



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO TÉCNICO EN INFORMÁTICA DE GESTIÓN

Título del proyecto:

DESARROLLO DE UN SISTEMA DE CONTROL DE ACCESOS A UN CENTRO

Héctor Napal Basterrica
Edurne Barrenechea Tartas
Pamplona, 24 de junio de 2010

*A mis padres y hermanos, por arroparme.
A mi compi, Patrizia, por estar siempre ahí.
A mi tutora, Edurne, por guiarme con paciencia.*

Prólogo

Ofrecemos el presente documento como la memoria de un Proyecto Final de Carrera para la titulación de Ingeniería Técnica en Informática de Gestión en la Universidad Pública de Navarra.

La elección de la empresa que hemos realizado no responde a un supuesto complejo o que busque el estudio de nuevas tecnologías, si no a optar por un proyecto de carácter cotidiano, que tenga una real demanda en el día a día y que a su vez nos permita manejar el mayor rango de conocimientos adquiridos a lo largo de nuestros años en la universidad, en especial todos los relacionados con la ingeniería del software, las bases de datos y la programación orientada a objetos.

En cuanto al carácter que hemos querido imprimir al proyecto, debemos decir que nuestra intención ha sido la de aunar el desarrollo técnico-formal y didáctico, representando así el formalismo que eximiríamos de cara al cliente y a la vez explicando, exponiendo y justificando todo el trabajo realizado, de modo que quien pueda llegar a utilizar esta memoria sea capaz de comprenderla a la perfección.

El desarrollo del proyecto atendiendo a los factores anteriormente citados ha dado lugar al presente escrito, que consta de dos tipos de documentos: los capítulos y los artefactos. En los primeros hemos descrito el esqueleto del desarrollo, mientras que con los segundos hemos acompañado a los primeros de extensas y clarificadoras explicaciones, así como de versiones simplificadas de todos aquellos documentos que tienen valor en el desarrollo de cualquier producto software.

Esperando que el lector disfrute de la lectura,

Héctor Napal Basterrica
Pamplona, 24 de junio de 2010.

Resumen

La presente memoria queda dividida en dos bloques claramente diferenciados, los cuales recogen la información de mayor relevancia extraída del proyecto software ejecutado y un anexo con todas las actividades realizadas en el proceso de desarrollo:

❖ Proyecto software:

En el capítulo 1 describimos de manera breve y precisa a la empresa para la que se realiza el proyecto que desarrollamos en esta memoria. A continuación aludimos a la situación en la que vive la empresa, su modo de funcionar día a día y qué factores determinan la necesidad de asumir un cambio en la gestión del control de accesos al centro. Dicha necesidad queda reflejada en una serie de objetivos y requerimientos que también describimos en esta sección.

En el siguiente capítulo hablamos de la elección de un proceso que guíe el desarrollo del producto software, eligiendo un responsable del proyecto, haciendo una recapitulación de la situación actual y qué pasos deben seguir.

En la sección número 3 recopilamos los principales artefactos generados durante el desarrollo software y que definen la esencia del producto elaborado. Aquí será donde el lector pueda contemplarlos ya completados y no parcialmente, como se muestran en el anexo adjunto.

Finalizamos este bloque incorporando dos capítulos en los que describimos las herramientas software empleadas en el desarrollo e instalación del sistema creado, así como un apartado bibliográfico con todos los recursos que hemos consultado.

❖ Anexo – Proceso de desarrollo:

Con el primer capítulo iniciamos la fase de elaboración, realizando un estudio detallado del problema que desemboca en el modelado de negocio y de los casos de uso, la definición de la arquitectura base, la planificación detallada del proyecto y un plan inicial de pruebas.

Las fases de planificación y diseño iniciales dan paso a la construcción del producto software, quedando reflejado todo el proceso en el segundo capítulo. En éste describimos cómo cada una de las iteraciones va dando lugar a sucesivas versiones del software, cada vez más refinadas, con mayor funcionalidad, comenzando con una línea arquitectónica básica hasta concluir con el producto software final.

A lo largo de la última sección describimos las iteraciones realizadas con vistas a realizar una óptima transferencia del producto elaborado, recogiendo las actividades de instalación, configuración, entrenamiento y futuro mantenimiento.

Índice

Prólogo

Resumen

Índice

Capítulo 1. Un antes y un después	1
1.1. Descripción del centro	1
1.2. Método actual de control de accesos al centro	1
1.3. Factores que promueven el desarrollo del sistema	1
1.4. Definición de objetivos, requerimientos y alcance general del proyecto	2
1.4.1. Objetivos y primeras determinaciones	
<i>Estudio inicial de la viabilidad del proyecto software</i>	3
Capítulo 2. El comienzo	9
2.1. Gestión del proyecto	9
2.1.1. El proceso de desarrollo	
2.1.2. El responsable del proyecto	
2.1.3. Las actividades de gestión	
2.2. Todo en orden	12
Capítulo 3. Principales artefactos	13
3.1. Introducción	13
3.2. Artefactos creados	13
<i>Especificación de los casos de uso</i>	14
<i>Planificación del proyecto</i>	148
<i>Esquema relacional de la base de datos</i>	154
<i>Arquitectura del sistema</i>	171
<i>Descripción de la API final</i>	181
<i>Plan de pruebas</i>	200
<i>Interfaz gráfica</i>	217
Capítulo 4. Herramientas	231
4.1. El entorno de desarrollo	231
4.2. Software ubicado en el sistema	237
Capítulo 5. Bibliografía	238
5.1. Libros y artículos de revistas	238
5.2. Sitios web	238

ANEXO – PROCESO DE DESARROLLO DE LA APLICACIÓN

Capítulo 1. Elaboración	1
1.1. Objetivos de la fase	1
1.2. Iteración E1: Estudio detallado del problema y modelado de negocio	2
1.3. Iteración E2: Captura de requisitos	2
1.4. Iteración E3: Definición de la arquitectura básica	2
1.5. Iteración E4: Planificación del proyecto	2
<i>Estudio detallado del problema software</i>	4
<i>Especificación del modelo de negocio</i>	13
<i>Especificación inicial de los casos de uso</i>	19
<i>Arquitectura preliminar</i>	64
<i>Planificación temporal y costes detallados del proyecto</i>	72
<i>Plan de pruebas inicial</i>	85
Capítulo 2. Construcción	94
2.1. Objetivos de la fase	94
2.2. Iteración C1: La base de datos	95
2.3. Iteración C2: API para la base de datos	95
2.4. Iteración C3 a C11: Implementación de los casos de uso	96
2.5. Iteración C12: Diseño, personalización y configuración	97
<i>Iteración C1: La base de datos</i>	98
<i>Iteración C2: Una API para la base de datos</i>	153
<i>Iteración C3: Accesos</i>	189
<i>Iteración C4: Empresas y personal</i>	244
<i>Iteración C5: Tarjetas, tipos de accesos y cargos</i>	358
<i>Iteración C6: Zonas, llaves y llaveros</i>	446
<i>Iteración C7: Préstamos</i>	584
<i>Iteración C8: Cuentas</i>	611
<i>Iteración C9: Partes de seguridad</i>	660
<i>Iteración C10: Notas</i>	703
<i>Iteración C11: Base de datos</i>	735
<i>Iteración C12: Diseño, personalización y configuración</i>	749
Capítulo 3. Transición	787
3.1. Objetivos de la fase	787
3.2. Iteración T1: Preparación de la release final	787
3.3. Iteración T2: Entrega	787
3.4. Iteración T3: Entrenamiento, soporte y mantenimiento	787
<i>Iteración T2: Entrega</i>	789

Capítulo 1

Un antes y un después

1.1. Descripción del centro

Informática Navarra S.A. es una sociedad pública creada por el Gobierno de Navarra dedicada a la producción, gestión y operación de equipamientos y aplicaciones informáticas, la prestación de servicios a usuarios, y la participación y colaboración con empresas del sector informático.

Fue constituida el pasado 7 de julio de 2008 por la sociedad pública Navarra Gestión de Administración S.A., y el hecho de que su capital sea público del Gobierno de Navarra exige que los estatutos y la creación de la sociedad reciban la conformidad del Ejecutivo, según la Ley foral que regula la Administración de la Comunidad Foral.

Tiene residencia en Pamplona, calle Salsipuedes s/n y como gerente a Jose Antonio Sánchez.

1.2. Método actual de control de accesos al centro

En la actualidad acceden al centro una gran variedad de personal perteneciente a empresas de todo ámbito para desarrollar su labor profesional.

El centro posee información y equipamiento de alta importancia y coste, lo que conlleva que sea necesario llevar un control de todo aquel que accede a él. Debido a la inexistencia de un sistema informático, este procedimiento es llevado a cabo de una forma rudimentaria, lenta y poco resolutive si se tiene en cuenta las amplias posibilidades que ofrece hoy en día el mundo de la informática. Todo el personal es identificado y registrado a través de documentos en papel, lo que impide totalmente llevar un control estricto, claro y sencillo de cualquier dato concerniente a los accesos si no es de forma manual.

1.3. Factores que promueven el desarrollo del sistema

Los responsables del control de accesos son conscientes del grado de inoperatividad del mismo, su falta de privacidad y la cantidad de limitaciones que tiene. Están dispuestos a mejorarlo, implantando un sistema informático que ayude a realizar dicha tarea, pero no pueden dedicar a ello una parte del tiempo de los empleados del centro, ni tampoco quieren hacer una inversión de alto coste en software y hardware.

Tras reunirse deciden dar un paso adelante y contratar un equipo informático con la finalidad de automatizar todo el proceso de control de accesos al centro, así como el tratamiento de toda la información concerniente a éstos. Además de estas premisas, se añaden otras que son mencionadas con detalle en el siguiente apartado.

1.4. Definición de objetivos, requerimientos y alcance general del proyecto

Son muchas y variadas las posibilidades que pueden dar solución al problema planteado. Tras una primera entrevista llevada a cabo entre Javier González Tablas (director general de Software Solutions), Juan Hernández Córdoba (analista de viabilidad y factibilidad de la empresa citada), Jose Antonio Sánchez (gerente de 'Informática Navarra S.A.') y Pedro Martínez Pérez (jefe de área de producción de la misma) se extraen los siguientes objetivos iniciales.

1.4.1. Objetivos y primeras determinaciones

Como en todo proyecto informático, se proponen los objetivos esenciales de resultado, tiempo, coste, calidad y satisfacción del cliente. Además de éstos, que englobarían el fin último del desarrollo, tenemos que señalar los principales cometidos:

Analizar, diseñar, implementar, implantar, configurar y mantener todo el software necesario que provea las siguientes funcionalidades principales:

- *Control de accesos al centro, con el consecuente tratamiento de ...*
 - *Empresas*
 - *Departamentos*
 - *Personal*
 - *Existencia y prestación de tarjetas, llaves y llaveros disponibles, prestados,...*
 - *Accesos realizados*
 - *...*
- *Redacción, visualización y almacenamiento de los partes de vigilancia del centro.*
- *Mensajería entre usuarios.*

El sistema deberá ser multiusuario, restringiendo la funcionalidad y acceso (medio y grado) a la información en función del tipo de usuario que utilice el sistema.

A raíz de esta primera reunión se pasa a acometer una evaluación general del problema planteado, realizando un estudio inicial de la viabilidad que tendrá el desarrollo del sistema, junto con sus costes y beneficios. Dicho estudio se puede observar a continuación y deberá ser aprobado por parte del cliente para que el desarrollo que documentamos en la presente memoria tenga lugar.



accessManager

Desarrollo de un sistema de control de accesos a un centro

Estudio inicial de la viabilidad del proyecto software

Fecha de entrega: 15 de enero de 2010
Editor: Juan Hernández Córdoba



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Estructura del documento

2. Vista general del proyecto

- 2.1. Deficiencias actuales
- 2.2. Restricciones
- 2.3. Metas y objetivos

3. Desarrollo del proyecto

- 3.1. Proceso de desarrollo
- 3.2. Personal involucrado
- 3.3. Planificación temporal
- 3.4. Estimación económica

4. Conclusiones

5. Aprobación y firmas



1. Introducción

1.1. Propósito

Hemos desarrollado este documento con objeto de estudiar la viabilidad del sistema que daría respuesta al problema planteado, realizando un estudio inicial antes de proceder a un análisis en profundidad del mismo. Además, aportamos una aproximación de los costes temporales y monetarios a los que éste dará lugar.

Para finalizar, y como fin para el que hemos elaborado el presente escrito, el cliente deberá realizar una evaluación de todos los datos que aquí aportamos, dando su aprobación para que iniciemos el desarrollo del proyecto o cancelándolo por no resultar viable o atractivo.

1.2. Audiencia

El proyecto ha sido ofertado por Jose Antonio Sánchez, gerente de Informática Navarra S.A., quien ha delegado toda la responsabilidad del desarrollo del mismo en Pedro Martínez Pérez (jefe de área de producción de la misma), que será con quien el equipo de desarrollo mantendrá contacto y el destinatario del presente estudio.

1.3. Estructura del documento

En la sección 2 realizamos una descripción global del problema software, restricciones y objetivos del sistema a desarrollar.

A continuación presentamos el proceso de desarrollo que se empleará, así como una aproximación de los recursos económico-temporales necesarios y el personal involucrado.

Damos fin a este documento con un breve apartado de conclusiones y la aprobación y firma por parte del cliente y del jefe de proyecto.

2. Vista general del proyecto

Informática Navarra S.A. desea adquirir un producto software con el principal fin de poder llevar a cabo el control de accesos a su sede, así como el manejo de toda la información necesaria para tal fin. A estos objetivos, se suman los secundarios, que constarían en poder gestionar la información del servicio de vigilancia y fuese una vía de comunicación entre usuarios. El soporte a desarrollar será el de un sistema multiusuario accesible a través de una aplicación interactiva.



2.1. Deficiencias actuales

Dada la inexistencia de cualquier tipo de sistema previo, todas las funcionalidades han de ser incorporadas como nuevas en el sistema:

- Acceso y tratamiento multiusuario de todo tipo de elementos relacionados con el control de accesos: personal, accesos, tarjetas, llaves,...
- Redacción, visualización y almacenamiento de los partes de vigilancia del centro.
- Mensajería entre usuarios.

2.2. Restricciones

2.1.1. Del proyecto

- El cliente no desea realizar una inversión de alto coste, con que deberá proporcionarse una solución de coste económico mínimo bajo unos umbrales aceptables de calidad.
- En cuanto a los recursos temporales, el desarrollo no requiere de presteza alguna.

2.1.2. De la aplicación

- Se realizará la compra de todo el equipo hardware necesario para poder implantar la solución.
- El número y concurrencia de usuarios será de tipo bajo, así como el volumen de datos manejados.

2.3. Metas y objetivos

Dar respuesta a las funcionalidades descritas anteriormente, que serán citadas con todo detalle en el documento de *Especificación inicial de los casos de uso*, de una forma económica, eficaz y fiable.

3. Desarrollo del proyecto

3.1. Proceso de desarrollo

Las características del problema junto con las restricciones iniciales impuestas determinan que se opte por el modelo de desarrollo orientado a objetos. La metodología RUP, con su contrastada eficiencia, será la que guíe todo el proceso de desarrollo.

3.2. Personal involucrado

Dada la envergadura del proyecto y la inexistencia de requisitos temporales, se opta por que trabaje una sola persona en el proyecto, realizando ésta todas las tareas de gestión, supervisión y desarrollo. Como único desarrollador, se asignará el trabajo a una persona experimentada en este tipo de proyectos, con la experiencia necesaria para dar respuesta a las demandas del cliente.



3.3. Planificación temporal

El ciclo de vida RUP presenta cuatro fases divididas en iteraciones, cada una de las cuales desarrollará en menor o mayor medida las disciplinas o flujos de trabajo fundamentales características.

Se estima la siguiente distribución de tiempos:

FASE	DURACIÓN
Estudio de oportunidad (inception)	1'5 semanas
Elaboración	3 semanas
Construcción	7'5 semanas
Transición	1'5 semanas
TOTAL	13'5 semanas

Estimación temporal de las fases RUP

3.4. Estimación económica

Las características del proyecto nos hacen prever los siguientes costes:

MANO DE OBRA			
CARGO	TIEMPO	SALARIO / SEMANA	TOTAL
Director del proyecto. Ingeniero de desarrollo y soporte	13'5 semanas	375 €	5063 €
MATERIA PRIMA			
MATERIA	CANTIDAD	VALOR / 5 AÑOS	TOTAL
PC's ambiente de desarrollo	1	499 €	26,87 €
Licencias software	1	4008,65 €	208,14 €
Servidor	1	450 €	450 €
TOTAL			5748,01 €

Costes estimados derivados del desarrollo del proyecto

4. Conclusiones

La empresa encargada resulta ser, en un principio, un proyecto convencional en la sociedad que nos concierne, no requiriendo una avanzada tecnología y que podrá ser desarrollado perfectamente bajo las condiciones iniciales impuestas por el cliente.

Desde el punto de vista coste-beneficio, resulta ser un sistema esencial en un centro de tales características y una inversión muy atractiva para los intereses de la empresa y las ventajas que proporciona.



5. Aprobaciones y firmas

Con las siguientes firmas queda aprobado *Estudio inicial de la viabilidad del proyecto software* por parte del cliente, constituyendo el inicio del desarrollo software del proyecto.

Pamplona, a 15 de enero de 2010.

Juan Hernández Córdoba
Analista de viabilidad y factibilidad

Jose Antonio Sánchez
Gerente de Informática Navarra S.A.

Capítulo 2

El comienzo

2.1. Gestión del proyecto

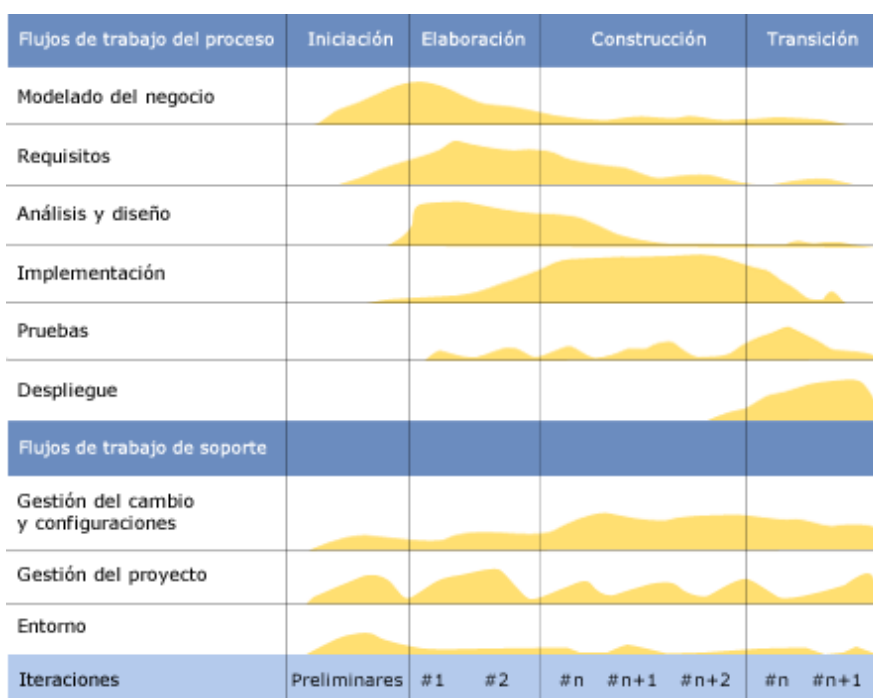
La aceptación de *Estudio inicial de la viabilidad del proyecto software* por parte del cliente da el pistoletazo de salida al desarrollo del producto software.

Todo problema con un mínimo grado de complejidad debe comenzar con una cuidada y meticulosa planificación del mismo, realizando todas las tareas de gestión previas y a las cuales dedicamos el presente capítulo.

2.1.1. El proceso de desarrollo

No son poco los modelos de desarrollo de software que han surgido a lo largo de la historia. En la empresa que nos atañe, una aplicación multiusuario, de complejidad media e interactiva, resulta de gran provecho optar por el *Modelo orientado a objetos*. Se trata de una tecnología en alza, que proporciona muchas y diversas ventajas, tantas que nos harían falta varias hojas de esta memoria para enumerarlas.

Una vez enfocados en la orientación a objetos, nos falta precisar. Metodologías que implementen este modelo son muchas, aunque una de ellas destaca por encima del resto: *RUP*. El *Rational Unified Process*, ideado por la compañía Rational (IBM) e hijo del *Proceso Unificado*, constituye el proceso de desarrollo orientado a objetos por antonomasia a día de hoy, incorporando un conjunto de *Best Practices* que no pueden faltar en ningún proyecto de carácter orientado a objetos. Consta de la siguiente distribución de fases y actividades iteradas:



Entre sus características esenciales debemos mencionar las de desarrollo iterativo e incremental' y control/dirección a través de los casos de uso, que nos permitirán:

- Afrontar los riesgos y dificultades de forma fraccionada, en iteraciones:
 Resulta obvio que es mucho más sencillo poder identificar los riesgos, así como definir las estrategias de contingencia para éstos, si afrontamos el desarrollo progresivamente, por partes, y no todo de golpe. Lo mismo ocurre si de lo que se trata es de desarrollar el sistema: dividimos la funcionalidad total en partes desarrolladas una a una, siendo la suma de las tareas individuales menor que la de la tarea en su totalidad (principio cesáreo de *Divide y vencerás*).
- Disponer de una verificación, evaluación y validación continuada por parte del cliente:
 El proceso iterativo permite obtener un conocimiento incremental de los requisitos, aumentar la calidad del producto mediante pruebas de verificación y efectuar una validación y desarrollo cooperativo constante con el cliente. Todo ello implica que la posibilidad de obtener un producto no válido será mínima, además que nos permitirá conocer el estado del proyecto en cada momento.
- Tener un control total y sencillo de todo el proceso a través de los casos de uso:
 Se define la funcionalidad del sistema a través de los casos de uso, que deberán ser implementados y probados.

El objeto de esta sección no es el de realizar una descripción en detalle de *RUP*, con que dejaremos aquí este breve inciso. No obstante, debe quedar claro que la elección de este proceso de desarrollo está completamente justificada y sobrada de motivos y ventajas.

2.1.2. El responsable del proyecto

En *Estudio inicial de la viabilidad del proyecto software* hicimos una escueta estimación de los recursos humanos destinados al proyecto. Recordemos que el grado de complejidad del sistema es medio, sin requisitos temporales, lo que hará que no se requiera más que un desarrollador experimentado en bases de datos y orientación a objetos. Tras reunirse los directivos de Software Solutions deciden asignar la dirección y desarrollo del proyecto a Héctor Napal Basterrica.

2.1.3. Las actividades de gestión

Dependiendo de la dimensión y grado de complejidad del sistema, el jefe del proyecto debe formalizar en diferente número y medida las actividades de gestión. En la siguiente página enumeramos las que creemos esenciales para esta empresa.

ACTIVIDAD	DESARROLLO
Asignación de recursos humanos	Todo el desarrollo es asignado a una única persona.
Asignación de recursos temporales	Remitimos a la calendarización detallada en el artefacto <i>Planificación temporal del proyecto</i> encuadrado en el siguiente capítulo.
Asignación de tareas	Todas las tareas son realizadas por el único trabajador involucrado en el proyecto.
Gestión y supervisión del proyecto	Constante, guiado por los casos de uso y asegurando en todo momento la máxima calidad y el mínimo riesgo.
Gestión de requisitos y cambios	Consiste en descubrir, organizar y documentar las necesidades reales que exigen los clientes al desarrollar el sistema, así como mantener un acuerdo continuo sobre los requisitos que cambian. Con ello conseguimos que los requisitos puedan tener una prioridad y seguimiento, siendo sencilla la detección de cualquier tipo de inconsistencia. Ésta actividad es de vital importancia y será realizada de forma continuada, comenzando con un modelado inicial de los casos de uso que será refinado y completado a lo largo de todo el desarrollo. <i>Véase Especificación inicial de los casos de uso, adjuntado tras el tercer capítulo de esta memoria.</i>
Control de calidad	Durante todo el proceso, llevando a cabo actividades de control de calidad que la aseguren. El resultado es un documento que planifica y recoge los resultados de todas las pruebas realizadas: <i>'Plan de pruebas'</i> .
Gestión de riesgos	La gestión de riesgos es un aspecto a tener muy en cuenta en casi cualquier proyecto de desarrollo software, ya que afecta a la agenda, recursos, calidad,... Esta actividad supone confeccionar un documento de <i>Plan de riesgos</i> en el que hacer un estudio pormenorizado de todos los posibles riesgos y las acciones establecidas como estrategias para mitigarlos (acciones de contingencia). En el caso que nos ocupa haremos una pequeña excepción debido al tipo y complejidad de proyecto que acometemos. Más adelante realizaremos un análisis en profundidad del problema, pero podemos avanzar que carece de sentido el pasar de un breve análisis de riesgos en cada iteración, y ésta será la línea que tomemos.
Gestión de la configuración	La necesaria para llevar a cabo la integración y gestión de versiones.

Actividades de gestión

2.2. Todo en orden

Llegados a este punto cabe recapitular y determinar en qué punto nos encontramos. Ateniéndonos a las fases *RUP*, podemos decir que hemos consumido la de Estudio de oportunidad (inception). Nos esperan las de Elaboración, Construcción y Transición, que se dividirán en iteraciones que desarrollarán las Disciplinas o flujos de trabajo fundamentales en menor o mayor medida, dando lugar a versiones cada vez más completas y refinadas del sistema, hasta conseguir el deseado. Hemos dedicado a cada una de ellas un capítulo del presente documento, de forma que se pueda seguir y entender todo el proceso de desarrollo, así como todo lo que conlleva, con el menor esfuerzo posible.

Capítulo 3

Principales artefactos

3.1. Introducción

RUP es un proceso de desarrollo software que genera artefactos de forma muy temprana y los va refinando iterativamente hasta que el ciclo de desarrollo finaliza. Con el fin de recoger y mostrar al lector el estado/resultado final de los documentos más representativos creamos esta sección.

3.2. Artefactos creados

Son varios los artefactos que podrían tener cabida tras esta página, pero hemos decidido recoger los siguientes, pues creemos que constituyen la esencia del sistema creado:

- **Especificación de los casos de uso**
Recoge una especificación completa y precisa de todos los casos de uso definidos a lo largo del proceso de desarrollo.
- **Planificación del proyecto**
Resumen de la planificación adoptada y seguida durante la fabricación del producto software.
- **Esquema relacional de la base de datos**
Define el modelo relacional que dará soporte a todo el modelo de datos que es necesario que accessManager maneje.
- **Arquitectura del sistema**
Muestra las estructuras arquitectónica y software que constituyen el sistema y que proporcionan un medio a través de la cual se resuelve el problema planteado.
- **Descripción de la API final**
Breve estudio de la API resultante tras haber completado todas las iteraciones de la etapa de Construcción y que será la encargada de proporcionar toda la funcionalidad que la interfaz gráfica de la aplicación necesite.
- **Plan de pruebas**
Recopila todas las pruebas realizadas a lo largo del proceso de desarrollo, demostrando que el producto creado implementa todos los casos de uso descritos en la especificación de los casos de uso bajo unos niveles aceptables de calidad.
- **Interfaz gráfica**
Exposición de la composición gráfica de la aplicación desarrollada, en la que ofrecemos un breve recorrido por todas las pestañas que la componen y cómo ha sido distribuida en éstas la funcionalidad que proporciona.



accessManager

Desarrollo de un sistema de control de accesos a un centro

Especificación de los casos de uso

Fecha de entrega: 16 de abril de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Estructura del documento

2. Actores del sistema

3. Descripciones de trazo grueso

4. Descripciones de trazo fino

- 4.1. Cuentas – Parte 1
- 4.2. Accesos
- 4.3. Empresas
- 4.4. Departamentos
- 4.5. Personal
- 4.6. Niveles
- 4.7. Colores
- 4.8. Tarjetas
- 4.9. Tipos de acceso
- 4.10. Cargos
- 4.11. Zonas del centro
- 4.12. Llaves
- 4.13. Llaveros
- 4.14. Almacenes de llaves
- 4.15. Préstamos pendientes
- 4.16. Cuentas del sistema
- 4.17. Tipos de usuario
- 4.18. Historial de accesos al sistema
- 4.19. Cuentas – Parte 2
- 4.20. Recordatorios y borrado
- 4.21. Partes de seguridad
- 4.22. Notas
- 4.23. Base de datos

5. Aprobaciones y firmas



1. Introducción

1.1. Propósito

Este documento nace con el propósito de recoger la información más relevante concerniente a la especificación de los casos que ha sido redactada a lo largo de todo el proceso de desarrollo.

1.2. Audiencia

La especificación de los casos de uso es el principal documento por el cual cliente y equipo de desarrollo pueden acordar qué características debe poseer el sistema a desarrollar.

Tras la definición inicial que realizamos en el capítulo 3, este documento ha ido creciendo y refinándose con cada una de las iteraciones realizadas hasta acabar siendo el que el lector posee en sus manos. Una vez concluido, el cliente obtiene un artefacto que define a la perfección toda la funcionalidad del producto software adquirido.

1.3. Estructura del documento

Comenzamos la redacción del documento identificando todos los tipos de usuarios diferentes que tiene el sistema, definiendo roles y para qué usan el mismo.

Una vez tenemos claros cuáles van a ser los actores, en la siguiente sección, la tercera, enumeramos los casos de uso en los que participarán, creando un conjunto de descripciones generales de éstos.

En el capítulo 4 recogemos las descripciones de trazo fino a las que han derivado las de trazo grueso y finalizamos el presente escrito con las firmas del cliente y el responsable del desarrollo.



2. Actores del sistema

Delimitamos sistema y entorno estableciendo los siguientes actores:

CAMPO	DESCRIPCIÓN
Nombre	Administrador
Descripción	Administrador del sistema.
Principales uso del sistema	<ul style="list-style-type: none">Realizar tareas de mantenimiento de la información, haciendo que ésta siempre sea veraz.Obtener toda la información necesaria para realizar su labor.Comunicarse con los demás usuarios del sistema.
Nombre	Vigilante
Descripción	Vigilante de seguridad del centro.
Principales uso del sistema	<ul style="list-style-type: none">Llevar un control eficiente de los accesos realizados al centro y de los préstamos que conllevan.Redactar los partes de seguridad de los servicios que realiza.Comunicarse con los demás usuarios del sistema.
Nombre	Autorizador
Descripción	Trabajador de Informática Navarra S.A. que ayuda al Administrador en su labor de mantenimiento del sistema, así como de hacerse cargo de parte del personal que acceda al centro.
Principales uso del sistema	<ul style="list-style-type: none">Gestionar todo tipo de información referente a los departamentos de los que se es responsable.Comunicarse con los demás usuarios del sistema.
Nombre	RP
Descripción	Responsable de personal de un departamento de una empresa.
Principales uso del sistema	Mantener actualizada continuamente la lista del personal propio que accede al centro.

Actores del sistema



3. Descripciones de trazo grueso

En las siguientes páginas proporcionamos las descripciones de trazo grueso generadas para los principales casos de uso y segmentadas con respecto a las diversas entidades reales existentes en el entorno del problema. Dicha división da lugar a grupos de casos de uso muy similares, constituyendo una buena aproximación a cómo dividiremos y acometeremos el trabajo en las siguientes iteraciones. La siguiente tabla resume éstos grupos y la prioridad que les hemos asignado:

CASOS DE USO			
GRUPO			PRIORIDAD
Cuentas			Imprescindible e importante
Accesos	Empresas	Departamentos	Imprescindible
Personal	Niveles	Colores	
Tarjetas	Tipos de acceso	Cargos	
Zonas del centro	Llaves	Llaveros	
Almacenes de llaves	Préstamos pendientes	Cuentas del sistema	
Tipos de usuario	Historial de accesos al sistema		
Recordatorios y borrado		Partes de seguridad	Importante
Notas		Base de datos	Deseable

Sin más preámbulos, exponemos a partir de la siguiente hoja las descripciones de trazo grueso creadas.

CUENTAS	
CAMPO	DESCRIPCIÓN
Caso de uso	1 - Login
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador, Vigilante y RP
Funcionalidad	El usuario accede al sistema, hecho que provoca que deba identificarse. El sistema le pide los datos y tras verificarlos despliega la aplicación. En caso de no existir el usuario, se le notifica el error y se permanece en la pantalla de Login.
Caso de uso	116 - Modificar datos de cuenta
Prioridad	Importante
Actor(es)	Administrador, Autorizador, Vigilante y RP
Funcionalidad	El usuario solicita cambiar alguno de sus datos de acceso (nombre o contraseña). El sistema le pide que introduzca sus datos de identificación actuales y los nuevos que desea utilizar. Tras comprobar que los primeros son correctos se realiza la actualización, en caso contrario se indica el error ocurrido y se permanece en la pantalla de Modificar datos de usuario.

Especificaciones de trazo grueso – Cuentas

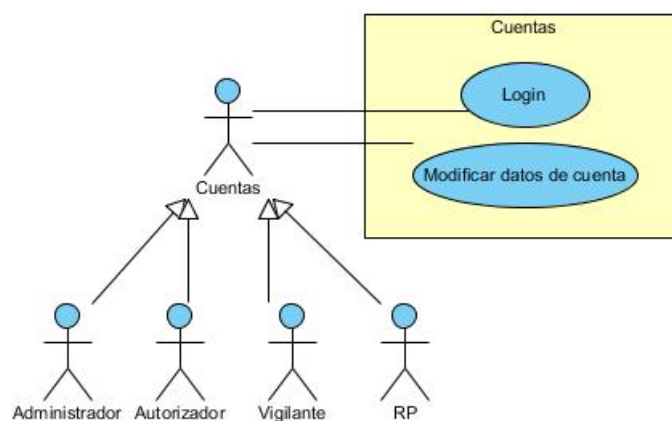


Diagrama de casos de uso – Cuenta



ACCESOS	
CAMPO	DESCRIPCIÓN
Caso de uso	2 - Visualizar accesos
Prioridad	Imprescindible
Actor(es)	Administrador y Vigilante
Funcionalidad	El usuario desea conocer los accesos realizados a la sede que cumplan unas condiciones concretas. Indica dicha solicitud al sistema, a lo que éste responde mostrándole el conjunto de accesos que cumplan con dichos requisitos.
Caso de uso	3 - Visualizar acceso
Prioridad	Imprescindible
Actor(es)	Administrador y Vigilante
Funcionalidad	El administrador o vigilante desea conocer toda la información referente a un acceso realizado al centro. Lo identifica, encargándose el sistema de mostrarle todos los datos de los que se compone.
Caso de uso	4 - Dar entrada
Prioridad	Imprescindible
Actor(es)	Administrador y Vigilante
Funcionalidad	El usuario pretende registrar el acceso al centro de una persona. El sistema le pide los datos, buscándola y determinando qué tarjeta le debe ser entregada. En caso oportuno se añaden las llaves y/o llaveros que se le prestan y se valida el acceso, quedando registrado.
Caso de uso	5 - Prestar llaves y/o llaveros
Prioridad	Imprescindible
Actor(es)	Administrador y Vigilante
Funcionalidad	Una persona que va a acceder o está ya en el centro necesita de llaves y/o llaveros para poder realizar sus labores. El usuario le presta lo que necesite, proporcionando al sistema persona y llaves prestadas, encargándose éste de grabar el préstamo.
Caso de uso	6 - Borrar acceso
Prioridad	Imprescindible
Actor(es)	Administrador y Vigilante
Funcionalidad	Es posible realizar el borrado de un acceso tras haberlo identificado.
Caso de uso	7 - Modificar acceso
Prioridad	Imprescindible
Actor(es)	Administrador y Vigilante
Funcionalidad	El usuario desea modificar algún dato de uno de los accesos registrados. Lo identifica y modifica a su gusto, aprobando que es correcto.
Caso de uso	8 - Borrar préstamo
Prioridad	Imprescindible
Actor(es)	Administrador y Vigilante
Funcionalidad	Se ha realizado el préstamo de una llave y/o llavero que no debiera haberse hecho. Para subsanar este hecho, el usuario selecciona el préstamo erróneo y le pide al sistema que lo elimine.

Especificaciones de trazo grueso – Accesos



Caso de uso	9 - Dar salida
Prioridad	Imprescindible
Actor(es)	Administrador y Vigilante
Funcionalidad	Se desea registrar que una persona se dispone a abandonar el centro. El sistema proporciona los datos del acceso y el usuario registra la salida indicando al sistema todos los elementos en préstamo devueltos.
Caso de uso	10 - Devolver llaves y/o llaveros
Prioridad	Imprescindible
Actor(es)	Administrador y Vigilante
Funcionalidad	Un trabajador desea devolver al vigilante parte o totalidad de las llaves y llaveros que le habían sido prestadas. El vigilante comunica al sistema qué le ha sido devuelto, identificando el prestamista y dando por cerrado el préstamo que se había iniciado con anterioridad.
Caso de uso	11 - Listar accesos
Prioridad	Imprescindible
Actor(es)	Administrador y Vigilante
Funcionalidad	Se desea obtener un listado con todos los accesos realizados al centro dependiendo de algún tipo de parámetro. El usuario acota la búsqueda y pide al sistema que le proporcione un listado con todos los accesos que la satisfagan.
Caso de uso	12 - Dar salida automática
Prioridad	Imprescindible
Actor(es)	Sistema
Funcionalidad	El sistema otorgará de salida a todos aquellos accesos del día anterior que careciesen de ella, estableciéndose la máxima en la que cada persona pudiese estar en el centro.

Especificaciones de trazo grueso – Accesos

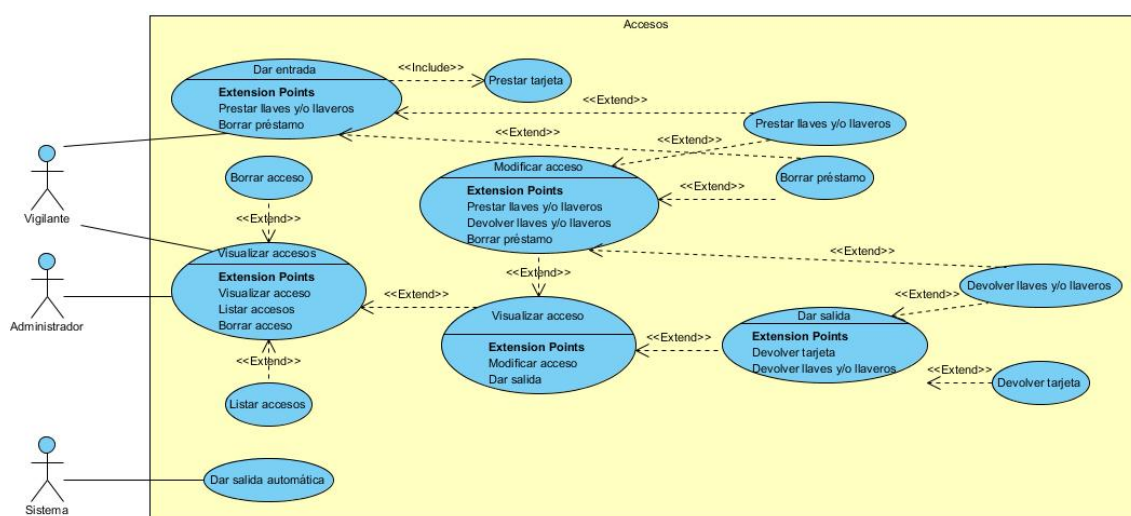


Diagrama de casos de uso – Accesos



EMPRESAS	
CAMPO	DESCRIPCIÓN
Caso de uso	13 - Visualizar empresas
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	El usuario puede visualizar de forma rápida y sencilla un listado con todas o parte de las empresas que trabajan a día de hoy en el centro.
Caso de uso	14 - Visualizar empresa
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	Se quiere visualizar toda la información relativa a una empresa. Tras identificarla el usuario pide al sistema que realice tal función, ofreciéndole éste toda la información de la que dispone.
Caso de uso	15 - Añadir empresa
Prioridad	Imprescindible
Actor(es)	Administrador y Autorizador
Funcionalidad	Una nueva empresa entra a formar parte de la “familia” de Informática Navarra S.A. El usuario le solicita al sistema darla de alta, pidiéndole éste todos los datos necesarios. Tras proporcionarlos y aprobar que son correctos, el sistema registra el alta.
Caso de uso	16 - Borrar empresa
Prioridad	Imprescindible
Actor(es)	Administrador y Autorizador
Funcionalidad	Una empresa deja de desarrollar sus labores en el centro. Por ello, el usuario desea darla de baja, identificándola y procediendo el sistema a su eliminación.
Caso de uso	17 - Modificar empresa
Prioridad	Imprescindible
Actor(es)	Administrador y Autorizador
Funcionalidad	El administrador o autorizador solicita al sistema modificar alguno de los datos que conforman el ente empresa en el sistema. Tras proporcionar los nuevos datos el sistema procede con la actualización de ésta.
Caso de uso	18 - Listar empresas
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	El usuario se dispone a obtener un listado de empresas del centro. Introduce los datos que cree conveniente para restringir la búsqueda y el sistema le proporciona el resultado deseado.

Especificaciones de trazo grueso – Empresas

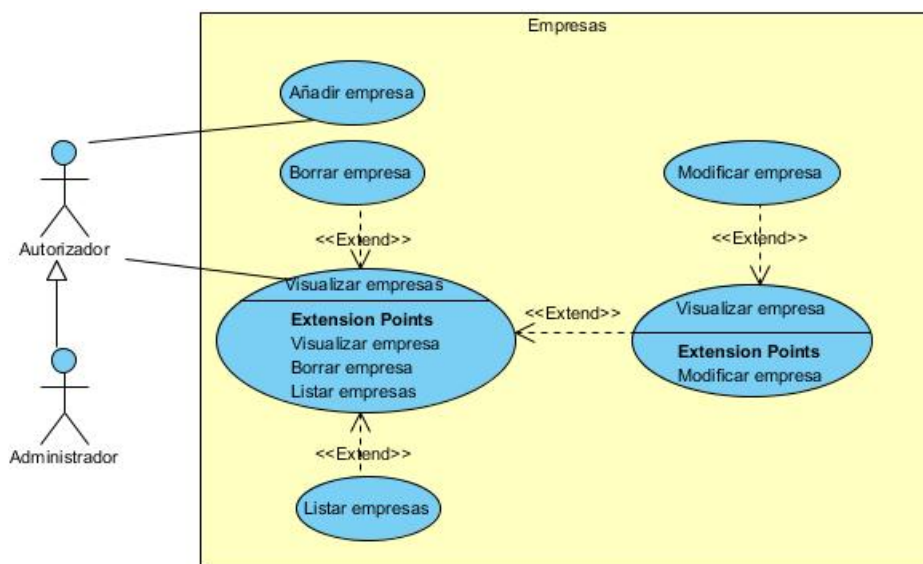


Diagrama de casos de uso – Empresas



DEPARTAMENTOS	
CAMPO	DESCRIPCIÓN
Caso de uso	19 - Visualizar departamentos
Prioridad	Imprescindible
Actor(es)	Administrador y Autorizador
Funcionalidad	El sistema provee la posibilidad de visualizar el conjunto de departamentos existentes en él y que cumplen con unas características determinadas; características que serán proporcionadas por el usuario y empleadas por la aplicación para realizar la búsqueda y mostrar los resultados.
Caso de uso	20 - Visualizar departamento
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	Se realiza una solicitud de visualización de toda la información de un departamento dado. El sistema obtiene toda la información y la ofrece la usuario.
Caso de uso	21 - Añadir departamento
Prioridad	Imprescindible
Actor(es)	Administrador y Autorizador
Funcionalidad	El usuario desea añadir a alguna de las empresas del centro un departamento nuevo que entra en escena en la vida diaria del centro. Lo solicita al sistema y éste le pide todos los datos necesarios. Tras proporcionárselos, un nuevo departamento es añadido a los ya existentes.
Caso de uso	22 - Borrar departamento
Prioridad	Imprescindible
Actor(es)	Administrador y Autorizador
Funcionalidad	El administrador o autorizador quiere borrar un departamento que consta en el sistema. Tras identificarlo, solicita al sistema que lo elimine, llevándose la tarea a buen puerto.
Caso de uso	23 - Modificar departamento
Prioridad	Imprescindible
Actor(es)	Administrador y Autorizador
Funcionalidad	Un departamento consta con datos erróneos, pretendiendo el usuario modificarlos. Tal solicitud hace que el sistema despliegue todos los datos que posee de éste, editando el usuario los datos deseados y afirmando que son correctos. La actualización tiene lugar de forma satisfactoria.
Caso de uso	24 - Listar departamentos
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	Se desea obtener un listado de departamentos con personal en el centro. El usuario acompaña dicha solicitud de todas las características que deban poseer éstos, elaborando el sistema el listado solicitado.

Especificaciones de trazo grueso – Departamentos

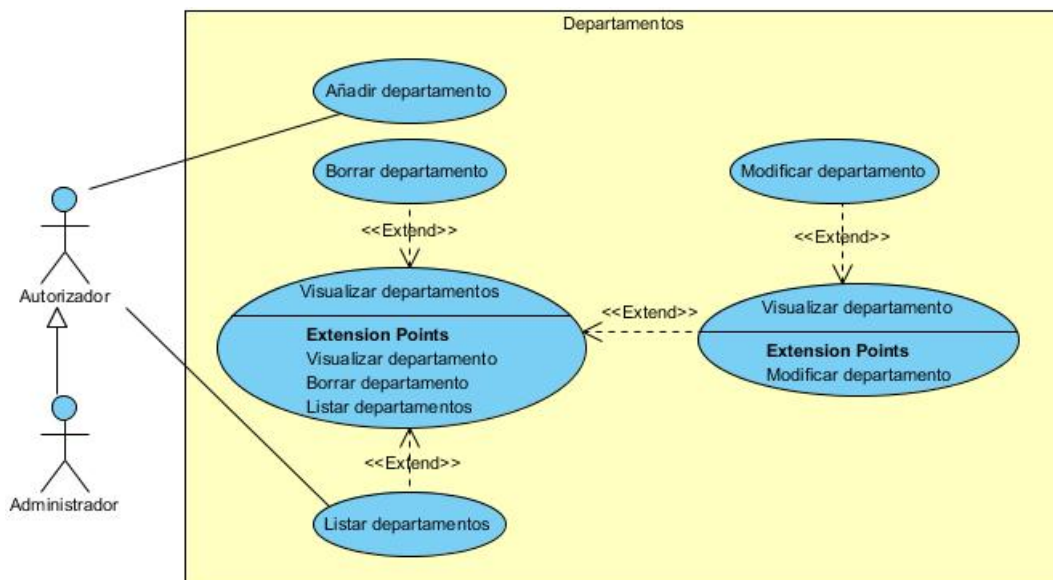


Diagrama de casos de uso – Departamentos



PERSONAL	
CAMPO	DESCRIPCIÓN
Caso de uso	25 - Visualizar personal
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	Será posible visualizar todo el personal que trabaja en el centro y acotarlo en función de diversos parámetros de búsqueda.
Caso de uso	26 - Visualizar persona
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	Surge la necesidad de acceder a los datos completos de una persona. El usuario la identifica y el sistema se encarga de obtenerlos y mostrárselos.
Caso de uso	27 - Añadir persona
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	El usuario desea dar de alta a una persona. Para ello proporciona al sistema todos los datos que éste le requiere, comprobando que son correctos y validando el alta.
Caso de uso	28 - Borrar persona
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	Se desea dar de baja a una persona. Se la identifica y se indica al sistema que proceda a su eliminación, dejando de formar parte del personal del centro.
Caso de uso	29 - Modificar persona
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	El usuario se percata de que algún dato perteneciente a una persona es erróneo, transmitiendo al sistema su deseo de modificarlo. Identifica al sujeto y corrige los datos que crea oportunos, haciendo que tenga en este momento todos sus datos descritos correctamente.
Caso de uso	30 - Añadir teléfono
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	El usuario identifica a una persona y solicita al sistema añadirle un número de teléfono. Lo teclea, pasando a ser almacenado con el resto de números.
Caso de uso	31 - Borrar teléfono
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	Una persona del centro deja de utilizar un número de teléfono, lo que lleva a su borrado ya que carece de sentido seguir almacenándolo. Se identifica qué número de teléfono entre todos los que posea debe ser borrado, llevando a cabo el sistema dicha operación.

Especificaciones de trazo grueso – Personal



Caso de uso	32 - Añadir E-Mail
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	Una persona pasa a tener una nueva cuenta de correo electrónico, que es registrada por el usuario mediante la identificación de la persona y el alta de la nueva dirección.
Caso de uso	33 - Borrar E-Mail
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	Por la razón que sea, una persona no desea recibir más comunicaciones a través de una cuenta de correo. El usuario solicita al sistema darla de baja, identificando persona y cuenta de correo. El sistema se encarga de eliminarla y que así no pueda ser utilizada en el futuro.
Caso de uso	34 - Añadir cargo
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	Un contacto de un departamento pasa a desarrollar un nuevo cargo, deseando el usuario que conste en el sistema. Para ello, identifica al contacto y solicita añadirle un nuevo cargo. El sistema le proporciona los cargos existentes, el usuario elige el que desee y éste queda adherido al contacto.
Caso de uso	35 - Borrar cargo
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	El usuario identifica un contacto y, tras seleccionar uno de los cargos que desempeña, solicita que sea borrado. El sistema le pide que apruebe el borrado, dejando de existir en el sistema en caso afirmativo.
Caso de uso	36 - Listar personal
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	El usuario desea obtener un listado de personas que posean unas características determinadas. Para ello, define las características del listado y el sistema lo elabora con los resultados obtenidos.

Especificaciones de trazo grueso – Personal

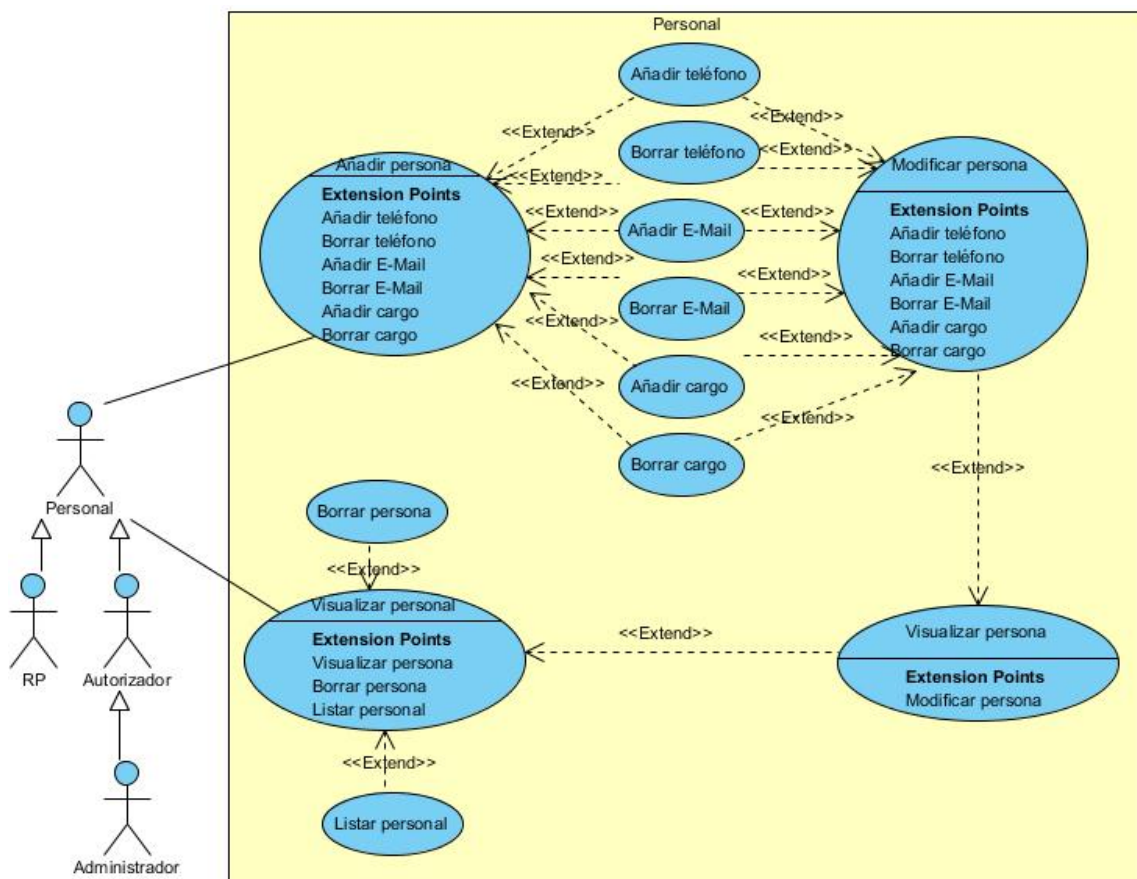


Diagrama de casos de uso – Personal



NIVELES	
CAMPO	DESCRIPCIÓN
Caso de uso	37 - Visualizar niveles
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador podrá visualizar todos los niveles existentes en el sistema.
Caso de uso	38 - Visualizar nivel
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Visualizar un nivel es tan fácil como identificarlo e indicar al sistema que muestre todos sus detalles.
Caso de uso	39 - Añadir nivel
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario solicita añadir un nuevo nivel a los ya existentes. El sistema le ofrece un formulario que rellena y valida, consiguiendo así su meta.
Caso de uso	40 - Borrar nivel
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El sistema permite borrar un nivel. Para conseguirlo, el usuario lo identifica y pide al sistema que lo elimine, hecho que realiza tras una breve aprobación por parte del usuario.
Caso de uso	41 - Modificar nivel
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Para modificar un nivel existente el administrador deberá identificarlo. Tras introducir los cambios que crea convenientes en los datos proporcionados por el sistema, el nivel queda grabado correctamente.
Caso de uso	42 – Dar de baja nivel
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador podrá dar de baja un nivel si éste ya no tiene cabida en el sistema pero se quiere seguir manteniendo la trazabilidad. Para ello realizará pasos análogos a los de borrar niveles.

Especificaciones de trazo grueso – Niveles

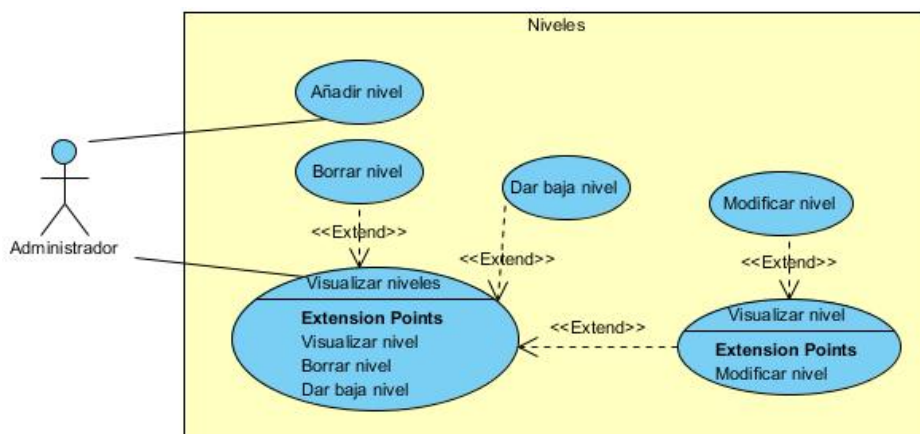


Diagrama de casos de uso – Niveles



COLORES	
CAMPO	DESCRIPCIÓN
Caso de uso	43 - Visualizar colores
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Debe ser posible visualizar de forma clara el conjunto de colores existentes en el sistema.
Caso de uso	44 - Visualizar color
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario está interesado en obtener todos los datos relativos a un color existente. Lo identifica y el sistema se los ofrece.
Caso de uso	45 - Añadir color
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario desea dar de alta un nuevo color. Realiza dicha solicitud al sistema, quien efectúa el alta tras recibir todos los datos necesarios.
Caso de uso	46 - Borrar color
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Un color deja de estar operativo y el administrador se dispone a eliminarlo del sistema. Lo identifica y solicita eliminarlo, tras lo cual el sistema le pide que confirme que ese es el color que desea eliminar. El administrador acepta y el color es borrado.
Caso de uso	47 - Modificar color
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Un color consta con datos erróneos, ante lo cual el administrador desea corregirlos. Identifica dicho color obteniendo todos sus datos en pantalla, edita los que considera equivocados y pide al sistema que los actualice.
Caso de uso	48 - Dar de baja color
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Al igual que se pueden eliminar colores, también debe ser posible darlos de baja. El usuario seleccionará un conjunto de colores y pedirá al sistema que los dé de baja, realizando dicha acción bajo confirmación previa.

Especificaciones de trazo grueso – Colores

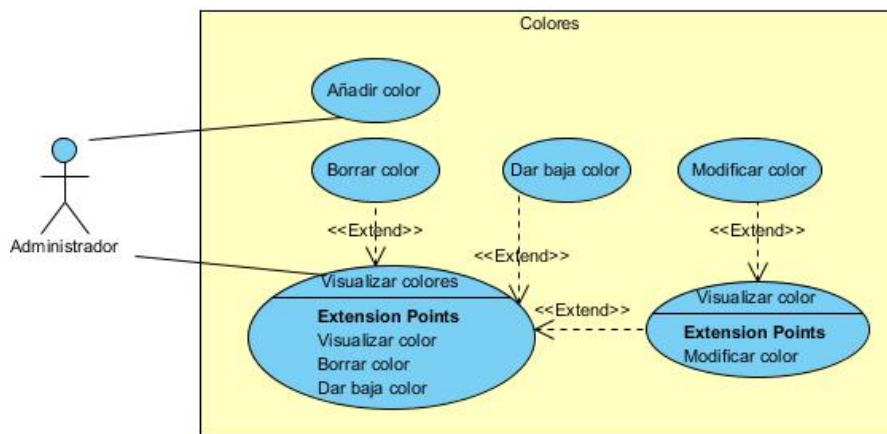


Diagrama de casos de uso – Colores



TARJETAS	
CAMPO	DESCRIPCIÓN
Caso de uso	49 - Visualizar tarjetas
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Será posible visualizar todas las tarjetas existentes en el centro.
Caso de uso	50 - Visualizar tarjeta
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	En caso de ser necesario, el administrador puede pedir al sistema que les muestre todos los datos de una tarjeta (color y numeración).
Caso de uso	51 - Añadir tarjeta
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Se dispone de una nueva tarjeta, queriendo darla de alta en el sistema para que pueda ser utilizada. Se indica a éste tal necesidad, a lo que responde pidiendo al usuario todos los datos necesarios. Éste se los proporciona y, tras aprobar que sean válidos, la tarjeta pasa a estar operativa.
Caso de uso	52 - Borrar tarjeta
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario selecciona una tarjeta y solicita al sistema darla de baja. Tras aprobar su borrado la tarjeta deja de existir en el sistema.
Caso de uso	53 - Modificar tarjeta
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Se desea modificar los datos de una tarjeta para poder darle otro uso distinto al que tiene. Para ello, el administrador identifica la tarjeta que quiere modificar, edita los datos oportunos y el sistema los recoge.
Caso de uso	54 - Dar baja tarjeta
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario quiere dar de baja una tarjeta. Selecciona la tarjeta e indica al sistema que proceda con su baja. El sistema realiza dicha acción si es aprobada por el usuario.

Especificaciones de trazo grueso – Tarjetas

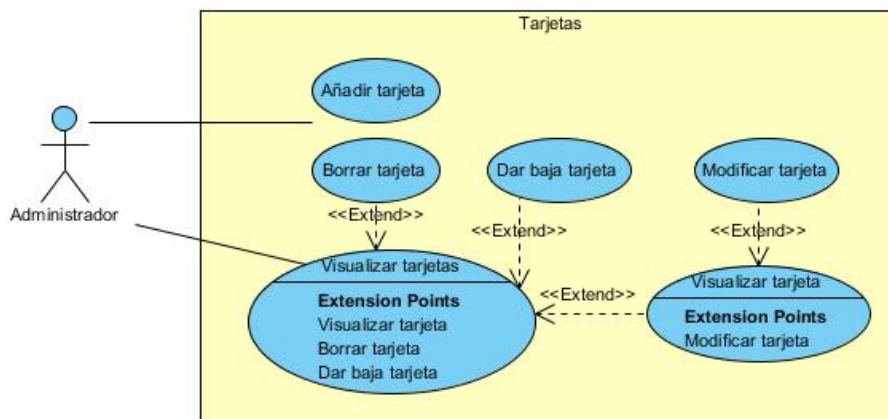


Diagrama de casos de uso – Tarjetas



TIPOS DE ACCESO	
CAMPO	DESCRIPCIÓN
Caso de uso	55 - Visualizar tipos de acceso
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El sistema deberá permitir que el administrador pueda visualizar todos los tipos de accesos existentes.
Caso de uso	56 - Visualizar tipo de acceso
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario desea conocer todos los datos de un tipo de acceso. Lo único que debe hacer es identificarlo e instar al sistema a que muestre todos los datos.
Caso de uso	57 - Añadir tipo de acceso
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Se desea otorgar un tipo de acceso inexistente a un departamento, teniendo como paso previo la realización del alta de dicho tipo de acceso. El usuario lo solicita al sistema, quien le insta a que rellene un formulario que le proporciona. Tras hacerlo y validarlo, el nuevo tipo de acceso queda registrado.
Caso de uso	58 - Borrar tipo de acceso
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Un tipo de acceso deja de ser válido, pretendiendo el administrador borrarlo. Lo identifica y traslada al sistema dicha necesidad, quien le pide que confirme que el tipo de acceso seleccionado es el que realmente quiere borrar. Tras aprobar el borrado, el tipo de acceso deja de existir en el sistema.
Caso de uso	59 - Modificar tipo de acceso
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador del sistema desea modificar un tipo de acceso que contiene datos erróneos. Para ello lo selecciona y, tras desplegar el sistema todos los datos, modifica lo necesario y aprueba los cambios.

Especificaciones de trazo grueso – Tipos de acceso

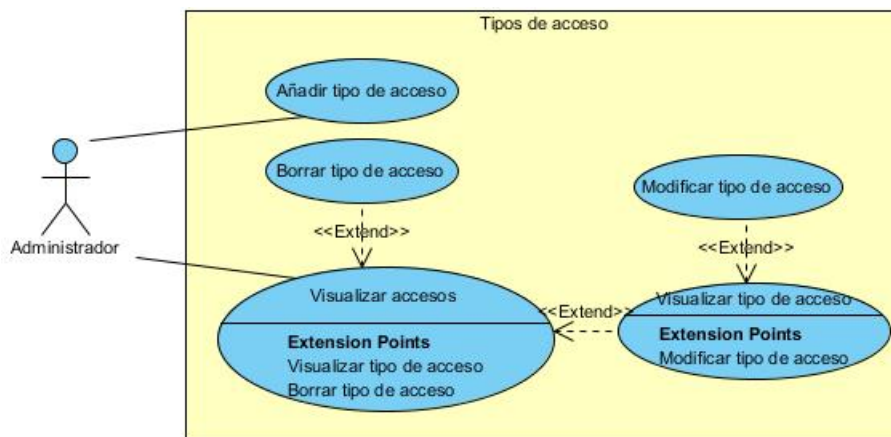


Diagrama de casos de uso – Tipos de acceso



CARGOS	
CAMPO	DESCRIPCIÓN
Caso de uso	60 - Visualizar cargos
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	No debe suponer esfuerzo alguno el que el usuario pueda visualizar todos los tipos de accesos existentes en el sistema.
Caso de uso	61 - Visualizar cargo
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador puede visualizar todos los datos referentes a un cargo mediante la identificación del mismo, encargándose el sistema de obtenerlos y mostrárselos.
Caso de uso	62 - Añadir cargo
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario decide introducir un nuevo cargo para un rango de personas con las que Informática Navarra S.A. pasará a mantener contacto. Traslada dicha solicitud al sistema, quien le pide todos los datos necesarios. El usuario se los proporciona y valida el alta, quedando el nuevo cargo registrado en el sistema.
Caso de uso	63 - Borrar cargo
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Se desea dar de baja un cargo que ha caído en desuso. Basta con identificarlo y pedir al sistema que lo elimine.
Caso de uso	64 - Modificar cargo
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Se decide modificar los datos de un cargo por no ser éstos incorrectos. Tras identificarlo, el usuario modifica los campos que crea oportunos y los valida, pasando a tener el cargo un nuevo estado.

Especificaciones de trazo grueso – Cargos

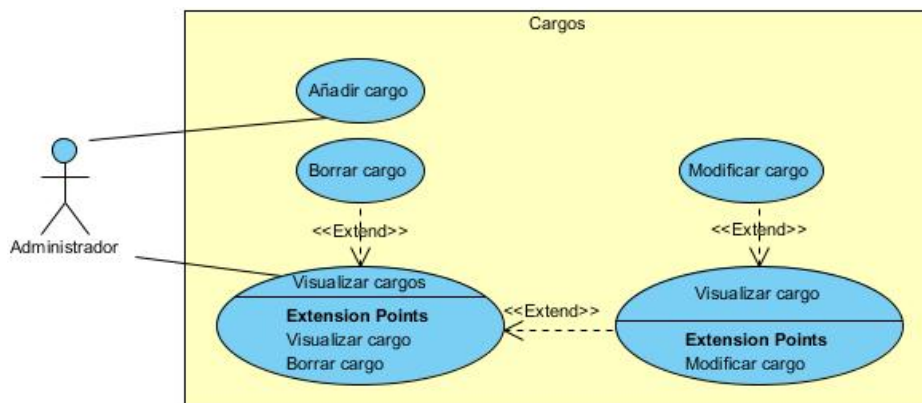


Diagrama de casos de uso – Cargos



ZONAS DEL CENTRO	
CAMPO	DESCRIPCIÓN
Caso de uso	65 - Visualizar zonas del centro
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario no debe tener el mínimo inconveniente para poder consultar de un simple vistazo todas las zonas que componen el centro.
Caso de uso	66 - Visualizar zona del centro
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Visualizar todos los datos relativos a una zona es tan sencillo como que el administrador la seleccione, encargándose el sistema de obtenerlos y mostrárselos.
Caso de uso	67 - Añadir zona del centro
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Surge una nueva zona en el centro que debe ser tratada al tener ésta cerraduras que abrir con las llaves existentes. Se comunica al sistema esta situación, pidiendo éste todos los datos que correspondan. El usuario los introduce y valida, quedando registrados de forma satisfactoria.
Caso de uso	68 - Borrar zona del centro
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario identifica una zona y solicita al sistema que la borre. Éste le pide que compruebe si la zona seleccionada es la que realmente desea borrar, a lo que el usuario responde afirmativamente. El borrado se produce y la zona deja de existir.
Caso de uso	69 - Modificar zona del centro
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Existen datos que no se corresponden con la realidad en alguna de las zonas existentes en el centro. El usuario identifica la zona afectada y el sistema muestra los datos de ésta para que puedan ser corregidos.

Especificaciones de trazo grueso – Zonas del centro

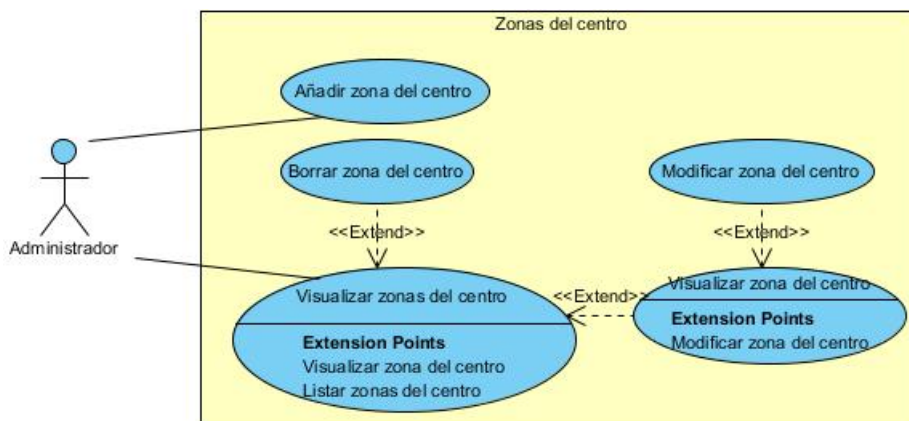


Diagrama de casos de uso – Zonas del centro



LLAVES	
CAMPO	DESCRIPCIÓN
Caso de uso	70 - Visualizar llaves
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Una de las funcionalidades que deberá ofrecer el sistema es la de mostrar los principales datos de las llaves registradas, pudiendo realizar una búsqueda sobre ellas.
Caso de uso	71 - Visualizar llave
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador desea obtener todos los datos relativos a una llave. Identifica la llave y el sistema se encarga de mostrarle todos los datos que posee de ella.
Caso de uso	72 - Añadir llave
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Se desea añadir una nueva llave a las ya existentes. El sistema pide al administrador todos los datos necesarios, aprobándolos el usuario y quedando registrados.
Caso de uso	73 - Borrar llave
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Una llave deja de tener validez, lo que promueve su eliminación. Para ello, basta con identificarla y aprobar su borrado.
Caso de uso	74 - Dar de baja llave
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	En aras de poder mantener la trazabilidad de todos los préstamos realizados en los accesos, existirá una funcionalidad que permitirá dar de baja llaves, quedando éstas registradas en el sistema pero no formando parte activa de la aplicación.
Caso de uso	75 - Modificar llave
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario tiene la necesidad de modificar los datos de una llave por no ser éstos correctos. Identifica la llave y el sistema se encarga de desplegar todos los datos de ésta. El usuario edita todo lo que estime conveniente e insta al sistema para que registre los cambios.
Caso de uso	76 - Listar llaves
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario podrá pedir a la aplicación que genere un listado con todas las llaves registradas en el sistema.

Especificaciones de trazo grueso – Llaves

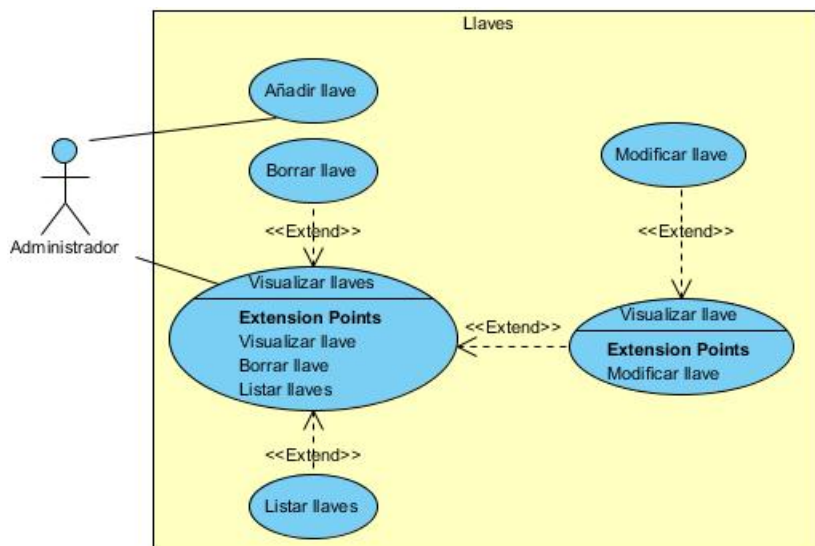


Diagrama de casos de uso – Llaves



LLAVEROS	
CAMPO	DESCRIPCIÓN
Caso de uso	77 - Visualizar llaveros
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador podrá acceder de manera sencilla a la visualización de todos los llaveros que ha registrado en el sistema y que permanecen activos.
Caso de uso	78 - Visualizar llavero
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario identifica un llavero y solicita visualizar todos sus datos, a lo que el sistema responde recuperándolos y mostrándoselos.
Caso de uso	79 - Añadir llavero
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	La dirección de Informática Navarra S.A. conforma un nuevo llavero, debiendo ser dado de alta en el sistema como paso previo para que pueda ser prestado a los trabajadores. Se solicita al sistema dicha alta proporcionándole todos los datos necesarios. Tras aprobar el usuario que todo es correcto se produce la inserción.
Caso de uso	80 - Borrar llavero
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Borrar un llavero es tan fácil como identificarlo y pedir al sistema que lo elimine. El único requerimiento que solicitará es la confirmación de que el llavero a borrar es el que el usuario realmente quiere borrar.
Caso de uso	81 - Dar de baja llavero
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Al igual que ocurre con las llaves, debe ser posible dar de baja un llavero siguiendo almacenando sus datos pero dejando de formar parte activa de la aplicación.
Caso de uso	82 - Modificar llavero
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Por algún motivo varía la composición de un llavero. El administrador solicita al sistema modificar dicho llavero, ofreciéndole éste todos los datos para que pueda editarlos. Una vez realizadas todos los cambios oportunos se aprueba la nueva composición del llavero.

Especificaciones de trazo grueso – Llaveros

Caso de uso	83 - Añadir llave
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Una nueva llave entra a formar parte del llavero. El administrador la selecciona y pide al sistema que la introduzca en el llavero indicado.
Caso de uso	84 - Borrar llave
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Al contrario que el caso de uso anterior, una llave deja de tener sentido en un llavero. Se comunica este hecho al sistema, quien se encarga de borrar la llave del llavero.
Caso de uso	85 - Listar llaveros
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador podrá realizar un listado con todos los llaveros que desee según los criterios de búsqueda suministrados a la aplicación.

Especificaciones de trazo grueso – Llaveros

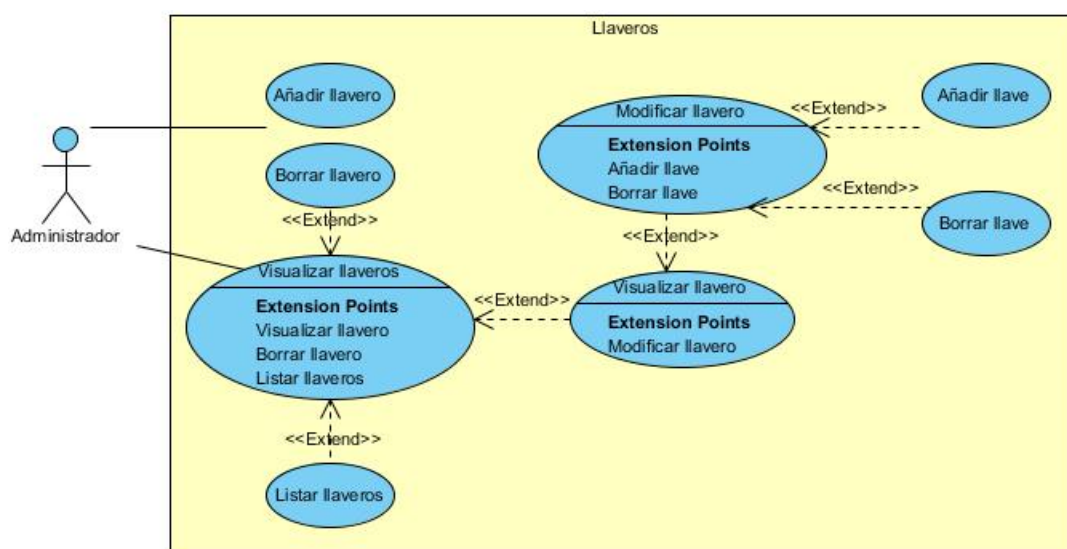


Diagrama de casos de uso – Llaveros



ALMACENES DE LLAVES	
CAMPO	DESCRIPCIÓN
Caso de uso	86 - Visualizar almacenes de llaves
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario tendrá la posibilidad de poder observar todos los almacenes de llaves existentes, con sus principales características y la relación de llaves y llaveros que albergan.
Caso de uso	87 - Visualizar almacén de llaves
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Se puede visualizar todos los datos de un almacén identificándolo e instando al sistema a que los muestre.
Caso de uso	88 - Añadir almacén de llaves
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	La dirección de la empresa decide crear un nuevo almacén de llaves para el centro. Para que pase a estar operativo, el administrador indica al sistema que desea añadirlo. Éste pide todos los datos necesarios, recibiendo por parte del administrador y llevando a cabo la inserción en el sistema.
Caso de uso	89 - Borrar almacén de llaves
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador identifica un almacén y solicita al sistema que lo borre. El sistema pide confirmación para que se lleve a cabo la eliminación del almacén y todas copias de las llaves que éste contiene. Tras otorgarla el usuario consigue su objetivo.
Caso de uso	90 - Modificar almacén de llaves
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Modificar un almacén de llaves es tan sencillo como identificarlo y editar los campos desplegados que se estimen oportunos.
Caso de uso	91 - Añadir llave
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Bien sea por que se realizan más copias o por que se introduce una nueva llave en la base de datos, deberá proporcionarse la posibilidad de añadir llaves a cualquier almacén. El usuario seleccionará la llave a introducir y el número de copias existentes, mientras que el sistema se encargará de almacenar dicha información.

Especificaciones de trazo grueso – Almacenes de llaves



Caso de uso	92 - Borrar llave
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Una llave deja de formar parte de un almacén. El usuario la selecciona dentro de éste e indica al sistema que la elimine, dejando de existir copia alguna.
Caso de uso	93 - Modificar copias llave
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Debe ser posible modificar el número de copias de una llave existentes dentro de un almacén con solo seleccionarla y variar el número de copias actuales.
Caso de uso	94 - Añadir llavero
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Al igual que las llaves, también debe ser posible añadir llaveros a los almacenes, siguiendo el mismo procedimiento que el llevado a cabo con éstas.
Caso de uso	95 - Borrar llavero
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	En el momento en que un llavero deje de existir dentro de un almacén, el usuario lo seleccionará e indicará al sistema que lo elimine del almacén. Tras una petición de ratificación por parte del sistema, la operación será llevada a cabo.
Caso de uso	96 - Modificar copias llavero
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Caso de uso idéntico al de las llaves, pero referido a los llaveros.
Caso de uso	97 - Listar almacenes de llaves
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	La visualización de los diversos almacenes existentes se acompañará de una vía para poder obtener listados con todos los elementos que almacenan.

Especificaciones de trazo grueso – Almacenes de llaves

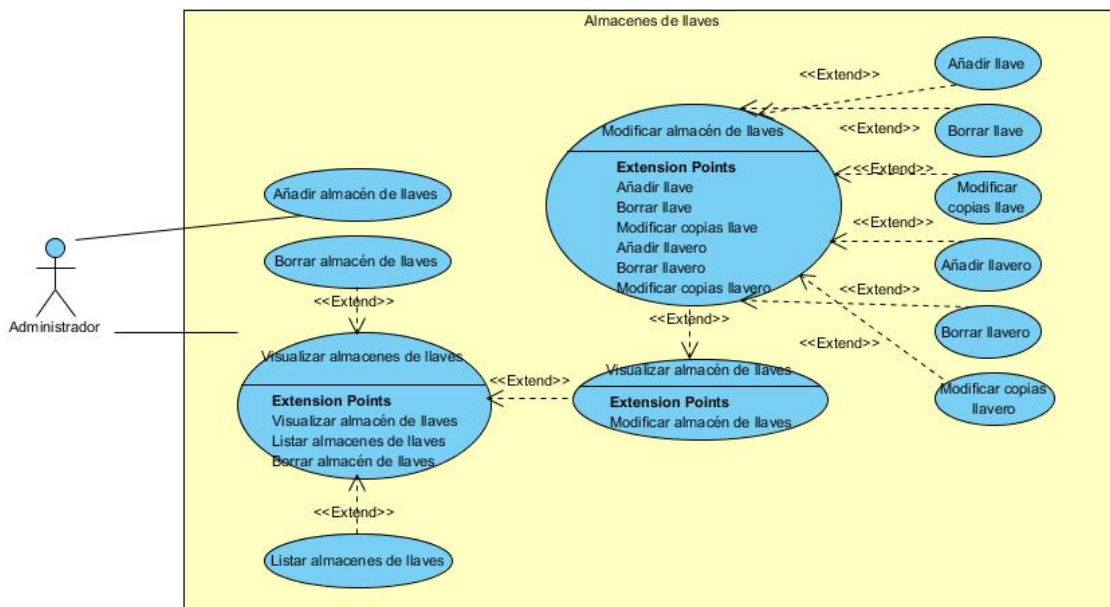


Diagrama de casos de uso – Almacenes de llaves



PRÉSTAMOS PENDIENTES	
CAMPO	DESCRIPCIÓN
Caso de uso	98 - Visualizar préstamos
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador desea visualizar qué préstamos están pendientes de devolución. Dicha pretensión queda resuelta gracias a que el sistema le proporciona una pantalla desde la que poder observar qué tarjetas, llaves y llaveros están a la espera de ser devueltos.
Caso de uso	99 - Visualizar datos del prestatario
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Una vez que el usuario disponga de los préstamos pendientes de devolución, podrá consultar qué persona es la prestataria en cada caso de una manera fácil y rápida, directa.
Caso de uso	100 - Devolver préstamo
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador deberá poder devolver los préstamos que desee desde la pantalla de consulta de préstamos, ya que en la de accesos no tendrá privilegios para editar información y aunque los tuviera no podría devolver préstamos de aquellos accesos que tengan salida.
Caso de uso	101 - Enviar recordatorio vía E-Mail
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	En caso de que el préstamo se prolongue en exceso o requiera que sea devuelto con la mayor brevedad posible, el sistema deberá proporcionar una vía que permita enviar un recordatorio vía E-Mail al prestatario en caso de disponer de alguna dirección de contacto.
Caso de uso	102 - Listar préstamos pendientes
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador podrá generar en cualquier momento un listado con todos los préstamos que están pendientes de devolución, combinando en él tarjetas, llaves y llaveros.

Especificaciones de trazo grueso – Préstamos pendientes

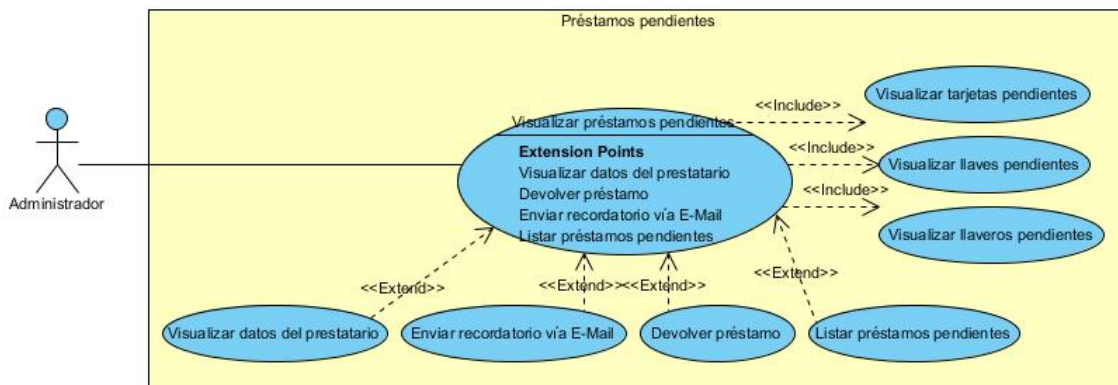


Diagrama de casos de uso – Préstamos pendientes



CUENTAS DEL SISTEMA	
CAMPO	DESCRIPCIÓN
Caso de uso	103 - Visualizar cuentas
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	La aplicación a desarrollar debe permitir visualizar todas las cuentas existentes en el sistema, así como sus principales datos, de forma que sea muy sencillo poder acceder a ellas para realizar las funciones que se estimen oportunas.
Caso de uso	104 - Visualizar cuenta
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario quiere visualizar toda la información relativa a una cuenta. Para ello, la identifica en el sistema y éste se encarga de recuperar y mostrar todos los datos de los que disponga.
Caso de uso	105 - Añadir cuenta
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador notifica al sistema que desea dar de alta una nueva cuenta. Éste le pide que le proporcione todos los datos necesarios, recogiendo e incorporando al sistema.
Caso de uso	106 - Borrar cuenta
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Es posible realizar el borrado de una cuenta identificándola en el sistema y pidiendo a éste que la elimine.
Caso de uso	107 - Modificar cuenta
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador desea modificar los datos de una cuenta. El sistema le pide que la identifique. El administrador obedece y obtiene así todos los datos de la cuenta seleccionada, modificando lo oportuno y pidiendo al sistema que realice dichos cambios.
Caso de uso	108 - Listar cuentas
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	La aplicación debe proporcionar un modo de poder obtener un listado de cuentas del sistema que cumplan unas condiciones establecidas. El usuario limitará la búsqueda e instará al sistema a que elabore el listado con los datos que obtenga.

Especificaciones de trazo grueso – Cuentas del sistema

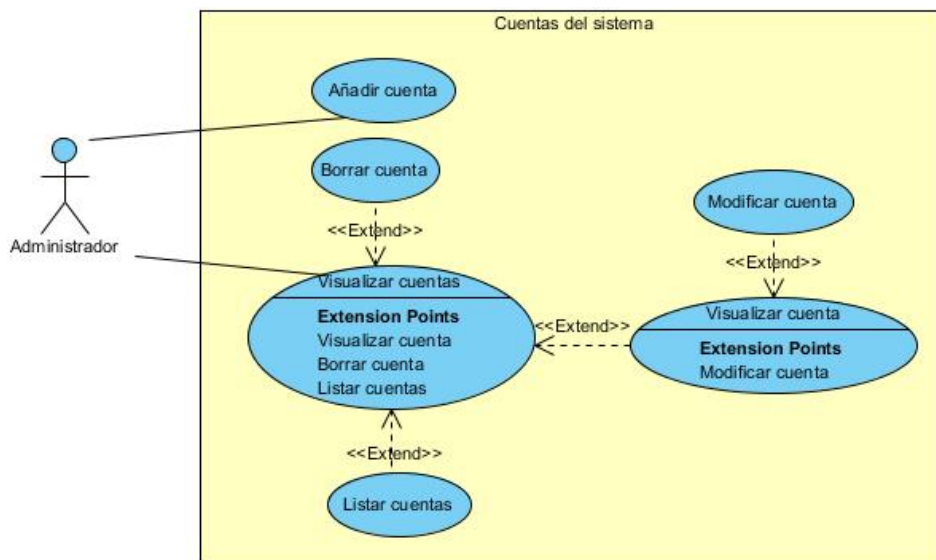


Diagrama de casos de uso – Cuentas del sistema



TIPOS DE USUARIO	
CAMPO	DESCRIPCIÓN
Caso de uso	109 - Visualizar tipos de usuario
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Al igual que toda la información manejada hasta el momento, los tipos de usuarios no se escapan de tener que poder ser visualizados.
Caso de uso	110 - Visualizar tipo de usuario
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario desea saber los datos completos de un tipo de usuario existente en el sistema. Lo selecciona y el sistema es el que se encarga de rescatarlos y mostrárselos.
Caso de uso	111 - Añadir tipo de usuario
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador desea añadir un nuevo tipo de usuario en el sistema. Éste le proporciona un formulario que deberá rellenar y validar, encargándose el sistema de registrarlo.
Caso de uso	112 - Borrar tipo de usuario
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Al contrario que el anterior caso de uso, así como se deben poder añadir tipos de usuarios, también deberá ser posible eliminarlos mediante su identificación y petición al sistema de que realice esta acción.
Caso de uso	113 - Modificar tipo de usuario
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Los datos de un tipo de usuario que consta en el sistema son erróneos. El administrador lo selecciona y pide al sistema poder modificarlo. Éste le muestra los datos actuales del tipo de usuario, siendo el administrador quien los modifica y el sistema quien actualiza los antiguos por éstos.

Especificaciones de trazo grueso – Tipos de usuarios

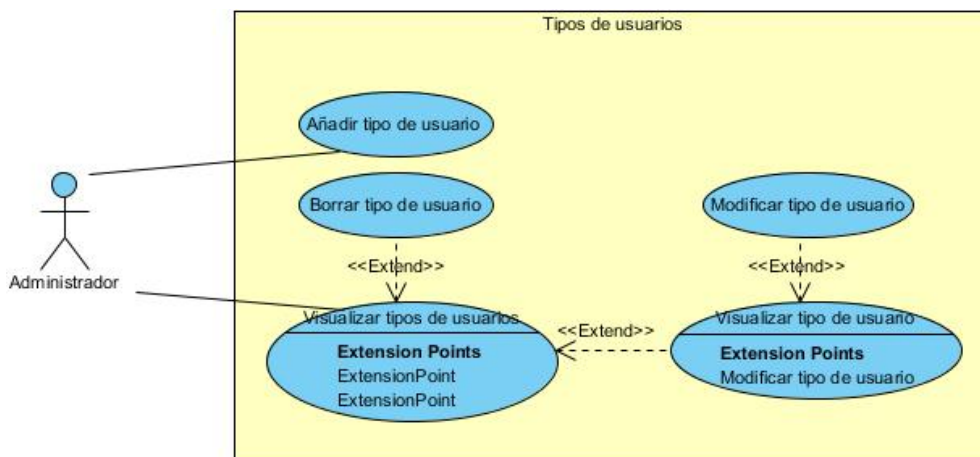


Diagrama de casos de uso – Tipos de usuarios

HISTORIAL DE ACCESOS AL SISTEMA	
CAMPO	DESCRIPCIÓN
Caso de uso	114 - Visualizar accesos al sistema
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador deberá poder consultar los accesos realizados al sistema por los diversos usuarios del mismo, de forma total o acotando esta con diversas variables (nombre, apellidos, fechas,...).
Caso de uso	115 - Listar accesos al sistema
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	La visualización de accesos llevada a cabo en el anterior punto podrá ser obtenida mediante un listado proporcionado por el sistema.

Especificaciones de trazo grueso – Historial de accesos al sistema

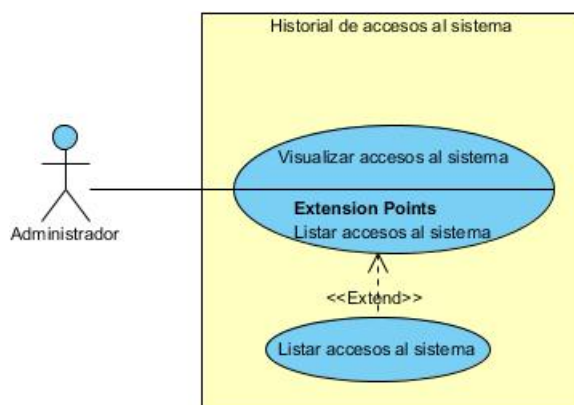


Diagrama de casos de uso – Historial de accesos al sistema

RECORDATORIOS Y BORRADO	
CAMPO	DESCRIPCIÓN
Caso de uso	117 - Recordar personal a punto de caducar
Prioridad	Importante
Actor(es)	Sistema
Funcionalidad	El sistema será capaz de enviar E-Mails a los departamentos cuyo personal está próximo a caducar según la política establecida por Informática Navarra S.A.
Caso de uso	118 - Recordar préstamos pendientes
Prioridad	Importante
Actor(es)	Sistema
Funcionalidad	También deberá existir un procedimiento por el cual toda aquella persona que tenga algún préstamo pendiente de devolución sea avisada en los casos que se estimen oportunos.
Caso de uso	119 - Borrar datos obsoletos
Prioridad	Importante
Actor(es)	Sistema
Funcionalidad	Para evitar que el sistema crezca de forma desmesurada, se deberá poder llevar a cabo el borrado de toda aquella información que carezca de sentido seguir almacenándola en el sistema.

Especificaciones de trazo grueso – Recordatorios y borrado

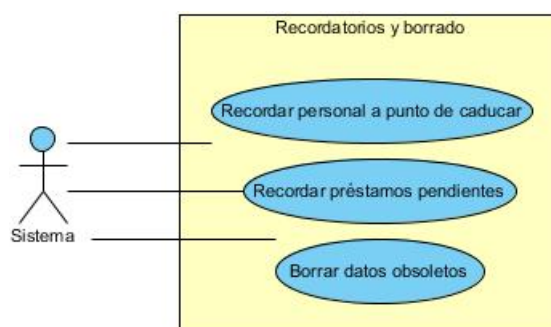


Diagrama de casos de uso – Recordatorios y borrado



PARTES DE SEGURIDAD	
CAMPO	DESCRIPCIÓN
Caso de uso	120 - Visualizar partes de seguridad
Prioridad	Importante
Actor(es)	Administrador y Vigilante
Funcionalidad	Se debe proporcionar una forma efectiva de poder visualizar todos los partes almacenados en el sistema, de modo que administrador y vigilante puedan acceder a ellos en el momento que deseen.
Caso de uso	121 - Visualizar parte de seguridad
Prioridad	Importante
Actor(es)	Administrador y Vigilante
Funcionalidad	El usuario del sistema desea consultar toda la información que consta en un parte de seguridad concreto. Lo identifica y el sistema es el que se encarga de proporcionar toda la información rescatada.
Caso de uso	122 - Añadir parte de seguridad
Prioridad	Importante
Actor(es)	Vigilante
Funcionalidad	El vigilante de seguridad desea crear un nuevo parte durante el desarrollo de un servicio. Transmite esta intención al sistema, quien le ofrece un formulario en el que puede empezar a redactar el parte y guardar todo lo que escriba.
Caso de uso	123 - Borrar parte de seguridad
Prioridad	Importante
Actor(es)	Administrador y Vigilante
Funcionalidad	El usuario identifica un parte de seguridad y solicita al sistema su borrado. Éste le pide que confirme que el parte seleccionado es el que realmente quiere borrar. El usuario da su conformidad y el parte deja de existir en el sistema.
Caso de uso	124 - Modificar parte de seguridad
Prioridad	Importante
Actor(es)	Vigilante
Funcionalidad	El vigilante desea modificar un parte de seguridad, que redactó con anterioridad, por estar éste inacabado o contener datos erróneos. Lo identifica y el sistema le muestra todos los datos que dispone de éste. El usuario edita y añade todo lo que estime oportuno, siendo el sistema quien registre los cambios.
Caso de uso	125 - Listar parte de seguridad
Prioridad	Importante
Actor(es)	Administrador y Vigilante
Funcionalidad	Tanto el administrador como el vigilante podrán descargar una copia del parte de seguridad que se encuentren visualizando.

Especificaciones de trazo grueso – Partes de seguridad

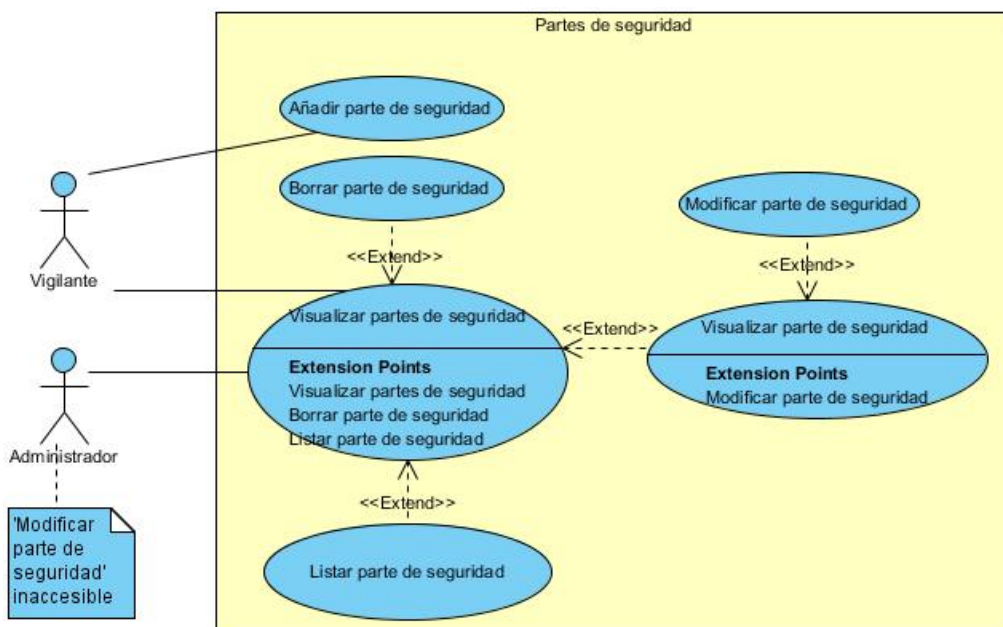


Diagrama de casos de uso – Partes de seguridad



NOTAS	
CAMPO	DESCRIPCIÓN
Caso de uso	126 - Redactar nota
Prioridad	Deseable
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	El usuario desea redactar una nota. El sistema le proporciona un formulario para que la redacte y elija los destinatarios. Tras validarla, el sistema se encarga de almacenarla y hacerla llegar a sus destinatarios.
Caso de uso	127 - Visualizar notas recibidas
Prioridad	Deseable
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	Se dispondrá de un listado de notas recibidas desde donde el usuario podrá acceder al contenido de las mismas.
Caso de uso	128 - Visualizar nota recibida
Prioridad	Deseable
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	El destinatario de una nota desea conocer todos sus detalles. Para ello la identifica y le pide al sistema que le muestre todos los datos de ésta.
Caso de uso	129 - Borrar nota recibida
Prioridad	Deseable
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	Es posible borrar una nota innecesaria con sólo identificarla y aprobar su eliminación.
Caso de uso	130 - Visualizar notas enviadas
Prioridad	Deseable
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	Así como se dispone de un almacén de entrada de notas, también se tendrá otro de notas enviadas, desde donde poder realizar la visualización y eliminación de éstas.
Caso de uso	131 - Visualizar nota enviada
Prioridad	Deseable
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	El redactor de la nota desea consultar en qué estado se encuentra una de las notas que ha redactado. Tras identificarla y realizar dicha petición al sistema, éste le mostrará la nota redactada y qué destinatarios la han leído y quiénes no.
Caso de uso	132 - Borrar nota enviada
Prioridad	Deseable
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	El usuario solicita al sistema descartar una nota de la cual es redactor mediante su identificación. A partir de entonces ésta no volverá a mostrarse al usuario.

Especificaciones de trazo grueso – Notas

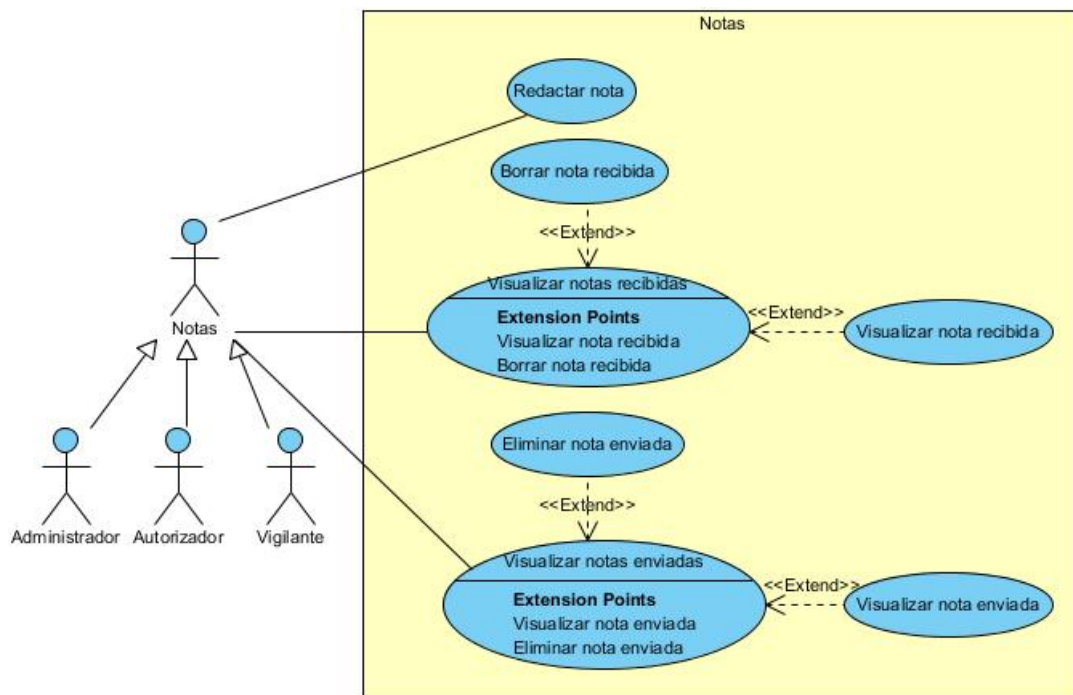


Diagrama de casos de uso – Notas

BASE DE DATOS	
CAMPO	DESCRIPCIÓN
Caso de uso	133 - Importar base de datos
Prioridad	Deseable
Actor(es)	Administrador
Funcionalidad	El administrador desea importar toda la información que posee en un archivo SQL. El sistema le insta a que indique la ruta del fichero a importar, petición que obedece consiguiendo introducir en el sistema toda la información deseada.
Caso de uso	134 - Exportar base de datos
Prioridad	Deseable
Actor(es)	Administrador
Funcionalidad	El usuario desea exportar toda la información existente en el sistema para que así pueda ser recuperada en un futuro. Indica dicha necesidad al sistema, quien genera un archivo SQL con toda la población de la base de datos.

Especificaciones de trazo grueso – Base de datos

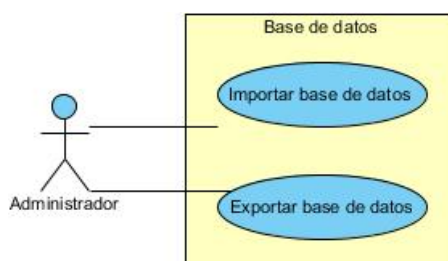


Diagrama de casos de uso – Base de datos



4. Descripciones de trazo fino

Con las descripciones que creamos en la sección anterior, cliente y equipo de desarrollo llegaron a un acuerdo sobre qué características debía tener la aplicación. Una vez que se dispuso de ellas ordenadas por prioridades, iteración a iteración se fueron seleccionando y profundizando en sus definiciones, creando lo que se llama “descripciones de trazo fino de los casos de uso” y cuyas semánticas recogemos a lo largo de las siguientes hojas.

4.1. Cuentas – Parte 1

Caso de uso 1: Login	
Descripción	<p>El primer paso para acceder al sistema es realizar un logueo, que conducirá a la pantalla principal de la aplicación bajo las condiciones impuestas por el tipo de usuario que realice el acceso al sistema.</p> <p>Realizar el logueo es sencillo, bastará con introducir el nombre de usuario y contraseña en la pantalla de inicio y presionar el botón de ‘Entrar’.</p>
Actores principal	Administrador, Vigilante, Autorizador y RP.
Pre-condiciones	El usuario desea acceder al sistema.
Flujos	Básico <ol style="list-style-type: none">1. El usuario introduce su nombre de usuario y contraseña y pulsa el botón ‘Entrar’.2. El sistema realiza la comprobación de la existencia de la cuenta, pasando a cerrar la ventana ‘Identificación’ y a abrir la de ‘Aplicación’ con las funcionalidades que le estén permitidas. (*)
	Alternativos (*) <p>El usuario y/o contraseña no existen o no casan con los de la base de datos:</p> <ul style="list-style-type: none">• El sistema indica al usuario del error acontecido y se pasa al punto de partida, permaneciendo inalterados los datos introducidos anteriormente por el usuario.
Post-condiciones	El sistema ha desplegado la pantalla desde la que el usuario podrá realizar las funciones que estime oportunas.

Caso de uso 1: Login



4.2. Accesos

Caso de uso 2: Visualizar accesos		
Descripción	Uno de los requisitos que impone el cliente es que se puedan observar el conjunto de accesos de forma sencilla y rápida, instantánea. Habitualmente se trabaja con los accesos al centro que se han realizado en el día, con que será conveniente aportar una simple forma de visualizarlos y hacer accesible al usuario el resto de accesos a través de algún paso adicional.	
Actores principal	Administrador y Vigilante.	
Pre-condiciones	El usuario se encuentra logueado en el sistema y desea consultar los accesos realizados al centro.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El administrador o vigilante desea consultar los accesos realizados en el centro en el día en que se encuentra. (*) 2. El sistema comprueba que las fechas proporcionadas son correctas y realiza la búsqueda de accesos según los parámetros establecidos, mostrándole al usuario el resultado a través de una tabla.
	Alternativos (*)	<p>El usuario desea consultar los accesos realizados entre días y horas específicas:</p> <ul style="list-style-type: none"> • Redefine los parámetros que delimitan la búsqueda y se pasa al punto 2.
Post-condiciones	El usuario obtiene por pantalla los accesos realizados al centro entre los días y horas indicados.	

Caso de uso 2: Visualizar accesos

Caso de uso 3: Visualizar acceso		
Descripción	Así como el usuario puede ver los datos característicos de todos aquellos accesos que cumplan con los parámetros establecidos, también debe poder visualizar todos los datos del acceso que desee.	
Actores principal	Administrador y Vigilante.	
Pre-condiciones	El usuario se encuentra logueado en el sistema y desea consultar todos los datos de uno de los accesos que visualiza en pantalla.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona un acceso en pantalla. 2. El sistema se encarga de recoger todos los datos referentes a ese acceso y mostrarlos.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario obtiene por pantalla todos los datos pertenecientes al acceso seleccionado.	

Caso de uso 3: Visualizar acceso



Caso de uso 4: Dar entrada	
Descripción	Todo acceso al centro queda registrado en estos momentos en hojas manejadas por los vigilantes. Aquí lo que haremos es evitar este trámite, siendo almacenados todos los accesos mediante la aplicación.
Actores principal	Vigilante.
Pre-condiciones	Una persona desea acceder al centro y el vigilante, logueado en el sistema, desea registrar el acceso.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario indica al sistema que desea registrar un nuevo acceso. 2. El sistema le proporciona un formulario en el que definir todos los datos. 3. El usuario rellena el formulario, respondiéndole el sistema con la tarjeta que deberá emplear en el acceso y si es necesario que avise a alguna persona del centro. 4. Acto seguido, el vigilante hace constar el motivo y otros datos del acceso, además de las llaves y llaveros que prestará a la persona, y termina pidiendo al sistema que registre el acceso. 5. El sistema, tras una previa aprobación, hace caso al vigilante y le indica que todo el proceso ha sido resuelto satisfactoriamente. (*)
	<p>Alternativos (*)</p> <p>El usuario introduce datos con un formato erróneo:</p> <ul style="list-style-type: none"> • El sistema avisa al usuario del error acontecido y vuelve al punto de partida. <p>El usuario no da su aprobación para que el acceso quede registrado en el sistema:</p> <ul style="list-style-type: none"> • Éste no realiza ninguna inserción y da por concluido el caso de uso.
Post-condiciones	El acceso (y préstamos) ha sido registrado con éxito y el vigilante ya puede permitir la entrada del trabajador al centro.

Caso de uso 4: Dar entrada



Caso de uso 5: Prestar llaves y/o llaveros	
Descripción	Tras haber realizado un acceso, la persona puede que haya olvidado pedir una llave o llavero, con que deberá existir la posibilidad de poder modificar los préstamos realizados en un acceso.
Actores principal	Vigilante.
Pre-condiciones	Una persona que ya se encuentra en el centro desea que le sea prestada una nueva llave o llavero. El vigilante, que se encuentra logueado en el sistema, se dispone a registrar el préstamo.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El vigilante selecciona el acceso que desea modificar. 2. El sistema le muestra toda la información acerca de éste y le permite añadir préstamos si es que el acceso no posee salida. Le muestra la lista de llaves y llaveros que la persona no posee y que le pueden ser prestados. (*) 3. El usuario selecciona las llaves y/o llaveros del nuevo préstamo. (*) 4. El caso de uso termina registrando el sistema los préstamos en la base de datos.
	<p>Alternativos (*)</p> <p>El acceso consta de salida:</p> <ul style="list-style-type: none"> • El sistema no permitirá modificarlo, dando por terminado el caso de uso. <p>No existen llaves o llaveros que poder prestar:</p> <ul style="list-style-type: none"> • El sistema no mostrará ninguna llave ni llavero y el usuario no tendrá más opción que cerrar la ventana y no poder realizar ningún préstamo.
Post-condiciones	El vigilante registra los préstamos solicitados y puede ofrecer a la persona las llaves y llaveros que necesita.

Caso de uso 5: Prestar llaves y/o llaveros



Caso de uso 6: Borrar acceso	
Descripción	No es lo normal, pero bien por error en la introducción de los datos o bien por necesidad, puede ser preciso eliminar algunos de los accesos que constan en la base de datos.
Actores principal	Autorizador y Vigilante.
Pre-condiciones	El vigilante ha introducido algún acceso de forma errónea o el autorizador desea eliminar algunos de los accesos por las razones que estime oportunas. Sea el usuario que sea, se encuentra logueado en el sistema y con intención de borrar algunos de los accesos existentes.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario selecciona de la tabla de accesos los accesos que desee eliminar y pide al sistema que los borre. 2. Éste pide confirmación al primero, por si los accesos seleccionados no fueran los que realmente desea eliminar. 3. Tras confirmar el usuario, el borrado tiene lugar, los accesos dejan de existir y el sistema refresca la tabla de accesos. (*)
	<p>Alternativos (*)</p> <p>El usuario ha seleccionado accesos que no debía y cancela la operación de borrado:</p> <ul style="list-style-type: none"> • El sistema no realiza ningún borrado y permanece en el punto previo a que el usuario pidiese el borrado.
Post-condiciones	El usuario consigue lo que desea: los accesos dejan de existir en el sistema y la tabla de accesos está actualizada con los existentes.

Caso de uso 6: Borrar acceso



Caso de uso 7: Modificar acceso		
Descripción	En el caso de uso 5 hablábamos de cómo modificar un acceso en cuanto a añadir más préstamos. Ahora tocaremos la funcionalidad de poder modificar el asunto y otros datos y posteriormente remataremos el trabajo con el borrado y devolución de préstamos.	
Actores principal	Vigilante.	
Pre-condiciones	El usuario se encuentra logueado en el sistema y desea modificar los atributos de 'Motivo' y 'Otros datos' de un acceso que no dispone de salida.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El vigilante selecciona el acceso a modificar. 2. El sistema ofrece toda la información en el formulario y hace los campos editables. 3. El usuario los modifica de forma que queden a su gusto y le pide al sistema que los grabe.
	Alternativos (*)	Ninguno
Post-condiciones	El acceso posee en sus atributos 'Motivo' y 'Otros datos' valores correctos.	

Caso de uso 7: Modificar acceso

Caso de uso 8: Borrar préstamo		
Descripción	Otra forma de poder modificar un acceso es la de eliminar uno o varios de los préstamos que se hayan realizado en éste.	
Actores principal	Vigilante.	
Pre-condiciones	El vigilante se encuentra logueado en el sistema, está visualizando un acceso y desea eliminar uno o varios de los préstamos que lo conforman.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El vigilante selecciona todos los préstamos que desea eliminar y pulsa el botón de eliminado. 2. El sistema le pide confirmación de que los accesos que ha seleccionado son los que realmente desea eliminar. (*) 3. El vigilante responde afirmativamente, lo que hace que los préstamos dejen de existir en la base de datos y la tabla de préstamos queda actualizada.
	Alternativos (*)	<p>El usuario decide que no desea eliminar alguno de los préstamos seleccionados:</p> <ul style="list-style-type: none"> • Pulsa 'No' cuando el sistema le pide la confirmación y el sistema permanece en el estado que se llegó al visualizar el acceso.
Post-condiciones	Las descritas en el tercer punto del flujo básico.	

Caso de uso 8: Borrar préstamo



Caso de uso 9: Dar salida					
Descripción	Complementariamente a poder dar entradas, también se debe poder dar salidas a todo el personal que abandone el centro, existiendo dos vías: <ol style="list-style-type: none"> 1. Salida indicando que prestamos se devuelven. 2. Salida en la que se devuelven todos los préstamos pendientes en el acceso. 				
Actores principal	Vigilante.				
Pre-condiciones	El vigilante, con la aplicación corriendo, observa que una persona desea abandonar el centro. Ante esta situación, se dispone a registrar esta salida.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> <ol style="list-style-type: none"> 1. El usuario selecciona el acceso al que desea otorgar salida. 2. El sistema le muestra todos los datos del acceso. 3. El usuario da salida al acceso seleccionando los préstamos devueltos. 4. El sistema pide confirmación para llevar a cabo la modificación del acceso. 5. El usuario la da. (*) </td> </tr> <tr> <td>Alternativos (*)</td> <td> El usuario no da el visto bueno para que la salida se lleve a efecto: <ul style="list-style-type: none"> • El sistema no realiza ninguna acción y da por finalizado el caso de uso. </td> </tr> </table>	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona el acceso al que desea otorgar salida. 2. El sistema le muestra todos los datos del acceso. 3. El usuario da salida al acceso seleccionando los préstamos devueltos. 4. El sistema pide confirmación para llevar a cabo la modificación del acceso. 5. El usuario la da. (*) 	Alternativos (*)	El usuario no da el visto bueno para que la salida se lleve a efecto: <ul style="list-style-type: none"> • El sistema no realiza ninguna acción y da por finalizado el caso de uso.
	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona el acceso al que desea otorgar salida. 2. El sistema le muestra todos los datos del acceso. 3. El usuario da salida al acceso seleccionando los préstamos devueltos. 4. El sistema pide confirmación para llevar a cabo la modificación del acceso. 5. El usuario la da. (*) 			
Alternativos (*)	El usuario no da el visto bueno para que la salida se lleve a efecto: <ul style="list-style-type: none"> • El sistema no realiza ninguna acción y da por finalizado el caso de uso. 				
Post-condiciones	El acceso pasa a disponer de salida y todos los préstamos indicados han sido marcados como devueltos.				

Caso de uso 9: Dar salida

Caso de uso 10: Devolver llaves y/o llaveros					
Descripción	Existen casos en los que los prestatarios querrán devolver parte o todos sus préstamos antes de terminar su jornada.				
Actores principal	Vigilante.				
Pre-condiciones	El vigilante, que se encuentra en uso de accessManager, desea marcar como devueltos alguno de los préstamos visualizados en algún acceso.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> <ol style="list-style-type: none"> 1. El usuario selecciona los préstamos que desea marcar como devueltos e indica al sistema que los devuelva. 2. El sistema pide una previa confirmación, tras la cual procede a grabar los préstamos indicados como devueltos. (*) </td> </tr> <tr> <td>Alternativos (*)</td> <td> El usuario cancela la devolución de los préstamos: <ul style="list-style-type: none"> • El sistema no realiza ninguna operación. </td> </tr> </table>	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona los préstamos que desea marcar como devueltos e indica al sistema que los devuelva. 2. El sistema pide una previa confirmación, tras la cual procede a grabar los préstamos indicados como devueltos. (*) 	Alternativos (*)	El usuario cancela la devolución de los préstamos: <ul style="list-style-type: none"> • El sistema no realiza ninguna operación.
	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona los préstamos que desea marcar como devueltos e indica al sistema que los devuelva. 2. El sistema pide una previa confirmación, tras la cual procede a grabar los préstamos indicados como devueltos. (*) 			
Alternativos (*)	El usuario cancela la devolución de los préstamos: <ul style="list-style-type: none"> • El sistema no realiza ninguna operación. 				
Post-condiciones	El vigilante de seguridad ha marcado como devueltos los préstamos que indicaba.				

Caso de uso 10: Devolver llaves y/o llaveros



Caso de uso 11: Listar accesos		
Descripción	El usuario puede consultar cuando desee la lista de los accesos que quiera, pero además de esta funcionalidad debemos ofrecer al usuario la posibilidad de que pueda guardar los resultados en un reporte en PDF.	
Actores principal	Administrador y Vigilante.	
Pre-condiciones	El usuario se encuentra logueado en el sistema y desea obtener un listado de todos los accesos que se encuentra visualizando.	
Flujos	Básico	<ol style="list-style-type: none">1. El usuario presiona el botón de 'Generar reporte' e indica el destino (ruta) en el que desea almacenarlo.2. El sistema genera el reporte con los datos principales de todos los accesos y lo almacena donde el usuario ha indicado.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario obtiene el reporte con todos los accesos que se encuentra visualizando.	

Caso de uso 11: Listar accesos

Caso de uso 12: Dar salida automática		
Descripción	El vigilante dará salida a la gran mayoría de los accesos, pero siempre puede quedar la posibilidad que el vigilante se deje alguno. Para evitar que los accesos queden sin salida y por tanto las personas que constan en ellos no puedan volver a acceder al centro, programamos la aplicación de modo que el sistema de salida a estos accesos automáticamente.	
Actores principal	Sistema.	
Pre-condiciones	El sistema se inicia por primera vez en todo el día.	
Flujos	Básico	El sistema busca los accesos que no disponen de salida y los recorre uno a uno proporcionándoles la máxima hora de salida que pudieran tener según el tipo de acceso de cada persona.
	Alternativos (*)	Ninguno.
Post-condiciones	Se consigue que todos los accesos del día anterior dispongan de entrada y salida, permaneciendo la tabla Acceso coherente.	

Caso de uso 12: Dar salida automática



4.3. Empresas

Caso de uso 13: Visualizar empresas		
Descripción	Lo primero que debemos conseguir para poder manejar todas las empresas y personal que accede al centro es poder visualizar las primeras.	
Actores principal	Administrador, Autorizador y RP.	
Pre-condiciones	El usuario se encuentra logueado en el sistema y desea visualizar qué empresas forman parte de la vida diaria del centro.	
Flujos	Básico	<ol style="list-style-type: none">3. El usuario accede a la pestaña de 'Empresas y personal' y define la búsqueda de empresa.4. El sistema se encarga de mostrarle todas las empresas existentes que satisfacen dichos filtros.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario visualiza en pantalla todas las empresas registradas en el sistema que cumplen con sus criterios de búsqueda.	

Caso de uso 13: Visualizar empresas

Caso de uso 14: Visualizar empresa		
Descripción	Una empresa no solo está compuesta de su nombre, sino de otros datos, departamentos y personal que la conforman.	
Actores principal	Administrador, Autorizador y RP.	
Pre-condiciones	El usuario se encuentra visualizando el conjunto de empresas registradas en el sistema y desea acceder a todos los datos de una de ellas.	
Flujos	Básico	<ol style="list-style-type: none">1. El usuario selecciona la empresa que quiere consultar.2. El sistema se encarga de mostrarle todos los datos secundarios, departamentos y personal que la conforman.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario consigue su objetivo: poder consultar a través del monitor todos los datos que componen la empresa seleccionada.	

Caso de uso 14: Visualizar empresa



Caso de uso 15: Añadir empresa	
Descripción	Como ya comentamos con anterioridad, el conjunto de empresas que trabaja en el centro es muy variable, lo que hace que debamos ofrecer al usuario una vía para insertar nuevas.
Actores principal	Administrador y Autorizador.
Pre-condiciones	El usuario se encuentra en el sistema, más concretamente en la tab de empresas y personal, y desea registrar una nueva empresa.
Flujos	Básico <ol style="list-style-type: none">1. El usuario pulsa el botón de 'Añadir empresa'.2. El sistema le proporciona un formulario con todos los campos que es necesario rellenar para dar de alta una nueva empresa.3. El usuario los rellena y pide al sistema que realice el alta.4. Éste le contesta mostrándole un mensaje de aviso con la operación que se encuentra a punto de realizar y solicitándole una confirmación previa para ello.5. El usuario la da. (*)6. El sistema registra la nueva empresa en el sistema. (*)
	Alternativos (*) <p>El usuario no da el visto bueno a la inserción:</p> <ul style="list-style-type: none">• El sistema no realiza ninguna acción y vuelve a la mitad del tercer punto sin haber modificado ningún campo. <p>El usuario introduce algún campo con formato erróneo:</p> <ul style="list-style-type: none">• El sistema le avisa de su error y queda a la espera de que continúe con el caso de uso o inicie otro.
Post-condiciones	El usuario consigue registrar la empresa en el sistema, pasando a ser ésta operativa a partir de ahora.

Caso de uso 15: Añadir empresa



Caso de uso 16: Borrar empresa		
Descripción	Así como se debe poder añadir empresas, también debe ser posible eliminarlas ya que no formarán parte del centro de por vida.	
Actores principal	Administrador y Autorizador.	
Pre-condiciones	Una o varias empresas dejan de formar parte del centro, lo que hace que el usuario se logué en el sistema con la intención de eliminarlas.	
Flujos	Básico	<ol style="list-style-type: none">1. El usuario selecciona del conjunto de empresas que el sistema muestra aquellas que desea eliminar y traslada al sistema su intención.2. Éste le pide que confirme la operación para que el borrado tenga lugar.3. El usuario responde 'Sí' y el caso de uso cumple su objetivo. (*)
	Alternativos (*)	El usuario aborta el borrado: <ul style="list-style-type: none">• El sistema no realiza ninguna acción y el caso de uso concluye.
Post-condiciones	El sistema ya no almacena el conjunto de empresas que el usuario mandó eliminar.	

Caso de uso 16: Borrar empresa



Caso de uso 17: Modificar empresa	
Descripción	No es extraño que una empresa pueda cambiar de dirección, o que veamos oportuno modificar su código, nombre,...
Actores principal	Administrador y Autorizador.
Pre-condiciones	El usuario se encuentra visualizando el conjunto de empresas existentes en el sistema y tiene intención de modificar los datos de alguna de ellas.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario selecciona la empresa a modificar. 2. El sistema le proporciona un formulario con los datos que posee actualmente. 3. El usuario edita todos los que estime erróneos y pide al sistema que lleve a cabo la actualización. 4. El sistema envía al usuario un diálogo indicándole que necesita una autorización previa para poder realizar la acción que demanda. 5. El usuario da su visto bueno. (*) 6. El sistema actualiza la empresa con los datos proporcionados. (*)
	<p>Alternativos (*)</p> <p>El usuario responde 'No' ante la advertencia del sistema:</p> <ul style="list-style-type: none"> • El caso de uso concluye sin lograr el fin para el que fue ideado. <p>Alguno de los datos introducidos por el usuario es erróneo:</p> <ul style="list-style-type: none"> • El sistema le avisa del error acontecido y permanece a la espera de nuevas acciones sin haber realizado ningún tipo de actualización.
Post-condiciones	La empresa que al comienzo de este caso de uso constaba con datos erróneos ahora lo hace con válidos.

Caso de uso 17: Modificar empresa



Caso de uso 18: Listar empresas		
Descripción	El usuario nos indica explícitamente que desea disponer de una vía con la que poder obtener un listado de todas las empresas participantes en la vida del centro.	
Actores principal	Administrador y Autorizador.	
Pre-condiciones	El usuario ha desplegado la aplicación y desea obtener un listado de las empresas que se encuentra visualizando.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario pide al sistema que genere el listado de empresas que visualiza en pantalla. 2. El sistema le insta a que le indique una ruta en la que depositar el reporte, a lo que el usuario responde ofreciéndosela. (*) 3. El sistema elabora el reporte solicitado y lo deposita donde el usuario indicó.
	Alternativos (*)	<p>El usuario no ofrece una ruta en la que depositar el listado:</p> <ul style="list-style-type: none"> • El sistema da por concluido el caso de uso sin elaborar ningún tipo de informe.
Post-condiciones	El usuario dispone del reporte que solicitó.	

Caso de uso 18: Listar empresas

4.4. Departamentos

Caso de uso 19: Visualizar departamentos		
Descripción	Se pueden realizar dos tipos de visualización de departamento: todos o los pertenecientes a una empresa indicada.	
Actores principal	Administrador, Autorizador y RP.	
Pre-condiciones	El usuario accede a accessManager con la intención de visualizar los departamentos existentes y se sitúa en la pestaña de empresas y personal.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario se sitúa en la pestaña de empresas y personal e indica al sistema que le muestre todos los departamentos. (*) 2. El sistema accede a la base de datos, recoge los que necesite y se los muestra por pantalla.
	Alternativos (*)	<p>El usuario selecciona una empresa concreta y/o redefine la búsqueda:</p> <ul style="list-style-type: none"> • El sistema continúa en el punto 2 pero obteniendo solo aquellos departamentos pertenecientes a la empresa seleccionada y/o que cumplan con los filtros de búsqueda indicados.
Post-condiciones	El usuario obtiene por pantalla todos los departamentos y personas que solicitó a través de los diversos campos de búsqueda.	

Caso de uso 19: Visualizar departamentos



Caso de uso 20: Visualizar departamento		
Descripción	Si alguien desea obtener toda la información de que consta un departamento, nosotros le ofreceremos una vía para ello.	
Actores principal	Administrador, Autorizador y RP.	
Pre-condiciones	El usuario loguea en el sistema y se sitúa en la tab 'Empresas y personal'.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario restringe la visualización de departamentos hasta conseguir visualizar el que desee. Lo selecciona e indica al sistema que muestre toda la información que disponga de éste. 2. El sistema accede a la base de datos y muestra por pantalla todos los datos de los que consta el departamento, así como el personal que forma parte de él.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario visualiza por pantalla toda la información que buscaba.	

Caso de uso 20: Visualizar departamento

Caso de uso 21: Añadir departamento		
Descripción	Al igual que las empresas varían, los departamentos también lo harán dentro de ellas.	
Actores principal	Administrador y Autorizador.	
Pre-condiciones	El usuario accede al sistema para dar de alta un nuevo departamento.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona la empresa a la que pertenecerá el departamento y pide al sistema añadir uno nuevo. 2. El sistema le ofrece un formulario que deberá rellenar. 3. El usuario obedece y tras concluir de editarlo pide al sistema que realice el alta. 4. El sistema responde con un mensaje de aviso, donde informa al usuario la operación que va a realizar y pidiéndole que certifique si es la deseada. 5. El usuario responde afirmativamente y el departamento es registrado en el sistema. (*)
	Alternativos (*)	<p>El usuario responde negativamente:</p> <ul style="list-style-type: none"> • El sistema no realiza ningún registro y el caso de uso finaliza. <p>El usuario introduce algún dato erróneo:</p> <ul style="list-style-type: none"> • El sistema le comunica el error que ha realizado y permanece a la espera de una nueva inserción o el inicio de otro caso de uso.
Post-condiciones	El departamento suministrado pasa a formar parte del sistema.	

Caso de uso 21: Añadir departamento



Caso de uso 22: Borrar departamento	
Descripción	Así como se pueden añadir departamentos, también deberá ser posible poder eliminarlos.
Actores principal	Administrador y Autorizador.
Pre-condiciones	El usuario visualiza en pantalla algunos departamentos que no debieran estar registrados, lo que hace que desee eliminarlos.
Flujos	Básico <ol style="list-style-type: none">1. El usuario selecciona los departamentos que desea eliminar y pide al sistema que los borre.2. El sistema insta al usuario a que confirme que desea realizar el borrado de los departamentos seleccionados.3. El usuario da el visto bueno ya que ve correcta la operación que se va a realizar. (*)4. El sistema elimina de la base de datos todos los departamentos indicados.
	Alternativos (*) <p>El usuario decide abortar el borrado por no estimarlo oportuno o haber seleccionado algún departamento que no debía:</p> <ul style="list-style-type: none">• El sistema da por concluido el caso de uso sin realizar ninguna operación sobre la base de datos.
Post-condiciones	Los departamentos que el usuario seleccionó ya no forman parte del sistema.

Caso de uso 22: Borrar departamento



Caso de uso 23: Modificar departamento		
Descripción	Acompañamos las operaciones de adición y borrado con la de modificación.	
Actores principal	Administrador y Autorizador.	
Pre-condiciones	El usuario se encuentra dentro del sistema visualizando los departamentos de una empresa y estima oportuno modificar uno de ellos ya que los datos que contiene no son correctos.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona la empresa y departamento a modificar y pide al sistema poder actualizarlo. 2. El sistema le proporciona todos los datos que lo conforman actualmente. 3. El usuario modifica los que no cree correctos y pasa el testigo al sistema. 4. El sistema realiza la actualización siempre y cuando el usuario dé su visto bueno. (*)
	Alternativos (*)	<p>El usuario no lo da:</p> <ul style="list-style-type: none"> • El sistema no realiza la actualización y queda a la espera de nuevas órdenes. <p>Alguno de los campos que el usuario ha editado consta con datos con formato erróneo:</p> <ul style="list-style-type: none"> • El sistema avisa al usuario del error que acaba de cometer y permanece al final del tercer punto de este caso de uso.
Post-condiciones	El usuario consigue que el departamento conste con los datos deseados.	

Caso de uso 23: Modificar departamento

Caso de uso 24: Listar departamentos		
Descripción	Caso de uso idéntico al 18, pero tomando como referencia los departamentos.	
Actores principal	Administrador y Autorizador.	
Pre-condiciones	El usuario despliega accessManager con la intención de obtener un listado de departamentos.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario se sitúa en la segunda pestaña, define la búsqueda de departamentos y pide al sistema que genere un reporte con los visualizados por pantalla. 2. El sistema obedece, pero previamente el usuario deberá ofrecerle una ruta donde depositarlo. (*)
	Alternativos (*)	<p>El usuario no ofrece ninguna ruta:</p> <ul style="list-style-type: none"> • El sistema salta al final del caso de uso sin elaborar ningún informe.
Post-condiciones	El usuario puede disfrutar del reporte solicitado en la ubicación que indicó durante el transcurso del caso de uso.	

Caso de uso 24: Listar departamentos



4.5. Personal

Caso de uso 25: Visualizar personal					
Descripción	Los últimos entes a tratar son las personas que forman parte de las empresas y departamentos anteriormente citados. En primer lugar facilitaremos la visualización de éste en función de los parámetros de búsqueda proporcionados.				
Actores principal	Administrador, Autorizador y RP.				
Pre-condiciones	El usuario de la aplicación accede a ella con la intención de visualizar el conjunto de personas que cumplen unas características concretas.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> <ol style="list-style-type: none"> 1. El usuario se sitúa en la pestaña de empresas y personal y define el conjunto de personas que desea visualizar (totales, pertenecientes a una empresa y/o pertenecientes a un departamento,...). 2. El sistema accede a la base de datos para mostrar por pantalla todos los datos que obtiene en función de los parámetros de búsqueda proporcionados. </td> </tr> <tr> <td>Alternativos (*)</td> <td>Ninguno.</td> </tr> </table>	Básico	<ol style="list-style-type: none"> 1. El usuario se sitúa en la pestaña de empresas y personal y define el conjunto de personas que desea visualizar (totales, pertenecientes a una empresa y/o pertenecientes a un departamento,...). 2. El sistema accede a la base de datos para mostrar por pantalla todos los datos que obtiene en función de los parámetros de búsqueda proporcionados. 	Alternativos (*)	Ninguno.
	Básico	<ol style="list-style-type: none"> 1. El usuario se sitúa en la pestaña de empresas y personal y define el conjunto de personas que desea visualizar (totales, pertenecientes a una empresa y/o pertenecientes a un departamento,...). 2. El sistema accede a la base de datos para mostrar por pantalla todos los datos que obtiene en función de los parámetros de búsqueda proporcionados. 			
Alternativos (*)	Ninguno.				
Post-condiciones	El usuario puede disfrutar en pantalla del conjunto de personas que quería.				

Caso de uso 25: Visualizar personal

Caso de uso 26: Visualizar persona					
Descripción	La muestra de los principales datos del personal puede resultar escasa para las pretensiones del usuario. Con este caso de uso le permitiremos que visualice toda la información de la persona que desee.				
Actores principal	Administrador.				
Pre-condiciones	El usuario se encuentra logueado en el sistema y con la necesidad de visualizar todos los datos correspondientes a una persona mostrada en pantalla.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> <ol style="list-style-type: none"> 1. El usuario selecciona la persona que desea consultar y pide al sistema que le proporcione todos los datos que disponga de ella. 2. El sistema obedece, recuperando toda la información oportuna de la base de datos y proporcionándole un formulario elaborado con ésta. </td> </tr> <tr> <td>Alternativos (*)</td> <td>Ninguno.</td> </tr> </table>	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona la persona que desea consultar y pide al sistema que le proporcione todos los datos que disponga de ella. 2. El sistema obedece, recuperando toda la información oportuna de la base de datos y proporcionándole un formulario elaborado con ésta. 	Alternativos (*)	Ninguno.
	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona la persona que desea consultar y pide al sistema que le proporcione todos los datos que disponga de ella. 2. El sistema obedece, recuperando toda la información oportuna de la base de datos y proporcionándole un formulario elaborado con ésta. 			
Alternativos (*)	Ninguno.				
Post-condiciones	El usuario consigue su objetivo: visualizar por pantalla todos los datos pertenecientes a la persona seleccionada.				

Caso de uso 26: Visualizar persona



Caso de uso 27: Añadir persona					
Descripción	Cuando una nueva persona entra a formar parte de algún tipo de empresa-departamento, ésta deberá ser registrada en el sistema si deseamos poder manejarla en las diferentes funcionalidades que ofrecerá accessManager.				
Actores principal	Administrador, Autorizador y RP.				
Pre-condiciones	Una nueva persona entra a formar parte de la vida diaria del centro, lo que lleva al usuario a querer registrarla.				
Flujos	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="background-color: #f2f2f2; width: 20%;">Básico</td> <td> <ol style="list-style-type: none"> 1. El usuario accede al sistema y se sitúa en la pestaña 'Empresas y personal'. A continuación selecciona la empresa (y departamento) que corresponda e indica al sistema que desea añadirle una nueva persona. 2. El sistema le ofrece un formulario diseñado en función del tipo de persona a insertar; formulario que el usuario rellena y devuelve. 3. El sistema pide al usuario que dé su aprobación a la operación que se encuentra a punto de ejecutarse. 4. El usuario la da. (*) 5. El sistema registra la nueva persona en la empresa (y departamento) indicada. (*) </td> </tr> <tr> <td style="background-color: #f2f2f2;">Alternativos (*)</td> <td> <p>El usuario no ve oportuno realizar el alta:</p> <ul style="list-style-type: none"> • El sistema no la lleva a cabo y permanece a la espera de nuevas órdenes. <p>El usuario proporciona algún dato erróneo:</p> <ul style="list-style-type: none"> • El sistema le avisa del error que acaba de cometer y pasa al final del punto 2. </td> </tr> </table>	Básico	<ol style="list-style-type: none"> 1. El usuario accede al sistema y se sitúa en la pestaña 'Empresas y personal'. A continuación selecciona la empresa (y departamento) que corresponda e indica al sistema que desea añadirle una nueva persona. 2. El sistema le ofrece un formulario diseñado en función del tipo de persona a insertar; formulario que el usuario rellena y devuelve. 3. El sistema pide al usuario que dé su aprobación a la operación que se encuentra a punto de ejecutarse. 4. El usuario la da. (*) 5. El sistema registra la nueva persona en la empresa (y departamento) indicada. (*) 	Alternativos (*)	<p>El usuario no ve oportuno realizar el alta:</p> <ul style="list-style-type: none"> • El sistema no la lleva a cabo y permanece a la espera de nuevas órdenes. <p>El usuario proporciona algún dato erróneo:</p> <ul style="list-style-type: none"> • El sistema le avisa del error que acaba de cometer y pasa al final del punto 2.
Básico	<ol style="list-style-type: none"> 1. El usuario accede al sistema y se sitúa en la pestaña 'Empresas y personal'. A continuación selecciona la empresa (y departamento) que corresponda e indica al sistema que desea añadirle una nueva persona. 2. El sistema le ofrece un formulario diseñado en función del tipo de persona a insertar; formulario que el usuario rellena y devuelve. 3. El sistema pide al usuario que dé su aprobación a la operación que se encuentra a punto de ejecutarse. 4. El usuario la da. (*) 5. El sistema registra la nueva persona en la empresa (y departamento) indicada. (*) 				
Alternativos (*)	<p>El usuario no ve oportuno realizar el alta:</p> <ul style="list-style-type: none"> • El sistema no la lleva a cabo y permanece a la espera de nuevas órdenes. <p>El usuario proporciona algún dato erróneo:</p> <ul style="list-style-type: none"> • El sistema le avisa del error que acaba de cometer y pasa al final del punto 2. 				
Post-condiciones	El usuario consigue realizar el alta en el sistema de la nueva persona.				

Caso de uso 27: Añadir persona



Caso de uso 28: Borrar persona	
Descripción	El borrado de personas es tan necesario como la adición, ya que durante la relación que mantengan Informática Navarra S.A. y cualquier empresa-departamento, dentro de ésta variará el flujo de personal destinado al centro.
Actores principal	Administrador, Autorizador y RP.
Pre-condiciones	Un conjunto de personas deja de acudir de manera definitiva al centro, lo que provoca que cualquiera de los tres actores quiera darlas de baja en el sistema.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario despliega accessManager y se dirige a la tab de empresas y personal. Una vez allí selecciona la empresa (y departamento) y el conjunto de personas que desea eliminar y pulsa el botón de borrado. 2. El sistema genera un mensaje para avisar al usuario de la operación que está a punto de ejecutar y le pide que la ratifique. 3. El usuario responde afirmativamente y el borrado de las personas seleccionadas es llevado a cabo. (*)
	<p>Alternativos (*)</p> <ul style="list-style-type: none"> • El usuario responde negativamente: <ul style="list-style-type: none"> • El sistema no realiza ninguna acción y da el caso de uso por finalizado.
Post-condiciones	El conjunto de personas que el usuario mandó borrar ya no forman parte del sistema.

Caso de uso 28: Borrar persona



Caso de uso 29: Modificar persona		
Descripción	Éste sí que será un caso de uso menos frecuente ya que el cambio de alguno de los datos que conforman una persona será un hecho aislado. Aun así, como puede ser necesario y el cliente nos los exige, implementaremos un caso de uso que dé solución a esta situación.	
Actores principal	Administrador, Autorizador y RP.	
Pre-condiciones	El usuario se encuentra en el sistema, más concretamente en la tab de empresas y personal, y visualiza una persona que consta con datos erróneos.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona la persona e indica al sistema que desea actualizar sus datos. 2. El sistema le proporciona un formulario con todos los datos que dispone de ésta actualmente. 3. El usuario modifica los datos que crea convenientes y pide al sistema que lleve a cabo la actualización. 4. Éste obedece siempre y cuando el usuario dé su visto bueno. (*)
	Alternativos (*)	<p>El usuario decide abortar la actualización:</p> <ul style="list-style-type: none"> • El sistema no modifica ningún registro de la base de datos y el caso de uso queda estancado a mitad del punto 3. <p>El usuario proporciona algún dato erróneo:</p> <ul style="list-style-type: none"> • El sistema no puede llevar a cabo la actualización solicitada, lo que hace que muestre un mensaje con el error cometido y quede a la espera de nuevas órdenes.
Post-condiciones	La actualización que el usuario buscaba ha sido llevada a buen puerto.	

Caso de uso 29: Modificar persona



Caso de uso 30: Añadir teléfono	
Descripción	accessManager permitirá llevar a cabo un registro de todos los números de teléfono que una persona disponga.
Actores principal	Administrador, Autorizador y RP.
Pre-condiciones	El usuario se encuentra en la pestaña 'Empresas y personal' y desea añadir un nuevo número de teléfono a una de las personas que visualiza.
Flujos	Básico <ol style="list-style-type: none"> 1. El usuario selecciona la persona e introduce el nuevo número de teléfono, indicando al sistema que lo registre en la base de datos. 2. El sistema obedece. (*)
	Alternativos (*) <p>El sistema no puede realizar la inserción ya que el número de teléfono proporcionado no dispone del formato deseado:</p> <ul style="list-style-type: none"> • El sistema avisa al usuario del error que ha cometido y le pide que corrija el número de teléfono que desea introducir, permaneciendo a la espera.
Post-condiciones	La persona seleccionada pasa a disponer de un nuevo número de teléfono registrado en el sistema.

Caso de uso 30: Añadir teléfono

Caso de uso 31: Borrar teléfono	
Descripción	En el momento de que una persona ya no disponga de un teléfono que fue registrado en el sistema será conveniente proceder con su eliminación, pues no tiene sentido almacenar información innecesaria.
Actores principal	Administrador, Autorizador y RP.
Pre-condiciones	Una persona deja de poseer un número de teléfono, disponiéndose el usuario a eliminarlo.
Flujos	Básico <ol style="list-style-type: none"> 1. El usuario loguea en el sistema, se sitúa en la pestaña de empresas y personal y define la búsqueda hasta dar con la persona deseada. Acto seguido selecciona de entre los existentes el conjunto de números de teléfono que desea borrar y clicla en el botón de borrado. 2. El sistema elimina los teléfonos del sistema previa autorización por parte del usuario. (*)
	Alternativos (*) <p>El usuario no da su autorización:</p> <ul style="list-style-type: none"> • No tiene lugar ningún borrado y el caso de uso concluye.
Post-condiciones	La persona seleccionada ya no dispone de los teléfonos eliminados entre los registrados.

Caso de uso 31: Borrar teléfono



Caso de uso 32: Añadir E-Mail					
Descripción	Nos encontramos en la misma situación que con los teléfonos, pero en este caso lo que trataremos serán las direcciones de correo electrónico del personal.				
Actores principal	Administrador, Autorizador y RP.				
Pre-condiciones	Cualquiera de los tres actores se encuentra con la aplicación accessManager desplegada y se dispone a asignar una nueva dirección de correo electrónico a una de las personas ya registradas.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> <ol style="list-style-type: none"> 1. El usuario accede a la segunda tab y selecciona la empresa (y departamento) que le lleve a visualizar a la persona deseada. La selecciona e indica el nuevo correo electrónico a registrar. 2. El sistema registra el E-Mail y se lo asigna a la persona seleccionada. (*) </td> </tr> <tr> <td>Alternativos (*)</td> <td> <p>No puede realizarse la adición ya que el E-Mail proporcionado posee un formato no válido:</p> <ul style="list-style-type: none"> • El sistema avisa al usuario de su error y permanece a la espera de que corrija la dirección o pase a ejecutar otro caso de uso. </td> </tr> </table>	Básico	<ol style="list-style-type: none"> 1. El usuario accede a la segunda tab y selecciona la empresa (y departamento) que le lleve a visualizar a la persona deseada. La selecciona e indica el nuevo correo electrónico a registrar. 2. El sistema registra el E-Mail y se lo asigna a la persona seleccionada. (*) 	Alternativos (*)	<p>No puede realizarse la adición ya que el E-Mail proporcionado posee un formato no válido:</p> <ul style="list-style-type: none"> • El sistema avisa al usuario de su error y permanece a la espera de que corrija la dirección o pase a ejecutar otro caso de uso.
	Básico	<ol style="list-style-type: none"> 1. El usuario accede a la segunda tab y selecciona la empresa (y departamento) que le lleve a visualizar a la persona deseada. La selecciona e indica el nuevo correo electrónico a registrar. 2. El sistema registra el E-Mail y se lo asigna a la persona seleccionada. (*) 			
Alternativos (*)	<p>No puede realizarse la adición ya que el E-Mail proporcionado posee un formato no válido:</p> <ul style="list-style-type: none"> • El sistema avisa al usuario de su error y permanece a la espera de que corrija la dirección o pase a ejecutar otro caso de uso. 				
Post-condiciones	La persona seleccionada dispone de una nueva dirección de correo electrónico registrada.				

Caso de uso 32: Añadir E-Mail

Caso de uso 33: Borrar E-Mail					
Descripción	El borrado de cualquier dirección de correo electrónico es semejante al realizado con los números de teléfono.				
Actores principal	Administrador, Autorizador y RP.				
Pre-condiciones	El usuario se encuentra visualizando una persona y observa que posee direcciones E-Mail que ha dejado de utilizar y se dispone a borrarlos.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> <ol style="list-style-type: none"> 1. El usuario selecciona el conjunto de direcciones que desea eliminar y clics en el botón de borrado. 2. El sistema se encarga de hacerlas desaparecer de la base de datos si el usuario da su visto bueno. (*) </td> </tr> <tr> <td>Alternativos (*)</td> <td> <p>El usuario no lo da:</p> <ul style="list-style-type: none"> • El sistema no realiza ningún borrado en la base de datos y da el caso de uso por finalizado, permaneciendo a la espera de nuevas órdenes. </td> </tr> </table>	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona el conjunto de direcciones que desea eliminar y clics en el botón de borrado. 2. El sistema se encarga de hacerlas desaparecer de la base de datos si el usuario da su visto bueno. (*) 	Alternativos (*)	<p>El usuario no lo da:</p> <ul style="list-style-type: none"> • El sistema no realiza ningún borrado en la base de datos y da el caso de uso por finalizado, permaneciendo a la espera de nuevas órdenes.
	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona el conjunto de direcciones que desea eliminar y clics en el botón de borrado. 2. El sistema se encarga de hacerlas desaparecer de la base de datos si el usuario da su visto bueno. (*) 			
Alternativos (*)	<p>El usuario no lo da:</p> <ul style="list-style-type: none"> • El sistema no realiza ningún borrado en la base de datos y da el caso de uso por finalizado, permaneciendo a la espera de nuevas órdenes. 				
Post-condiciones	El borrado de direcciones E-Mail ha sido realizado correctamente.				

Caso de uso 33: Borrar E-Mail



Caso de uso 34: Añadir cargo	
Descripción	En este caso haremos alusión a las personas con las que Informática Navarra S.A. mantiene contacto.
Actores principal	Administrador, Autorizador y RP.
Pre-condiciones	El usuario se encuentra logueado en accessManager en disposición de asignar un nuevo conjunto de cargos a uno de los contactos existentes.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario selecciona la empresa y departamento en la que se encuentra encuadrado el contacto para así visualizarlo en la tabla destinada a éstos. Lo selecciona y pide al sistema poder añadirle un nuevo cargo. 2. El sistema le ofrece la lista de cargos existentes, seleccionando el usuario los que desee añadir. 3. El sistema pide al usuario que confirme la operación de adición que va a ser realizada, tras la cual el sistema añade los cargos indicados al contacto seleccionado y da el caso de uso por finalizado. (*)
	<p>Alternativos (*)</p> <p>El usuario aborta la operación de adición:</p> <ul style="list-style-type: none"> • El sistema no realiza ninguna acción y pasa al final del segundo punto del flujo básico.
Post-condiciones	El contacto pasa a disponer de los cargos que poseía junto con los nuevos asignados.

Caso de uso 34: Añadir cargo

Caso de uso 35: Borrar cargo	
Descripción	No es muy común, pero sí que es verdad que las responsabilidades de cualquier contacto con respecto a Informática Navarra S.A. pueden variar. De ahí el origen de este caso de uso.
Actores principal	Administrador, Autorizador y RP.
Pre-condiciones	El usuario, logueado en el sistema y visualizando el conjunto de contactos de una empresa-departamento percibe que un contacto dispone de algún tipo de cargo que ya no desempeña y se dispone a eliminarlo.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario selecciona el conjunto de cargos desempeñados por un contacto que no deberían constar en el sistema y pulsa sobre el botón de borrado. 2. El sistema se encarga de eliminarlos y dar por finalizado exitosamente el caso de uso solo sí el usuario a dado su autorización previa. (*)
	<p>Alternativos (*)</p> <p>El usuario contesta 'No' ante el diálogo de aviso:</p> <ul style="list-style-type: none"> • El sistema no borra ningún cargo, dando por concluido el caso de uso en ejecución.
Post-condiciones	El contacto ya no dispone de cargos que no desempeña en la actualidad.

Caso de uso 35: Borrar cargo



Caso de uso 36: Listar personal		
Descripción	Para terminar los casos de uso que implementaremos en esta pestaña, el cliente nos pide que le proporcionemos una vía para generar un listado con el personal que se encuentre visualizando en pantalla.	
Actores principal	Administrador, Autorizador y RP.	
Pre-condiciones	El usuario se encuentra situado en la pestaña 'Empresas y personal' y ve interesante generar un informe con las personas que visualiza.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario pide al sistema que elabore un reporte con los usuarios que muestra en la tabla de personal. 2. Éste le responde pidiéndole que indique una ruta en la que depositar el informe solicitado. 3. El usuario la ofrece y el sistema se encarga de crear y depositar el reporte en ésta. (*)
	Alternativos (*)	<p>El usuario no proporciona ninguna ruta, cancelando el cuadro de selección:</p> <ul style="list-style-type: none"> • El sistema no elabora ningún informe y el caso de uso llega a su fin.
Post-condiciones	El usuario dispone del informe solicitado en la ruta indicada.	

Caso de uso 36: Listar personal

4.6. Niveles

Caso de uso 37: Visualizar niveles		
Descripción	Todos los niveles deben poder ser visualizados de forma directa, de un solo vistazo.	
Actores principal	Administrador.	
Pre-condiciones	El administrador se encuentra logueado en el sistema y desea consultar los niveles existentes en el sistema.	
Flujos	Básico	<ol style="list-style-type: none"> 5. El usuario accede al apartado de 'Niveles'. 6. El sistema muestra todos los niveles existentes, además de otros datos.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario conoce todos los niveles existentes.	

Caso de uso 37: Visualizar niveles



Caso de uso 38: Visualizar nivel		
Descripción	El caso de uso anterior nos daba la posibilidad de visualizar los principales datos de todos los niveles existentes de un vistazo. En esta ocasión posibilitaremos que el usuario pueda observar todos los detalles de un nivel seleccionado.	
Actores principal	Administrador.	
Pre-condiciones	El administrador desea consultar todos los detalles de uno de los niveles que observa en la tabla de niveles.	
Flujos	Básico	<ol style="list-style-type: none"> 3. El usuario pulsa el nivel del que desea consultar sus datos. 4. El sistema rescata todos los datos que posee del nivel y se los muestra al usuario.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario disfruta de todos los detalles que buscaba.	

Caso de uso 38: Visualizar nivel

Caso de uso 39: Añadir nivel		
Descripción	Si surge un nuevo nivel en el centro, debemos proporcionar la posibilidad de poder registrarlo en el sistema.	
Actores principal	Administrador.	
Pre-condiciones	El administrador desea almacenar un nuevo nivel en el sistema.	
Flujos	Básico	<ol style="list-style-type: none"> 7. El usuario indica al sistema que quiere registrar un nuevo nivel. 8. El sistema proporciona al usuario un formulario con el que dar de alta el nuevo nivel. 9. El usuario rellena el formulario, pide al sistema que realice el alta y aprueba el aviso previo. (*) 10. Si todo es correcto se lleva a cabo el alta del nuevo nivel. (*)
	Alternativos (*)	<p>El usuario no da su conformidad a que se lleve a cabo el alta del nivel:</p> <ul style="list-style-type: none"> • El sistema aborta el proceso y finaliza el caso de uso. <p>Los datos introducidos ya existen o son incorrectos:</p> <ul style="list-style-type: none"> • El sistema avisa al usuario de esta situación y retorna al punto 2, sin haber modificado los campos redactados.
Post-condiciones	El administrador consigue registrar el nuevo nivel en el sistema.	

Caso de uso 39: Añadir nivel



Caso de uso 40: Borrar nivel		
Descripción	Si existe un nivel que no debiera constar en la base de datos, el usuario podrá seleccionarlo y pedir al sistema que lo elimine.	
Actores principal	Administrador.	
Pre-condiciones	El administrador se encuentra logueado en el sistema y en la pestaña de niveles observa que existe alguno que no debiera, con que se dispone a eliminarlo.	
Flujos	Básico	4. El usuario selecciona los niveles que no debieran existir y pide al sistema que los elimine. 5. El sistema pide al usuario que confirme que los niveles seleccionados son los que realmente quiere eliminar. 6. Los niveles seleccionados son eliminados si el usuario da su conformidad. (*)
	Alternativos (*)	El usuario no da su conformidad al borrado: <ul style="list-style-type: none"> • El sistema no realiza acción alguna.
Post-condiciones	El administrador ha conseguido eliminar los niveles que no debían constar en el sistema.	

Caso de uso 40: Borrar nivel

Caso de uso 41: Modificar nivel		
Descripción	Un nivel puede variar a lo largo de su existencia, con que debe ser posible modificar los valores de los atributos que lo componen.	
Actores principal	Administrador.	
Pre-condiciones	El usuario desea modificar los datos de un nivel existente.	
Flujos	Básico	7. El administrador selecciona el nivel a modificar y el sistema despliega los datos de éste. 8. Se modifican los datos oportunos y se pide al sistema que lleve a cabo la actualización. 9. El sistema realiza la actualización del nivel en la base de datos si el administrador da su conformidad. (*)
	Alternativos (*)	El administrador no da el visto bueno o introduce los datos con un formato no válido: <ul style="list-style-type: none"> • El sistema avisa del error y no realiza modificación alguna.
Post-condiciones	El usuario consigue actualizar el nivel indicado y que conste como deseaba.	

Caso de uso 41: Modificar nivel



Caso de uso 42: Dar de baja nivel		
Descripción	Si un nivel deja de existir y se quiere seguir manteniendo la trazabilidad, debe existir una forma de darlo de baja de modo que permanezca inoperativo pero que siga constando en el sistema.	
Actores principal	Administrador.	
Pre-condiciones	El administrador se encuentra logueado en el sistema, en la tab de niveles y dispuesto a dar de baja a alguno de ellos.	
Flujos	Básico	<ol style="list-style-type: none"> 4. El usuario selecciona los niveles que desea dar de baja y traslada dicha petición al sistema. 5. Éste pide al usuario que confirme que los niveles seleccionados son los que realmente quiere dar de baja. (*) 6. El sistema da de baja de todos los niveles indicados tras recibir la confirmación.
	Alternativos (*)	El administrador no da su conformidad: <ul style="list-style-type: none"> • El sistema no realiza ninguna baja.
Post-condiciones	El sistema ha dado de baja a todos los niveles seleccionados por el administrador.	

Caso de uso 42: Dar de baja nivel

4.7. Colores

Caso de uso 43: Visualizar colores		
Descripción	Pasamos de los niveles a los colores. Las condiciones en las que deberán visualizarse todos los colores existentes son análogas a las de los niveles: todos de un vistazo, fácilmente.	
Actores principal	Administrador.	
Pre-condiciones	El usuario desea observar qué colores se encuentran registrados en el sistema.	
Flujos	Básico	<ol style="list-style-type: none"> 3. El administrador accede al apartado de 'Colores'. 4. El sistema muestra todos los colores existentes.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario visualiza todos los colores activos.	

Caso de uso 43: Visualizar colores



Caso de uso 44: Visualizar color		
Descripción	Quizá al usuario no le baste con observar los datos principales de cada color, sino que desee observar todos los datos de uno concreto.	
Actores principal	Administrador.	
Pre-condiciones	El administrador se encuentra en la tab de 'Colores' y desea conocer todos los datos de uno de ellos.	
Flujos	Básico	<ol style="list-style-type: none"> 3. El usuario selecciona un color e insta al sistema a que le muestre todos los detalles del mismo. 4. El sistema rescata todos los datos que posee del color y se los muestra al usuario por pantalla.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario puede consultar los datos que desee.	

Caso de uso 44: Visualizar color

Caso de uso 45: Añadir color		
Descripción	No es lo usual, pero siempre puede darse el caso de que llegue al centro un nuevo juego de tarjetas, con que tenemos que tener la posibilidad de poder añadir nuevos colores para así luego registrar las tarjetas.	
Actores principal	Administrador.	
Pre-condiciones	El administrador se encuentra logueado y desea introducir un nuevo color.	
Flujos	Básico	<ol style="list-style-type: none"> 6. El usuario pide al sistema introducir un nuevo color. 7. El sistema le proporciona un formulario con el que registrarlo. 8. El usuario rellena todos los campos e indica al sistema que realice el alta. 9. El caso de uso termina registrando el sistema el nuevo color en el caso de que el usuario haya dado su visto bueno. (*)
	Alternativos (*)	<p>El usuario no da su visto bueno:</p> <ul style="list-style-type: none"> • El sistema no realiza ninguna acción y el caso de uso termina.
Post-condiciones	El administrador consigue dar de alta el color que deseaba.	

Caso de uso 45: Añadir color



Caso de uso 46: Borrar color	
Descripción	Nos ocupa el caso contrario: eliminar algún color debido a que ya no tiene sentido seguir almacenándolo (perdiendo la consecuente trazabilidad).
Actores principal	Administrador.
Pre-condiciones	El usuario se encuentra en la pestaña de tarjetas y se percata de que existen colores que no debieran, con lo que se dispone a eliminarlos.
Flujos	Básico <ol style="list-style-type: none"> 5. El administrador selecciona los colores que desea eliminar de la tabla de colores, pidiendo al sistema que los elimine. 6. El sistema solicita al usuario que dé su visto bueno para proceder con el borrado. 7. El usuario acepta y los colores dejan de existir. (*)
	Alternativos (*) <p>El usuario no da su visto bueno:</p> <ul style="list-style-type: none"> • El sistema no borra ningún color y el caso de uso finaliza.
Post-condiciones	El usuario consigue borrar del sistema los colores que se proponía.

Caso de uso 46: Borrar color

Caso de uso 47: Modificar color	
Descripción	Uno de los casos de uso detallados en la <i>Especificación inicial de los casos de uso</i> era que todo color pudiera ser modificado. Esto se debe a que puede darse el caso de que un color (juego de tarjetas) vea modificado su nivel, proporcionando así un menor o mayor grado de acceso, más que el color cambie de coloración.
Actores principal	Administrador.
Pre-condiciones	El usuario necesita modificar los datos de un color, con que abre la aplicación y se dirige a la pestaña de tarjetas.
Flujos	Básico <ol style="list-style-type: none"> 5. El administrador selecciona el color que desea modificar y pide al sistema poder modificarlo. 6. Éste le proporciona todos los datos del color, editando el usuario todo lo que estime oportuno. 7. El administrador traslada al sistema su intención de que haga permanentes los cambios y ratifica el aviso enviado por el sistema. (*)
	Alternativos (*) <p>El usuario no da su visto bueno:</p> <ul style="list-style-type: none"> • El sistema no borra ningún color y el caso de uso concluye.
Post-condiciones	El sistema lleva a cabo la modificación indicada y el usuario puede dar por cumplido su objetivo.

Caso de uso 47: Modificar color



Caso de uso 48: Dar de baja color		
Descripción	Es el turno de desarrollar las bajas de colores.	
Actores principal	Administrador.	
Pre-condiciones	El administrador desea dar de baja algún color activo en el sistema.	
Flujos	Básico	<ol style="list-style-type: none">3. El usuario selecciona el conjunto de colores que desea dar de baja, permaneciendo almacenados en este pero dejando de forma parte del día a día en el centro. Clica en el botón 'Dar de baja'.4. El sistema le envía un mensaje avisándole de que va a proceder con la baja de los colores indicados, a lo que el usuario responde afirmativamente (*).
	Alternativos (*)	El usuario no responde afirmativamente: <ul style="list-style-type: none">• El sistema no procede con la baja de ningún color y vuelve al punto de partida.
Post-condiciones	El administrador logra dar de baja los colores que desea.	

Caso de uso 48: Dar de baja color

4.8. Tarjetas

Caso de uso 49: Visualizar tarjetas		
Descripción	La visualización de las tarjetas sigue la misma lógica que los niveles y los colores. Procedamos con ella.	
Actores principal	Administrador.	
Pre-condiciones	El administrador se encuentra logueado en el sistema y desea visualizar los juegos de tarjetas de los que disponen los vigilantes para llevar a cabo el proceso de accesos.	
Flujos	Básico	<ol style="list-style-type: none">3. El administrador accede al apartado 'Tarjetas'.4. El sistema es el encargado de mostrarle todas las tarjetas existentes en el sistema y que están activas, sin fecha de baja.
	Alternativos (*)	Ninguno.
Post-condiciones	El administrador observa todo el rango de tarjetas existentes.	

Caso de uso 49: Visualizar tarjetas



Caso de uso 50: Visualizar tarjeta		
Descripción	No sólo debe bastar con ver los datos más representativos de cada tarjeta, sino que el administrador debe poder consultar todos los datos de los que esté compuesta.	
Actores principal	Administrador.	
Pre-condiciones	El usuario se encuentra en el tab de tarjetas y se dispone a obtener los datos de una concreta.	
Flujos	Básico	3. El administrador clics sobre la tarjeta que desea consultar y pide al sistema que le proporcione todos los datos que disponga de ésta. 4. El sistema recupera todos los datos posibles y se los muestra al usuario.
	Alternativos (*)	Ninguno.
Post-condiciones	El administrador obtiene por pantalla todos los datos de la tarjeta que deseaba consultar.	

Caso de uso 50: Visualizar tarjeta

Caso de uso 51: Añadir tarjeta		
Descripción	Este hecho sí que puede llegar a ser más habitual, debido a que el constante préstamo de tarjetas puede derivar en la pérdida de éstas, y por tanto en que sea necesario ir introduciendo nuevas.	
Actores principal	Administrador.	
Pre-condiciones	El administrador dispone de una nueva tarjeta a dar de alta en el sistema para que pueda utilizarse.	
Flujos	Básico	6. El usuario clics sobre el botón de 'Dar de alta tarjeta', lo que hace que el sistema muestre un formulario con el que poder llevar a cabo la tarea. 7. Tras disponer del formulario completado y la petición y aprobación del cliente, el sistema puede proceder con el alta de la tarjeta. (*)
	Alternativos (*)	El usuario no da su conformidad con el alta de la tarjeta: <ul style="list-style-type: none"> Se interrumpe el proceso y da por concluido el caso de uso. Los datos introducidos no son válidos (el código de tarjeta no está comprendido entre 0 y 999): <ul style="list-style-type: none"> El sistema avisa al usuario del error ocurrido y se retorna al punto de partida.
Post-condiciones	El administrador ha conseguido dar de alta la tarjeta que deseaba.	

Caso de uso 51: Añadir tarjeta



Caso de uso 52: Borrar tarjeta	
Descripción	Es cierto que lo normal será que el administrador dé de baja las tarjetas y no las borre, pero puede llegar a darse el primer caso.
Actores principal	Administrador.
Pre-condiciones	El usuario desea eliminar alguna de las tarjetas registradas en el sistema.
Flujos	<p>Básico</p> <p>4. El administrador selecciona de la tabla de tarjetas aquellas que desea eliminar y pide al sistema que las elimine.</p> <p>5. Éste procede a su borrado, pero previamente debe de haber recibido una confirmación por parte del administrador. (*)</p>
	<p>Alternativos (*)</p> <p>El administrador opta por abortar la eliminación:</p> <ul style="list-style-type: none"> El sistema no borra ningún dato y el caso de uso finaliza sin más consecuencias.
Post-condiciones	El sistema no dispone de las tarjetas eliminadas por el usuario.

Caso de uso 52: Borrar tarjeta

Caso de uso 53: Modificar tarjeta	
Descripción	No es lo normal, pero también debemos tratar el caso de que una tarjeta pueda cambiar de numeración o color.
Actores principal	Administrador.
Pre-condiciones	El usuario quiere modificar la numeración y/o color de una de las tarjetas.
Flujos	<p>Básico</p> <p>5. El administrador selecciona la tarjeta que desea modificar y traslada al sistema su petición.</p> <p>6. El sistema le muestra todos los datos de ésta; datos que edita el administrador.</p> <p>7. Se produce un paso previo de confirmación y el sistema procede con la actualización de la tarjeta. (*)</p>
	<p>Alternativos (*)</p> <p>El administrador ha introducido datos con el formato equivocado:</p> <ul style="list-style-type: none"> El sistema avisa del error y no actualiza ningún registro. <p>El administrador niega que se realice la modificación:</p> <ul style="list-style-type: none"> El sistema no lleva a cabo ninguna acción y el caso de uso finaliza sin producir cambios.
Post-condiciones	La tarjeta que el usuario seleccionó y modificó queda alterada en la base de datos.

Caso de uso 53: Modificar tarjeta



Caso de uso 54: Dar de baja tarjeta					
Descripción	Ésta utilidad será bastante más utilizada que sus dos predecesoras: todas y cada una de las veces que se extravíe una tarjeta.				
Actores principal	Administrador.				
Pre-condiciones	El administrador desea dar de baja una tarjeta ya que ésta se ha extraviado.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> <ol style="list-style-type: none"> El usuario selecciona un rango de tarjetas y pide al sistema que las dé de baja. El sistema le pregunta al usuario si de verdad quiere dar de baja las tarjetas seleccionadas. El usuario responde que sí. (*) El sistema procede con la baja de todas las tarjetas proporcionadas. </td> </tr> <tr> <td>Alternativos (*)</td> <td> El usuario responde que no: <ul style="list-style-type: none"> El sistema no modifica ningún dato y da por concluido el caso de uso. </td> </tr> </table>	Básico	<ol style="list-style-type: none"> El usuario selecciona un rango de tarjetas y pide al sistema que las dé de baja. El sistema le pregunta al usuario si de verdad quiere dar de baja las tarjetas seleccionadas. El usuario responde que sí. (*) El sistema procede con la baja de todas las tarjetas proporcionadas. 	Alternativos (*)	El usuario responde que no: <ul style="list-style-type: none"> El sistema no modifica ningún dato y da por concluido el caso de uso.
	Básico	<ol style="list-style-type: none"> El usuario selecciona un rango de tarjetas y pide al sistema que las dé de baja. El sistema le pregunta al usuario si de verdad quiere dar de baja las tarjetas seleccionadas. El usuario responde que sí. (*) El sistema procede con la baja de todas las tarjetas proporcionadas. 			
Alternativos (*)	El usuario responde que no: <ul style="list-style-type: none"> El sistema no modifica ningún dato y da por concluido el caso de uso. 				
Post-condiciones	El administrador ha conseguido dar de baja las tarjetas seleccionadas.				

Caso de uso 54: Dar de baja tarjeta

4.9. Tipos de acceso

Caso de uso 55: Visualizar tipos de acceso					
Descripción	El cliente requiere que puedan visualizarse los principales datos de todos los tipos de acceso y todos los datos que conforman cada tipo (caso de uso siguiente).				
Actores principal	Administrador.				
Pre-condiciones	El usuario desea visualizar los principales datos de todos los tipos de acceso existentes.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> <ol style="list-style-type: none"> El usuario se dirige al tab de tarjetas. El sistema muestra al usuario los datos que busca en una tabla. </td> </tr> <tr> <td>Alternativos (*)</td> <td>Ninguno.</td> </tr> </table>	Básico	<ol style="list-style-type: none"> El usuario se dirige al tab de tarjetas. El sistema muestra al usuario los datos que busca en una tabla. 	Alternativos (*)	Ninguno.
	Básico	<ol style="list-style-type: none"> El usuario se dirige al tab de tarjetas. El sistema muestra al usuario los datos que busca en una tabla. 			
Alternativos (*)	Ninguno.				
Post-condiciones	El usuario disfruta de los principales datos de todos los tipos de acceso residentes en la base de datos.				

Caso de uso 55: Visualizar tipos de acceso



Caso de uso 56: Visualizar tipo de acceso					
Descripción	Completamos el caso de uso anterior posibilitando al administrador que pueda visualizar todos los datos que componen un tipo de acceso.				
Actores principal	Administrador.				
Pre-condiciones	El administrador se encuentra logueado en el sistema y acude a la pestaña de tipos de acceso en búsqueda de todos los datos de un tipo de acceso.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> 3. El administrador selecciona un tipo de acceso y pide al sistema que le ofrezca todos los datos de los que disponga. 4. El sistema acude a la base de datos, rescata todos los datos del tipo de acceso y se los hace llegar al usuario a través de un formulario. </td> </tr> <tr> <td>Alternativos (*)</td> <td>Ninguno.</td> </tr> </table>	Básico	3. El administrador selecciona un tipo de acceso y pide al sistema que le ofrezca todos los datos de los que disponga. 4. El sistema acude a la base de datos, rescata todos los datos del tipo de acceso y se los hace llegar al usuario a través de un formulario.	Alternativos (*)	Ninguno.
	Básico	3. El administrador selecciona un tipo de acceso y pide al sistema que le ofrezca todos los datos de los que disponga. 4. El sistema acude a la base de datos, rescata todos los datos del tipo de acceso y se los hace llegar al usuario a través de un formulario.			
Alternativos (*)	Ninguno.				
Post-condiciones	El administrador obtiene todos los datos que estaba buscando.				

Caso de uso 56: Visualizar tipo de acceso

Caso de uso 57: Añadir tipo de acceso					
Descripción	No podemos limitar los tipos de acceso a que sólo existan los iniciales, con que a partir de ahora realizaremos los casos de uso que permitan su manejo.				
Actores principal	Administrador.				
Pre-condiciones	Surge un nuevo tipo de acceso, con lo que el administrador arranca la aplicación y se dirige a la pestaña de tarjetas.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> 3. El usuario pide al sistema introducir un nuevo tipo de acceso. 4. El sistema le proporciona un formulario mediante el cual obtiene todos los datos de éste. Acto seguido envía al usuario una petición de confirmación de alta de tipo de acceso. 5. El usuario responde afirmativamente y el sistema se encarga de registrar el nuevo tipo de acceso. (*) </td> </tr> <tr> <td>Alternativos (*)</td> <td> El usuario introduce algún dato con un formato erróneo: <ul style="list-style-type: none"> El sistema avisa del error acontecido y no realiza ninguna inserción. El usuario responde negativamente ante la petición de confirmación: <ul style="list-style-type: none"> El sistema no realiza ninguna acción más y da por concluido en caso de uso. </td> </tr> </table>	Básico	3. El usuario pide al sistema introducir un nuevo tipo de acceso. 4. El sistema le proporciona un formulario mediante el cual obtiene todos los datos de éste. Acto seguido envía al usuario una petición de confirmación de alta de tipo de acceso. 5. El usuario responde afirmativamente y el sistema se encarga de registrar el nuevo tipo de acceso. (*)	Alternativos (*)	El usuario introduce algún dato con un formato erróneo: <ul style="list-style-type: none"> El sistema avisa del error acontecido y no realiza ninguna inserción. El usuario responde negativamente ante la petición de confirmación: <ul style="list-style-type: none"> El sistema no realiza ninguna acción más y da por concluido en caso de uso.
	Básico	3. El usuario pide al sistema introducir un nuevo tipo de acceso. 4. El sistema le proporciona un formulario mediante el cual obtiene todos los datos de éste. Acto seguido envía al usuario una petición de confirmación de alta de tipo de acceso. 5. El usuario responde afirmativamente y el sistema se encarga de registrar el nuevo tipo de acceso. (*)			
Alternativos (*)	El usuario introduce algún dato con un formato erróneo: <ul style="list-style-type: none"> El sistema avisa del error acontecido y no realiza ninguna inserción. El usuario responde negativamente ante la petición de confirmación: <ul style="list-style-type: none"> El sistema no realiza ninguna acción más y da por concluido en caso de uso. 				
Post-condiciones	El administrador consigue que el nuevo tipo de acceso pase a formar parte del sistema.				

Caso de uso 57: Añadir tipo de acceso



Caso de uso 58: Borrar tipo de acceso		
Descripción	Borrar un tipo de acceso es tan fácil como puede ser borrar un nivel o una tarjeta.	
Actores principal	Administrador.	
Pre-condiciones	El usuario no desea seguir almacenando alguno de los tipos de acceso existentes en el sistema.	
Flujos	Básico	<ol style="list-style-type: none"> 4. El administrador selecciona los tipos de acceso que quiere eliminar e indica al sistema que proceda. 5. El sistema pide al usuario que confirme que los tipos de acceso seleccionados son realmente los que desea eliminar. 6. El administrador responde afirmativamente ante la advertencia. (*) 7. El sistema procede con el borrado de los tipos de acceso indicados.
	Alternativos (*)	<p>El administrador responde con un no:</p> <ul style="list-style-type: none"> • El sistema aborta cualquier tipo de operación y el caso de uso finaliza.
Post-condiciones	El usuario consigue lo que se proponía: eliminar los tipos de acceso no deseados.	

Caso de uso 58: Borrar tipo de acceso

Caso de uso 59: Modificar tipo de acceso		
Descripción	Es bastante probable que los tipos de acceso vayan variando a lo largo de la vida del centro para adaptarse a las nuevas necesidades que surjan.	
Actores principal	Administrador.	
Pre-condiciones	El administrador quiere cambiar alguna de las características de un tipo de acceso.	
Flujos	Básico	<ol style="list-style-type: none"> 3. El usuario selecciona el tipo de acceso a modificar y transmite al sistema su intención. 4. El sistema le muestra todos los datos del tipo de acceso. 5. El usuario edita lo que estima oportuno y solicita grabar los cambios. 6. El sistema indica al usuario que va a modificar el acceso y que necesita previamente una autorización. 7. El usuario la da y el sistema procede a actualizar el tipo de acceso. (*)
	Alternativos (*)	<p>El usuario no da su aprobación:</p> <ul style="list-style-type: none"> • El sistema no realiza ninguna acción y el caso de uso finaliza.
Post-condiciones	El tipo de acceso pasa a tener el estado deseado.	

Caso de uso 59: Modificar tipo de acceso



4.10. Cargos

Caso de uso 60: Visualizar cargos		
Descripción	Llegamos al último conjunto de elementos que trataremos en esta pestaña: los cargos.	
Actores principal	Administrador.	
Pre-condiciones	El usuario desea poder consultar todos los cargos existentes en el sistema.	
Flujos	Básico	<ol style="list-style-type: none">El usuario pide al sistema poder visualizar todos los cargos existentes.El sistema accede a la base de datos y obtiene los datos requeridos, mostrándoselos al usuario.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario visualiza todos los datos que buscaba.	

Caso de uso 60: Visualizar cargos

Caso de uso 61: Visualizar cargo		
Descripción	Es hora de permitir al usuario que visualice todos los datos del cargo que desee.	
Actores principal	Administrador.	
Pre-condiciones	El administrador quiere consultar todos los datos de un cargo. Inicia la aplicación y se sitúa en la pestaña de cargos.	
Flujos	Básico	<ol style="list-style-type: none">El administrador selecciona un cargo y pide al sistema que le muestre todos los datos que posea de éste.El sistema acude a la base de datos para rescatarlos y se los proporciona al administrador.
	Alternativos (*)	Ninguno.
Post-condiciones	El administrador visualiza todos los datos del cargo.	

Caso de uso 61: Visualizar cargo



Caso de uso 62: Añadir cargo	
Descripción	Si Informática Navarra S.A. decide guardar contactos que desempeñen un nuevo cargo, previamente deberá ser dado de alta éste en la base de datos.
Actores principal	Administrador.
Pre-condiciones	El administrador necesita añadir un nuevo cargo a los existentes. Arranca la aplicación, loguea y se dirige al tab de cargos.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario traslada al sistema su intención de dar de alta un nuevo cargo. 2. El sistema le proporciona un formulario que debe rellenar. 3. Tras completarlo, el usuario le pide al sistema que registre el nuevo cargo. 4. El sistema envía al usuario un mensaje que debe aceptar para proceder con el alta. 5. El usuario acepta el mensaje y el cargo pasa a formar parte de la base de datos. (*)
	<p>Alternativos (*)</p> <p>El usuario introduce datos que no poseen el formato adecuado:</p> <ul style="list-style-type: none"> • El sistema avisa del error y no realiza ninguna inserción. <p>El usuario no acepta el mensaje de confirmación:</p> <ul style="list-style-type: none"> • El sistema finaliza el caso de uso sin realizar ninguna acción.
Post-condiciones	El administrador dispone del nuevo cargo registrado en el sistema.

Caso de uso 62: Añadir cargo



Caso de uso 63: Borrar cargo	
Descripción	Puede llegar el momento en el que el administrador opte por prescindir de todos los contactos que desempeñen algún tipo de cargo.
Actores principal	Administrador.
Pre-condiciones	El usuario se encuentra en el tab de cargos dispuesto a eliminar uno o varios de ellos.
Flujos	Básico <ol style="list-style-type: none">1. El administrador selecciona los cargos que desea borrar e indica al sistema que proceda.2. Éste le avisa que va a eliminar los cargos seleccionados y le pregunta si de está seguro de querer realizar tal acción.3. El administrador le contesta afirmativamente y todos los cargos seleccionados dejan de existir. (*)
	Alternativos (*) <p>El administrador decide abortar el borrado respondiendo negativamente al aviso:</p> <ul style="list-style-type: none">• El sistema también aborta y el caso de uso no llega a completarse.
Post-condiciones	El administrador consigue eliminar los cargos que pretendía.

Caso de uso 63: Borrar cargo



Caso de uso 64: Modificar cargo	
Descripción	Este caso de uso resulta menos frecuente, pero no por ello podemos olvidarlo.
Actores principal	Administrador.
Pre-condiciones	El administrador desea modificar un cargo por constar éste con datos erróneos.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario se sitúa en la pestaña de cargos y selecciona el que desea modificar. Pulsa el botón de 'Modificar cargo' y el sistema le proporciona todos los datos actuales del cargo. 2. Una vez que el usuario ha modificado lo que estima oportuno, pide al sistema que actualice el cargo. 3. El sistema genera un aviso que el usuario debe aceptar como paso previo a la modificación. 4. El usuario da el visto bueno y la actualización tiene lugar. (*)
	<p>Alternativos (*)</p> <p>El usuario introduce algún nuevo dato con un formato indeseado:</p> <ul style="list-style-type: none"> • El sistema avisa del error y no realiza ninguna acción, volviendo al punto de partida. <p>El usuario responde negativamente ante la petición de confirmación:</p> <ul style="list-style-type: none"> • El sistema no envía ningún mensaje ni actualiza ningún registro, pero sí que da por concluido el caso de uso.
Post-condiciones	El administrador consigue modificar el cargo que se proponía con los datos proporcionados al sistema.

Caso de uso 64: Modificar cargo



4.11. Zonas del centro

Caso de uso 65: Visualizar zonas del centro		
Descripción	Comenzaremos tratando en primer lugar las zonas del centro, que son a las que están asociadas las llaves y llaveros. Al igual que ocurría con otros elementos, se nos presentan casos de uso que requieren poder observar datos globales y concretos. En este caso trataremos el poder observar los datos principales de todas las zonas del centro, mientras que en el siguiente lo haremos con todos los datos de una zona.	
Actores principal	Administrador.	
Pre-condiciones	El usuario quiere visualizar los principales datos de todas las zonas que componen la sede de Informática Navarra S.A.	
Flujos	Básico	7. El administrador loguea en el sistema y se dirige a la sección de zonas del centro. 8. El sistema recopila todos los datos necesarios y se los muestra al administrador.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario visualiza en pantalla todos los datos que buscaba.	

Caso de uso 65: Visualizar zonas del centro

Caso de uso 66: Visualizar zona del centro		
Descripción	Una vez que se pueden visualizar todas las zonas, es el momento de ofrecer una vía para conocer todos los datos de cualquiera de ellas.	
Actores principal	Administrador.	
Pre-condiciones	El administrador desea observar todos los datos que componen una de las zonas que se encuentra visualizando.	
Flujos	Básico	1. El usuario selecciona la zona a consultar y pide al sistema que le muestre todos sus datos. 2. El sistema recupera todos los datos de los que dispone y se los ofrece al usuario.
	Alternativos (*)	Ninguno.
Post-condiciones	El administrador observa en un formulario ofrecido por el sistema todos los datos que componen la zona seleccionada.	

Caso de uso 66: Visualizar zona del centro



Caso de uso 67: Añadir zona del centro		
Descripción	No es normal la redistribución de zonas, pero debemos tener en cuenta que el centro no tiene por qué permanecer inalterado durante toda su vida. Puede que nazcan nuevas zonas con intención de particionar en mayor grado la sede, puede que sea necesario modificar algunas de ellas,...	
Actores principal	Administrador.	
Pre-condiciones	El administrador se encuentra logueado en el sistema y se dispone a dar de alta una nueva zona.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El administrador traslada al sistema sus pretensiones. 2. El sistema le ofrece un formulario con el que poder recoger todos los datos de la nueva zona. 3. El administrador lo rellena y devuelve al sistema, quien le pide al usuario una confirmación de que todos los datos introducidos son correctos y que está seguro de querer realizar la inserción. 4. El administrador da su visto bueno. (*) 5. El sistema procede con la adición de la zona. (*)
	Alternativos (*)	<p>El administrador no da su visto bueno:</p> <ul style="list-style-type: none"> • El sistema no realiza ninguna operación y el caso de uso da por concluido. <p>El administrador introduce algún dato con formato erróneo:</p> <ul style="list-style-type: none"> • El sistema avisa al usuario de su error y vuelve al final del punto 2 sin haber modificado ningún campo.
Post-condiciones	El administrador consigue su objetivo y la nueva zona entra a formar parte del sistema.	

Caso de uso 67: Añadir zona del centro



Caso de uso 68: Borrar zona del centro	
Descripción	Una zona puede dejar de existir en el centro ya sea porque otra la embeba o porque no se desee controlar con la aplicación (cosa muy poco probable). Éste es un hecho que controlaremos con accessManager.
Actores principal	Administrador.
Pre-condiciones	El usuario necesita borrar una zona del sistema ya que ésta ha dejado de existir.
Flujos	Básico <ol style="list-style-type: none">1. El usuario loguea en el sistema, se dirige a la pestaña de zonas, selecciona aquellas que desea borrar y pulsa el botón de borrado.2. El sistema envía al usuario un mensaje de advertencia avisándole de la operación que va a realizar.3. El usuario aprueba el borrado y éste tiene lugar en el sistema. (*)
	Alternativos (*) <p>El usuario no aprueba el borrado:</p> <ul style="list-style-type: none">• El sistema no elimina ningún registro de la base de datos y el caso de uso concluye.
Post-condiciones	El sistema ya no dispone de las zonas que el usuario mandó borrar.

Caso de uso 68: Borrar zona del centro



Caso de uso 69: Modificar zona del centro	
Descripción	Ahora trataremos el registro de los cambios que pueden darse en los datos que componen una zona, como son su código, descripción o nivel necesario para acceder a ésta.
Actores principal	Administrador.
Pre-condiciones	El administrador loguea en el sistema con la pretensión de modificar los datos de una de las zonas que se encuentra observando.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El administrador selecciona la zona que desea modificar y el sistema le proporciona un formulario con todos los datos que la componen. 2. El administrador edita todos los datos que considera erróneos y pide al sistema que actualice la zona con los datos que le proporciona. 3. El sistema avisa al usuario de que va a modificar la zona seleccionada con los datos indicados y le pregunta si está seguro de querer llevar a cabo esta operación. 4. El administrador asiente. (*) 5. El sistema actualiza todos los registros necesarios en la base de datos. (*)
	<p>Alternativos (*)</p> <p>El usuario no ve con buenos ojos realizar la actualización:</p> <ul style="list-style-type: none"> • El sistema da por concluido el caso de uso sin realizar ningún tipo de operación. <p>La actualización no puede llevarse a cabo ya que el usuario ha introducido algún dato con formato erróneo:</p> <ul style="list-style-type: none"> • El sistema le avisa de su error y pasa al final del primer punto del flujo básico.
Post-condiciones	Los cambios indicados surten efecto y la zona seleccionada dispone en estos momentos de datos válidos.

Caso de uso 69: Modificar zona del centro



4.12. Llaves

Caso de uso 70: Visualizar llaves					
Descripción	Tras disponer de toda la funcionalidad necesaria para llevar a cabo una correcta gestión de las zonas, podemos pasar a tratar todo lo que ellas suponen: llaves, llaveros y almacenes. En primer lugar nos ocupamos de las llaves, ofreciendo una vía sencilla de poder observar los principales datos de todas las existentes.				
Actores principal	Administrador.				
Pre-condiciones	El usuario precisa conocer qué llaves están relacionadas con una zona.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> <ol style="list-style-type: none"> 1. El administrador selecciona la zona de la que desea conocer sus llaves. 2. El sistema accede a la base de datos y recupera los datos más representativos de éstas, mostrándolos por pantalla. </td> </tr> <tr> <td>Alternativos (*)</td> <td>Ninguno.</td> </tr> </table>	Básico	<ol style="list-style-type: none"> 1. El administrador selecciona la zona de la que desea conocer sus llaves. 2. El sistema accede a la base de datos y recupera los datos más representativos de éstas, mostrándolos por pantalla. 	Alternativos (*)	Ninguno.
	Básico	<ol style="list-style-type: none"> 1. El administrador selecciona la zona de la que desea conocer sus llaves. 2. El sistema accede a la base de datos y recupera los datos más representativos de éstas, mostrándolos por pantalla. 			
Alternativos (*)	Ninguno.				
Post-condiciones	El usuario puede visualizar los principales datos de todas las llaves pertenecientes a una zona.				

Caso de uso 70: Visualizar llaves

Caso de uso 71: Visualizar llave					
Descripción	La visualización de los principales datos de las llaves puede que no deje satisfecho al usuario, con que no estaría de más el poder visualizar todos los atributos al completo.				
Actores principal	Administrador.				
Pre-condiciones	El usuario se encuentra en la pestaña de zonas y desea que el sistema le proporcione todos los datos que componen una llave de una zona dada.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> <ol style="list-style-type: none"> 1. El administrador selecciona la llave a consultar. 2. El sistema se encarga de recuperar de la base de datos todos los datos que guarda de ella. A continuación genera un formulario con éstos y se lo muestra al administrador. </td> </tr> <tr> <td>Alternativos (*)</td> <td>Ninguno.</td> </tr> </table>	Básico	<ol style="list-style-type: none"> 1. El administrador selecciona la llave a consultar. 2. El sistema se encarga de recuperar de la base de datos todos los datos que guarda de ella. A continuación genera un formulario con éstos y se lo muestra al administrador. 	Alternativos (*)	Ninguno.
	Básico	<ol style="list-style-type: none"> 1. El administrador selecciona la llave a consultar. 2. El sistema se encarga de recuperar de la base de datos todos los datos que guarda de ella. A continuación genera un formulario con éstos y se lo muestra al administrador. 			
Alternativos (*)	Ninguno.				
Post-condiciones	El usuario visualiza en pantalla todos los datos de la llave seleccionada.				

Caso de uso 71: Visualizar llave



Caso de uso 72: Añadir llave	
Descripción	Está claro que el conjunto de llaves que permiten el acceso total a las zonas no permanecerá inmutable. Por ello los cuatro siguientes casos de uso.
Actores principal	Administrador.
Pre-condiciones	Una nueva llave entra en juego en la sede y es obligación del administrador registrarla en el sistema.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario indica al sistema que desea dar de alta una nueva llave. 2. El sistema le proporciona un formulario con todos los campos necesarios. 3. El usuario rellena el formulario y pide que se realice el registro. 4. El sistema indica que necesita de una aprobación previa para poder realizar el alta solicitada. 5. El usuario da su visto bueno y la adición de la llave al conjunto de llaves existentes en la zona es una realidad. (*)
	<p>Alternativos (*)</p> <p>El usuario ha introducido algún dato con un formato no válido:</p> <ul style="list-style-type: none"> • El sistema se percata del error, lo exterioriza y no realiza la inserción solicitada. <p>El usuario cambia de idea y decide abortar el alta de la llave:</p> <ul style="list-style-type: none"> • El sistema interrumpe el flujo básico y salta al final del caso de uso sin haber sido éste realizado.
Post-condiciones	El administrador ha conseguido dar de alta una nueva llave asociándola a una de las zonas existentes.

Caso de uso 72: Añadir llave



Caso de uso 73: Borrar llave					
Descripción	Si alguna de las cerraduras que pueden llegar a poblar una de las zonas desaparece (alguna puerta deja de necesitar llave, algún armario que deja de existir,...) también deberá hacerlo del sistema. No tiene sentido almacenar algo que no existe, es incoherente, innecesario.				
Actores principal	Administrador.				
Pre-condiciones	El administrador loguea en el sistema persiguiendo borrar alguna de las llaves existentes en una zona dada.				
Flujos	<table border="1"> <tr> <td style="text-align: center;">Básico</td> <td> <ol style="list-style-type: none"> 1. El usuario selecciona la zona en la que desea eliminar llaves. 2. El sistema le muestra las llaves existentes. 3. El usuario selecciona todas las llaves de las que desea prescindir y pide al sistema que proceda con la operación de borrado. 4. El sistema despliega un dialogo de aviso en el que informa al usuario del borrado y pide que lo apruebe si lo estima oportuno. 5. El usuario responde 'Sí' ante esta pregunta. (*) 6. El sistema elimina las llaves seleccionadas de la base de datos. </td> </tr> <tr> <td style="text-align: center;">Alternativos (*)</td> <td> <p>El usuario selecciona 'No':</p> <ul style="list-style-type: none"> • El sistema finaliza el caso de uso sorteando el sexto paso. </td> </tr> </table>	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona la zona en la que desea eliminar llaves. 2. El sistema le muestra las llaves existentes. 3. El usuario selecciona todas las llaves de las que desea prescindir y pide al sistema que proceda con la operación de borrado. 4. El sistema despliega un dialogo de aviso en el que informa al usuario del borrado y pide que lo apruebe si lo estima oportuno. 5. El usuario responde 'Sí' ante esta pregunta. (*) 6. El sistema elimina las llaves seleccionadas de la base de datos. 	Alternativos (*)	<p>El usuario selecciona 'No':</p> <ul style="list-style-type: none"> • El sistema finaliza el caso de uso sorteando el sexto paso.
	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona la zona en la que desea eliminar llaves. 2. El sistema le muestra las llaves existentes. 3. El usuario selecciona todas las llaves de las que desea prescindir y pide al sistema que proceda con la operación de borrado. 4. El sistema despliega un dialogo de aviso en el que informa al usuario del borrado y pide que lo apruebe si lo estima oportuno. 5. El usuario responde 'Sí' ante esta pregunta. (*) 6. El sistema elimina las llaves seleccionadas de la base de datos. 			
Alternativos (*)	<p>El usuario selecciona 'No':</p> <ul style="list-style-type: none"> • El sistema finaliza el caso de uso sorteando el sexto paso. 				
Post-condiciones	El sistema ya no dispone de las llaves eliminadas por el administrador.				

Caso de uso 73: Borrar llave



Caso de uso 74: Dar de baja llave	
Descripción	El borrado anteriormente citado implicaría la pérdida de la trazabilidad entre accesos y llaves empleadas. Para evitarlo se diseñará un sistema por el cual la llave deje de formar parte de la aplicación (muestra en pestaña) pero no del sistema.
Actores principal	Administrador.
Pre-condiciones	El administrador quiere dar de baja una llave ya que esta no será necesaria de aquí en adelante.
Flujos	Básico <ol style="list-style-type: none">1. El administrador se sitúa en la pestaña de zonas, selecciona aquella donde desea dar de baja llaves y selecciona las que estime oportuno. Acto seguido, traslada al sistema su intención de darlas de baja.2. El sistema procede con la baja de éstas siempre y cuando el administrador acepte el mensaje de aviso que le es mostrado por pantalla. (*)
	Alternativos (*) <p>El administrador finalmente decide no dar de baja las llaves que ha seleccionado, respondiendo negativamente ante el diálogo en pantalla:</p> <ul style="list-style-type: none">• El sistema no ejecuta ninguna acción y da por finalizado el caso de uso.
Post-condiciones	El administrador consigue dar de baja las llaves que seleccionó.

Caso de uso 74: Dar de baja llave



Caso de uso 75: Modificar llave		
Descripción	Las llaves están compuestas de código, descripción y nota, que son atributos que no tienen por qué permanecer inalterados. Puede ser necesario cambiar el código de una llave, editar la descripción de ésta por ser errónea,...	
Actores principal	Administrador.	
Pre-condiciones	El usuario se encuentra en el sistema, más concretamente en la tab de zonas visualizando el conjunto de llaves que forman parte de una zona, y desea modificar los datos que conforman una llave al no ser éstos válidos.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona la llave que desea modificar. 2. El sistema le ofrece todos los datos que la componen en un formulario. 3. El usuario edita los que estima conveniente y pide al sistema que proceda con la actualización. 4. El sistema insta al usuario a que dé su visto bueno para que la actualización tenga lugar. 5. El usuario obedece, la da y el caso de uso finaliza con la consecución del fin para el que fue ideado. (*)
	Alternativos (*)	<p>El usuario ha proporcionado datos que no puede aceptar el sistema por ser erróneos en su formato:</p> <ul style="list-style-type: none"> • El sistema avisa al usuario del error ocurrido y retorna al final del punto 2 sin haber modificado dato alguno. <p>El usuario no da su visto bueno a la actualización:</p> <ul style="list-style-type: none"> • El sistema no actualiza ningún registro de la base de datos y da por concluido el caso de uso.
Post-condiciones	El usuario consigue que la llave conste con datos válidos y no erróneos, como los que tenía antes de iniciarse este caso de uso.	

Caso de uso 75: Modificar llave



Caso de uso 76: Listar llaves		
Descripción	El cliente pide expresamente al equipo de desarrollo que le ofrezca una vía para poder obtener reportes de las llaves que existen en el sistema.	
Actores principal	Administrador.	
Pre-condiciones	El usuario se encuentra observando las llaves que forman parte del sistema y se dispone a obtener un reporte de éstas.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario pide al sistema que genere un informe con las llaves que se encuentra visualizando en pantalla. 2. El sistema pide al usuario la ubicación donde deberá depositar el reporte a generar. 3. El usuario proporciona la ruta. (*) 4. El sistema elabora el informe y lo deposita en la ubicación indicada.
	Alternativos (*)	<p>El usuario no proporciona ninguna ubicación, cancelando el diálogo de selección e ruta:</p> <ul style="list-style-type: none"> • El caso de uso finaliza sin la generación de ningún informe.
Post-condiciones	El usuario dispone del reporte solicitado en la ruta indicada.	

Caso de uso 76: Listar llaves

4.13. Llaveros

Caso de uso 77: Visualizar llaveros		
Descripción	Completadas las llaves debemos hacer lo mismo con los llaveros. Los casos de uso serán similares, solo que algo más numerosos y complicados ya que nos encontramos ante objetos compuestos.	
Actores principal	Administrador.	
Pre-condiciones	El administrador necesita conocer el conjunto de llaveros existentes. Para ello, loguea en el sistema y se dirige a la tab de zonas.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El administrador pide al sistema que le muestre el conjunto de llaveros existentes. 2. El sistema acude a la base de datos y recupera de ella todos los datos necesarios para mostrárselos al administrador por pantalla y satisfacer su necesidad.
	Alternativos (*)	Ninguno.
Post-condiciones	El administrador visualiza los llaveros existentes, habiendo conseguido realizar el caso de uso con éxito.	

Caso de uso 77: Visualizar llaveros



Caso de uso 78: Visualizar llavero		
Descripción	Observar el conjunto de llaveros existentes puede no ser del todo suficiente, ya que existen datos que se escapan al anterior caso de uso, como pueden ser el conjunto de llaves que los conforman.	
Actores principal	Administrador.	
Pre-condiciones	El usuario requiere al sistema que le muestre todos los datos disponibles sobre un llavero que se encuentra visualizando.	
Flujos	Básico	<ol style="list-style-type: none">1. El usuario selecciona el llavero que desea visualizar.2. El sistema, gracias a la base de datos, muestra por pantalla todos los datos que el usuario estaba buscando.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario observa por pantalla todos los datos que describen el llavero seleccionado.	

Caso de uso 78: Visualizar llavero



Caso de uso 79: Añadir llavero	
Descripción	Si el conjunto de llaves puede variar, el conjunto de llaveros no es menos.
Actores principal	Administrador.
Pre-condiciones	El administrador se encuentra logueado en el sistema y precisa dar de alta un nuevo llavero.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El administrador selecciona las llaves que conformarán el nuevo llavero y pide al sistema crear uno nuevo con éstas. 2. El sistema le proporciona un formulario para que el administrador termine de definir el nuevo llavero. 3. El administrador lo rellena y pide al sistema que realice el alta. 4. El sistema requiere al administrador una autorización previa para realizar la operación que éste ha solicitado. 5. El administrador la otorga. (*) 6. El sistema realiza el registro del nuevo llavero en el sistema. (*)
	<p>Alternativos (*)</p> <p>El administrador decide abortar la inserción del nuevo llavero:</p> <ul style="list-style-type: none"> • El sistema obedece, no realiza ninguna acción y la ejecución del caso de uso finaliza. <p>El sistema se percata de que el usuario ha introducido algún dato con un formato no deseado:</p> <ul style="list-style-type: none"> • Le avisa del error que acaba de cometer y pasa al final del punto 2 sin modificar ningún dato del formulario.
Post-condiciones	El sistema alberga un nuevo llavero con las características que el administrador ha proporcionado.

Caso de uso 79: Añadir llavero



Caso de uso 80: Borrar llavero	
Descripción	Borrar un llavero es otra de las funcionalidades que debemos proporcionar, ya que éstos no perdurarán eternamente.
Actores principal	Administrador.
Pre-condiciones	El usuario accede al sistema con la intención de realizar el borrado de alguno de los llaveros que no quiere que permanezcan en el sistema.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El administrador accede a la sección de llaveros y selecciona aquellos que desea eliminar, iniciando el proceso de borrado con la pulsación del botón creado para tal fin. 2. El sistema despliega un dialogo de aviso en el que advierte al administrador de que va a proceder con el borrado de los llaveros seleccionados, necesitando previamente una autorización para asegurarse que el borrado es el que el administrador realmente quiere realizar. 3. El administrador da su visto bueno. (*) 4. El sistema elimina del sistema todos los llaveros seleccionados.
	<p>Alternativos (*)</p> <p>El administrador cambia de idea y no ve necesario realizar el borrado:</p> <ul style="list-style-type: none"> • El sistema aborta el caso de uso y se llega al final de éste.
Post-condiciones	El usuario consigue su objetivo: eliminar los llaveros seleccionados.

Caso de uso 80: Borrar llavero

Caso de uso 81: Dar de baja llavero	
Descripción	Al igual que ocurría con las llaves, la eliminación de llaveros implica la pérdida de la trazabilidad entre los accesos y éstos, con que si se quiere seguir manteniéndola debe proporcionarse una forma de dar de baja llaveros.
Actores principal	Administrador.
Pre-condiciones	El administrador loguea en el sistema con el objetivo de dar de baja alguno de los llaveros que residen en éste.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario accede a la pestaña donde se muestran los llaveros y selecciona aquellos que desea dar de baja, pidiendo al sistema que realice dicha acción. 2. El sistema obedece siempre y cuando reciba una respuesta afirmativa al mensaje de aviso que enviará al usuario explicándole la operación que se dispone a realizar. (*)
	<p>Alternativos (*)</p> <p>El usuario responde negativamente al aviso:</p> <ul style="list-style-type: none"> • El sistema da por concluido el caso de uso no llevando a cabo ninguna acción.
Post-condiciones	El administrador consigue dar de baja los llaveros que pretendía.

Caso de uso 81: Dar de baja llavero



Caso de uso 82: Modificar llavero	
Descripción	Con los tres siguientes casos de uso haremos capaz al usuario de poder modificar los datos o composición de cualquiera de los llaveros que se proponga.
Actores principal	Administrador.
Pre-condiciones	El usuario inicia este caso de uso persiguiendo modificar alguno de los datos que componen uno de los llaveros que se encuentra visualizando.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El administrador selecciona el llavero a modificar y transmite al sistema su deseo. 2. El sistema despliega un formulario con todos los datos del llavero, de modo que el administrador pueda editar lo que estime oportuno. 3. Tras la edición, el administrador pide al sistema que realice la actualización de campos. 4. El sistema envía al administrador un aviso pidiéndole que confirme que desea modificar el llavero seleccionado. 5. El administrador responde positivamente y la actualización tiene lugar en la base de datos. (*)
	<p>Alternativos (*)</p> <p>El administrador introduce datos con un formato erróneo:</p> <ul style="list-style-type: none"> • El sistema avisa al usuario del error acontecido y retorna al final del punto 2. <p>El administrador aborta la actualización contestando negativamente a la advertencia:</p> <ul style="list-style-type: none"> • El sistema no modifica ningún registro en la base de datos y da por concluido el caso de uso.
Post-condiciones	El administrador consigue que el llavero que constaba con datos erróneos ahora lo haga con válidos.

Caso de uso 82: Modificar llavero



Caso de uso 83: Añadir llave	
Descripción	Informática Navarra S.A. puede estimar oportuno modificar algún llavero añadiéndole más llaves.
Actores principal	Administrador.
Pre-condiciones	El usuario quiere modificar uno de los llaveros que se encuentra visualizando mediante la adición de alguna nueva llave existente.
Flujos	Básico <ol style="list-style-type: none"> 1. El usuario selecciona las llaves y llavero en el que insertarlas, instando al sistema a que realice los cambios necesarios para llevar a cabo esta actualización. 2. El sistema cumple con las intenciones del usuario siempre y cuando éste conteste 'Sí' a su dialogo de aviso. (*)
	Alternativos (*) <p>El usuario contesta 'No' ya que finalmente no desea añadir las llaves al llavero:</p> <ul style="list-style-type: none"> • El caso de uso finaliza sin ejecutarse ninguna sentencia sobre la base de datos.
Post-condiciones	El administrador consigue añadir las llaves al llavero seleccionado, pasando a constar éste con un conjunto de llaves distinto al que poseía antes de comenzar este caso de uso.

Caso de uso 83: Añadir llave

Caso de uso 84: Borrar llave	
Descripción	También puede suceder que alguna llave deje de constar en algún llavero, siendo necesario implementar el caso de uso opuesto al anterior.
Actores principal	Administrador.
Pre-condiciones	El administrador se encuentra registrado en el sistema, visualizando un llavero y desea borrar algunas de las llaves que aúna.
Flujos	Básico <ol style="list-style-type: none"> 1. El usuario selecciona las llaves que desea que dejen de formar parte del llavero y pide al sistema que las elimine. 2. El sistema pide al usuario que confirme sus pretensiones. 3. El usuario responde afirmativamente. (*) 4. El sistema se encarga de realizar el borrado.
	Alternativos (*) <p>El usuario contesta negativamente:</p> <ul style="list-style-type: none"> • Se salta al final del caso de uso quedando el llavero inalterado.
Post-condiciones	Las llaves seleccionadas por el usuario dejan de formar parte del llavero.

Caso de uso 84: Borrar llave



Caso de uso 85: Listar llaveros		
Descripción	Otro de los requisitos que el cliente nos impone es el que pueda listar todos los llaveros registrados en el sistema.	
Actores principal	Administrador.	
Pre-condiciones	El usuario se encuentra en la sección de llaveros y se dispone a obtener un reporte de éstos.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El administrador pide al sistema que elabore un informe con todos los llaveros de los que dispone. 2. El sistema responde pidiéndole que le ofrezca una ruta en la que depositar dicho informe. 3. El usuario se la proporciona. (*) 4. El sistema genera el reporte solicitado y lo deposita en la ubicación indicada.
	Alternativos (*)	El usuario no se la proporciona: <ul style="list-style-type: none"> • El sistema no elabora ningún tipo de informe y da por finalizada la ejecución del caso de uso.
Post-condiciones	El usuario puede disfrutar del informe solicitado en la ubicación señalada.	

Caso de uso 85: Listar llaveros

4.14. Almacenes de llaves

Caso de uso 86: Visualizar almacenes de llaves		
Descripción	El usuario debe poder visualizar toda la información referente a los distintos almacenes existentes y qué elementos almacenan. En un principio, se define el poder observar los almacenes que existen y a continuación poder visualizar toda la información de cada almacén.	
Actores principal	Administrador.	
Pre-condiciones	El administrador necesita conocer qué almacenes se encuentran registrados en el sistema.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario loguea y accede a la sección de almacenes. 2. El sistema se encarga de recuperar de la base de datos los principales datos de los almacenes existentes y mostrárselos.
	Alternativos (*)	Ninguno.
Post-condiciones	El administrador visualiza en pantalla todos los almacenes de datos existentes.	

Caso de uso 86: Visualizar almacenes de llaves

Caso de uso 87: Visualizar almacén de llaves		
Descripción	Una vez que el cliente puede obtener todos los almacenes existentes, debería poder seleccionar uno y visualizar toda su composición.	
Actores principal	Administrador.	
Pre-condiciones	El administrador precisa conocer el contenido de alguno de los almacenes que visualiza en pantalla.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El administrador selecciona el almacén del que desea obtener toda su información. 2. El sistema accede a la base de datos, rescata toda la información que dispone de éste y la muestra en pantalla.
	Alternativos (*)	Ninguno.
Post-condiciones	El administrador obtiene la información que buscaba y el caso de uso finaliza con éxito.	

Caso de uso 87: Visualizar almacén de llaves

Caso de uso 88: Añadir almacén de llaves		
Descripción	No es lo normal, pero nunca se puede desechar el hecho de que pueda surgir algún nuevo almacén en el que Informática Navarra S.A. desee almacenar llaves y llaveros.	
Actores principal	Administrador.	
Pre-condiciones	El usuario loguea en el sistema con la intención de realizar el alta de un nuevo almacén de llaves.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario acude a la sección de almacenes e indica al sistema que desea realizar el alta de uno nuevo. 2. El sistema le proporciona un formulario que deberá rellenar. 3. El usuario obedece y tras completarlo le pide al sistema que proceda con el alta. 4. Éste inserta en la base de datos un registro con el nuevo almacén previa autorización por parte del usuario. (*)
	Alternativos (*)	<p>El usuario proporciona algún dato que no cumple con el formato esperado:</p> <ul style="list-style-type: none"> • El sistema avisa al usuario de su error y vuelve al final del punto 2 sin modificar ningún campo. <p>El usuario cambia de idea y aborta dar de alta el nuevo almacén:</p> <ul style="list-style-type: none"> • El caso de uso salta del inicio al final del punto 4, pasando por alto la adición al sistema del nuevo almacén.
Post-condiciones	El usuario consigue que el nuevo almacén sea registrado en la base de datos.	

Caso de uso 88: Añadir almacén de llaves



Caso de uso 89: Borrar almacén de llaves		
Descripción	Reducir el número de almacenes en los que guardar llaves y llaveros debería ser posible en la aplicación informática.	
Actores principal	Administrador.	
Pre-condiciones	El usuario loguea en el sistema con el objetivo de borrar alguno de los almacenes existentes.	
Flujos	Básico	<ol style="list-style-type: none">1. El administrador selecciona los almacenes que desea eliminar y comunica al sistema su intención.2. Éste consulta al administrador para tener la certeza de que está seguro de querer realizar la acción propuesta.3. El administrador contesta afirmativamente, permitiendo que el sistema ejecute la tarea. (*)
	Alternativos (*)	El administrador responde negativamente: <ul style="list-style-type: none">• El sistema no realiza ningún tipo de borrado y da por concluido el flujo básico del caso de uso.
Post-condiciones	El sistema ya no dispone de los almacenes de datos eliminados por el usuario.	

Caso de uso 89: Borrar almacén de llaves



Caso de uso 90: Modificar almacén de llaves	
Descripción	A continuación tratamos todos los casos de uso que pueden ocasionar la modificación del estado de cualquiera de los almacenes existentes.
Actores principal	Administrador.
Pre-condiciones	El usuario se percata de que uno de los almacenes consta con datos erróneos y se dirige a la sección dedicada a éstos persiguiendo actualizarlo de modo que todos los datos de éste sean correctos.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El administrador selecciona el almacén cuyos datos desea modificar y transmite al sistema su querer. 2. Éste le muestra un formulario con los datos del almacén para que pueda editarlos a su antojo. 3. Una vez el administrador dispone de todo listo, pide al sistema que proceda con la actualización. 4. El sistema envía un aviso al administrador indicándole la operación que se dispone a realizar y pidiendo permiso para ello. 5. El administrador da el visto bueno. (*) 6. La actualización tiene lugar. (*)
	<p>Alternativos (*)</p> <p>El administrador no da el visto bueno:</p> <ul style="list-style-type: none"> • El caso de uso finaliza sin completarse. <p>El administrador introduce algún dato con formato erróneo:</p> <ul style="list-style-type: none"> • El sistema le avisa del error que acaba de cometer y retorna al final del punto 2.
Post-condiciones	El usuario hace que el almacén cuyos datos eran erróneos en un principio ahora sean correctos.

Caso de uso 90: Modificar almacén de llaves



Caso de uso 91: Añadir llave	
Descripción	La distribución de llaves en los distintos almacenes variará a lo largo del tiempo. Esto hace que debamos implementar funciones de adición, sustracción y modificación del número de copias de cualquier llave.
Actores principal	Administrador.
Pre-condiciones	El usuario se encuentra en la sección dedicada a los almacenes y quiere añadir a uno de ellos una llave.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario selecciona la llave y almacén que desea relacionar y clica en el botón de 'Añadir llave'. 2. El sistema le proporciona un campo para que indique el número de copias de la llave que residirán en el almacén. 3. El usuario las indica y pide al sistema que proceda. 4. Éste lo hace pero antes debe recibir el visto bueno por parte del usuario. (*)
	<p>Alternativos (*)</p> <p>El usuario indica un número de copias no comprendido entre 1 y 99:</p> <ul style="list-style-type: none"> • El sistema finaliza el caso de uso con un mensaje de error. <p>El usuario acaba viendo innecesario introducir la llave en el almacén, con lo que contesta 'No' al mensaje de advertencia:</p> <ul style="list-style-type: none"> • El sistema concluye sin realizar ninguna inserción y sin mostrar ningún tipo de aviso.
Post-condiciones	El administrador dispone de copias de la llave en el almacén indicado.

Caso de uso 91: Añadir llave



Caso de uso 92: Borrar llave	
Descripción	La extracción de una llave de un almacén debe ser reflejado también en el sistema.
Actores principal	Administrador.
Pre-condiciones	El administrador se encuentra logueado en la aplicación y desea borrar alguna de las llaves que pueblan un almacén concreto.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario selecciona las llaves que desea eliminar del almacén que se encuentra visualizando e indica al sistema que efectúe las operaciones necesarias para ello. 2. El sistema despliega un diálogo en el que indica al usuario las llaves que se dispone a borrar e instándole a que confirme la operación. 3. El usuario la confirma y el borrado surte efecto. (*)
	<p>Alternativos (*)</p> <p>El usuario contesta con un 'No':</p> <ul style="list-style-type: none"> • El sistema no elimina ningún registro de la base de datos.
Post-condiciones	El usuario ha conseguido que las llaves seleccionadas no permanezcan en el almacén.

Caso de uso 92: Borrar llave

Caso de uso 93: Modificar copias llave	
Descripción	Puede no variar la existencia de la llave pero sí el número de copias de las que se dispone. De ahí nace este caso de uso.
Actores principal	Administrador.
Pre-condiciones	El usuario se dispone a modificar el número de copias existentes de una llave en un almacén.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El administrador selecciona la llave dentro del almacén y pide al sistema poder modificar su número de copias. 2. El sistema le muestra en un campo el número de copias registradas y el administrador lo edita con el número de copias deseadas. 3. La actualización tiene lugar siempre y cuando el administrador dé su visto bueno (*).
	<p>Alternativos (*)</p> <p>El administrador introduce un número de copias no permitido:</p> <ul style="list-style-type: none"> • El sistema indica el error acontecido y no actualiza el número de copias. <p>El administrador aborta la actualización:</p> <ul style="list-style-type: none"> • El caso de uso llega a su fin sin haber efectuado cambio alguno en el sistema.
Post-condiciones	El usuario consigue actualizar el número de copias existentes de una llave en un almacén.

Caso de uso 93: Modificar copias llave



Caso de uso 94: Añadir llavero		
Descripción	Los llaveros poseen las mismas necesidades que las llaves, por lo que debemos crear casos de uso análogos para ellos.	
Actores principal	Administrador.	
Pre-condiciones	El usuario se registra en el sistema y accede a la sección de almacenes ante la necesidad de añadir un llavero a uno de ellos.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El administrador selecciona el llavero y almacén que desea relacionar y clicla el botón 'Añadir llavero'. 2. El sistema le proporciona un campo en el que indicar el número de copias del llavero a introducir en el almacén. 3. El administrador rellena el campo y pide al sistema que realice la inserción. 4. Éste muestra por pantalla un diálogo en el que avisa al administrador de la acción que realizará si éste le da permiso para ello. 5. El administrador lo otorga. (*) 6. El sistema inserta el llavero en el almacén indicado. (*)
	Alternativos (*)	<p>El usuario no da su visto bueno a la inserción:</p> <ul style="list-style-type: none"> • El sistema no realiza ninguna acción y el caso de uso termina. <p>A la hora de realizar la inserción, el sistema se percata de que el número de copias indicado es incorrecto:</p> <ul style="list-style-type: none"> • Avisa al administrador del error que ha cometido y permanece inalterado a la espera de que el usuario vuelva a intentar la inserción o pase a ejecutar otro caso de uso.
Post-condiciones	La base de datos se ve alterada mediante la introducción de un llavero en uno de sus almacenes.	

Caso de uso 94: Añadir llavero



Caso de uso 95: Borrar llavero	
Descripción	El borrado de llaveros será similar al realizado con las llaves.
Actores principal	Administrador.
Pre-condiciones	El administrador accede a la aplicación con ánimo de borrar alguno de los llaveros que constan en un almacén.
Flujos	Básico <ol style="list-style-type: none">1. El administrador despliega el contenido del almacén que desea modificar y selecciona de él aquellos llaveros que desee desechar. Acto seguido, pulsa el botón 'Borrar llavero'.2. El sistema responde avisándole de que va a eliminar los llaveros que indica, pero que antes necesita asegurarse de que la selección es correcta.3. El administrador le indica que lo es, hecho que desencadena que el caso de uso finalice correctamente. (*)
	Alternativos (*)
Post-condiciones	El almacén ya no dispone de los llaveros borrados por el usuario.

Caso de uso 95: Borrar llavero



Caso de uso 96: Modificar copias llavero		
Descripción	Cuando el número de copias de un llavero varíe el usuario deberá ejecutar este caso de uso.	
Actores principal	Administrador.	
Pre-condiciones	El usuario precisa modificar el número de copias existentes de un llavero en un almacén. Para ello loguea en el sistema y se dirige a la sección de llaveros.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona el almacén y llavero que desea modificar. 2. Tras recibir la orden de modificación, el sistema hace visible un campo con el número de copias disponibles del llavero, número que el usuario modifica y pide al sistema que almacene. 3. El sistema realiza la actualización si el usuario da su visto bueno. (*)
	Alternativos (*)	<p>El usuario introduce un número de copias no permitido:</p> <ul style="list-style-type: none"> • El sistema le avisa del error que ha cometido y permanece inalterado a la espera de acontecimientos. <p>El sistema aborta la modificación ya que no la estima oportuna:</p> <ul style="list-style-type: none"> • El sistema no realiza ninguna acción y la ejecución del caso de uso concluye.
Post-condiciones	El usuario ha conseguido modificar el número de copias de un llavero existentes en uno de los almacenes.	

Caso de uso 96: Modificar copias llavero



Caso de uso 97: Listar almacenes de llaves		
Descripción	El cliente desea poder obtener siempre que lo necesite un informe con todos los almacenes existentes y los elementos que éstos almacenan.	
Actores principal	Administrador.	
Pre-condiciones	El administrador precisa conocer el estado de los almacenes a través de un reporte.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El administrador pide al sistema que elabore un informe con los almacenes existentes y las llaves y llaveros que guardan. 2. El sistema le pide que indique en qué ubicación desea recibir el reporte. 3. El usuario selecciona el <i>path</i> a través de un cuadro de diálogo. (*) 4. El sistema se encarga de generar el reporte y depositarlo en la ruta indicada.
	Alternativos (*)	<p>El usuario no selecciona ninguna ubicación, cancelando el cuadro de diálogo:</p> <ul style="list-style-type: none"> • El caso de uso finaliza.
Post-condiciones	El administrador puede disfrutar del informe requerido en la ubicación indicada.	

Caso de uso 97: Listar almacenes de llaves

4.15. Préstamos pendientes

Caso de uso 98: Visualizar préstamos		
Descripción	Todos los accesos que se lleven a cabo en el centro conllevarán el préstamo de tarjetas y/o llaves y llaveros. Es mucho el material que cambia de manos, con lo que parece muy razonable crear un apartado dentro de la aplicación destinado a llevar su control.	
Actores principal	Administrador.	
Pre-condiciones	El usuario desea conocer qué préstamos están pendientes de devolución.	
Flujos	Básico	<ol style="list-style-type: none"> 9. El administrador accede a la pestaña de préstamos. 10. El sistema se encarga de realiza la búsqueda y mostrarle qué material está pendiente de devolución.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario visualiza todos los préstamos que en estos momentos están pendientes de ser devueltos.	

Caso de uso 98: Visualizar préstamos



Caso de uso 99: Visualizar datos del prestatario					
Descripción	Una vez obtenidos los datos de todos los préstamos que están pendientes de devolución, puede ser necesario conocer los principales datos del prestatario de cara a poder pedirle que devuelva todo el material lo antes posible.				
Actores principal	Administrador.				
Pre-condiciones	El administrador se encuentra logueado en el sistema y por las razones que sean necesita conocer los datos de la persona que tenga en préstamo algún tipo de material (tarjeta y/o llaves y/o llaveros).				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> 5. El usuario pulsa el préstamo de cuyo prestatario necesita conocer sus datos personales. 6. El sistema accede a la base de datos y le proporciona al usuario los datos que busca, permitiendo el envío de un recordatorio vía E-Mail al prestatario en caso de que se disponga de alguna dirección de correo electrónico suya. </td> </tr> <tr> <td>Alternativos (*)</td> <td>Ninguno.</td> </tr> </table>	Básico	5. El usuario pulsa el préstamo de cuyo prestatario necesita conocer sus datos personales. 6. El sistema accede a la base de datos y le proporciona al usuario los datos que busca, permitiendo el envío de un recordatorio vía E-Mail al prestatario en caso de que se disponga de alguna dirección de correo electrónico suya.	Alternativos (*)	Ninguno.
	Básico	5. El usuario pulsa el préstamo de cuyo prestatario necesita conocer sus datos personales. 6. El sistema accede a la base de datos y le proporciona al usuario los datos que busca, permitiendo el envío de un recordatorio vía E-Mail al prestatario en caso de que se disponga de alguna dirección de correo electrónico suya.			
Alternativos (*)	Ninguno.				
Post-condiciones	El administrador consigue lo que buscaba: visualizar la información del prestatario y disponer de la posibilidad de enviarle un correo electrónico si es que se dispone de alguna dirección.				

Caso de uso 99: Visualizar datos del prestatario

Caso de uso 100: Devolver préstamo					
Descripción	La devolución de préstamos deberá irse realizando conforme los trabajadores entreguen de vuelta el material que en su día tomaron prestado. Como el vigilante solo podrá editar los accesos que él vaya dando y se encuentren sin salida y el administrador no tendrá privilegios de modificación en la pestaña de accesos, es necesario que se pueda registrar la devolución de préstamos desde esta pestaña.				
Actores principal	Administrador.				
Pre-condiciones	El usuario recibe de un trabajador algún préstamo pendiente y se dispone a registrar la devolución.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> 11. El usuario accede a la pestaña de préstamos y selecciona el que acaba de ser entregado, indicando dicho evento al sistema. 12. Éste le pide aprobación para marcar como devuelto todo aquel material que esté seleccionado en las tablas de préstamos. 13. El usuario le indica que todo es correcto y el sistema puede proceder. (*) </td> </tr> <tr> <td>Alternativos (*)</td> <td> El usuario le expresa al sistema que no está satisfecho con las devoluciones que se van a realizar: <ul style="list-style-type: none"> El sistema no realiza ninguna acción y da por concluido el caso de uso. </td> </tr> </table>	Básico	11. El usuario accede a la pestaña de préstamos y selecciona el que acaba de ser entregado, indicando dicho evento al sistema. 12. Éste le pide aprobación para marcar como devuelto todo aquel material que esté seleccionado en las tablas de préstamos. 13. El usuario le indica que todo es correcto y el sistema puede proceder. (*)	Alternativos (*)	El usuario le expresa al sistema que no está satisfecho con las devoluciones que se van a realizar: <ul style="list-style-type: none"> El sistema no realiza ninguna acción y da por concluido el caso de uso.
	Básico	11. El usuario accede a la pestaña de préstamos y selecciona el que acaba de ser entregado, indicando dicho evento al sistema. 12. Éste le pide aprobación para marcar como devuelto todo aquel material que esté seleccionado en las tablas de préstamos. 13. El usuario le indica que todo es correcto y el sistema puede proceder. (*)			
Alternativos (*)	El usuario le expresa al sistema que no está satisfecho con las devoluciones que se van a realizar: <ul style="list-style-type: none"> El sistema no realiza ninguna acción y da por concluido el caso de uso. 				
Post-condiciones	La devolución de material efectuada consta en el sistema.				

Caso de uso 100: Devolver préstamo



Caso de uso 101: Enviar recordatorio vía E-Mail		
Descripción	En el caso de uso 99 comentamos que deberá existir una manera de poder enviar un correo electrónico al prestatario en caso de que se disponga de alguna dirección electrónica suya almacenada.	
Actores principal	Administrador.	
Pre-condiciones	Se ha completado el caso de uso 99, el sistema dispone de alguna dirección de correo electrónico almacenada del prestatario y el usuario estima oportuno enviarle un mensaje recordándole los préstamos que tiene pendientes.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El administrador clic en el botón de 'Enviar recordatorio'. 2. El sistema, tras recibir la aprobación por parte del administrador, elabora un E-Mail con los datos del prestatario y los préstamos que tiene pendientes y lo envía a todas las direcciones de correo electrónico almacenadas de éste.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario ha enviado un correo electrónico al prestatario que deseaba indicándole qué préstamos tiene pendientes y pidiéndole que los devuelva lo antes posible.	

Caso de uso 101: Enviar recordatorio vía E-Mail

Caso de uso 102: Listar préstamos pendientes		
Descripción	El cliente nos pide que le ofrezcamos una forma de poder listar parte o la totalidad de los préstamos que están pendientes de devolución, de modo que pueda consultarlos sin necesidad de accessManager.	
Actores principal	Administrador.	
Pre-condiciones	El administrador quiere obtener un listado con parte de los préstamos que están pendientes de devolución.	
Flujos	Básico	<ol style="list-style-type: none"> 7. El usuario accede al sistema y se sitúa en la pestaña 'Préstamos'. Selecciona de qué préstamos desea que esté formado el listado y pide al sistema que lo genere. 8. Éste abre un diálogo en el que el usuario especificará dónde almacenar el listado. (*) 9. Tras esto, el sistema genera el listado con los datos requeridos y lo deposita en la ruta introducida por el usuario.
	Alternativos (*)	El usuario cancela el diálogo de selección de ruta: <ul style="list-style-type: none"> • El caso de uso finaliza.
Post-condiciones	El administrador dispone del listado solicitado.	

Caso de uso 102: Listar préstamos pendientes.



4.16. Cuentas del sistema

Caso de uso 103: Visualizar cuentas		
Descripción	El administrador debe tener una vía para saber qué cuentas existen en el sistema y qué usuarios las poseen.	
Actores principal	Administrador.	
Pre-condiciones	El administrador se encuentra logueado en el sistema y desea consultar qué cuentas están registradas.	
Flujos	Básico	<ol style="list-style-type: none">1. El usuario accede a la pestaña 'Cuentas'.2. El sistema rescata toda la información necesaria y la muestra al usuario.
	Alternativos (*)	Ninguno.
Post-condiciones	El administrador visualiza los principales datos de las cuentas existentes en el sistema.	

Caso de uso 103: Visualizar cuentas

Caso de uso 104: Visualizar cuenta		
Descripción	Además de poder visualizar el conjunto de cuentas, puede que el administrador desee conocer todos los datos que componen una de ellas.	
Actores principal	Administrador.	
Pre-condiciones	El usuario se encuentra visualizando el conjunto de cuentas registradas en el sistema y desea conocer los datos de una de ellas.	
Flujos	Básico	<ol style="list-style-type: none">1. El administrador selecciona una cuenta y pide al sistema que le muestre toda la información que posea de ésta.2. El sistema recupera todos los datos y se los proporciona al usuario.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario dispone de todos los datos que buscaba.	

Caso de uso 104: Visualizar cuenta



Caso de uso 105: Añadir cuenta	
Descripción	La lista de usuarios que harán uso de accessManager a lo largo de toda la vida del centro será muy variable. Puede que surjan nuevos administradores, autorizadores, vigilantes,... que dejen o entren a trabajar nuevas empresas y departamentos... está claro que se hace imprescindible tener una forma de poder crear cuentas nuevas.
Actores principal	Administrador.
Pre-condiciones	Surge la necesidad de crear un nuevo usuario y el administrador se dispone a ello.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El administrador indica al sistema que desea crear una nueva cuenta. 2. El sistema le proporciona un formulario que el usuario deberá rellenar. 3. Tras editar todos los campos, el usuario le pide al sistema que realice el alta de la cuenta. 4. El sistema le contesta pidiéndole una autorización previa. 5. El usuario la otorga y así el sistema puede proceder con el alta de la nueva cuenta. (*)
	<p>Alternativos (*)</p> <p>El usuario introduce algún campo con un formato indeseado:</p> <ul style="list-style-type: none"> • El sistema le avisa del error que acaba de cometer y vuelve al punto 2 sin modificar ningún campo. <p>El usuario no da su visto bueno al registro de la cuenta:</p> <ul style="list-style-type: none"> • El sistema no realiza ninguna inserción en la base de datos y da por concluido el caso de uso.
Post-condiciones	El sistema dispone de una nueva cuenta.

Caso de uso 105: Añadir cuenta



Caso de uso 106: Borrar cuenta	
Descripción	Como dijimos en el caso de uso anterior, la lista de usuarios varía, y así como se proporciona la posibilidad de añadir cuentas, también debe existir la de eliminarlas.
Actores principal	Administrador.
Pre-condiciones	El usuario se encuentra en la pestaña de cuentas y desea eliminar alguna de ellas.
Flujos	Básico <ol style="list-style-type: none">1. El usuario selecciona las cuentas que no desea que persistan en el sistema y le pide a éste que las borre.2. El sistema le envía un mensaje al usuario avisándole de la acción que se dispone a realizar y pidiéndole que confirme si está seguro de eliminar las cuentas que ha seleccionado.3. El usuario contesta afirmativamente. (*)4. El sistema borra las cuentas seleccionadas.
	Alternativos (*) <p>El usuario contesta negativamente:</p> <ul style="list-style-type: none">• El sistema no realiza ningún borrado y el caso de uso finaliza.
Post-condiciones	El usuario consigue que las cuentas que seleccionó dejen de formar parte del sistema.

Caso de uso 106: Borrar cuenta



Caso de uso 107: Modificar cuenta	
Descripción	¿Qué pasa si se desea otorgar un tipo de usuario diferente a una cuenta para que ésta vea reducidos o aumentados sus privilegios? Parece claro que el que haya una forma de modificar el tipo de usuario (o cualquiera de sus otros datos) resulta más atractivo que tener que borrar y añadir una nueva cuenta.
Actores principal	Administrador.
Pre-condiciones	El usuario no está conforme con las características de una cuenta.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario selecciona la cuenta que desea modificar y traslada al sistema su intención. 2. Éste despliega un formulario con todos los datos de la cuenta seleccionada. 3. El usuario edita todo lo que estime oportuno y le pide al sistema que actualice el registro. 4. El sistema le pregunta si está seguro de querer llevar a cabo los cambios solicitados. 5. El usuario le indica que sí y el sistema procede. (*)
	<p>Alternativos (*)</p> <p>El usuario introduce algún campo erróneo:</p> <ul style="list-style-type: none"> • El sistema le informa del error que ha cometido y el caso de uso termina. <p>El usuario opta por abortar la modificación:</p> <ul style="list-style-type: none"> • Ésta no tiene lugar.
Post-condiciones	El usuario consigue que la cuenta que deseaba modificar ahora conste con los datos correctos.

Caso de uso 107: Modificar cuenta

Caso de uso 108: Listar cuentas	
Descripción	Uno de los requisitos que nos exige el cliente es que pueda tener la posibilidad de listar las cuentas existentes en el sistema.
Actores principal	Administrador.
Pre-condiciones	El administrador accede al sistema, se sitúa en el tab 'Cuentas' y se dispone a obtener un reporte con todas las que existen en estos momentos.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El administrador pide al sistema poder disponer de un reporte en PDF con todas las cuentas de accessManager. 2. El sistema le pide al usuario la ubicación en la que almacenar el listado. 3. El usuario se la proporciona y el sistema se encarga de elaborarlo. (*)
	<p>Alternativos (*)</p> <p>El usuario opta por no ofrecer una ruta:</p> <ul style="list-style-type: none"> • El sistema no realiza ningún reporte y el caso de uso termina.
Post-condiciones	El administrador dispone del reporte solicitado en la ruta indicada.

Caso de uso 108: Listar cuentas



4.17. Tipos de usuario

Caso de uso 109: Visualizar tipos de usuario		
Descripción	Éste y los cuatro casos de uso que le siguen no tienen tanta importancia pero resulta de utilidad desarrollarlos.	
Actores principal	Administrador.	
Pre-condiciones	El usuario desea visualizar de qué tipos de usuarios se componen las cuentas del sistema.	
Flujos	Básico	<ol style="list-style-type: none">1. El administrador loguea en el sistema y se sitúa en la pestaña de cuentas.2. El sistema accede a la base de datos para obtener y mostrar al usuario los datos que requiere.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario dispone de los datos de todos los tipos de usuario registrados en el sistema.	

Caso de uso 109: Visualizar tipos de usuario

Caso de uso 110: Visualizar tipo de usuario		
Descripción	Además de proporcionar una vía para poder visualizar todos los tipos de usuarios existentes, también hay que proporcionar otra de cara a que el administrador pueda consultar los datos de uno específico.	
Actores principal	Administrador.	
Pre-condiciones	El administrador se encuentra visualizando los tipos de usuario y quiere obtener toda la información de uno de ellos.	
Flujos	Básico	<ol style="list-style-type: none">1. El usuario selecciona el tipo de usuario a consultar.2. El sistema recupera los datos de éste y se los proporciona al usuario en un formulario.
	Alternativos (*)	Ninguno.
Post-condiciones	El administrador visualiza todos los datos que conforman el tipo de usuario que seleccionó.	

Caso de uso 110: Visualizar tipo de usuario



Caso de uso 111: Añadir tipo de usuario	
Descripción	El caso de uso 111 nos pide que sea posible añadir un tipo de usuario a los ya existentes.
Actores principal	Administrador.
Pre-condiciones	El usuario necesita dar de alta un nuevo tipo de usuario. Abre accessManager, se loguea y se dirige a la tab de cuentas.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario indica al sistema su intención de añadir un nuevo tipo de usuario. 2. El sistema le proporciona un formulario que deberá rellenar y aprobar para que el alta tenga lugar. (*)
	<p>Alternativos (*)</p> <p>El usuario rellena el formulario con algún dato con formato no válido:</p> <ul style="list-style-type: none"> • El sistema informa al usuario de que se ha producido un error y que el alta no ha podido realizarse. • Se vuelve al final del punto 1. <p>El usuario finalmente no aprueba el alta del nuevo tipo de usuario:</p> <ul style="list-style-type: none"> • El sistema no realiza ninguna inserción y el caso de uso finaliza.
Post-condiciones	El sistema pasa a almacenar el nuevo tipo de usuario.

Caso de uso 111: Añadir tipo de usuario

Caso de uso 112: Borrar tipo de usuario	
Descripción	Pueden ser varios los motivos que puedan llevar al administrador a eliminar un tipo de usuario.
Actores principal	Administrador.
Pre-condiciones	El administrador desea eliminar algún tipo de usuario.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El administrador selecciona los tipos de usuario que no desea que sigan formando parte del sistema, y le indica a éste que los elimine. 2. El sistema lanza una advertencia al administrador avisándole de la acción que está a punto de llevarse a cabo. 3. El administrador da el visto bueno y la acción tiene lugar. (*)
	<p>Alternativos (*)</p> <p>El administrador no da el visto bueno:</p> <ul style="list-style-type: none"> • El sistema no borra ningún registro de la base de datos y da por concluido el caso de uso.
Post-condiciones	El sistema ya no dispone de los tipos de usuario eliminados.

Caso de uso 112: Borrar tipo de usuario



Caso de uso 113: Modificar tipo de usuario	
Descripción	Este caso de uso permitirá al usuario modificar cualquiera de los datos de los que consta un tipo de usuario.
Actores principal	Administrador.
Pre-condiciones	Un tipo de usuario consta con datos erróneos, el administrador se da cuenta y se dispone a corregirlo.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario selecciona el tipo de usuario que desea editar. 2. El sistema le proporciona todos los datos del tipo de usuario a través de un formulario. 3. El usuario modifica los que estima oportunos y pide al sistema que lleve a cabo la actualización. (*) 4. El sistema le pide una autorización para poder actuar. 5. El usuario se la da. (*)
	<p>Alternativos (*)</p> <p>El usuario modifica los datos con otros que tienen un formato no deseado:</p> <ul style="list-style-type: none"> • El sistema le indica el error y que no puede realizar la actualización. <p>El usuario responde negativamente ante el aviso por parte del sistema:</p> <ul style="list-style-type: none"> • Éste no realiza ninguna acción y da por concluido el caso de uso.
Post-condiciones	El administrador consigue que el tipo de usuario que constaba con datos erróneos ahora lo haga con válidos.

Caso de uso 113: Modificar tipo de usuario

4.18. Historial de accesos al sistema

Caso de uso 114: Visualizar accesos al sistema	
Descripción	El administrador del sistema debe tener la posibilidad de consultar qué usuarios han accedido a él.
Actores principal	Administrador.
Pre-condiciones	El usuario loguea en el sistema con la intención de conocer qué usuarios lo han utilizado.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El administrador selecciona la pestaña de cuentas y define la búsqueda que desea realizar. 2. El sistema se encarga de mostrarle qué usuarios han accedido al sistema según los parámetros de búsqueda proporcionados y los datos residentes en la base de datos.
	<p>Alternativos (*)</p> <p>Ninguno.</p>
Post-condiciones	El usuario visualiza los accesos al sistema que cumplen con los parámetros establecidos en la búsqueda.

Caso de uso 114: Visualizar accesos al sistema



Caso de uso 115: Listar accesos al sistema	
Descripción	El administrador puede consultar qué accesos se han producido en el sistema bajo unas condiciones dadas, pero también nos pide que pueda almacenar los resultados de esta búsqueda a través de un reporte en PDF.
Actores principal	Administrador.
Pre-condiciones	El usuario se encuentra en el sistema, y más concretamente en la tab 'Cuentas' con la intención de obtener un reporte con los datos que visualiza en la tabla de accesos al sistema.
Flujos	Básico <ol style="list-style-type: none">1. El administrador pide al sistema que genere un reporte con los datos que pueblan la tabla de accesos al sistema.2. Éste le requiere una ruta en la que depositar el listado PDF.3. El usuario le ofrece la deseada y le indica que genere el reporte. (*)
	Alternativos (*) <p>El usuario no le indica ninguna ruta al decidir que finalmente no desea obtener el reporte:</p> <ul style="list-style-type: none">• El sistema da por finalizado el caso de uso sin realizar ninguna operación.
Post-condiciones	El usuario obtiene el reporte que deseaba en la ubicación que ha proporcionado a accessManager.

Caso de uso 115: Listar accesos al sistema



4.19. Cuentas - Parte 2

Caso de uso 116: Modificar datos de cuenta		
Descripción	Las cuentas se generan con un nombre de usuario asignado por el administrador y una contraseña por defecto ("cambiar"). Esta más que claro que será necesario que los usuarios cambien al menos una vez los datos de sus cuentas, y más veces si lo estiman oportuno.	
Actores principal	Administrador, Autorizador, Vigilante y RP.	
Pre-condiciones	El usuario recibe una cuenta para usar en el sistema o ve necesario cambiar los datos que conforman alguna de las que tenga. Para ello loguea en el sistema con la cuenta que desea modificar.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario indica al sistema que desea actualizar los datos de su cuenta. 2. Éste le muestra un dialogo con sus datos actuales. 3. El usuario decide que dato desea cambiar (nombre de usuario o contraseña). 4. Si el usuario elige cambiar: <ol style="list-style-type: none"> a. El nombre de usuario: <ul style="list-style-type: none"> • El sistema le proporciona un formulario en el que deberá escribir su contraseña actual y el nuevo nombre de usuario. b. La contraseña: <ul style="list-style-type: none"> • El sistema le proporciona un formulario en el que deberá escribir su contraseña actual y su nueva contraseña por dos ocasiones. 5. Si los datos actuales son válidos, el sistema avisa al usuario que va a proceder a actualizar los datos indicados, pidiéndole una aprobación previa. (*) 6. El usuario da su aprobación. (*) 7. El sistema recoge los datos y realiza la modificación. (*)
	Alternativos (*)	<p>El usuario termina por no dar su aprobación:</p> <ul style="list-style-type: none"> • El sistema da por concluido en este punto el caso de uso. <p>La contraseña introducida no es válida o el usuario introduce un nombre de usuario existente o dos contraseñas distintas o un nombre de usuario o contraseña con un formato indeseado:</p> <ul style="list-style-type: none"> • El sistema avisa del error y vuelve al punto 2, sin modificar ningún dato.
Post-condiciones	El usuario dispone de su cuenta actualizada con los datos que desea.	

Caso de uso 116: Modificar datos de cuenta



4.20. Recordatorios y borrado

Caso de uso 117: Recordar personal a punto de caducar		
Descripción	En el momento de diseñar la base de datos que da soporte a todo el sistema, comentamos que las empresas que trabajan en la sede de Informática Navarra S.A. deben actualizar de forma periódica a su personal. Hasta el desarrollo de este proyecto, el jefe de área de producción era el encargado de avisar diariamente a los departamentos cuyo personal estaba a punto de caducar, pero una vez que accessManager esté en funcionamiento podremos automatizar esta tarea.	
Actores principal	Sistema.	
Pre-condiciones	El usuario que accede al sistema es el primer vigilante en hacerlo a lo largo del día.	
Flujos	Básico	El sistema obtiene qué departamentos tienen su personal a <i>d1</i> , <i>d2</i> y <i>d3</i> días de caducar y envía un E-Mail a todos los responsables de personal de éstos para avisarles de la situación.
	Alternativos (*)	Ninguno.
Post-condiciones	El sistema insta a los responsables de personal de los departamentos implicados a que actualicen su personal si no quieren que éste sea eliminado, liberando al administrador de esta responsabilidad.	

Caso de uso 117: Recordar personal a punto de caducar

Caso de uso 118: Recordar préstamos pendientes		
Descripción	La tarea anteriormente citada no es la única que realiza el administrador. Recordemos que existirán préstamos que no sean devueltos a la salida del centro. Informática Navarra S.A. no puede dar por perdido todo este material, lo que hace que el jefe de área de producción, además de recordar a los departamentos que le envíen la lista de su personal, también es el encargado de ir recordando a todas aquellas personas que tengan préstamos pendientes que los devuelvan con la mayor brevedad posible. Le ayudaremos también en esta tarea.	
Actores principal	Sistema.	
Pre-condiciones	El vigilante del primer turno del día arranca la aplicación.	
Flujos	Básico	El sistema obtiene el conjunto de trabajadores que poseen préstamos pendientes desde hace <i>d1</i> , <i>d2</i> y <i>d3</i> días y a todos aquellos de los que disponga de alguna dirección de correo electrónico les envía un mensaje informándoles del material que tienen pendiente de devolución.
	Alternativos (*)	Ninguno.
Post-condiciones	Los prestatarios que cumplen con las características impuestas son avisados vía E-Mail, ayudando así al jefe de área de producción en la recuperación del material "extraviado".	

Caso de uso 118: Recordar préstamos pendientes



Caso de uso 119: Borrar datos obsoletos					
Descripción	accessManager es al fin y al cabo un almacén de datos que alguien debe mantener si quiere conservarlo en condiciones óptimas; no podemos dejar que crezca a su libre albedrío. Es verdad que siempre deberá existir un administrador del sistema que lo conozca y pueda realizar sobre éste un mantenimiento de garantías, pero nunca está de más que ayudemos con aquellas tareas que se puedan automatizar.				
Actores principal	Administrador.				
Pre-condiciones	El sistema registra el acceso al sistema del primer vigilante del día.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> accessManager obtiene y borra: <ul style="list-style-type: none"> • Todos aquellos trabajadores ocasionales cuyo permiso de acceso ha caducado. • Todos aquellos departamentos cuya fecha máxima de renovación de personal haya expirado. • Todas aquellas personas que no constan en ninguna tabla especializada. • Todos los accesos, llaveros, llaves, niveles, colores y tarjetas (en desuso, dados de baja) cuya existencia en la base de datos excede de <i>d</i> días. </td> </tr> <tr> <td>Alternativos (*)</td> <td>Ninguno.</td> </tr> </table>	Básico	accessManager obtiene y borra: <ul style="list-style-type: none"> • Todos aquellos trabajadores ocasionales cuyo permiso de acceso ha caducado. • Todos aquellos departamentos cuya fecha máxima de renovación de personal haya expirado. • Todas aquellas personas que no constan en ninguna tabla especializada. • Todos los accesos, llaveros, llaves, niveles, colores y tarjetas (en desuso, dados de baja) cuya existencia en la base de datos excede de <i>d</i> días. 	Alternativos (*)	Ninguno.
	Básico	accessManager obtiene y borra: <ul style="list-style-type: none"> • Todos aquellos trabajadores ocasionales cuyo permiso de acceso ha caducado. • Todos aquellos departamentos cuya fecha máxima de renovación de personal haya expirado. • Todas aquellas personas que no constan en ninguna tabla especializada. • Todos los accesos, llaveros, llaves, niveles, colores y tarjetas (en desuso, dados de baja) cuya existencia en la base de datos excede de <i>d</i> días. 			
Alternativos (*)	Ninguno.				
Post-condiciones	La población residente en la base de datos está actualizada y es la idónea para el correcto funcionamiento del sistema.				

Caso de uso 119: Borrar datos obsoletos



4.21. Partes de seguridad

Caso de uso 120: Visualizar partes de seguridad	
Descripción	El centro está custodiado constantemente por vigilantes jurados que deben realizar un parte de incidencias por servicio. Esta tarea se realizaba antes de forma manual y en el momento que bien el administrador o bien el vigilante deseaban consultar cualquier dato acudían al “taco” de partes. Ahora, al informatizar esta tarea y ser tan grande el volumen de partes de seguridad que pueden llegar a almacenarse, se debe dotar a la aplicación de un método que permita visualizar todos o parte de ellos, para después seleccionar el que se desea visionar (tratado en el siguiente caso de uso).
Actores principal	Administrador y Vigilante.
Pre-condiciones	El usuario accede al sistema y se dirige a la pestaña ‘Partes de seguridad’ con la intención de visionar los existentes.
Flujos	Básico 5. El sistema muestra en un principio todos los partes de seguridad. 6. El vigilante restringe la muestra de partes mediante los parámetros de búsqueda de los que dispone si lo cree preciso. (*) 7. El sistema actualiza los partes visionados mostrando solo los que cumplan con los criterios establecidos.
	Alternativos (*) El usuario no filtra la búsqueda ya que los resultados que observa por pantalla cumplen con sus expectativas: <ul style="list-style-type: none">• El sistema da por finalizado el caso de uso sin realizar ninguna acción más.
Post-condiciones	El usuario visualiza en pantalla el conjunto de partes de seguridad que deseaba.

Caso de uso 120: Visualizar partes de seguridad



Caso de uso 121: Visualizar parte de seguridad		
Descripción	Una vez que se visualiza los partes de seguridad que se desean, es necesario completar el proceso de visionado dando la posibilidad al usuario de acceder a la información que éstos almacenan.	
Actores principal	Administrador y Vigilante.	
Pre-condiciones	El usuario se encuentra logueado en el sistema, más concretamente en la tab de partes de seguridad, y desea visualizar los datos que componen un parte que observa en pantalla.	
Flujos	Básico	6. El usuario selecciona el parte de seguridad del que desea obtener los datos que lo componen. 7. El sistema se encarga de acudir a la base de datos, rescatar los datos del parte y mostrárselos al usuario en pantalla.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario ha conseguido lo que buscaba: visualizar todos los datos del parte de seguridad seleccionado.	

Caso de uso 121: Visualizar parte de seguridad

Caso de uso 122: Añadir parte de seguridad		
Descripción	Un nuevo vigilante de seguridad comienza su labor diaria en el centro y a lo largo de su turno de trabajo se dispone a crear el parte de seguridad que debe redactar.	
Actores principal	Vigilante.	
Pre-condiciones	El vigilante se encuentra logueado en el sistema y visita la sección destinada a los partes de seguridad.	
Flujos	Básico	4. El vigilante transmite al sistema su intención de crear un nuevo parte de seguridad. 5. El sistema le proporciona un formulario para tal fin. 6. El vigilante lo rellena y pide al sistema que lo almacene. 7. El sistema obedecerá, pero antes deberá recibir una confirmación previa. (*)
	Alternativos (*)	El usuario proporciona datos que no tienen el formato esperado por el sistema: <ul style="list-style-type: none"> El sistema le avisa del error y vuelve al punto 2, sin haber modificado ningún dato. El usuario termina por no dar su visto bueno a la inserción del nuevo parte: <ul style="list-style-type: none"> El sistema da por concluido el caso de uso sin realizar ninguna inserción.
Post-condiciones	El sistema dispone de un nuevo parte de seguridad en su tabla Parte_Seguridad.	

Caso de uso 122: Añadir parte de seguridad



Caso de uso 123: Borrar parte de seguridad	
Descripción	Ésta será una funcionalidad muy poco utilizada y dirigida especialmente al administrador, aunque el vigilante también podrá hacer uso de ella. No es normal que cualquiera borre un parte de seguridad, más aún el vigilante, que dispone de la opción de edición, pero el cliente nos ha pedido que proveamos la posibilidad de borrar los partes de seguridad que se desee almacenar.
Actores principal	Administrador y Vigilante.
Pre-condiciones	El usuario quiere eliminar alguno de los partes de seguridad que visualiza en pantalla.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 3. El usuario selecciona los partes de seguridad a eliminar y pide al sistema que llevé a cabo la operación de borrado. 4. El sistema envía un mensaje al usuario avisándole de qué partes va a eliminar y si está seguro de continuar con la acción. 5. El usuario responde afirmativamente ante esta pregunta. (*) 6. El sistema borra los partes de seguridad indicados.
	<p>Alternativos (*)</p> <p>El usuario responde negativamente:</p> <ul style="list-style-type: none"> • El sistema no borra ningún elemento de la base de datos y el caso de uso llega a su fin.
Post-condiciones	accessManager ya no dispone de los partes de seguridad que el usuario ha seleccionado y eliminado.

Caso de uso 123: Borrar parte de seguridad



Caso de uso 124: Modificar parte de seguridad	
Descripción	Esta sí que será una funcionalidad muy utilizada, ya que lo normal es que un vigilante de seguridad abra un parte de seguridad al iniciar su servicio y lo vaya completando a medida que éste transcurre.
Actores principal	Vigilante.
Pre-condiciones	Surge alguna incidencia que el vigilante debe anotar o éste percata de que un parte posee algún dato erróneo, lo que lleva al vigilante a entrar en el sistema, buscar el parte y seleccionarlo.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 3. El vigilante indica al sistema que desea editar el parte que se encuentra visualizando. 4. El sistema hace los campos del formulario editables. (*) 5. El usuario edita todos lo que estima oportuno y pide al sistema que los registre. 6. El sistema obedece tras una previa aprobación por parte del usuario. (*)
	<p>Alternativos (*)</p> <p>El sistema no permite la edición del formulario ya que el usuario no dispone de privilegios suficientes.</p> <p>El usuario introduce datos con el formato indebido:</p> <ul style="list-style-type: none"> • El sistema avisa al usuario del error y no realiza ninguna actualización. <p>El usuario aborta la edición del parte:</p> <ul style="list-style-type: none"> • El caso de uso finaliza y ningún registro es modificado.
Post-condiciones	El parte de seguridad, que en un principio no era correcto, ahora sí lo es tras ser editado por el vigilante.

Caso de uso 124: Modificar parte de seguridad



Caso de uso 125: Listar parte de seguridad	
Descripción	Para que administrador y vigilantes de seguridad puedan disponer de copias en papel de los partes de seguridad que redactan, el cliente nos pide que proporcionemos una vía para poder exportarlos en formato PDF.
Actores principal	Administrador y Vigilante.
Pre-condiciones	El administrador o vigilante se encuentra visualizando un parte de seguridad y cree conveniente reportarlo.
Flujos	Básico 4. El usuario pide al sistema que exporte el parte que se visualiza en pantalla. 5. El sistema le pide al usuario que indique la ruta en donde desea almacenarlo. 6. El vigilante la indica. (*) 7. El sistema se encarga de crear el reporte con el parte de seguridad.
	Alternativos (*) El usuario no proporciona ninguna ruta: <ul style="list-style-type: none">• El sistema no elabora ningún reporte y el caso de uso finaliza.
Post-condiciones	El usuario puede disfrutar del parte de seguridad en formato PDF.

Caso de uso 125: Listar parte de seguridad



4.22. Notas

Caso de uso 126: Redactar nota	
Descripción	Ahora, en la iteración destinada a las notas, vamos proporcionar un servicio sencillo de mensajería. En primer lugar describimos cómo un trabajador podría enviar un mensaje a otro.
Actores principal	Administrador, Autorizador y Vigilante.
Pre-condiciones	Un trabajador quiere enviar un mensaje a otro utilizando el servicio de mensajería de accessManager. Para ello loguea en el sistema y se sitúa en la tab de notas.
Flujos	Básico <ol style="list-style-type: none">1. El usuario indica al sistema que desea redactar una nota.2. El sistema le proporciona un formulario con el que redactarla e indicar a qué destinatarios irá dirigida.3. El usuario la escribe e indica sus receptores.4. El sistema recoge toda la información y pide al usuario que apruebe el envío.5. El usuario da su visto bueno y el sistema se encarga de enviar la nota. (*)
	Alternativos (*) <p>El usuario escribe una nota con alguno de sus datos con un formato erróneo:</p> <ul style="list-style-type: none">• El sistema indica el error ocurrido y no envía ninguna nota, pasándose al punto 2 con el formulario intacto. <p>El usuario ve que se ha equivocado en alguno de los campos, o por las razones que sea decide abortar el envío de la nota contestando negativamente ante el aviso de envío:</p> <ul style="list-style-type: none">• El sistema cierra el diálogo de aviso y no ejecuta ninguna acción más.
Post-condiciones	El usuario ha enviado la nota a los destinatarios que deseaba.

Caso de uso 126: Redactar nota



Caso de uso 127: Visualizar notas recibidas	
Descripción	Una vez que tenemos la posibilidad de enviar notas, pasamos al administración de éstas, tanto las de carácter entrante (recibidas) como las de carácter saliente (enviadas). En cuanto a las primeras cabe decir que no podemos proporcionar simultáneamente una visión de todas ellas con todos los datos que las componen, ya que la cantidad de notas que un usuario puede disponer es ilimitada. Nos encontramos ante un caso muy similar al de los partes de seguridad. Por tanto, atenderemos al visionado de todas las notas recibidas y el contenido de cada una de ellas en éste y el siguiente caso de uso respectivamente.
Actores principal	Administrador, Autorizador y Vigilante.
Pre-condiciones	El usuario accede a la pestaña de notas con la intención de comprobar sus notas entrantes.
Flujos	Básico 1. El usuario accede a la sección de notas entrantes. 2. El sistema se encarga de obtener todas las notas entrantes de éste y mostrárselas.
	Alternativos (*) Ninguno.
Post-condiciones	El usuario conoce qué notas componen su “bandeja de entrada”.

Caso de uso 127: Visualizar notas recibidas

Caso de uso 128: Visualizar nota recibida	
Descripción	Tras disponer de la lista de notas que tenemos entrantes, nos queda proporcionar la posibilidad de leerlas.
Actores principal	Administrador, Autorizador y Vigilante.
Pre-condiciones	El usuario se encuentra visionando las notas que posee entrantes y desea realizar la lectura de alguna de ellas.
Flujos	Básico 1. El usuario selecciona la nota que desea leer. 2. El sistema acude a la base de datos para recuperar los datos de ésta y mostrárselos al usuario a través de un formulario.
	Alternativos (*) Ninguno.
Post-condiciones	El usuario visualiza por pantalla toda la composición de la nota.

Caso de uso 128: Visualizar nota recibida



Caso de uso 129: Borrar nota recibida					
Descripción	Las notas recibidas no tienen por qué ser almacenadas de manera indefinida; el usuario cuando no necesite más de una de ellas debe poder eliminarla.				
Actores principal	Administrador, Autorizador y Vigilante.				
Pre-condiciones	El usuario se encuentra logueado en el sistema, más concretamente visualizando sus notas entrantes en la pestaña 'Notas', y desea dejar de conservar algunas de ellas.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> 8. El usuario selecciona las notas que desea eliminar de su "bandeja de entrada", trasladando al sistema su intención de no seguir conservándolas. 9. El sistema le envía un mensaje indicándole que va a proceder con el borrado de éstas, pero que previamente necesita su autorización. 10. El usuario la da y así el sistema puede finalizar el caso de uso con el objetivo cumplido. (*) </td> </tr> <tr> <td>Alternativos (*)</td> <td> El usuario decide no eliminar las notas respondiendo 'No': <ul style="list-style-type: none"> El sistema no elimina ningún registro de la base de datos y da por concluido el caso de uso. </td> </tr> </table>	Básico	8. El usuario selecciona las notas que desea eliminar de su "bandeja de entrada", trasladando al sistema su intención de no seguir conservándolas. 9. El sistema le envía un mensaje indicándole que va a proceder con el borrado de éstas, pero que previamente necesita su autorización. 10. El usuario la da y así el sistema puede finalizar el caso de uso con el objetivo cumplido. (*)	Alternativos (*)	El usuario decide no eliminar las notas respondiendo 'No': <ul style="list-style-type: none"> El sistema no elimina ningún registro de la base de datos y da por concluido el caso de uso.
	Básico	8. El usuario selecciona las notas que desea eliminar de su "bandeja de entrada", trasladando al sistema su intención de no seguir conservándolas. 9. El sistema le envía un mensaje indicándole que va a proceder con el borrado de éstas, pero que previamente necesita su autorización. 10. El usuario la da y así el sistema puede finalizar el caso de uso con el objetivo cumplido. (*)			
Alternativos (*)	El usuario decide no eliminar las notas respondiendo 'No': <ul style="list-style-type: none"> El sistema no elimina ningún registro de la base de datos y da por concluido el caso de uso. 				
Post-condiciones	Las notas borradas dejan de constar en la lista de notas entrantes del usuario.				

Caso de uso 129: Borrar nota recibida

Caso de uso 130: Visualizar notas enviadas					
Descripción	Todo lo que proponemos para la recepción de notas también lo haremos para el envío, además con una funcionalidad nueva añadida que comentaremos en el siguiente caso de uso. En éste tratamos el visionado de todas las notas que hemos enviado y seguimos conservando.				
Actores principal	Administrador, Autorizador y Vigilante.				
Pre-condiciones	El usuario despliega accessManager y se dirige a la pestaña 'Notas' con la idea de observar las que ha enviado y sigue almacenando.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> 1. El usuario accede a la sección de notas enviadas. 2. El sistema recupera de la base de datos la lista de notas de las que dispone el usuario como enviadas y las muestra por pantalla. </td> </tr> <tr> <td>Alternativos (*)</td> <td>Ninguno.</td> </tr> </table>	Básico	1. El usuario accede a la sección de notas enviadas. 2. El sistema recupera de la base de datos la lista de notas de las que dispone el usuario como enviadas y las muestra por pantalla.	Alternativos (*)	Ninguno.
	Básico	1. El usuario accede a la sección de notas enviadas. 2. El sistema recupera de la base de datos la lista de notas de las que dispone el usuario como enviadas y las muestra por pantalla.			
Alternativos (*)	Ninguno.				
Post-condiciones	El usuario consigue lo que se proponía: visualizar qué notas dispone en el "buzón de salida".				

Caso de uso 130: Visualizar notas enviadas



Caso de uso 131: Visualizar nota enviada		
Descripción	El usuario, así como puede visualizar qué notas ha recibido, también podrá visualizar las que ha enviado. El visionado de la información con la que fue redactada quizá no tenga tanto sentido como pueda ser el de las recibidas, pero el usuario nos pide que pueda visualizar qué personas la han leído, y esto sí que puede ser de gran utilidad.	
Actores principal	Administrador, Autorizador y Vigilante.	
Pre-condiciones	El usuario se encuentra visualizando las notas que componen su "bandeja de salida" y desea consultar una nota de las que la componen, bien sea por recordar qué escribió en ella o por comprobar que destinatarios la han leído.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario clics sobre la nota que desea visualizar. 2. El sistema se encarga de desplegar en pantalla todos los datos que posee de ésta.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario visualiza en pantalla todos los datos que buscaba.	

Caso de uso 131: Visualizar nota enviada

Caso de uso 132: Borrar nota enviada		
Descripción	A medida que transcurre el tiempo el usuario reusará seguir conservando notas que redactó, ya sea porque todos sus destinatarios la hayan leído, porque no le interese estar informado sobre su situación,... Entonces, proporcionaremos una forma de poder borrar notas enviadas al igual que la proporcionamos con las recibidas.	
Actores principal	Administrador, Autorizador y Vigilante.	
Pre-condiciones	El usuario loguea en el sistema, clics la pestaña de notas y se dirige a la sección de notas enviadas.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona el conjunto de notas que no desea conservar y clics sobre el botón de borrado. 2. El sistema pregunta al usuario si está seguro de querer continuar con este borrado. 3. El usuario contesta afirmativamente. (*) 4. El sistema actualiza la base de datos de modo que el usuario no visualice más las notas seleccionadas pero sus receptores puedan seguir conservándolas.
	Alternativos (*)	El usuario contesta negativamente: <ul style="list-style-type: none"> • El sistema no modifica ningún dato y el caso de uso finaliza.
Post-condiciones	El usuario descarta las notas que había seleccionado, no volviendo a disponer de ellas.	

Caso de uso 132: Borrar nota enviada



4.23. Base de datos

Caso de uso 133: Importar base de datos	
Descripción	Cabe la posibilidad de que la base de datos deba ser restaurada o surja un conjunto de registros que deban ser insertados. Parece muy mala idea ir insertando todas y cada una de las tuplas una a una mediante la interfaz gráfica, con que dotaremos de la funcionalidad de importación que puede proporcionar el SGBD pero sin tener que pasar por éste.
Actores principal	Administrador.
Pre-condiciones	El administrador de la base de datos desea importar un fichero con código SQL con sentencias a ejecutar sobre la base de datos.
Flujos	Básico 6. El usuario indica al sistema que desea importar un fichero SQL. 7. El sistema abre un diálogo en el que el usuario deberá buscar y seleccionar el fichero a importar. (*) 8. El sistema realiza la importación de todo el código existente en el fichero indicado. (*)
	Alternativos (*) El usuario cancela la selección del fichero: <ul style="list-style-type: none">El sistema no realiza ninguna acción y el caso de uso concluye. El fichero SQL contiene algún tipo de error: <ul style="list-style-type: none">El sistema avisa del error que ha sucedido y aborta la importación de código en el punto en que éste se produjo.
Post-condiciones	El administrador importa a la base de datos todo el código SQL que residía en el fichero seleccionado.

Caso de uso 133: Importar base de datos

Caso de uso 134: Exportar base de datos					
Descripción	La población residente en la base de datos irá aumentando a medida que los días se sucedan. El administrador no puede arriesgarse a perder toda esta valiosa información con que deberá realizar copias de la misma periódicamente. Con este caso de uso pretendemos proporcionarle una vía fácil y cómo mediante la cual exportar toda la población a un fichero sin tener que mediar con el SGBD.				
Actores principal	Administrador.				
Pre-condiciones	El administrador desea exportar todo el contenido de la base de datos a un fichero.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> <ol style="list-style-type: none"> 3. El administrador transmite al sistema su intención. 4. El sistema abre un diálogo en el que el administrador deberá indicar la ruta y el nombre del fichero en el que depositar toda la información. 5. El administrador indicar la ruta y el sistema realiza la exportación solicitada. (*) </td> </tr> <tr> <td>Alternativos (*)</td> <td> El administrador cancela el cuadro de selección de fichero: <ul style="list-style-type: none"> • El sistema no realiza ninguna exportación. </td> </tr> </table>	Básico	<ol style="list-style-type: none"> 3. El administrador transmite al sistema su intención. 4. El sistema abre un diálogo en el que el administrador deberá indicar la ruta y el nombre del fichero en el que depositar toda la información. 5. El administrador indicar la ruta y el sistema realiza la exportación solicitada. (*) 	Alternativos (*)	El administrador cancela el cuadro de selección de fichero: <ul style="list-style-type: none"> • El sistema no realiza ninguna exportación.
	Básico	<ol style="list-style-type: none"> 3. El administrador transmite al sistema su intención. 4. El sistema abre un diálogo en el que el administrador deberá indicar la ruta y el nombre del fichero en el que depositar toda la información. 5. El administrador indicar la ruta y el sistema realiza la exportación solicitada. (*) 			
Alternativos (*)	El administrador cancela el cuadro de selección de fichero: <ul style="list-style-type: none"> • El sistema no realiza ninguna exportación. 				
Post-condiciones	El administrador consigue lo que perseguía: crear un fichero con toda la población existente en la base de datos.				

Caso de uso 127: Visualizar notas recibidas

5. Aprobaciones y firmas

Con las siguientes firmas queda aprobado el documento '*Especificación de los casos de uso*' por parte del cliente y del equipo de desarrollo.

Pamplona, a 16 de abril de 2010.



Héctor Napal Basterrica
Responsable del proyecto



Pedro Martínez Pérez
Jefe de área de producción de Informática Navarra S.A.



accessManager

Desarrollo de un sistema de control de accesos a un centro

Planificación del proyecto

Fecha de entrega: 16 de abril de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Estructura del documento

2. El proceso de desarrollo

3. Iteraciones

4. Artefactos



1. Introducción

1.1. Propósito

Redactamos este artefacto con el objetivo de exponer brevemente todo el ciclo de desarrollo que hemos seguido para la confección del producto software encargado.

1.2. Audiencia

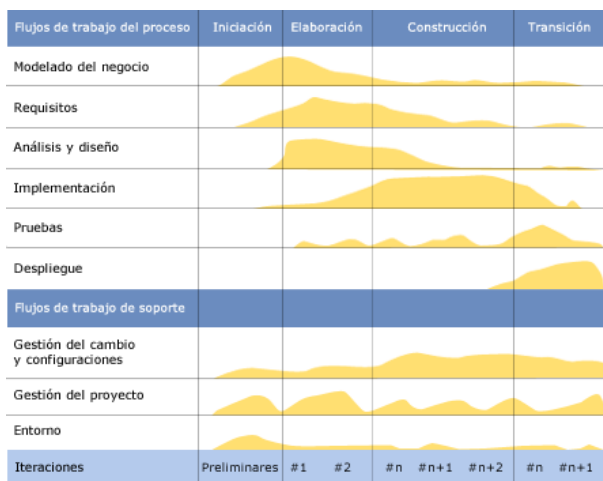
El destinatario no es otro que el cliente. No es un documento de gran interés, pero servirá para justificar la distribución de tiempos utilizada, relacionar las secciones en que se divide la documentación con las iteraciones en las que ha sido realizada,... a fin de cuentas, clarificar todo el proceso seguido y los resultados obtenidos en él.

1.3. Estructura del documento

Iniciamos la redacción de este documento exponiendo las fases de que consta el proceso *RUP* empleado en el desarrollo del producto y qué tiempos abarcan. A continuación mostramos cómo hemos dividido cada una de éstas fases en iteraciones que nos han permitido obtener los artefactos expuestos en la última sección, la cuarta.

2. El proceso de desarrollo

Al inicio de esta memoria explicamos que las circunstancias y el tipo de producto a desarrollar hacían muy recomendable escoger *RUP* como proceso de desarrollo:



Proceso de desarrollo RUP estándar



La configurabilidad de este proceso permite que pueda ser adaptado a cualquier tipo de proyecto y necesidades. En lo que a nosotros respecta, en un principio dividimos las fases de la siguiente forma:

FASE	DURACIÓN
Estudio de oportunidad (inception)	1 semana
Elaboración	3 semanas
Construcción	8 semanas
Transición	2 semanas
TOTAL	14 semanas

Coste temporal final de las fases RUP

3. Iteraciones

Una vez disponemos de la división en fases, éstas se fragmentan a su vez en las siguientes iteraciones:

	Nombre de tarea	Duración	Comienzo	Fin
1	+ Estudio de oportunidad (inception)	7 días	jue 07/01/10	vie 15/01/10
7	- Elaboración	16 días	lun 18/01/10	lun 08/02/10
8	+ Iteración E1: Estudio detallado del problema y modelado de	2 días	lun 18/01/10	mar 19/01/10
10	+ Iteración E2: Captura de requisitos	8 días	mié 20/01/10	vie 29/01/10
14	+ Iteración E3: Definición de la arquitectura básica	2 días	lun 01/02/10	mar 02/02/10
18	+ Iteración E4: Planificación del proyecto	4 días	mié 03/02/10	lun 08/02/10
23	Arquitectura	0 días	lun 08/02/10	lun 08/02/10
24	- Construcción	41 días	mar 09/02/10	mar 06/04/10
25	+ Iteración C1: La base de datos	4 días	mar 09/02/10	vie 12/02/10
31	+ Iteración C2: API para la base de datos	5 días	lun 15/02/10	vie 19/02/10
36	+ Iteración C3: Accesos	5 días	lun 22/02/10	vie 26/02/10
42	+ Iteración C4: Empresas y personal	5 días	lun 01/03/10	vie 05/03/10
48	+ Iteración C5: Tarjetas, tipos de acceso y cargos	3 días	lun 08/03/10	mié 10/03/10
54	+ Iteración C6: Zonas, llaves y llaveros	5 días	jue 11/03/10	mié 17/03/10
60	+ Iteración C7: Préstamos	2 días	jue 18/03/10	vie 19/03/10
66	+ Iteración C8: Cuentas	2 días	lun 22/03/10	mar 23/03/10
72	+ Iteración C9: Partes de seguridad	3 días	mié 24/03/10	vie 26/03/10
78	+ Iteración C10: Notas	3 días	lun 29/03/10	mié 31/03/10
84	+ Iteración C11: Base de datos	2 días	jue 01/04/10	vie 02/04/10
90	+ Iteración C12: Diseño, personalización y configuración	2 días	lun 05/04/10	mar 06/04/10
96	Capacidad operacional	0 días	mar 06/04/10	mar 06/04/10
97	- Transición	8 días	mié 07/04/10	vie 16/04/10
98	+ Iteración T1: Preparación de la release final	2 días	mié 07/04/10	jue 08/04/10
100	+ Iteración T2: Entrega	3 días	vie 09/04/10	mar 13/04/10
102	+ Iteración T3: Entrenamiento, soporte y mantenimiento	3 días	mié 14/04/10	vie 16/04/10
104	Liberación del producto	0 días	vie 16/04/10	vie 16/04/10

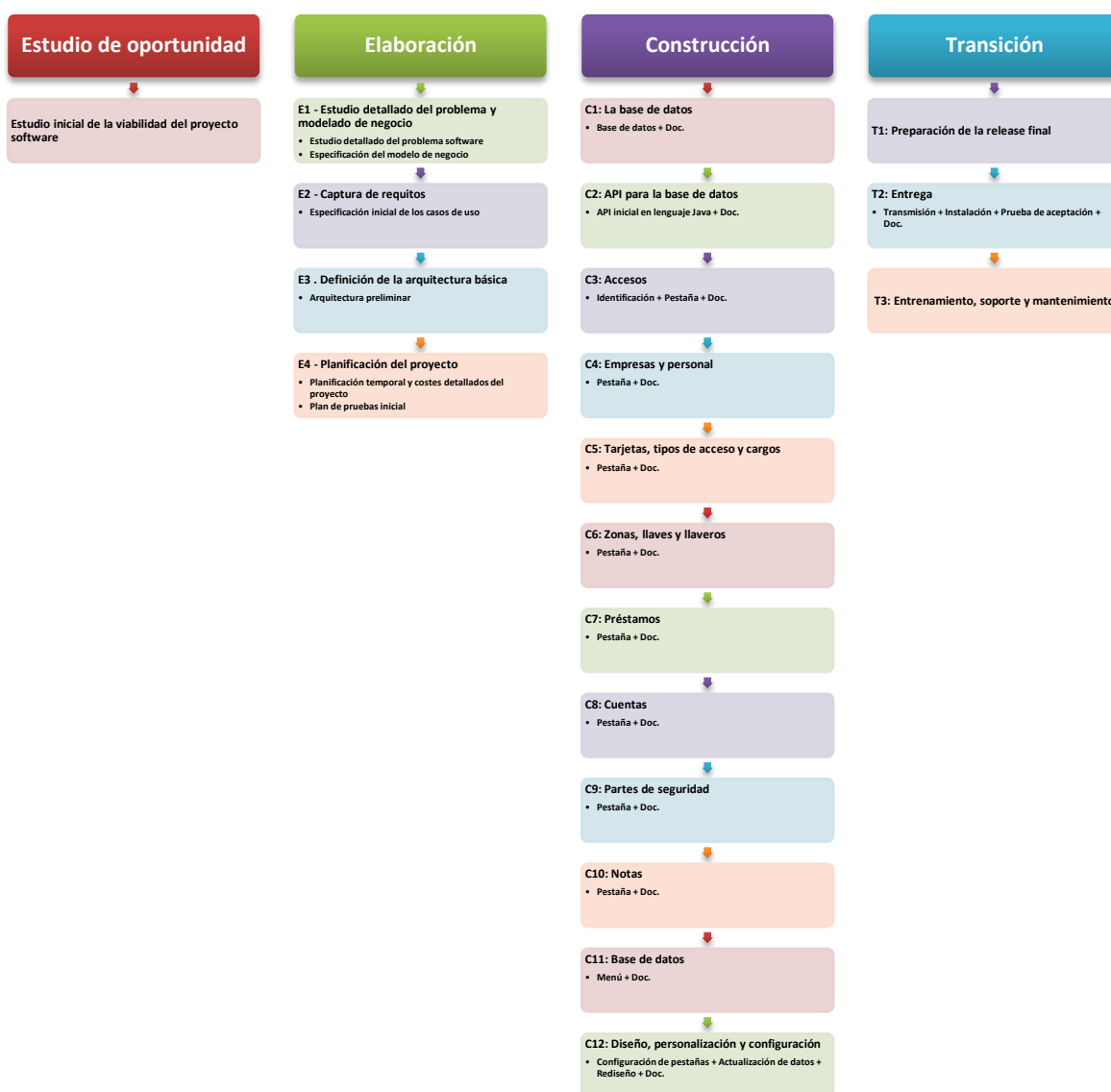
Esquema temporal de las iteraciones



4. Artefactos

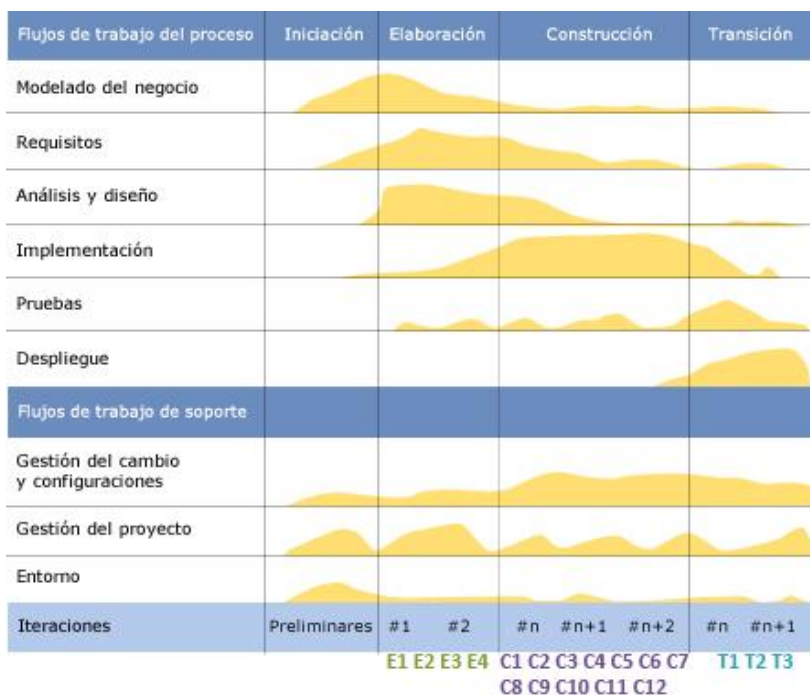
¿Qué resultados se consiguen en cada iteración? De forma muy resumida podríamos decir que el *Estudio de oportunidad* es la fase preliminar al inicio del desarrollo, donde se estudia si el sistema es viable y atractivo para el cliente. Una vez aceptado da comienzo la fase de *Elaboración*, destinada a estudiar y definir el sistema a construir, además de planificar todo el proceso que permita obtenerlo. A lo largo de la *Construcción* elaboramos iterativa e incrementalmente el sistema ideado, mientras que en la de *Transición* realizamos la transferencia y explicación del mismo de modo que el cliente pueda disponer de él, ponerlo en marcha y manejarlo eficazmente.

Concretando cada una de las iteraciones tendríamos el siguiente esquema...





... que también puede ser llevado sobre el primer gráfico expuesto en la página 5:



Proceso de desarrollo RUP seguido

Pamplona, a 16 de abril de 2010.

Héctor Napal Basterrica
Responsable del proyecto



accessManager
Desarrollo de un sistema de control de accesos a un centro

Esquema relacional de la base de datos

Fecha de entrega: 16 de abril de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Estructura del documento

2. Estudio del dominio de información ('Supuesto')

3. Diseño del modelo de datos

- 3.1. Esquema relacional
- 3.2. Breve descripción
- 3.3. Disparadores



1. Introducción

1.1. Propósito

Este documento nace con la intención de recoger de manera breve y precisa toda la documentación de interés necesaria para llegar a entender el modelo de datos que maneja el sistema desarrollado.

1.2. Audiencia

Se trata de un documento de consulta que necesitarán conocer y entender los futuros mantenedores del sistema para así poder realizar eficazmente sus funciones. El documento en su conjunto debería ser revisado por lo menos en una ocasión por todos ellos y de forma continuada el esquema relacional que en él se aporta.

1.3. Estructura del documento

En el capítulo 2 exponemos el supuesto que recoge todo el modelo de datos que ha sido necesario tratar para poder dar solución al sistema desarrollado. A continuación, en el tercero, exponemos el diseño que hemos realizado, aportando el esquema relacional elaborado junto con una breve explicación de las tablas que lo componen y los disparadores que lo complementan.

2. Estudio del dominio de información ('Supuesto')

Este es el modelo de datos recogido:

La **sede** consta de zonas.

El edificio puede constar de una o varias zonas, de las cuales bastará con almacenar su descripción y el nivel mínimo de acceso necesario para poder acceder a ella.



Sede de Informática Navarra S.A.

Varias zonas forman un nivel que es accesible única y exclusivamente si se dispone de una tarjeta que lo permita.



Plantas de los edificios

Los niveles requieren una numeración y una descripción que ayude a su comprensión.

El conjunto de **tarjetas** (numeración y color) del que se dispone está coloreado, correspondiéndose cada color a un único nivel.



Tarjetas coloreadas

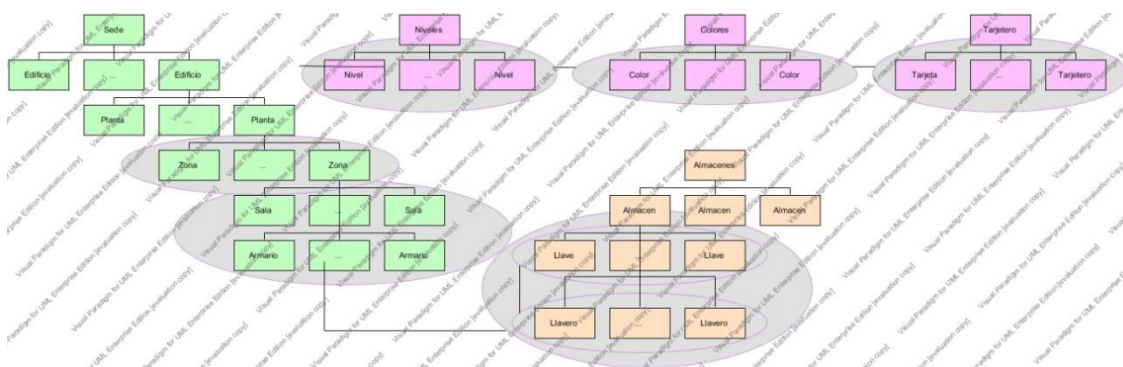
Una tarjeta sólo puede estar coloreada con un color y cada color sólo puede corresponder a un nivel de acceso. No obstante, pueden existir tarjetas coloreadas con distintos colores que proporcionen el mismo nivel. Cada tarjeta puede ser prestada a una persona, la cual puede acumular un número indefinido de préstamos.

También se dispone de un conjunto de **llaves** y **llaveros** disponibles en varios **almacenes**, que permitirán abrir puertas y armarios de las salas que componen cada zona.

Cada llave puede tener una o varias copias repartidas en uno o varios llaveros, además de poder tener existencia propia de manera individual (no asociada a ningún llavero). Todas las llaves y llaveros pueden ser almacenados de manera indistinta en uno o varios almacenes, de los cuales sólo es necesario almacenar una breve descripción. A partir de cada llave deberá poder identificarse a qué zona pertenece, qué abre, en qué lugares se encuentra almacenada y cuántas copias existen de la misma. Para los llaveros sólo será necesario saber qué llaves los componen, dónde están almacenados y de cuántas copias se disponen.

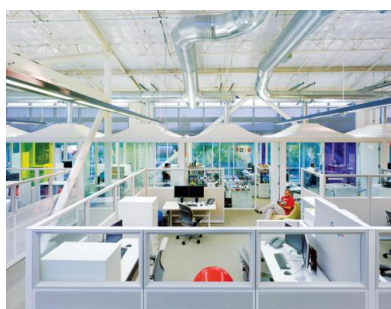


Armario de llaves



Sede, niveles, colores, tarjetas, llaves y llaveros

En cuanto a la vida diaria del centro, éste acoge el día a día de Informática Navarra S.A. y de varias **empresas**. Éstas se dividen en **departamentos**, a los cuales se les asigna un **tipo de acceso** en función de la actividad que realicen y el cual les permitirá acceder al centro bajo unas condiciones concretas.



Una empresa (CIF, nombre y dirección de sus oficinas centrales) consta de uno o varios departamentos, de los que se exige guardar la siguiente información:

- Nombre.
- Tipo de acceso que posee.
- Periodicidad con la que debe reafirmar la lista de personal que va a acceder al centro.
- Fecha en la que se realizó la última actualización de personal.
- DNI del trabajador (autorizador) de Informática Navarra S.A. que se hace responsable de la misma.
- Llaves que les está permitido tomar en préstamo a su personal a la hora de acceder al centro.



Los tipos de acceso son un combinado entre días, horas y nivel de acceso, así como de si es necesaria una autorización telefónica para acceder al centro (dependiendo del tipo de actividad que se realice) y qué tipo de actividad está permitido desarrollar. Se le asignará un tipo de acceso a cada departamento en función de la actividad que vaya a realizar, pudiendo ser asignado un mismo tipo a varios de ellos.

El **personal** que accede al centro se divide en trabajadores de Informática Navarra S.A, personal perteneciente a empresas (departamentos, en concreto) y personal que acude al centro de manera puntual (ocasional).

- Dentro de los primeros cabe distinguir a aquellos que se responsabilizarán de parte de los departamentos que trabajan en el centro (autorizadores) y a aquellos que ejercerán de administradores del sistema.
- En los segundos se encuentran encuadrados los vigilantes de seguridad, que realizarán las labores de vigilancia del centro y redactarán los **partes de seguridad** pertinentes (fecha, DNI del redactor e incidencias acontecidas en el servicio), pudiendo pertenecer en cada momento a una sola empresa-departamento de vigilancia contratada por el centro. Lo mismo ocurre con los trabajadores que acuden periódicamente al centro, pero no con los contactos con los que Informática Navarra S.A mantiene correspondencia, que podrán desempeñar uno o varios cargos dentro de cada empresa-departamento para la que trabajen. Entre estos últimos tendrán una gran importancia los responsables de personal, que serán los encargados de mantener actualizado el personal de las empresas-departamento en los que desempeñan tal cargo.
- El personal ocasional es aquel que acude al centro por motivos puntuales y también estará capacitado para obtener llaves y llaveros en préstamo.



Datos comunes a todo el personal son el DNI, apellidos, nombre y DNI del usuario que lo da de alta en el sistema. Como datos opcionales se tienen los números de teléfono o direcciones de correo electrónico que suministre cada persona para poder recibir comunicaciones o ser contactada en un momento determinado.

De manera especializada tendríamos los siguientes atributos:

- *Personal residente:* El nivel de acceso que posee, por si algún día olvida traer su tarjeta y es necesario prestarle una.
- *Personal periódico:* Empresa y departamento para el que trabaja.
- *Vigilantes:* Código de vigilante.
- *Contactos:* Empresas y departamentos a los que pertenecen junto con el cargo que desempeñan en cada caso. De los tipos de cargos sólo almacenaremos una breve descripción.
- *Personal ocasional:* Tipo de acceso otorgado, fecha de caducidad del permiso, notas a tener en cuenta y empresa a la que pertenece.



Todo el personal descrito con anterioridad (salvo los contactos) accede al centro para realizar sus labores. Dichos **accesos** deben quedar registrados según la política establecida en el documento de *Estudio detallado del problema software*.

Los accesos vienen determinados por la persona que accede al centro, la fecha y horas de entrada y salida, el motivo del acceso, la tarjeta que utiliza para moverse por el centro, otros datos de interés y la persona que ha permitido y registrado el acceso (normalmente será el vigilante).

No existe ningún tipo de restricción sobre el número de accesos y salidas del centro que un trabajador puede realizar con una tarjeta, pero si del tipo de acceso, que quedará determinado por el asignado al departamento al que pertenezca el trabajador periódico o el objeto de la visita realizada por la persona ocasional, o por el nivel de acceso en el caso del personal residente.

Al acceder al centro el trabajador puede requerir que le sea prestada alguna llave o llavero. En caso de existir copias y que el préstamo sea lícito, éste deberá ser registrado en el sistema (basta con suministrar al sistema la relación entre acceso y ente prestado).



Para que el sistema sea viable será necesario un mantenimiento y actualización constante por parte de los **usuarios** del mismo. Éstos, que se dividen en administradores, autorizadores, vigilantes y responsables de personal, dispondrán cada uno de una cuenta con la cual poder acceder al sistema y realizar las funciones que se les ha asignado. Será obligatorio llevar un registro de todos los accesos al sistema.

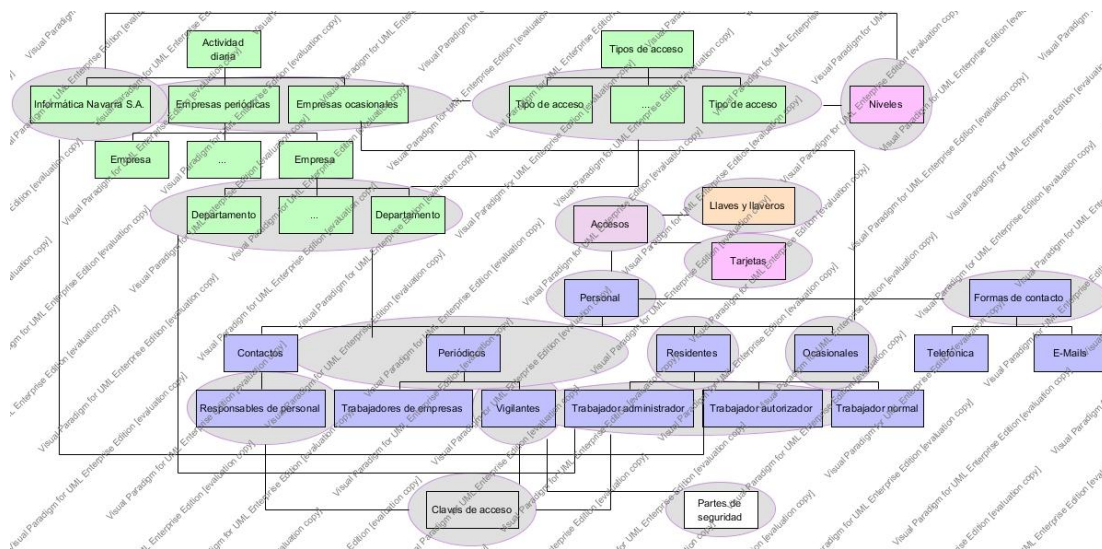


Cada cuenta estará determinada por un nombre de usuario y una contraseña, que permitirán loguear en el sistema, y un tipo de usuario, del que se desea almacenar una breve descripción y que marcará a qué funcionalidades puede acceder éste. Una cuenta sólo puede ofrecer las funcionalidades establecidas para un rol determinado.

La aplicación permitirá la comunicación inter-usuario a través de un sistema de **notas**. Todo usuario de ésta (que no del sistema) podrá redactar notas y dirigirlas al conjunto de usuarios que estime oportuno.



Una nota está compuesta por la fecha de redacción, el DNI del autor, el tema y el texto que la compone. Además deberá diseñarse una estructura que permita reflejar la lectura y escritura de notas, ofreciendo las funcionalidades descritas en el modelo de casos de uso. Un usuario puede redactar o no notas destinadas a uno o varios destinatarios.

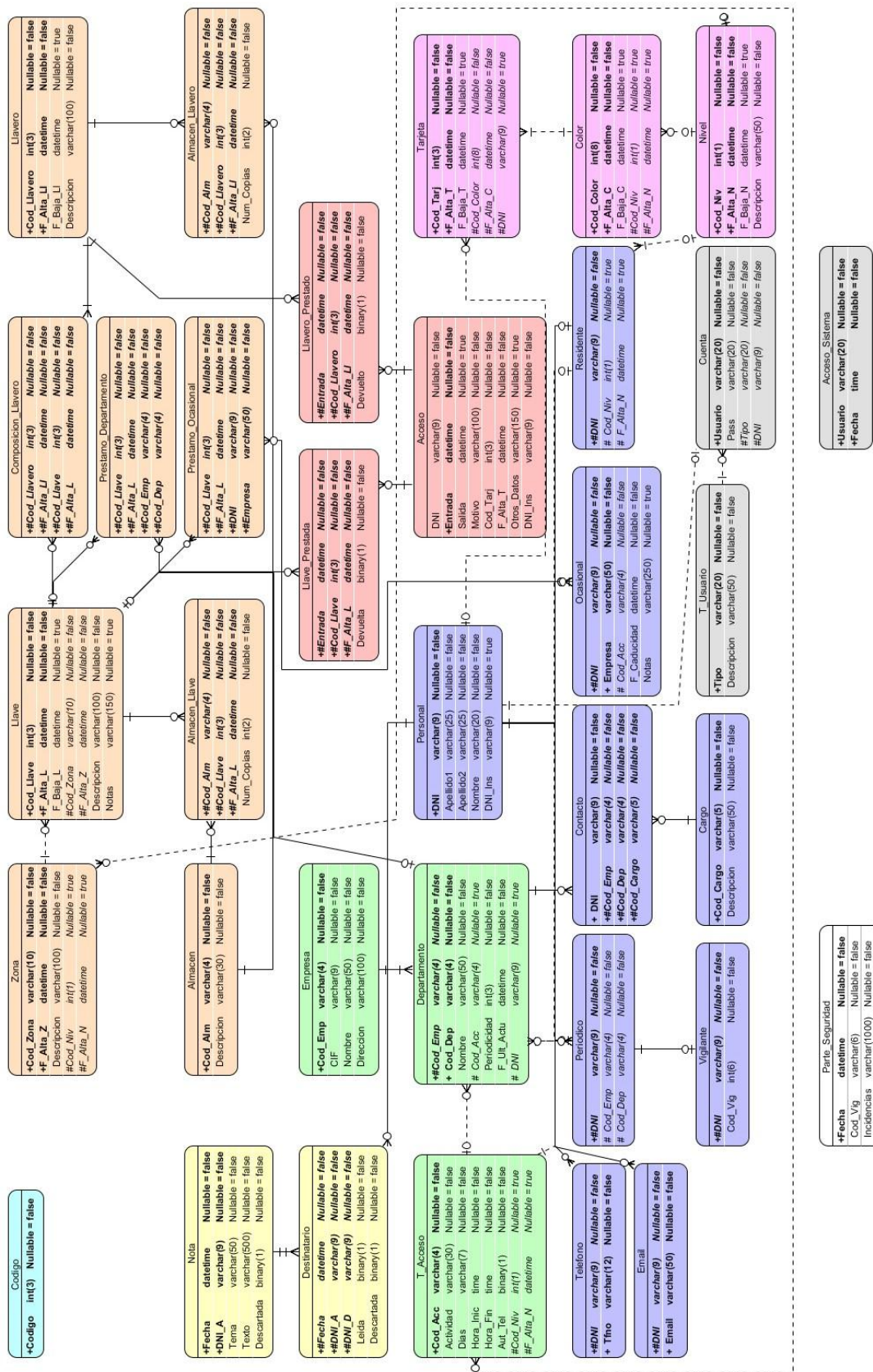


Empresas, departamentos, tipos de acceso, niveles, personal, accesos, usuarios y notas

3. Diseño del modelo de datos

3.1. Esquema relacional

La siguiente página muestra el esquema relacional ideado y codificado que da soporte a todo el sistema.



Esquema relacional de la base de datos

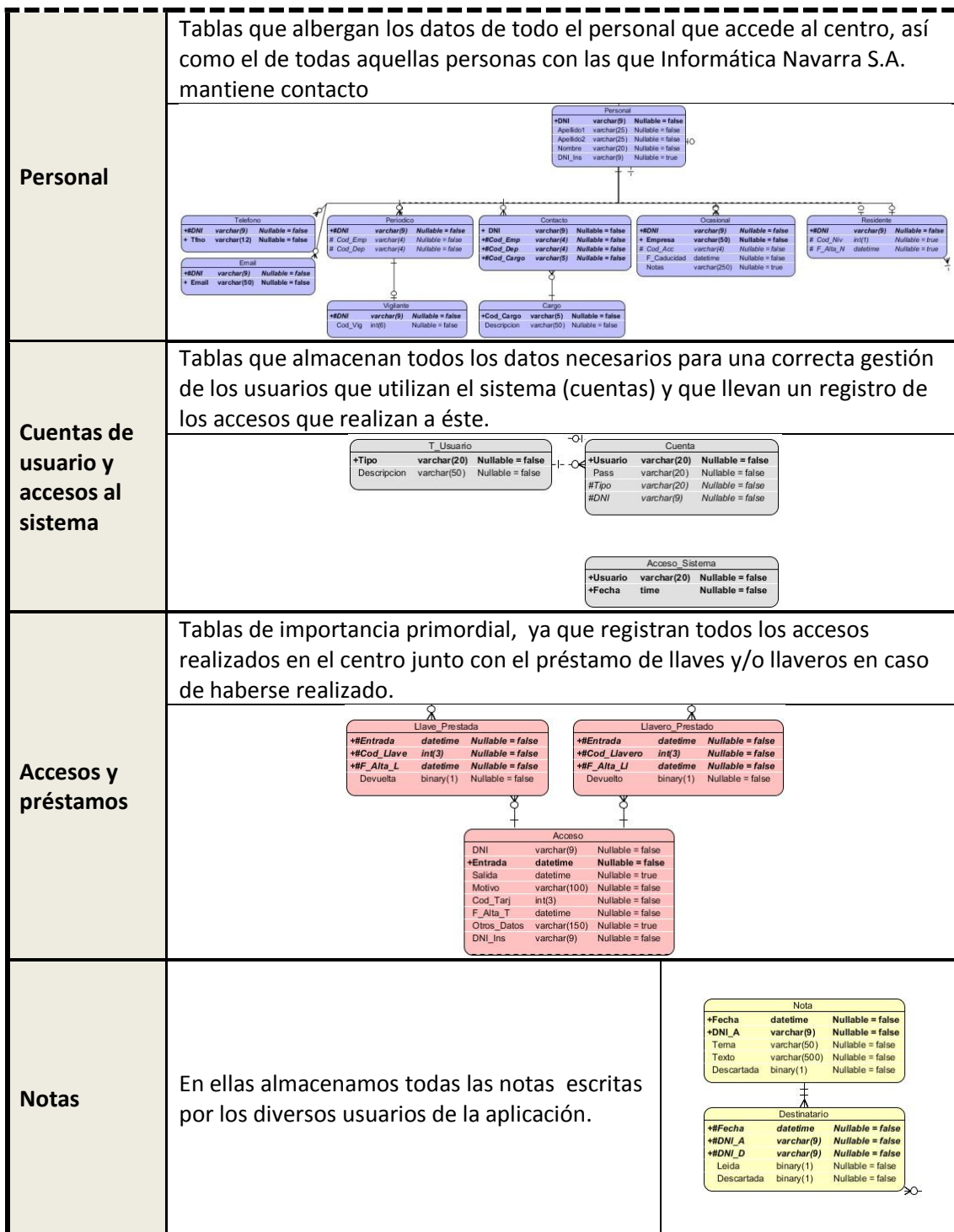


3.2. Breve descripción

Podríamos resumir la funcionalidad de cada grupo de tablas en la siguiente tabla:

TABLAS DEL ESQUEMA RELACIONAL		
ELEMENTOS	DESCRIPCIÓN	
Tarjetas, colores y niveles de acceso	<p>En ellas almacenamos toda la información referente a los diversos niveles de acceso que existen en el centro, junto con las tarjetas coloreadas que lo proporcionan.</p>	<pre> Tarjeta +Cod_Tarj int(3) Nullable = false +F_Alta_T datetime Nullable = false F_Baja_T datetime Nullable = true #Cod_Color int(8) Nullable = false #F_Alta_C datetime Nullable = false #DNI varchar(9) Nullable = true Color +Cod_Color int(8) Nullable = false +F_Alta_C datetime Nullable = false F_Baja_C datetime Nullable = true #Cod_Niv int(1) Nullable = true #F_Alta_N datetime Nullable = true Nivel +Cod_Niv int(1) Nullable = false +F_Alta_N datetime Nullable = false F_Baja_N datetime Nullable = true Descripcion varchar(50) Nullable = false </pre>
Zonas, llaves y llaveros	<p>En ellas registramos las zonas que constituyen la sede, así como todas las llaves y llaveros que serán usados en ésta y cómo se encuentran almacenados.</p>	<pre> Zona +Cod_Zona varchar(10) Nullable = false +F_Alta_Z datetime Nullable = false Descripcion varchar(100) Nullable = false #Cod_Niv int(1) Nullable = true #F_Alta_N datetime Nullable = true Llave +Cod_Llave int(3) Nullable = false +F_Alta_L datetime Nullable = true F_Baja_L datetime Nullable = false #Cod_Zona varchar(10) Nullable = false #F_Alta_Z datetime Nullable = false Descripcion varchar(100) Nullable = false Notas varchar(150) Nullable = true Llavero +Cod_Llavero int(3) Nullable = false +F_Alta_Ll datetime Nullable = false F_Baja_Ll datetime Nullable = true Descripcion varchar(100) Nullable = false Almacen +Cod_Alm varchar(4) Nullable = false Descripcion varchar(30) Nullable = false Prestamo +Cod_Llave int(3) Nullable = false +F_Alta_L datetime Nullable = false #Cod_Emp varchar(4) Nullable = false #Cod_Dep varchar(4) Nullable = false Almacen_Llave +Cod_Alm varchar(4) Nullable = false #Cod_Llave int(3) Nullable = false #F_Alta_L datetime Nullable = false Num_Copias int(2) Nullable = false Almacen_Llavero +Cod_Alm varchar(4) Nullable = false #Cod_Llavero int(3) Nullable = false #F_Alta_Ll datetime Nullable = false Num_Copias int(2) Nullable = false </pre>
Empresas, departamentos y tipos de acceso	<p>Tablas en las que almacenamos toda la información referente a las empresas que trabajan en el centro. Dicha información consta de las empresas en sí mismas, los departamentos que las forman y el tipo de acceso que poseen éstos.</p>	<pre> Empresa +Cod_Emp varchar(4) Nullable = false CIF varchar(9) Nullable = false Nombre varchar(50) Nullable = false Direccion varchar(100) Nullable = false T_Acceso +Cod_Acc varchar(4) Nullable = false Actividad varchar(30) Nullable = false Dias varchar(7) Nullable = false Hora_Inic time Nullable = false Hora_Fin time Nullable = false Aut_Tel binary(1) Nullable = false #Cod_Niv int(1) Nullable = true #F_Alta_N datetime Nullable = true Departamento +Cod_Emp varchar(4) Nullable = false +Cod_Dep varchar(4) Nullable = false Nombre varchar(50) Nullable = false #Cod_Acc varchar(4) Nullable = true Periodicidad int(3) Nullable = false F_Util_Actu datetime Nullable = false #DNI varchar(9) Nullable = true </pre>

Tablas que conforman la base de datos



Tablas que conforman la base de datos



Partes de seguridad	Tabla que almacena todos los partes de seguridad redactados por los vigilantes del centro.	<table border="1"><thead><tr><th colspan="3">Parte_Seguridad</th></tr></thead><tbody><tr><td>*Fecha</td><td>datetime</td><td>Nullable = false</td></tr><tr><td>Cod_Vig</td><td>varchar(6)</td><td>Nullable = false</td></tr><tr><td>Incidencias</td><td>varchar(1000)</td><td>Nullable = false</td></tr></tbody></table>	Parte_Seguridad			*Fecha	datetime	Nullable = false	Cod_Vig	varchar(6)	Nullable = false	Incidencias	varchar(1000)	Nullable = false
Parte_Seguridad														
*Fecha	datetime	Nullable = false												
Cod_Vig	varchar(6)	Nullable = false												
Incidencias	varchar(1000)	Nullable = false												
Códigos	Tabla auxiliar que será utilizada en futuras consultas.	<table border="1"><thead><tr><th colspan="3">Codigo</th></tr></thead><tbody><tr><td>*Codigo</td><td>int(3)</td><td>Nullable = false</td></tr></tbody></table>	Codigo			*Codigo	int(3)	Nullable = false						
Codigo														
*Codigo	int(3)	Nullable = false												

Tablas que conforman la base de datos

3.3. Disparadores

Se dispone de los disparadores mostrados en las siguientes páginas con el fin de mantener la coherencia de la información que alberga la base de datos:



TARJETAS, COLORES Y NIVELES DE ACCESO				
TABLA	NOMBRE	MOMENTO	EVENTO	ACCIONES A REALIZAR
Nivel	ins_nivel	Antes	Inserción	Previamente a la inserción tenemos que dar de baja todos aquellos niveles (cero o uno) que dispongan del mismo código de nivel que el que va a ser introducido.
	actu_nivel	Después	Actualiz.	Si la actualización de un nivel implica la baja de éste, haremos que dicha baja se propague a través de los colores y tarjetas que los representan.
<ul style="list-style-type: none"> • Actualizar Nivel asignando al nivel cuyo código sea el que nos disponemos a introducir su fecha de baja a la actual. • Actualizar Colores asignando a los colores y tarjetas que estén relacionados con el nivel dado de baja su fecha de baja a la actual. 				
Color	ins_col	Antes	Inserción	Antes de insertar un color debemos dar de baja a aquellos colores que existan en el sistema con el mismo código que el nuevo (cero o uno), así como las tarjetas con las que éstos estén relacionados.
	actu_col	Después	Actualiz.	Después de actualizar o eliminar un color tendremos que comprobar que todos los niveles siguen disponiendo de algún color. En caso de no ser así, los eliminaremos.
	borr_col	Después	Borrado	
<ul style="list-style-type: none"> • Actualizar Color asignando al color cuyo código sea el que nos disponemos a introducir su fecha de baja a la actual. • Actualizar Tarjeta asignando a las tarjetas coloreadas con el color dado de baja la misma fecha de baja que a éste. • Eliminar de Nivel todos los niveles que no dispongan de colores. 				
Tarjeta	ins_tarj	Antes	Inserción	Antes de insertar cualquier tarjeta debemos asegurar que no existirán dos con el mismo código. En este caso daremos de baja la existente antes de añadir la nueva.
	actu_tarj	Después	Actualiz.	Tras actualizar o eliminar una tarjeta debemos asegurar que todo color y todo nivel disponen de alguna tarjeta y algún color respectivamente.
	borr_tarj	Después	Borrado	
<ul style="list-style-type: none"> • Actualizar Tarjeta asignando a la tarjeta cuyo código sea el que nos disponemos a introducir su fecha de baja a la actual. • Eliminar de Color todos los colores que no dispongan de tarjetas. • Eliminar de Nivel todos los niveles que no dispongan de colores. 				

Disparadores relativos a las tablas de tarjetas, colores y niveles de acceso

accessManager · Esquema relacional de la base de datos · Página 13 de 17



ZONAS, LLAVES Y LLAVEROS				
TABLA	NOMBRE	MOMENTO	EVENTO	ACCIONES A REALIZAR
Zona	borr_zona	Después	Borrado	Si borramos una zona debemos eliminar del sistema todos aquellos llaveros que queden sin llaves y todos los almacenes que pudieran quedar vacíos.
<ul style="list-style-type: none"> • <i>Eliminar de Llavero todos los llaveros cuyos códigos no aparezcan referenciados en Composicion_Llavero.</i> • <i>Eliminar de Almacen todos los almacenes cuyos códigos no aparezcan referenciados en Almacen_Llave o Almacen_Llavero.</i> 				
Llave	ins_llave	Antes	Inserción	Al igual que nos sucede con las tarjetas, colores y niveles, antes de insertar cualquier llave debemos dar de baja (en caso de que exista) aquella que pueble la tabla Llaves y tenga el mismo código que la que nos disponemos a insertar.
<ul style="list-style-type: none"> • <i>Actualizar Llave asignando a la llave cuyo código sea el que nos disponemos a introducir su fecha de baja a la actual.</i> 				
Llave	borr_l	Después	Borrado	Se deberán realizar las mismas comprobaciones que las llevadas a cabo con el disparador <i>borr_zona</i> , ya que el borrado de una llave podría hacer que un llavero se quedase sin llaves o un almacén sin elementos que almacenar.
<ul style="list-style-type: none"> • <i>Eliminar de Llavero todos los llaveros cuyos códigos no aparezcan referenciados en Composicion_Llavero.</i> • <i>Eliminar de Almacen todos los almacenes cuyos códigos no aparezcan referenciados en Almacen_Llave o Almacen_Llavero.</i> 				
Almacen_Llave	borr_al	Después	Borrado	Si eliminamos una relación almacén-llave no estando la llave contenida en ningún almacén o llavero más, debemos eliminarla. Si además dejamos a un almacén vacío, procederá eliminarlo.
<ul style="list-style-type: none"> • <i>Eliminar de Llave todas las llaves cuyos códigos no aparezcan referenciados en Composicion_Llavero o Almacen_Llave.</i> • <i>Eliminar de Almacen todos los almacenes cuyos códigos no aparezcan referenciados en Almacen_Llave o Almacen_Llavero.</i> 				

Disparadores relativos a las tablas de zonas, llaves y llaveros



Llavero	ins_llavero	Antes	Inserción	Tal y como ocurría con las llaves, si nos disponemos a insertar un llavero cuyo código ya exista en la base de datos, deberemos dar de baja el llavero que lo posea para después pasar a realizar la inserción del nuevo.
<ul style="list-style-type: none"> • <i>Actualizar Llavero asignando al llavero cuyo código sea el que nos disponemos a introducir su fecha de baja a la actual.</i> 				
Llavero	borr_ll	Después	Borrado	Toda aquella llave que tras un borrado de llavero no corresponda a ningún otro llavero o tenga existencia propia deberá ser eliminada. Además realizaremos el mantenimiento de la tabla Almacen, borrando todos aquellos que no dispongan de elementos que albergar.
<ul style="list-style-type: none"> • <i>Eliminar de Llave todas las llaves cuyos códigos no aparezcan referenciados en Composicion_Llavero o Almacen_Llave.</i> • <i>Eliminar de Almacen todos los almacenes cuyos códigos no aparezcan referenciados en Almacen_Llave o Almacen_Llavero.</i> 				
Almacen_Llavero	borr_all	Después	Borrado	Si eliminamos una relación almacén-llavero no estando el llavero contenido en ningún almacén más, entonces debemos eliminarlo, así como las llaves no dispongan de existencia propia o conjunta a un llavero y aquellos almacenes que queden sin elementos que almacenar.
<ul style="list-style-type: none"> • <i>Eliminar de Llavero todos los llaveros cuyos códigos no aparezcan referenciados en Almacen_Llavero.</i> • <i>Eliminar de Llave todas las llaves cuyos códigos no aparezcan referenciados en Composicion_Llavero o Almacen_Llave.</i> • <i>Eliminar de Almacen todos los almacenes cuyos códigos no aparezcan referenciados en Almacen_Llave o Almacen_Llavero.</i> 				

Disparadores relativos a las tablas de zonas, llaves y llaveros



EMPRESAS, DEPARTAMENTOS Y TIPOS DE ACCESOS				
TABLA	NOMBRE	MOMENTO	EVENTO	ACCIONES A REALIZAR
Departamento	borr_dep	Después	Borrado	<p>Eliminar todos trabajadores periódicos y contactos que hayan quedado “huérfanos” de departamentos en los que trabajar.</p> <p>Si el departamento a borrar es el único que queda en la empresa, la eliminamos.</p> <p>Si existen cargos sin desempeñar, los borramos.</p>
<ul style="list-style-type: none"> • <i>Eliminar de Personal aquellas personas que no aparezcan referenciadas en ninguna de sus tablas especializadas.</i> • <i>Eliminar de Empresa todas las empresas cuyos códigos no aparezcan referenciados en Departamento.</i> • <i>Eliminar de Cargo aquellos cargos que no aparezcan referenciados en Contacto.</i> 				

Disparadores relativos a las tablas de empresas, departamentos y tipos de acceso

PERSONAL				
TABLA	NOMBRE	MOMENTO	EVENTO	ACCIONES A REALIZAR
Contacto	borr_contacto	Después	Borrado	<p>Borrar todos los contactos que no desempeñen su labora para ninguna empresa-departamento y todos los cargos que no estén relacionados con alguna persona.</p>
<ul style="list-style-type: none"> • <i>Eliminar de Personal la persona recientemente eliminada si no consta como contacto en otra empresa-departamento.</i> • <i>Eliminar de Cargo aquellos cargos que no aparezcan referenciados en Contacto.</i> 				

Disparadores relativos a las tablas de personal

CUENTAS DE USUARIO Y ACCESOS AL SISTEMA				
TABLA	NOMBRE	MOMENTO	EVENTO	ACCIONES A REALIZAR
Cuenta	borr_cuenta	Después	Borrado	<p>Borrar todos los tipos de usuarios que no dispongan de ninguna cuenta que los representen.</p>
<ul style="list-style-type: none"> • <i>Eliminar de T_Usuario aquellos tipos de usuario que no aparezcan referenciados en Cuenta.</i> 				

Disparadores relativos a las tablas de cuentas de usuario y accesos al sistema



ACCESOS Y PRÉSTAMOS				
TABLA	NOMBRE	MOMENTO	EVENTO	ACCIONES A REALIZAR
Acceso	ins_acc	Después	Inserción	Registramos que la tarjeta ofrecida a la persona entrante al centro la posee en préstamo.
<ul style="list-style-type: none">• <i>Actualizar Tarjeta, asignando a la tarjeta prestada el DNI de su prestatario, es decir, que Tarjeta.DNI = NEW.DNI donde Tarjeta.Cod_Tarj = NEW.Cod_Tarj.</i>				

Disparadores relativos a las tablas de accesos y préstamos

NOTAS				
TABLA	NOMBRE	MOMENTO	EVENTO	ACCIONES A REALIZAR
Nota	actu_notas	Después	Actualiz.	Si el redactor de una nota la descarta habiendo sido ésta descartada anteriormente por todos sus destinatarios, la borraremos ya que carece de sentido seguir almacenándola si nadie puede acceder a ella.
<ul style="list-style-type: none">• <i>Eliminar de Nota aquellas notas que hayan sido descartadas por su autor y todos sus receptores.</i>				
Destinatario	actu_dest	Después	Actualiz.	Debemos eliminar todas aquellas notas que hayan sido descartadas por su autor y todos sus destinatarios. Si en el caso anterior tratábamos que el autor era el último en descartarla, en éste manejaremos el caso en que el destinatario sea el último en descartarla.
<ul style="list-style-type: none">• <i>Eliminar de Nota aquellas notas que hayan sido descartadas por su autor y todos sus receptores.</i>				

Disparadores relativos a las tablas de notas

Pamplona, a 16 de abril de 2010.

Héctor Napal Basterrica
Responsable del proyecto



accessManager
Desarrollo de un sistema de control de accesos a un centro

Arquitectura del sistema

Fecha de entrega: 16 de abril de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Estructura del documento

2. Arquitectura del sistema

- 2.1. Vista de distribución (hardware)
- 2.2. Vista lógica (diseño)
- 2.3. Vista de realización (código fuente)
- 2.4. Vista de procesos (obj, dll, lib, com,...)
- 2.5. Vista de los casos de uso (escenarios)



1. Introducción

1.1. Propósito

A lo largo de las siguientes páginas realizaremos una breve explicación de la arquitectura que dispone el sistema desarrollado, exponiendo varias vistas arquitectónicas con el fin de que el futuro lector de este documento sea capaz de entender el sistema a la perfección y desde todos los puntos de vista que requiera.

1.2. Audiencia

Este artefacto va dirigido a cualquier persona que necesite entender la arquitectura que compone el sistema, bien para su estudio o para su futuro mantenimiento. El mismo será entregado con el resto de la documentación en la segunda iteración de la fase de Transición.

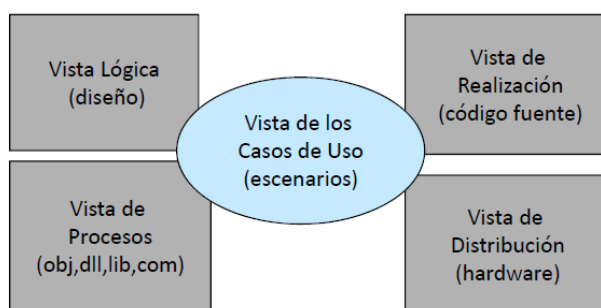
1.3. Estructura del documento

El escrito que usted se encuentra leyendo consta de una única sección dividida en cinco apartados, en los que se explicará la estructura del sistema a través del modelo de '4+1 vistas' ideado por *Philippe Kruchten*.



2. Arquitectura del sistema

La arquitectura de un sistema es el resultado de ensamblar un conjunto de componentes para satisfacer las necesidades que originan el desarrollo del producto software. Resulta muy complejo capturar la arquitectura en un solo modelo, por lo que suelen emplearse múltiples vistas (perspectivas) para ello. Entre las cualidades que atesora el proceso de desarrollo *RUP* se encuentra la especificación de la arquitectura mediante el famoso modelo de '4+1 vistas' de *Philippe Kruchten*:



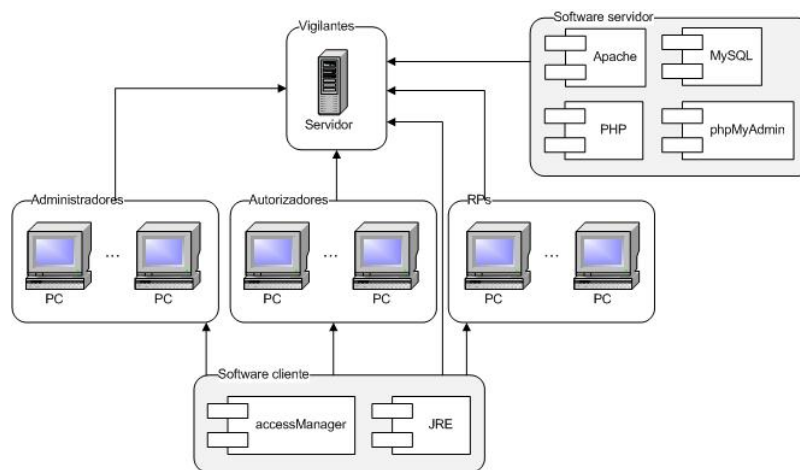
Modelo de 4+1 vistas

A continuación hacemos una pequeña aproximación a de cada una de ellas y una muestra de las que creemos más importantes para la comprensión de nuestro sistema.

2.1. Vista de distribución (hardware)

Describe los diferentes recursos hardware y la implementación del software en éstos. Constituye un mapeo entre las arquitecturas hardware y software.

En la segunda iteración de la fase de Transición ya establecimos el despliegue que realizaremos en la sede de Informática Navarra S.A. para conseguir el sistema deseado. Conviene recordar que crearemos una arquitectura cliente-servidor con la siguiente distribución:



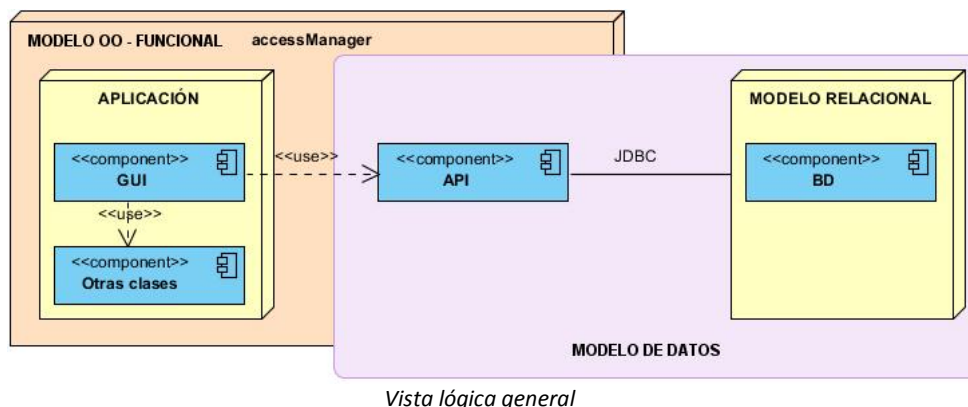
Vista de distribución



2.2. Vista lógica (diseño)

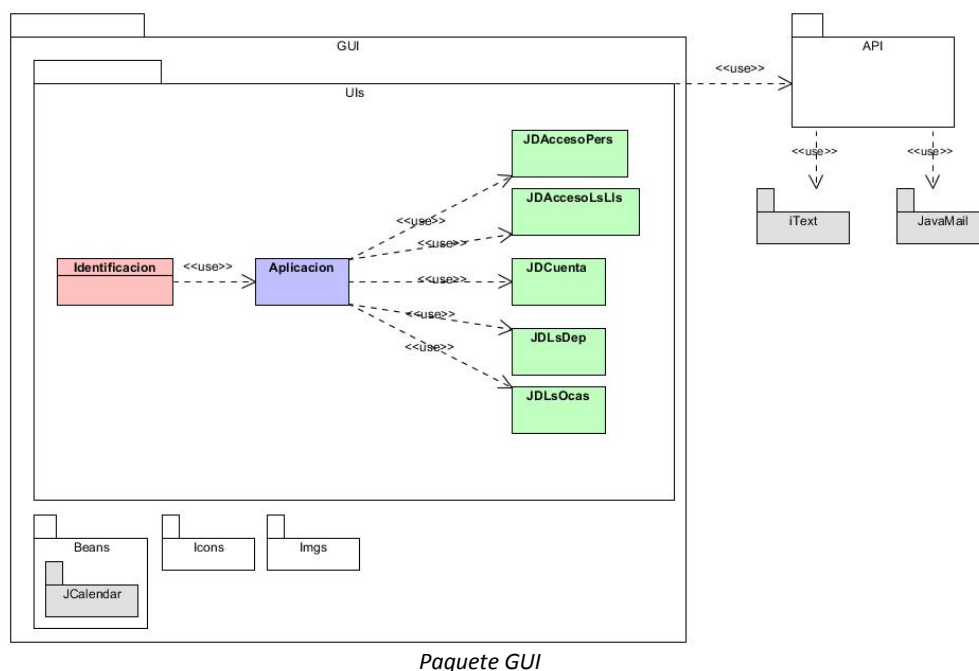
Describe los aspectos estáticos y dinámicos de un sistema en términos de clases y objetos. Recoge el modelo de objetos, clases, E/R,...

Podemos describir el sistema de la siguiente forma:



El modelo relacional fue descrito en el anterior artefacto y la API será descrita en el posterior, con que si el lector desea conocerlos en profundidad le aconsejamos que los revise.

De cara a la aplicación gráfica que coordinará todos los componentes software disponemos de la siguiente estructura...





... cuyas clases vienen especificadas de la siguiente forma:

Identificacion
+Identificacion() : Identificacion
+initComponents()
+main(args : String [])

Estructura de Identificacion.java

JDAccesoPers
-pulsarExit : boolean
-UGUI : UtilidadesGUI
+JDAccesoPers(parent : Frame, modal : boolean) : JDAccesoPers
-initComponents()
-iniciarDatos()
+getDNI() : String
+getPulsarExit() : boolean
+main(args : String [])

Estructura de JDAccesoPers.java

JDAccesoLsLIs
-DNI : String
-Entrada : String
-pulsarExit : boolean
-UGUI : UtilidadesGUI
-UBD : UtilidadesBD
+JDAccesoLsLIs(parent : Frame, modal : boolean, DNI : String, Entrada : String) : JDAccesoLsLIs
-initComponents()
-iniciarDatos()
+getLsLIs() : HashMap<String, GLL_Dev>
+getPulsarExit() : boolean
+main(args : String [])

Estructura de JDAccesoLsLIs.java

JDCuenta
-Usuario : String
+JDCuenta(parent : Frame, modal : boolean, Usuario : String) : JDCuenta
-initComponents()
-iniciarDatos()
+main(args : String [])

Estructura de JDCuenta.java

JDLsDep
-Cod_Emp : String
-Cod_Dep : String
-UGUI : UtilidadesGUI
-UBD : UtilidadesBD
+JDLsDep(parent : Frame, modal : boolean, Cod_Emp : String, Cod_Dep : String) : JDLsDep
-initComponents()
-iniciarDatos()
+main(args : String [])

Estructura de JDLsDep.java



JDLsOcas
-DNI : String -Empresa : String -UGUI : UtilidadesGUI -UBD : UtilidadesBD
+JDLsOcas(parent : Frame, modal : boolean, DNI : String, Empresa : String) : JDLsOcas -initComponents() -iniciarDatos() +main(args : String [])

Estructura de JDLsOcas.java

Aplicacion
-Usuario : String -Tipo_Usuario : String -DNI_Usuario : String -DNI_Usu : String -HM_GLLDs : HashMap<String, GLL_Dev> -DNI_Personal : String -DNI_Prest : String -Tipo_Usu : String -Fecha_Parte : String -Cartero : Cartero -UGUI : UtilidadesGUI -UBD : UtilidadesBD -UCAL : UtilidadesCalendar
+Aplicacion(Usuario : String, Tipo_Usuario : String, DNI_Usuario : String) : Aplicacion -initComponets() -iniciarDatos() -pintar() -visualizacion() -iniciarAccesos() -comprobarFechasHorasAccesos() -mostrarAcceso() -mostrarDatosPersonalesAcceso(acc : Acceso) : int -mostrarTarjeta(acc : Acceso) -mostrarPrestamos(acc : Acceso) -mostrarAutorizTfno() -borrarFormAcceso() -borrarDatosPersonales() -borrarTarjeta() -borrarPrestamos() -borrarAutorizTfno() -iniciarEmpresasPersonal() -refrescarTablasEmpresasPersonal(tablas : String) -refrescarJTEmpresas(CIF : String, Nom_Emp : String, Nom_Dep : String, DNI : String) -refrescarJDepartamentos(Nom_Emp : String, Nom_Dep : String, Actividad : String, DNI : String) -refrescarJTEmpleados(Nom_Emp : String, Nom_Dep : String, Filtro : String) -refrescarJTContactos(Nom_Emp : String, Nom_Dep : String, Persona : String) -refrescarJTTFns(DNI : String) -refrescarJTEmails(DNI : String) -refrescarJTLsLisPers() -refrescarJTLsLisPers1() -refrescarJTCargos(DNI : String) -mostrarDP(DNI : String) -mostrarForm() -mostrarJPanelsPersonal(Paneles : String) -borrarDatosEmp() -borrarDatosDep() -borrarFormDP() -refrescarCBActivos() -refrescarCBNivs(jcb : JComboBox) -refrescarCBEmpresas()

Estructura de Aplicacion.java - Parte 1



```
-refrescarCBOcasional()
-refrescarCBContactos()
-refrescarBotonesEmpPers()
-iniciarTACs()
-refrescarTarjetas()
-refrescarColores()
-refrescarNiveles()
-refrescarTAccesos()
-refrescarTCargos()
-activDesactivBotonesTACs()
-iniciarZonas()
-refrescarZonas()
-deshabilitarBotonesZonas()
-refrescarAlmacenes()
-vaciarTablasZonas()
-vaciarTablasPrestables()
-activarTablasPrestablesActuales()
-activarTablasPrestablesPosibles()
-insertarPrestables(Llave : Llave)
-iniciarPrestamos()
-refrescarTarjsPrest()
-refrescarLsLlsPrest()
-mostrarDatosPrestatario(Prestatario : Personal)
-iniciarPartes()
-refrescarCBVigs()
-refrescarTPartes()
-comprobarFechasHorasPartes()
-iniciarCuentas()
-refrescarCBDNls()
-refrescarCBTUsuarios()
-refrescarCuentas()
-refrescarTCuentas()
-refrescarTUsuarios()
-refrescarTAccsSist()
-comprobarFechasHorasAccsSistema()
-iniciarNotas()
-refrescarContactos()
-refrescarBandejas()
-refrescarBandejaEntrada()
-refrescarBandejaSalida()
-mostrarNota(nota : Nota)
-borrarFormNotaLect()
-borrarFormNotaEscr()
-enviarNota()
+mantenerAcceso()
-enviarEmails()
-eliminarDatosAntiguos(Dias : int)
+main(args : String [])
```

Estructura de Aplicacion.java - Parte 2

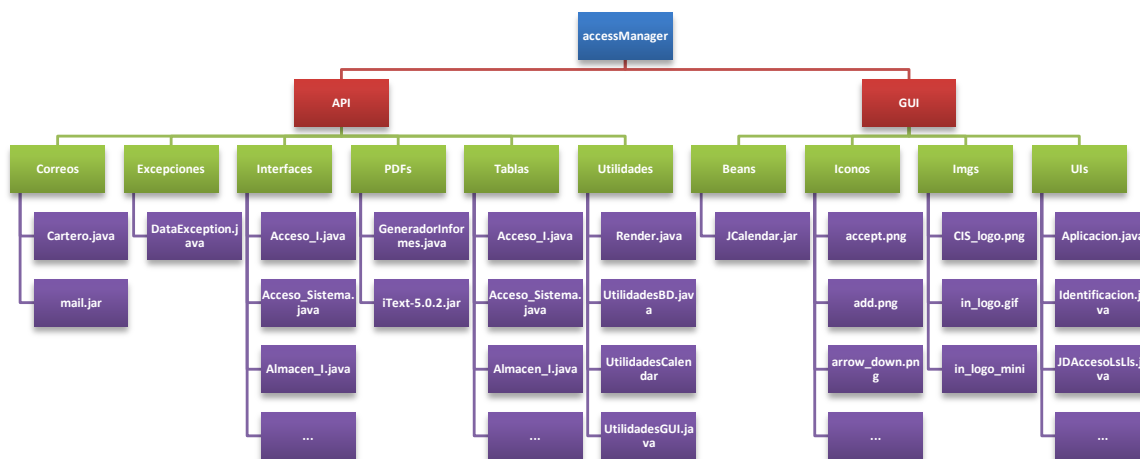
Estimamos innecesario profundizar en el estudio de cada clase, pues todos y cada uno de los métodos que implementan han sido documentados cuidadosamente en el código fuente, siendo éste de gran utilidad y fácil consulta en caso de que el lector desee conocer la funcionalidad que proporciona cualquiera de ellos. Para una breve revisión todo lo expuesto será más que suficiente.



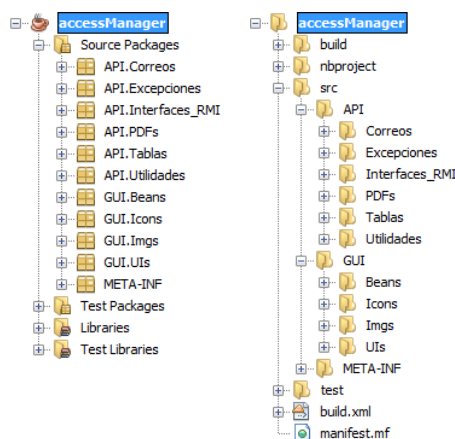
2.3. Vista de realización (código fuente)

Se preocupa de la organización de los componentes en el entorno de desarrollo, mostrando el reparto de clases en componentes y éstos en subsistemas.

En nuestro desarrollo hemos optado por implementar cada clase del diagrama de clases (tanto de la aplicación como de la API) en un fichero de código fuente, obteniendo la siguiente estructura jerárquica con los principales artefactos:



Estructura jerárquica de artefactos (código fuente, librerías e imágenes)



Estructuras jerárquicas en el entorno de desarrollo NetBeans



2.4. Vista de procesos (obj, dll, lib, com,...)

Representa la descomposición en flujos de ejecución, la sincronización entre los flujos y la asignación de objetos y clases en dichos flujos.

accessManager será una aplicación Java que correrá en cada uno de los ordenadores en los que sea ejecutada, creando la máquina virtual de Java tantas instancias de ésta como veces sea lanzada. Debido al tipo de sistema creado resulta innecesario realizar un estudio desde esta perspectiva.

2.5. Vista de los casos de uso (escenarios)

Unifican las cuatro vistas precedentes, recogiendo toda la funcionalidad que deberá proporcionar el producto software final al usuario. Para más información aconsejamos al lector que acuda al documento *Especificación de los casos de uso* enmarcado en este mismo capítulo.

Pamplona, a 16 de abril de 2010.

Héctor Napal Basterrica
Responsable del proyecto



accessManager
Desarrollo de un sistema de control de accesos a un centro

Descripción de la API final

Fecha de entrega: 16 de abril de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Estructura del documento

2. Composición de la API final

- 2.1. Nuevas clases
- 2.2. Clases iniciales refinadas



1. Introducción

1.1. Propósito

El propósito que perseguimos con este documento es el de describir el estado final en el que queda la API Java iniciada al comienzo del proceso de desarrollo y refinada a lo largo de éste, que será manejada por la clase principal para conseguir proporcionar toda la funcionalidad para la que ha sido ideada.

1.2. Audiencia

Descripción de la API final se encuentra encuadrado entre los principales artefactos surgentes del producto software desarrollado y supondrá de vital importancia de cara a las futuras actividades de mantenimiento necesarias para que el sistema concebido satisfaga los requisitos cambiantes del cliente.

1.3. Estructura del documento

La estructura del documento es breve y sencilla, disponiendo de un único capítulo en el que expondremos en primer lugar el diagrama de clases que representa la estructura de la API final, a continuación realizaremos un breve estudio de las nuevas clases creadas durante el desarrollo y que no existían en la API inicial y finalizaremos con la muestra de la estructura final de cada una de las clases iniciales que han sido refinadas.



2. Composición de la API final

Si tuviésemos que destacar el componente software que más grado de evolución ha sufrido a lo largo de todo el proceso de desarrollo, ese sería la API. En la segunda iteración de la fase de Construcción pudimos observar cómo evolucionar del modelo relacional al orientado a objetos, creando una API que evolucionaría con el tiempo. Una vez que hemos completado el desarrollo de toda la aplicación y disponemos de los artefactos finales, podemos ofrecer una visión mucho más detallada de la API resultante de añadir a la inicial todos los métodos que hemos especificado en cada iteración. En la siguiente página exponemos el diagrama global de clases y a continuación un breve estudio de las nuevas y un repaso del resto.



Para simplificar el esquema y facilitar su entendimiento, se ha omitido:
 UtilidadesBD:
 Usado por: Acceso_Sistema, Almacen, Color, Cuenta, Edificio, Empresa, Gen_LLI, Nivel, Parte_Seguridad, Personal, T_Acceso, T_Usuario, Tarjeta
 UtilidadesCalendar:
 Usado por: Almacen, Edificio, Empresa, Residente
 UtilidadesBD:
 Descendientes: Acceso_Sistema, Departamento, Nota, Parte_Seguridad, Zona
 Usado por: Permiso
 DosFechas:
 Descendientes: Acceso, Gen_LLI, Nivel, T_Acceso, Tarjeta
 Nota: Cada clase del diagrama implementa su correspondiente interfaz, la cual hereda de Remote.

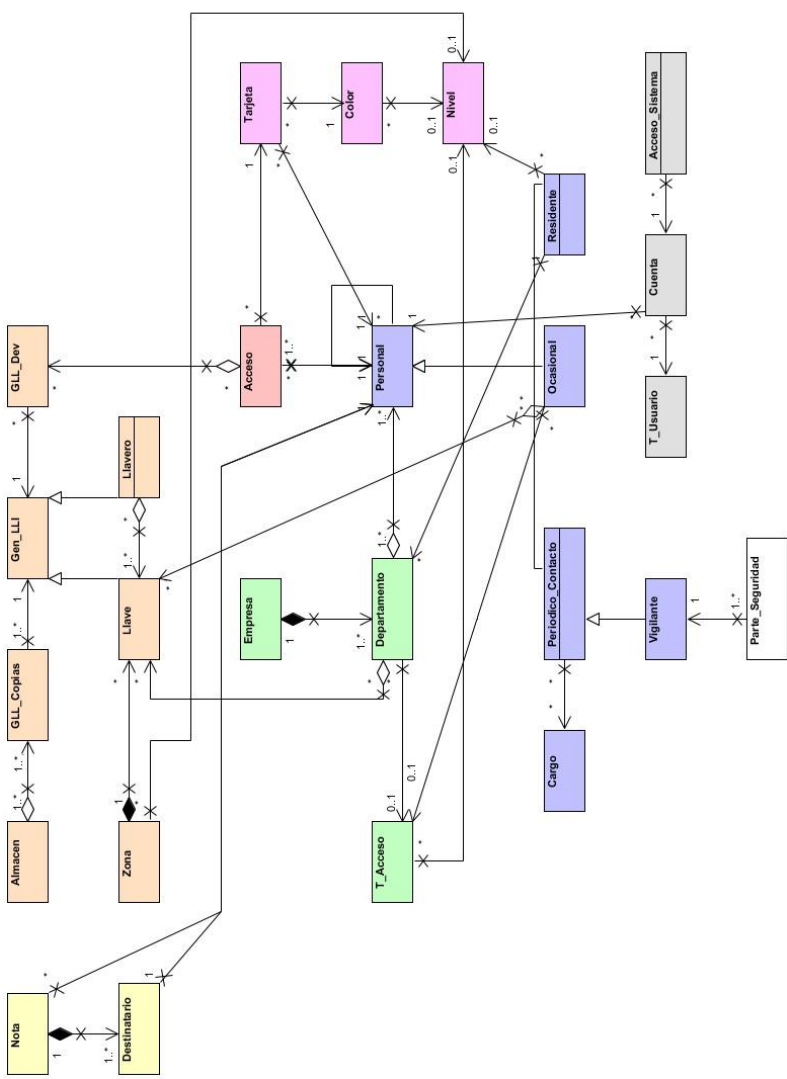
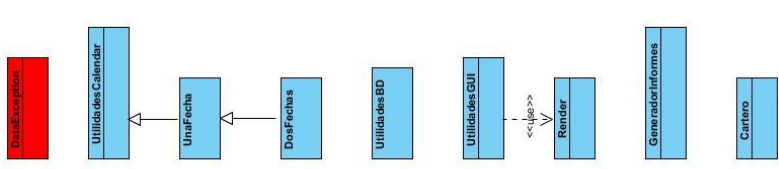


Diagrama de clases de la API final



2.1. Nuevas clases

Debemos mencionar cuatro clases nuevas que surgen a lo largo del diseño de accessManager: *UtilidadesGUI*, *Render*, *GeneradorInformes* y *Cartero*:

UTILIDADESGUI	
CAMPO	DESCRIPCIÓN
Descrip.	<p>Como su nombre indica, se trata de una clase que nos facilitará programar la interfaz gráfica ofreciendo funcionalidades relacionados con ésta, como pueden ser el llenado y pintado de JTables y JComboBoxes:</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p style="text-align: center;">UtilidadesGUI</p> <pre> +rsToTableModel(rs : ResultSet) : DefaultTableModel +rellenarTabla(tabla : JTable, rs : ResultSet) +pintarTabla(tabla : JTable, al : ArrayList) +rellenarTablaTelefonos(tabla : JTable, al : ArrayList<String>) +rellenarTablaPrestamos(tabla : JTable, hm : HashMap<String, GLL_Dev>) +rellenarTablaEmpresas(tabla : JTable, rs : ResultSet, hm : HashMap<Integer, Empresa>) +rellenarTablaDepartamentos(tabla : JTable, hm : HashMap<Integer, Empresa>, Emp_i : int) +rellenarTablaLs(tabla : JTable, hm : HashMap<String, Llave>) +rellenarJCB(JCB : JComboBox, rs : ResultSet) </pre> </div>
Atribs.	<p>---</p>
Méts.	<p><i>rsToTableModel</i> <i>rellenarTabla</i> <i>pintarTabla</i> <i>rellenarTablaTelefonos</i> <i>rellenarTablaPrestamos</i> <i>rellenarTablaEmpresas</i> <i>rellenarTablaDepartamentos</i> <i>rellenarTablaLs</i> <i>rellenarJCB</i></p> <p>Crear un DefaultTableModel a partir de un ResultSet. Rellenar una JTable con un ResultSet. Pintar una JTable según un ArrayList de colores. Rellenar la JTable de teléfonos con un ArrayList. Rellenar la JTable de préstamos con un HashMap. Rellenar la JTable de empresas con las empresas... Rellenar la JTable de departamentos... Rellenar la JTable de llaves con un HashMap de llaves. Rellenar un JComboBox con un ResultSet.</p>

Estructura de UtilidadesGUI.java



RENDER						
CAMPO	DESCRIPCIÓN					
Descrip.	Clase empleada por <i>UtilidadesGUI</i> y que se encarga de pintar las celdas de cualquier <i>JTable</i> a partir del <i>ArrayList</i> de colores que recibe en su construcción: <table border="1" style="margin: 10px auto; width: 80%;"> <thead> <tr> <th>Render</th> </tr> </thead> <tbody> <tr> <td>-colores : <i>ArrayList</i></td> </tr> <tr> <td>+Render(<i>Colores</i> : <i>ArrayList</i>) : <i>Render</i></td> </tr> <tr> <td>+getTableCellRendererComponent(<i>table</i> : <i>JTable</i>, <i>value</i> : <i>Object</i>, <i>isSelected</i> : <i>boolean</i>, <i>hasFocus</i> : <i>boolean</i>, <i>row</i> : <i>int</i>, <i>column</i> : <i>int</i>)</td> </tr> </tbody> </table>		Render	-colores : <i>ArrayList</i>	+Render(<i>Colores</i> : <i>ArrayList</i>) : <i>Render</i>	+getTableCellRendererComponent(<i>table</i> : <i>JTable</i> , <i>value</i> : <i>Object</i> , <i>isSelected</i> : <i>boolean</i> , <i>hasFocus</i> : <i>boolean</i> , <i>row</i> : <i>int</i> , <i>column</i> : <i>int</i>)
Render						
-colores : <i>ArrayList</i>						
+Render(<i>Colores</i> : <i>ArrayList</i>) : <i>Render</i>						
+getTableCellRendererComponent(<i>table</i> : <i>JTable</i> , <i>value</i> : <i>Object</i> , <i>isSelected</i> : <i>boolean</i> , <i>hasFocus</i> : <i>boolean</i> , <i>row</i> : <i>int</i> , <i>column</i> : <i>int</i>)						
Atribs.	<i>colores</i>	<i>ArrayList</i> con los colores que se irán empleando para pintar las celdas de la <i>JTable</i> recogida.				
Méts.	<i>Render</i> <i>getTableCellRendererComponent</i>	Constructor. Obtener la celda coloreada según ' <i>colores</i> '.				

Estructura de Render.java

GENERADORINFORME									
CAMPO	DESCRIPCIÓN								
Descrip.	Concebida en la iteración 'C3 - Accesos' y ampliada en 'C4 - Empresas y personal' y 'C9 - Partes de seguridad', <i>GeneradorInforme</i> es la clase encargada en el sistema de generar reportes en formato PDF a partir de las <i>Strings</i> y <i>JTables</i> que recibe en sus métodos: <table border="1" style="margin: 10px auto; width: 80%;"> <thead> <tr> <th>GeneradorInformes</th> </tr> </thead> <tbody> <tr> <td>-Ruta : <i>String</i></td> </tr> <tr> <td>+GeneradorInformes(<i>Ruta</i> : <i>String</i>)</td> </tr> <tr> <td>+generarInforme(<i>Titulo</i> : <i>String</i>, <i>Texto</i> : <i>String</i>, <i>Tabla</i> : <i>JTable</i>)</td> </tr> <tr> <td>+generarInformeAvanzado(<i>Titulo</i> : <i>String</i>, <i>Explicacion</i> : <i>String</i>, <i>Subtitulo1</i> : <i>String</i>, <i>Subtitulo2</i> : <i>String</i>, <i>Subtitulo3</i> : <i>String</i>)</td> </tr> <tr> <td>+JTable2PdfPTable(<i>Tabla</i> : <i>JTable</i>) : <i>PdfPTable</i></td> </tr> <tr> <td>+generarParte(<i>Panel</i> : <i>JPanel</i>)</td> </tr> </tbody> </table>		GeneradorInformes	-Ruta : <i>String</i>	+GeneradorInformes(<i>Ruta</i> : <i>String</i>)	+generarInforme(<i>Titulo</i> : <i>String</i> , <i>Texto</i> : <i>String</i> , <i>Tabla</i> : <i>JTable</i>)	+generarInformeAvanzado(<i>Titulo</i> : <i>String</i> , <i>Explicacion</i> : <i>String</i> , <i>Subtitulo1</i> : <i>String</i> , <i>Subtitulo2</i> : <i>String</i> , <i>Subtitulo3</i> : <i>String</i>)	+JTable2PdfPTable(<i>Tabla</i> : <i>JTable</i>) : <i>PdfPTable</i>	+generarParte(<i>Panel</i> : <i>JPanel</i>)
GeneradorInformes									
-Ruta : <i>String</i>									
+GeneradorInformes(<i>Ruta</i> : <i>String</i>)									
+generarInforme(<i>Titulo</i> : <i>String</i> , <i>Texto</i> : <i>String</i> , <i>Tabla</i> : <i>JTable</i>)									
+generarInformeAvanzado(<i>Titulo</i> : <i>String</i> , <i>Explicacion</i> : <i>String</i> , <i>Subtitulo1</i> : <i>String</i> , <i>Subtitulo2</i> : <i>String</i> , <i>Subtitulo3</i> : <i>String</i>)									
+JTable2PdfPTable(<i>Tabla</i> : <i>JTable</i>) : <i>PdfPTable</i>									
+generarParte(<i>Panel</i> : <i>JPanel</i>)									
Atribs.	<i>Ruta</i>	Ruta en la que depositar el reporte generado.							
Méts.	<i>GeneradorInformes</i> <i>generarInforme</i> <i>generarInformeAvanzado</i> <i>JTable2PdfPTable</i> <i>generarParte</i>	Constructor. Generar reporte PDF dados un título, texto y tabla. Generar reporte PDF dados título,... y tres tablas. Pasar <i>JTable</i> a <i>PdfPTable</i> (propia de <i>iText</i>). Generar reporte PDF dado un <i>JPanel</i> .							

Estructura de GeneradorInformes.java



CARTERO		
CAMPO	DESCRIPCIÓN	
Descrip.	<p>Clase encargada de realizar el envío de E-Mails ayudándose de la base de datos y la funcionalidad ofrecida por el paquete <i>JavaMail</i>:</p> <pre> class Cartero -Propiedades : Properties -Usuario : String -Pass : String +Cartero(DirSMTP : String, TLS : String, Puerto : String, DirEmail : String, Autent : String, Usuario : String, Pass : String) : Cartero +getPropiedades() : Properties +setPropiedades(Propiedades : properties) +getUsuario() : String +setUsuario(Usuario : String) +getPass() : String +setPass() : String +enviarCorreo(Emisor : String, Receptores : ArrayList, Tema : String, Texto : String, Codificacion : String, Tipo : String) +enviarCorreo(Receptores : ArrayList, Tema : String, Texto : String) +enviarCorreosRenovacionPersonal(Dias : int) +enviarCorreosPendientes(DNI : String) +enviarCorreosPrestamosPendientes(Dias : int) </pre>	
Atribs.	<p><i>Propiedades</i> <i>Usuario</i> <i>Pass</i></p>	<p>Propiedades de la cuenta de correo electrónico. Nombre de usuario. Contraseña.</p>
Méts.	<p><i>Cartero</i> <i>getXXXX</i> <i>setXXXX</i> <i>enviarCorreo</i> <i>enviarCorreo</i> <i>enviarCorreosRenovacionPersonal</i> <i>enviarCorreosPendientes</i> <i>enviarCorreosPrestamosPendientes</i></p>	<p>Constructor. Obtener valores de los atributos. Establecer valores de los atributos. Enviar correo (tipo simple). Enviar correo (tipo avanzado). Enviar correos a los departamentos con personal... Enviar correo a una persona con sus préstamos... Enviar correo a todo el personal cuyos préstamos...</p>

Estructura de Cartero.java

2.2. Clases iniciales refinadas

Excluyendo las cuatro clases anteriormente expuestas, el resto de la estructura de la API no se ve variada en cuanto a su número e interrelación pero sí en su estructura (número de métodos que atesoran). Como nuestro propósito no es otro más que el de mostrar la composición final de la API y no su estudio pormenorizado, a lo largo de las siguientes páginas exponemos la estructura final de cada clase e instamos al lector a que acuda al código fuente si desea conocer la funcionalidad de algún método específico.



Tarjeta
-Cod_Tarj : int
+Tarjeta() : Tarjeta
+Tarjeta(Cod_Tarj : int, F_Alta_T : Calendar, Color : Color, Persona : Persona) : Tarjeta
+Tarjeta(Cod_Tarj : int, F_Alta_T : Calendar, F_Baja_T : Calendar, Color : Color, Persona : Persona) : Tarjeta
+Tarjeta(Cod_Tarj : int, F_Alta_T : String) : Tarjeta
+Tarjeta(Cod_Tarj : int) : Tarjeta
+getCodTarj() : int
+setCodTarj(Cod_Tarj : int)
+getColor() : Color
+setColor(Color : Color)
+getPersona() : Persona
+setPersona(Persona : Persona)
+toString() : String
+insertTarjetaBD() : int
+updateTarjetaBD(Tarjeta : Tarjeta) : int
+devTarjetaBD() : int
+darBajaTarjetaBD() : int
+removeTarjetaBD() : int
+obtenerTarjetas() : ResultSet
+obtenerColoresTarjs() : ArrayList
+obtenerColoresTarjsPrest() : ArrayList
+obtenerPrestamos() : ResultSet
+numTarjPordefecto() : int
+numTarjAcc(Color : int) : int
+eliminarTarjetas(Dias : int) : int

Estructura de Tarjeta.java

Color
-Cod_Color : int
+Color() : Color
+Color(Cod_Color : int, F_Alta_C : Calendar, Nivel : Nivel) : Color
+Color(Cod_Color : int, F_Alta_C : Calendar, F_Baja_C : Calendar, Nivel : Nivel) : Color
+Color(Cod_Color : int, F_Alta_C : String) : Color
+Color(Cod_Color : int) : Color
+getCodColor() : int
+setCodColor(Cod_Color : int)
+getNivel() : Nivel
+setNivel(Niv : Nivel)
+toString() : String
+insertColorBD() : int
+updateColorBD(Color : Color) : int
+darBajaColorBD() : int
+removeColorBD() : int
+obtenerCeldasColores() : ResultSet
+obtenerColores() : ArrayList
+colorPorDefecto() : int
+colorPorDefecto(DNI : String) : int
+eliminarColores(Dias : int) : int

Estructura de Color.java

Nivel
-Cod_Niv : int
-Descripcion : String
+Nivel() : Nivel
+Nivel(Cod_Niv : int, F_Alta_N : Calendar, Descripcion : String) : Nivel
+Nivel(Cod_Niv : int, F_Alta_N : Calendar, F_Baja_N : Calendar, Descripcion : String) : Nivel
+Nivel(Cod_Niv : int) : Nivel
+Nivel(Cod_Niv : int, F_Alta_N : String) : Nivel
+getCodNiv() : int
+setCodNiv(Cod_Niv : int)
+getDescripcion() : String
+setDescripcion(Descripcion : String)
+toString() : String
+insertNivelBD() : int
+updateNivelBD(Nivel : Nivel) : int
+darBajaNivelBD() : int
+removeNivelBD() : int
+obtenerNiveles() : ResultSet
+obtenerNivelesJCB() : ResultSet
+nivelMinimo() : int
+nivelPorDefecto() : int
+eliminarNiveles(Dias : int) : int

Estructura de Nivel.java



Zona
-Cod_Zona : String
-Descripcion : String
+Zona() : Zona
+Zona(Cod_Zona : String, F_Alta_Z : Calendar, Descripcion : String, Llaves : List) : Zona
+Zona(Cod_Zona : String, F_Alta_Z : Calendar, Descripcion : String, Nivel : Nivel, Llaves : List) : Zona
+Zona(Cod_Zona : String, F_Alta_Z : String) : Zona
+Zona(Cod_Zona : String) : Zona
+getCodZona() : string
+setCodZona(Cod_Zona : String)
+getDescripcion() : string
+setDescripcion(Descripcion : String) : string
+getNivel() : Nivel
+setNivel(Nivel : Nivel)
+getLlaves() : List<Llave>
+setLlaves(Llaves : List<Llave>)
+addLlave(Llave : Llave)
+updateLlave(Llave_old : Llave, Llave_new : Llave)
+removeLlave(Llave : Llave)
+getLlave(Key_Llave : String) : Llave
+toString() : String
+insertLlaveBD(Llave : Llave) : int
+updateLlaveBD(Llave_old : Llave, Llave_new : Llave) : int
+darBajaLlaveBD() : int
+removeLlaveBD(Llave : Llave) : int
+insertLlavesBD() : int
+updateLlavesBD() : int
+removeLlavesBD() : int
+getLlavesBD() : HashMap<String, Llave>
+insertZonaBD() : int
+updateZonaBD(Zona : Zona) : int
+removeZonaBD() : int
+obtenerZonas() : ResultSet
+obtenerLlaves() : ResultSet

Estructura de Zona.java

Almacen
-Cod_Alm : String
-Descripcion : String
+Almacen() : Almacen
+Almacen(Cod_Alm : String, Descripcion : String, Llaves_Llaveros : HashMap<String, GLL_Copias>) : Almacen
+Almacen(Descripcion : String) : Almacen
+getCodAlm() : String
+setCodAlm(Cod_Alm : String)
+getDescripcion() : String
+setDescripcion(Descripcion : String)
+getGLLCs() : HashMap<String, GLL_Copias>
+setGLLCs(GLLCs : HashMap<String, GLL_Copias>)
+addGLLC(GLLC : GLL_Copias)
+subsGLLC(GLLC : GLL_Copias)
+updateGLLC(GLLC_old : GLL_Copias, GLLC_new : GLL_Copias)
+removeGLLC(GLLC : GLL_Copias)
+getGLLC(Key_GLLC) : GLL_Copias
+toString() : String
+insertGLLCBD(GLLC : GLL_Copias) : int
+updateGLLCBD(GLLC_old : GLL_Copias, GLLC_new : GLL_Copias) : int
+removeGLLCBD(GLLC : GLL_Copias) : int
+getGLLCBD(GLLC_Key : String) : GLL_Copias
+insertGLLCsBD() : int
+insertGLLCsBD(GLLCs : HashMap<String, GLL_Copias>) : int
+updateGLLCsBD(GLLCs : HashMap<String, GLL_Copias>) : int
+removeGLLCsBD() : int
+getGLLCsBD() : HashMap<String, GLL_Copias>
+insertAlmacenBD() : int
+updateAlmacenBD(Almacen : Almacen) : int
+removeAlmacenBD() : int
+removeAlmacenBD(Almacen : Almacen) : int
+obtenerAlmacenes() : ResultSet
+obtenerLlaves(Filtro : String) : ResultSet
+obtenerLlaveros(Filtro : String) : ResultSet
+obtenerLlavesTotales(Filtro : String) : ResultSet
+obtenerLlaverosTotales(Filtro : String) : ResultSet
+obtenerCodLis() : ResultSet

Estructura de Almacen.java



GLL_Copias
-Num_Copias : int
+GLL_Copias(Llave_Llavero : Gen_LLI, Num_Copias : int) : GLL_Copias
+getLlaveLlavero() : Gen_LLI
+setLlaveLlavero(Llave_Llavero : Gen_LLI)
+getNumCopias() : int
+setNumCopias(Num_Copias : int)
+toString() : String

Estructura de GLL_Copias.java

Gen_LLI
-Codigo : int
-Descripcion : String
+Gen_LLI() : Gen_LLI
+Gen_LLI(F_Alta : Calendar) : Gen_LLI
+Gen_LLI(F_Alta : Calendar, F_Baja : Calendar) : Gen_LLI
+getCodigo() : int
+setCodigo(Codigo : int)
+getDescripcion() : String
+setDescripcion(Descripcion : String)
+toString()

Estructura de Gen_LLI.java

Llave
-Notas : String = null
+Llave(Cod_Llave : int, F_Alta_L : Calendar, Descripcion : String, Departamentos : HashMap<String, Departamento>, Ocasionales : HashMap<String, Ocasional>) : Llave
+Llave(Cod_Llave : int, F_Alta_L : Calendar, F_Baja_L : Calendar, Descripcion : String, Notas : String, Departamentos : HashMap<String, Departamento>, Ocasionales : HashMap<String, Ocasional>) : Llave
+Llave(Cod_Llave : int, F_Alta_L : String) : Llave
+Llave(Cod_Llave : int) : Llave
+getNotas() : string
+setNotas(Notas : String)
+toString() : string
+obtenerLlaves() : String
+obtenerLlaves(DNI : String, Entrada : String, Filtro : String) : ResultSet
+obtenerPrestamos() : ResultSet
+obtenerEmpDepas() : ResultSet
+obtenerOcasional() : ResultSet
+obtenerEmpDepasResto() : ResultSet
+obtenerOcasionalResto() : ResultSet
+eliminarPrestables() : int
+codigoPorDefecto() : int
+eliminarLlaves(Dias : int) : int

Estructura de Llave.java

Llavero
+Llavero(Cod_Llavero : int, F_Alta_LI : Calendar, Descripcion : String, Llaves : HashMap<String, Llave>) : Llavero
+Llavero(Cod_Llavero : int, F_Alta_LI : Calendar, F_Baja_LI : Calendar, Descripcion : String, Llaves : HashMap<String, Llave>) : Llavero
+Llavero(Cod_Llavero : int) : Llavero
+getLlaves() : HashMap<String, Llave>
+setLlaves(Llaves : HashMap<String, Llave>)
+addLlave(Llave : Llave)
+updateLlave(L_old : Llave, L_new : Llave)
+removeLlave(Llave : Llave)
+getLlave(Key_Llave : String) : Llave
+toString() : String
+insertLlaveBD(Llave : Llave) : int
+updateLlaveBD(Llave_old : Llave, Llave_new : Llave) : int
+removeLlaveBD(Llave : Llave) : int
+getLlavesBD(Cod_Llavero : int, F_Alta_LI : String) : HashMap<String, Llave>
+insertLlaveroBD() : int
+updateDatosBD(Llavero : Llavero) : int
+updateLlaveroBD(Llavero : Llavero) : int
+darBajaLlaveroBD() : int
+removeLlaveroBD() : int
+removeLlaveroBD(Llavero : Llavero) : int
+obtenerLlaveros(DNI : String, Entrada : String, Filtro : String) : ResultSet
+obtenerPrestamos() : ResultSet
+obtenerLlaves(Filtro : String) : ResultSet
+codigoPorDefecto() : int
+eliminarLlaveros(Dias : int) : int

Estructura de Llavero.java



Empresa
-Cod_Emp : String
-CIF : String
-Nombre : String
-Direccion : String
+Empresa() : Empresa
+Empresa(Cod_Emp : String, CIF : String, Nombre : String, Direccion : String, Departamentos : HashMap<String, Departamento>) : Empresa
+Empresa(Cadena : String, Codigo : boolean) : Empresa
+getCodEmp() : String
+setCodEmp(Cod_Emp : String)
+getCIF() : String
+setCIF(CIF : String)
+getNombre() : String
+setNombre(Nombre : String)
+getDireccion() : String
+setDireccion(Direccion : String)
+getDepartamentos() : List<Departamento>
+setDepartamentos(Departamentos : HashMap<String, Departamento>)
+addDepartamento(Departamento : Departamento)
+updateDepartamento(Dep_old : Departamento, Dep_new : Departamento)
+removeDepartamento(Departamento : Departamento)
+getDepartamento(Cod_Dep : String) : Departamento
+toString() : String
+insertLlaveBD(Departamento : Departamento, Llave : Llave) : int
+updateLlaveBD(Departamento : Departamento, Llave_old : Llave, Llave_new : Llave) : int
+removeLlaveBD(Departamento : Departamento, Llave : Llave) : int
+insertLlavesBD(Departamento : Departamento) : int
+updateLlavesBD(Departamento : Departamento) : int
+removeLlavesBD(Departamento : Departamento) : int
+getLlavesBD(String : Cod_Dep) : int
+insertPersonaCargoBD(Departamento : Departamento, Persona : Periodico_Contacto, Cargo : Cargo) : int
+updatePersonaCargoBD(Departamento : Departamento, Persona : Periodico_Contacto, Cargo_old : Cargo, Cargo_new : Cargo) : int
+removePersonaCargoBD(Departamento : Departamento, Persona : Periodico_Contacto, Cargo : Cargo) : int
+insertPersonaCargosBD(Departamento : Departamento, Persona : Periodico_Contacto) : int
+updatePersonaCargosBD(Departamento : Departamento, Persona : Periodico_Contacto, Cargos : HashMap<String, Cargo>) : int
+removePersonaCargosBD(Departamento : Departamento, Persona : Periodico_Contacto) : int
+insertPeriodicoContactoBD(Departamento : Departamento, Persona_old : Periodico_Contacto, Persona_new : Periodico_Contacto) : int
+removePeriodicoContactoBD(Departamento : Departamento, Persona : Periodico_Contacto) : int
+getPeriodicoContactosBD(Cod_Dep : String) : HashMap<String, Persona>
+insertVigilanteBD(Departamento : Departamento, Vigilante : Vigilante) : int
+removeVigilanteBD(DNI : String) : int
+getVigilantesBD(Cod_Dep : String) : HashMap<String, Persona>
+insertPersonaBD(Departamento : Departamento, Persona : Persona) : int
+updatePersonaBD(Departamento : Departamento, Persona_old : Persona, Persona_new : Persona) : int
+removePersonaBD(Departamento : Departamento, Persona : Persona) : int
+insertPersonalBD(Departamento : Departamento) : int
+updatePersonalBD(Departamento : Departamento) : int
+removePersonalBD(Departamento : Departamento) : int
+getPersonalBD(Cod_Dep : String) : HashMap<String, Persona>
+insertDepartamentoBD(Departamento : Departamento) : int
+updateDepartamentoBD(Dep_old : Departamento, Dep_new : Departamento) : int
+removeDepartamentoBD(Departamento : Departamento) : int
+getDepartamentoBD(Cod_Dep : String) : Departamento
+insertDepartamentosBD() : int
+updateDepartamentosBD(Departamentos : HashMap<String, Departamento>) : int
+removeDepartamentosBD() : int
+getDepartamentosBD() : HashMap<String, Departamento>
+insertEmpresaBD() : int
+updateDatosBD(Empresa : Empresa) : int
+updateEmpresaBD(Empresa : Empresa) : int
+removeEmpresaBD() : int
+removeEmpresaBD(Empresa : Empresa) : int
+datosEmpresas(CIF : String, Nom_Emp : String, Nom_Dep : String, DNI : String) : ResultSet
+datosDepartamentos(Nom_Emp : String, Nom_Dep : String, Actividad : String, DNI : String) : ResultSet
+datosEmpleados(Nom_Emp : String, Nom_Dep : String, Filtro : String) : ResultSet
+datosContactos(Nom_Emp : String, Nom_Dep : String, Filtro : String) : ResultSet
+obtenerCodEmp() : String
+obtenerCodDep(Nom_Dep : String) : String
+obtenerNomEmps() : ResultSet
+obtenerNomDeps() : ResultSet
+restablecerDatos(nombre : String)
+obtenerLlaves(Cod_Dep : String) : ResultSet
+obtenerLsPorPrestarDep(Cod_Dep : String, Filtro String) : ResultSet

Estructura de Empresa.java



Departamento
-Cod_Dep : String -Nombre : String -Periodicidad : int
+Departamento() : Departamento +Departamento(Cod_Dep : String, Nombre : String, Acceso : T_Acceso, Periodicidad : int, F_Ult_Actu : Calendar, Autorizador : Residente, Personal : HashMap<String, Personal>) : Departamento +Departamento(Cod_Dep : String, Nombre : String, Acceso : T_Acceso, Periodicidad : int, F_Ult_Actu : Calendar, Autorizador : Residente, Personal : HashMap<String, Personal>, Llaves : HashMap<String, Llave>) : Departamento +getCodDep() : String +setCodDep(Cod_Dep) +getNombre() : String +setNombre(Nombre : String) +getAcceso() : T_Acceso +setAcceso(Acceso : T_Acceso) +getPeriodicidad() : int +setPeriodicidad(Periodicidad : int) +getAutorizador() : Residente +setAutorizador(Autorizador : Residente) +getPersonal() : HashMap<String, Personal> +setPersonal(Personal : HashMap<String, Personal>) +addPersonal(Persona : Persona) +updatePersona(Pers_old : Personal, Pers_new : Personal) +removePersona(Persona : Personal) +getPersona(DNI : String) : Personal +getLlaves() : HashMap<String, Llave> +setLlaves(Llaves : HashMap<String, Llave>) +addLlave(Llave : Llave) +updateLlave(Llave_old : Llave, Llave_new : Llave) +removeLlave(Llave : Llave) +getLlave(Key_Llave : String) : Llave +toString() : String +obtenerDatosRenovacionPersona(Dias : int) +eliminarDepartamentos() : int

Estructura de Departamento.java

T_Acceso
-Cod_Acc : String -Actividad : String -Dias : String -Aut_Tel : boolean
+T_Acceso() : T_Acceso +T_Acceso(Cod_Acc : String, Actividad : String, Dias : String, Hora_Inic : Time, Hora_Fin : Time, Aut_Tel : boolean, Cod_Niv : int, F_Alta_N : Calendar) : T_Acceso +T_Acceso(Cod_Acc : String) : T_Acceso +getCodAcc() : String +setCodAcc(Cod_Acc : String) +getActividad() : String +setActividad(Actividad : String) +getDias() : String +setDias(Dias : String) +getAutTel() : boolean +setAutTel(Aut_Tel : boolean) +getNivel() : Nivel +setNivel(Nivel : Nivel) +timeToString(Hora : Calendar) : String +stringToTime(Hora : String) : Calendar +toString() : String +insertTAccesoBD() : int +updateTAccesoBD(T_Acceso : T_Acceso) : int +removeTAccesoBD() : int +obtenerTAccesos() : ResultSet

Estructura de T_Acceso.java



```
Personal
-DNI : String
-Apellido1 : String
-Apellido2 : String
-Nombre : String
-Telefonos : ArrayList<String>
-Emails : ArrayList<String>
+Personal_Permission() : Personal
+Personal_Permission(DNI : String, Apellido1 : String, Apellido2 : String, Nombre : String, Insertador : Personal) : Personal
+Personal_Permission(DNI : String, Apellido1 : String, Apellido2 : String, Nombre : String, Insertador : Personal, Telefonos : ArrayList<String>, Emails : ArrayList<String>) : Personal
+Personal(DNI : String) : Personal
+getDNI() : String
+setDNI(DNI : String)
+getApellido1() : String
+setApellido1(Apellido1 : String)
+getApellido2() : String
+setApellido2(Apellido2 : String)
+getNombre() : String
+setNombre(Nombre : String)
+getInsertador() : Personal
+setInsertador(Insertador : Personal)
+getTelefonos() : ArrayList<String>
+setTelefonos(Telefonos : ArrayList<String>)
+addTelefono(Telefono : String)
+updateTelefono(Tfno_old : String, Tfno_new : String)
+removeTelefono(Telefono : String)
+getEmails() : ArrayList<String>
+setEmails(Emails : ArrayList<String>)
+addEmail(Email : String)
+updateEmail(Email_old : String, Email_new : String)
+removeEmail(Email : String)
+toString() : String
+insertTelefonoBD(Telefono : String) : int
+updateTelefonoBD(Tfno_old : String, Tfno_new : String) : int
+removeTelefonoBD(String Telefono) : int
+insertTelefonosBD() : int
+insertTelefonosBD(Telefonos : ArrayList<String>) : int
+updateTelefonosBD(Telefonos : ArrayList<String>) : int
+removeTelefonosBD() : int
+removeTelefonosBD(Persona : Personal) : int
+insertEmailBD(Email : String) : int
+updateEmailBD(Email_old : String, Email_new : String) : int
+removeEmailBD(Email : String) : int
+insertEmailsBD() : int
+insertEmailsBD(Emails : ArrayList<String>) : int
+updateEmailsBD(Emails : ArrayList<String>) : int
+removeEmailsBD() : int
+removeEmailsBD(Persona : Personal) : int
+insertPersonalPermisoBD() : int
+updatePersonalPermisoBD(Persona : Personal) : int
+removePersonalPermisoBD() : int
+obtenerPersonalCuentas(Filtro : String) : ResultSet
+obtenerPersonalControl(Filtro : String) : ResultSet
+obtenerPersonalControl() : ArrayList<String>
+obtenerDNI() : String
+obtenerDNI(apellidosNombre : String) : String
+obtenerTrns() : ResultSet
+obtenerEmails() : ResultSet
+obtenerLsPrestables() : ResultSet
+obtenerCargos() : ResultSet
+obtenerTipo() : String
+obtenerEmpresa() : String
+obtenerResponsable() : String
+existe() : boolean
+obtenerInsertadores() : ResultSet
+obtenerDatosPrestamosPendientes() : ResultSet
+obtenerDatosPrestamosPendientes(Dias : int) : ResultSet
+eliminarPersonal() : int
```

Estructura de Personal.java

```
Periodico_Contacto
+Periodico_Contacto() : Periodico_Contacto()
+Periodico_Contacto(DNI : String, Apellido1 : String, Apellido2 : String, Nombre : String, Insertador : Personal, Telefonos : ArrayList, Emails : ArrayList) : Periodico_Contacto
+Periodico_Contacto(DNI : String, Apellido1 : String, Apellido2 : String, Nombre : String, Insertador : Personal, Cargos : ArrayList<Cargo>, Telefonos : ArrayList, Emails : ArrayList) : Periodico_Contacto
+Periodico_Contacto(DNI : String) : Periodico_Contacto
+Periodico_Contacto(DNI : String, Cargos : ArrayList<Cargo>) : Periodico_Contacto
+getCargos() : ArrayList<Cargo>
+setCargos(Cargos : ArrayList<Cargo>)
+addCargo(Cargo : Cargo)
+updateCargo(Cargo_old : Cargo, Cargo_new : Cargo)
+removeCargo(Cargo : Cargo)
+toString() : String
+obtenerContactos() : ResultSet
+actualizarFUA()
```

Estructura de Periodico_Contacto.java



Cargo
-Cod_Cargo : String
-Descripcion : String
+Cargo(Cod_Cargo : String, Descripcion : String) : Cargo
+Cargo(Cod_Cargo : String) : Cargo
+getCodCargo() : String
+setCodCargo(CodCargo : String)
+getDescripcion() : String
+setDescripcion(Descripcion : String)
+toString()
+insertCargoBD() : int
+updateCargo(Cargo : Cargo) : int
+removeCargo() : int
+obtenerCargos() : ResultSet
+obtenerCodCargo(Descripcion : String) : String
+obtenerDescripCargos() : ResultSet

Estructura de Cargo.java

Vigilante
-Cod_Vig : int
+Vigilante() : Vigilante
+Vigilante(DNI : String, Apellido1 : String, Apellido2 : String, Nombre : String, Insertador : Personal, Cod_Vig : int, Telefonos : ArrayList, Emails : ArrayList) : Vigilante
+Vigilante(DNI : String, Cod_Vig : int) : Vigilante
+Vigilante(DNI : String) : Vigilante
+getCodVig() : String
+setCodVig(Cod_Vig : int)
+toString() : String
+existeVigilante() : boolean
+obtenerCodsVigilantes() : ResultSet
+obtenerDNI(Cod_Vig : int) : String

Estructura de Vigilante.java

Ocasional
-Empresa : String
-Acceso : T_Acceso
-Notas : String = null
+Ocasional() : Ocasional
+Ocasional(DNI : String, Apellido1 : String, Apellido2 : String, Nombre : String, Insertador : Personal, Empresa : String, Acceso : T_Acceso, F_Caducidad : Calendar, Telefonos : ArrayList, Emails : ArrayList) : Ocasional
+Ocasional(DNI : String, Apellido1 : String, Apellido2 : String, Nombre : String, Insertador : Personal, Empresa : String, Acceso : T_Acceso, F_Caducidad : Calendar, Notas : String, Telefonos : ArrayList, Emails : ArrayList) : Ocasional
+Ocasional(DNI : String, Empresa : String) : Ocasional
+Ocasional(DNI : String) : Ocasional
+getEmpresa() : String
+setEmpresa(Empresa : String)
+getAcceso() : T_Acceso
+setAcceso(Acceso : T_Acceso)
+getFCaducidad() : Calendar
+setFCaducidad(F_Caducidad : Calendar)
+getNotas() : String
+setNotas(Notas : String)
+getLlaves() : HashMap<String, Llave>
+setLlaves(Llaves : HashMap<String, Llave>)
+addLlave(Llave : Llave)
+updateLlave(Llave_old : Llave, Llave_new : Llave)
+removeLlave(Llave : Llave)
+getLlave(Key_Llave : String) : Llave
+toString() : String
+insertLlaveBD(Llave : Llave) : int
+updateLlaveBD(Llave_old : Llave, Llave_new : Llave) : int
+removeLlaveBD(Llave : Llave) : int
+insertLlavesBD() : int
+updateLlavesBD() : int
+removeLlavesBD() : int
+getLlavesBD() : HashMap<String, Llave>
+insertOcasionalBD() : int
+updateOcasionalBD(Permiso_Ocasional : Ocasional) : int
+removeOcasionalBD() : int
+obtenerEmpresas(DNI : String) : ResultSet
+obtenerOcasional() : ResultSet
+eliminarOcasional() : int
+obtenerLlaves() : ResultSet
+obtenerLsPorPrestarOcas(Filtro : String) : ResultSet

Estructura de Ocasional.java

Residente
+Residente(DNI : String, Apellido1 : String, Apellido2 : String, Nombre : String, Insertador : Personal, Nivel : Nivel) : Residente
+Residente(DNI : String, Apellido1 : String, Apellido2 : String, Nombre : String, Insertador : Personal, Emails : List<String>, Nivel : Nivel) : Residente
+getNivel() : Nivel
+setNivel(Nivel : Nivel)
+toString() : String
+insertResidenteBD() : int
+updateResidenteBD(Residente : Residente)
+removeResidenteBD() : int

Estructura de Residente.java



Acceso_Sistema
+Acceso_Sistema(): Acceso_Sistema
+Acceso_Sistema(Cuenta : Cuenta, Fecha : Calendar) : Acceso_Sistema
+getCuenta() : Cuenta
+setCuenta(Cuenta : Cuenta)
+toString() : String
+insertAccesoSistemaBD() : int
+updateAccesoSistemaBD(Acceso_Sistema : Acceso_Sistema) : int
+removeAccesoSistemaBD() : int
+primerVigilante() : boolean
+obtenerAccesosSistema(Fecha_Inic : String, Fecha_Fin : String, Hora_Inic : String, Hora_Fin : String, Filtro : String) : ResultSet

Estructura de Acceso_Sistema.java

Cuenta
-Usuario : String
-Pass : String = "cambiar"
+Cuenta() : Cuenta
+Cuenta(Usuario : String, Tipo : T_Usuario, Persona : Persona) : Cuenta
+Cuenta(Usuario : String, Pass : String, Tipo : T_Usuario, Persona : Persona) : Cuenta
+Cuenta(Usuario : String) : Cuenta
+getUsuario() : String
+setUsuario(Usuario : String)
+getPass() : String
+setPass(Pass : String)
+getTipo() : String
+setTipo(Tipo : String)
+getPersona() : Persona
+setPersona(Persona : Persona)
+toString() : String
+insertCuentaBD() : int
+updateCuentaBD(Cuenta : Cuenta) : int
+removeCuentaBD() : int
+checkPass(Pass : String) : String
+obtenerCuentas() : ResultSet
+obtenerDNIsAdmAut() : ResultSet

Estructura de Cuenta.java

T_Usuario
-Tipo : String
-Descripcion : String
+T_Usuario() : T_Usuario
+T_Usuario(Tipo : String, Descripcion : String) : T_Usuario
+T_Usuario(Tipo : String) : T_Usuario
+getTipo() : String
+setTipo(Tipo : String)
+getDescripcion() : String
+setDescripcion(Descripcion : String)
+toString() : String
+insertTUsuarioBD() : int
+updateTUsuarioBD(T_Usuario : T_Usuario) : int
+removeTUsuarioBD() : int
+obtenerCodsTUsuarios() : ResultSet
+obtenerTUsuarios() : ResultSet

Estructura de T_Usuario.java



```

Acceso
-Motivo : String
-Otros_Datos : String = null
+Acceso() : Acceso
+Acceso(Persona : Personal, Entrada : Calendar, Motivo : String, Tarjeta : Tarjeta, Insertador : Personal) : Acceso
+Acceso(Persona : Personal, Entrada : Calendar, Motivo : String, Tarjeta : Tarjeta, GLLDs : HashMap<String, GLL_Dev>, Insertador : Personal) : Acceso
+Acceso(Persona : Personal, Entrada : Calendar, Salida : Calendar, Motivo : String, Tarjeta : Tarjeta, GLLDs : HashMap<String, GLL_Dev>, Otros_Datos : String, Insertador : Personal) : Acceso
+Acceso(Entrada : String) : Acceso
+getPersona() : Persona
+setPersona(Persona : Personal)
+poseeSalida() : boolean
+getMotivo() : String
+setMotivo(Motivo : String)
+getTarjeta() : Tarjeta
+setTarjeta(Tarjeta : Tarjeta)
+getGLLDs() : HashMap<String, GLL_Dev>
+setGLLDs(Llaves_Llaveros : HashMap<String, GLL_Dev>)
+addGLLD(GLLD : GLL_Dev)
+updateGLLD(LLI_old : GLL_Dev, LLI_new : GLL_Dev)
+removeGLLD(GLLD : GLL_Dev)
+getGLLD(Key_GLLD : String) : GLL_Dev
+devGLLD(Key_GLLD : String)
+getOtrosDatos() : String
+setOtrosDatos(Otros_Datos : String)
+getInsertador() : Personal
+setInsertador(Personal_Permission)
+toString() : String
+insertGLLDBD(GLLD : GLL_Dev) : int
+updateGLLDBD(LLI_old : GLL_Dev, LLI_new : GLL_Dev) : int
+devGLLDBD(GLLD : GLL_Dev) : int
+devGLLDBD(Tipo : String,Codigo : int) : int
+removeGLLDBD(GLLD : GLL_Dev) : int
+getGLLDBD(GLLD_Key : String) : GLL_Dev
+getGLLDBD(Tipo : String, Codigo : int) : GLL_Dev
+insertGLLDsBD() : int
+insertGLLDsBD(GLLDs : HashMap<String, GLL_Dev>) : int
+updateGLLDsBD(GLLDs : HashMap<String, GLL_Dev>) : int
+removeGLLDsBD() : int
+getGLLDsBD() : HashMap<String, GLL_Dev>
+insertAccesoBD() : int
+updateAccesoBD(Acceso : Accesos) : int
+updateDatosAccesoBD(Acceso : Acceso) : int
+removeAccesoBD() : int
+removeAccesoBD(Acceso : Accesos) : int
+obtenerAccesosSinSalida() : ArrayList<Acceso>
+darSalidaAutomatica()
+obtenerFAltal(Codigo : int) : String
+obtenerAccesos(Entrada : boolean, Fecha_Inic : String, Fecha_Fin : String, Hora_Inic : String, Hora_Fin : String, Filtro : String) : ResultSet
+obtenerTarjetaAcceso(Entrada : String) : ResultSet
+obtenerLasJasAcceso() : ResultSet
+darSalida()
+devolverLlavesLlaveros()
+devolverLlaves()
+devolverLlaveros()
+eliminarAccesos(Dias : int) : int

```

Estructura de Acceso.java

```

GLL_Dev
-Dev : boolean
+GLL_Dev(Llave_Llavero : Gen_LLI) : GLL_Dev
+GLL_Dev(Llave_Llavero : Gen_LLI, Dev : boolean) : GLL_Dev
+getLlaveLlavero() : Gen_LLI
+setLlaveLlavero(Llave_Llavero : Gen_LLI)
+getDev() : boolean
+setDev(Dev : boolean)
+toString() : String

```

Estructura de GLL_Dev.java



Nota
-Tema : String -Texto : String
+Nota() : Nota +Nota(Fecha : Calendar, Redactor : Persona, Tema : String, Texto : String, Destinatarios : HashMap<String, Destinatario>) : Nota +Nota(Fecha : String, Redactor : String) : Nota +getRedactor() : Persona +setRedactor(Redactor : Persona) +getTema() : String +setTema(Tema : String) +getTexto() : String +setTexto(Texto : String) +getDescartada() : boolean +setDescartada(Descartada : boolean) +getDestinatarios() : HashMap<String, Destinatario> +setDestinatarios(Destinataros : HashMap<String, Destinatario>) +addDestinatario(Destinataro : Destinatario) +updateDestinatario(Dest_old : Destinatario, Dest_new : Destinatario) +removeDestinatario(Destinataro : Destinatario) +getDestinatario(DNI : String) : Destinatario +toString() +insertDestinatarioBD(Destinataro : Destinatario) : int +updateDestinatarioBD(Dest_old : Destinatario, Dest_new : Destinatario) : int +removeDestinatarioBD(Destinataro : Destinatario) : int +getDestinatarioBD(DNI : String) : Destinatario +insertDestinatariosBD(Destinataros : HashMap<String, Destinatario>) : int +updateDestinatariosBD(Destinataros : HashMap<String, Destinatario>) : int +removeDestinatariosBD() : int +getDestinatariosBD() : HashMap<String, Destinatario> +insertNotaBD() : int +updateNotaBD(Nota : Nota) : int +removeNotaBD() : int +removeNotaBD(Nota : Nota) : int +obtenerDestinatarios(Filtro : String) : ResultSet +obtenerBandejaEntrada(DNI : String) : ResultSet +obtenerBandejaSalida(DNI : String) +leida(DNI : String) +descartada(DNI : String) +descartada()

Estructura de Nota.java

Destinatario
-Leida : boolean = False -Descartada : boolean = False
+Destinatario(Receptor : Persona) : Destinatario +Destinatario(Receptor : Persona, Leida : boolean, Descartada : boolean) : Destinatario +getReceptor() : Persona +setReceptor(Receptor : Persona) +getLeida() : boolean +setLeida(Leida : boolean) +getDescartada() : boolean +setDescartada(Descartada : boolean) +toString() : String

Estructura de Destinatario.java

Parte_Seguridad
-Incidencias : String
+Parte_Seguridad(Fecha : Calendar, Vigilante : Vigilante, Incidencias : String) : Parte_Seguridad +Parte_Seguridad(Fecha : Calendar) : Parte_Seguridad +getVigilante() : Vigilante +setVigilante(Vigilante : Vigilante) +getIncidencias() : String +setIncidencias(Incidencias : String) +toString() : String +insertParteSeguridadBD() : int +updateParteSeguridadBD(Parte_Seguridad : Parte_Seguridad) : int +removeParteSeguridadBD() : int +obtenerPartes(Fecha_Inic : String, Fecha_Fin : String, Hora_Inic : String, Hora_Fin : String, Cod_Vig : String) : ResultSet +obtenerPartesEditables(DNI : String) : ArrayList

Estructura de Parte_Seguridad.java



Utilidades Calendar
+calendarToString(Fecha : Calendar) : String
+stringToCalendar(Fecha : String) : Calendar
+stringToTime(Fecha : String) : Calendar

Estructura de UtilidadesCalendar.java

UnaFecha
#Fecha_i = Now()
+UnaFecha() : UnaFecha
+UnaFecha(Fecha_i : Calendar) : UnaFecha
+getFecha() : Calendar
+setFecha(Fecha_i : Calendar)
+toString() : String

Estructura de UnaFecha.java

DosFechas
-Fecha_j = null
+DosFechas() : DosFechas
+DosFechas(Fecha_j : Calendar) : DosFechas
+DosFechas(Fecha_i : Calendar, Fecha_j : Calendar) : DosFechas
+getFechaJ() : Calendar
+setFechaJ(Fecha_j : Calendar)
+toString() : String

Estructura de DosFechas.java

UtilidadesBD
-DireccionURL : String
-Usuario : String
-Pass : String
#Con : Connection
+abrirConexion() : Statement
+cerrarConexion()
+ejecutarQuery(Query : String) : ResultSet
+ejecutarQuery(DireccionURL : String, Usuario : String, Pass : String, Query : String) : ResultSet
+ejecutarUpdate(Query : String)
+ejecutarUpdate(DireccionURL : String, Usuario : String, Pass : String, Query : String)

Estructura de UtilidadesBD.java

DataException
+toString() : String

Estructura de DataException.java

Pamplona, a 16 de abril de 2010.

Héctor Napal Basterrica
Responsable del proyecto

accessManager · Descripción de la API final · Página 19 de 19



accessManager

Desarrollo de un sistema de control de accesos a un centro

Plan de pruebas

Fecha de entrega: 16 de abril de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Estructura del documento

2. Requerimientos de las pruebas

3. Estrategias de prueba

4. Recursos

5. Concreción de las pruebas

6. Resultados de las pruebas

- 6.1. Iteración C1: La base de datos
- 6.2. Iteración C2: API para la base de datos
- 6.3. Iteración C3: Accesos
- 6.4. Iteración C4: Empresas y personal
- 6.5. Iteración C5: Tarjetas, tipos de acceso y cargos
- 6.6. Iteración C6: Zonas, llaves y llaveros
- 6.7. Iteración C7: Préstamos
- 6.8. Iteración C8: Cuentas
- 6.9. Iteración C9: Partes de seguridad
- 6.10. Iteración C10: Notas
- 6.11. Iteración C11: Base de datos
- 6.12. Iteración C12: Diseño, personalización y configuración
- 6.13. Prueba de aceptación
- 6.14. Prueba de concurrencia



1. Introducción

1.1. Propósito

Este documento pretende ser un resumen de todo el plan de pruebas llevado a cabo en el desarrollo de accessManager, recogiendo especificaciones, resultados y conclusiones obtenidas en la realización de las pruebas.

Los principales objetivos que nos planteamos en este artefacto son:

- Identificar los elementos que se deben probar.
- Definir estrategias a seguir para realizar las pruebas.
- Estimar los recursos necesarios para llevar a cabo el proceso de pruebas.
- Recoger los resultados obtenidos en cada una de las pruebas, junto con una evaluación de éstos.

No debemos olvidar que la finalidad última de este documento es la de probar que toda la especificación de requisitos ha sido implementada por el equipo de desarrollo, creando con ello un producto válido para las pretensiones del cliente.

1.2. Audiencia

Este es un documento que consta de mucho valor para el cliente, pues es una fiel prueba de la calidad del producto que se encuentra en desarrollo y que obtendrá en el futuro. Será mostrado al cliente a lo largo de todas las iteraciones, demostrándole que los diversos prototipos son desarrollados con unos niveles aceptables de calidad y que toda la funcionalidad implementada se ejecuta de manera correcta y eficiente, según lo establecido.

1.3. Estructura del documento

En los tres primeros capítulos realizamos una breve introducción al proceso de pruebas, identificando qué tipo de pruebas debemos realizar, los pasos que debemos seguir y qué recursos necesitamos, respectivamente.

A continuación mostrando una tabla que relaciona los elementos a probar en cada iteración con los documentos que recogen las pruebas ejecutadas, poniendo punto y final al documento con una enumeración de todas las pruebas realizadas y los resultados y conclusiones que hemos obtenido de ellas.



2. Requerimientos de las pruebas

Es buena práctica comenzar enumerando todos los tipos de pruebas que debemos realizar:

PRUEBAS			
TIPO	INSTANTE	OBJETIVO	DESCRIPCIÓN
De unidad	Con cada componente software desarrollado.	Probar la especificación de los elementos software.	Pruebas funcionales (de caja negra) y estructurales (de caja blanca) que validen la eficacia funcional de cada componente desarrollado.
De integración	Con cada componente software desarrollado.	Probar la documentación de diseño y arquitectura.	Validar la eficacia funcional del sistema y subsistemas que lo componen.
De sistema	Al finalizar cada iteración.	Probar la especificación de requisitos del software.	Comprobar que se han incorporado todos los casos de uso, que el funcionamiento es correcto y robusto y que la documentación a entregar al usuario es la adecuada.
De aceptación	En la segunda iteración de la fase de Transición.	Probar los requisitos de usuario (contrato).	Evaluar conjuntamente con el cliente el producto, haciéndole ver que es el apropiado.

Tipos de pruebas a realizar

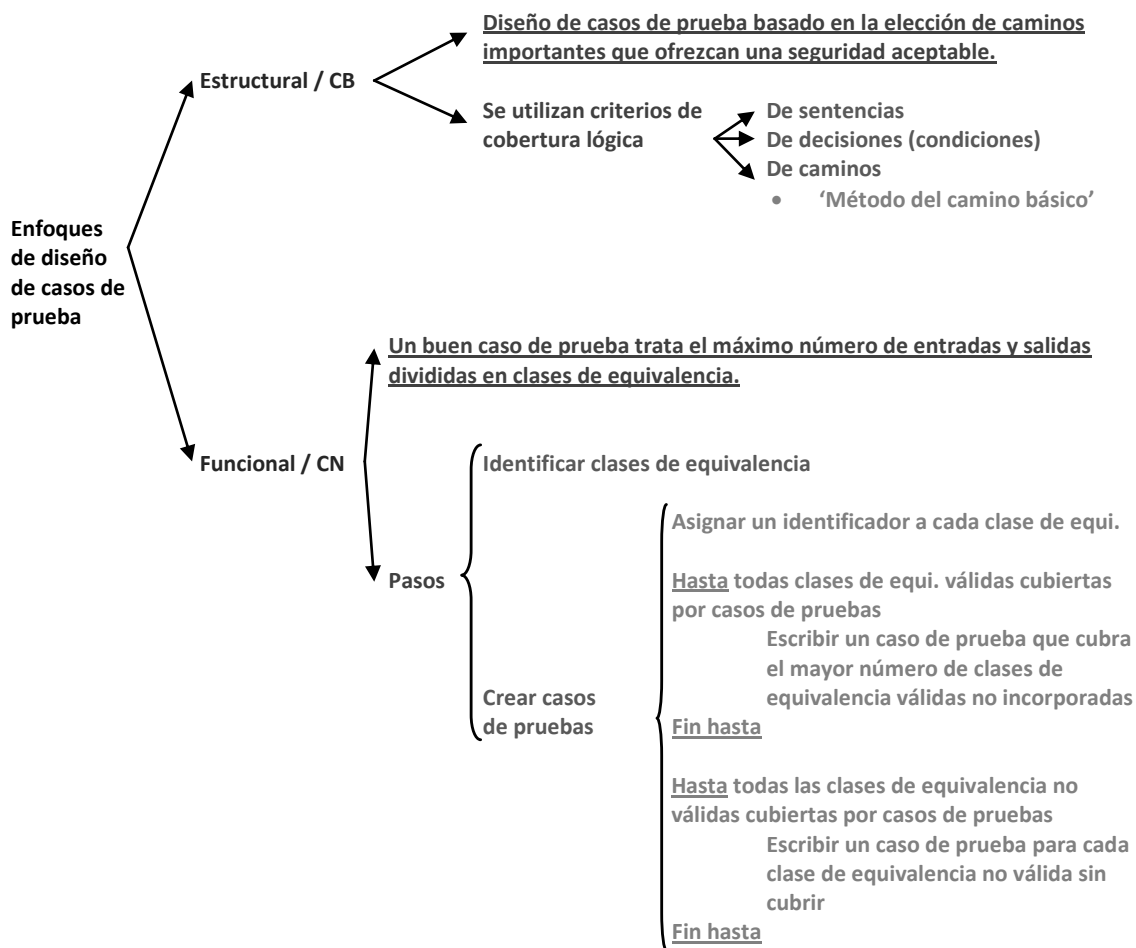


3. Estrategias de prueba

En la sección anterior hemos descrito qué tipos de pruebas debemos realizar, con que ahora es el turno de definir cómo las realizaremos. A continuación presentamos los procedimientos y técnicas que emplearemos en el desarrollo de las mismas.

PRUEBAS	
TIPO	PROCEDIMIENTOS Y TÉCNICAS
De unidad	<p>Cada elemento software será probado individualmente mediante los enfoques de diseño de casos de prueba usuales y que esquematizamos a continuación de esta tabla.</p> <p>El enfoque práctico será el siguiente:</p> <ol style="list-style-type: none">1. Si existen combinaciones de entrada → Crear grafos causa-efecto y añadir los casos de prueba oportunos.2. Identificar las clases de equivalencia empleando la técnica de análisis de valores límite. Añadir los casos de prueba no incluidos anteriormente.3. Añadir casos de prueba mediante la técnica de conjetura de errores.4. Ejecutar los casos de prueba generados hasta el momento (de caja negra) y analizar la cobertura obtenida.5. Examinar la lógica del programa para añadir los casos de prueba (de caja blanca) precisos para cumplir el criterio de cobertura elegido.
De integración	<p>El desarrollo de las pruebas de integración y de sistema se realizarán durante todas y cada una de las iteraciones de construcción del producto, teniendo la siguiente secuencia de actividades:</p> <ol style="list-style-type: none">1. Planificar las pruebas de cada iteración.<ol style="list-style-type: none">1.1. <i>De integración (para cada construcción).</i>1.2. <i>De sistema (al final de cada iteración).</i>2. Diseñar las pruebas para cada construcción (caso de uso).<ol style="list-style-type: none">2.1. <i>Casos de prueba.</i>2.2. <i>Procedimientos de prueba.</i>
De sistema	<ol style="list-style-type: none">3. Implementar las pruebas.4. Ejecutar las pruebas.5. Evaluar los resultados obtenidos.
De aceptación	<p>Se prueban todos y cada uno de los casos de uso con el cliente, con datos válidos y erróneos y mostrándole que el sistema se comporta según lo previsto.</p>

Procedimientos y técnicas a utilizar en las pruebas



4. Recursos

Estos son los recursos necesarios para llevar a cabo todo el desarrollo de las pruebas:

HARDWARE			
TIPO	CANTIDAD	TIPO DE PRUEBA	DESCRIPCIÓN
PC	1	Todas	Computadora con la que poder realizar el diseño e implementación de las pruebas.
PC	11	De sistema	Ordenadores para poder realizar la prueba final de todo el producto en unas condiciones de concurrencia máxima.

Recursos hardware



HUMANOS		
ROL	CANTIDAD	RESPONSABILIDADES
Diseñador	1	Identificar, priorizar, diseñar e implementar los casos de prueba, recogiendo todo en el 'Plan de pruebas'.
Probador (tester)	12	Ejecutar las pruebas, recogiendo los resultados obtenidos y realizando una evaluación de los mismos.

Recursos humanos

5. Concreción de las pruebas

Una vez establecidas qué pruebas se deben realizar, cómo deben ser desarrolladas y los recursos necesarios, pasamos a concretarlas en los artefactos que las desarrollarán:

PRUEBAS			
ITERACIÓN	ELEMENTOS A PROBAR	TIPOS DE PRUEBAS	DOCUMENTO ASOCIADO
C1	Implementación de la base de datos.	De unidad.	<i>Iteración C1: La base de datos</i>
C2	Implementación de la API para la base de datos.	De unidad.	<i>Iteración C2: API para la base de datos</i>
C3	Todos los casos de uso referentes a los accesos al centro.	De unidad, integración y sistema.	<i>Iteración C3: Accesos</i>
C4	Ident. con empresas y personal.	De unidad, integración y sistema.	<i>Iteración C4: Empresas y personal</i>
C5	Ident. con tarjetas, colores, niveles, tipos de acceso y cargos.	De unidad, integración y sistema.	<i>Iteración C5: Tarjetas, tipos de acceso y cargos</i>
C6	Ident. con zonas, llaves y llaveros.	De unidad, integración y sistema.	<i>Iteración C6: Zonas, llaves y llaveros</i>
C7	Ident. con préstamos.	De unidad, integración y sistema.	<i>Iteración C7: Préstamos</i>
C8	Ident. con cuentas.	De unidad, integración y sistema.	<i>Iteración C8: Cuentas</i>
C9	Ident. con partes de seguridad.	De unidad, integración y sistema.	<i>Iteración C9: Partes de seguridad</i>

Iteraciones, elementos, tipos de pruebas y documentos asociados



C10	Ident. con notas.	De unidad, integración y sistema.	<i>Iteración C10: Notas</i>
C11	Ident. con base de datos.	De unidad, integración y sistema.	<i>Iteración C11: Base de datos</i>
C12	Comprobar que el diseño es el correcto, que la muestra de los componentes de la aplicación se realiza acorde al tipo de usuario registrado y que los detalles de configuración tienen efecto.	De integración y sistema.	<i>Iteración C12: Diseño, personalización y configuración</i>
T1	Asegurar la idoneidad de toda la documentación y software elaborado.	---	<i>Iteración T1: Preparación de la release final</i>
T2	Probar toda la aplicación con el cliente, creándole confianza en que el software desarrollado está capacitado para dar respuesta a sus pretensiones.	De aceptación.	<i>Iteración T2: Entrega</i>
T3	Asegurar que el entrenamiento realizado da lugar a operadores que saben manejar a la perfección el sistema, además de proporcionarles manuales que puedan ayudarlos en futuras dudas.	---	<i>Iteración T3: Entrenamiento, soporte y mantenimiento</i>

Iteraciones, elementos, tipos de pruebas y documentos asociados



6. Resultados de las pruebas

A continuación se resumen todas las pruebas realizadas en cada iteración junto con los resultados obtenidos y las conclusiones que se extraen de ellas:

6.1. Iteración C1: La base de datos

Nº	Prueba	Resultado
1.01	Disparador <i>ins_nivel</i>	✓
1.02	Disparador <i>actu_nivel</i>	✓
1.03	Disparador <i>ins_col</i>	✓
1.04	Disparadores <i>actu_col</i> y <i>borr_col</i>	✓
1.05	Disparador <i>ins_tarj</i>	✓
1.06	Disparadores <i>actu_tarj</i> y <i>borr_tarj</i>	✓
1.07	Disparador <i>borr_zona</i>	✓
1.08	Disparador <i>ins_llave</i>	✓
1.09	Disparador <i>borr_l</i>	✓
1.10	Disparador <i>borr_al</i>	✓
1.11	Disparador <i>ins_llavero</i>	✓
1.12	Disparador <i>borr_ll</i>	✓
1.13	Disparador <i>borr_all</i>	✓
1.14	Disparador <i>borr_dep</i>	✓
1.15	Disparador <i>borr_contacto</i>	✓
1.16	Disparador <i>borr_cuenta</i>	✓
1.17	Disparador <i>ins_acc</i>	✓
1.18	Disparador <i>actu_nota</i>	✓
1.19	Disparador <i>actu_dest</i>	✓

El funcionamiento de todos los disparadores es el deseado, encargándose éstos de realizar toda la funcionalidad que les asignamos.

Resultados de las pruebas realizadas en la iteración 'C1 - La base de datos'

6.2. Iteración C2: API para la base de datos

Nº	Prueba	Resultado
2.01	Comunicación API - BD	✓

La comunicación entre la API y la base de datos es correcta, permitiendo utilizar la segunda desde la primera para obtener los datos almacenados.

Resultados de las pruebas realizadas en la iteración 'C2 - API para la base de datos'



6.3. Iteración C3: Accesos

Nº	Prueba	Resultado
3.01	Caso de uso 1: Login	✓
3.02	Caso de uso 2: Visualizar accesos	✓
3.03	Caso de uso 3: Visualizar acceso	✓
3.04	Caso de uso 4: Dar entrada	✓
3.05	Caso de uso 5: Prestar llaves y/o llaveros	✓
3.06	Caso de uso 6: Borrar acceso	✓
3.07	Caso de uso 7: Modificar acceso	✓
3.08	Caso de uso 8: Borrar préstamo	✓
3.09	Caso de uso 9: Dar salida	✓
3.10	Caso de uso 10: Devolver llaves y/o llaveros	✓
3.11	Caso de uso 11: Listar accesos	✓
3.12	Caso de uso 12: Dar acceso automático	✓

Todos los casos de uso correspondientes a los accesos (del 1 al 12) han sido desarrollados y probados de forma satisfactoria, comportándose todo de acuerdo a lo establecido en *Especificación de los casos de uso*.

Resultados de las pruebas realizadas en la iteración 'C3 - Accesos'



6.4. Iteración C4: Empresas y personal

Nº	Prueba	Resultado
4.01	Casos de uso 13, 19 y 25: Visualizaciones	✓
4.02	Caso de uso 14: Visualizar empresa	✓
4.03	Caso de uso 15: Añadir empresa	✓
4.04	Caso de uso 16: Borrar empresa	✓
4.05	Caso de uso 17: Modificar empresa	✓
4.06	Caso de uso 20: Visualizar departamento	✓
4.07	Caso de uso 21: Añadir departamento	✓
4.08	Caso de uso 22: Borrar departamento	✓
4.09	Caso de uso 23: Modificar departamento	✓
4.10	Caso de uso 26: Visualizar persona	✓
4.11	Caso de uso 27: Añadir persona	✓
4.12	Caso de uso 28: Borrar persona	✓
4.13	Caso de uso 29: Modificar persona	✓
4.14	Casos de uso 30, 31, 32 y 33: Añadir o borrar teléfono o E-Mail	✓
4.15	Casos de uso 34 y 35: Añadir o borrar cargo	✓
4.16	Casos de uso 18, 24 y 36: Listar empresas, departamentos y personal	✓

He aquí los resultados obtenidos en la ejecución de pruebas correspondiente a la cuarta iteración de la fase de Construcción. Todas ellas han sido superadas obteniendo resultados positivos y éstas, junto con las que ya hemos realizado, suponen un fiel reflejo de que en estos momentos disponemos de un prototipo software cuyos componentes interaccionan a la perfección para concretar el fin para el que han sido creados. La implementación de todos los casos de uso propuesta como uno de los objetivos a alcanzar ha sido documentada de forma extensa y precisa en el correspondiente escrito de iteración, cuyo contenido será más que suficiente para poder desarrollar las futuras tareas de mantenimiento.

Resultados de las pruebas realizadas en la iteración 'C4 - Empresas y personal'



6.5. Iteración C5: Tarjetas, tipos de acceso y cargos

Nº	Prueba	Resultado
5.01	Casos de uso 37, 38, 43, 44, 49, 50, 55, 60 y 61: Visualizaciones	✓
5.02	Caso de uso 39: Añadir nivel	✓
5.03	Caso de uso 40: Borrar nivel	✓
5.04	Caso de uso 41: Modificar nivel	✓
5.05	Caso de uso 42: Dar de baja nivel	✓
5.06	Caso de uso 45: Añadir color	✓
5.07	Caso de uso 46: Borrar color	✓
5.08	Caso de uso 46: Modificar color	✓
5.09	Caso de uso 48: Dar de baja color	✓
5.10	Caso de uso 51: Añadir tarjeta	✓
5.11	Caso de uso 52: Borrar tarjeta	✓
5.12	Caso de uso 53: Modificar tarjeta	✓
5.13	Caso de uso 54: Dar de baja tarjeta	✓
5.14	Caso de uso 56: Visualizar tipo de acceso	✓
5.15	Caso de uso 57: Añadir tipo de acceso	✓
5.16	Caso de uso 58: Borrar tipo de acceso	✓
5.17	Caso de uso 59: Modificar tipo de acceso	✓
5.18	Caso de uso 62: Añadir cargo	✓
5.19	Caso de uso 63: Borrar cargo	✓
5.20	Caso de uso 64: Modificar cargo	✓

La quinta iteración de la fase de Construcción ha dado lugar a las pruebas sobre estas líneas citadas. Como vemos, todas ellas han sido realizadas con éxito, consiguiendo una perfecta integración de componentes y que el sistema recoja todos los casos de uso que nos propusimos.

Resultados de las pruebas realizadas en la iteración 'C5 - Tarjetas, tipos de acceso y cargos'



6.6. Iteración C6: Zonas, llaves y llaveros

Nº	Prueba	Resultado
6.01	Casos de uso 65, 70a, 77 y 86: Visualizaciones	✓
6.02	Casos de uso 66 y 70b: Visualizar zona del centro y sus llaves	✓
6.03	Caso de uso 67: Añadir zona del centro	✓
6.04	Caso de uso 68: Borrar zona del centro	✓
6.05	Caso de uso 69: Modificar zona del centro	✓
6.06	Caso de uso 71: Visualizar llave	✓
6.07	Caso de uso 72: Añadir llave	✓
6.08	Caso de uso 73: Borrar llave	✓
6.09	Caso de uso 74: Dar de baja llave	✓
6.10	Caso de uso 75: Modificar llave	✓
6.11	Caso de uso 78: Visualizar llavero	✓
6.12	Caso de uso 79: Añadir llavero	✓
6.13	Caso de uso 80: Borrar llavero	✓
6.14	Caso de uso 81: Dar de baja llavero	✓
6.15	Caso de uso 82: Modificar llavero	✓
6.16	Caso de uso 83: Añadir llave	✓
6.17	Caso de uso 84: Borrar llave	✓
6.18	Caso de uso 87: Visualizar almacén de llaves	✓
6.19	Caso de uso 88: Añadir almacén de llaves	✓
6.20	Caso de uso 89: Borrar almacén de llaves	✓
6.21	Caso de uso 90: Modificar almacén de llaves	✓
6.22	Caso de uso 91: Añadir llave	✓
6.23	Caso de uso 92: Borrar llave	✓
6.24	Caso de uso 93: Modificar copias llave	✓
6.25	Caso de uso 94: Añadir llavero	✓
6.26	Caso de uso 95: Borrar llavero	✓
6.27	Caso de uso 96: Modificar copias llavero	✓
6.28	Caso de uso 76, 85 y 97: Listar llaves, llaveros y almacenes	✓

Estos 28 casos de prueba realizados y ejecutados a lo largo de la iteración 'C6 – Zonas, llaves y llaveros' ponen de manifiesto que el trabajo realizado en ella es correcto, habiendo cumplido con los objetivos propuestos, consiguiendo un software robusto y una documentación adecuada para el futuro mantenimiento.

Resultados de las pruebas realizadas en la iteración 'C6 - Zonas, llaves y llaveros'



6.7. Iteración C7: Préstamos

Nº	Prueba	Resultado
7.01	Caso de uso 98: Visualizar préstamos	✓
7.02	Caso de uso 99: Visualizar datos del prestatario	✓
7.03	Caso de uso 100: Devolver préstamo	✓
7.04	Caso de uso 101: Enviar recordatorio vía E-Mail	✓
7.05	Caso de uso 102: Listar préstamos pendientes	✓

Los casos de uso desarrollados en esta iteración han superado todas las pruebas de forma positiva, revelando que el sistema se comporta conforme a lo establecido y diseñado y que la integración entre todos los componentes sigue siendo óptima.

Resultados de las pruebas realizadas en la iteración 'C7 - Préstamos'

6.8. Iteración C8: Cuentas

Nº	Prueba	Resultado
8.01	Casos de uso 103, 104, 109, 110 y 114: Visualizaciones	✓
8.02	Caso de uso 105: Añadir cuenta	✓
8.03	Caso de uso 106: Borrar cuenta	✓
8.04	Caso de uso 107: Modificar cuenta	✓
8.05	Caso de uso 108: Listar cuentas	✓
8.06	Caso de uso 111: Añadir tipo de usuario	✓
8.07	Caso de uso 112: Borrar tipo de usuario	✓
8.08	Caso de uso 113: Modificar tipo de usuario	✓
8.09	Caso de uso 115: Listar accesos al sistema	✓
8.10	Caso de uso 116: Modificar datos de cuenta	✓

Las pruebas ejecutadas a lo largo de la octava iteración ponen de manifiesto que los casos de uso del 103 al 116 han sido implementados satisfactoriamente, cumpliendo los requisitos de eficiencia, robustez y calidad exigidos.

Resultados de las pruebas realizadas en la iteración 'C8 - Cuentas'



6.9. Iteración C9: Partes de seguridad

Nº	Prueba	Resultado
9.01	Casos de uso 117, 118 y 119: Inicialización	✓
9.02	Caso de uso 120: Visualizar partes de seguridad	✓
9.03	Caso de uso 121: Visualizar parte de seguridad	✓
9.04	Caso de uso 122: Añadir parte de seguridad	✓
9.05	Caso de uso 123: Borrar parte de seguridad	✓
9.06	Caso de uso 124: Modificar parte de seguridad	✓
9.07	Caso de uso 125: Listar parte de seguridad	✓

La iteración C9 ha dado lugar al diseño y ejecución de las pruebas arriba citadas, obteniendo en todas ellas resultados positivos y llegando a la conclusión de que el software desarrollado en esta iteración manifiesta un nivel de calidad aceptable. La aplicación sigue mostrándose robusta, interactuando todos sus componentes de la forma deseada y proporcionando toda la funcionalidad que debía estar desarrollada en este punto del proceso de desarrollo.

Resultados de las pruebas realizadas en la iteración 'C9 - Partes de seguridad'

6.10. Iteración C10: Notas

Nº	Prueba	Resultado
10.01	Caso de uso 126: Redactar nota	✓
10.02	Caso de uso 127: Visualizar notas recibidas	✓
10.03	Caso de uso 128: Visualizar nota recibida	✓
10.04	Caso de uso 129: Borrar nota recibida	✓
10.05	Caso de uso 130: Visualizar notas enviadas	✓
10.06	Caso de uso 131: Visualizar nota enviada	✓
10.07	Caso de uso 132: Borrar nota enviada	✓

He aquí las pruebas que demuestran que el servicio de mensajería ideado para la aplicación cumple con toda la funcionalidad y calidad requerida, permitiendo ser utilizado sin ningún tipo de inconveniente.

Resultados de las pruebas realizadas en la iteración 'C10 - Notas'



6.11. Iteración C11: Base de datos

Nº	Prueba	Resultado
11.01	Caso de uso 133: Importar base de datos	✓
11.02	Caso de uso 134: Exportar base de datos	✓

Las pruebas diseñadas y ejecutadas a lo largo de la iteración 'C11 - Base de datos' muestran que los casos de uso que en ella se han implementado han sido ideados y desarrollados correctamente, de acuerdo a los requerimientos expresados por el cliente y bajo unos criterios de calidad aceptable.

Resultados de las pruebas realizadas en la iteración 'C11 - Base de datos'

6.12. Iteración C12: Diseño, personalización y configuración

Nº	Prueba	Resultado
12.01	Inicialización de la aplicación según el tipo de usuario	✓
12.02	Refresco de los datos de la aplicación	✓

La última iteración de la fase de Construcción ha sido dedicada al diseño, personalización y configuración de la aplicación con el fin de ponerla a punto para transferirla al cliente. En ella hemos realizado dos pruebas, una de inicialización de la aplicación y otra de refresco de los datos, obteniendo en ambas los resultados esperados.

Resultados de las pruebas realizadas en la iteración 'C12 - Diseño, configuración y personalización'

6.13. Prueba de aceptación

La segunda iteración de la fase de Transición acoge la mayor y más importante prueba que puede realizarse en cualquier proyecto informático: la de aceptación. Ésta tiene por objeto probar los requisitos de usuario y como procedimiento la evaluación conjunta con el cliente del producto software desarrollado, haciéndole ver que éste es el ideal para él, que le proporciona toda la funcionalidad que en él busca.

Tras haber instalado y configurado todo el sistema se realiza la prueba de todos y cada uno de los casos de uso implementados, quedando el cliente nuevamente muy satisfecho con nuestra labor.

6.14. Prueba de concurrencia

Para concluir con todo el proceso de producción del sistema y su implantación llevamos a cabo la prueba simultánea del sistema con varios usuarios de modo que conseguimos simular el funcionamiento real que tendrá el sistema en un futuro muy cercano.

Se prueban varios casos de uso paralelamente y el sistema siempre responde con corrección ante cualquier situación, lo que hace que demos por superada esta prueba.



Pamplona, a 16 de abril de 2010.

Héctor Napal Basterrica
Responsable del proyecto

Pedro Martínez Pérez
*Jefe de área de producción de
Informática Navarra S.A.*



accessManager
Desarrollo de un sistema de control de accesos a un centro

Interfaz gráfica

Fecha de entrega: 16 de abril de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Estructura del documento

2. Interfaz gráfica de usuario

- 2.1. Diálogo de identificación
- 2.2. Accesos
- 2.3. Empresas y personal
- 2.4. Tarjetas, tipos de acceso y cargos
- 2.5. Zonas, llaves y llaveros
- 2.6. Préstamos
- 2.7. Cuentas
- 2.8. Partes de seguridad
- 2.9. Notas
- 2.10. Menú 'accessManager'



1. Introducción

1.1. Propósito

Mostrar al lector los resultados tangibles que experimentará el usuario de accessManager, realizando una breve descripción de todos los componentes gráficos que constituyen la aplicación y cómo ha sido distribuida su funcionalidad entre ellos.

1.2. Audiencia

Este documento va dirigido a todas aquellas personas que en un futuro sean usuarios de la aplicación.

1.3. Estructura del documento

Interfaz gráfica resulta ser un escrito muy breve y sencillo redactado con aras a explicar de forma global la interfaz gráfica que permitirá al usuario interactuar con el sistema. Está compuesto por un único capítulo que va recorriendo todas las pestañas en que accessManager se divide y explicando una a una el fin con el que han sido diseñadas; qué funcionalidades proporcionan.



2. Interfaz gráfica de usuario

Como punto final a esta recolección de los principales artefactos obtenidos del proceso de desarrollo no podía faltar una breve exposición de la aplicación gráfica elaborada, la cual se dividirá en pestañas recogiendo en cada una funcionalidades que tratan sobre el mismo grupo de datos (datos relacionados=).

2.1. Diálogo de identificación

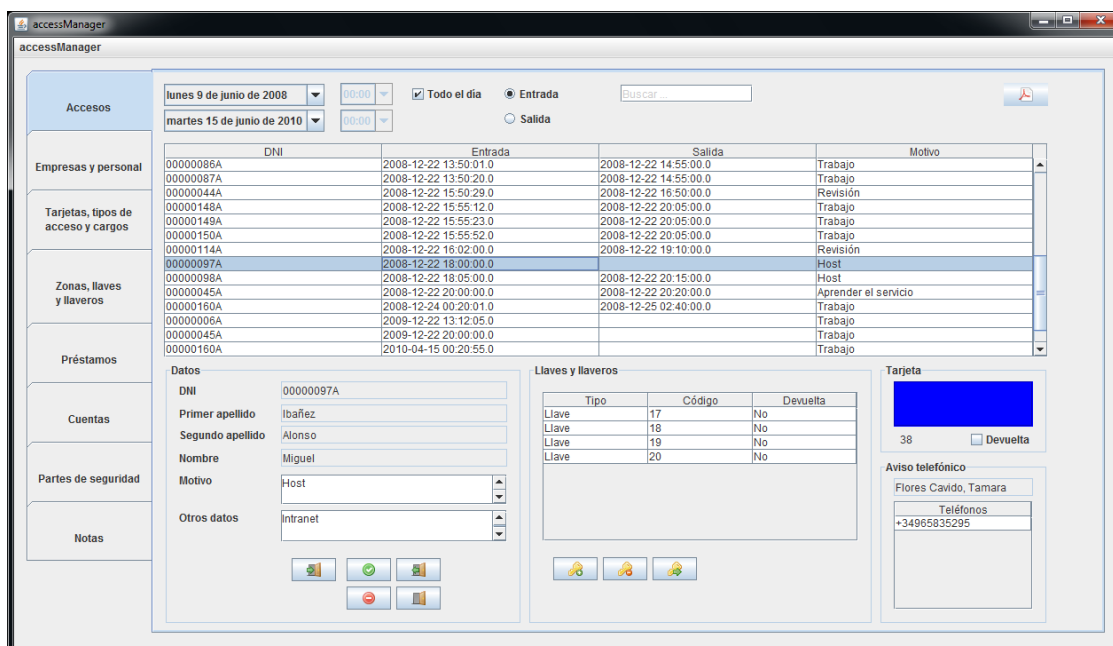
Como todo sistema que atesore información de uso restringido, se ha realizado este diálogo mediante el cual el usuario podrá acceder al sistema:



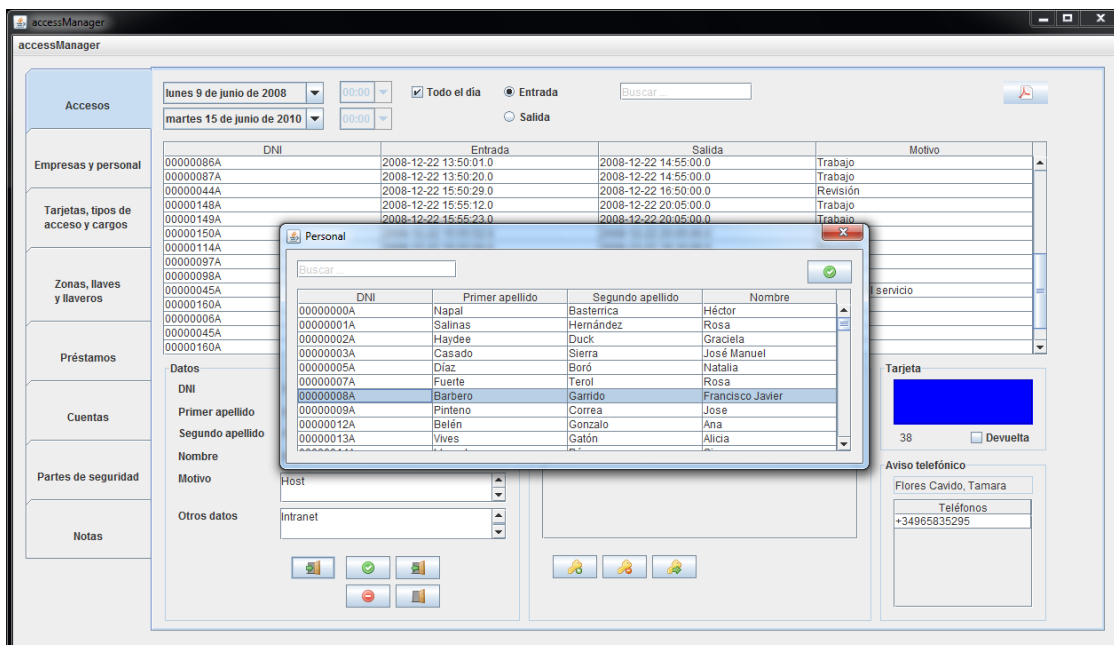
Interfaz de acceso al sistema

2.2. Accesos

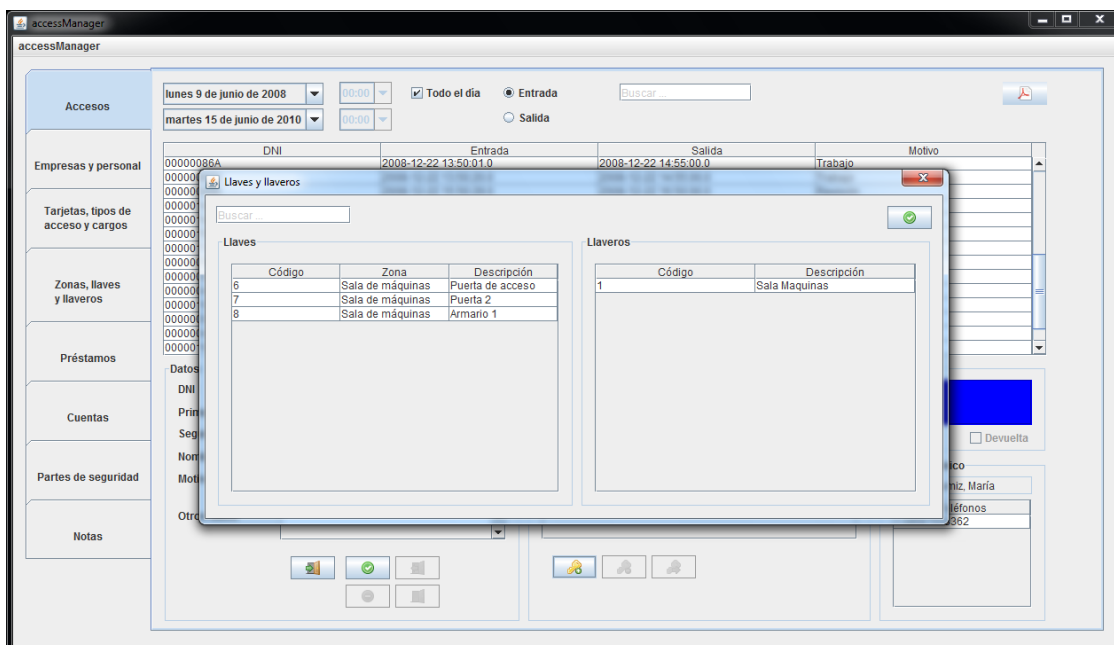
El principal propósito por el que fue diseñado accessManager es poder tratar toda la información concerniente a los accesos realizados por el personal al centro. La interfaz diseñada para la ocasión está compuesta por una sección con los parámetros que el usuario puede usar para redefinir la búsqueda sobre el total de accesos realizados almacenados, una tabla que muestra los resultados obtenidos de dicha búsqueda y un conjunto de paneles que exponen la información del acceso seleccionado en cada momento:



Interfaz de la pestaña 'Accesos'



Interfaz de selección de persona

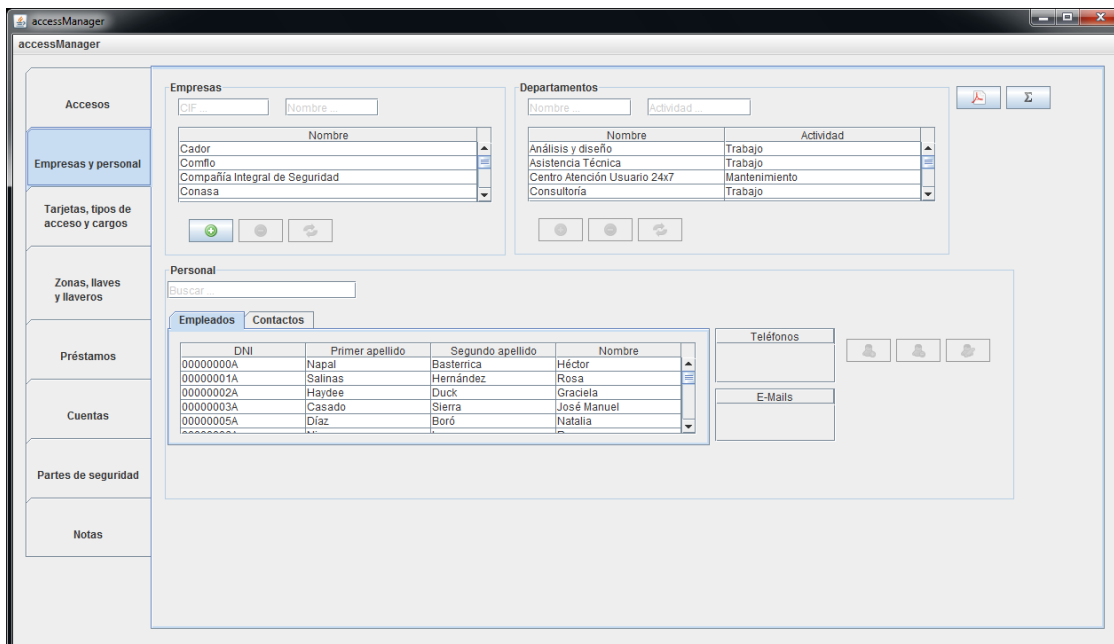


Interfaz de selección de préstamos

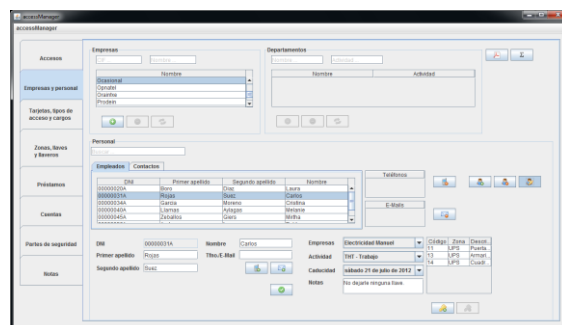
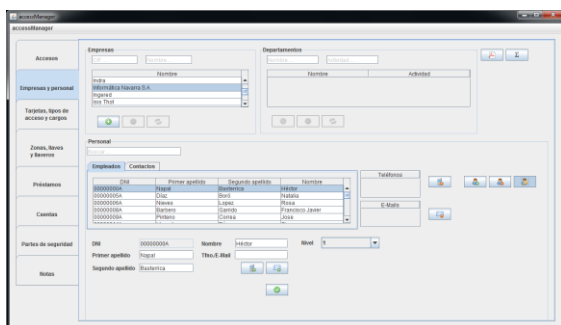
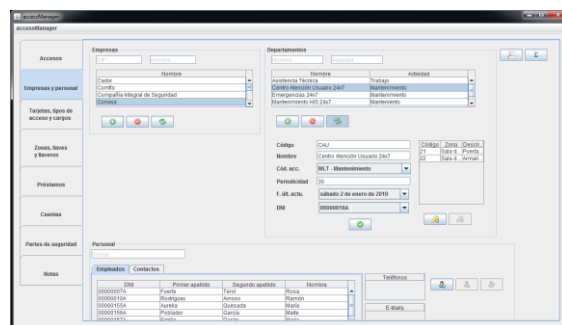
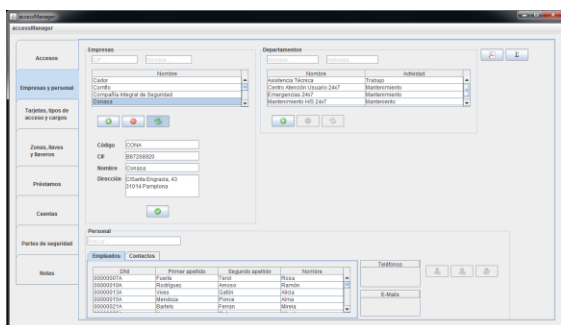


2.3. Empresas y personal

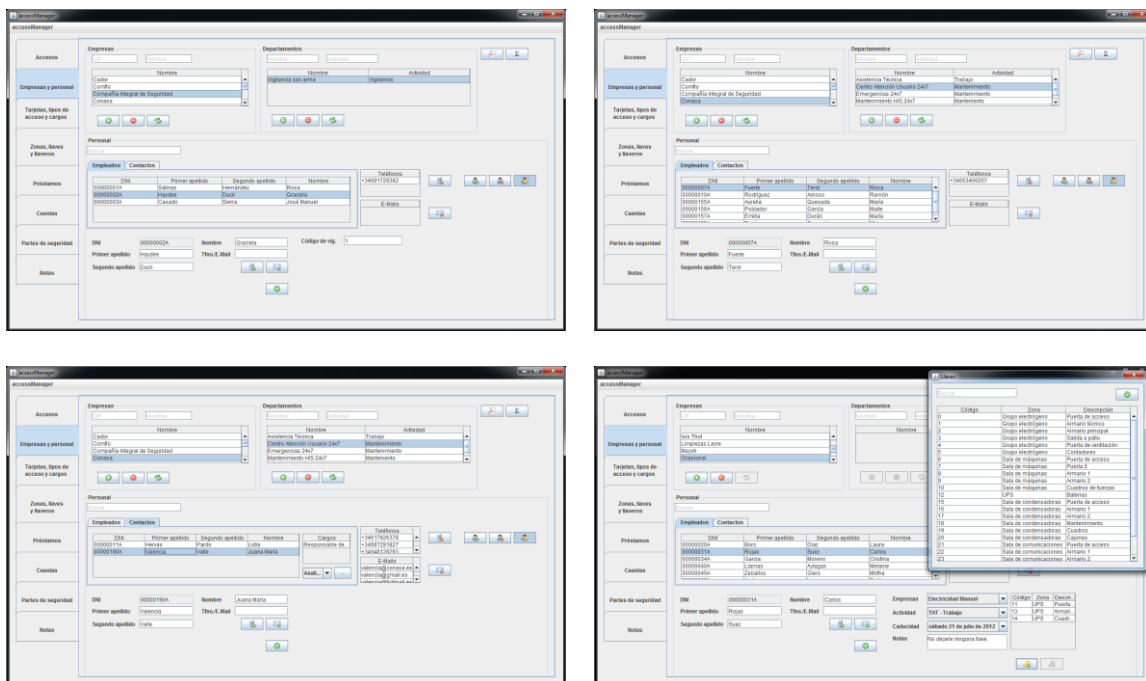
Los accesos son realizados por todo el personal que pertenece a las diversas empresas y departamentos registrados en el sistema. Para poder manipularlos creamos esta pestaña, que está compuesta por tres secciones: una para realizar todas las operaciones necesarias sobre las empresas, otra donde manipularemos los departamentos que las conforman y la inferior, donde el usuario podrá visualizar, añadir, eliminar y editar todo el personal derivado de éstas:



Interfaz inicial de la pestaña 'Empresas y personal'



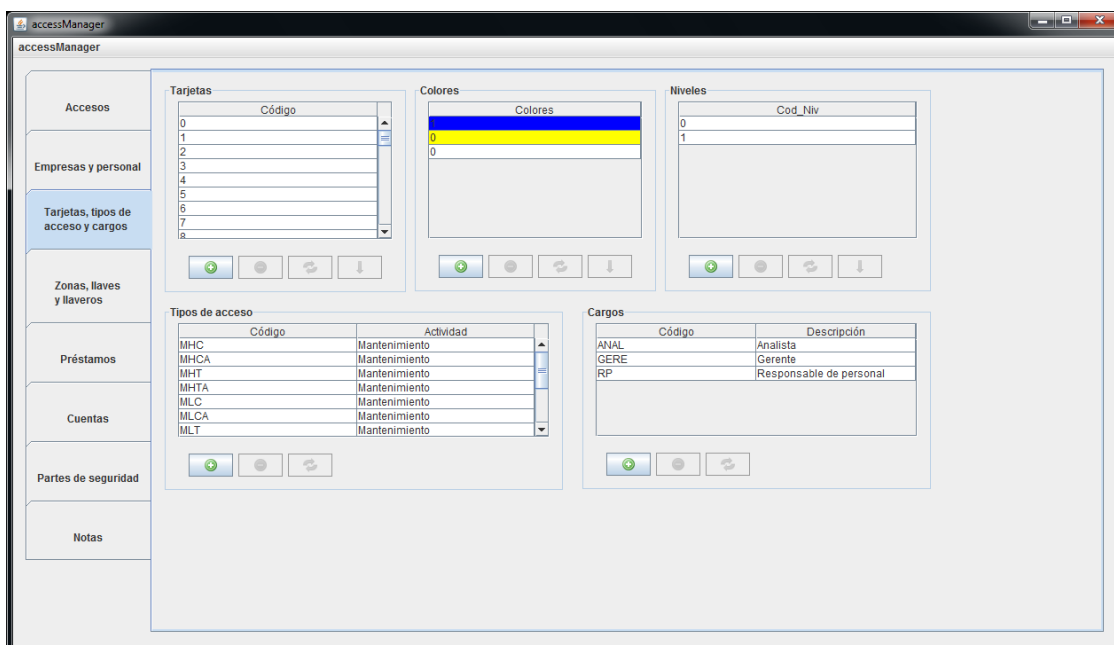
Interfaces de la pestaña 'Empresas y personal'



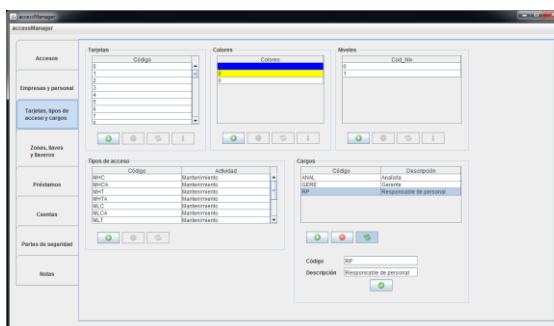
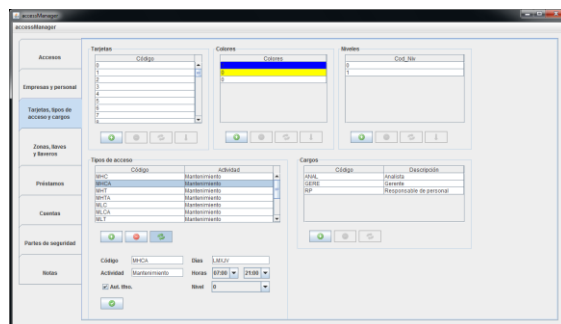
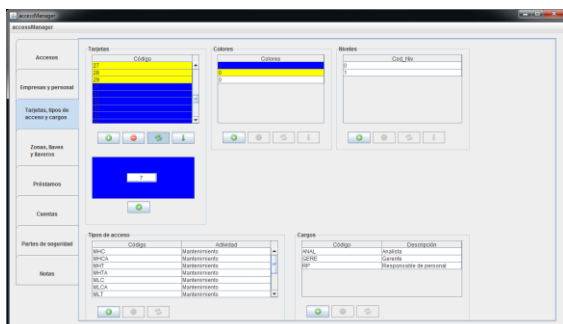
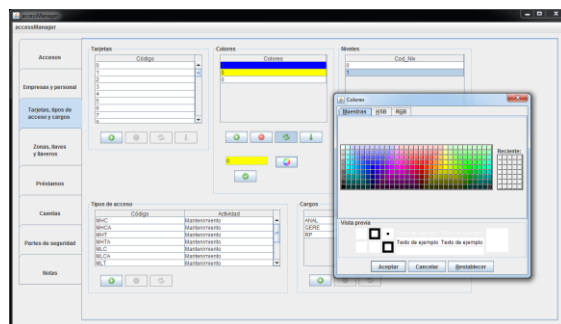
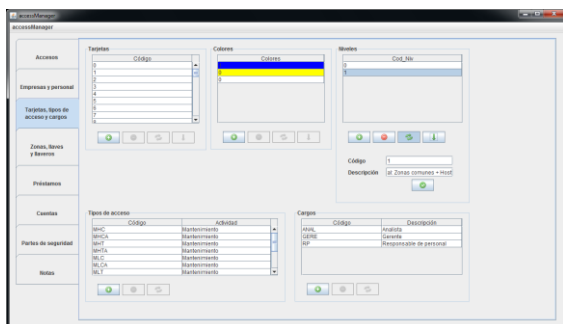
Interfases de la pestaña 'Empresas y personal'

2.4. Tarjetas, tipos de acceso y cargos

No basta solo con manipular el personal que puede acceder al centro, sino que también se debe prestar atención al resto de componentes que toman parte en el proceso de acceso: las tarjetas y tipos de acceso que proporcionan. Para poder manejar todo este conjunto de información, así como los cargos que desempeñan los contactos recogidos en la anterior pestaña, el usuario deberá dirigirse a la pestaña 'Tarjetas, tipos de acceso y cargos':



Interfaz inicial de la pestaña 'Tarjetas, tipos de acceso y cargos'

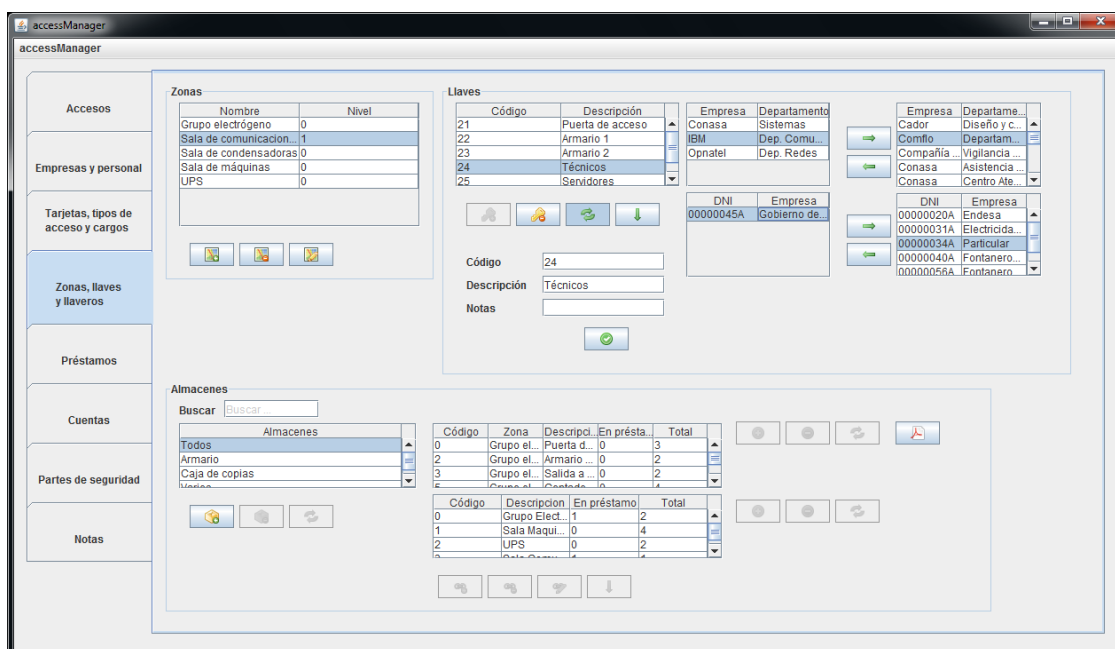


Interfaces de la pestaña 'Tarjetas, tipos de acceso y cargos'



2.5. Zonas, llaves y llaveros

La sede de Informática Navarra S.A. está dividida en zonas accesibles en función del nivel de acceso que proporcione la tarjeta en préstamo y las llaves y/o llaveros de los que se disponga. 'Zonas, llaves y llaveros' recoge toda la funcionalidad que el usuario pueda necesitar para llevar a cabo una correcta gestión de las zonas que componen el centro, las llaves asociadas a éstas junto con los departamentos y personal ocasional a los que pueden ser prestadas, los llaveros que pueden crear y cómo llaves y llaveros son almacenados en un conjunto de almacenes. El orden seguido en esta explicación es el mismo que empleamos para disponer los componentes gráficos sobre la interfaz:

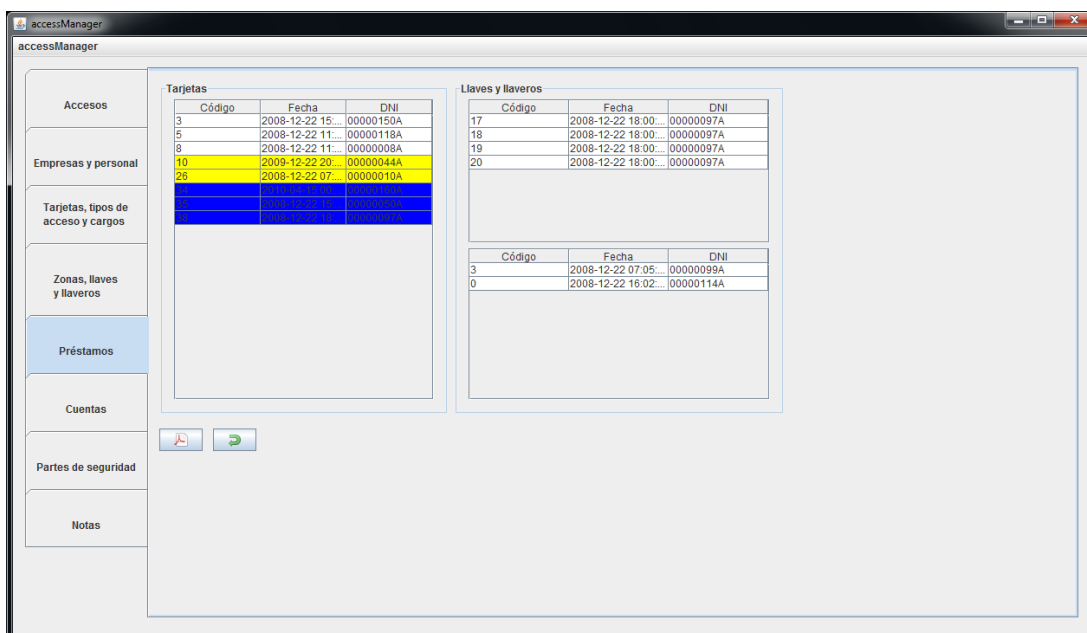


Interfaz de la pestaña 'Zonas, llaves y llaveros'



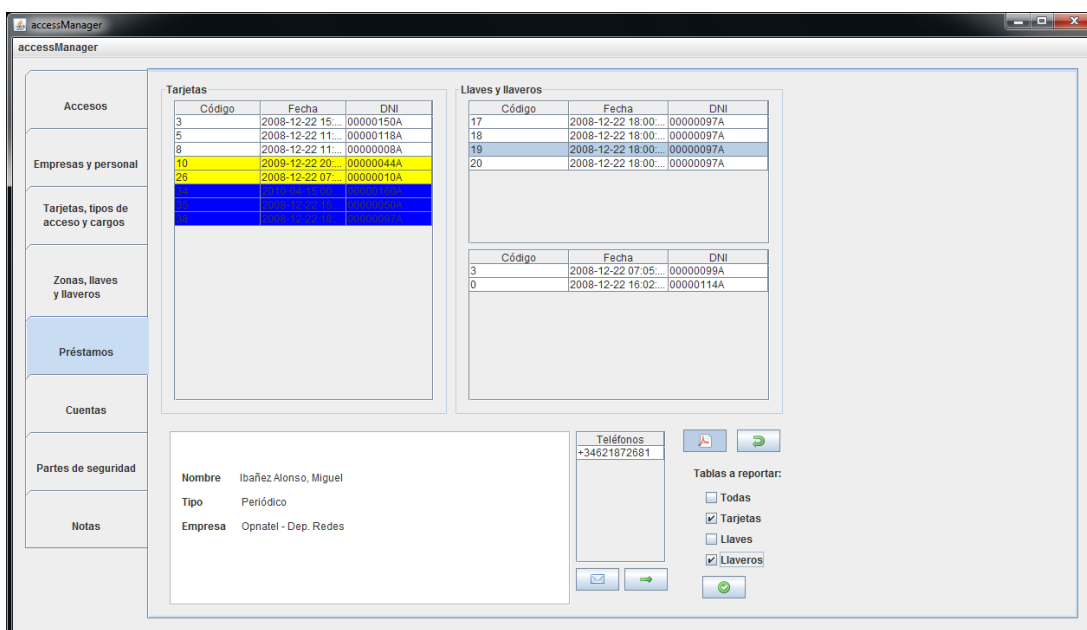
2.6. Préstamos

Parte de los préstamos que el vigilante realiza no serán devueltos por los prestatarios, lo que hace que ésta pestaña sea de gran utilidad al administrador de cara a saber qué material está prestado y requerirlo de vuelta lo antes posible. La interfaz inicial dispone de dos columnas, la primera con las tarjetas y la segunda con las llaves y llaveros prestados:



Interfaz inicial de la pestaña 'Préstamos'

Al clicar sobre cualquier préstamo la interfaz se expandirá ofreciendo los datos del prestatario y dos botones con los que marcar el préstamo como devuelto y enviar un E-Mail al prestatario recordándole sus préstamos pendientes:



Interfaz de la pestaña 'Préstamos'



2.7. Cuentas

La pestaña 'Cuentas' dispone de un carácter puramente administrativo y servirá al usuario que se encargue de la administración de la aplicación para manipular las cuentas existentes, los tipos de usuario a las que están asociadas y además visualizar quienes de ellos han accedido al sistema según unos parámetros de búsqueda que podrá redefinir a su gusto:

DNI	Tipo	Usuario	Contraseña
00000000A	AUTORIZADOR	a	*****
00000000A	ADMINISTRADOR	hnapalb	*****
00000001A	VIGILANTE	rsalinash	*****
00000002A	VIGILANTE	ghaydeed	*****
00000003A	VIGILANTE	jcasados	*****
00000018A	AUTORIZADOR	mgamiz	*****
00000028A	AUTORIZADOR	lcarmona	*****
00000043A	AUTORIZADOR	tcavido	*****
00000163A	RP	jgarciap	*****
00000168A	DD	rdacasaquiza	*****

Tipo	Descripción
ADMINISTRADOR	Administrador del sistema
AUTORIZADOR	Autorizador de departamentos
RP	Responsable de personal de de...
VIGILANTE	Vigilante del centro

Usuario	Tipo	DNI	Fecha
hnapalb	ADMINISTRADOR	00000000A	2010-06-15 10:12:12.0
hnapalb	ADMINISTRADOR	00000000A	2010-06-15 11:42:45.0
hnapalb	ADMINISTRADOR	00000000A	2010-06-15 11:43:37.0
hnapalb	ADMINISTRADOR	00000000A	2010-06-15 11:44:13.0
hnapalb	ADMINISTRADOR	00000000A	2010-06-15 11:45:21.0
hnapalb	ADMINISTRADOR	00000000A	2010-06-15 11:49:13.0
hnapalb	ADMINISTRADOR	00000000A	2010-06-15 11:50:26.0
hnapalb	ADMINISTRADOR	00000000A	2010-06-15 11:51:32.0

Interfaz de la pestaña 'Cuentas'

Teniendo en cuenta a todo el conjunto de usuarios, el menú 'accessManager' permitirá que éstos puedan modificar los datos de su cuenta a través del siguiente diálogo:

Usuario y contraseña

hnapalb

Autenticación

Contraseña actual

Nuevos datos

Contraseña

Reescribe la contraseña

Interfaz del diálogo 'Cuenta'



2.8. Partes de seguridad

Si antes los vigilantes estaban obligados a redactar sus partes de vigilancia con papel y bolígrafo, a partir de ahora lo harán desde la pestaña 'Partes de seguridad'. La parte izquierda de la misma ha sido diseñada a modo de expositor de los partes almacenados según los criterios de búsqueda definidos, mientras que la derecha ha sido confeccionada de modo que simule los partes de seguridad que hasta ahora eran redactados:

The screenshot shows the 'accessManager' application window. On the left is a sidebar with menu items: Accesos, Empresas y personal, Tarjetas, tipos de acceso y cargos, Zonas, llaves y llaveros, Préstamos, Cuentas, Partes de seguridad (highlighted), and Notas. The main area is divided into two sections. The top section contains search and filter options: 'Cod. Vig.' set to 'Todos', 'Activar fechas' checkbox, and 'Todo el día' checkbox. Below these are date and time pickers for 'martes 15 de junio de 2010'. A table lists security parts with columns for 'Fecha' and 'Hora'. The bottom section displays a detailed view of a security part for 'Código de vigilante 1' on 'viernes 16 de abril de 2010' at '19:00'. The 'Incidencias' field contains the following text:

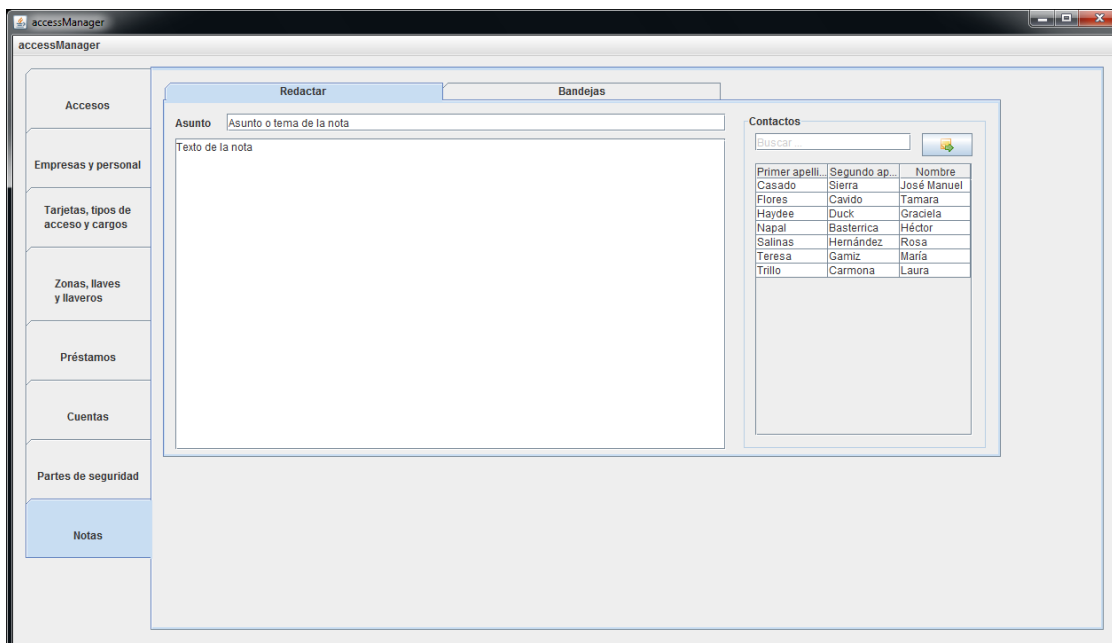
19:00 - Inicio servicio s/n aparente.
20:00 - Sonda ambiente 20,4°.
22:00 - Ronda por el centro s/n.
07:00 - Finalizo servicio s/n

Interfaz de la pestaña 'Partes de seguridad'



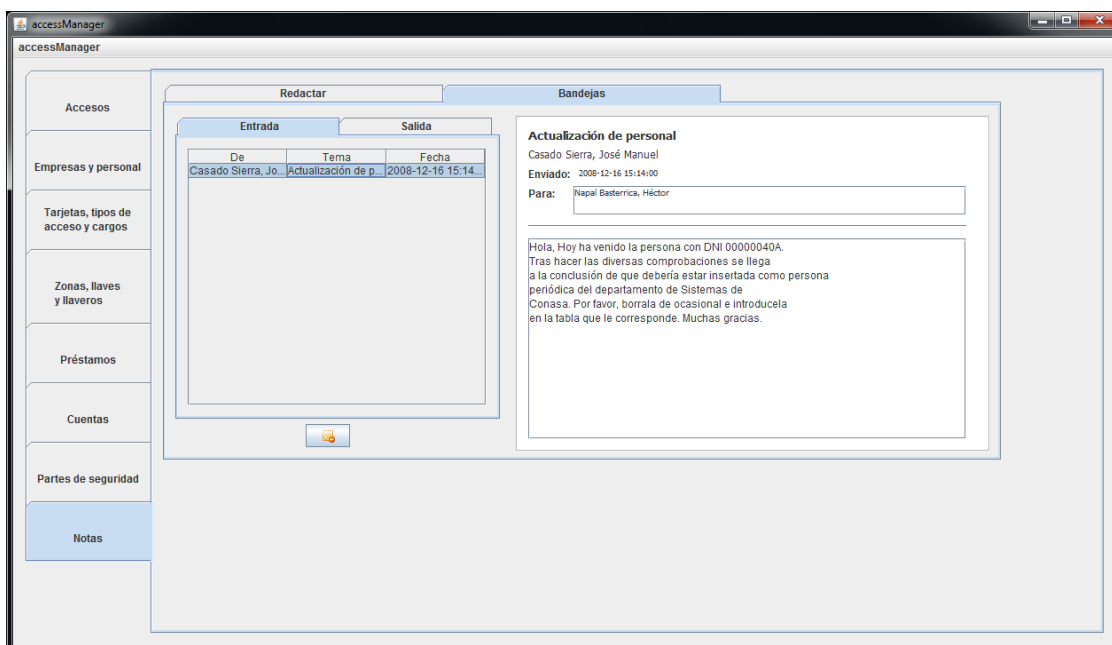
2.9. Notas

Como última pestaña, accesible por todos los usuarios salvo los responsables de personal, tenemos la de 'Notas'. Ésta ofrece un sistema simple de mensajería que simula cualquier gestor básico de correo electrónico. Está compuesta por dos pestañas, una que permitirá enviar escribir notas...



Interfaz de la pestaña 'Notas': Redactar

... y la otra que permitirá consultar las bandejas de entrada (notas recibidas) y salida (notas enviadas):



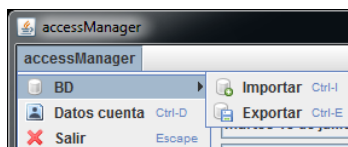
Interfaz de la pestaña 'Notas': Bandejas



2.10. Menú 'accessManager'

Compuesto por tres submenús permite:

- Importar y exportar código SQL hacia y desde la base de datos del sistema.
- Acceder al diálogo con el que modificar los datos de la cuenta (explicado en el punto 2.7).
- Ofrecer otra vía de salida de la aplicación.



Interfaz del menú 'accessManager'

Pamplona, a 16 de abril de 2010.


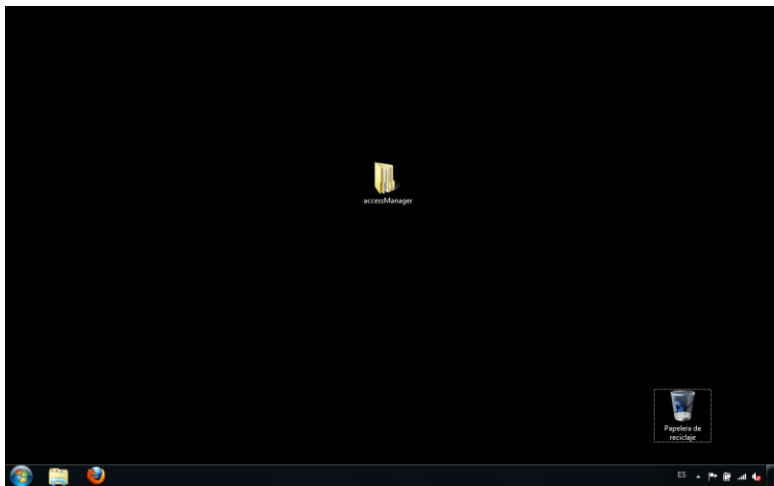
Héctor Napal Basterrica
Responsable del proyecto

Capítulo 4


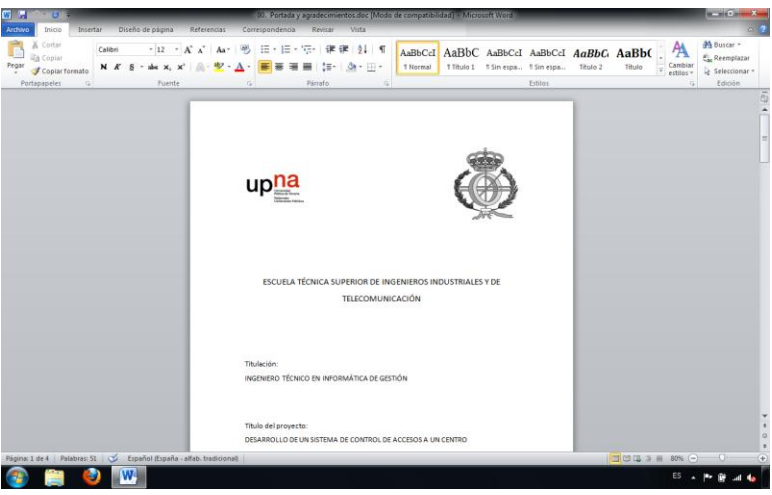

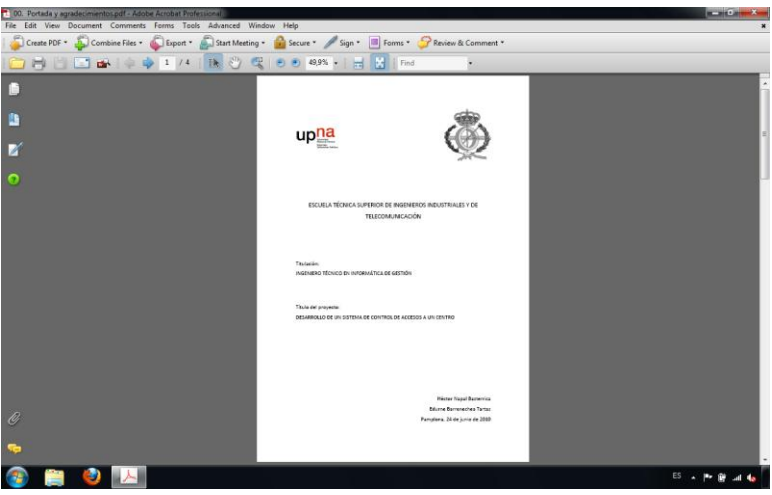
Herramientas

4.1. El entorno de desarrollo


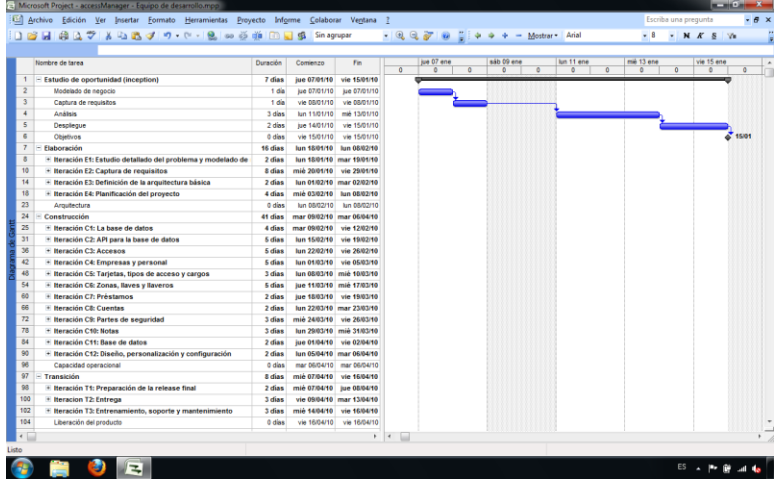

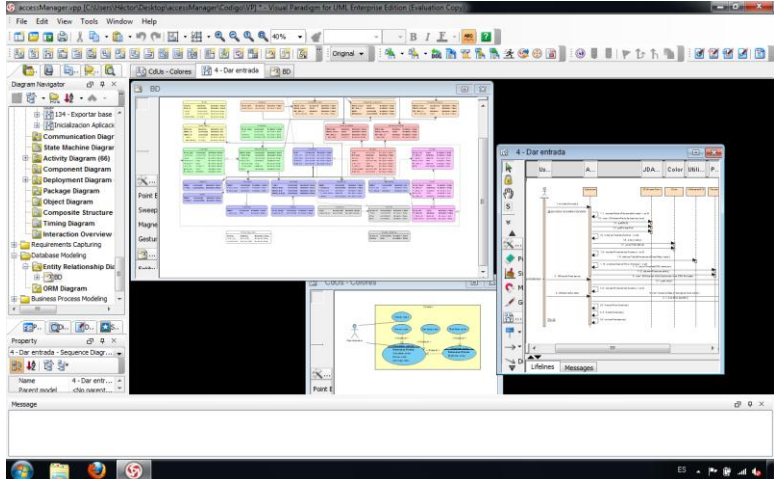
La realización de éste proyecto no podría haber sido posible sin el empleo de las herramientas software que describimos a continuación:

APLICACIONES	
CAMPO	DESCRIPCIÓN
Nombre	Microsoft Windows 
Versión	7
Desarrollador	Microsoft
Descripción	Última versión del sistema operativo para uso en PC desarrollado por Microsoft.
URL	http://windows.microsoft.com/es-ES/windows-7?os=win7
Imagen	


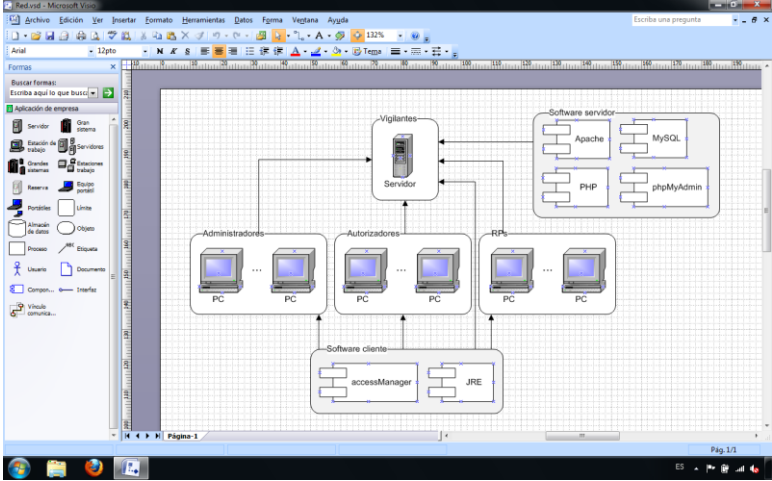

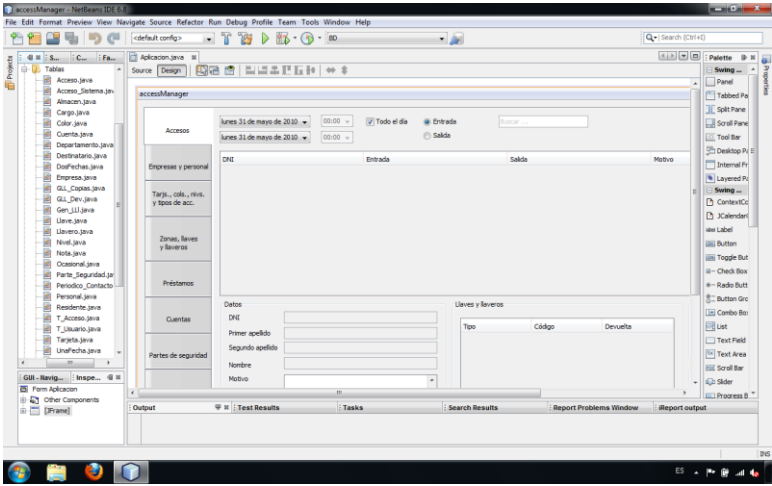
Aplicaciones software empleadas en el desarrollo

APLICACIONES	
CAMPO	DESCRIPCIÓN
Nombre	Microsoft Word 
Versión	2010
Desarrollador	Microsoft Corporation
Descripción	Conocido procesador de textos de la compañía de Redmond.
URL	http://emea.microsoftstore.com/es/es-ES/Microsoft/Office/Suites-2010
Imagen	
Nombre	Adobe Acrobat Professional 
Versión	8
Desarrollador	Adobe Systems
Descripción	Programa informático diseñado para visualizar, crear, modificar y controlar archivos con el formato PDF.
URL	http://www.adobe.com/es/products/acrobat/?promoid=BPBHY
Imagen	


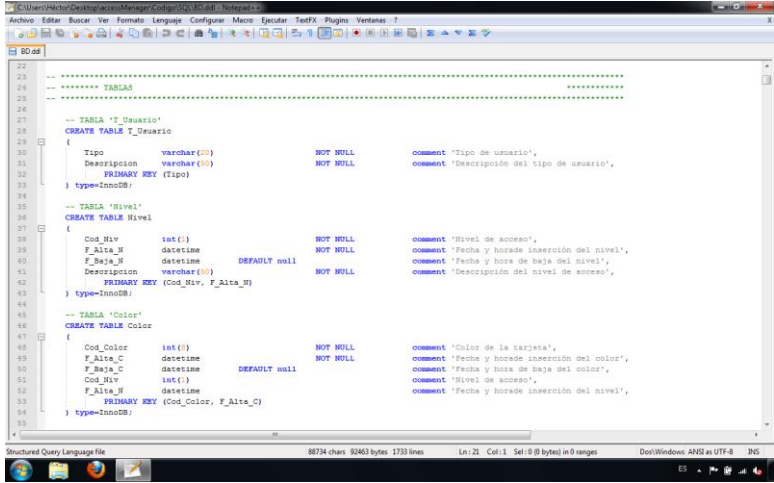

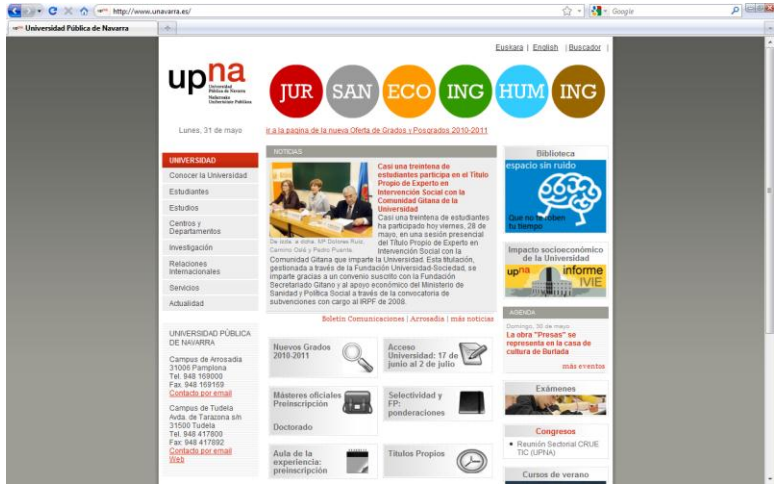
Aplicaciones software empleadas en el desarrollo

APLICACIONES	
CAMPO	DESCRIPCIÓN
Nombre	Microsoft Project 
Versión	2007
Desarrollador	Microsoft Corporation
Descripción	Software de administración de proyectos que aplica los procedimientos descritos en el <i>Management Body of Knowledge (PMBok)</i> del <i>Project Management Institute (PMI)</i> .
URL	http://office.microsoft.com/es-es/project/FX100487773082.aspx
Imagen	
Nombre	Visual Paradigm Suite 
Versión	4.2
Desarrollador	Visual Paradigm International
Descripción	Aplicación de diagramado que permite analizar y diseñar un amplio rango de diagramas circunscritos al desarrollo de software.
URL	http://www.visual-paradigm.com/
Imagen	


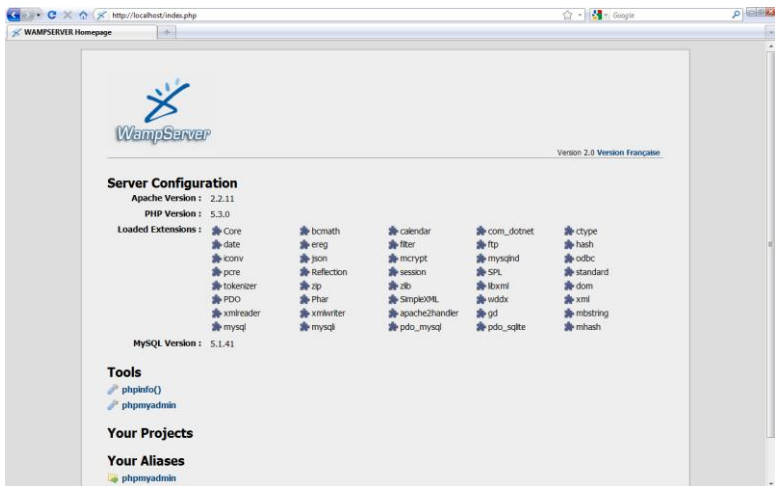
Aplicaciones software empleadas en el desarrollo

APLICACIONES	
CAMPO	DESCRIPCIÓN
Nombre	Microsoft Visio 
Versión	2007, Standard
Desarrollador	Microsoft Corporation
Descripción	Software de dibujo vectorial que permite realizar diagramas de bases de datos, de flujo de programas, UML,...
URL	http://emea.microsoftstore.com/es/es-ES/Microsoft/Office-Visio-Standard-2007
Imagen	
Nombre	NetBeans IDE 
Versión	6.8
Desarrollador	Oracle Corporation
Descripción	Entorno de desarrollo integrado fundado y patrocinado por Oracle Corporation que permite construir aplicaciones en una amplia variedad de lenguajes, entre ellos Java.
URL	http://www.netbeans.org/
Imagen	




Aplicaciones software empleadas en el desarrollo

APLICACIONES	
CAMPO	DESCRIPCIÓN
Nombre	NotePad++ 
Versión	5.6.6
Desarrollador	NotePad++ Project
Descripción	Editor de código fuente basado en el componente de edición Scintilla. Está escrito en C++ utilizando directamente la API de win32 y STL y se distribuye bajo los términos de la Licencia Pública General de GNU.
URL	http://notepad-plus.sourceforge.net/es/site.htm
Imagen	
Nombre	Mozilla Firefox 
Versión	3.6
Desarrollador	Fundación Mozilla
Descripción	Navegador web libre multiplataforma descendiente de Mozilla Application Suite, desarrollado por la Corporación Mozilla, la Fundación Mozilla y un gran número de voluntarios externos.
URL	http://www.mozilla-europe.org/es/firefox/
Imagen	

Aplicaciones software empleadas en el desarrollo

APLICACIONES	
CAMPO	DESCRIPCIÓN
Nombre	WAMPServer 
Versión	2.0i
Desarrollador	Proyecto WAMP
Descripción	<p>Aplicación que crea un sistema de infraestructura de Internet (“multiservidor”) con las siguientes herramientas:</p> <ul style="list-style-type: none"> • Apache (servidor web). • MySQL (gestor de bases de datos). • PHP (módulo de visualización de páginas en formato PHP). • phpMyAdmin (administrador de MySQL vía web).
URL	http://www.wampserver.com/en/
Imagen	

Aplicaciones software empleadas en el desarrollo

LIBRERÍAS JAVA	
CAMPO	DESCRIPCIÓN
Nombre	Connector/J 
Versión	5.1.12
Desarrollador	MySQL AB
Descripción	Adaptador encargado de convertir las peticiones desde el programa Java a un protocolo que pueda entender MySQL.
URL	http://www.mysql.com/downloads/connector/j/
Nombre	iText 
Versión	5.0.2
Desarrollador	Bruno Lowagie, Paulo Soares y otros
Descripción	Biblioteca Open Source desarrollada para permitir la creación y manipulación de archivos PDF, RTF y HTML en Java.
URL	http://itextpdf.com/
Nombre	JavaMail 
Versión	1.4.3
Desarrollador	Oracle Corporation
Descripción	Paquete desarrollado para facilitar el envío y recepción de E-Mail desde código java.
URL	http://java.sun.com/products/javamail/

Librerías Java empleadas en el desarrollo

4.2. Software ubicado en el sistema

El sistema se compone de dos capas en las que el servidor deberá disponer de la aplicación WAMPServer instalada y configurada tal y como se explicó en el artefacto T2 – *Entrega*, y las máquinas clientes deberán disponer de la aplicación accessManager y el Java Runtime Environment:

APLICACIONES	
CAMPO	DESCRIPCIÓN
Nombre	Java Runtime Environment (JRE)
Versión	6.20
Desarrollador	Oracle Corporation
Descripción	Conjunto de utilidades (entre las que se encuentra la Java Virtual machine) que permite la ejecución de programas Java.
URL	http://www.java.com/es/

Aplicaciones software ubicadas en las máquinas clientes

Capítulo 5

Bibliografía

5.1. Libros y artículos de revistas

- [1] F. Alonso, L. Martínez y F. J. Segovia, *Introducción a la ingeniería del software – Modelos de desarrollo de programas*, 1 ed.: Delta Publicaciones, 2005.
- [2] P. Kruchten, *Architectural Blueprints - The 4+1 View Model of Software Architecture*, IEEE Software - Institute of Electrical and Electronics Engineers, pp. 42-50, Noviembre 1995.
- [3] S. Birnam, *Java 2 Distribuido – Desarrollo de bases de datos*, 1 ed.: Pearson Educación, 2003.
- [4] A. Silberschatz, H. F. Korth y S. Sudarshan, *Database Systems Concepts*, 4 ed.: McGraw-Hill Higher Education, 2001.
- [5] B. Shuwartz, *MySQL Avanzado*, 2 ed.: Anaya Multimedia, 2009.
- [6] C. S. Horstmann y G. Cornell, *Core Java 1: Volume I - Fundamentals*, 8 ed.: Prentice Hall PTR, 2007.
- [7] C. S. Horstmann y G. Cornell, *Core Java 2: Volume II - Advanced Features*, 5 ed.: Prentice Hall PTR, 2001.
- [8] H. Napal, *Apuntes de la asignatura ‘Gestión de bases de datos’ (Código 41209)*, 2005.
- [9] H. Napal, *Práctica 1: Instalación y configuración de las aplicaciones en Apuntes de la asignatura ‘Laboratorio de interfaces de redes’ (Código 41311)*, 2007.

5.2. Sitios web

- [1] E. Barrenechea, "Inteligencia artificial, planificación y gestión de proyectos informáticos y gestión de bases de datos" Pamplona, 2010, pp. Página web de la profesora Edurne Barrenechea Tartas, profesora titular de la Universidad Pública de Navarra en el área de ‘Ciencias de la computación e inteligencia artificial’, donde se puede encontrar una amplia gama de apuntes acerca de la gestión de proyectos informáticos y el diseño y gestión de bases de datos.
[<http://www.ayc.unavarra.es/edurneb/>]
- [2] M. V. Mohedano, "Fundamentos de informática, ficheros y bases de datos y laboratorio de bases de datos" Pamplona, 2010, pp. Página web de la profesora M^a Victoria Mohedano Salillas, profesora titular de la Universidad Pública de Navarra en el área de ‘Ciencias de la computación e inteligencia artificial’, que pone a disposición del interesado un buena variedad de apuntes relacionados con el estudio de las bases de datos, y que pueden ser un buen complemento a los existentes en la página web anteriormente citada.
[<http://www.ayc.unavarra.es/vmohedano/>]

- [3] O. Corporation, "MySQL official web page" 2010,pp. Se trata de otra de las páginas de Oracle Corporation en la que albergan todo el contenido referente a MySQL. Aquí puede encontrarse todo lo que se desee acerca de este potente SGBD firmado bajo licencia GPL.
[<http://www.mysql.com>]
- [4] O. Corporation, "Developer resources for Java Technology" 2010, pp. Página web de Oracle Corporation en la que reside variada información de este popular y extendido lenguaje de programación, además de muchas aplicaciones para descargar con las que poder interactuar y programar en Java.
[<http://java.sun.com>]
- [5] Java API, "Java 2 Platform Standard Ed 5.0" 2010, p. Sitio web que aloja toda la documentación relativa a la API estándar de Java empleada en el desarrollo del proyecto.
[<http://java.sun.com/j2se/1.5.0/docs/api/>]
- [6] JavaMail API, "JavaMail API documentation" 2010, p. Sitio web con toda la documentación relativa a la API del paquete JavaMail utilizado para el envío de E-Mails desde la aplicación.
[<http://java.sun.com/products/javamail/javadocs/index.html>]
- [7] iText API, "iText API documentation" 2010, p. Sitio web que da soporte a toda la documentación del paquete iText empleado en este proyecto para la creación de informes en formato.
[<http://api.itextpdf.com/>]
- [8] Chuidiang, "Ejemplos Java y C/Linux" 2010, p. Sitio web que alberga una amplia gama de ejemplos Java..
[<http://www.chuidiang.com/>]
- [9] Wikipedia, "Wikipedia web page" 2010, p. Sitio web de la enciclopedia libre y políglota más utilizada del mundo. Es actualmente la mayor y más popular obra de consulta en Internet, y fuente de una gran cantidad de información y conceptos que han sido consultados en la elaboración de la presente memoria.
[<http://es.wikipedia.org/wiki/Wikipedia:Portada>]
- [10] Sourceforge, "Sourceforge web page" 2010, p. Sitio web de la famosa central de desarrollos de software que controla y gestiona varios proyectos de software libre y actúa como un interesante y recomendable repositorio de código fuente.
[<http://sourceforge.net/>]



ANEXO - PROCESO DE DESARROLLO DE LA APLICACIÓN

Capítulo 1

Elaboración

1.1. Objetivos de la fase

Segunda fase del proceso de desarrollo, en la que ya hemos cubierto la etapa de Estudio de oportunidad (inception) y nos disponemos a afrontar la de Elaboración. En ella atenderemos las actividades básicas que nos proporcionarán unos cimientos sobre los que sustentar el proyecto, teniendo como objetivos:

- Realizar un documento de *Estudio detallado del problema software*, que profundice y especifique el problema con un nivel de detalle importante.
- Acometer un breve modelado de negocio que permita representar y entender el contexto de la empresa como paso previo a la especificación de la mayor parte de los casos de uso.
- Definir una arquitectura básica que dé respuesta al problema planteado.
- Planificar temporalmente el proyecto de una forma detallada según recursos disponibles, estableciendo un plan de actividades que permita organizar y realizar la supervisión del proyecto.
- Iniciar el *Plan de pruebas*, que recogerá todas las pruebas que se realizarán junto con los resultados obtenidos en ellas, así como las evaluaciones y conclusiones derivadas. Es verdad que aún no se ha desarrollado ningún tipo de componente software que deba ser probado, pero una vez que se dispone de información suficiente acerca del proyecto podemos comenzar a esbozar este documento e ir refinando y completándolo a lo largo de todo el desarrollo. Concretamente, en ésta etapa, identificaremos las fases y tipos de pruebas a realizar.

Como vemos, ésta es una etapa de “puesta en marcha” en la que se especifica qué se quiere construir, determinando la línea base de la arquitectura.

Se propone realizar cuatro iteraciones en las que se obtendremos los siguientes documentos:

- *Estudio detallado del problema software*
- *Especificación del modelo de negocio*
- *Especificación inicial de los casos de uso*
- *Arquitectura preliminar*
- *Planificación temporal y costes detallados del proyecto*
- *Plan de pruebas*

Se debe tener muy en cuenta que el proceso *RUP* genera artefactos muy tempranamente, pero éstos no serán completados hasta etapas tardías del desarrollo. Para poder disponer de los documentos finales diríjase al capítulo *Principales artefactos* de la presente memoria.

1.2. Iteración E1: Estudio detallado del problema y modelado de negocio

Las dos reuniones previas que permitieron obtener la información necesaria para realizar un análisis general del problema y aceptar posteriormente el desarrollo del mismo, dan paso a otra reunión que nos ha permitido definir con mucha mayor precisión el ámbito del problema. Dicho estudio se recoge en el documento *Estudio detallado del problema software* que acompaña al presente capítulo.

Como paso previo a la captura de requisitos, también realizamos un pequeño documento que recoge un modelado de negocio inicial, que servirá para comprender los procesos de negocio de la organización. Invitamos al lector a que ojee el artefacto *Especificación del modelo de negocio* y adquiera así un conocimiento más profundo y sólido de la organización y los procesos que en ella se desarrollan.

1.3. Iteración E2: Captura de requisitos

Tras el estudio de la organización procedemos con un paso de vital importancia en todo proceso de desarrollo, y en especial en *RUP*: el modelado de los casos de uso a través de la especificación de los mismos. Al estar todo el proceso de desarrollo controlado/dirigido por los casos de uso, adquiere una importancia máxima realizar esta etapa con la mayor precisión y completitud posible. Esto no quita para que se trate de un modelo inicial, en el que es verdad que se especifican la mayor parte de los casos de uso, pero no es menos verdad que será refinado y extendido a lo largo de todo el proceso iterativo.

Toda la información relativa a la captura de requisitos queda recogida en *Especificación inicial de los casos de uso*.

1.4. Iteración E3: Definición de la arquitectura básica

Con la funcionalidad y objetivos del sistema bien definidos, debemos marcar una arquitectura básica entorno a la cual podamos construir el sistema. No será definitiva, pues deberá ser refinada en la fase de construcción hasta obtener el sistema final, pero sí que se deberá definir un escenario de funcionamiento. Para más detalles consultar el antepenúltimo documento de este capítulo.

1.5. Iteración E4: Planificación del proyecto

Una vez tenemos asentadas las bases del proyecto, sabiendo qué se debe construir y con un esbozo del cómo, llega la hora de establecer un calendario en el que distribuyamos todo el proceso en el tiempo. Se identificarán procesos, actividades y tareas, junto con una estimación de tiempos que servirán para que el desarrollador realice, supervise y controle el proyecto, realizando los reajustes oportunos a medida que éste se desarrolle. Además, se establecerá un hito por fase con todos los objetivos que deben cumplirse al llegar a éstos.

Para finalizar, y una vez que se dispone de información precisa del proyecto, se realiza una estimación detallada de costes monetarios que es suministrada al cliente.

Toda la información de esta iteración está disponible en el penúltimo artefacto, de nombre *Planificación temporal y costes detallados del proyecto*.



accessManager
Desarrollo de un sistema de control de accesos a un centro

Estudio detallado del problema software

Fecha de entrega: 19 de enero de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Definiciones, acrónimos y abreviaciones
- 1.4. Estructura del documento

2. El día a día en Informática Navarra S.A.

- 2.1. El centro
- 2.2. La entrada y salida
- 2.3. La seguridad
- 2.4. La comunicación

3. Descripción global del producto

- 3.1. Perspectiva del producto
- 3.2. Beneficios

4. Usuarios afectados por el proyecto

- 4.1. Cliente (co-desarrollador)
- 4.2. Usuarios



1. Introducción

1.1. Propósito

El objeto de este documento es recoger, analizar y profundizar en el dominio del problema software, identificando el papel del software dentro de éste y describiendo los objetivos y necesidades de alto nivel, así como los usuarios afectados.

1.2. Audiencia

Este es un documento interno, careciente de interés para el cliente y que surge de la aceptación del estudio inicial y de una posterior reunión mantenida con éste.

1.3. Definiciones, acrónimos y abreviaciones

- *RUP* Rational Unified Process Proceso Unificado Racional

1.4. Estructura del documento

En el siguiente apartado realizamos una descripción de la sede de Informática Navarra S.A., analizando detalladamente todo el proceso que se produce en el acceso al centro y constatando la ineficacia del mismo.
A continuación hablamos de la perspectiva global del producto, así como los beneficios que supondrá, para finalizar enunciando qué usuarios serán afectados por la implantación del nuevo sistema y qué roles adoptarán en éste.

2. El día a día en Informática Navarra S.A.

2.1. El centro

Informática Navarra S.A. dispone de un centro de operaciones en Pamplona dotado de información y equipamiento de alta importancia y coste, lo que hace que se deba llevar un control de accesos de todo el personal que trabaja en él.

Éste acoge el trabajo diario de su plantilla junto con el de varias empresas más. Éstas se dividen en departamentos que poseen un conjunto de trabajadores destinados al centro. Éstos, junto con los primeros y el personal ocasional (aquel que accede al centro de una forma puntual para algún fin específico) forman todo el personal que desempeña su labor en el centro, amén de los vigilantes de seguridad. En el primer grupo debemos distinguir al jefe de área de producción, que es el encargado de que las políticas de acceso y vigilancia se desarrollen con respecto a lo establecido, y a un grupo de personas afines a él a las que ha confiado la tarea de ayudarle en la gestión de todas las empresas y su personal. Su labor es la de hacerse cargo de algunas de las



empresas y departamentos, siendo necesario que sean avisadas en algunos de los accesos que realizan sus trabajadores.

El centro posee de varias zonas encuadradas en diferentes niveles de acceso, de modo que un trabajador podrá acceder a unas u otras en función del nivel de acceso que le otorgue la tarjeta asignada en su acceso al centro. Las tarjetas están coloreadas en función del nivel de acceso que proveen, pudiendo corresponder varias a un mismo nivel.

El acceso permitido a una zona no implica que se pueda acceder a toda ella, quedando partes (p.e.: despachos) y elementos de ésta (p.e.: armarios) pendientes de que el trabajador disponga de la llave que permita su apertura. Todas las llaves y llaveros necesarios pueden ser obtenidas mediante préstamo por parte del vigilante de seguridad.

2.2. La entrada y salida

El acceso a la sede consta de dos puertas, una de apertura manual y otra automática, que sólo puede ser desbloqueada si alguno de los lectores de tarjetas que la acompaña (por uno y otro lado, para entrar y para salir) ha reconocido previamente una tarjeta como válida. En éste lugar es donde se realiza el control de accesos al centro, cuyos principales escenarios quedan resumidos en la siguiente tabla:

INSA	HORA	TFNO	AUTO	TARJ	LLAV	Entrada	Salida
NO	NO	X	X	X	X	No puede acceder	---
NO	SÍ	NO	X	SÍ	SÍ	Prestar tarjeta y llaves	Devolver <tarjeta> y <llaves>
NO	SÍ	NO	X	SÍ	NO	Prestar tarjeta	<Devolver tarjeta>
NO	SÍ	SÍ	NO	X	X	No puede acceder	---
NO	SÍ	SÍ	SÍ	SÍ	NO	Prestar tarjeta	<Devolver tarjeta>
NO	SÍ	SÍ	SÍ	SÍ	SÍ	Prestar tarjeta y llaves	Devolver <tarjeta> y <llaves>
SÍ	X	-	-	NO	NO	Puede acceder	---
SÍ	X	-	-	NO	SÍ	Prestar llaves	<Devolver llaves>
SÍ	NO	-	-	SÍ	X	No puede acceder	---
SÍ	SÍ	-	-	SÍ	NO	Prestar tarjeta	<Devolver tarjeta>
SÍ	SÍ	-	-	SÍ	SÍ	Prestar tarjeta y llaves	Devolver <tarjeta> y <llaves>

Escenarios de acceso posibles

INSA	Trabajador de 'Informática Navarra S.A.'	/	Resto de trabajadores.
HORA	Acceso en horario de trabajo.	/	Acceso fuera del horario de trabajo.
TFNO	Aviso telefónico requerido.	/	Aviso telefónico no requerido.
AUTO	Acceso autorizado por el responsable.	/	Acceso no autorizado.
TARJ	El trabajador necesita una tarjeta.	/	El trabajador no necesita tarjeta.
LLAV	El trabajador necesita alguna llave.	/	El trabajador no necesita llaves.

Nota: Todo préstamo debe ser registrado, así como todo acceso que conlleve préstamo de tarjeta.



El proceso es el siguiente:

1. Entradas: Una persona se dispone a acceder al centro.
 - 1.1. Si pertenece a Informática Navarra S.A., lo normal es que el trabajador porte consigo su tarjeta personal que le permite acceder al centro, aunque puede darse el caso de que no la lleve consigo debido a un descuido, pérdida temporal, ...
 - 1.1.1. Si ha traído consigo su tarjeta no tiene más que pasar la tarjeta por el lector y acceder al centro sin necesidad de que tal acceso quede registrado.
 - 1.1.2. En el caso de que no disponga de tarjeta:
 - Se dirige al vigilante de seguridad identificándose mediante su DNI y explicándole la situación.
 - El vigilante busca el nivel de acceso que dicho trabajador posee en el centro, comprueba que se encuentra dentro de su horario laboral en el centro y le otorga una tarjeta que proporcione el nivel que posee, registrando el acceso y motivo del préstamo de la tarjeta.
 - El trabajador accede al centro mediante la tarjeta que se le ha proporcionado.
 - 1.2. Si es un trabajador no perteneciente a Informática Navarra S.A., requiere siempre una tarjeta para poder acceder al centro:
 - Se dirige al vigilante de seguridad identificándose mediante su DNI y explicándole el motivo de su visita, haciéndole saber si se trata de un trabajador de alguna empresa o si es de tipo ocasional.
 - El vigilante, asistido por dos listas (una de personal de empresas y otra de personal ocasional), busca al trabajador en la que corresponda, obteniendo así el nivel de acceso que posee, si se encuentra dentro de su horario laboral en el centro y si es necesaria una autorización telefónica previa para su acceso (en el caso del personal ocasional siempre lo será).
 - 1.2.1. Si el acceso requiere aviso telefónico:
 - El vigilante llama a la persona responsable del departamento del que procede el empleado o a la que ha autorizado el acceso del trabajador ocasional.
 - 1.2.1.1. Si el responsable da el visto bueno, se continúa con el flujo de sucesos que siguen.
 - 1.2.1.2. Si el responsable no da el visto bueno, se le deniega el acceso.



- El vigilante asigna una tarjeta que se corresponda con el nivel de acceso que posee el trabajador y registra el acceso.
- El trabajador pasa la tarjeta por el lector de la puerta automática, la abre y accede al centro.

De forma complementaria a todos los accesos permitidos, el trabajador puede necesitar de alguna llave o llavero para llevar a cabo su labor. En tal caso se dirige al vigilante de seguridad para que le preste lo necesario. Éste se asegurará de que el trabajador pueda poseer dichas llaves o llaveros, apuntando el préstamo y proporcionándoselos.

Notas:

- *El nivel de acceso otorgado a todos los trabajadores de un departamento será el mismo, mientras que el otorgado a los de tipo ocasional será específico de cada trabajador.*
- *Las listas con todo el personal autorizado a acceder al centro, así como sus respectivos niveles de acceso, son confeccionadas en la actualidad por el jefe de área de producción de Informática Navarra S.A.*
- *Puede ocurrir que un trabajador salga del edificio olvidando devolver alguna llave, llavero o tarjeta.*
Al final del día el vigilante realiza un repaso de todos los accesos comprobando que todos constan de entrada y salida y qué parte del material prestado ha sido devuelto. Aquellos accesos que no constan de salida, se les proporciona una acorde con la hora máxima de salida de su horario laboral. Además, de aquí surge una lista con todo el material que no ha sido devuelto y que el vigilante requerirá a los responsables en el siguiente acceso que realicen.
Por regla general todo material debe ser devuelto a la salida, exceptuando los olvidos que pueden suceder.

2. Salidas: Un trabajador desea abandonar el centro.

En este caso el proceso es similar para todos los trabajadores:

- Todas aquellas personas a las que ha sido necesario prestar una tarjeta es registrada su salida y la devolución de tarjeta.
- Toda persona que posee alguna llave en préstamo debe devolverla antes de abandonar el centro.

2.3. La seguridad

Informática Navarra S.A. dispone de un contrato firmado con Compañía Internacional de seguridad (CIS) por el cual la segunda empresa se hace responsable de la seguridad del centro de operaciones, encargándose del proceso de control de accesos. Los vigilantes de seguridad destinados al centro realizan las labores de:

- Control de todo el proceso de accesos descrito con anterioridad.
- Redacción de un parte de seguridad por servicio en el que se describen todas las incidencias acontecidas en el mismo.



Al comenzar el día, el vigilante de seguridad que se encuentre en servicio tendrá la obligación de hacer llegar al jefe de área de producción todos los accesos y partes de seguridad redactados el día anterior.

2.4. La comunicación

Para poder realizar la gestión de accesos y seguridad del centro, jefe de área de producción, autorizadores, vigilantes y responsables de personal de las diversas empresas llevan a cabo una comunicación continua por varias vías: en persona, vía telefónica, por e-mail, mediante notas,... Obtienen el resultado esperado pero de forma un tanto caótica y arriesgada.

3. Descripción global del producto

3.1. Perspectiva del producto

Toda la información que entra en juego en el control de accesos y préstamos es tratada por los vigilantes y responsables del centro a través de listados manuales, lo que hace que tanto la inserción de nueva información como la modificación de la existente sea del todo ineficaz. Lo mismo ocurre sí de lo que se trata es de rescatar información específica.

No menos preocupante es la practicidad del manejo de todos los partes de seguridad, por no hablar del grado de mejora que se podría obtener en la comunicación entre todos los responsables.

Por todo lo aludido con anterioridad, se pide desarrollar un producto software de calidad que sea capaz de automatizar todo el proceso descrito en la sección 2.2, redefiniéndolo y mejorándolo, además de proveer una vía de comunicación eficaz entre los diversos usuarios y de ser un medio de redacción, almacenamiento y consulta de los partes de seguridad producidos por los vigilantes del centro.¹

El sistema constará de una aplicación interactiva multiusuario que emplearán todos los tipos de usuarios (enumerados en la sección 4.2 de este artefacto) para realizar las tareas antes descritas, en menor o mayor grado dependiendo del tipo de usuario.

¹ Para más detalles es posible consultar toda la funcionalidad requerida por el cliente en el documento *Especificación inicial de los casos de uso*.



3.2. Beneficios

Los beneficios que podrán obtener los usuarios a partir del producto son muchos, entre los que se encuentran:

- Eficacia en el registro de entradas y salidas de personal, así como la asignación de tarjetas, llaves y llaveros a los diversos accesos.
- Flexibilidad en la generación de listados acerca de los elementos de información manejados.
- Redacción y visualización de los partes de seguridad, que detallan las incidencias ocurridas en el servicio.
- Comunicación entre los diversos usuarios de la aplicación.

4. Usuarios afectados por el proyecto

4.1. Cliente (co-desarrollador)

Llevaremos a cabo el desarrollo cooperativo, así como el seguimiento y validación continuada de todo el proyecto con Pedro Martínez Pérez, jefe de área de producción de Informática Navarra S.A.

4.2. Usuarios

A continuación mostramos una tabla que resume los usuarios afectados por la implantación del nuevo sistema y sus principales características.

CAMPO	DESCRIPCIÓN
Nombre	Jefe de área de producción
Descripción	Jefe de área de producción de Informática Navarra S.A.
Tipo	Gurú.
Responsabilidades	Encargarse de que el acceso y vigilancia del centro se realice conforme a las políticas de acceso y vigilancia establecidas.
Rol que asumirá	Administrador
Funcionalidad	Casi total. Salvo redactar partes de seguridad, podrá tener acceso a cualquier tipo de información para visualizarla, incorporarla, modificarla o eliminarla.

Usuarios afectados



Nombre	Ayudante del jefe de área de producción
Descripción	Trabajador específico dentro de Informática Navarra S.A.
Tipo	Usuario experto.
Responsabilidades	Trabajador perteneciente a Informática Navarra S.A. en quien el jefe de área de producción ha confiado el cometido de ayudarlo en la gestión de parte de las empresas y personal que trabajan en el centro. Tiene la potestad de poder dar acceso a empresas (departamentos de éstas) para que desempeñen su labor, haciéndose responsable de las mismas.
Rol que asumirá	Autorizador
Funcionalidad	Acceso a toda la información que concierne a las empresas (departamentos) de las que es responsable y posibilidad de comunicarse con los demás usuarios de la aplicación.
Nombre	Vigilante de seguridad
Descripción	Vigilante jurado encargado de la vigilancia del centro.
Tipo	Usuario experto.
Responsabilidades	Realizar las funciones de vigilancia del centro, redactando los partes de seguridad oportunos y registrando todos los accesos permitidos.
Rol que asumirá	Vigilante
Funcionalidad	Toda la necesaria para dar entrada y salida al personal que realiza sus labores en el centro, así como permitir que pueda realizar el préstamo de material de forma sencilla y satisfactoria. A esto hay que añadir la posibilidad de redactar los partes resultados de sus servicios de vigilancia y poder comunicarse con el resto de usuarios.
Nombre	Recursos humanos (resto de empresas)
Descripción	Responsable de personal de uno o varios departamentos pertenecientes a una empresa.
Tipo	Usuario casual.
Responsabilidades	Mantener actualizado constantemente el listado del personal perteneciente a su departamentos y que realiza sus labores en la sede de Informática Navarra S.A.
Rol que asumirá	RP
Funcionalidad	Visualización y edición de la lista del personal del que es responsable y que accede al centro.

Usuarios afectados

Pamplona, a 19 de enero de 2010.


Héctor Napal Basterrica
Responsable del proyecto

accessManager · Estudio detallado del problema software · Página 9 de 9



accessManager

Desarrollo de un sistema de control de accesos a un centro

Especificación del modelo de negocio

Fecha de entrega: 19 de enero de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Estructura del documento

2. Organización de la empresa

3. Modelo de casos de uso y objetos del negocio principales

- 3.1. Gestionar listas
- 3.2. Entrada de una persona
- 3.3. Salida de una persona
- 3.4. Resto de procesos



1. Introducción

1.1. Propósito

La finalidad de este escrito es el de desarrollar todo el modelado de negocio correspondiente a la empresa Informática Navarra S.A., de modo que quede delimitada y se comprenda el contexto de la misma en su totalidad. Para tal fin hemos elaborado los modelos de Casos de uso del negocio y Objetos del negocio, que detallamos en las siguientes secciones y que pueden ser complementados con toda la información recogida en *Estudio detallado del problema software*.

1.2 Audiencia

Se trata de un documento interno a Software Solutions que nos servirá de apoyo en la fase de comprensión del problema, careciendo de total importancia para el cliente.

1.3 Estructura del documento

En primer lugar mostramos la parte esencial de la organización de la empresa y la cual nos será necesario entender, para a continuación esbozar el modelo de negocio con un nivel de detalle suficiente.

2. Organización de la empresa

Se hace muy aconsejable comenzar delimitando el personal que forma parte del dominio del problema, para así simplificar la visión de éste. Las labores de control de accesos, vigilancia del centro y comunicación quedan restringidas a la siguiente organización jerárquica, en la que todas las personas cooperan para llevarlas a buen puerto. Destaca por encima de todas la importancia del jefe de área de producción, que dirige a sus ayudantes y vigilantes, mantiene correspondencia con los responsables de personal de los diversos departamentos de las empresas que trabajan en la sede y finalmente ofrece resultados periódicos a su superior.



Organigrama empresarial parcial



3. Modelo de casos de uso y objetos del negocio principales

Para no extendernos en demasía y simplificar el modelado de negocio, a continuación resumimos los dos modelos que lo componen, estudiando los procesos más importantes y complejos a través de sus respectivos diagramas junto con unas breves explicaciones de los mismos.

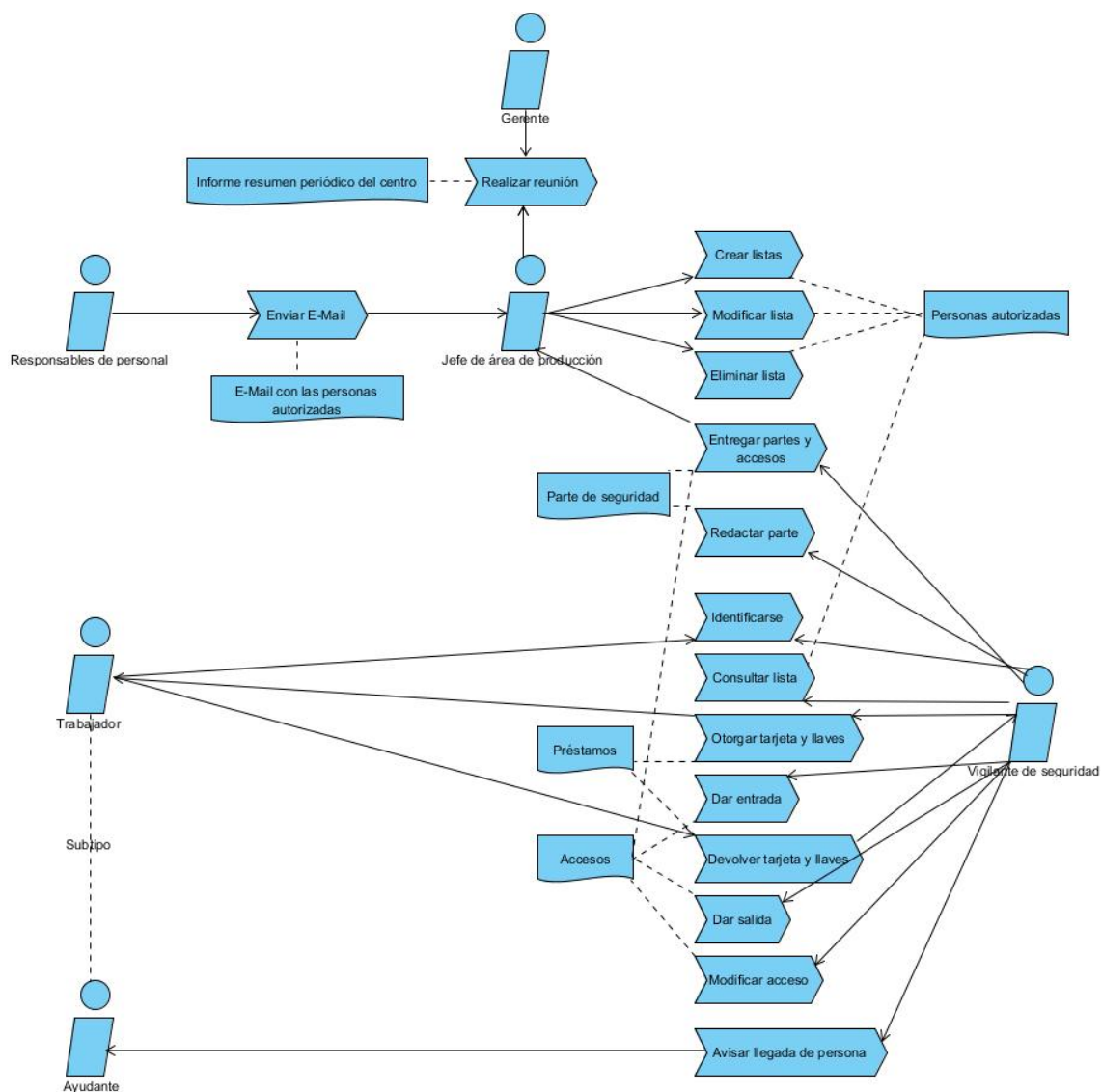


Diagrama de casos de uso y objetos del negocio – Global

3.1. Gestionar listas

Las listas que contienen todas las personas autorizadas a entrar al centro son elaboradas (creadas, mantenidas y eliminadas) por el jefe de área de producción a partir de los E-Mails recibidos por parte de los responsables del personal de los diversos departamentos de las empresas que trabajan en la sede de Informática Navarra S.A.

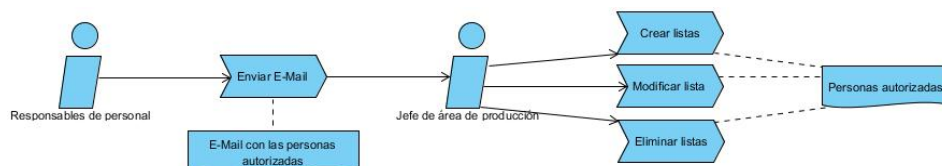


Diagrama de casos de uso y objetos del negocio – Gestionar listas

3.2. Entrada de una persona

La llegada al centro de una persona inicia todo el proceso de acceso. Ésta se dirige al vigilante identificándose mediante su DNI. El vigilante comprueba que consta en alguna de las listas, si se encuentra dentro de su horario laboral y si es necesaria una autorización telefónica. Tras realizar las comprobaciones necesarias le otorga la tarjeta y llaves y/o llaveros que necesite, registrando su entrada en una hoja de accesos y dándole vía libre.

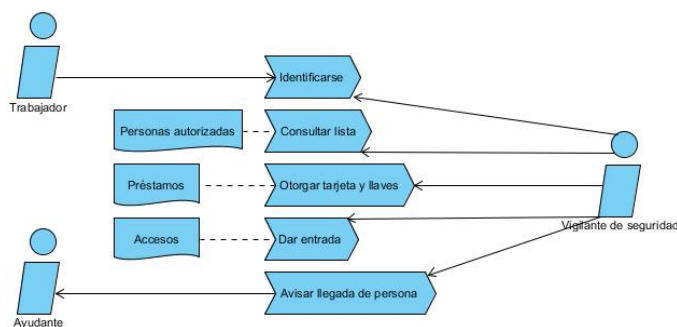


Diagrama de casos de uso y objetos del negocio – Entrada de una persona

3.3. Salida de una persona

A la salida del centro el trabajador devuelve la tarjeta y llaves y/o llaveros que le hayan sido prestados. A continuación, el vigilante de seguridad registra la devolución, busca en la hoja de accesos a la persona dándole salida y le permite su marcha.

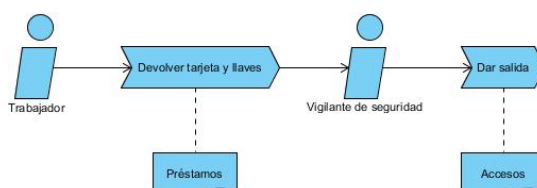


Diagrama de casos de uso y objetos del negocio – Salida de una persona



3.4. Resto de procesos

Los demás procesos son todos de tipo simple y pueden ser entendidos fácilmente observando el diagrama de casos de uso y de objetos del negocio global.

Pamplona, a 19 de enero de 2010.

Héctor Napal Basterrica
Responsable del proyecto



accessManager

Desarrollo de un sistema de control de accesos a un centro

Especificación inicial de los casos de uso

Fecha de entrega: 29 de enero de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Estructura del documento

2. Actores del sistema

3. Definición de los casos de uso

4. Aprobaciones y firmas



1. Introducción

1.1. Propósito

A lo largo de las siguientes páginas realizamos la especificación inicial de los casos de uso que deberán ser implementados por el nuevo sistema. Éste es un documento clave que deberá ser tenido en cuenta y refinado a lo largo de todo el desarrollo software.

1.2. Audiencia

Su interés es de tipo general, debiendo ser aprobado por el cliente y sirviendo de base para el trabajo del equipo de desarrollo.

1.3. Estructura del documento

En la segunda sección identificamos y describimos los actores que usarán en un futuro el sistema.

A continuación, y como finalización del documento, llevamos a cabo una especificación de los principales casos de uso, detallando qué funcionalidad proporcionan y las relaciones que existen entre ellos.

2. Actores del sistema

Delimitamos sistema y entorno estableciendo los siguientes actores:

CAMPO	DESCRIPCIÓN
Nombre	Administrador
Descripción	Administrador del sistema.
Principales uso del sistema	<ul style="list-style-type: none">Realizar tareas de mantenimiento de la información, haciendo que ésta siempre sea veraz.Obtener toda la información necesaria para realizar su labor.Comunicarse con los demás usuarios del sistema.
Nombre	Vigilante
Descripción	Vigilante de seguridad del centro.
Principales uso del sistema	<ul style="list-style-type: none">Llevar un control eficiente de los accesos realizados al centro y de los préstamos que conllevan.Redactar los partes de seguridad de los servicios que realiza.Comunicarse con los demás usuarios del sistema.

Actores del sistema



CAMPO	DESCRIPCIÓN
Nombre	Autorizador
Descripción	Trabajador de Informática Navarra S.A. que ayuda al Administrador en su labor de mantenimiento del sistema, así como de hacerse cargo de parte del personal que acceda al centro.
Principales uso del sistema	<ul style="list-style-type: none"> • Gestionar todo tipo de información referente a los departamentos de los que se es responsable. • Comunicarse con los demás usuarios del sistema.
Nombre	RP
Descripción	Responsable de personal de un departamento de una empresa.
Principales uso del sistema	Mantener actualizada continuamente la lista del personal propio que accede al centro.

Actores del sistema

3. Definición de los casos de uso

En las siguientes páginas proporcionamos las descripciones de trazo grueso generadas para los principales casos de uso y segmentadas con respecto a las diversas entidades reales existentes en el entorno del problema. Dicha división da lugar a grupos de casos de uso muy similares, constituyendo una buena aproximación a cómo dividiremos y acometeremos el trabajo en las siguientes iteraciones. La siguiente tabla resume éstos grupos y la prioridad que les hemos asignado:

CASOS DE USO			
GRUPO			PRIORIDAD
Cuentas			Imprescindible e importante
Accesos	Empresas	Departamentos	Imprescindible
Personal	Niveles	Colores	
Tarjetas	Tipos de acceso	Cargos	
Zonas del centro	Llaves	Llaveros	
Almacenes de llaves	Préstamos pendientes	Cuentas del sistema	
Tipos de usuario	Historial de accesos al sistema		
Recordatorios y borrado		Partes de seguridad	Importante
Notas		Base de datos	Deseable



Esta especificación, junto con el estudio detallado realizado con anterioridad y el supuesto del modelo de datos que generaremos más adelante, serán las armas que empleemos para definir de forma detallada y precisa todo el problema que nos ocupa.

Sin más preámbulos, veamos los casos de uso que implementaremos:

CUENTAS	
CAMPO	DESCRIPCIÓN
Caso de uso	1 - Login
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador, Vigilante y RP
Funcionalidad	El usuario accede al sistema, hecho que provoca que deba identificarse. El sistema le pide los datos y tras verificarlos despliega la aplicación. En caso de no existir el usuario, se le notifica el error y se permanece en la pantalla de Login.
Caso de uso	116 - Modificar datos de cuenta
Prioridad	Importante
Actor(es)	Administrador, Autorizador, Vigilante y RP
Funcionalidad	El usuario solicita cambiar alguno de sus datos de acceso (nombre o contraseña). El sistema le pide que introduzca sus datos de identificación actuales y los nuevos que desea utilizar. Tras comprobar que los primeros son correctos se realiza la actualización, en caso contrario se indica el error ocurrido y se permanece en la pantalla de Modificar datos de usuario.

Especificaciones de trazo grueso – Cuentas

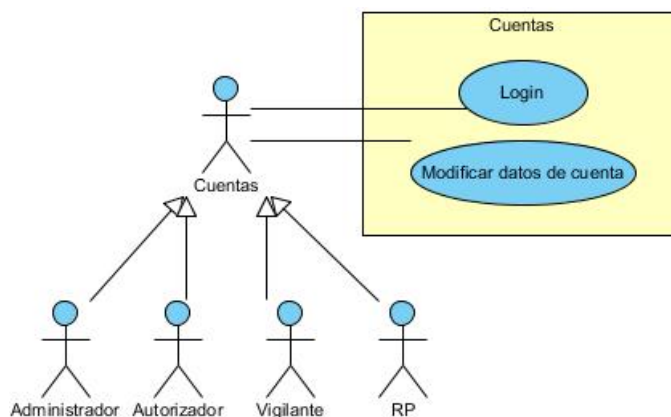


Diagrama de casos de uso – Cuenta



ACCESOS	
CAMPO	DESCRIPCIÓN
Caso de uso	2 - Visualizar accesos
Prioridad	Imprescindible
Actor(es)	Administrador y Vigilante
Funcionalidad	El usuario desea conocer los accesos realizados a la sede que cumplan unas condiciones concretas. Indica dicha solicitud al sistema, a lo que éste responde mostrándole el conjunto de accesos que cumplan con dichos requisitos.
Caso de uso	3 - Visualizar acceso
Prioridad	Imprescindible
Actor(es)	Administrador y Vigilante
Funcionalidad	El administrador o vigilante desea conocer toda la información referente a un acceso realizado al centro. Lo identifica, encargándose el sistema de mostrarle todos los datos de los que se compone.
Caso de uso	4 - Dar entrada
Prioridad	Imprescindible
Actor(es)	Administrador y Vigilante
Funcionalidad	El usuario pretende registrar el acceso al centro de una persona. El sistema le pide los datos, buscándola y determinando qué tarjeta le debe ser entregada. En caso oportuno se añaden las llaves y/o llaveros que se le prestan y se valida el acceso, quedando registrado.
Caso de uso	5 - Prestar llaves y/o llaveros
Prioridad	Imprescindible
Actor(es)	Administrador y Vigilante
Funcionalidad	Una persona que va a acceder o está ya en el centro necesita de llaves y/o llaveros para poder realizar sus labores. El usuario le presta lo que necesite, proporcionando al sistema persona y llaves prestadas, encargándose éste de grabar el préstamo.
Caso de uso	6 - Borrar acceso
Prioridad	Imprescindible
Actor(es)	Administrador y Vigilante
Funcionalidad	Es posible realizar el borrado de un acceso tras haberlo identificado.
Caso de uso	7 - Modificar acceso
Prioridad	Imprescindible
Actor(es)	Administrador y Vigilante
Funcionalidad	El usuario desea modificar algún dato de uno de los accesos registrados. Lo identifica y modifica a su gusto, aprobando que es correcto.
Caso de uso	8 - Borrar préstamo
Prioridad	Imprescindible
Actor(es)	Administrador y Vigilante
Funcionalidad	Se ha realizado el préstamo de una llave y/o llavero que no debiera haberse hecho. Para subsanar este hecho, el usuario selecciona el préstamo erróneo y le pide al sistema que lo elimine.

Especificaciones de trazo grueso – Accesos



Caso de uso	9 - Dar salida
Prioridad	Imprescindible
Actor(es)	Administrador y Vigilante
Funcionalidad	Se desea registrar que una persona se dispone a abandonar el centro. El sistema proporciona los datos del acceso y el usuario registra la salida indicando al sistema todos los elementos en préstamo devueltos.
Caso de uso	10 - Devolver llaves y/o llaveros
Prioridad	Imprescindible
Actor(es)	Administrador y Vigilante
Funcionalidad	Un trabajador desea devolver al vigilante parte o totalidad de las llaves y llaveros que le habían sido prestadas. El vigilante comunica al sistema qué le ha sido devuelto, identificando el prestamista y dando por cerrado el préstamo que se había iniciado con anterioridad.
Caso de uso	11 - Listar accesos
Prioridad	Imprescindible
Actor(es)	Administrador y Vigilante
Funcionalidad	Se desea obtener un listado con todos los accesos realizados al centro dependiendo de algún tipo de parámetro. El usuario acota la búsqueda y pide al sistema que le proporcione un listado con todos los accesos que la satisfagan.
Caso de uso	12 - Dar salida automática
Prioridad	Imprescindible
Actor(es)	Sistema
Funcionalidad	El sistema otorgará de salida a todos aquellos accesos del día anterior que careciesen de ella, estableciéndose la máxima en la que cada persona pudiese estar en el centro.

Especificaciones de trazo grueso – Accesos

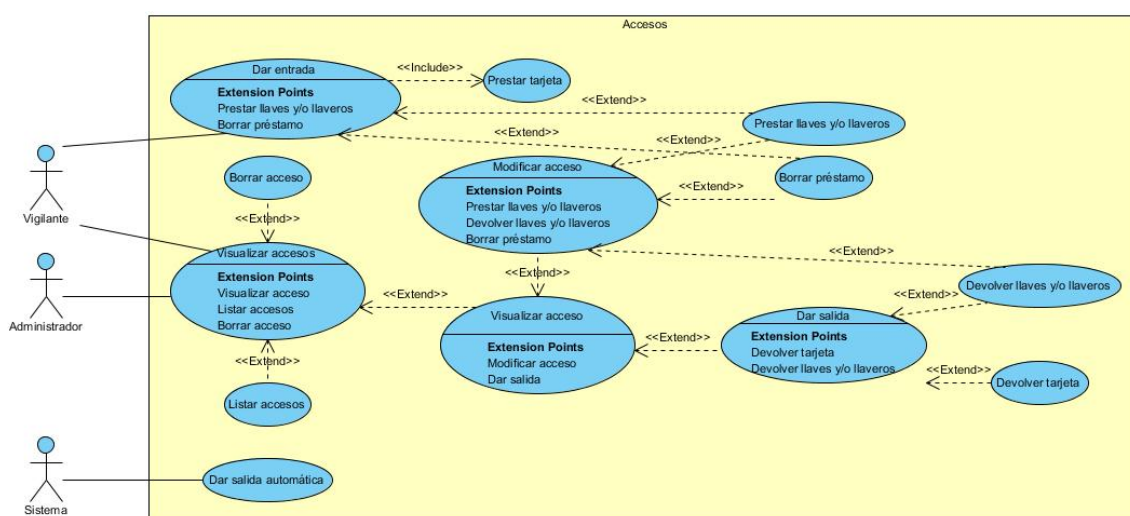


Diagrama de casos de uso – Accesos



EMPRESAS	
CAMPO	DESCRIPCIÓN
Caso de uso	13 - Visualizar empresas
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	El usuario puede visualizar de forma rápida y sencilla un listado con todas o parte de las empresas que trabajan a día de hoy en el centro.
Caso de uso	14 - Visualizar empresa
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	Se quiere visualizar toda la información relativa a una empresa. Tras identificarla el usuario pide al sistema que realice tal función, ofreciéndole éste toda la información de la que dispone.
Caso de uso	15 - Añadir empresa
Prioridad	Imprescindible
Actor(es)	Administrador y Autorizador
Funcionalidad	Una nueva empresa entra a formar parte de la “familia” de Informática Navarra S.A. El usuario le solicita al sistema darla de alta, pidiéndole éste todos los datos necesarios. Tras proporcionarlos y aprobar que son correctos, el sistema registra el alta.
Caso de uso	16 - Borrar empresa
Prioridad	Imprescindible
Actor(es)	Administrador y Autorizador
Funcionalidad	Una empresa deja de desarrollar sus labores en el centro. Por ello, el usuario desea darla de baja, identificándola y procediendo el sistema a su eliminación.
Caso de uso	17 - Modificar empresa
Prioridad	Imprescindible
Actor(es)	Administrador y Autorizador
Funcionalidad	El administrador o autorizador solicita al sistema modificar alguno de los datos que conforman el ente empresa en el sistema. Tras proporcionar los nuevos datos el sistema procede con la actualización de ésta.
Caso de uso	18 - Listar empresas
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	El usuario se dispone a obtener un listado de empresas del centro. Introduce los datos que cree conveniente para restringir la búsqueda y el sistema le proporciona el resultado deseado.

Especificaciones de trazo grueso – Empresas

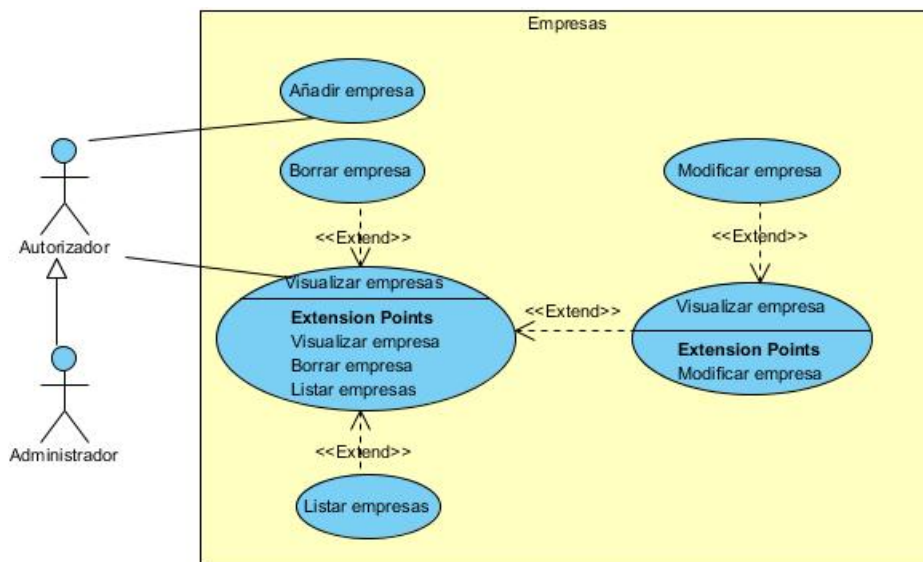


Diagrama de casos de uso – Empresas



DEPARTAMENTOS	
CAMPO	DESCRIPCIÓN
Caso de uso	19 - Visualizar departamentos
Prioridad	Imprescindible
Actor(es)	Administrador y Autorizador
Funcionalidad	El sistema provee la posibilidad de visualizar el conjunto de departamentos existentes en él y que cumplen con unas características determinadas; características que serán proporcionadas por el usuario y empleadas por la aplicación para realizar la búsqueda y mostrar los resultados.
Caso de uso	20 - Visualizar departamento
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	Se realiza una solicitud de visualización de toda la información de un departamento dado. El sistema obtiene toda la información y la ofrece la usuario.
Caso de uso	21 - Añadir departamento
Prioridad	Imprescindible
Actor(es)	Administrador y Autorizador
Funcionalidad	El usuario desea añadir a alguna de las empresas del centro un departamento nuevo que entra en escena en la vida diaria del centro. Lo solicita al sistema y éste le pide todos los datos necesarios. Tras proporcionárselos, un nuevo departamento es añadido a los ya existentes.
Caso de uso	22 - Borrar departamento
Prioridad	Imprescindible
Actor(es)	Administrador y Autorizador
Funcionalidad	El administrador o autorizador quiere borrar un departamento que consta en el sistema. Tras identificarlo, solicita al sistema que lo elimine, llevándose la tarea a buen puerto.
Caso de uso	23 - Modificar departamento
Prioridad	Imprescindible
Actor(es)	Administrador y Autorizador
Funcionalidad	Un departamento consta con datos erróneos, pretendiendo el usuario modificarlos. Tal solicitud hace que el sistema despliegue todos los datos que posee de éste, editando el usuario los datos deseados y afirmando que son correctos. La actualización tiene lugar de forma satisfactoria.
Caso de uso	24 - Listar departamentos
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	Se desea obtener un listado de departamentos con personal en el centro. El usuario acompaña dicha solicitud de todas las características que deban poseer éstos, elaborando el sistema el listado solicitado.

Especificaciones de trazo grueso – Departamentos

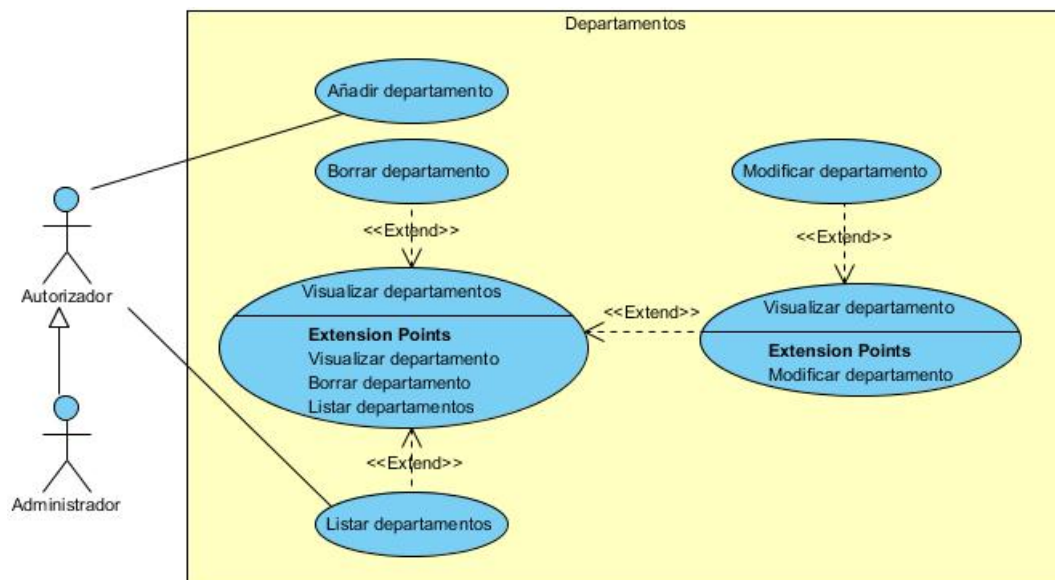


Diagrama de casos de uso – Departamentos



PERSONAL	
CAMPO	DESCRIPCIÓN
Caso de uso	25 - Visualizar personal
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	Será posible visualizar todo el personal que trabaja en el centro y acotarlo en función de diversos parámetros de búsqueda.
Caso de uso	26 - Visualizar persona
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	Surge la necesidad de acceder a los datos completos de una persona. El usuario la identifica y el sistema se encarga de obtenerlos y mostrárselos.
Caso de uso	27 - Añadir persona
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	El usuario desea dar de alta a una persona. Para ello proporciona al sistema todos los datos que éste le requiere, comprobando que son correctos y validando el alta.
Caso de uso	28 - Borrar persona
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	Se desea dar de baja a una persona. Se la identifica y se indica al sistema que proceda a su eliminación, dejando de formar parte del personal del centro.
Caso de uso	29 - Modificar persona
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	El usuario se percata de que algún dato perteneciente a una persona es erróneo, transmitiendo al sistema su deseo de modificarlo. Identifica al sujeto y corrige los datos que crea oportunos, haciendo que tenga en este momento todos sus datos descritos correctamente.
Caso de uso	30 - Añadir teléfono
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	El usuario identifica a una persona y solicita al sistema añadirle un número de teléfono. Lo teclea, pasando a ser almacenado con el resto de números.
Caso de uso	31 - Borrar teléfono
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	Una persona del centro deja de utilizar un número de teléfono, lo que lleva a su borrado ya que carece de sentido seguir almacenándolo. Se identifica qué número de teléfono entre todos los que posea debe ser borrado, llevando a cabo el sistema dicha operación.

Especificaciones de trazo grueso – Personal



Caso de uso	32 - Añadir E-Mail
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	Una persona pasa a tener una nueva cuenta de correo electrónico, que es registrada por el usuario mediante la identificación de la persona y el alta de la nueva dirección.
Caso de uso	33 - Borrar E-Mail
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	Por la razón que sea, una persona no desea recibir más comunicaciones a través de una cuenta de correo. El usuario solicita al sistema darla de baja, identificando persona y cuenta de correo. El sistema se encarga de eliminarla y que así no pueda ser utilizada en el futuro.
Caso de uso	34 - Añadir cargo
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	Un contacto de un departamento pasa a desarrollar un nuevo cargo, deseando el usuario que conste en el sistema. Para ello, identifica al contacto y solicita añadirle un nuevo cargo. El sistema le proporciona los cargos existentes, el usuario elige el que desee y éste queda adherido al contacto.
Caso de uso	35 - Borrar cargo
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	El usuario identifica un contacto y, tras seleccionar uno de los cargos que desempeña, solicita que sea borrado. El sistema le pide que apruebe el borrado, dejando de existir en el sistema en caso afirmativo.
Caso de uso	36 - Listar personal
Prioridad	Imprescindible
Actor(es)	Administrador, Autorizador y RP
Funcionalidad	El usuario desea obtener un listado de personas que posean unas características determinadas. Para ello, define las características del listado y el sistema lo elabora con los resultados obtenidos.

Especificaciones de trazo grueso – Personal

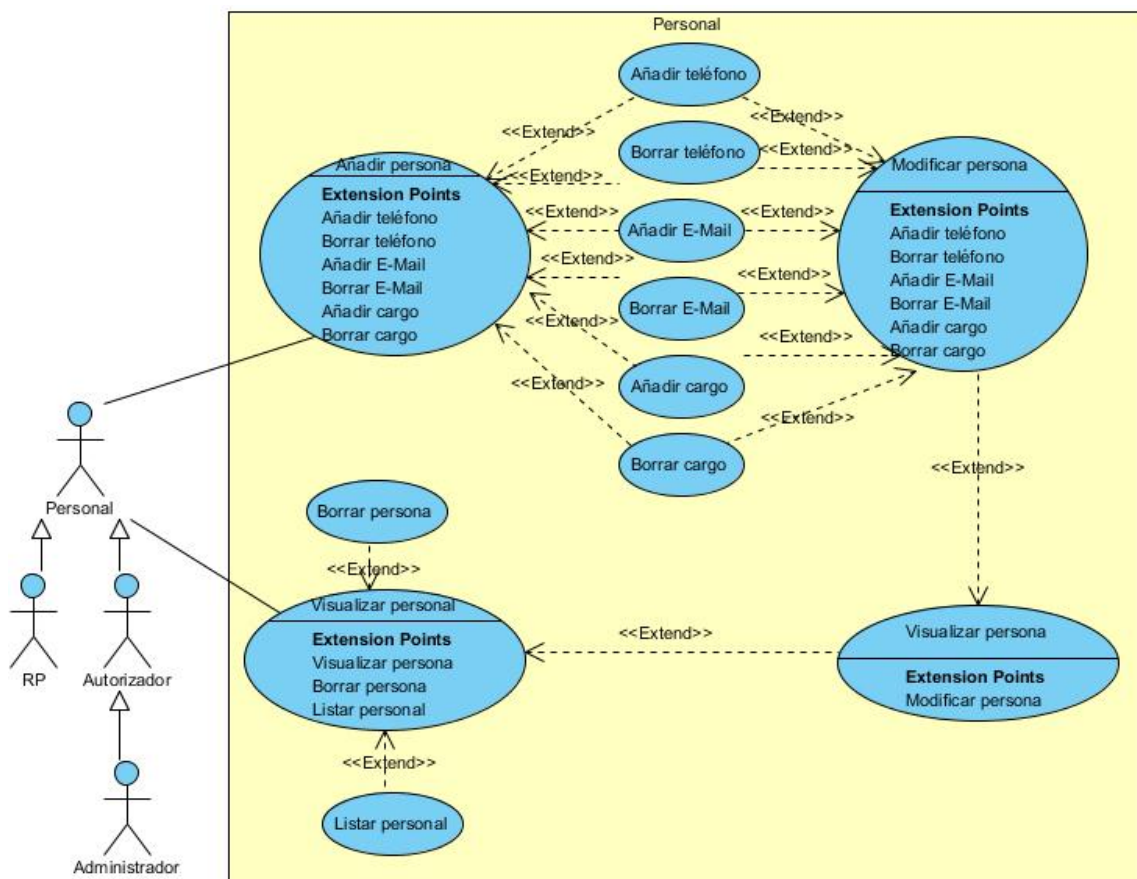


Diagrama de casos de uso – Personal



NIVELES	
CAMPO	DESCRIPCIÓN
Caso de uso	37 - Visualizar niveles
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador podrá visualizar todos los niveles existentes en el sistema.
Caso de uso	38 - Visualizar nivel
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Visualizar un nivel es tan fácil como identificarlo e indicar al sistema que muestre todos sus detalles.
Caso de uso	39 - Añadir nivel
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario solicita añadir un nuevo nivel a los ya existentes. El sistema le ofrece un formulario que rellena y valida, consiguiendo así su meta.
Caso de uso	40 - Borrar nivel
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El sistema permite borrar un nivel. Para conseguirlo, el usuario lo identifica y pide al sistema que lo elimine, hecho que realiza tras una breve aprobación por parte del usuario.
Caso de uso	41 - Modificar nivel
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Para modificar un nivel existente el administrador deberá identificarlo. Tras introducir los cambios que crea convenientes en los datos proporcionados por el sistema, el nivel queda grabado correctamente.
Caso de uso	42 – Dar de baja nivel
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador podrá dar de baja un nivel si éste ya no tiene cabida en el sistema pero se quiere seguir manteniendo la trazabilidad. Para ello realizará pasos análogos a los de borrar niveles.

Especificaciones de trazo grueso – Niveles

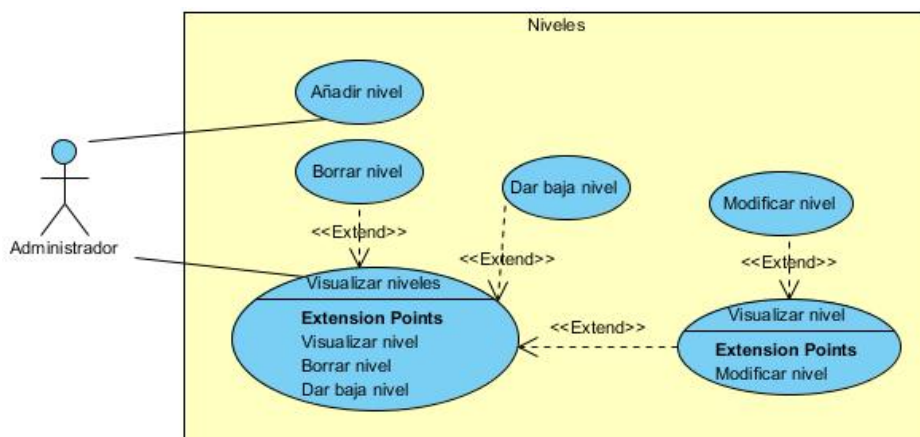


Diagrama de casos de uso – Niveles



COLORES	
CAMPO	DESCRIPCIÓN
Caso de uso	43 - Visualizar colores
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Debe ser posible visualizar de forma clara el conjunto de colores existentes en el sistema.
Caso de uso	44 - Visualizar color
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario está interesado en obtener todos los datos relativos a un color existente. Lo identifica y el sistema se los ofrece.
Caso de uso	45 - Añadir color
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario desea dar de alta un nuevo color. Realiza dicha solicitud al sistema, quien efectúa el alta tras recibir todos los datos necesarios.
Caso de uso	46 - Borrar color
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Un color deja de estar operativo y el administrador se dispone a eliminarlo del sistema. Lo identifica y solicita eliminarlo, tras lo cual el sistema le pide que confirme que ese es el color que desea eliminar. El administrador acepta y el color es borrado.
Caso de uso	47 - Modificar color
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Un color consta con datos erróneos, ante lo cual el administrador desea corregirlos. Identifica dicho color obteniendo todos sus datos en pantalla, edita los que considera equivocados y pide al sistema que los actualice.
Caso de uso	48 - Dar de baja color
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Al igual que se pueden eliminar colores, también debe ser posible darlos de baja. El usuario seleccionará un conjunto de colores y pedirá al sistema que los dé de baja, realizando dicha acción bajo confirmación previa.

Especificaciones de trazo grueso – Colores

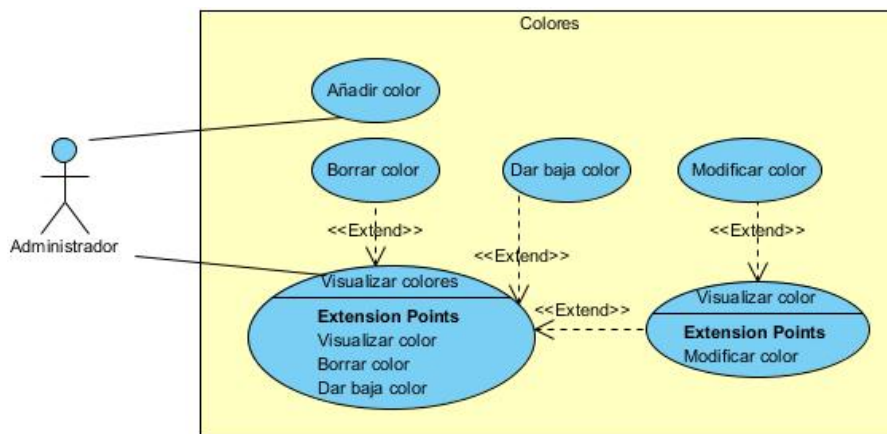


Diagrama de casos de uso – Colores



TARJETAS	
CAMPO	DESCRIPCIÓN
Caso de uso	49 - Visualizar tarjetas
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Será posible visualizar todas las tarjetas existentes en el centro.
Caso de uso	50 - Visualizar tarjeta
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	En caso de ser necesario, el administrador puede pedir al sistema que les muestre todos los datos de una tarjeta (color y numeración).
Caso de uso	51 - Añadir tarjeta
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Se dispone de una nueva tarjeta, queriendo darla de alta en el sistema para que pueda ser utilizada. Se indica a éste tal necesidad, a lo que responde pidiendo al usuario todos los datos necesarios. Éste se los proporciona y, tras aprobar que sean válidos, la tarjeta pasa a estar operativa.
Caso de uso	52 - Borrar tarjeta
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario selecciona una tarjeta y solicita al sistema darla de baja. Tras aprobar su borrado la tarjeta deja de existir en el sistema.
Caso de uso	53 - Modificar tarjeta
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Se desea modificar los datos de una tarjeta para poder darle otro uso distinto al que tiene. Para ello, el administrador identifica la tarjeta que quiere modificar, edita los datos oportunos y el sistema los recoge.
Caso de uso	54 - Dar baja tarjeta
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario quiere dar de baja una tarjeta. Selecciona la tarjeta e indica al sistema que proceda con su baja. El sistema realiza dicha acción si es aprobada por el usuario.

Especificaciones de trazo grueso – Tarjetas

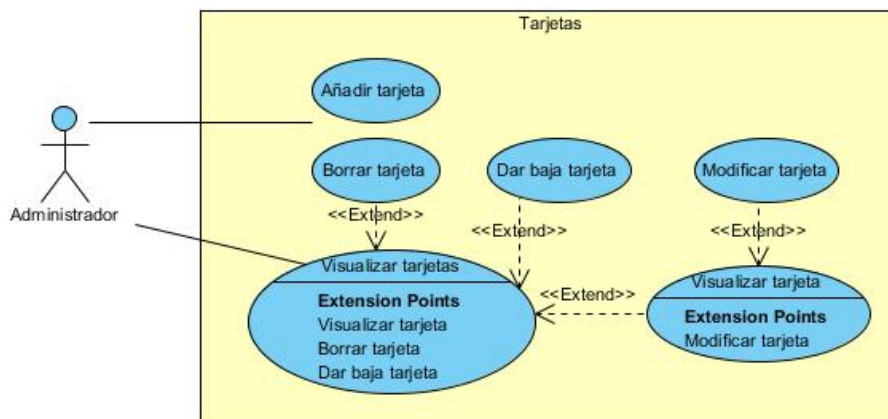


Diagrama de casos de uso – Tarjetas



TIPOS DE ACCESO	
CAMPO	DESCRIPCIÓN
Caso de uso	55 - Visualizar tipos de acceso
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El sistema deberá permitir que el administrador pueda visualizar todos los tipos de accesos existentes.
Caso de uso	56 - Visualizar tipo de acceso
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario desea conocer todos los datos de un tipo de acceso. Lo único que debe hacer es identificarlo e instar al sistema a que muestre todos los datos.
Caso de uso	57 - Añadir tipo de acceso
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Se desea otorgar un tipo de acceso inexistente a un departamento, teniendo como paso previo la realización del alta de dicho tipo de acceso. El usuario lo solicita al sistema, quien le insta a que rellene un formulario que le proporciona. Tras hacerlo y validarlo, el nuevo tipo de acceso queda registrado.
Caso de uso	58 - Borrar tipo de acceso
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Un tipo de acceso deja de ser válido, pretendiendo el administrador borrarlo. Lo identifica y traslada al sistema dicha necesidad, quien le pide que confirme que el tipo de acceso seleccionado es el que realmente quiere borrar. Tras aprobar el borrado, el tipo de acceso deja de existir en el sistema.
Caso de uso	59 - Modificar tipo de acceso
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador del sistema desea modificar un tipo de acceso que contiene datos erróneos. Para ello lo selecciona y, tras desplegar el sistema todos los datos, modifica lo necesario y aprueba los cambios.

Especificaciones de trazo grueso – Tipos de acceso

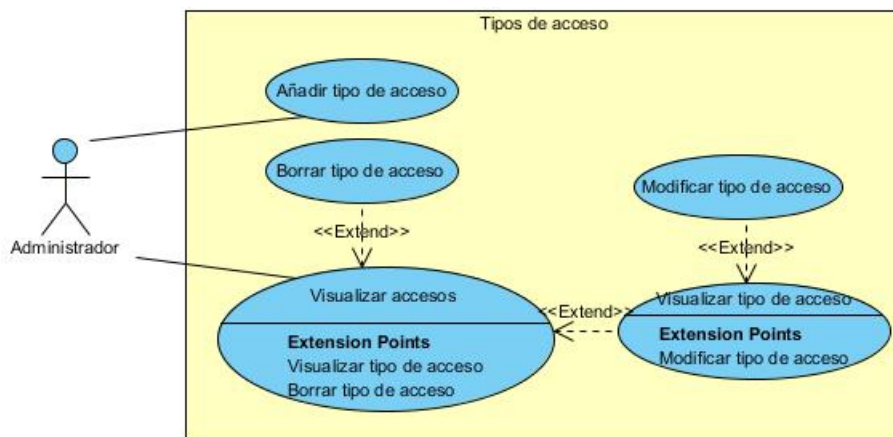


Diagrama de casos de uso – Tipos de acceso



CARGOS	
CAMPO	DESCRIPCIÓN
Caso de uso	60 - Visualizar cargos
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	No debe suponer esfuerzo alguno el que el usuario pueda visualizar todos los tipos de accesos existentes en el sistema.
Caso de uso	61 - Visualizar cargo
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador puede visualizar todos los datos referentes a un cargo mediante la identificación del mismo, encargándose el sistema de obtenerlos y mostrárselos.
Caso de uso	62 - Añadir cargo
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario decide introducir un nuevo cargo para un rango de personas con las que Informática Navarra S.A. pasará a mantener contacto. Traslada dicha solicitud al sistema, quien le pide todos los datos necesarios. El usuario se los proporciona y valida el alta, quedando el nuevo cargo registrado en el sistema.
Caso de uso	63 - Borrar cargo
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Se desea dar de baja un cargo que ha caído en desuso. Basta con identificarlo y pedir al sistema que lo elimine.
Caso de uso	64 - Modificar cargo
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Se decide modificar los datos de un cargo por no ser éstos incorrectos. Tras identificarlo, el usuario modifica los campos que crea oportunos y los valida, pasando a tener el cargo un nuevo estado.

Especificaciones de trazo grueso – Cargos

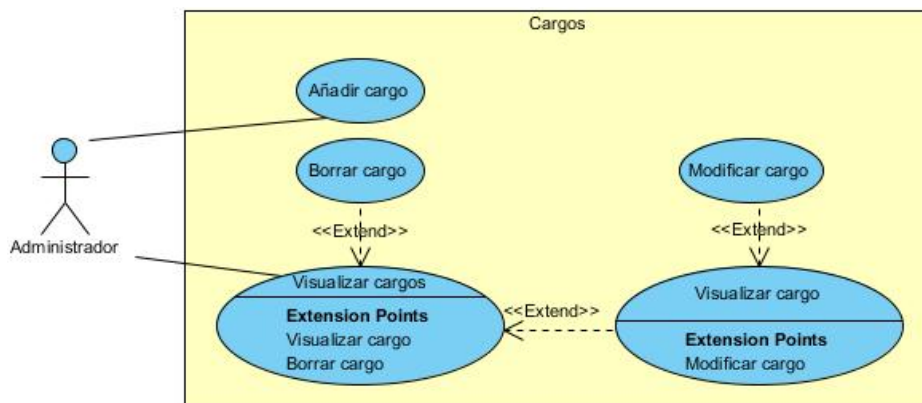


Diagrama de casos de uso – Cargos



ZONAS DEL CENTRO	
CAMPO	DESCRIPCIÓN
Caso de uso	65 - Visualizar zonas del centro
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario no debe tener el mínimo inconveniente para poder consultar de un simple vistazo todas las zonas que componen el centro.
Caso de uso	66 - Visualizar zona del centro
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Visualizar todos los datos relativos a una zona es tan sencillo como que el administrador la seleccione, encargándose el sistema de obtenerlos y mostrárselos.
Caso de uso	67 - Añadir zona del centro
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Surge una nueva zona en el centro que debe ser tratada al tener ésta cerraduras que abrir con las llaves existentes. Se comunica al sistema esta situación, pidiendo éste todos los datos que correspondan. El usuario los introduce y valida, quedando registrados de forma satisfactoria.
Caso de uso	68 - Borrar zona del centro
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario identifica una zona y solicita al sistema que la borre. Éste le pide que compruebe si la zona seleccionada es la que realmente desea borrar, a lo que el usuario responde afirmativamente. El borrado se produce y la zona deja de existir.
Caso de uso	69 - Modificar zona del centro
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Existen datos que no se corresponden con la realidad en alguna de las zonas existentes en el centro. El usuario identifica la zona afectada y el sistema muestra los datos de ésta para que puedan ser corregidos.

Especificaciones de trazo grueso – Zonas del centro

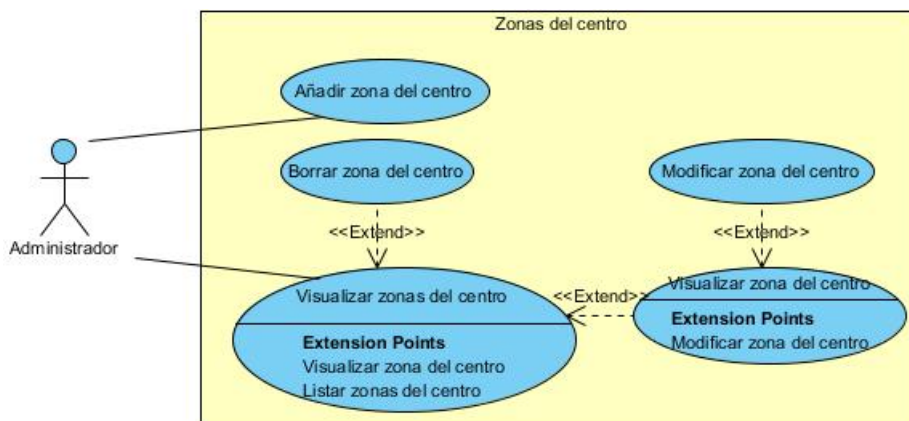


Diagrama de casos de uso – Zonas del centro



LLAVES	
CAMPO	DESCRIPCIÓN
Caso de uso	70 - Visualizar llaves
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Una de las funcionalidades que deberá ofrecer el sistema es la de mostrar los principales datos de las llaves registradas, pudiendo realizar una búsqueda sobre ellas.
Caso de uso	71 - Visualizar llave
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador desea obtener todos los datos relativos a una llave. Identifica la llave y el sistema se encarga de mostrarle todos los datos que posee de ella.
Caso de uso	72 - Añadir llave
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Se desea añadir una nueva llave a las ya existentes. El sistema pide al administrador todos los datos necesarios, aprobándolos el usuario y quedando registrados.
Caso de uso	73 - Borrar llave
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Una llave deja de tener validez, lo que promueve su eliminación. Para ello, basta con identificarla y aprobar su borrado.
Caso de uso	74 - Dar de baja llave
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	En aras de poder mantener la trazabilidad de todos los préstamos realizados en los accesos, existirá una funcionalidad que permitirá dar de baja llaves, quedando éstas registradas en el sistema pero no formando parte activa de la aplicación.
Caso de uso	75 - Modificar llave
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario tiene la necesidad de modificar los datos de una llave por no ser éstos correctos. Identifica la llave y el sistema se encarga de desplegar todos los datos de ésta. El usuario edita todo lo que estime conveniente e insta al sistema para que registre los cambios.
Caso de uso	76 - Listar llaves
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario podrá pedir a la aplicación que genere un listado con todas las llaves registradas en el sistema.

Especificaciones de trazo grueso – Llaves

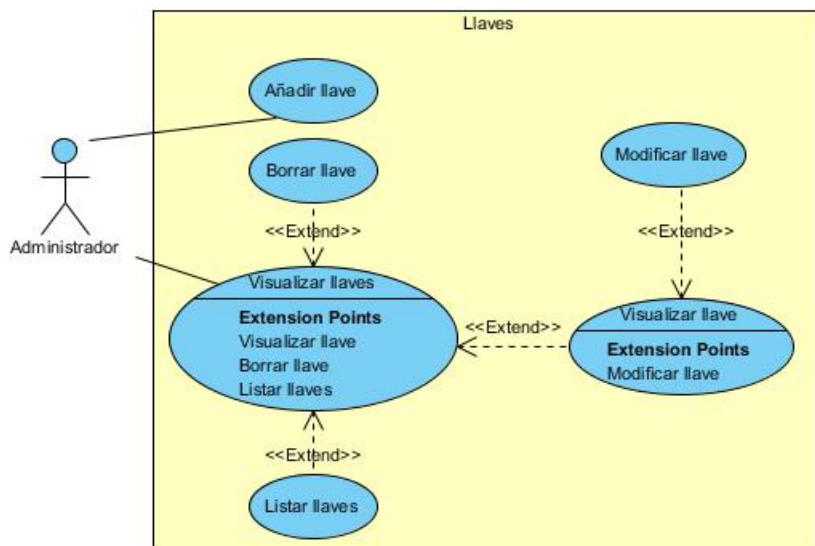


Diagrama de casos de uso – Llaves



LLAVEROS	
CAMPO	DESCRIPCIÓN
Caso de uso	77 - Visualizar llaveros
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador podrá acceder de manera sencilla a la visualización de todos los llaveros que ha registrado en el sistema y que permanecen activos.
Caso de uso	78 - Visualizar llavero
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario identifica un llavero y solicita visualizar todos sus datos, a lo que el sistema responde recuperándolos y mostrándoselos.
Caso de uso	79 - Añadir llavero
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	La dirección de Informática Navarra S.A. conforma un nuevo llavero, debiendo ser dado de alta en el sistema como paso previo para que pueda ser prestado a los trabajadores. Se solicita al sistema dicha alta proporcionándole todos los datos necesarios. Tras aprobar el usuario que todo es correcto se produce la inserción.
Caso de uso	80 - Borrar llavero
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Borrar un llavero es tan fácil como identificarlo y pedir al sistema que lo elimine. El único requerimiento que solicitará es la confirmación de que el llavero a borrar es el que el usuario realmente quiere borrar.
Caso de uso	81 - Dar de baja llavero
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Al igual que ocurre con las llaves, debe ser posible dar de baja un llavero siguiendo almacenando sus datos pero dejando de formar parte activa de la aplicación.
Caso de uso	82 - Modificar llavero
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Por algún motivo varía la composición de un llavero. El administrador solicita al sistema modificar dicho llavero, ofreciéndole éste todos los datos para que pueda editarlos. Una vez realizadas todos los cambios oportunos se aprueba la nueva composición del llavero.

Especificaciones de trazo grueso – Llaveros

Caso de uso	83 - Añadir llave
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Una nueva llave entra a formar parte del llavero. El administrador la selecciona y pide al sistema que la introduzca en el llavero indicado.
Caso de uso	84 - Borrar llave
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Al contrario que el caso de uso anterior, una llave deja de tener sentido en un llavero. Se comunica este hecho al sistema, quien se encarga de borrar la llave del llavero.
Caso de uso	85 - Listar llaveros
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador podrá realizar un listado con todos los llaveros que desee según los criterios de búsqueda suministrados a la aplicación.

Especificaciones de trazo grueso – Llaveros

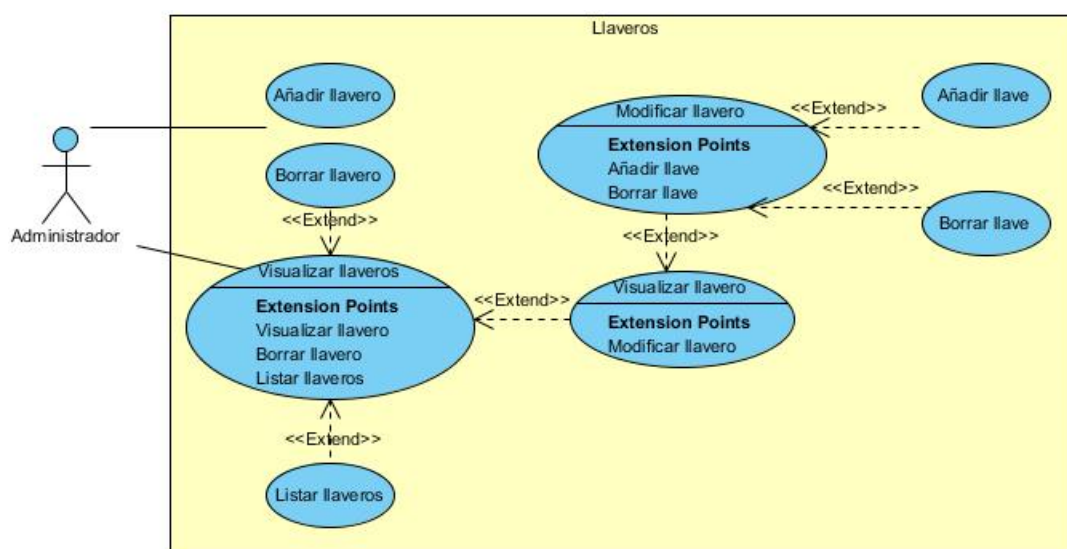


Diagrama de casos de uso – Llaveros



ALMACENES DE LLAVES	
CAMPO	DESCRIPCIÓN
Caso de uso	86 - Visualizar almacenes de llaves
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario tendrá la posibilidad de poder observar todos los almacenes de llaves existentes, con sus principales características y la relación de llaves y llaveros que albergan.
Caso de uso	87 - Visualizar almacén de llaves
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Se puede visualizar todos los datos de un almacén identificándolo e instando al sistema a que los muestre.
Caso de uso	88 - Añadir almacén de llaves
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	La dirección de la empresa decide crear un nuevo almacén de llaves para el centro. Para que pase a estar operativo, el administrador indica al sistema que desea añadirlo. Éste pide todos los datos necesarios, recibiendo los por parte del administrador y llevando a cabo la inserción en el sistema.
Caso de uso	89 - Borrar almacén de llaves
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador identifica un almacén y solicita al sistema que lo borre. El sistema pide confirmación para que se lleve a cabo la eliminación del almacén y todas copias de las llaves que éste contiene. Tras otorgarla el usuario consigue su objetivo.
Caso de uso	90 - Modificar almacén de llaves
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Modificar un almacén de llaves es tan sencillo como identificarlo y editar los campos desplegados que se estimen oportunos.
Caso de uso	91 - Añadir llave
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Bien sea por que se realizan más copias o por que se introduce una nueva llave en la base de datos, deberá proporcionarse la posibilidad de añadir llaves a cualquier almacén. El usuario seleccionará la llave a introducir y el número de copias existentes, mientras que el sistema se encargará de almacenar dicha información.

Especificaciones de trazo grueso – Almacenes de llaves



Caso de uso	92 - Borrar llave
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Una llave deja de formar parte de un almacén. El usuario la selecciona dentro de éste e indica al sistema que la elimine, dejando de existir copia alguna.
Caso de uso	93 - Modificar copias llave
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Debe ser posible modificar el número de copias de una llave existentes dentro de un almacén con solo seleccionarla y variar el número de copias actuales.
Caso de uso	94 - Añadir llavero
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Al igual que las llaves, también debe ser posible añadir llaveros a los almacenes, siguiendo el mismo procedimiento que el llevado a cabo con éstas.
Caso de uso	95 - Borrar llavero
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	En el momento en que un llavero deje de existir dentro de un almacén, el usuario lo seleccionará e indicará al sistema que lo elimine del almacén. Tras una petición de ratificación por parte del sistema, la operación será llevada a cabo.
Caso de uso	96 - Modificar copias llavero
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Caso de uso idéntico al de las llaves, pero referido a los llaveros.
Caso de uso	97 - Listar almacenes de llaves
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	La visualización de los diversos almacenes existentes se acompañará de una vía para poder obtener listados con todos los elementos que almacenan.

Especificaciones de trazo grueso – Almacenes de llaves

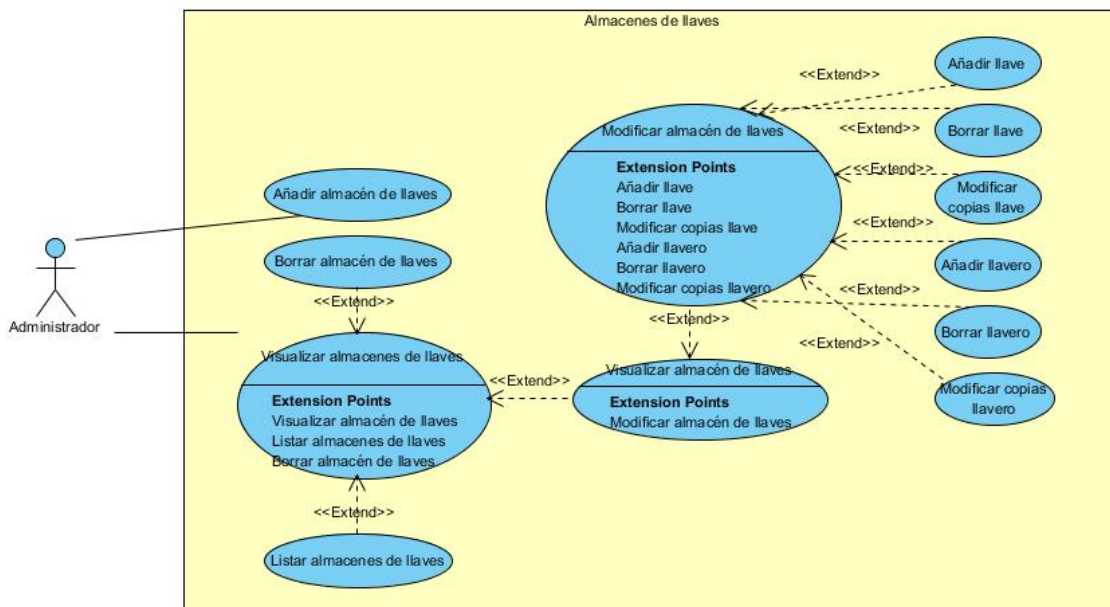


Diagrama de casos de uso – Almacenes de llaves



PRÉSTAMOS PENDIENTES	
CAMPO	DESCRIPCIÓN
Caso de uso	98 - Visualizar préstamos
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador desea visualizar qué préstamos están pendientes de devolución. Dicha pretensión queda resuelta gracias a que el sistema le proporciona una pantalla desde la que poder observar qué tarjetas, llaves y llaveros están a la espera de ser devueltos.
Caso de uso	99 - Visualizar datos del prestatario
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Una vez que el usuario disponga de los préstamos pendientes de devolución, podrá consultar qué persona es la prestataria en cada caso de una manera fácil y rápida, directa.
Caso de uso	100 - Devolver préstamo
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador deberá poder devolver los préstamos que desee desde la pantalla de consulta de préstamos, ya que en la de accesos no tendrá privilegios para editar información y aunque los tuviera no podría devolver préstamos de aquellos accesos que tengan salida.
Caso de uso	101 - Enviar recordatorio vía E-Mail
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	En caso de que el préstamo se prolongue en exceso o requiera que sea devuelto con la mayor brevedad posible, el sistema deberá proporcionar una vía que permita enviar un recordatorio vía E-Mail al prestatario en caso de disponer de alguna dirección de contacto.
Caso de uso	102 - Listar préstamos pendientes
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador podrá generar en cualquier momento un listado con todos los préstamos que están pendientes de devolución, combinando en él tarjetas, llaves y llaveros.

Especificaciones de trazo grueso – Préstamos pendientes

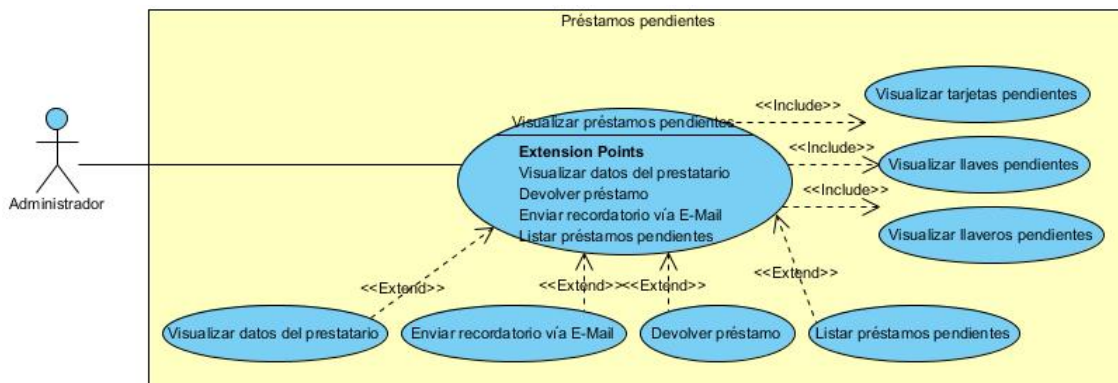


Diagrama de casos de uso – Préstamos pendientes



CUENTAS DEL SISTEMA	
CAMPO	DESCRIPCIÓN
Caso de uso	103 - Visualizar cuentas
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	La aplicación a desarrollar debe permitir visualizar todas las cuentas existentes en el sistema, así como sus principales datos, de forma que sea muy sencillo poder acceder a ellas para realizar las funciones que se estimen oportunas.
Caso de uso	104 - Visualizar cuenta
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario quiere visualizar toda la información relativa a una cuenta. Para ello, la identifica en el sistema y éste se encarga de recuperar y mostrar todos los datos de los que disponga.
Caso de uso	105 - Añadir cuenta
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador notifica al sistema que desea dar de alta una nueva cuenta. Éste le pide que le proporcione todos los datos necesarios, recogiénolos e incorporánolos al sistema.
Caso de uso	106 - Borrar cuenta
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Es posible realizar el borrado de una cuenta identificándola en el sistema y pidiendo a éste que la elimine.
Caso de uso	107 - Modificar cuenta
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador desea modificar los datos de una cuenta. El sistema le pide que la identifique. El administrador obedece y obtiene así todos los datos de la cuenta seleccionada, modificando lo oportuno y pidiendo al sistema que realice dichos cambios.
Caso de uso	108 - Listar cuentas
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	La aplicación debe proporcionar un modo de poder obtener un listado de cuentas del sistema que cumplan unas condiciones establecidas. El usuario limitará la búsqueda e instará al sistema a que elabore el listado con los datos que obtenga.

Especificaciones de trazo grueso – Cuentas del sistema

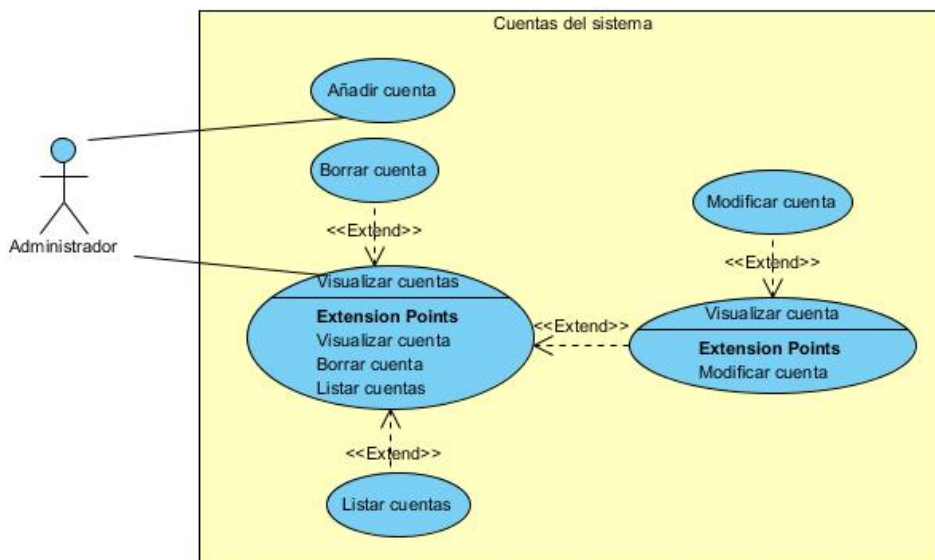


Diagrama de casos de uso – Cuentas del sistema



TIPOS DE USUARIO	
CAMPO	DESCRIPCIÓN
Caso de uso	109 - Visualizar tipos de usuario
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Al igual que toda la información manejada hasta el momento, los tipos de usuarios no se escapan de tener que poder ser visualizados.
Caso de uso	110 - Visualizar tipo de usuario
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El usuario desea saber los datos completos de un tipo de usuario existente en el sistema. Lo selecciona y el sistema es el que se encarga de rescatarlos y mostrárselos.
Caso de uso	111 - Añadir tipo de usuario
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador desea añadir un nuevo tipo de usuario en el sistema. Éste le proporciona un formulario que deberá rellenar y validar, encargándose el sistema de registrarlo.
Caso de uso	112 - Borrar tipo de usuario
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Al contrario que el anterior caso de uso, así como se deben poder añadir tipos de usuarios, también deberá ser posible eliminarlos mediante su identificación y petición al sistema de que realice esta acción.
Caso de uso	113 - Modificar tipo de usuario
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	Los datos de un tipo de usuario que consta en el sistema son erróneos. El administrador lo selecciona y pide al sistema poder modificarlo. Éste le muestra los datos actuales del tipo de usuario, siendo el administrador quien los modifica y el sistema quien actualiza los antiguos por éstos.

Especificaciones de trazo grueso – Tipos de usuarios

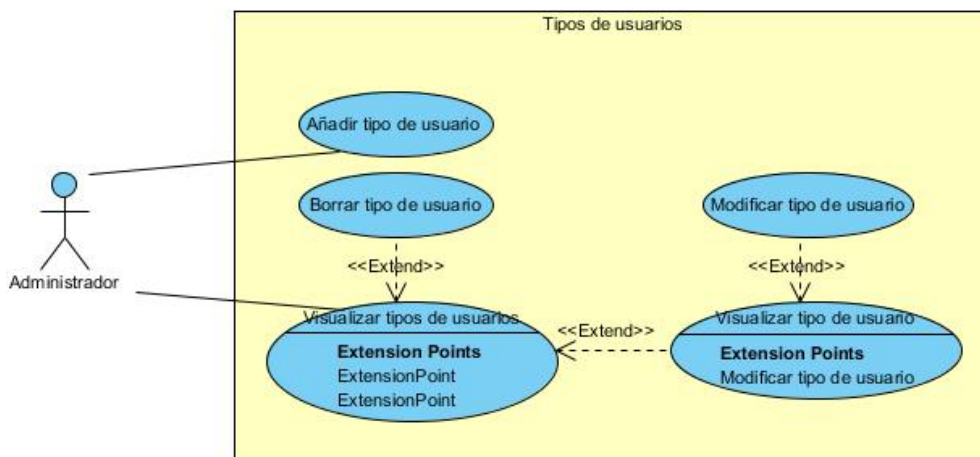


Diagrama de casos de uso – Tipos de usuarios

HISTORIAL DE ACCESOS AL SISTEMA	
CAMPO	DESCRIPCIÓN
Caso de uso	114 - Visualizar accesos al sistema
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	El administrador deberá poder consultar los accesos realizados al sistema por los diversos usuarios del mismo, de forma total o acotando esta con diversas variables (nombre, apellidos, fechas,...).
Caso de uso	115 - Listar accesos al sistema
Prioridad	Imprescindible
Actor(es)	Administrador
Funcionalidad	La visualización de accesos llevada a cabo en el anterior punto podrá ser obtenida mediante un listado proporcionado por el sistema.

Especificaciones de trazo grueso – Historial de accesos al sistema

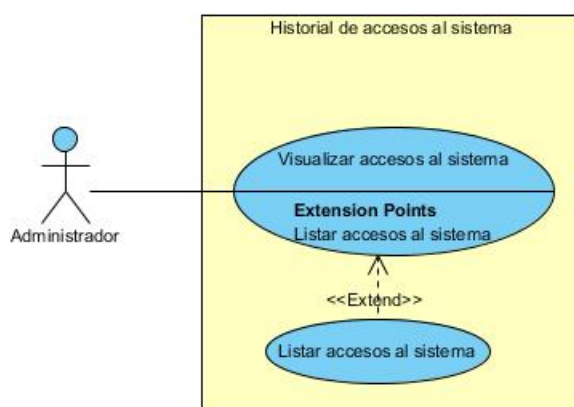


Diagrama de casos de uso – Historial de accesos al sistema

RECORDATORIOS Y BORRADO	
CAMPO	DESCRIPCIÓN
Caso de uso	117 - Recordar personal a punto de caducar
Prioridad	Importante
Actor(es)	Sistema
Funcionalidad	El sistema será capaz de enviar E-Mails a los departamentos cuyo personal está próximo a caducar según la política establecida por Informática Navarra S.A.
Caso de uso	118 - Recordar préstamos pendientes
Prioridad	Importante
Actor(es)	Sistema
Funcionalidad	También deberá existir un procedimiento por el cual toda aquella persona que tenga algún préstamo pendiente de devolución sea avisada en los casos que se estimen oportunos.
Caso de uso	119 - Borrar datos obsoletos
Prioridad	Importante
Actor(es)	Sistema
Funcionalidad	Para evitar que el sistema crezca de forma desmesurada, se deberá poder llevar a cabo el borrado de toda aquella información que carezca de sentido seguir almacenándola en el sistema.

Especificaciones de trazo grueso – Recordatorios y borrado

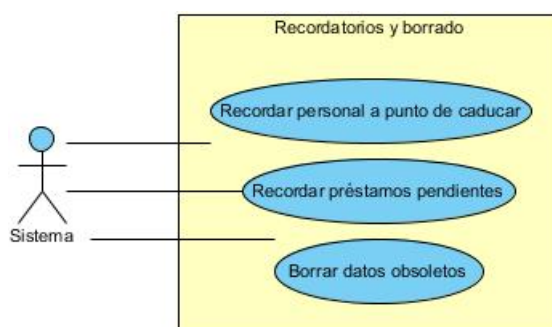


Diagrama de casos de uso – Recordatorios y borrado



PARTES DE SEGURIDAD	
CAMPO	DESCRIPCIÓN
Caso de uso	120 - Visualizar partes de seguridad
Prioridad	Importante
Actor(es)	Administrador y Vigilante
Funcionalidad	Se debe proporcionar una forma efectiva de poder visualizar todos los partes almacenados en el sistema, de modo que administrador y vigilante puedan acceder a ellos en el momento que deseen.
Caso de uso	121 - Visualizar parte de seguridad
Prioridad	Importante
Actor(es)	Administrador y Vigilante
Funcionalidad	El usuario del sistema desea consultar toda la información que consta en un parte de seguridad concreto. Lo identifica y el sistema es el que se encarga de proporcionar toda la información rescatada.
Caso de uso	122 - Añadir parte de seguridad
Prioridad	Importante
Actor(es)	Vigilante
Funcionalidad	El vigilante de seguridad desea crear un nuevo parte durante el desarrollo de un servicio. Transmite esta intención al sistema, quien le ofrece un formulario en el que puede empezar a redactar el parte y guardar todo lo que escriba.
Caso de uso	123 - Borrar parte de seguridad
Prioridad	Importante
Actor(es)	Administrador y Vigilante
Funcionalidad	El usuario identifica un parte de seguridad y solicita al sistema su borrado. Éste le pide que confirme que el parte seleccionado es el que realmente quiere borrar. El usuario da su conformidad y el parte deja de existir en el sistema.
Caso de uso	124 - Modificar parte de seguridad
Prioridad	Importante
Actor(es)	Vigilante
Funcionalidad	El vigilante desea modificar un parte de seguridad, que redactó con anterioridad, por estar éste inacabado o contener datos erróneos. Lo identifica y el sistema le muestra todos los datos que dispone de éste. El usuario edita y añade todo lo que estime oportuno, siendo el sistema quien registre los cambios.
Caso de uso	125 - Listar parte de seguridad
Prioridad	Importante
Actor(es)	Administrador y Vigilante
Funcionalidad	Tanto el administrador como el vigilante podrán descargar una copia del parte de seguridad que se encuentren visualizando.

Especificaciones de trazo grueso – Partes de seguridad

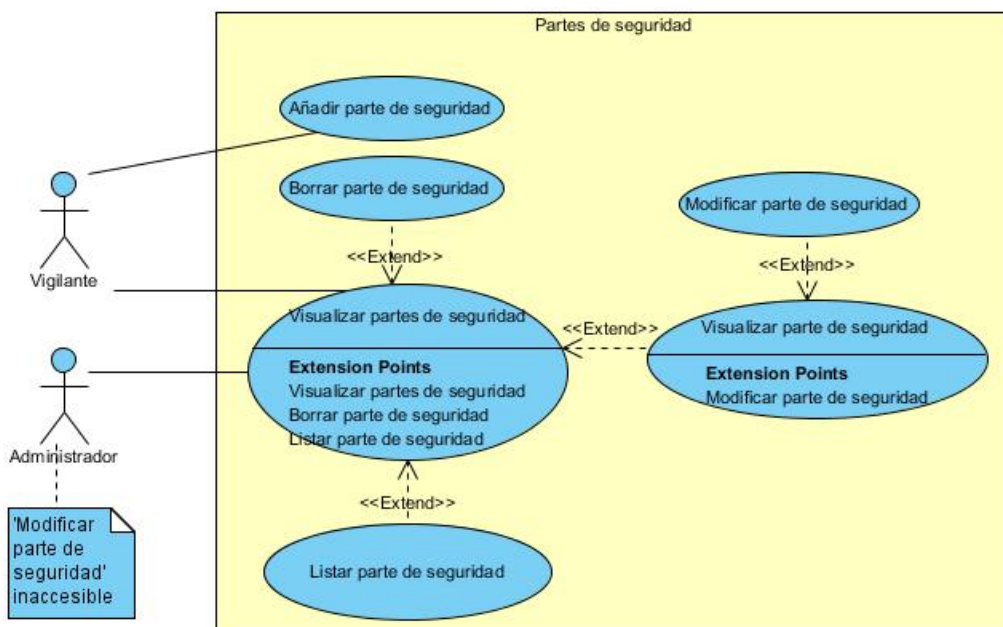


Diagrama de casos de uso – Partes de seguridad



NOTAS	
CAMPO	DESCRIPCIÓN
Caso de uso	126 - Redactar nota
Prioridad	Deseable
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	El usuario desea redactar una nota. El sistema le proporciona un formulario para que la redacte y elija los destinatarios. Tras validarla, el sistema se encarga de almacenarla y hacerla llegar a sus destinatarios.
Caso de uso	127 - Visualizar notas recibidas
Prioridad	Deseable
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	Se dispondrá de un listado de notas recibidas desde donde el usuario podrá acceder al contenido de las mismas.
Caso de uso	128 - Visualizar nota recibida
Prioridad	Deseable
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	El destinatario de una nota desea conocer todos sus detalles. Para ello la identifica y le pide al sistema que le muestre todos los datos de ésta.
Caso de uso	129 - Borrar nota recibida
Prioridad	Deseable
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	Es posible borrar una nota innecesaria con sólo identificarla y aprobar su eliminación.
Caso de uso	130 - Visualizar notas enviadas
Prioridad	Deseable
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	Así como se dispone de un almacén de entrada de notas, también se tendrá otro de notas enviadas, desde donde poder realizar la visualización y eliminación de éstas.
Caso de uso	131 - Visualizar nota enviada
Prioridad	Deseable
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	El redactor de la nota desea consultar en qué estado se encuentra una de las notas que ha redactado. Tras identificarla y realizar dicha petición al sistema, éste le mostrará la nota redactada y qué destinatarios la han leído y quiénes no.
Caso de uso	132 - Borrar nota enviada
Prioridad	Deseable
Actor(es)	Administrador, Autorizador y Vigilante
Funcionalidad	El usuario solicita al sistema descartar una nota de la cual es redactor mediante su identificación. A partir de entonces ésta no volverá a mostrarse al usuario.

Especificaciones de trazo grueso – Notas

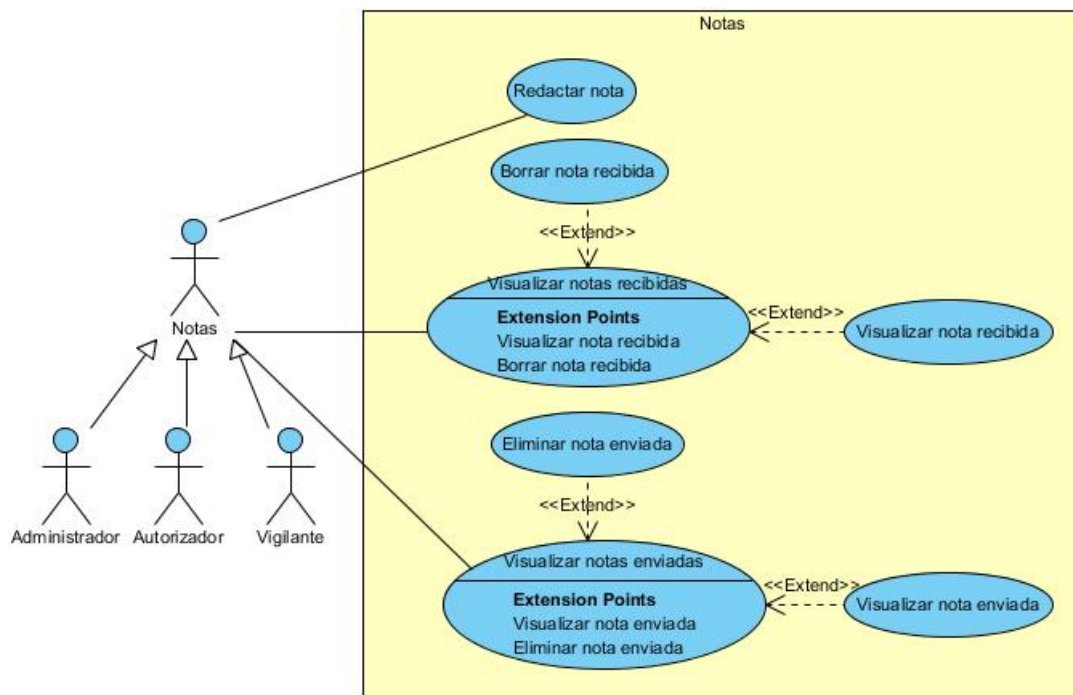


Diagrama de casos de uso – Notas

BASE DE DATOS	
CAMPO	DESCRIPCIÓN
Caso de uso	133 - Importar base de datos
Prioridad	Deseable
Actor(es)	Administrador
Funcionalidad	El administrador desea importar toda la información que posee en un archivo SQL. El sistema le insta a que indique la ruta del fichero a importar, petición que obedece consiguiendo introducir en el sistema toda la información deseada.
Caso de uso	134 - Exportar base de datos
Prioridad	Deseable
Actor(es)	Administrador
Funcionalidad	El usuario desea exportar toda la información existente en el sistema para que así pueda ser recuperada en un futuro. Indica dicha necesidad al sistema, quien genera un archivo SQL con toda la población de la base de datos.

Especificaciones de trazo grueso – Base de datos

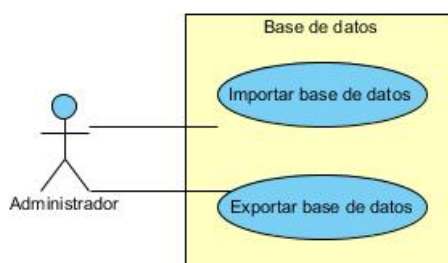



Diagrama de casos de uso – Base de datos

4. Aprobaciones y firmas

Con las siguientes firmas queda aprobado *'Especificación inicial de los casos de uso'* por parte del cliente, constituyendo la base contractual y de desarrollo principal del proyecto.

Pamplona, a 29 de enero de 2010.


 Héctor Napal Basterrica
Responsable del proyecto


 Pedro Martínez Pérez
Jefe de área de producción de Informática Navarra S.A.



accessManager
Desarrollo de un sistema de control de accesos a un centro

Arquitectura preliminar

Fecha de entrega: 2 de febrero de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Definiciones, acrónimos y abreviaciones
- 1.4. Estructura del documento

2. Evolución del 'Análisis' a la arquitectura hardware-software

- 2.1. Tipo de arquitectura
- 2.2. Elección del modelo de la base de datos

3. Análisis por capas



1. Introducción

1.1. Propósito

Elaboramos este breve escrito con el objeto de establecer una arquitectura básica que sirva de soporte para el sistema en construcción. Hay que tener en cuenta que se trata de un modelo inicial, una base que podrá y deberá ser refinada a lo largo del proceso de desarrollo del producto software.

1.2. Audiencia

Se trata de un documento de carácter interno, aunque en un futuro será entregado al cliente junto con toda la documentación fundamental del proyecto.

1.3. Definiciones, acrónimos y abreviaciones

- | | | |
|-------|--------------------------|---|
| • DBA | Data Base Administrator | Administrador de bases de datos |
| • NFS | Network File System | Sistema de ficheros en red |
| • FTP | File Transfer Protocol | Protocolo de transmisión de ficheros |
| • RMI | Remote Method Invocation | Mecanismo de invocación remota de métodos |

1.4. Estructura del documento

En la siguiente sección realizamos una breve aproximación a la arquitectura global que presentará todo el sistema, junto con los componentes que dispondrán. Más adelante, en la sección 3, profundizamos en dicha estructura observándola desde un breve análisis de las capas de las que dispone.

2. Evolución del 'Análisis' a la arquitectura hardware-software

El diseño de la distribución del sistema responde a varias razones que tratamos de explicar.

2.1. Tipo de arquitectura

Nos encontramos en un escenario en el que varias personas establecen una comunicación bidireccional con un sistema para conseguir diversos objetivos. Todos ellos deberán trabajar sobre los mismos datos, hecho que nos obliga a establecer un "almacén común de datos", y con ello una máquina servidor contenedora de una base de datos.



2.2. Elección del modelo de la base de datos

Tenemos pues, una **arquitectura cliente-servidor** en el que sabemos que unos clientes harán peticiones a una máquina servidor que alojará una base de datos. Profundicemos más en este tema.

La creación de una aplicación de bases de datos distribuida es algo más que el simple desarrollo de una aplicación. Implica comprender cómo interactúan los sistemas a través de las redes, bajo qué condiciones el rendimiento de las bases de datos es óptimo, los puntos fuertes y débiles de todos los componentes de nuestro sistema, cómo sacar el máximo rendimiento,...

Debemos hacer una “planificación para el futuro”, diseñando nuestro sistema de forma que sea lo suficientemente flexible para admitir cambios y no meternos así en problemas posteriores.

Demos un repaso rápido a los modelos de aplicaciones de bases de datos más populares, para así optar por el que más se amolde a nuestras pretensiones.

Modelo de dos niveles

Muchas aplicaciones de bases de datos utilizan un modelo de dos niveles, ya que se trata de un modelo relativamente sencillo de implementar y diseñar.

En este modelo se establece una arquitectura cliente-servidor en el que el software del primero mantiene una conexión persistente directa con una base de datos a lo largo de la vida de la aplicación. Esta conexión permanece viva tanto si el usuario está utilizando activamente la aplicación como si no, estando compuesta frecuentemente de un protocolo del fabricante ejecutándose sobre un protocolo de transporte. Esto la convierte en una conexión bastante congestionada, ya que el protocolo del fabricante puede estar realizando comprobaciones de errores y revisiones de conexiones continuas, además de lo que ofrezca el protocolo de transporte.

Como podemos ver hay una gran cantidad de sobrecarga sólo en lo que respecta al mantenimiento de la conexión.

Para desarrollar una aplicación que utilice este modelo, cada usuario necesita la instalación del software y de las bibliotecas de cliente de la base de datos compiladas para la plataforma del cliente.



Modelo de n niveles

En respuesta a los problemas de licencias, congestión y dificultad de instalación y configuración de las máquinas clientes, la tendencia en la computación de bases de datos ha pasado a la solución de los n niveles.

En este modelo, un equipo intermedio o agente gestiona las solicitudes de los clientes utilizando un protocolo ligero y redireccionándolas después a la base de datos, utilizando el protocolo del fabricante.

Ventajas:

- **Conexiones sin estado entre los servidores y los clientes.** En primer lugar está el hecho de que la conexión con lo que los clientes consideran el servidor (ahora el nivel intermedio) es sin estado (no permanece abierta). Por tanto, durante los periodos de inactividad de la aplicación, ni el cliente ni el servidor utilizarán recursos manteniendo una conexión no utilizada.
- **No se utilizan protocolos específicos del fabricante para las comunicaciones entre el cliente y el servidor;** lo que hace que las máquinas clientes no sufran tanta congestión y no sea necesario instalar en cada una de ellas las bibliotecas necesarias.
- **Las conexiones de bases de datos abiertas persistentes entre el nivel intermedio y el servidor de la base de datos.** Gran parte de la sobrecarga en el acceso a una base de datos se produce durante la apertura inicial de la conexión con ella. Al eliminar esta tarea en el cliente, los tiempos de carga iniciales descienden.
- **JDBC garantiza la independencia de la base de datos.** Como el driver está implementado en Java, podemos ejecutarlo y, por tanto, conectar con la base de datos desde cualquier plataforma. Además, si en un futuro decidimos cambiar de fabricante de bases de datos o necesitamos acceder a varias bases de datos de distribuidores diferentes desde el nivel intermedio, todo lo que tenemos que hacer es cambiar los URL de JDBC y ya está.
- **Escalabilidad y flexibilidad.** Con este diseño, nuestro sistema puede crecer para adjuntar varios orígenes de datos, independientemente del fabricante. Nuestro nivel intermedio puede comunicarse con varias bases de datos, del mismo o diferente fabricante.
- **Sencillez en la programación.** Las aplicaciones clientes serán más sencillas en sus estructuras y de desarrollo más rápido.

Recordemos que las conexiones entre los clientes y el nivel intermedio son sin estado. Es decir, sólo mantienen una conexión con el servidor de nivel intermedio el tiempo necesario para conseguir lo que quiere y, después, cierran la conexión. Esto es un gran avance, ya que asegurarse de que los drivers del software cliente de la base de datos



están cargados de la forma apropiada, en el sitio correcto, con la versión adecuada y con las variables de ruta del usuario y de entorno configuradas debidamente supone casi más trabajo que el desarrollo de la aplicación completa. El único lugar donde necesitamos mantener drivers específicos del distribuidor actualizados es en el nivel intermedio, un equipo sobre el que tenemos el control.

Desde la perspectiva de la administración, el tener que mantener otro servidor puede no parecer deseable. Sin embargo, el nivel intermedio no requiere el mismo nivel de mantenimiento que un servidor de bases de datos comercial normal, además de ser de una complejidad baja, siendo la configuración y el soporte de este servidor casi exactamente igual que la administración de un servidor web, lo que hace que no debiera ser muy difícil de mantener por cualquier DBA, o persona con unos conocimientos básicos en informática.

Además de actuar como nivel intermedio, en un futuro podríamos utilizar este servidor para que haga las funciones de servidor de archivos o de aplicaciones (NFS, Samba, FTP,...).

Como vemos, la programación distribuida, y en concreto el modelo de bases de datos en n niveles, trae consigo muchas ventajas, aunque también inconvenientes que deberemos tener en cuenta:

- Hemos introducido potencialmente un único punto de fallo.
- Podrían “caer”:
 - Nivel intermedio
 - Solución: Podríamos diseñar el cliente para que pasara a otro servidor de nivel intermedio o a ejecutarse en el mismo servidor de bases de datos. O podríamos cargar código dinámicamente para pasar temporalmente a un modelo de dos niveles.
 - Servidor de bases de datos
 - Solución: Podríamos diseñar nuestro nivel intermedio para detectar el fallo de la base de datos, alertar al usuario o incluso permitirle continuar mientras almacenamos en la memoria temporal las transacciones para su inserción posterior en la base de datos.
 - Red
 - Sin red no funciona nada. No podemos hacer gran cosa desde la perspectiva del software.



En el caso que nos ocupa podríamos optar por cualquiera de los dos modelos:

- Modelo de dos niveles: Nos encontramos ante el desarrollo de una aplicación y base de datos para un entorno cerrado y con un número reducido de usuarios, lo que hace que las peticiones que tiene que atender, y por tanto los recursos que tiene que gastar no sean elevados y fácilmente subsanables por este modelo.
- Modelo de n niveles: Se trata de un desarrollo más complejo que el anterior, mucho más completo y eficiente en muchos casos, pero no en el nuestro; no nos parece una solución acorde con el problema que nos ocupa. Aun así, con el fin de facilitar una futura extensión del sistema, codificaremos el mismo implementando todas las interfaces necesarias para que los futuros mantenedores puedan crear un sistema de n niveles con el menor esfuerzo posible a través del uso de Java RMI.

Tras el estudio realizado decidimos emplear una arquitectura cliente-servidor como la que sigue:

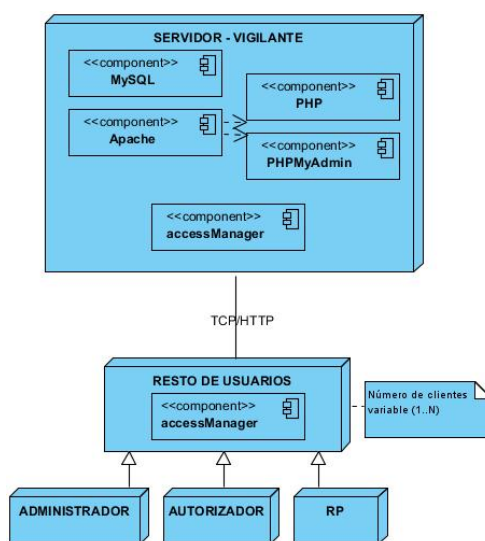


Diagrama de despliegue inicial del sistema

Donde:

- *Servidor - Vigilante*: Sería el servidor del sistema, instalado en la máquina que utilizará el vigilante para realizar sus labores. No creemos conveniente asignar una máquina con la única función de ejercer de servidor, ya que las características del sistema no son complejas y la concurrencia de usuarios es baja.
 - *MySQL*: Servidor de bases de datos MySQL.
 - *Apache*: Servidor web.
 - *PHP*: Módulo de PHP necesario para que Apache pueda mostrar páginas desarrolladas en ese lenguaje.
 - *phpMyAdmin*: Herramienta escrita en PHP para manejar la administración de MySQL a través del navegador Web.
 - *accessManager*: Aplicación Java a desarrollar.

accessManager · Arquitectura preliminar · Página 7 de 8

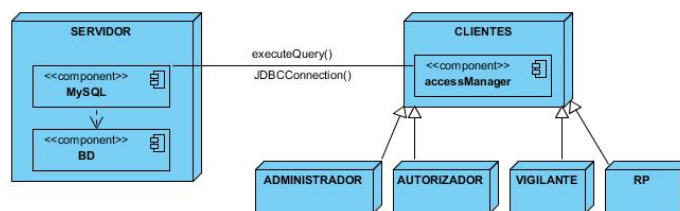


- *Resto de usuarios*: Refiriéndose al resto de usuarios que utilizarán el sistema, que son administradores, autorizadores y responsables de personal. Ubicados en cualquier localización geográfica accederán al servidor a través de la red mediante accessManager.

3. Análisis por capas

El estudio de capas del sistema ayuda a clarificar su estructura y ver cómo interactúan entre ellas. En nuestro caso, podemos observar una arquitectura de dos capas:

- Capa de aplicación / Cliente
Alojada físicamente en el ordenador del usuario. Se trata de la aplicación accessManager, con la que manipularemos todos los datos.
- Capa de datos / Servidor
Máquina en la que corren todos los servidores necesarios para ofrecer la funcionalidad deseada.



Análisis por capas

Pamplona, a 2 de febrero de 2010.

Héctor Napal Basterrica
Responsable del proyecto



accessManager

Desarrollo de un sistema de control de accesos a un centro

Planificación temporal y costes detallados del proyecto

Fecha de entrega: 5 de febrero de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Estructura del documento

2. Fases

3. Objetivos, artefactos e hitos

4. Calendarización

- 4.1. Calendario del equipo de desarrollo
- 4.2. Calendario del cliente

5. Estimación detallada de costes económicos



1. Introducción

1.1. Propósito

Realizamos la redacción de este documento para asentar en buena medida la planificación de todo el proceso de desarrollo a realizar para obtener el producto software deseado, así como una serie de premisas que deberán tenerse en cuenta. Dicha planificación se basará en la descripción de las fases que tendrán lugar, junto con los hitos que se desean obtener en cada una de ellas, y dos calendarios: uno con las actividades que las componen y otro con las actividades que el cliente deberá conocer y en las que deberá tomar parte.

1.2. Audiencia

Se trata de un documento vital para la correcta gestión del proyecto por nuestra parte, careciendo de valor buena parte del mismo para el cliente. En consecuencia, no se proporcionará este artefacto al cliente, pero si su calendario antes descrito.

1.3. Estructura del documento

Iniciamos el presente escrito con una explicación de las fases de las que constará el proceso de desarrollo, realizando una estimación del tiempo que conllevará cada una, así como un reconocimiento de los objetivos, artefactos e hitos que poseen.

Plasmamos toda esta definición de elementos en un calendario, dividiendo las fases en iteraciones y éstas en todos los flujos de trabajo fundamentales que desarrollarán. La descomposición de cada uno de éstos en actividades es llevada a cabo en cada uno de los documentos de iteración.

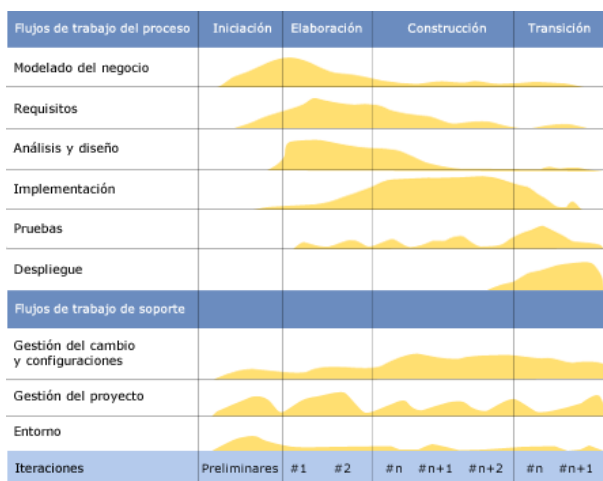
Además, en cada fase pueden observarse actividades en las que el cliente deberá tomar parte, lo que hace que finalicemos este documento derivando del calendario general otro específico para el cliente, el cual se lo entregaremos y explicaremos.

2. Fases

El proceso de desarrollo constará de cuatro fases divididas en iteraciones, cada una de las cuales desarrollará en menor o mayor medida las *Disciplinas o flujos de trabajo fundamentales* características del *Proceso Unificado Racional*. Este proceso iterativo guiado por los casos de uso implementará el sistema de forma incremental, generando los artefactos de manera temprana, refinándolos con cada iteración y llegando al final del mismo con todos completados.



La siguiente figura representa de manera muy significativa todo el proceso de desarrollo, relacionando fases y disciplinas:



Proceso de desarrollo RUP

Como bien se apuntó en el estudio inicial, la distribución aproximada de tiempos en nuestro caso es la siguiente:

FASE	DURACIÓN
Estudio de oportunidad (inception)	1'5 semanas
Elaboración	3 semanas
Construcción	7'5 semanas
Transición	1'5 semanas
TOTAL	13'5 semanas

Estimación temporal inicial de las fases RUP

Ésta estimación varía para ajustarse a todas las iteraciones que debemos realizar, quedando de la siguiente forma:

FASE	DURACIÓN
Estudio de oportunidad (inception)	1 semana
Elaboración	3 semanas
Construcción	8 semanas
Transición	2 semanas
TOTAL	14 semanas

Coste temporal final de las fases RUP



3. Objetivos, artefactos e hitos

Cada fase presenta unos objetivos a cumplir, unos artefactos a generar (iniciar o refinar) y un hito que marca el final de la misma y cuyas metas deben ser satisfechas a través de todo el trabajo desarrollado. Resumimos todo ello en la siguiente tabla:

FASE	METAS
Estudio de oportunidad (inception)	<p>Objetivos:</p> <ul style="list-style-type: none">• Establecimiento del ámbito y objetivos del proyecto, determinando la funcionalidad y capacidades del producto.• Plan de proyecto y coste estimado. <p>Artefactos:</p> <ul style="list-style-type: none">• Examen inicial de la viabilidad del proyecto software, en el que como su nombre bien indica, se realiza un examen inicial del proyecto, así como de su viabilidad. Su aprobación determinará si éste se desarrolla o no lo hace por no ser viable o atractivo desde el punto de vista coste-beneficio. <p>Hito:</p> <ul style="list-style-type: none">• Objetivos: Cliente y equipo de desarrollo aprueban el proyecto y obtienen una visión global común del problema y su solución.

Principales objetivos, artefactos e hitos de la fase de Estudio de oportunidad (inception)



FASE	METAS
Elaboración	<p>Objetivos:</p> <ul style="list-style-type: none">• Estudio en profundidad del dominio del problema y de la funcionalidad que deberá tener el futuro sistema, estableciendo la línea base de la arquitectura.<ul style="list-style-type: none">○ Especificación de la mayor parte de los casos de uso.○ Definición de la arquitectura básica del sistema (escenarios de funcionamiento posibles).○ Planificación temporal detallada del proyecto. <p>Artefactos:</p> <ul style="list-style-type: none">• Estudio detallado del problema software, en el que se profundiza en el ámbito del problema, refinando la funcionalidad que deberá tener.• Especificación del modelo de negocio, para comprender los procesos de negocio de la organización, mostrando su contexto y límites.• Especificación inicial de los casos de uso, que sirve para averiguar qué se debe construir; qué debe y qué no debe hacer el sistema.• Arquitectura preliminar, con un mapeo inicial entre arquitectura y software a desplegar.• Planificación temporal del proyecto, que comprende una planificación temporal precisa de todo el proceso de desarrollo, a nivel interno (equipo de desarrollo) y externo (cliente).• Plan de pruebas, donde se detalla las pruebas que tendrán lugar para verificar y validar el sistema construido, así como los resultados obtenidos y qué conclusiones se pueden obtener de ellos. <p>Hito:</p> <ul style="list-style-type: none">• Arquitectura: Se dispone de la arquitectura base que tendrá el sistema a desarrollar, siendo refinada e implementada en la siguiente fase.

Principales objetivos, artefactos e hitos de la fase de Elaboración



FASE	METAS
Construcción	<p>Objetivos:</p> <ul style="list-style-type: none">• Refinamiento iterativo e incremental de la arquitectura básica hasta obtener el producto software deseado; lo que viene a ser: transformar la línea base de la arquitectura en un sistema completo (producto y toda su documentación asociada). <p>Cada iteración:</p> <ul style="list-style-type: none">○ Establece sus objetivos en función de la evaluación de las iteraciones precedentes.○ Corresponde a uno o más casos de uso.○ Reproduce el ciclo de vida tradicional a menor escala, constando de las actividades de:<ul style="list-style-type: none">▪ Análisis▪ Diseño▪ Codificación y pruebas▪ Evaluación de la entrega ejecutable▪ Preparación de la entrega <p>Tras estas actividades se lleva a cabo una evaluación de la release elaborada junto con el cliente, aprobando el desarrollo o proponiendo mejoras, nuevos requisitos,... Este hecho, que se repetirá en muchas de las iteraciones, consta de gran importancia en la:</p> <ul style="list-style-type: none">• Especificación incremental e iterativa de requisitos.• Verificación y validación continuada del producto. <p>Artefactos (con cada iteración):</p> <ul style="list-style-type: none">• Ejecutable (prototipo) y documentación de la release. <p>Hito:</p> <ul style="list-style-type: none">• Capacidad operacional: Se obtiene un sistema con toda la funcionalidad requerida por el cliente.

Principales objetivos, artefactos e hitos de la fase de Construcción



FASE	METAS
Transición	<p>Objetivos:</p> <ul style="list-style-type: none">• Liberación del producto y entrega al usuario para su uso final. Consta de:<ul style="list-style-type: none">○ Instalación, configuración y prueba de todo el software.○ Entrega y explicación de toda la documentación del proyecto al cliente.○ Entrenamiento y explicación de los manuales a los usuarios.○ Soporte y mantenimiento futuro opcional (a negociar con el cliente). <p>Artefactos:</p> <ul style="list-style-type: none">• Producto software final (ejecutables y documentación del proyecto).• Documentos y manuales del sistema. <p>Hito:</p> <ul style="list-style-type: none">• Liberación del producto: El producto ha sido liberado y el cliente ya puede disfrutar de él.

Principales objetivos, artefactos e hitos de la fase de Transición

4. Calendarización

4.1. Calendario del equipo de desarrollo

En la siguiente hoja se expone el calendario establecido de cara a desarrollar todo el sistema software. Para evitar mostrar un calendario excesivamente detallado lo que haremos será crear uno con las fases y actividades principales e instar al lector a que pueda examinarlo con mayor detalle recorriendo las diversas planificaciones realizadas en cada iteración.

La división de la fase de Construcción ha sido realizada en función de las pestañas del JTabbedPane que compondrá el principal Frame de la aplicación, agrupando los casos de uso en éstas y desarrollándolas en orden decreciente según la importancia de los mismos.

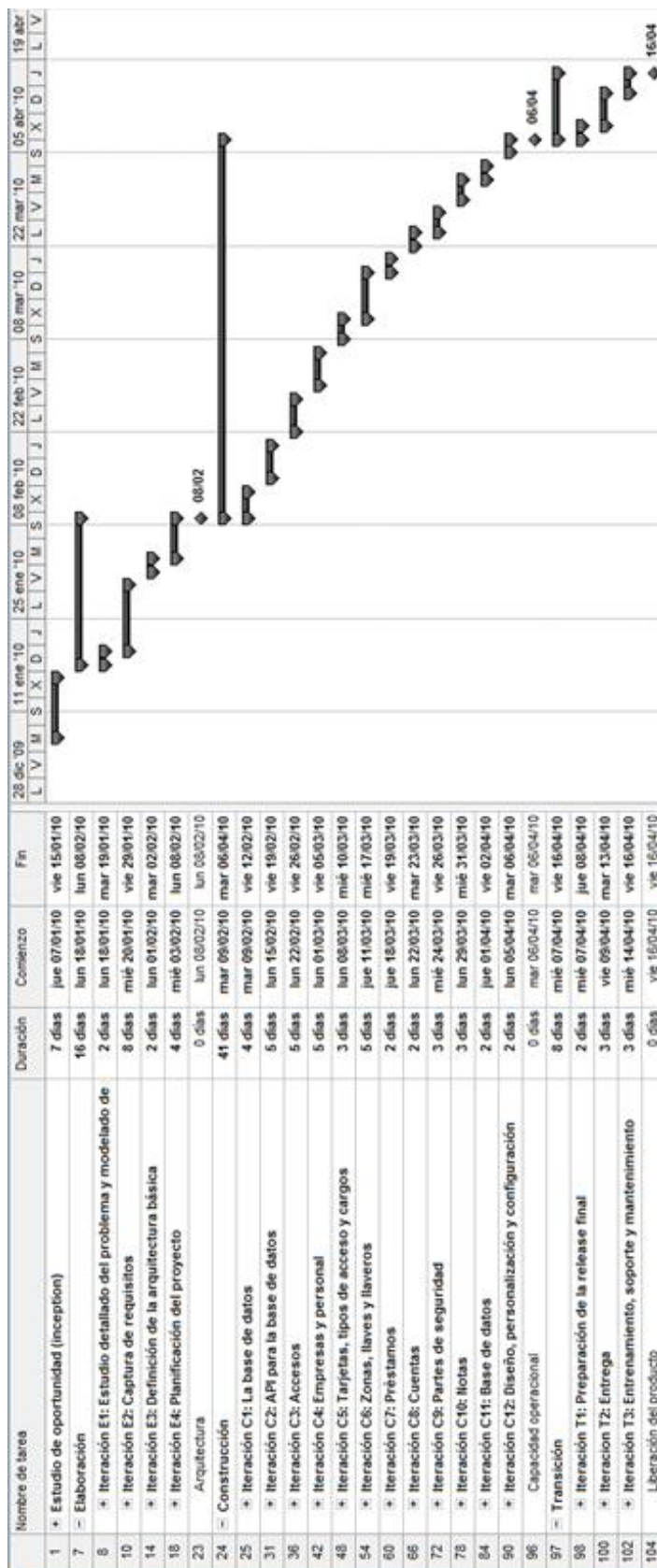


Diagrama de Gantt



4.2. Calendario del cliente

Al emplear el proceso de desarrollo RUP, la participación del cliente en el desarrollo deberá ser continuada, de modo que podamos refinar los requisitos y recibir la aprobación de éste constantemente, reduciendo al mínimo el riesgo de obtener un producto indeseado. Este será su calendario por fases:

CALENDARIO DEL CLIENTE				
FASE	ITERACIÓN	EVENTO	MOTIVO	FECHA
Estudio de oportunidad (inception)	Única	Definición del problema	Reunión preliminar entre el analista de viabilidad y el cliente en la que éste comunicará su intención de contratar los servicios de Software Solutions para que resuelva el problema al que se enfrenta. Ambos deberán adquirir una visión común de éste.	07/01/10
		Estudio inicial de viabilidad	Entrega y explicación al cliente del estudio inicial de viabilidad del sistema llevado a cabo.	14/01/10
		Aprobación o rechazo del proyecto	Tras un día de evaluación y reflexión por parte del cliente, el analista y éste se vuelven a reunir para solucionar dudas y que el cliente decida aprobar o rechazar el desarrollo del proyecto.	15/02/10

Calendario del cliente



FASE	ITERACIÓN	EVENTO	MOTIVO	FECHA
Elaboración	E1	Comprensión del problema	Responsable de proyecto y cliente realizan una tercera reunión para delimitar y detallar perfectamente todo el problema software, consiguiendo ambos una misma visión del mismo y sirviendo de base para que el primero cree el documento <i>Estudio detallado del problema software</i> .	18/01/10
	E2	Aproximación al modelo inicial de casos de uso	Se muestra al cliente un primer boceto de lo que será el modelo inicial de casos de uso, para que pueda evaluarlo y sugerir modificaciones.	27/01/10
		Muestra, debate y aprobación de la especificación inicial de los casos de uso	Se rediseña la <i>Especificación inicial de los casos de uso</i> , volviendo a mostrársela al cliente. Se evalúa el texto con la intención de que el cliente lo apruebe y se disponga de una base sobre la que comenzar a trabajar. No es necesario recordar que dicho modelo se completará a lo largo del ciclo de desarrollo, variando su contenido.	29/01/10
	E4	Calendario de actuación y presupuesto detallado	Entrega y explicación al cliente del calendario con las principales actividades en las que deberá participar para asistir al equipo de desarrollo, ya sea evaluando algún artefacto o prototipo, aprobando decisiones,... además del presupuesto económico total estimado para la realización del proyecto software.	08/02/10

Calendario del cliente



FASE	ITERACIÓN	EVENTO	MOTIVO	FECHA
Construcción	C3 a C12	Evaluación del prototipo	Evaluación y prueba conjunta del prototipo elaborado, sugiriendo cambios (refinamiento) en la especificación de casos de uso y dando el visto bueno para continuar con el proyecto si el cliente está satisfecho.	26/02/10 05/03/10 10/03/10 17/03/10 19/03/10 23/03/10 26/03/10 31/03/10 06/04/10
Transición	T2	Entrega e instalación	Entrega e instalación del sistema desarrollado, así como la transmisión y explicación de toda la documentación asociada.	09/04/10
		Pruebas de aceptación	Prueba del sistema en el lugar de instalación. Se probarán los requisitos de usuario, verificando que todos los casos de uso han sido incorporados y que el sistema es el que el cliente realmente quiere y necesita.	12/04/10 13/04/10
	T3	Entrenamiento	Charla con cada grupo de usuarios del sistema en la que se les mostrará cómo operar con el sistema.	14/04/10
		Soporte	Explicación en profundidad de la estructura del sistema al cliente y personal encargado de su mantenimiento, de modo que puedan comprenderlo de cara a futuras actividades.	15/04/10
		Mantenimiento	En caso de que el cliente nos otorgue la tarea del mantenimiento del sistema en el tiempo, realizaremos una reunión en la que estableceremos todas las actividades que se llevarán a cabo, su coste,...	16/04/10

Calendario del cliente



5. Estimación detallada de costes económicos

Con el proyecto a punto de entrar en la fase de Construcción y toda la información básica generada, realizamos la siguiente estimación de recursos económicos, mucho más detallada y precisa que la que generamos en *Estudio inicial de la viabilidad del proyecto software* y haciéndola llegar al cliente:

MANO DE OBRA			
CARGO	TIEMPO	SALARIO / SEMANA	TOTAL
Director del proyecto – Ingeniero de desarrollo y soporte	14 semanas	375 €	5250 €
Subtotal mano de obra			5250 €
MATERIA PRIMA			
MATERIA	CANTIDAD	VALOR / 5 AÑOS	TOTAL
PC's ambiente de desarrollo	1	499 €	26,87 €
Licencias software	1	4008,65 €	215,85 €
Microsoft Windows 7 – Professional Edition	1	309 €	16,64 €
Adobe Acrobat Professional	1	648,44 €	34,92 €
Office Word 2007	1	289 €	15,56 €
Office Project Pro 2007	1	1299 €	69,94 €
Visual Paradigm Suite 4.2	1	1463,21 €	78,79 €
Servidor	1	299 €	479 €
Acer X38109 + Hacer X233HB TFT	1	479 €	479 €
Subtotal mano de obra			721,72 €
TOTAL			5821,64 €

Costes estimados derivados del desarrollo del proyecto

Tal y como podemos observar el primer presupuesto que realizamos fue bastante preciso ya que difiere mínimamente del realizado anteriormente.

Pamplona, a 5 de febrero de 2010.

Héctor Napal Basterrica
Responsable del proyecto



accessManager
Desarrollo de un sistema de control de accesos a un centro

Plan de pruebas inicial

Fecha de entrega: 29 de enero de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Estructura del documento

2. Requerimientos de las pruebas

3. Estrategias de prueba

4. Recursos

5. Concreción de las pruebas



1. Introducción

1.1. Propósito

Este documento pretende ser un resumen de todo el plan de pruebas llevado a cabo en el desarrollo de accessManager, recogiendo especificaciones, resultados y conclusiones obtenidas en la realización de las pruebas.

Los principales objetivos que nos planteamos en este artefacto son:

- Identificar los elementos que se deben probar.
- Definir estrategias a seguir para realizar las pruebas.
- Estimar los recursos necesarios para llevar a cabo el proceso de pruebas.
- Recoger los resultados obtenidos en cada una de las pruebas, junto con una evaluación de éstos.

No debemos olvidar que la finalidad última de este documento es la de probar que toda la especificación de requisitos ha sido implementada por el equipo de desarrollo, creando con ello un producto válido para las pretensiones del cliente.

1.2. Audiencia

Este es un documento que consta de mucho valor para el cliente, pues es una fiel prueba de la calidad del producto que se encuentra en desarrollo y que obtendrá en el futuro. Será mostrado al cliente a lo largo de todas las iteraciones, demostrándole que los diversos prototipos son desarrollados con unos niveles aceptables de calidad y que toda la funcionalidad implementada se ejecuta de manera correcta y eficiente, según lo establecido.

1.3. Estructura del documento

En los tres primeros capítulos realizamos una breve introducción al proceso de pruebas, identificando qué tipo de pruebas debemos realizar, los pasos que debemos seguir y qué recursos necesitamos, respectivamente.

A continuación mostrando una tabla que relaciona los elementos a probar en cada iteración con los documentos que recogen las pruebas ejecutadas, poniendo punto y final al documento con una enumeración de todas las pruebas realizadas y los resultados y conclusiones que hemos obtenido de ellas.¹

¹ La enumeración es llevada a cabo en el documento *Plan de pruebas* incluido en los artefactos que acompañan al tercer capítulo de la presente memoria.



2. Requerimientos de las pruebas

Es buena práctica comenzar enumerando todos los tipos de pruebas que debemos realizar:

PRUEBAS			
TIPO	INSTANTE	OBJETIVO	DESCRIPCIÓN
De unidad	Con cada componente software desarrollado.	Probar la especificación de los elementos software.	Pruebas funcionales (de caja negra) y estructurales (de caja blanca) que validen la eficacia funcional de cada componente desarrollado.
De integración	Con cada componente software desarrollado.	Probar la documentación de diseño y arquitectura.	Validar la eficacia funcional del sistema y subsistemas que lo componen.
De sistema	Al finalizar cada iteración.	Probar la especificación de requisitos del software.	Comprobar que se han incorporado todos los casos de uso, que el funcionamiento es correcto y robusto y que la documentación a entregar al usuario es la adecuada.
De aceptación	En la segunda iteración de la fase de Transición.	Probar los requisitos de usuario (contrato).	Evaluar conjuntamente con el cliente el producto, haciéndole ver que es el apropiado.

Tipos de pruebas a realizar

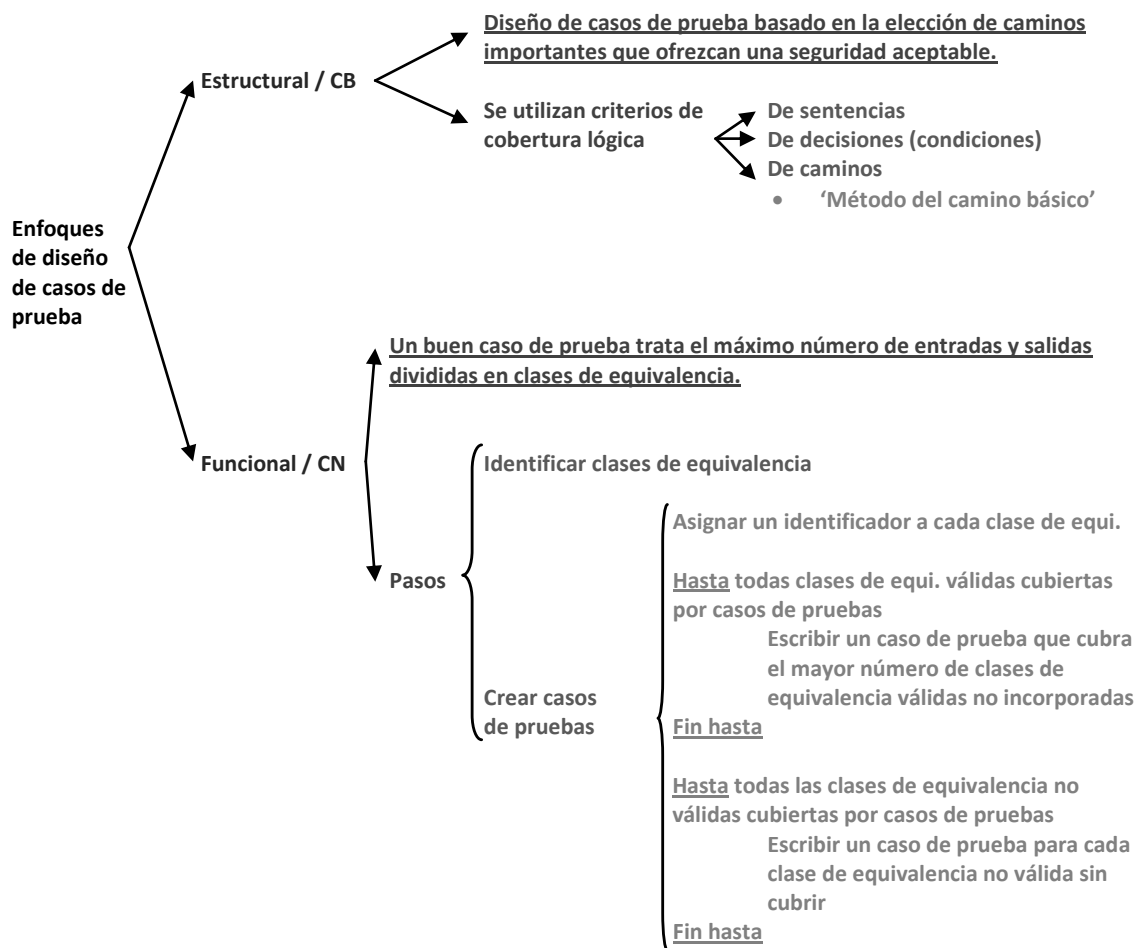


3. Estrategias de prueba

En la sección anterior hemos descrito qué tipos de pruebas debemos realizar, con que ahora es el turno de definir cómo las realizaremos. A continuación presentamos los procedimientos y técnicas que emplearemos en el desarrollo de las mismas.

PRUEBAS	
TIPO	PROCEDIMIENTOS Y TÉCNICAS
De unidad	<p>Cada elemento software será probado individualmente mediante los enfoques de diseño de casos de prueba usuales y que esquematizamos a continuación de esta tabla.</p> <p>El enfoque práctico será el siguiente:</p> <ol style="list-style-type: none">1. Si existen combinaciones de entrada → Crear grafos causa-efecto y añadir los casos de prueba oportunos.2. Identificar las clases de equivalencia empleando la técnica de análisis de valores límite. Añadir los casos de prueba no incluidos anteriormente.3. Añadir casos de prueba mediante la técnica de conjetura de errores.4. Ejecutar los casos de prueba generados hasta el momento (de caja negra) y analizar la cobertura obtenida.5. Examinar la lógica del programa para añadir los casos de prueba (de caja blanca) precisos para cumplir el criterio de cobertura elegido.
De integración	<p>El desarrollo de las pruebas de integración y de sistema se realizarán durante todas y cada una de las iteraciones de construcción del producto, teniendo la siguiente secuencia de actividades:</p> <ol style="list-style-type: none">1. Planificar las pruebas de cada iteración.<ol style="list-style-type: none">1.1. <i>De integración (para cada construcción).</i>1.2. <i>De sistema (al final de cada iteración).</i>2. Diseñar las pruebas para cada construcción (caso de uso).<ol style="list-style-type: none">2.1. <i>Casos de prueba.</i>2.2. <i>Procedimientos de prueba.</i>
De sistema	<ol style="list-style-type: none">3. Implementar las pruebas.4. Ejecutar las pruebas.5. Evaluar los resultados obtenidos.
De aceptación	<p>Se prueban todos y cada uno de los casos de uso con el cliente, con datos válidos y erróneos y mostrándole que el sistema se comporta según lo previsto.</p>

Procedimientos y técnicas a utilizar en las pruebas



4. Recursos

Estos son los recursos necesarios para llevar a cabo todo el desarrollo de las pruebas:

HARDWARE			
TIPO	CANTIDAD	TIPO DE PRUEBA	DESCRIPCIÓN
PC	1	Todas	Computadora con la que poder realizar el diseño e implementación de las pruebas.
PC	11	De sistema	Ordenadores para poder realizar la prueba final de todo el producto en unas condiciones de concurrencia máxima.

Recursos hardware



HUMANOS		
ROL	CANTIDAD	RESPONSABILIDADES
Diseñador	1	Identificar, priorizar, diseñar e implementar los casos de prueba, recogiendo todo en el 'Plan de pruebas'.
Probador (tester)	12	Ejecutar las pruebas, recogiendo los resultados obtenidos y realizando una evaluación de los mismos.

Recursos humanos

5. Concreción de las pruebas

Una vez establecidas qué pruebas se deben realizar, cómo deben ser desarrolladas y los recursos necesarios, pasamos a concretarlas en los artefactos que las desarrollarán:

PRUEBAS			
ITERACIÓN	ELEMENTOS A PROBAR	TIPOS DE PRUEBAS	DOCUMENTO ASOCIADO
C1	Implementación de la base de datos.	De unidad.	<i>Iteración C1: La base de datos</i>
C2	Implementación de la API para la base de datos.	De unidad.	<i>Iteración C2: API para la base de datos</i>
C3	Todos los casos de uso referentes a los accesos al centro.	De unidad, integración y sistema.	<i>Iteración C3: Accesos</i>
C4	Ident. con empresas y personal.	De unidad, integración y sistema.	<i>Iteración C4: Empresas y personal</i>
C5	Ident. con tarjetas, colores, niveles, tipos de acceso y cargos.	De unidad, integración y sistema.	<i>Iteración C5: Tarjetas, tipos de acceso y cargos</i>
C6	Ident. con zonas, llaves y llaveros.	De unidad, integración y sistema.	<i>Iteración C6: Zonas, llaves y llaveros</i>
C7	Ident. con préstamos.	De unidad, integración y sistema.	<i>Iteración C7: Préstamos</i>
C8	Ident. con cuentas.	De unidad, integración y sistema.	<i>Iteración C8: Cuentas</i>
C9	Ident. con partes de seguridad.	De unidad, integración y sistema.	<i>Iteración C9: Partes de seguridad</i>

Iteraciones, elementos, tipos de pruebas y documentos asociados



C10	Ident. con notas.	De unidad, integración y sistema.	<i>Iteración C10: Notas</i>
C11	Ident. con base de datos.	De unidad, integración y sistema.	<i>Iteración C11: Base de datos</i>
C12	Comprobar que el diseño es el correcto, que la muestra de los componentes de la aplicación se realiza acorde al tipo de usuario registrado y que los detalles de configuración tienen efecto.	De integración y sistema.	<i>Iteración C12: Diseño, personalización y configuración</i>
T1	Asegurar la idoneidad de toda la documentación y software elaborado.	---	<i>Iteración T1: Preparación de la release final</i>
T2	Probar toda la aplicación con el cliente, creándole confianza en que el software desarrollado está capacitado para dar respuesta a sus pretensiones.	De aceptación.	<i>Iteración T2: Entrega</i>
T3	Asegurar que el entrenamiento realizado da lugar a operadores que saben manejar a la perfección el sistema, además de proporcionarles manuales que puedan ayudarlos en futuras dudas.	---	<i>Iteración T3: Entrenamiento, soporte y mantenimiento</i>

Iteraciones, elementos, tipos de pruebas y documentos asociados

Hasta aquí llega el *Plan de pruebas inicial*, en el que hemos establecido las bases de lo que serán las pruebas de aseguramiento de calidad del software. En caso de que el lector desee observar de manera extendida las pruebas realizadas, le aconsejamos que consulte los artefactos enumerados en la tabla anterior. Para una versión simplificada de todas las pruebas realizadas, diríjase al capítulo donde se recogen los principales documentos de este desarrollo software y consulte el artefacto *Plan de pruebas*.



Pamplona, a 29 de enero de 2010.

Héctor Napal Basterrica
Responsable del proyecto

Capítulo 2

Construcción

2.1. Objetivos de la fase

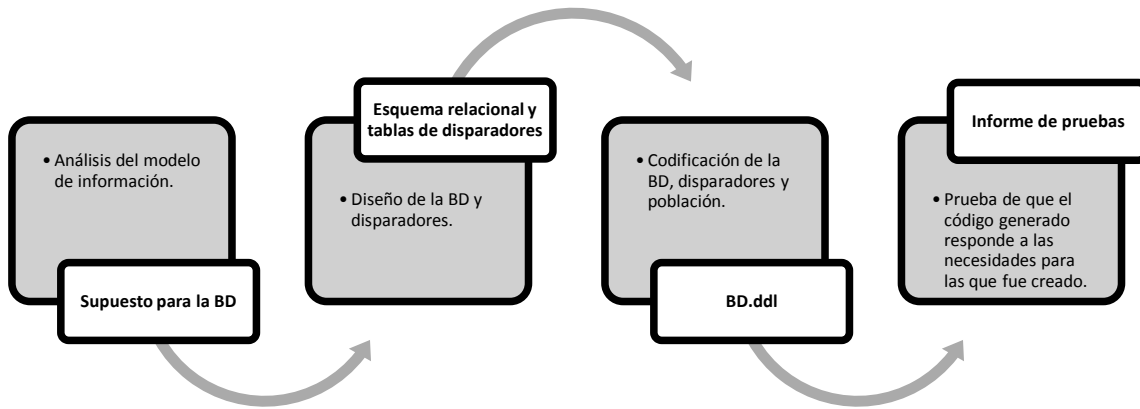
Llegamos a la fase de construcción, fase sucesora a las de Estudio de oportunidad (inception) y Elaboración, en la que elaboraremos el producto software que dará respuesta a todas las necesidades y requisitos que hemos venido describiendo. Nos encontramos ante un proceso complejo, en el que pasaremos de la *Línea base de la arquitectura* a un producto software completo, y el cual documentamos en las siguientes páginas. Cabe recordar que estamos empleando el proceso de desarrollo *RUP*, que guía la construcción en iteraciones; iteraciones que describimos a continuación y explicamos con total detalle en los posteriores artefactos.

Antes de entrar en materia debemos aclarar algunos puntos:

- Los artefactos han sido elaborados de modo que para cada...
 - ... iteración:
 - Se debería realizar un estudio de los riesgos que conlleva. En nuestro caso éste ha sido documentado para las dos siguientes iteraciones, pero obviado en el resto debido a la similitud de los riesgos que conllevan todas las demás. Esta parte de documentación ha sido suprimida para facilitar la lectura de esta memoria, pero toda iteración ha conllevado su respectivo estudio de riesgos.
 - .. caso de uso se describan:
 - Interfaces gráficas necesarias.
 - Estrategias a seguir.
 - Métodos a añadir a la futura API y qué funcionalidad deben proporcionar.
 - Diagramas de actividades y de secuencia que aclaren y describan con mayor detalle las estrategias mencionadas. Aquí hemos omitido toda llamada necesaria para mostrar los datos en la futura interfaz gráfica con la intención de simplificar los diagramas y hacer que el lector atienda a los métodos realmente importantes (los que generan la información, distintos todos ellos, no los que la muestran, que son siempre los mismos).

2.2. Iteración C1: La base de datos

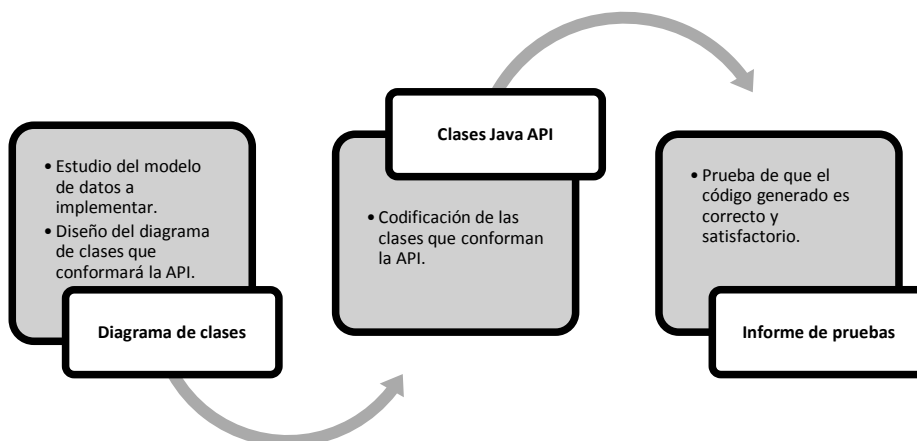
Comenzamos la Construcción creando los cimientos que sustenten todo el posterior desarrollo. En nuestro caso eso implica elaborar una base de datos que permita recoger todo el modelo de datos que entrará en juego posteriormente. Por tanto, dedicaremos la primera iteración a analizar el modelo de información, diseñando y codificando una base de datos, acompañándola de una serie de disparadores y una población base y probando que todo el trabajo realizado es correcto.



Iteración C1: La base de datos - Proceso

2.3. Iteración C2: Una API para la base de datos

El siguiente paso es crear una API en lenguaje Java que nos permita manejar la base de datos, permitiendo una cómoda programación de la aplicación y produciendo un mantenimiento futuro eficiente y sencillo. La idea es representar el mismo modelo que representamos con el esquema relacional pero a través de uno orientado a objetos. Cada clase se corresponderá con una o varias tablas de la base de datos, ofreciendo métodos de alto nivel que proporcionen toda la funcionalidad DML. Más adelante se añadirán a éstos métodos que permitan desarrollar toda la funcionalidad de la aplicación, por tanto debemos tener en cuenta que se trata de un diagrama inicial que será extendido en futuras iteraciones.



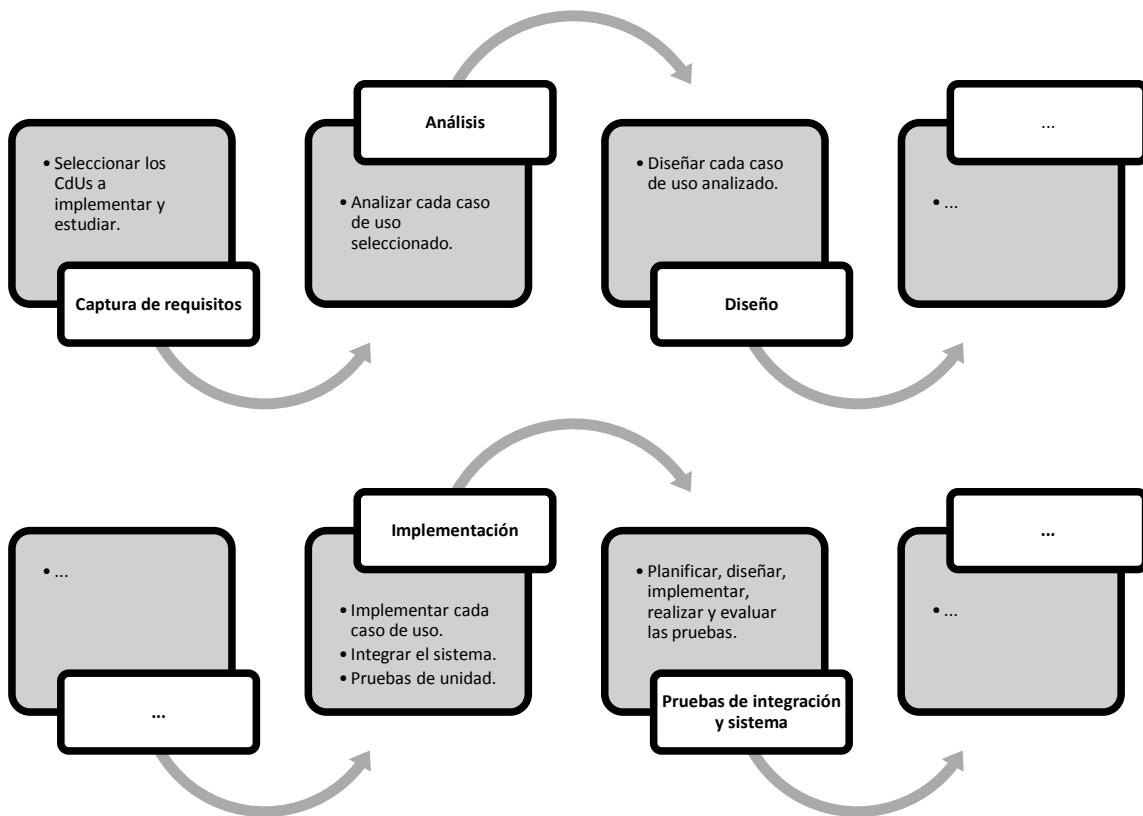
Iteración C2: Una API para la base de datos - Proceso

2.4. Iteraciones C3 a C11: Implementación de los casos de uso

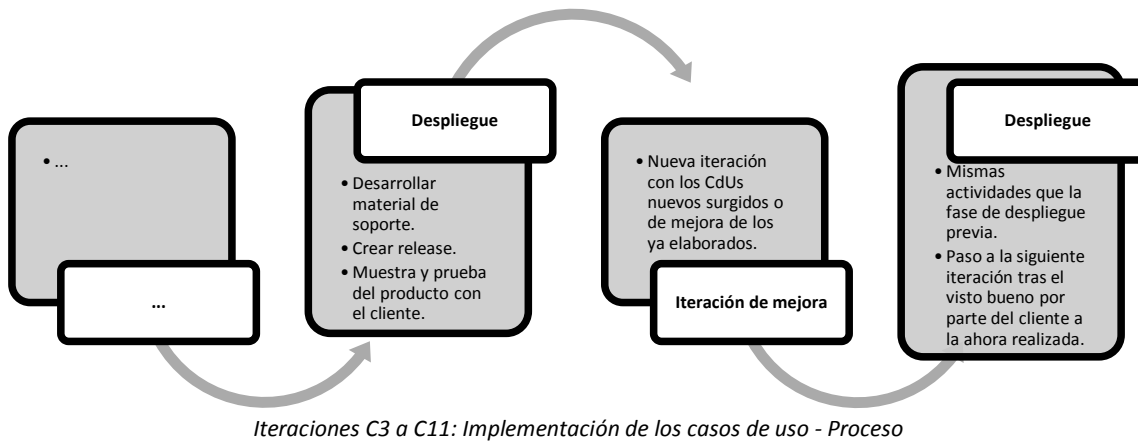
Una vez que disponemos de la base de datos y la API que la maneja, es hora de analizar, diseñar, implementar y probar todos y cada uno de los casos de uso (CdUs) descritos en el modelo de casos de uso. Recordemos que en el hicimos una agrupación de éstos en función de los entes que trataban y de la importancia de los mismos. Ese ese el orden es el que seguiremos en todo el desarrollo, realizando en cada iteración un grupo y el consiguiente el refinamiento. Recordamos todas las iteraciones brevemente:

- C3: Accesos
- C4: Mejora de Accesos
- C5: Empresas y personal
- C6: Mejora de Empresas y personal
- C7 : Tarjetas, tipos de accesos y cargos
- C8: Mejora de Tarjetas, tipos de acceso y cargos
- C9: Zonas, llaves y llaveros
- C10: Mejora de Zonas, llaves y llaveros
- C11: Préstamos
- C12: Mejora de Préstamos
- C13: Cuentas
- C14: Mejora de cuentas
- C15: Partes de seguridad
- C16: Mejora de Partes de seguridad
- C17: Notas
- C18: Mejora de Notas
- C19: Base de datos
- C20: Mejora de Base de datos

Para cada uno de los grupos tenemos el siguiente proceso iterado...



Iteraciones C3 a C11: Implementación de los casos de uso - Proceso



... que irá generando clases y extendiendo la API diseñada en la iteración C2¹.

2.5. Iteración C12: Diseño, personalización y configuración

Tras haber desarrollado toda la funcionalidad acordada con el cliente a través del documento de especificación de los casos de uso, acometeremos actividades de diseño de la aplicación, personalizaremos la misma en función de los diversos tipos de usuarios que existirán y realizaremos los últimos detalles necesarios de configuración. Al finalizar esta iteración habremos obtenido el producto software final, teniendo todo listo para pasar a la siguiente y última fase del proceso de desarrollo: la de Transición.

¹ Para evitar sobrecargar esta memoria, se explicarán los métodos creados en cada iteración sin ver visualmente cómo van creciendo las clases. El resultado de la API final podrá ser consultada en la sección de *Principales artefactos*.



accessManager

Desarrollo de un sistema de control de accesos a un centro

Iteración C1: La base de datos

Fecha de entrega: 12 de febrero de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Definiciones, acrónimos y abreviaciones
- 1.4. Estructura del documento

2. Requisitos

- 2.1. Planificación de la iteración
- 2.2. Gestión de riesgos

3. Análisis y diseño

- 3.1. Estudio del dominio de información ('Supuesto')
- 3.2. Diseño de la base de datos
- 3.3. Diseño de los disparadores

4. Implementación y pruebas

- 4.1. Codificación e integración
- 4.2. Pruebas

5. Conclusiones



1. Introducción

1.1. Propósito

El propósito con el que hemos elaborado este escrito es el de documentar con un nivel de detalle aceptable todas las actividades realizadas a lo largo de la iteración C1; iteración cuyo objetivo es el análisis, diseño e implementación de todo el modelo de datos que dará soporte al sistema.

1.2. Audiencia

Se trata de un documento de carácter interno hasta el final del desarrollo.

1.3. Definiciones, acrónimos y abreviaciones

- | | | | |
|---|----|-------------------------|----------------|
| • | BD | Base de datos | |
| • | RP | Responsable de personal | |
| • | PK | Primary key | Clave primaria |
| • | FK | Foreign key | Clave foránea |

1.4. Estructura del documento

Como paso inicial en toda iteración, comenzamos planificándola y realizando un estudio de los riesgos que conllevará y cómo los afrontaremos.

A continuación pasamos a estudiar el dominio de información que deberá manejar el sistema, diseñando una base de datos que será codificada, integrada y probada en el posterior apartado.

Para finalizar incluimos una breve sección de conclusiones en la que evaluamos el desarrollo y resultados que hemos conseguido tras la iteración, decidiendo continuar con el proceso de desarrollo o rehacer parte del camino andado.



2. Requisitos

2.1. Planificación de la iteración

Debido a su importancia vital, dedicaremos una minuciosa y detallada iteración al estudio y desarrollo del modelo de datos. Éstas son las tareas de las que consta, los pasos y herramientas a emplear, además de los artefactos que deberán obtenerse:

PLAN DE ITERACIÓN C1: LA BASE DE DATOS			
ASPECTO	DESCRIPCIÓN		
Objetivos	Diseño e implementación de la BD.		
Artefactos de entrada	<ul style="list-style-type: none"> • <i>Estudio detallado del problema software</i> • <i>Especificación del modelo de negocio</i> • <i>Especificación inicial de los casos de uso</i> • <i>Planificación temporal del proyecto</i> 		
Artefactos modificados o creados	<ul style="list-style-type: none"> • El presente documento, que aporta entre otras cosas el esquema relacional de la BD y la descripción de los disparadores que dotarán de funcionalidad a la misma. • Ficheros BD.dll y disparadores.sql con la implementación de las tablas y disparadores, así como una población que permita simular el uso real de la base de datos. • Pruebas documentadas en el <i>Plan de pruebas</i>. 		
Calendarización detallada	Requisitos + Pruebas	09/02/10	<ul style="list-style-type: none"> • Planificación de la iteración y gestión de riesgos. • Reunión con el cliente para estudiar el modelo de datos. • Planificación de las pruebas.
	Análisis + Diseño + Pruebas	10/02/10	<ul style="list-style-type: none"> • Análisis y diseño de la base de datos, disparadores y población de prueba. • Inicio de toda la documentación.
		11/02/10	<ul style="list-style-type: none"> • Diseño de pruebas.
	Implementación + Pruebas	12/02/10	<ul style="list-style-type: none"> • Codificación de todo el diseño. • Ejecución y evaluación de pruebas. • Finalización de toda la documentación.

Planificación de la iteración



Nombre de tarea	Duración	Comienzo	Fin	lun 08 feb	mié 10 feb	vie 12 feb
+ Estudio de oportunidad (inception)	7 días	jue 07/01/10	vie 15/01/10	0	0	0
+ Elaboración	16 días	lun 18/01/10	lun 08/02/10			
- Construcción	41 días	mar 09/02/10	mar 06/04/10			
- Iteración C1: La base de datos	4 días	mar 09/02/10	vie 12/02/10			
Captura de requisitos	1 día	mar 09/02/10	mar 09/02/10			
Análisis	1 día	mié 10/02/10	mié 10/02/10			
Diseño	1 día	jue 11/02/10	jue 11/02/10			
Implementación	1 día	vie 12/02/10	vie 12/02/10			
Pruebas	4 días	mar 09/02/10	vie 12/02/10			
+ Iteración C2: API para la base de datos	5 días	lun 15/02/10	vie 19/02/10			
+ Iteración C3: Accesos	5 días	lun 22/02/10	vie 26/02/10			
+ Iteración C4: Empresas y personal	5 días	lun 01/03/10	vie 05/03/10			
+ Iteración C5: Tarjetas, tipos de acceso y cargos	3 días	lun 08/03/10	mié 10/03/10			
+ Iteración C6: Zonas, llaves y llaveros	5 días	jue 11/03/10	mié 17/03/10			
+ Iteración C7: Préstamos	2 días	jue 18/03/10	vie 19/03/10			
+ Iteración C8: Cuentas	2 días	lun 22/03/10	mar 23/03/10			
+ Iteración C9: Partes de seguridad	3 días	mié 24/03/10	vie 26/03/10			
+ Iteración C10: Notas	3 días	lun 29/03/10	mié 31/03/10			
+ Iteración C11: Base de datos	2 días	jue 01/04/10	vie 02/04/10			
+ Iteración C12: Diseño, personalización y configuración	2 días	lun 05/04/10	mar 06/04/10			
Capacidad operacional	0 días	mar 06/04/10	mar 06/04/10			
+ Transición	8 días	mié 07/04/10	vie 16/04/10			

Recursos	Temporales	4 días	
	Humanos	1 persona	
	Software	<i>Sistema operativo</i>	Microsoft Windows 7
		<i>Ofimática</i>	Microsoft Word 2010 Adobe Acrobat Professional 8
		<i>Diseño</i>	Visual Paradigm 4.2
		<i>Codificación</i>	Notepad++ 5.6.6
		<i>Multiservidor web</i>	WampServer 2.0i
<i>Explorador web</i>	Mozilla Firefox 3.6		
Hardware	1 PC		

Planificación de la iteración



2.2. Gestión de riesgos

Estos son los riesgos y planes de contingencia elaborados para esta iteración:

IDENTIFICACIÓN DE RIESGOS		
TIPO	EXISTENCIA	DESCRIPCIÓN
Tecnología y herramientas	No se observan	Las herramientas a emplear en el diseño son un ordenador provisto del siguiente software: <ul style="list-style-type: none"> • Visual Paradigm Suite, para diseñar la BD. • Multiservidor web (WampServer), para chequear el correcto diseño y funcionamiento de la BD. • Microsoft Word y Adobe Acrobat, para la correcta documentación y redacción de toda la información. Se trata de un material sencillo que siempre tendremos disponible.
Requerimientos	Se observan	El aspecto cambiante de los requerimientos puede suponer rehacer el diseño. El cliente nos indica que el sistema que desean desarrollar es el especificado en los casos de uso y que éstos tienen pocas posibilidades de cambiar. No obstante, existe el riesgo de que éstos puedan cambiar y sea necesario modificar el modelo de datos inicial.
Estimación	Se observan	Se ha designado un total de cuatro días para la realización de ésta iteración. Se trata de un tiempo prudente, pero siempre cabe la posibilidad de que éste haya podido ser infraestimado a causa de la evaluación errónea de la complejidad del 'Supuesto'. Los recursos económicos disponibles suponen un pequeño riesgo si es necesaria una reestructuración de la calendarización causada por una infraestimación del tiempo necesario para esta iteración.

Identificación de riesgos

ANÁLISIS DE RIESGOS			
TIPO	RIESGOS POSIBLES	PROBABILIDAD	EFFECTOS
Estimación	Infraestimación del tiempo requerido, y por tanto de los recursos económicos necesarios.	Moderada	Tolerables
Requerimientos	El aspecto cambiante de los requerimientos puede suponer rehacer el diseño.	Baja	Tolerables – Serios

Análisis de riesgos



PLANES DE CONTINGENCIA	
TIPO	ESTRATEGIA
Estimación	En caso que sea necesario prolongar en el tiempo esta iteración, se intentará reestructurar la calendarización del proyecto de modo que se consiga finalizar en la fecha prevista o que la fecha final diste lo menos posible de ésta, anulando o disminuyendo en lo posible los efectos en los recursos económicos estimados.
Requerimientos	Ésta es una situación que debemos asumir y ante la cual poco podemos hacer. Nuestra única arma para disminuir en lo posible los efectos que pueda causar un rediseño de la base de datos es diseñar un esquema relacional factorizado al máximo, escalable y extendible fácilmente, de modo que los posibles cambios futuros afecten al menor número de tablas y por tanto tengan un impacto mínimo en el sistema.

Planes de contingencia

3. Análisis y diseño

3.1. Estudio del dominio de información ('Supuesto')

A continuación se describe el modelo de datos al que el sistema deberá dar cabida:

La **sede** consta de zonas.

El edificio puede constar de una o varias zonas, de las cuales bastará con almacenar su descripción y el nivel mínimo de acceso necesario para poder acceder a ella.



Sede de Informática Navarra S.A.

Varias zonas forman un nivel que es accesible única y exclusivamente si se dispone de una tarjeta que lo permita.



Plantas de los edificios

Los niveles requieren una numeración y una descripción que ayude a su comprensión.

El conjunto de **tarjetas** (numeración y color) del que se dispone está coloreado, correspondiéndose cada color a un único nivel.



Tarjetas coloreadas

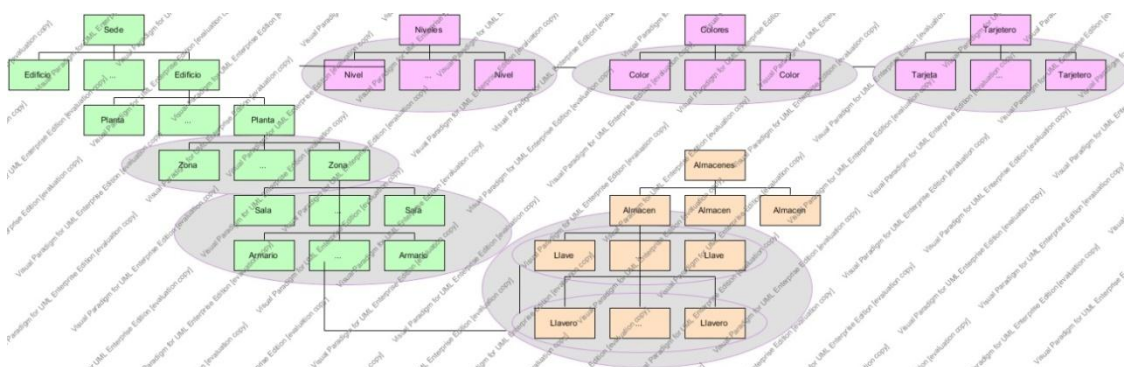
Una tarjeta sólo puede estar coloreada con un color y cada color sólo puede corresponder a un nivel de acceso. No obstante, pueden existir tarjetas coloreadas con distintos colores que proporcionen el mismo nivel. Cada tarjeta puede ser prestada a una persona, la cual puede acumular un número indefinido de préstamos.

También se dispone de un conjunto de **llaves** y **llaveros** disponibles en varios **almacenes**, que permitirán abrir puertas y armarios de las salas que componen cada zona.

Cada llave puede tener una o varias copias repartidas en uno o varios llaveros, además de poder tener existencia propia de manera individual (no asociada a ningún llavero). Todas las llaves y llaveros pueden ser almacenados de manera indistinta en uno o varios almacenes, de los cuales sólo es necesario almacenar una breve descripción. A partir de cada llave deberá poder identificarse a qué zona pertenece, qué abre, en qué lugares se encuentra almacenada y cuántas copias existen de la misma. Para los llaveros sólo será necesario saber qué llaves los componen, dónde están almacenados y de cuántas copias se disponen.

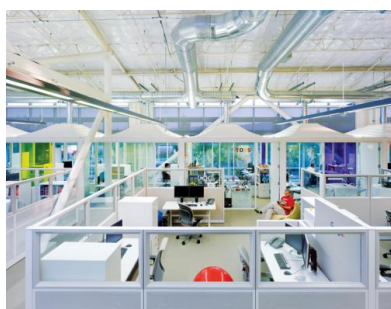


Armario de llaves



Sede, niveles, colores, tarjetas, llaves y llaveros

En cuanto a la vida diaria del centro, éste acoge el día a día de Informática Navarra S.A. y de varias **empresas**. Éstas se dividen en **departamentos**, a los cuales se les asigna un **tipo de acceso** en función de la actividad que realicen y el cual les permitirá acceder al centro bajo unas condiciones concretas.



Una empresa (CIF, nombre y dirección de sus oficinas centrales) consta de uno o varios departamentos, de los que se exige guardar la siguiente información:

- Nombre.
- Tipo de acceso que posee.
- Periodicidad con la que debe reafirmar la lista de personal que va a acceder al centro.
- Fecha en la que se realizó la última actualización de personal.
- DNI del trabajador (autorizador) de Informática Navarra S.A. que se hace responsable de la misma.
- Llaves que les está permitido tomar en préstamo a su personal a la hora de acceder al centro.



Los tipos de acceso son un combinado entre días, horas y nivel de acceso, así como de si es necesaria una autorización telefónica para acceder al centro (dependiendo del tipo de actividad que se realice) y qué tipo de actividad está permitido desarrollar. Se le asignará un tipo de acceso a cada departamento en función de la actividad que vaya a realizar, pudiendo ser asignado un mismo tipo a varios de ellos.

El **personal** que accede al centro se divide en trabajadores de Informática Navarra S.A, personal perteneciente a empresas (departamentos, en concreto) y personal que acude al centro de manera puntual (ocasional).

- Dentro de los primeros cabe distinguir a aquellos que se responsabilizarán de parte de los departamentos que trabajan en el centro (autorizadores) y a aquellos que ejercerán de administradores del sistema.
- En los segundos se encuentran encuadrados los vigilantes de seguridad, que realizarán las labores de vigilancia del centro y redactarán los **partes de seguridad** pertinentes (fecha, DNI del redactor e incidencias acontecidas en el servicio), pudiendo pertenecer en cada momento a una sola empresa-departamento de vigilancia contratada por el centro. Lo mismo ocurre con los trabajadores que acuden periódicamente al centro, pero no con los contactos con los que Informática Navarra S.A mantiene correspondencia, que podrán desempeñar uno o varios cargos dentro de cada empresa-departamento para la que trabajen. Entre estos últimos tendrán una gran importancia los responsables de personal, que serán los encargados de mantener actualizado el personal de las empresas-departamento en los que desempeñan tal cargo.
- El personal ocasional es aquel que acude al centro por motivos puntuales y también estará capacitado para obtener llaves y llaveros en préstamo.



Datos comunes a todo el personal son el DNI, apellidos, nombre y DNI del usuario que lo da de alta en el sistema. Como datos opcionales se tienen los números de teléfono o direcciones de correo electrónico que suministre cada persona para poder recibir comunicaciones o ser contactada en un momento determinado.

De manera especializada tendríamos los siguientes atributos:

- *Personal residente:* El nivel de acceso que posee, por si algún día olvida traer su tarjeta y es necesario prestarle una.
- *Personal periódico:* Empresa y departamento para el que trabaja.
- *Vigilantes:* Código de vigilante.
- *Contactos:* Empresas y departamentos a los que pertenecen junto con el cargo que desempeñan en cada caso. De los tipos de cargos sólo almacenaremos una breve descripción.
- *Personal ocasional:* Tipo de acceso otorgado, fecha de caducidad del permiso, notas a tener en cuenta y empresa a la que pertenece.



Todo el personal descrito con anterioridad (salvo los contactos) accede al centro para realizar sus labores. Dichos **accesos** deben quedar registrados según la política establecida en el documento de *Estudio detallado del problema software*.

Los accesos vienen determinados por la persona que accede al centro, la fecha y horas de entrada y salida, el motivo del acceso, la tarjeta que utiliza para moverse por el centro, otros datos de interés y la persona que ha permitido y registrado el acceso (normalmente será el vigilante).

No existe ningún tipo de restricción sobre el número de accesos y salidas del centro que un trabajador puede realizar con una tarjeta, pero si del tipo de acceso, que quedará determinado por el asignado al departamento al que pertenezca el trabajador periódico o el objeto de la visita realizada por la persona ocasional, o por el nivel de acceso en el caso del personal residente.

Al acceder al centro el trabajador puede requerir que le sea prestada alguna llave o llavero. En caso de existir copias y que el préstamo sea lícito, éste deberá ser registrado en el sistema (basta con suministrar al sistema la relación entre acceso y ente prestado).



Para que el sistema sea viable será necesario un mantenimiento y actualización constante por parte de los **usuarios** del mismo. Éstos, que se dividen en administradores, autorizadores, vigilantes y responsables de personal, dispondrán cada uno de una cuenta con la cual poder acceder al sistema y realizar las funciones que se les ha asignado. Será obligatorio llevar un registro de todos los accesos al sistema.

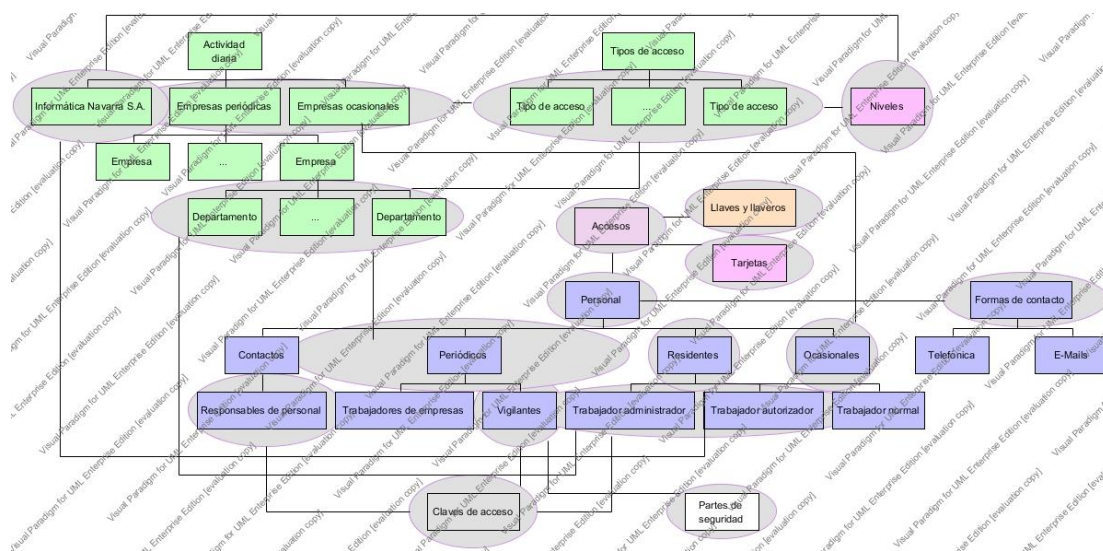


Cada cuenta estará determinada por un nombre de usuario y una contraseña, que permitirán loguear en el sistema, y un tipo de usuario, del que se desea almacenar una breve descripción y que marcará a qué funcionalidades puede acceder éste. Una cuenta sólo puede ofrecer las funcionalidades establecidas para un rol determinado.

La aplicación permitirá la comunicación inter-usuario a través de un sistema de **notas**. Todo usuario de ésta (que no del sistema) podrá redactar notas y dirigirlas al conjunto de usuarios que estime oportuno.



Una nota está compuesta por la fecha de redacción, el DNI del autor, el tema y el texto que la compone. Además deberá diseñarse una estructura que permita reflejar la lectura y escritura de notas, ofreciendo las funcionalidades descritas en el modelo de casos de uso. Un usuario puede redactar o no notas destinadas a uno o varios destinatarios.

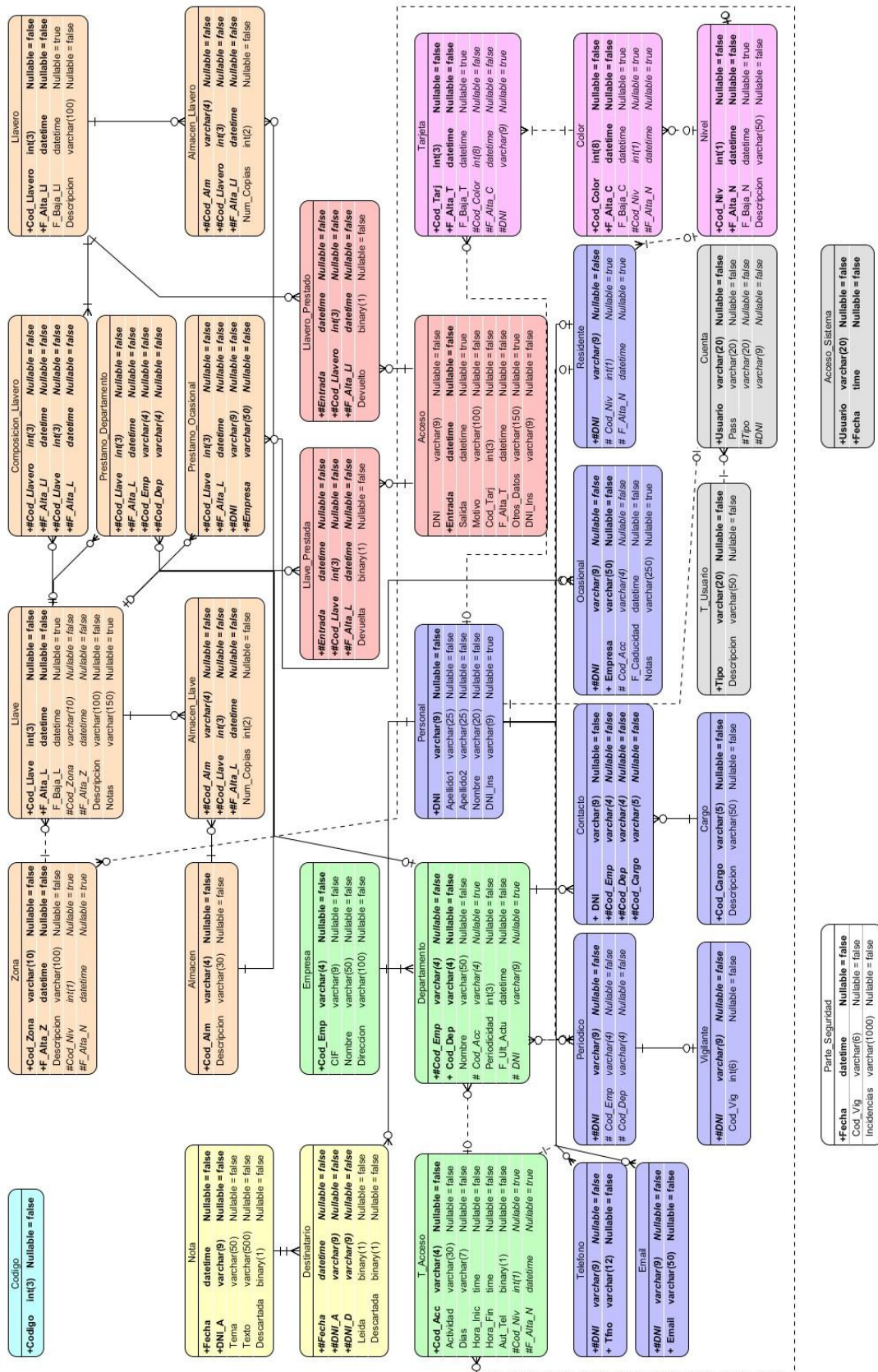


Empresas, departamentos, tipos de acceso, niveles, personal, accesos, usuarios y notas

3.2. Diseño de la base de datos

Tras conocer y comprender todo el dominio de información que debe manejarse con el futuro sistema, procede realizar un conciso análisis, diseño e implementación del modelo de datos que lo refleje, utilizando para ello el modelo relacional presentado por Edgar Frank Codd.

Hemos considerado que el esquema mostrado en la siguiente página nos daría el soporte necesario para desarrollar el sistema.



Esquema relacional de la BD



A continuación realizamos un breve estudio de cada una de las tablas que componen el esquema relacional, describiendo su funcionalidad y estructura. Además, realizamos un breve estudio de la interdependencia entre ellas, haciendo hincapié en los efectos que tendrán las actualizaciones y borrados de tuplas en ellas.

TARJETAS, COLORES Y NIVELES DE ACCESO		
TABLA	DESCRIPCIÓN	
Todas	En ellas almacenaremos toda la información referente a los diversos niveles de acceso que existen en el centro, junto con las tarjetas coloreadas que lo proporcionan.	<pre> classDiagram class Tarjeta { +Cod_Tarj int(3) +F_Alta_T datetime F_Baja_T datetime #Cod_Color int(8) #F_Alta_C datetime #DNI varchar(9) } class Color { +Cod_Color int(8) +F_Alta_C datetime F_Baja_C datetime #Cod_Niv int(1) #F_Alta_N datetime } class Nivel { +Cod_Niv int(1) +F_Alta_N datetime F_Baja_N datetime Descripción varchar(50) } Tarjeta "1" -- "1" Color Color "1" -- "1" Nivel </pre>
Tarjeta	Función	Tabla que almacena las diversas tarjetas registradas junto con su color.
	Cod_Tarj	Número de tarjeta.
	F_Alta_T	Fecha y hora de inserción de la tarjeta.
	F_Baja_T	Fecha y hora de baja de la tarjeta. Valor pred.: 'null'.
	Cod_Color	Código del color de la tarjeta.
	F_Alta_C	Fecha y hora de inserción del color de la tarjeta.
	DNI	DNI de la persona que la posee en préstamo en estos momentos.
	Actualizaciones y borrados	FK (Cod_Color,F_Alta_C) REFERENCIA Color: <ul style="list-style-type: none"> ON UPDATE CASCADE: Todo color actualizado en Color hará que se actualicen las respectivas tuplas en Tarjetas. ON DELETE CASCADE: Todo color borrado en Color hará que sea eliminada toda tupla existente en Tarjetas con Cod_Color = Color_borrado.

Tarjetas, colores y niveles de acceso



Color	Función	Tabla que almacena los diversos colores de tarjetas existentes junto con los niveles de acceso a los que están asociadas.
	Cod_Color	Código de color.
	F_Alta_C	Fecha y horade inserción del color.
	F_Baja_C	Fecha y hora de baja del color. Valor pred.: 'null'.
	Cod_Niv	Nivel de acceso que proporciona.
	F_Alta_N	Fecha y hora de inserción del nivel.
	Actualizaciones y borrados	FK (Cod_Niv,F_Alta_N) REFERENCIA Nivel: <ul style="list-style-type: none">• ON UPDATE CASCADE: Todo nivel actualizado en Nivel hará que se actualicen las respectivas tuplas en Color.• ON DELETE SET NULL: Todo nivel borrado en Nivel hará que se actualicen a null las respectivas tuplas en Color, dejando así los colores libres para poder ser asignados a otros niveles en un futuro.
Nivel	Función	Tabla que almacena los diversos niveles de acceso existentes en el centro además de su descripción.
	Cod_Niv	Nivel de acceso.
	F_Alta_N	Fecha y hora de inserción del nivel.
	F_Baja_N	Fecha y hora de baja del nivel. Valor pred.: 'null'.
	Descripcion	Descripción del nivel.
	Actualizaciones y borrados	Ninguna clave foránea existente.

Tarjetas, colores y niveles de acceso



ZONAS, LLAVES Y LLAVEROS																			
TABLA	DESCRIPCIÓN																		
Todas	<p>En ellas registramos las zonas que constituyen la sede, así como todas las llaves y llaveros que serán usados en ésta y cómo se encuentran almacenados.</p>																		
	<table border="1"> <tr> <td>Función</td> <td>Tabla que almacena las diversas zonas que conforman la sede de Informática Navarra S.A..</td> </tr> <tr> <td>Cod Zona</td> <td>Código de zona.</td> </tr> <tr> <td>F Alta Z</td> <td>Fecha y hora de inserción de la zona.</td> </tr> <tr> <td>Descripcion</td> <td>Descripción de la zona.</td> </tr> <tr> <td>Cod_Niv</td> <td>Nivel necesario para acceder a ella.</td> </tr> <tr> <td>F_Alta_N</td> <td>Fecha y hora de inserción del nivel.</td> </tr> <tr> <td>Actualizaciones y borrados</td> <td> FK (Cod_Niv,F_Alta_N) REFERENCIA Nivel: <ul style="list-style-type: none"> ON UPDATE CASCADE: La actualización de un nivel desencadenará la actualización en cascada de todas las tuplas de esta tabla que lo referencien. ON DELETE SET NULL: Si un nivel es borrado, todas las zonas que posean dicho nivel pasarán a tener un tipo de nivel 'null', a la espera de que les sea asignado uno nuevo. </td> </tr> </table>	Función	Tabla que almacena las diversas zonas que conforman la sede de Informática Navarra S.A..	Cod Zona	Código de zona.	F Alta Z	Fecha y hora de inserción de la zona.	Descripcion	Descripción de la zona.	Cod_Niv	Nivel necesario para acceder a ella.	F_Alta_N	Fecha y hora de inserción del nivel.	Actualizaciones y borrados	FK (Cod_Niv,F_Alta_N) REFERENCIA Nivel: <ul style="list-style-type: none"> ON UPDATE CASCADE: La actualización de un nivel desencadenará la actualización en cascada de todas las tuplas de esta tabla que lo referencien. ON DELETE SET NULL: Si un nivel es borrado, todas las zonas que posean dicho nivel pasarán a tener un tipo de nivel 'null', a la espera de que les sea asignado uno nuevo. 				
Función	Tabla que almacena las diversas zonas que conforman la sede de Informática Navarra S.A..																		
Cod Zona	Código de zona.																		
F Alta Z	Fecha y hora de inserción de la zona.																		
Descripcion	Descripción de la zona.																		
Cod_Niv	Nivel necesario para acceder a ella.																		
F_Alta_N	Fecha y hora de inserción del nivel.																		
Actualizaciones y borrados	FK (Cod_Niv,F_Alta_N) REFERENCIA Nivel: <ul style="list-style-type: none"> ON UPDATE CASCADE: La actualización de un nivel desencadenará la actualización en cascada de todas las tuplas de esta tabla que lo referencien. ON DELETE SET NULL: Si un nivel es borrado, todas las zonas que posean dicho nivel pasarán a tener un tipo de nivel 'null', a la espera de que les sea asignado uno nuevo. 																		
Llave	<table border="1"> <tr> <td>Función</td> <td>Tabla que almacena los datos relevantes de todas las llaves existentes.</td> </tr> <tr> <td>Cod Llave</td> <td>Número de llave.</td> </tr> <tr> <td>F Alta L</td> <td>Fecha y hora de inserción de la llave.</td> </tr> <tr> <td>F_Baja_L</td> <td>Fecha y hora de baja de la llave. Valor pred.: 'null'.</td> </tr> <tr> <td>Cod_Zona</td> <td>Zona a la que pertenece.</td> </tr> <tr> <td>F_Alta_Z</td> <td>Fecha y hora de inserción de la zona.</td> </tr> <tr> <td>Descripción</td> <td>Descripción de la cerradura que abre.</td> </tr> <tr> <td>Notas</td> <td>Otros datos a tener en cuenta. Valor pred.: 'null'.</td> </tr> <tr> <td>Actualizaciones y borrados</td> <td> FK (Cod_Zona,F_Alta_Z) REFERENCIA Zona: <ul style="list-style-type: none"> ON UPDATE CASCADE: Si una zona cambia de PK es lógico que todas las FKs que la referencien se actualicen para seguir manteniendo la integridad referencial. ON DELETE CASCADE: El borrado de una zona hará que también se eliminen todas las llaves ya que carece de sentido almacenar llaves de zonas en inexistentes. </td> </tr> </table>	Función	Tabla que almacena los datos relevantes de todas las llaves existentes.	Cod Llave	Número de llave.	F Alta L	Fecha y hora de inserción de la llave.	F_Baja_L	Fecha y hora de baja de la llave. Valor pred.: 'null'.	Cod_Zona	Zona a la que pertenece.	F_Alta_Z	Fecha y hora de inserción de la zona.	Descripción	Descripción de la cerradura que abre.	Notas	Otros datos a tener en cuenta. Valor pred.: 'null'.	Actualizaciones y borrados	FK (Cod_Zona,F_Alta_Z) REFERENCIA Zona: <ul style="list-style-type: none"> ON UPDATE CASCADE: Si una zona cambia de PK es lógico que todas las FKs que la referencien se actualicen para seguir manteniendo la integridad referencial. ON DELETE CASCADE: El borrado de una zona hará que también se eliminen todas las llaves ya que carece de sentido almacenar llaves de zonas en inexistentes.
	Función	Tabla que almacena los datos relevantes de todas las llaves existentes.																	
	Cod Llave	Número de llave.																	
	F Alta L	Fecha y hora de inserción de la llave.																	
	F_Baja_L	Fecha y hora de baja de la llave. Valor pred.: 'null'.																	
	Cod_Zona	Zona a la que pertenece.																	
	F_Alta_Z	Fecha y hora de inserción de la zona.																	
	Descripción	Descripción de la cerradura que abre.																	
Notas	Otros datos a tener en cuenta. Valor pred.: 'null'.																		
Actualizaciones y borrados	FK (Cod_Zona,F_Alta_Z) REFERENCIA Zona: <ul style="list-style-type: none"> ON UPDATE CASCADE: Si una zona cambia de PK es lógico que todas las FKs que la referencien se actualicen para seguir manteniendo la integridad referencial. ON DELETE CASCADE: El borrado de una zona hará que también se eliminen todas las llaves ya que carece de sentido almacenar llaves de zonas en inexistentes. 																		

Zonas, llaves y llaveros



Llavero	Función	Tabla que almacena todos los llaveros existentes, así como la descripción de las llaves que aglutina.
	<u>Cod Llavero</u>	Número de llavero.
	<u>F Alta LI</u>	Fecha y hora de inserción del llavero.
	<u>F Baja LI</u>	Fecha y hora de baja del llavero. Valor pred.: 'null'.
	Descripcion	Descripción del llavero.
	Actualizaciones y borrados	Ninguna clave foránea existente.
Composicion _Llavero	Función	Tabla que relaciona llaveros y llaves; qué llaves forman cada llavero.
	<u>Cod Llavero</u>	Número de llavero.
	<u>F Alta LI</u>	Fecha y hora de inserción del llavero.
	<u>Cod Llave</u>	Número de llave.
	<u>F Alta L</u>	Fecha y hora de inserción de la llave.
	Actualizaciones y borrados	<p>FK (Cod_Llavero,F_Alta_LI) REFERENCIA Llavero:</p> <ul style="list-style-type: none"> • ON UPDATE CASCADE: Si un llavero cambia de numeración, se producirá una actualización en cascada. • ON DELETE CASCADE: El borrado de un llavero provocará el borrado de todas las tuplas que detallen las llaves que posee. <p>FK (Cod_Llave,F_Alta_L) REFERENCIA Llave, y su política de integridad referencial es análoga a la de los llaveros.</p>
Almacen _Llave	Función	Tabla que almacena la relación de llaves “seltas” y los lugares en los que están almacenadas.
	<u>Cod Alm</u>	Lugar de almacenamiento.
	<u>Cod Llave</u>	Número de llave.
	<u>F Alta L</u>	Fecha y hora de inserción de la llave.
	Num_Copias	Número de copias existentes.
	Actualizaciones y borrados	<p>FK (Cod_Alm) REFERENCIA Almacen:</p> <ul style="list-style-type: none"> • ON UPDATE CASCADE: Si un almacén cambia de PK, lógicamente también lo harán las tuplas de aquellas llaves que estén almacenadas en éste. • ON DELETE CASCADE: Si un almacén deja de existir, no se mantendrá ningún registro de las llaves que albergaba. <p>FK (Cod_Llave,F_Alta_L) REFERENCIA Llave, y su política de integridad referencial es análoga a la explicada anteriormente: ON UPDATE CASCADE ON DELETE CASCADE.</p>

Zonas, llaves y llaveros



Almacen _Llavero	Función	Tabla que almacena la relación de llaveros y los lugares en los que éstos son almacenados.
	<u>Cod Alm</u>	Lugar de almacenamiento.
	<u>Cod Llavero</u>	Número de llavero.
	<u>F Alta LI</u>	Fecha y hora de inserción del llavero.
	Num_Copias	Número de copias existentes.
	Actualizaciones y borrados	Política de integridad referencial idéntica a la optada con las llaves.
Almacén	Función	Tabla que recoge los lugares en que pueden estar almacenadas las diversas llaves y llaveros.
	<u>Cod Alm</u>	Código de almacén.
	Descripción	Descripción del almacén.
	Actualizaciones y borrados	Ninguna clave foránea existente.
Prestamo _Departam.	Función	Tabla que recoge qué llaves pueden ser prestadas a qué departamentos.
	<u>Cod Llave</u>	Número de llave.
	<u>F Alta L</u>	Fecha y hora de inserción de la llave.
	<u>Cod Emp</u>	Código de empresa
	<u>Cod Dep</u>	Código de departamento.
	Actualizaciones y borrados	FK (Cod_Llave,F_Alta_L) REFERENCIA Llave: <ul style="list-style-type: none"> ON UPDATE CASCADE: Si una llave cambia de numeración, aquí también deberá llevarse a cabo dicho cambio. ON DELETE CASCADE: Si borramos una llave no tiene sentido almacenar a qué departamentos puede ser prestada. <p>Lo mismo ocurre con las empresas-departamentos.</p>
Prestamo _Ocasional	Función	Tabla que recoge qué llaves pueden ser prestadas a qué personas ocasionales.
	<u>Cod Llave</u>	Número de llave.
	<u>F Alta L</u>	Fecha y hora de inserción de la llave.
	<u>DNI</u>	DNI de la persona ocasional.
	<u>Empresa</u>	Empresa para la que trabaja la persona ocasional.
	Actualizaciones y borrados	FK (Cod_Llave,F_Alta_L) REFERENCIA Llave, encontrándonos en la misma situación que su tabla homóloga Prestamo_Departamento. FK (DNI,Empresa) REFERENCIA Ocasional, ON UPDATE CASCADE y ON DELETE CASCADE, al igual que como ocurre con los departamentos.

Zonas, llaves y llaveros



EMPRESAS, DEPARTAMENTOS Y TIPOS DE ACCESO		
TABLA	DESCRIPCIÓN	
Todas	<p>Tablas en las que almacenamos toda la información referente a las empresas que trabajan en el centro. Dicha información consta de las empresas en sí mismas, los departamentos que las forman y el tipo de acceso que poseen éstos.</p>	
Empresa	Función	Tabla que almacena las empresas que tienen departamentos trabajando en el centro.
	<u>Cod Emp</u>	Código de la empresa.
	CIF	Código de identificación fiscal de la empresa.
	Nombre	Nombre de la empresa.
	Dirección	Dirección de las oficinas de la empresa.
Actualizaciones y borrados	Ninguna clave foránea existente.	

Empresas, departamentos y tipos de acceso



Departamento	Función	Tabla que almacena todos los departamentos que constituyen las empresas que trabajan en el centro.
	<u>Cod_Emp</u>	Código de empresa.
	<u>Cod_Dep</u>	Código de departamento.
	Nombre	Nombre del departamento.
	<u>Cod_Acc</u>	Código de acceso que posee.
	Periodicidad	Periodicidad con la que debe renovar la lista de su personal que trabaja en el centro.
	F_Ult_Actu	Fecha y hora de la última actualización de personal realizada.
	<u>DNI</u>	DNI del autorizador responsable del departamento en el centro.
	Actualizaciones y borrados	<p>FK (Cod_Emp) REFERENCIA Empresa:</p> <ul style="list-style-type: none"> • ON UPDATE CASCADE: Cada actualización de la FK provocará una actualización en cascada para seguir manteniendo la integridad referencial en ambas tablas. • ON DELETE CASCADE: El borrado de una empresa también provocará un borrado en Departamento, ya que una vez eliminada una empresa no necesitamos continuar almacenando información de los departamentos que la componen. <p>FK (Cod_Acc) REFERENCIA T_Acceso:</p> <ul style="list-style-type: none"> • ON UPDATE CASCADE: Cada actualización en la tabla referenciada provocará una actualización en cascada. • ON DELETE SET NULL: El borrado de tuplas en T_Acceso provocará que los departamentos con dicho tipo de acceso pasen a tener un tipo de acceso 'null', estando a la espera de que se les vuelva a asignar uno para poder volver a acceder al centro. <p>FK (DNI) REFERENCIA Residente:</p> <ul style="list-style-type: none"> • ON UPDATE CASCADE: La actualización de un DNI en Residente causará la actualización en cascada en esta tabla. • ON DELETE SET NULL: Al ser borrado su responsable, dicho atributo pasará a ser 'null', estableciéndose lo antes posible un nuevo responsable para los departamentos "huérfanos".

Empresas, departamentos y tipos de acceso



T_Acceso	Función	Tabla que almacena todos los tipos de acceso que pueden ser asignados a los departamentos y personas ocasionales que acceden al centro.
	Cod_Acc	Código de acceso.
	Actividad	Actividad para la cual se concede el acceso.
	Dias	Días de la semana en lo que está permitido el acceso.
	Hora_Ini	Hora de inicio a partir de la cual está permitido el acceso.
	Hora_Fin	Hora hasta la que está permitido el acceso al centro.
	Aut_Tel	Denota si es necesario el aviso telefónico a la persona responsable del departamento cada vez que un empleado de ésta accede a la sede.
	Cod_Niv	Nivel de acceso al centro.
	F_Alta_N	Fecha y hora de inserción del nivel.
Actualizaciones y borrados	Ninguna clave foránea existente.	

Empresas, departamentos y tipos de acceso



PERSONAL															
TABLA	DESCRIPCIÓN														
Todas	<p>Tablas que albergan los datos de todo el personal que accede al centro, así como el de todas aquellas personas con las que Informática Navarra S.A. mantiene contacto</p>														
Personal	<table border="1"> <tr> <td>Función</td> <td>Tabla que almacena los datos comunes a todo el personal.</td> </tr> <tr> <td>DNI</td> <td>DNI.</td> </tr> <tr> <td>Apellido1</td> <td>Primer apellido.</td> </tr> <tr> <td>Apellido2</td> <td>Segundo apellido.</td> </tr> <tr> <td>Nombre</td> <td>Nombre.</td> </tr> <tr> <td>DNI_Ins</td> <td>DNI del usuario que ha realizado el alta de esta persona en la base de datos.</td> </tr> <tr> <td>Actualizaciones y borrados</td> <td>Ninguna clave foránea existente.</td> </tr> </table>	Función	Tabla que almacena los datos comunes a todo el personal.	DNI	DNI.	Apellido1	Primer apellido.	Apellido2	Segundo apellido.	Nombre	Nombre.	DNI_Ins	DNI del usuario que ha realizado el alta de esta persona en la base de datos.	Actualizaciones y borrados	Ninguna clave foránea existente.
	Función	Tabla que almacena los datos comunes a todo el personal.													
	DNI	DNI.													
	Apellido1	Primer apellido.													
	Apellido2	Segundo apellido.													
	Nombre	Nombre.													
DNI_Ins	DNI del usuario que ha realizado el alta de esta persona en la base de datos.														
Actualizaciones y borrados	Ninguna clave foránea existente.														
Residente	<table border="1"> <tr> <td>Función</td> <td>Especialización de la tabla Personal en la que se describen los atributos propios del personal residente del centro, que es el que trabaja en él a diario y forma parte de la plantilla de Informática Navarra S.A.</td> </tr> <tr> <td>DNI</td> <td>DNI.</td> </tr> <tr> <td>Cod_Niv</td> <td>Nivel de acceso al centro que posee.</td> </tr> <tr> <td>F_Alta_N</td> <td>Fecha y hora de inserción del nivel.</td> </tr> <tr> <td>Actualizaciones y borrados</td> <td> <p>FK (DNI) REFERENCIA Personal:</p> <ul style="list-style-type: none"> ON UPDATE CASCADE: La actualización de un DNI provocará la actualización en cascada en esta tabla. ON DELETE CASCADE: Trataremos el borrado de personal en cascada, ya que la eliminación de una persona implica el borrado de todos sus datos. <p>FK (Cod_Niv,F_Alta_N) REFERENCIA Nivel:</p> <ul style="list-style-type: none"> ON UPDATE CASCADE: La actualización de un nivel tendrá que verse reflejada en esta tabla. ON DELETE SET NULL: Si borramos un nivel, todos los niveles de los residentes pasarán a ser 'null' hasta que les sea asignado un nuevo nivel. </td> </tr> </table>	Función	Especialización de la tabla Personal en la que se describen los atributos propios del personal residente del centro, que es el que trabaja en él a diario y forma parte de la plantilla de Informática Navarra S.A.	DNI	DNI.	Cod_Niv	Nivel de acceso al centro que posee.	F_Alta_N	Fecha y hora de inserción del nivel.	Actualizaciones y borrados	<p>FK (DNI) REFERENCIA Personal:</p> <ul style="list-style-type: none"> ON UPDATE CASCADE: La actualización de un DNI provocará la actualización en cascada en esta tabla. ON DELETE CASCADE: Trataremos el borrado de personal en cascada, ya que la eliminación de una persona implica el borrado de todos sus datos. <p>FK (Cod_Niv,F_Alta_N) REFERENCIA Nivel:</p> <ul style="list-style-type: none"> ON UPDATE CASCADE: La actualización de un nivel tendrá que verse reflejada en esta tabla. ON DELETE SET NULL: Si borramos un nivel, todos los niveles de los residentes pasarán a ser 'null' hasta que les sea asignado un nuevo nivel. 				
	Función	Especialización de la tabla Personal en la que se describen los atributos propios del personal residente del centro, que es el que trabaja en él a diario y forma parte de la plantilla de Informática Navarra S.A.													
	DNI	DNI.													
	Cod_Niv	Nivel de acceso al centro que posee.													
	F_Alta_N	Fecha y hora de inserción del nivel.													
Actualizaciones y borrados	<p>FK (DNI) REFERENCIA Personal:</p> <ul style="list-style-type: none"> ON UPDATE CASCADE: La actualización de un DNI provocará la actualización en cascada en esta tabla. ON DELETE CASCADE: Trataremos el borrado de personal en cascada, ya que la eliminación de una persona implica el borrado de todos sus datos. <p>FK (Cod_Niv,F_Alta_N) REFERENCIA Nivel:</p> <ul style="list-style-type: none"> ON UPDATE CASCADE: La actualización de un nivel tendrá que verse reflejada en esta tabla. ON DELETE SET NULL: Si borramos un nivel, todos los niveles de los residentes pasarán a ser 'null' hasta que les sea asignado un nuevo nivel. 														

Personal



Periodico _Contacto	Función	Especialización de Personal que almacena los datos comunes a los trabajadores periódicos y contactos.
	<u>DNI</u>	DNI.
	<u>Cod_Emp</u>	Código de empresa para la que trabaja.
	<u>Cod_Dep</u>	Código de departamento para el que trabaja.
	Actualizaciones y borrados	<p>FK (DNI) REFERENCIA Personal: ON UPDATE CASCADE y ON DELETE CASCADE, por las mismas razones expuestas para la tabla Residente.</p> <p>FK (Cod_Emp,Cod_Dep) REFERENCIA Departamento:</p> <ul style="list-style-type: none"> ON UPDATE CASCADE: Al modificar la PK de un departamento debe realizarse una actualización en cascada en esta tabla para que puedan seguir relacionándose el departamento con todo su personal. ON DELETE CASCADE: Si prescindimos de un departamento también lo haremos de todo el personal que tenga que ver con él.
Vigilante	Función	Especialización de Periodico que almacena los datos específicos de los vigilantes de seguridad que realizan sus labores en el centro.
	<u>DNI</u>	DNI.
	<u>Cod_Vig</u>	Código de vigilante.
	Actualizaciones y borrados	FK (DNI) REFERENCIA Periodico, y como subtipo tiene la política de ON UPDATE CASCADE ON DELETE CASCADE.
Contacto	Función	Tabla que relaciona todos los contactos existentes con los cargos que desempeñan en sus respectivas empresas y departamentos.
	<u>DNI</u>	DNI.
	<u>Cod_Emp</u>	Código de empresa para la que trabaja.
	<u>Cod_Dep</u>	Código de departamento para el que trabaja.
	<u>Cod_Cargo</u>	Cargo desempeñado.
Actualizaciones y borrados	FK (DNI) REFERENCIA Personal, (Cod_Emp,Cod_Dep) REFERENCIA Departamento y (Cod_Cargo) REFERENCIA Cargo, llevando a cabo todas estas relaciones una actualización y borrado en cascada.	
Cargo	Función	Tabla que almacena los diversos cargos de los contactos con los que Informática Navarra S.A. mantiene relación, junto con sus correspondientes descripciones.
	<u>Cod_Cargo</u>	Código de cargo.
	Descripcion	Descripción del cargo.
	Actualizaciones y borrados	Ninguna clave foránea existente.

Personal



Ocasional	Función	Subtipo de Personal que almacena los datos propios del personal ocasional, que es aquel que accede al centro de forma puntual.
	<u>DNI</u>	DNI.
	<u>Empresa</u>	Empresa para la que trabaja.
	Cod_Acc	Código de acceso proporcionado.
	F_Caducidad	Fecha hasta la cual le está permitido el acceso.
	Notas	Otros datos.
	Actualizaciones y borrados	<p>FK (DNI) REFERENCIA Personal, y se maneja al igual que el resto de especializaciones del personal.</p> <p>FK (Cod_Acc) REFERENCIA T_Acceso, y mantenemos la siguiente política de actualización y borrado:</p> <ul style="list-style-type: none"> • ON UPDATE CASCADE: Si cambia la PK de un tipo de acceso, también cambiarán las FKs que referencien a la tupla actualizada. • ON DELETE SET NULL: En el caso de que se decida prescindir de un tipo de acceso, aquellas tuplas de permisos que tengan ese tipo de acceso actualizarán dicha FK a 'null', quedando a la espera de que les sea asignado uno nuevo.
Telefono	Función	Tabla que almacena los números de teléfono de todas aquellas personas que los proporcionen.
	<u>DNI</u>	DNI.
	Telefono	Número de teléfono de contacto.
	Actualizaciones y borrados	<p>FK (DNI) REFERENCES Personal:</p> <ul style="list-style-type: none"> • ON UPDATE CASCADE: Si una persona cambia de DNI también lo hará ese mismo DNI en esta tabla, para que así puedan seguir relacionándose personas y números de teléfono. • ON DELETE CASCADE: El borrado de una persona conlleva el borrado de todos sus datos personales.
Email	Función	Tabla que almacena las direcciones de correo electrónico de todas aquellas personas que las proporcionen.
	<u>DNI</u>	DNI.
	Email	Dirección de correo electrónico de contacto.
	Actualizaciones y borrados	Las mismas que las llevadas a cabo en la anterior tabla.

Personal



CUENTAS DE USUARIO Y ACCESOS AL SISTEMA		
TABLA	DESCRIPCIÓN	
Todas	<p>Tablas que almacenan todos los datos necesarios para una correcta gestión de los usuarios que utilizan el sistema (cuentas) y que llevan un registro de los accesos que realizan a éste.</p>	
	<pre> erDiagram T_Usuario --o{ Cuenta : "1" T_Usuario --o{ Acceso_Sistema : "1" Cuenta --o{ Acceso_Sistema : "1" </pre> <p>ER diagram showing relationships between T_Usuario, Cuenta, and Acceso_Sistema tables.</p>	
Cuenta	Función	Tabla que almacena todos los datos de las cuentas existentes en el sistema.
	Usuario	Nombre de usuario.
	Pass	Contraseña. Valor pred.: 'cambiar'.
	Tipo	Tipo de cuenta (Administrador, Autorizador,...).
	DNI	DNI de la persona a la que está asociada la cuenta.
	Actualizaciones y borrados	<p>FK (Tipo) REFERENCIA T_Usuario:</p> <ul style="list-style-type: none"> ON UPDATE CASCADE: Toda tupla actualizada en T_Usuario hará que se actualicen las tuplas que la referencien en Cuenta. ON DELETE CASCADE: Todo tipo de usuario borrado en el sistema hará que las cuentas pertenecientes a ese tipo también sean eliminadas. <p>FK (DNI) REFERENCIA Personal:</p> <ul style="list-style-type: none"> ON UPDATE CASCADE: Todo DNI actualizado en Personal provocará una actualización en cascada en esta tabla para que puedan seguir asociándose cuentas a personas. ON DELETE CASCADE: Si una persona deja de formar parte del centro no tiene sentido que siga poseyendo cuenta alguna.
T_Usuario	Función	Tabla que almacena los diversos roles que pueden adoptar los usuarios del sistema.
	Tipo	Tipo de usuario.
	Descripción	Descripción del tipo de usuario.
	Actualizaciones y borrados	Ninguna clave foránea existente.
Acceso_Sistema	Función	Tabla que contiene todos los accesos realizados por los diversos usuarios.
	Usuario	Nombre de usuario.
	Fecha	Fecha y hora en la que se realiza el acceso al sistema.
	Actualizaciones y borrados	FK (Usuario) REFERENCIA Cuenta: ON UPDATE CASCADE y ON DELETE CASCADE.

Cuentas de usuario



ACCESOS Y PRÉSTAMOS		
TABLA	DESCRIPCIÓN	
Todas	<p>Tablas de importancia primordial, ya que registran todos los accesos realizados en el centro junto con el préstamo de llaves y/o llaveros en caso de haberse realizado.</p>	
	<pre> erDiagram Llave_Prestada --o{ Acceso : "1" Llavero_Prestado --o{ Acceso : "1" Llave_Prestada { datetime Entrada PK int3 Cod_Llave PK datetime F_Alta_L PK binary1 Devuelta } Llavero_Prestado { datetime Entrada PK int3 Cod_Llavero PK datetime F_Alta_Ll PK binary1 Devuelto } Acceso { varchar9 DNI datetime Entrada PK datetime Salida varchar100 Motivo int3 Cod_Tarj int3 F_Alta_T varchar150 Otros_Datos varchar9 DNI_Ins } </pre>	
Acceso	Función	Tabla que registra todos los accesos llevados a cabo en el centro.
	DNI	DNI de la persona que accede.
	Entrada	Fecha y hora de entrada.
	Salida	Fecha y hora de salida. Valor pred.: 'null'.
	Motivo	Motivo de acceso.
	Cod_Tarj	Número de tarjeta empleada.
	F_Alta_T	Fecha y hora de inserción de la tarjeta.
	Otros_Datos	Otros datos de interés. Valor pred.: 'null'.
DNI_Ins	DNI del usuario que realiza la inserción del acceso.	
Actualizaciones y borrados	Ninguna clave foránea existente.	

Accesos y préstamos



Llave _Prestada	Función	Tabla que almacena los diversos préstamos de llaves realizados a las personas que acceden al centro.
	<u>Entrada</u>	Fecha y hora de acceso.
	<u>Cod Llave</u>	Número de llave prestada.
	<u>F Alta L</u>	Fecha y hora de inserción de la llave.
	Devuelta	Denota si la llave ha sido devuelta.
	Actualizaciones y borrados	<p>FK (Entrada) REFERENCIA Acceso:</p> <ul style="list-style-type: none"> • ON UPDATE CASCADE: Si se produce una actualización de alguna PK de Acceso, aquí también tendrá lugar ya que se debe mantener la integridad referencial de ambas tablas para conocer las llaves prestadas en cada acceso. • ON DELETE CASCADE: El borrado será en cascada, pues carece de utilidad alguna almacenar las llaves prestadas de un acceso desconocido. <p>FK (Cod_Llave,F_Alta_L) REFERENCIA Llave:</p> <ul style="list-style-type: none"> • ON UPDATE CASCADE: Si una llave cambia de numeración, sus tuplas en esta tabla también lo harán. • ON DELETE CASCADE: Si una llave es borrada, las tuplas que la referencien en esta tabla también serán borradas.
Llavero _Prestado	Función	Tabla que almacena los llaveros prestados en los diversos accesos realizados por el personal que trabaja en el centro.
	<u>Entrada</u>	Fecha y hora de acceso.
	<u>Cod Llavero</u>	Número de llavero prestado.
	<u>F Alta Ll</u>	Fecha y hora de inserción del llavero.
	Devuelto	Denota si el llavero ha sido devuelto.
	Actualizaciones y borrados	Sigue la misma lógica que el préstamo de llaves, solo que con los llaveros.

Accesos y préstamos



NOTAS	
TABLA	DESCRIPCIÓN
Todas	<p>En ellas almacenaremos todas las notas escritas por los diversos usuarios de la aplicación.</p> <pre> Nota +Fecha datetime Nullable = false +DNI_A varchar(9) Nullable = false Tema varchar(50) Nullable = false Texto varchar(500) Nullable = false Descartada binary(1) Nullable = false Destinatario +Fecha datetime Nullable = false +DNI_A varchar(9) Nullable = false +DNI_D varchar(9) Nullable = false Leida binary(1) Nullable = false Descartada binary(1) Nullable = false </pre>
Nota	<p>Función</p> <p>Tabla que almacena las notas escritas por los diversos usuarios de la aplicación.</p>
	<p>Fecha</p> <p>Fecha y hora de redacción.</p>
	<p>DNI A</p> <p>DNI del autor.</p>
	<p>Tema</p> <p>Tema/título de la nota.</p>
	<p>Texto</p> <p>Texto/contenido de la nota.</p>
	<p>Descartada</p> <p>Denota si la nota ha sido descartada por su autor.</p>
<p>Actualizaciones y borrados</p> <p>Ninguna clave foránea existente.</p>	
Destinatario	<p>Función</p> <p>Tabla que almacena los diversos destinatarios de cada nota y el estado en que se encuentra ésta con respecto a cada uno.</p>
	<p>Fecha</p> <p>Fecha y hora de redacción.</p>
	<p>DNI A</p> <p>DNI del autor.</p>
	<p>DNI D</p> <p>DNI del destinatario.</p>
	<p>Leida</p> <p>Denota si ha sido marcada como 'leída'. Valor pred.: null.</p>
	<p>Descartada</p> <p>Denota si ha sido marcada como 'descartada'. Valor pred.: null.</p>
<p>Actualizaciones y borrados</p> <p>FK (Fecha,DNI_A) REFERENCIA Nota:</p> <ul style="list-style-type: none"> ON UPDATE CASCADE: Cada nota que sea cambiada de fecha, hora y/o autor hará que todas las tuplas de sus destinatarios sufran el cambio en esta tabla. ON DELETE CASCADE: Si eliminamos una nota, también se eliminarán las tuplas que relativas a sus destinatarios. 	

Notas



PARTES DE SEGURIDAD																			
TABLA	DESCRIPCIÓN																		
Parte _Seguridad	Función	Tabla que almacena todos los partes de seguridad redactados por los vigilantes del centro.	<table border="1"><thead><tr><th colspan="4">Parte_Seguridad</th></tr></thead><tbody><tr><td>+Fecha</td><td>datetime</td><td>Nullable = false</td><td></td></tr><tr><td>Cod_Vig</td><td>varchar(6)</td><td>Nullable = false</td><td></td></tr><tr><td>Incidencias</td><td>varchar(1000)</td><td>Nullable = false</td><td></td></tr></tbody></table>	Parte_Seguridad				+Fecha	datetime	Nullable = false		Cod_Vig	varchar(6)	Nullable = false		Incidencias	varchar(1000)	Nullable = false	
	Parte_Seguridad																		
	+Fecha	datetime	Nullable = false																
	Cod_Vig	varchar(6)	Nullable = false																
	Incidencias	varchar(1000)	Nullable = false																
Fecha	Fecha y hora de comienzo del servicio.																		
DNI	Código de vigilante del vigilante redactor del parte.																		
Incidencias	Incidencias acontecidas durante el servicio.																		
Actualizaciones y borrados	Ninguna clave foránea existente.																		

Partes de seguridad

CÓDIGOS											
TABLA	DESCRIPCIÓN										
Codigo	Función	Tabla auxiliar que será utilizada en futuras consultas.	<table border="1"><thead><tr><th colspan="4">Codigo</th></tr></thead><tbody><tr><td>+Codigo</td><td>int(3)</td><td>Nullable = false</td><td></td></tr></tbody></table>	Codigo				+Codigo	int(3)	Nullable = false	
	Codigo										
	+Codigo	int(3)	Nullable = false								
Código	Código										
Actualizaciones y borrados	Ninguna clave foránea existente.										

Códigos

3.3. Diseño de los disparadores

Tras disponer de la estructura en la que almacenar los datos, dirigimos nuestros pasos a otorgar a la base de datos de la mayor funcionalidad requerida posible a través de disparadores. A lo largo de las siguientes páginas se documentan las necesidades y cómo son resueltas a través de los llamados “triggers”.



TARJETAS, COLORES Y NIVELES DE ACCESO				
TABLA	NOMBRE	MOMENTO	EVENTO	ACCIONES A REALIZAR
Nivel	ins_nivel	Antes	Inserción	Previamente a la inserción tenemos que dar de baja todos aquellos niveles (cero o uno) que dispongan del mismo código de nivel que el que va a ser introducido.
	actu_nivel	Después	Actualiz.	Si la actualización de un nivel implica la baja de éste, haremos que dicha baja se propague a través de los colores y tarjetas que los representan.
<ul style="list-style-type: none"> • Actualizar Nivel asignando al nivel cuyo código sea el que nos disponemos a introducir su fecha de baja a la actual. • Actualizar Colores asignando a los colores y tarjetas que estén relacionados con el nivel dado de baja su fecha de baja a la actual. 				
Color	ins_col	Antes	Inserción	Antes de insertar un color debemos dar de baja a aquellos colores que existan en el sistema con el mismo código que el nuevo (cero o uno), así como las tarjetas con las que éstos estén relacionados.
	actu_col	Después	Actualiz.	Después de actualizar o eliminar un color tendremos que comprobar que todos los niveles siguen disponiendo de algún color. En caso de no ser así, los eliminaremos.
	borr_col	Después	Borrado	
<ul style="list-style-type: none"> • Actualizar Color asignando al color cuyo código sea el que nos disponemos a introducir su fecha de baja a la actual. • Actualizar Tarjeta asignando a las tarjetas coloreadas con el color dado de baja la misma fecha de baja que a éste. • Eliminar de Nivel todos los niveles que no dispongan de colores. 				
Tarjeta	ins_tarj	Antes	Inserción	Antes de insertar cualquier tarjeta debemos asegurar que no existirán dos con el mismo código. En este caso daremos de baja la existente antes de añadir la nueva.
	actu_tarj	Después	Actualiz.	Tras actualizar o eliminar una tarjeta debemos asegurar que todo color y todo nivel disponen de alguna tarjeta y algún color respectivamente.
	borr_tarj	Después	Borrado	
<ul style="list-style-type: none"> • Actualizar Tarjeta asignando a la tarjeta cuyo código sea el que nos disponemos a introducir su fecha de baja a la actual. • Eliminar de Color todos los colores que no dispongan de tarjetas. • Eliminar de Nivel todos los niveles que no dispongan de colores. 				

Disparadores relativos a las tablas de tarjetas, colores y niveles de acceso
accessManager · Iteración C1: La base de datos · Página 30 de 55



ZONAS, LLAVES Y LLAVEROS				
TABLA	NOMBRE	MOMENTO	EVENTO	ACCIONES A REALIZAR
Zona	borr_zona	Después	Borrado	Si borramos una zona debemos eliminar del sistema todos aquellos llaveros que queden sin llaves y todos los almacenes que pudieran quedar vacíos.
<ul style="list-style-type: none"> • <i>Eliminar de Llavero todos los llaveros cuyos códigos no aparezcan referenciados en Composicion_Llavero.</i> • <i>Eliminar de Almacen todos los almacenes cuyos códigos no aparezcan referenciados en Almacen_Llave o Almacen_Llavero.</i> 				
Llave	ins_llave	Antes	Inserción	Al igual que nos sucede con las tarjetas, colores y niveles, antes de insertar cualquier llave debemos dar de baja (en caso de que exista) aquella que pueble la tabla Llaves y tenga el mismo código que la que nos disponemos a insertar.
<ul style="list-style-type: none"> • <i>Actualizar Llave asignando a la llave cuyo código sea el que nos disponemos a introducir su fecha de baja a la actual.</i> 				
Llave	borr_l	Después	Borrado	Se deberán realizar las mismas comprobaciones que las llevadas a cabo con el disparador <i>borr_zona</i> , ya que el borrado de una llave podría hacer que un llavero se quedase sin llaves o un almacén sin elementos que almacenar.
<ul style="list-style-type: none"> • <i>Eliminar de Llavero todos los llaveros cuyos códigos no aparezcan referenciados en Composicion_Llavero.</i> • <i>Eliminar de Almacen todos los almacenes cuyos códigos no aparezcan referenciados en Almacen_Llave o Almacen_Llavero.</i> 				
Almacen_Llave	borr_al	Después	Borrado	Si eliminamos una relación almacén-llave no estando la llave contenida en ningún almacén o llavero más, debemos eliminarla. Si además dejamos a un almacén vacío, procederá eliminarlo.
<ul style="list-style-type: none"> • <i>Eliminar de Llave todas las llaves cuyos códigos no aparezcan referenciados en Composicion_Llavero o Almacen_Llave.</i> • <i>Eliminar de Almacen todos los almacenes cuyos códigos no aparezcan referenciados en Almacen_Llave o Almacen_Llavero.</i> 				

Disparadores relativos a las tablas de zonas, llaves y llaveros



Llavero	ins_llavero	Antes	Inserción	Tal y como ocurría con las llaves, si nos disponemos a insertar un llavero cuyo código ya exista en la base de datos, deberemos dar de baja el llavero que lo posea para después pasar a realizar la inserción del nuevo.
<ul style="list-style-type: none"> • <i>Actualizar Llavero asignando al llavero cuyo código sea el que nos disponemos a introducir su fecha de baja a la actual.</i> 				
Llavero	borr_ll	Después	Borrado	Toda aquella llave que tras un borrado de llavero no corresponda a ningún otro llavero o tenga existencia propia deberá ser eliminada. Además realizaremos el mantenimiento de la tabla Almacen, borrando todos aquellos que no dispongan de elementos que albergar.
<ul style="list-style-type: none"> • <i>Eliminar de Llave todas las llaves cuyos códigos no aparezcan referenciados en Composicion_Llavero o Almacen_Llave.</i> • <i>Eliminar de Almacen todos los almacenen cuyos códigos no aparezcan referenciados en Almacen_Llave o Almacen_Llavero.</i> 				
Almacen_Llavero	borr_all	Después	Borrado	Si eliminamos una relación almacén-llavero no estando el llavero contenido en ningún almacén más, entonces debemos eliminarlo, así como las llaves no dispongan de existencia propia o conjunta a un llavero y aquellos almacenen que queden sin elementos que almacenar.
<ul style="list-style-type: none"> • <i>Eliminar de Llavero todos los llaveros cuyos códigos no aparezcan referenciados en Almacen_Llavero.</i> • <i>Eliminar de Llave todas las llaves cuyos códigos no aparezcan referenciados en Composicion_Llavero o Almacen_Llave.</i> • <i>Eliminar de Almacen todos los almacenen cuyos códigos no aparezcan referenciados en Almacen_Llave o Almacen_Llavero.</i> 				

Disparadores relativos a las tablas de zonas, llaves y llaveros



EMPRESAS, DEPARTAMENTOS Y TIPOS DE ACCESOS				
TABLA	NOMBRE	MOMENTO	EVENTO	ACCIONES A REALIZAR
Departamento	borr_dep	Después	Borrado	<p>Eliminar todos trabajadores periódicos y contactos que hayan quedado “huérfanos” de departamentos en los que trabajar.</p> <p>Si el departamento a borrar es el único que queda en la empresa, la eliminamos.</p> <p>Si existen cargos sin desempeñar, los borramos.</p>
<ul style="list-style-type: none"> • <i>Eliminar de Personal aquellas personas que no aparezcan referenciadas en ninguna de sus tablas especializadas.</i> • <i>Eliminar de Empresa todas las empresas cuyos códigos no aparezcan referenciados en Departamento.</i> • <i>Eliminar de Cargo aquellos cargos que no aparezcan referenciados en Contacto.</i> 				

Disparadores relativos a las tablas de empresas, departamentos y tipos de acceso

PERSONAL				
TABLA	NOMBRE	MOMENTO	EVENTO	ACCIONES A REALIZAR
Contacto	borr_contacto	Después	Borrado	<p>Borrar todos los contactos que no desempeñen su labora para ninguna empresa-departamento y todos los cargos que no estén relacionados con alguna persona.</p>
<ul style="list-style-type: none"> • <i>Eliminar de Personal la persona recientemente eliminada si no consta como contacto en otra empresa-departamento.</i> • <i>Eliminar de Cargo aquellos cargos que no aparezcan referenciados en Contacto.</i> 				

Disparadores relativos a las tablas de personal

CUENTAS DE USUARIO Y ACCESOS AL SISTEMA				
TABLA	NOMBRE	MOMENTO	EVENTO	ACCIONES A REALIZAR
Cuenta	borr_cuenta	Después	Borrado	<p>Borrar todos los tipos de usuarios que no dispongan de ninguna cuenta que los representen.</p>
<ul style="list-style-type: none"> • <i>Eliminar de T_Usuario aquellos tipos de usuario que no aparezcan referenciados en Cuenta.</i> 				

Disparadores relativos a las tablas de cuentas de usuario y accesos al sistema



ACCESOS Y PRÉSTAMOS				
TABLA	NOMBRE	MOMENTO	EVENTO	ACCIONES A REALIZAR
Acceso	ins_acc	Después	Inserción	Registramos que la tarjeta ofrecida a la persona entrante al centro la posee en préstamo.
<ul style="list-style-type: none"> • <i>Actualizar Tarjeta, asignando a la tarjeta prestada el DNI de su prestatario, es decir, que Tarjeta.DNI = NEW.DNI donde Tarjeta.Cod_Tarj = NEW.Cod_Tarj.</i> 				

Disparadores relativos a las tablas de accesos y préstamos

NOTAS				
TABLA	NOMBRE	MOMENTO	EVENTO	ACCIONES A REALIZAR
Nota	actu_notas	Después	Actualiz.	Si el redactor de una nota la descarta habiendo sido ésta descartada anteriormente por todos sus destinatarios, la borraremos ya que carece de sentido seguir almacenándola si nadie puede acceder a ella.
<ul style="list-style-type: none"> • <i>Eliminar de Nota aquellas notas que hayan sido descartadas por su autor y todos sus receptores.</i> 				
Destinatario	actu_dest	Después	Actualiz.	Debemos eliminar todas aquellas notas que hayan sido descartadas por su autor y todos sus destinatarios. Si en el caso anterior tratábamos que el autor era el último en descartarla, en éste manejaremos el caso en que el destinatario sea el último en descartarla.
<ul style="list-style-type: none"> • <i>Eliminar de Nota aquellas notas que hayan sido descartadas por su autor y todos sus receptores.</i> 				

Disparadores relativos a las tablas de notas

4. Implementación

4.1. Codificación e integración

La codificación de las tablas y disparadores resulta automática a partir del esquema relacional y las tablas de disparadores diseñadas. Toda ella ha sido realizada en el fichero *BD.ddl*, a excepción de los disparadores multi-acción, que han sido codificados en *Disparadores.sql*. Se crea una población de tamaño medio también en el primer fichero, para así poder simular la base de datos y



realizar todas las pruebas que sean necesarias a lo largo del desarrollo software.

En cuanto a la integración no cabe mencionar nada, ya que nos encontramos en el inicio del camino, en la primera iteración de la fase de Construcción, con que no existe ningún componente donde debamos integrar la base de datos.

4.2. Pruebas

La codificación de la base de datos resulta sencilla y carece de puntos críticos que debamos probar. Lo que sí que haremos es probar los disparadores, asegurando que realizan las funciones para las que los diseñamos y mantienen la coherencia de la base de datos que planificamos.

A continuación recopilamos todas las pruebas realizadas, llevando el resultado de las mismas también al *Plan de pruebas*.

Caso de prueba	1.01 - Disparador <i>ins_nivel</i>																																
Objetivo	Probar que la inserción de un nivel cuyo código existe provoca la baja del nivel que lo representa actualmente en la base de datos y acto seguido se realiza la inserción del nuevo nivel.																																
Actividades	Realizar la inserción de un nivel cuyo código ya exista en la base de datos y chequear que el que existía ha sido dado de baja y el nuevo insertado.																																
Técnicas y herramientas	Utilidad phpMyAdmin.																																
Datos de entrada	<ul style="list-style-type: none"> Estado inicial de la tabla: <table border="1"> <thead> <tr> <th>Cod_Niv</th> <th>Nivel de acceso</th> <th>F_Alta_N</th> <th>Fecha y hora de inserción del nivel</th> <th>F_Baja_N</th> <th>Fecha y hora de baja del nivel</th> <th>Descripcion</th> <th>Descripción del nivel de acceso</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Acceso restringido: Zonas comunes</td> <td></td> </tr> <tr> <td>1</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Acceso total: Zonas comunes + Host</td> <td></td> </tr> </tbody> </table> <ul style="list-style-type: none"> Disparador <i>ins_nivel</i>. Inserción de un nivel: <p>Ejecutar la(s) consulta(s) SQL en la base de datos bd: <input type="text"/></p> <pre>INSERT INTO Color VALUES (-1,CURDATE(),NULL,0,'2010-01-01 00:00:01');</pre>	Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y hora de inserción del nivel	F_Baja_N	Fecha y hora de baja del nivel	Descripcion	Descripción del nivel de acceso	0		2010-01-01 00:00:01		NULL		Acceso restringido: Zonas comunes		1		2010-01-01 00:00:01		NULL		Acceso total: Zonas comunes + Host									
Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y hora de inserción del nivel	F_Baja_N	Fecha y hora de baja del nivel	Descripcion	Descripción del nivel de acceso																										
0		2010-01-01 00:00:01		NULL		Acceso restringido: Zonas comunes																											
1		2010-01-01 00:00:01		NULL		Acceso total: Zonas comunes + Host																											
Salida esperada	La tabla deberá poseer dos niveles con código 0, solo que uno tendrá fecha de baja y el otro no.																																
Salida observada	La esperada:																																
	<table border="1"> <thead> <tr> <th>Cod_Niv</th> <th>Nivel de acceso</th> <th>F_Alta_N</th> <th>Fecha y hora de inserción del nivel</th> <th>F_Baja_N</th> <th>Fecha y hora de baja del nivel</th> <th>Descripcion</th> <th>Descripción del nivel de acceso</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>2010-06-17 00:00:01</td> <td></td> <td>Acceso restringido: Zonas comunes</td> <td></td> </tr> <tr> <td>0</td> <td></td> <td>2010-06-17 00:00:00</td> <td></td> <td>NULL</td> <td></td> <td>Nuevo nivel</td> <td></td> </tr> <tr> <td>1</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Acceso total: Zonas comunes + Host</td> <td></td> </tr> </tbody> </table>	Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y hora de inserción del nivel	F_Baja_N	Fecha y hora de baja del nivel	Descripcion	Descripción del nivel de acceso	0		2010-01-01 00:00:01		2010-06-17 00:00:01		Acceso restringido: Zonas comunes		0		2010-06-17 00:00:00		NULL		Nuevo nivel		1		2010-01-01 00:00:01		NULL		Acceso total: Zonas comunes + Host	
Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y hora de inserción del nivel	F_Baja_N	Fecha y hora de baja del nivel	Descripcion	Descripción del nivel de acceso																										
0		2010-01-01 00:00:01		2010-06-17 00:00:01		Acceso restringido: Zonas comunes																											
0		2010-06-17 00:00:00		NULL		Nuevo nivel																											
1		2010-01-01 00:00:01		NULL		Acceso total: Zonas comunes + Host																											
Evaluación/conclusión	El disparador <i>ins_nivel</i> funciona correctamente.																																

*Caso de prueba para el disparador *ins_nivel**



Caso de prueba	1.02 - Disparador <i>actu_nivel</i>																																																																																																																																																																																																																																																																																																																												
Objetivo	Comprobar que si un nivel es dado de baja también lo serán los colores y tarjetas que lo representan.																																																																																																																																																																																																																																																																																																																												
Actividades	Dar de baja un nivel y comprobar que los colores y tarjetas con los que está relacionado también son dados de baja.																																																																																																																																																																																																																																																																																																																												
Técnicas y herramientas	Utilidad phpMyAdmin.																																																																																																																																																																																																																																																																																																																												
Datos de entrada	<ul style="list-style-type: none"> Estado inicial de las tablas: <table border="1"> <thead> <tr> <th>Cod_Niv</th> <th>Nivel de acceso</th> <th>F_Alta_N</th> <th>Fecha y horade inserción del nivel</th> <th>F_Baja_N</th> <th>Fecha y hora de baja del nivel</th> <th>Descripcion</th> <th>Descripción del nivel de acceso</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Acceso restringido: Zonas comunes</td> <td></td> </tr> <tr> <td>1</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Acceso total: Zonas comunes + Host</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Cod_Color</th> <th>Color de la tarjeta</th> <th>F_Alta_C</th> <th>Fecha y horade inserción del color</th> <th>F_Baja_C</th> <th>Fecha y hora de baja del color</th> <th>Cod_Niv</th> <th>Nivel de acceso</th> <th>F_Alta_N</th> <th>Fecha y horade inserción del nivel</th> </tr> </thead> <tbody> <tr> <td>-16776961</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>1</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> </tr> <tr> <td>-256</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>0</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> </tr> <tr> <td>-1</td> <td></td> <td>2010-06-17 00:00:00</td> <td></td> <td>NULL</td> <td></td> <td>0</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Cod_Tarj</th> <th>Número de tarjeta</th> <th>F_Alta_T</th> <th>Fecha y hora de inserción de la tarjeta</th> <th>F_Baja_T</th> <th>Fecha y hora de baja de la tarjeta</th> <th>Cod_Color</th> <th>Color de la tarjeta</th> <th>F_Alta_C</th> <th>Fecha y hora de inserción del color</th> <th>DNI</th> <th>DNI de la persona que la tiene en su posesión</th> </tr> </thead> <tbody> <tr><td>49</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>-16776961</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td></tr> <tr><td>38</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>-16776961</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>00000097A</td><td></td></tr> <tr><td>37</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>-16776961</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td></tr> <tr><td>36</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>-16776961</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td></tr> <tr><td>35</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>-16776961</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>00000050A</td><td></td></tr> <tr><td>34</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>-16776961</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>00000160A</td><td></td></tr> <tr><td>33</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>-16776961</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td></tr> <tr><td>32</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>-16776961</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td></tr> <tr><td>31</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>-16776961</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td></tr> <tr><td>39</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>-16776961</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td></tr> <tr><td>40</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>-16776961</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td></tr> <tr><td>48</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>-16776961</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td></tr> <tr><td>47</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>-16776961</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td></tr> <tr><td>46</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>-16776961</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td></tr> <tr><td>45</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>-16776961</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td></tr> <tr><td>44</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>-16776961</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td></tr> <tr><td>43</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>-16776961</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td></tr> <tr><td>42</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>-16776961</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td></tr> <tr><td>41</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>-16776961</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td></tr> <tr><td>30</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>-16776961</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td></tr> </tbody> </table> <ul style="list-style-type: none"> Disparador <i>actu_nivel</i>. Baja de un nivel: <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: <input type="text"/> UPDATE Nivel SET F_Baja_N = CURDATE() WHERE Cod_Niv = 1 AND F_Alta_N = '2010-01-01 00:00:01';</pre>	Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y horade inserción del nivel	F_Baja_N	Fecha y hora de baja del nivel	Descripcion	Descripción del nivel de acceso	0		2010-01-01 00:00:01		NULL		Acceso restringido: Zonas comunes		1		2010-01-01 00:00:01		NULL		Acceso total: Zonas comunes + Host		Cod_Color	Color de la tarjeta	F_Alta_C	Fecha y horade inserción del color	F_Baja_C	Fecha y hora de baja del color	Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y horade inserción del nivel	-16776961		2010-01-01 00:00:01		NULL		1		2010-01-01 00:00:01		-256		2010-01-01 00:00:01		NULL		0		2010-01-01 00:00:01		-1		2010-06-17 00:00:00		NULL		0		2010-01-01 00:00:01		Cod_Tarj	Número de tarjeta	F_Alta_T	Fecha y hora de inserción de la tarjeta	F_Baja_T	Fecha y hora de baja de la tarjeta	Cod_Color	Color de la tarjeta	F_Alta_C	Fecha y hora de inserción del color	DNI	DNI de la persona que la tiene en su posesión	49		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL		38		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		00000097A		37		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL		36		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL		35		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		00000050A		34		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		00000160A		33		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL		32		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL		31		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL		39		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL		40		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL		48		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL		47		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL		46		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL		45		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL		44		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL		43		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL		42		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL		41		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL		30		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL	
Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y horade inserción del nivel	F_Baja_N	Fecha y hora de baja del nivel	Descripcion	Descripción del nivel de acceso																																																																																																																																																																																																																																																																																																																						
0		2010-01-01 00:00:01		NULL		Acceso restringido: Zonas comunes																																																																																																																																																																																																																																																																																																																							
1		2010-01-01 00:00:01		NULL		Acceso total: Zonas comunes + Host																																																																																																																																																																																																																																																																																																																							
Cod_Color	Color de la tarjeta	F_Alta_C	Fecha y horade inserción del color	F_Baja_C	Fecha y hora de baja del color	Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y horade inserción del nivel																																																																																																																																																																																																																																																																																																																				
-16776961		2010-01-01 00:00:01		NULL		1		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																																																					
-256		2010-01-01 00:00:01		NULL		0		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																																																					
-1		2010-06-17 00:00:00		NULL		0		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																																																					
Cod_Tarj	Número de tarjeta	F_Alta_T	Fecha y hora de inserción de la tarjeta	F_Baja_T	Fecha y hora de baja de la tarjeta	Cod_Color	Color de la tarjeta	F_Alta_C	Fecha y hora de inserción del color	DNI	DNI de la persona que la tiene en su posesión																																																																																																																																																																																																																																																																																																																		
49		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL																																																																																																																																																																																																																																																																																																																			
38		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		00000097A																																																																																																																																																																																																																																																																																																																			
37		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL																																																																																																																																																																																																																																																																																																																			
36		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL																																																																																																																																																																																																																																																																																																																			
35		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		00000050A																																																																																																																																																																																																																																																																																																																			
34		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		00000160A																																																																																																																																																																																																																																																																																																																			
33		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL																																																																																																																																																																																																																																																																																																																			
32		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL																																																																																																																																																																																																																																																																																																																			
31		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL																																																																																																																																																																																																																																																																																																																			
39		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL																																																																																																																																																																																																																																																																																																																			
40		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL																																																																																																																																																																																																																																																																																																																			
48		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL																																																																																																																																																																																																																																																																																																																			
47		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL																																																																																																																																																																																																																																																																																																																			
46		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL																																																																																																																																																																																																																																																																																																																			
45		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL																																																																																																																																																																																																																																																																																																																			
44		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL																																																																																																																																																																																																																																																																																																																			
43		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL																																																																																																																																																																																																																																																																																																																			
42		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL																																																																																																																																																																																																																																																																																																																			
41		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL																																																																																																																																																																																																																																																																																																																			
30		2010-01-01 00:00:01		NULL		-16776961		2010-01-01 00:00:01		NULL																																																																																																																																																																																																																																																																																																																			
Salida esperada	La tabla deberá poseer dos niveles con código 0, solo que uno tendrá fecha de baja y el otro no.																																																																																																																																																																																																																																																																																																																												
Salida observada	La esperada. Así quedan las tablas afectadas:																																																																																																																																																																																																																																																																																																																												
	<table border="1"> <thead> <tr> <th>Cod_Niv</th> <th>Nivel de acceso</th> <th>F_Alta_N</th> <th>Fecha y horade inserción del nivel</th> <th>F_Baja_N</th> <th>Fecha y hora de baja del nivel</th> <th>Descripcion</th> <th>Descripción del nivel de acceso</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Acceso restringido: Zonas comunes</td> <td></td> </tr> <tr> <td>1</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>2010-06-17 00:00:00</td> <td></td> <td>Acceso total: Zonas comunes + Host</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Cod_Color</th> <th>Color de la tarjeta</th> <th>F_Alta_C</th> <th>Fecha y horade inserción del color</th> <th>F_Baja_C</th> <th>Fecha y hora de baja del color</th> <th>Cod_Niv</th> <th>Nivel de acceso</th> <th>F_Alta_N</th> <th>Fecha y horade inserción del nivel</th> </tr> </thead> <tbody> <tr> <td>-16776961</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>2010-06-17 00:00:00</td> <td></td> <td>1</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> </tr> <tr> <td>-256</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>0</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> </tr> <tr> <td>-1</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>0</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> </tr> </tbody> </table>	Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y horade inserción del nivel	F_Baja_N	Fecha y hora de baja del nivel	Descripcion	Descripción del nivel de acceso	0		2010-01-01 00:00:01		NULL		Acceso restringido: Zonas comunes		1		2010-01-01 00:00:01		2010-06-17 00:00:00		Acceso total: Zonas comunes + Host		Cod_Color	Color de la tarjeta	F_Alta_C	Fecha y horade inserción del color	F_Baja_C	Fecha y hora de baja del color	Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y horade inserción del nivel	-16776961		2010-01-01 00:00:01		2010-06-17 00:00:00		1		2010-01-01 00:00:01		-256		2010-01-01 00:00:01		NULL		0		2010-01-01 00:00:01		-1		2010-01-01 00:00:01		NULL		0		2010-01-01 00:00:01																																																																																																																																																																																																																																																													
Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y horade inserción del nivel	F_Baja_N	Fecha y hora de baja del nivel	Descripcion	Descripción del nivel de acceso																																																																																																																																																																																																																																																																																																																						
0		2010-01-01 00:00:01		NULL		Acceso restringido: Zonas comunes																																																																																																																																																																																																																																																																																																																							
1		2010-01-01 00:00:01		2010-06-17 00:00:00		Acceso total: Zonas comunes + Host																																																																																																																																																																																																																																																																																																																							
Cod_Color	Color de la tarjeta	F_Alta_C	Fecha y horade inserción del color	F_Baja_C	Fecha y hora de baja del color	Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y horade inserción del nivel																																																																																																																																																																																																																																																																																																																				
-16776961		2010-01-01 00:00:01		2010-06-17 00:00:00		1		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																																																					
-256		2010-01-01 00:00:01		NULL		0		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																																																					
-1		2010-01-01 00:00:01		NULL		0		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																																																					



Cod_Tarj Número de tarjeta	F_Alta_T Fecha y hora de inserción de la tarjeta	F_Baja_T Fecha y hora de baja de la tarjeta	Cod_Color Color de la tarjeta	F_Alta_C Fecha y hora de inserción del color	DNI DNI de la persona que la tiene en su posesión
49	2010-01-01 00:00:01	2010-06-17 00:00:00	-16776961	2010-01-01 00:00:01	NULL
38	2010-01-01 00:00:01	2010-06-17 00:00:00	-16776961	2010-01-01 00:00:01	0000097A
37	2010-01-01 00:00:01	2010-06-17 00:00:00	-16776961	2010-01-01 00:00:01	NULL
36	2010-01-01 00:00:01	2010-06-17 00:00:00	-16776961	2010-01-01 00:00:01	NULL
35	2010-01-01 00:00:01	2010-06-17 00:00:00	-16776961	2010-01-01 00:00:01	0000050A
34	2010-01-01 00:00:01	2010-06-17 00:00:00	-16776961	2010-01-01 00:00:01	00000160A
33	2010-01-01 00:00:01	2010-06-17 00:00:00	-16776961	2010-01-01 00:00:01	NULL
32	2010-01-01 00:00:01	2010-06-17 00:00:00	-16776961	2010-01-01 00:00:01	NULL
31	2010-01-01 00:00:01	2010-06-17 00:00:00	-16776961	2010-01-01 00:00:01	NULL
39	2010-01-01 00:00:01	2010-06-17 00:00:00	-16776961	2010-01-01 00:00:01	NULL
40	2010-01-01 00:00:01	2010-06-17 00:00:00	-16776961	2010-01-01 00:00:01	NULL
48	2010-01-01 00:00:01	2010-06-17 00:00:00	-16776961	2010-01-01 00:00:01	NULL
47	2010-01-01 00:00:01	2010-06-17 00:00:00	-16776961	2010-01-01 00:00:01	NULL
46	2010-01-01 00:00:01	2010-06-17 00:00:00	-16776961	2010-01-01 00:00:01	NULL
45	2010-01-01 00:00:01	2010-06-17 00:00:00	-16776961	2010-01-01 00:00:01	NULL
44	2010-01-01 00:00:01	2010-06-17 00:00:00	-16776961	2010-01-01 00:00:01	NULL
43	2010-01-01 00:00:01	2010-06-17 00:00:00	-16776961	2010-01-01 00:00:01	NULL
42	2010-01-01 00:00:01	2010-06-17 00:00:00	-16776961	2010-01-01 00:00:01	NULL
41	2010-01-01 00:00:01	2010-06-17 00:00:00	-16776961	2010-01-01 00:00:01	NULL
30	2010-01-01 00:00:01	2010-06-17 00:00:00	-16776961	2010-01-01 00:00:01	NULL

Evaluación/
conclusión

El disparador *actu_nivel* funciona correctamente.

Caso de prueba para el disparador actu_nivel



Caso de prueba	1.03 - Disparador <i>ins_col</i>																																																		
Objetivo	Verificar que al insertar un color el sistema realiza la baja de todos los colores que pudieran existir con dicho código.																																																		
Actividades	Insertar un color cuyo código exista y comprobar que el que existía es dado de baja y el nuevo entra a formar parte del sistema.																																																		
Técnicas y herramientas	Utilidad phpMyAdmin.																																																		
Datos de entrada	<ul style="list-style-type: none">Estado inicial de la tabla: <table border="1"><thead><tr><th>Cod_Color</th><th>Color de la tarjeta</th><th>F_Alta_C</th><th>Fecha y horade inserción del color</th><th>F_Baja_C</th><th>Fecha y hora de baja del color</th><th>Cod_Niv</th><th>Nivel de acceso</th><th>F_Alta_N</th><th>Fecha y horade inserción del nivel</th></tr></thead><tbody><tr><td>-16776961</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>1</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr><tr><td>-256</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>0</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr><tr><td>-1</td><td></td><td>2010-06-17 00:00:00</td><td></td><td>NULL</td><td></td><td>0</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr></tbody></table>Disparador <i>ins_col</i>.Inserción de un col: Ejecutar la(s) consulta(s) SQL en la base de datos bd: @ <pre>INSERT INTO Nivel VALUES (0,CURDATE(),NULL,'Nuevo nivel')</pre>	Cod_Color	Color de la tarjeta	F_Alta_C	Fecha y horade inserción del color	F_Baja_C	Fecha y hora de baja del color	Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y horade inserción del nivel	-16776961		2010-01-01 00:00:01		NULL		1		2010-01-01 00:00:01		-256		2010-01-01 00:00:01		NULL		0		2010-01-01 00:00:01		-1		2010-06-17 00:00:00		NULL		0		2010-01-01 00:00:01											
Cod_Color	Color de la tarjeta	F_Alta_C	Fecha y horade inserción del color	F_Baja_C	Fecha y hora de baja del color	Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y horade inserción del nivel																																										
-16776961		2010-01-01 00:00:01		NULL		1		2010-01-01 00:00:01																																											
-256		2010-01-01 00:00:01		NULL		0		2010-01-01 00:00:01																																											
-1		2010-06-17 00:00:00		NULL		0		2010-01-01 00:00:01																																											
Salida esperada	Deberán existir dos colores con código -1, uno con fecha de baja y otro sin ella.																																																		
Salida observada	La esperada: <table border="1"><thead><tr><th>Cod_Color</th><th>Color de la tarjeta</th><th>F_Alta_C</th><th>Fecha y horade inserción del color</th><th>F_Baja_C</th><th>Fecha y hora de baja del color</th><th>Cod_Niv</th><th>Nivel de acceso</th><th>F_Alta_N</th><th>Fecha y horade inserción del nivel</th></tr></thead><tbody><tr><td>-16776961</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>1</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr><tr><td>-256</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>0</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr><tr><td>-1</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>2010-06-17 00:00:00</td><td></td><td>0</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr><tr><td>-1</td><td></td><td>2010-06-17 00:00:00</td><td></td><td>NULL</td><td></td><td>0</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr></tbody></table>	Cod_Color	Color de la tarjeta	F_Alta_C	Fecha y horade inserción del color	F_Baja_C	Fecha y hora de baja del color	Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y horade inserción del nivel	-16776961		2010-01-01 00:00:01		NULL		1		2010-01-01 00:00:01		-256		2010-01-01 00:00:01		NULL		0		2010-01-01 00:00:01		-1		2010-01-01 00:00:01		2010-06-17 00:00:00		0		2010-01-01 00:00:01		-1		2010-06-17 00:00:00		NULL		0		2010-01-01 00:00:01	
Cod_Color	Color de la tarjeta	F_Alta_C	Fecha y horade inserción del color	F_Baja_C	Fecha y hora de baja del color	Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y horade inserción del nivel																																										
-16776961		2010-01-01 00:00:01		NULL		1		2010-01-01 00:00:01																																											
-256		2010-01-01 00:00:01		NULL		0		2010-01-01 00:00:01																																											
-1		2010-01-01 00:00:01		2010-06-17 00:00:00		0		2010-01-01 00:00:01																																											
-1		2010-06-17 00:00:00		NULL		0		2010-01-01 00:00:01																																											
Evaluación/ conclusión	El disparador <i>ins_col</i> funciona correctamente.																																																		

Caso de prueba para el disparador ins_col



Caso de prueba	1.04 - Disparadores <i>actu_col</i> y <i>borr_col</i>																																																																
Objetivo	Comprobar que al actualizar o borrar un color sólo permanecen en el sistema aquellos niveles relacionados con algún color.																																																																
Actividades	Insertar dos registros color-nivel relacionados, actualizar o eliminar el color y ver que también desaparece el nivel.																																																																
Técnicas y herramientas	Utilidad phpMyAdmin.																																																																
Datos de entrada	<ul style="list-style-type: none"> Registros de prueba: <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: @</pre> <pre>INSERT INTO Nivel VALUES (9,'2010-01-01 00:00:01',NULL,'Descripcion de prueba'); INSERT INTO Color VALUES (-100,'2010-01-01 00:00:01',NULL,9,'2010-01-01 00:00:01');</pre> <ul style="list-style-type: none"> Disparadores <i>actu_col</i> y <i>borr_col</i>. Actualización o borrado del color insertado: <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: @</pre> <pre>UPDATE Color SET Cod_Niv = 0, F_Alta_N = '2010-01-01 00:00:01' WHERE Cod_color = -100 AND F_Alta_C = '2010-01-01 00:00:01'</pre> <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: @</pre> <pre>DELETE FROM Color WHERE Cod_Color = -100 AND F_Alta_C = '2010-01-01 00:00:01'</pre>																																																																
Salida esperada	En el caso de la actualización, el color deja de existir; en el otro caso, ninguna de la dos tuplas consta en la base de datos.																																																																
Salida observada	<p>La esperada:</p> <table border="1"> <thead> <tr> <th>Cod_Color</th> <th>Color de la tarjeta</th> <th>F_Alta_C</th> <th>Fecha y horade inserción del color</th> <th>F_Baja_C</th> <th>Fecha y hora de baja del color</th> <th>Cod_Niv</th> <th>Nivel de acceso</th> <th>F_Alta_N</th> <th>Fecha y horade inserción del nivel</th> </tr> </thead> <tbody> <tr> <td>-16776961</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>1</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> </tr> <tr> <td>-256</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>0</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> </tr> <tr> <td>-100</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>0</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Cod_Niv</th> <th>Nivel de acceso</th> <th>F_Alta_N</th> <th>Fecha y horade inserción del nivel</th> <th>F_Baja_N</th> <th>Fecha y hora de baja del nivel</th> <th>Descripcion</th> <th>Descripción del nivel de acceso</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Acceso restringido: Zonas comunes</td> <td></td> </tr> <tr> <td>1</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Acceso total: Zonas comunes + Host</td> <td></td> </tr> </tbody> </table>	Cod_Color	Color de la tarjeta	F_Alta_C	Fecha y horade inserción del color	F_Baja_C	Fecha y hora de baja del color	Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y horade inserción del nivel	-16776961		2010-01-01 00:00:01		NULL		1		2010-01-01 00:00:01		-256		2010-01-01 00:00:01		NULL		0		2010-01-01 00:00:01		-100		2010-01-01 00:00:01		NULL		0		2010-01-01 00:00:01		Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y horade inserción del nivel	F_Baja_N	Fecha y hora de baja del nivel	Descripcion	Descripción del nivel de acceso	0		2010-01-01 00:00:01		NULL		Acceso restringido: Zonas comunes		1		2010-01-01 00:00:01		NULL		Acceso total: Zonas comunes + Host	
Cod_Color	Color de la tarjeta	F_Alta_C	Fecha y horade inserción del color	F_Baja_C	Fecha y hora de baja del color	Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y horade inserción del nivel																																																								
-16776961		2010-01-01 00:00:01		NULL		1		2010-01-01 00:00:01																																																									
-256		2010-01-01 00:00:01		NULL		0		2010-01-01 00:00:01																																																									
-100		2010-01-01 00:00:01		NULL		0		2010-01-01 00:00:01																																																									
Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y horade inserción del nivel	F_Baja_N	Fecha y hora de baja del nivel	Descripcion	Descripción del nivel de acceso																																																										
0		2010-01-01 00:00:01		NULL		Acceso restringido: Zonas comunes																																																											
1		2010-01-01 00:00:01		NULL		Acceso total: Zonas comunes + Host																																																											
Evaluación/ conclusión	Los disparadores <i>actu_col</i> y <i>borr_tarj</i> funcionan correctamente.																																																																

*Casos de prueba para los disparadores *actu_col* y *borr_col**



Caso de prueba	1.05 - Disparador <i>ins_tarj</i>																																										
Objetivo	Chequear que la inserción de una tarjeta repetida provoca que el sistema dé de baja la existente antes de registrar la nueva.																																										
Actividades	Insertar una tarjeta con código existente y comprobar que la vieja ha sido dada de baja.																																										
Técnicas y herramientas	Utilidad phpMyAdmin.																																										
Datos de entrada	<ul style="list-style-type: none"> Estado inicial de la tabla: <table border="1"> <thead> <tr> <th>Cod_Tarj</th> <th>F_Alta_T</th> <th>F_Baja_T</th> <th>Cod_Color</th> <th>F_Alta_C</th> <th>DNI</th> </tr> <tr> <td>Numero de tarjeta</td> <td>Fecha y hora de inserción de la tarjeta</td> <td>Fecha y hora de baja de la tarjeta</td> <td>Color de la tarjeta</td> <td>Fecha y hora de inserción del color</td> <td>DNI de la persona que la tiene en su posesión</td> </tr> </thead> <tbody> <tr> <td>0</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>-1</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> </tr> <tr> <td>1</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>-1</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> </tr> <tr> <td>2</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>-1</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> </tr> <tr> <td>3</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>-1</td> <td>2010-01-01 00:00:01</td> <td>00000150A</td> </tr> </tbody> </table> <ul style="list-style-type: none"> Disparador <i>ins_tarj</i>. Inserción de una tarjeta: <p>Ejecutar la(s) consulta(s) SQL en la base de datos bd: <input type="text"/></p> <pre>INSERT INTO Tarjeta VALUES (0,CURDATE(),NULL,-1,'2010-01-01 00:00:01',NULL);</pre>	Cod_Tarj	F_Alta_T	F_Baja_T	Cod_Color	F_Alta_C	DNI	Numero de tarjeta	Fecha y hora de inserción de la tarjeta	Fecha y hora de baja de la tarjeta	Color de la tarjeta	Fecha y hora de inserción del color	DNI de la persona que la tiene en su posesión	0	2010-01-01 00:00:01	NULL	-1	2010-01-01 00:00:01	NULL	1	2010-01-01 00:00:01	NULL	-1	2010-01-01 00:00:01	NULL	2	2010-01-01 00:00:01	NULL	-1	2010-01-01 00:00:01	NULL	3	2010-01-01 00:00:01	NULL	-1	2010-01-01 00:00:01	00000150A						
Cod_Tarj	F_Alta_T	F_Baja_T	Cod_Color	F_Alta_C	DNI																																						
Numero de tarjeta	Fecha y hora de inserción de la tarjeta	Fecha y hora de baja de la tarjeta	Color de la tarjeta	Fecha y hora de inserción del color	DNI de la persona que la tiene en su posesión																																						
0	2010-01-01 00:00:01	NULL	-1	2010-01-01 00:00:01	NULL																																						
1	2010-01-01 00:00:01	NULL	-1	2010-01-01 00:00:01	NULL																																						
2	2010-01-01 00:00:01	NULL	-1	2010-01-01 00:00:01	NULL																																						
3	2010-01-01 00:00:01	NULL	-1	2010-01-01 00:00:01	00000150A																																						
Salida esperada	El primer registro de la tabla pasará a tener fecha de baja '2010-06-17 00:00:00' y el sistema realizará la inserción del nuevo.																																										
Salida observada	La esperada:																																										
	<table border="1"> <thead> <tr> <th>Cod_Tarj</th> <th>F_Alta_T</th> <th>F_Baja_T</th> <th>Cod_Color</th> <th>F_Alta_C</th> <th>DNI</th> </tr> <tr> <td>Numero de tarjeta</td> <td>Fecha y hora de inserción de la tarjeta</td> <td>Fecha y hora de baja de la tarjeta</td> <td>Color de la tarjeta</td> <td>Fecha y hora de inserción del color</td> <td>DNI de la persona que la tiene en su posesión</td> </tr> </thead> <tbody> <tr> <td>0</td> <td>2010-01-01 00:00:01</td> <td>2010-06-17 00:00:00</td> <td>-1</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> </tr> <tr> <td>0</td> <td>2010-06-17 00:00:00</td> <td>NULL</td> <td>-1</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> </tr> <tr> <td>1</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>-1</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> </tr> <tr> <td>2</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>-1</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> </tr> <tr> <td>3</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>-1</td> <td>2010-01-01 00:00:01</td> <td>00000150A</td> </tr> </tbody> </table>	Cod_Tarj	F_Alta_T	F_Baja_T	Cod_Color	F_Alta_C	DNI	Numero de tarjeta	Fecha y hora de inserción de la tarjeta	Fecha y hora de baja de la tarjeta	Color de la tarjeta	Fecha y hora de inserción del color	DNI de la persona que la tiene en su posesión	0	2010-01-01 00:00:01	2010-06-17 00:00:00	-1	2010-01-01 00:00:01	NULL	0	2010-06-17 00:00:00	NULL	-1	2010-01-01 00:00:01	NULL	1	2010-01-01 00:00:01	NULL	-1	2010-01-01 00:00:01	NULL	2	2010-01-01 00:00:01	NULL	-1	2010-01-01 00:00:01	NULL	3	2010-01-01 00:00:01	NULL	-1	2010-01-01 00:00:01	00000150A
Cod_Tarj	F_Alta_T	F_Baja_T	Cod_Color	F_Alta_C	DNI																																						
Numero de tarjeta	Fecha y hora de inserción de la tarjeta	Fecha y hora de baja de la tarjeta	Color de la tarjeta	Fecha y hora de inserción del color	DNI de la persona que la tiene en su posesión																																						
0	2010-01-01 00:00:01	2010-06-17 00:00:00	-1	2010-01-01 00:00:01	NULL																																						
0	2010-06-17 00:00:00	NULL	-1	2010-01-01 00:00:01	NULL																																						
1	2010-01-01 00:00:01	NULL	-1	2010-01-01 00:00:01	NULL																																						
2	2010-01-01 00:00:01	NULL	-1	2010-01-01 00:00:01	NULL																																						
3	2010-01-01 00:00:01	NULL	-1	2010-01-01 00:00:01	00000150A																																						
Evaluación/conclusión	El disparador <i>ins_tarj</i> funciona correctamente.																																										

Caso de prueba para el disparador ins_tarj



Caso de prueba	1.06 - Disparadores <i>actu_tarj</i> y <i>borr_tarj</i>																								
Objetivo	Comprobar que al actualizar o borrar una tarjeta se eliminan todos aquellos colores que no dispongan de tarjetas y todos aquellos niveles que no dispongan de colores.																								
Actividades	Insertar tres registros relacionados de modo que la eliminación de la tarjeta debiera eliminar los tres. Actualizamos o eliminamos la tarjeta y vemos si ocurre lo que esperamos.																								
Técnicas y herramientas	Utilidad phpMyAdmin.																								
Datos de entrada	<ul style="list-style-type: none">Registros de prueba: <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: [?] INSERT INTO Nivel VALUES (9,'2010-01-01 00:00:01',NULL,'Descripcion de prueba'); INSERT INTO Color VALUES (-100,'2010-01-01 00:00:01',NULL,9,'2010-01-01 00:00:01'); INSERT INTO Tarjeta VALUES (999,'2010-01-01 00:00:01',NULL,-100,'2010-01-01 00:00:01',NULL)</pre>Disparadores <i>actu_tarj</i> y <i>borr_tarj</i>.Actualización o borrado de la tarjeta insertada: <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: [?] UPDATE Tarjeta SET Cod_Color = '-1', F_Alta_C = '2010-01-01 00:00:01' WHERE Cod_Tarj = 999 AND F_Alta_T = '2010-01-01 00:00:01' Ejecutar la(s) consulta(s) SQL en la base de datos bd: [?] DELETE FROM Tarjeta WHERE Cod_Tarj = 999 AND F_Alta_T = '2010-01-01 00:00:01';</pre>																								
Salida esperada	Ninguna de la dos/tres tuplas introducidas como datos de entrada existe en la base de datos.																								
Salida observada	La esperada: <table border="1"><thead><tr><th>Cod_Niv</th><th>Nivel de acceso</th><th>F_Alta_N</th><th>Fecha y hora de inserción del nivel</th><th>F_Baja_N</th><th>Fecha y hora de baja del nivel</th><th>Descripcion</th><th>Descripción del nivel de acceso</th></tr></thead><tbody><tr><td>0</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>Acceso restringido: Zonas comunes</td><td></td></tr><tr><td>1</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>Acceso total: Zonas comunes + Host</td><td></td></tr></tbody></table>	Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y hora de inserción del nivel	F_Baja_N	Fecha y hora de baja del nivel	Descripcion	Descripción del nivel de acceso	0		2010-01-01 00:00:01		NULL		Acceso restringido: Zonas comunes		1		2010-01-01 00:00:01		NULL		Acceso total: Zonas comunes + Host	
Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y hora de inserción del nivel	F_Baja_N	Fecha y hora de baja del nivel	Descripcion	Descripción del nivel de acceso																		
0		2010-01-01 00:00:01		NULL		Acceso restringido: Zonas comunes																			
1		2010-01-01 00:00:01		NULL		Acceso total: Zonas comunes + Host																			
Evaluación/ conclusión	Los disparadores <i>actu_tarj</i> y <i>borr_tarj</i> funcionan correctamente.																								

*Casos de prueba para los disparadores *actu_tarj* y *borr_tarj**



Caso de prueba	1.07 - Disparador <i>borr_zona</i>																																																								
Objetivo	Verificar que al borrar una zona desaparecen los llaveros que queden sin llaves y los almacenes vacíos.																																																								
Actividades	Probar a borrar una zona cuyas llaves formen un llavero almacenado como único elemento en un almacén y cerciorarnos que tanto el llavero como el almacén desaparecen de la base de datos.																																																								
Técnicas y herramientas	Utilidad phpMyAdmin.																																																								
Datos de entrada	<ul style="list-style-type: none"> • Creamos un almacén con un llavero que solo disponga de llaves de una zona: <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: @ INSERT INTO Almacen VALUES ('PRU','Almacén de prueba'); INSERT INTO Llavero VALUES (9, '2010-01-01 00:00:01', NULL, 'Llavero de prueba'); INSERT INTO Composicion_Llavero VALUES (9, '2010-01-01 00:00:01', 0, '2010-01-01 00:00:01'); INSERT INTO Composicion_Llavero VALUES (9, '2010-01-01 00:00:01', 1, '2010-01-01 00:00:01'); INSERT INTO Almacen_Llavero VALUES ('PRU', 9, '2010-01-01 00:00:01', 1);</pre> <ul style="list-style-type: none"> • Disparador <i>borr_zona</i>. • Borramos la zona cuyas llaves pertenecen al llavero: <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: @ DELETE FROM Zona WHERE Cod_Zona = "GRUPELEC";</pre>																																																								
Salida esperada	El llavero y el almacén que toman parte en el caso de prueba deberían dejar de pertenecer a las tablas Llavero y Almacén respectivamente.																																																								
Salida observada	<p>La esperada:</p> <table border="1"> <thead> <tr> <th>Cod_Alm</th> <th>Código de almacén</th> <th>Descripcion</th> <th>Descripción del almacén</th> </tr> </thead> <tbody> <tr> <td>A</td> <td></td> <td>Armario</td> <td></td> </tr> <tr> <td>CC</td> <td></td> <td>Caja de copias</td> <td></td> </tr> <tr> <td>V</td> <td></td> <td>Varios</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Cod_Llavero</th> <th>Número de llavero</th> <th>F_Alta_Ll</th> <th>Fecha y hora de inserción del llavero</th> <th>F_Baja_Ll</th> <th>Fecha y hora de baja del llavero</th> <th>Descripcion</th> <th>Descripción del llavero</th> </tr> </thead> <tbody> <tr> <td></td> <td>0</td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Grupo Electrogeno</td> <td></td> </tr> <tr> <td></td> <td>1</td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Sala Maquinas</td> <td></td> </tr> <tr> <td></td> <td>2</td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>UPS</td> <td></td> </tr> <tr> <td></td> <td>3</td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Sala Comunicaciones</td> <td></td> </tr> </tbody> </table>	Cod_Alm	Código de almacén	Descripcion	Descripción del almacén	A		Armario		CC		Caja de copias		V		Varios		Cod_Llavero	Número de llavero	F_Alta_Ll	Fecha y hora de inserción del llavero	F_Baja_Ll	Fecha y hora de baja del llavero	Descripcion	Descripción del llavero		0	2010-01-01 00:00:01		NULL		Grupo Electrogeno			1	2010-01-01 00:00:01		NULL		Sala Maquinas			2	2010-01-01 00:00:01		NULL		UPS			3	2010-01-01 00:00:01		NULL		Sala Comunicaciones	
Cod_Alm	Código de almacén	Descripcion	Descripción del almacén																																																						
A		Armario																																																							
CC		Caja de copias																																																							
V		Varios																																																							
Cod_Llavero	Número de llavero	F_Alta_Ll	Fecha y hora de inserción del llavero	F_Baja_Ll	Fecha y hora de baja del llavero	Descripcion	Descripción del llavero																																																		
	0	2010-01-01 00:00:01		NULL		Grupo Electrogeno																																																			
	1	2010-01-01 00:00:01		NULL		Sala Maquinas																																																			
	2	2010-01-01 00:00:01		NULL		UPS																																																			
	3	2010-01-01 00:00:01		NULL		Sala Comunicaciones																																																			
Evaluación/ conclusión	El disparador <i>borr_zona</i> funciona correctamente.																																																								

*Caso de prueba para el disparador *borr_zona**



Caso de prueba	1.08 - Disparador <i>ins_llave</i>																																			
Objetivo	Comprobar que la inserción de una nueva llave tiene como paso previo la baja de otra si es que ambas comparten código.																																			
Actividades	Probar a insertar una llave con código existente para así comprobar que la inserción de la nueva tiene como primera consecuencia la baja de la existente.																																			
Técnicas y herramientas	Utilidad phpMyAdmin.																																			
Datos de entrada	<ul style="list-style-type: none"> Estado inicial de la tabla: <table border="1"> <thead> <tr> <th>Cod_Llave</th> <th>F_Alta_L Fecha y hora de inserción de la llave</th> <th>F_Baja_L Fecha y hora de baja de la llave</th> <th>Cod_Zona Zona a la que pertenece</th> <th>F_Alta_Z Fecha y hora de inserción de la zona</th> <th>Descripcion Descripción de la cerradura que abre</th> <th>Notas Otros datos a tener en cuenta</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>GRUPELEC</td> <td>2010-01-01 00:00:01</td> <td>Puerta de acceso</td> <td>NULL</td> </tr> <tr> <td>1</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>GRUPELEC</td> <td>2010-01-01 00:00:01</td> <td>Armario técnico</td> <td>NULL</td> </tr> <tr> <td>2</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>GRUPELEC</td> <td>2010-01-01 00:00:01</td> <td>Armario principal</td> <td>NULL</td> </tr> <tr> <td>3</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>GRUPELEC</td> <td>2010-01-01 00:00:01</td> <td>Salida a patio</td> <td>NULL</td> </tr> </tbody> </table> <ul style="list-style-type: none"> Disparador <i>ins_llave</i>. Inserción de una llave: <p>Ejecutar la(s) consulta(s) SQL en la base de datos bd: <input type="text"/></p> <pre>INSERT INTO Llave VALUES (0,CURDATE(),NULL,'GRUPELEC','2010-01-01 00:00:01','Llave de prueba',NULL)</pre>	Cod_Llave	F_Alta_L Fecha y hora de inserción de la llave	F_Baja_L Fecha y hora de baja de la llave	Cod_Zona Zona a la que pertenece	F_Alta_Z Fecha y hora de inserción de la zona	Descripcion Descripción de la cerradura que abre	Notas Otros datos a tener en cuenta	0	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Puerta de acceso	NULL	1	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Armario técnico	NULL	2	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Armario principal	NULL	3	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Salida a patio	NULL
Cod_Llave	F_Alta_L Fecha y hora de inserción de la llave	F_Baja_L Fecha y hora de baja de la llave	Cod_Zona Zona a la que pertenece	F_Alta_Z Fecha y hora de inserción de la zona	Descripcion Descripción de la cerradura que abre	Notas Otros datos a tener en cuenta																														
0	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Puerta de acceso	NULL																														
1	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Armario técnico	NULL																														
2	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Armario principal	NULL																														
3	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Salida a patio	NULL																														
Salida esperada	La llave con código 0 deberá estar representada en dos tuplas, una con fecha de baja y otra sin ella.																																			
Salida observada	La esperada:																																			
	<table border="1"> <thead> <tr> <th>Cod_Llave</th> <th>F_Alta_L Fecha y hora de inserción de la llave</th> <th>F_Baja_L Fecha y hora de baja de la llave</th> <th>Cod_Zona Zona a la que pertenece</th> <th>F_Alta_Z Fecha y hora de inserción de la zona</th> <th>Descripcion Descripción de la cerradura que abre</th> <th>Notas Otros datos a tener en cuenta</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>2010-01-01 00:00:01</td> <td>2010-06-17 00:00:00</td> <td>GRUPELEC</td> <td>2010-01-01 00:00:01</td> <td>Puerta de acceso</td> <td>NULL</td> </tr> <tr> <td>0</td> <td>2010-06-17 00:00:00</td> <td>NULL</td> <td>GRUPELEC</td> <td>2010-01-01 00:00:01</td> <td>Llave de prueba</td> <td>NULL</td> </tr> <tr> <td>1</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>GRUPELEC</td> <td>2010-01-01 00:00:01</td> <td>Armario técnico</td> <td>NULL</td> </tr> <tr> <td>2</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>GRUPELEC</td> <td>2010-01-01 00:00:01</td> <td>Armario principal</td> <td>NULL</td> </tr> </tbody> </table>	Cod_Llave	F_Alta_L Fecha y hora de inserción de la llave	F_Baja_L Fecha y hora de baja de la llave	Cod_Zona Zona a la que pertenece	F_Alta_Z Fecha y hora de inserción de la zona	Descripcion Descripción de la cerradura que abre	Notas Otros datos a tener en cuenta	0	2010-01-01 00:00:01	2010-06-17 00:00:00	GRUPELEC	2010-01-01 00:00:01	Puerta de acceso	NULL	0	2010-06-17 00:00:00	NULL	GRUPELEC	2010-01-01 00:00:01	Llave de prueba	NULL	1	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Armario técnico	NULL	2	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Armario principal	NULL
Cod_Llave	F_Alta_L Fecha y hora de inserción de la llave	F_Baja_L Fecha y hora de baja de la llave	Cod_Zona Zona a la que pertenece	F_Alta_Z Fecha y hora de inserción de la zona	Descripcion Descripción de la cerradura que abre	Notas Otros datos a tener en cuenta																														
0	2010-01-01 00:00:01	2010-06-17 00:00:00	GRUPELEC	2010-01-01 00:00:01	Puerta de acceso	NULL																														
0	2010-06-17 00:00:00	NULL	GRUPELEC	2010-01-01 00:00:01	Llave de prueba	NULL																														
1	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Armario técnico	NULL																														
2	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Armario principal	NULL																														
Evaluación/ conclusión	El disparador <i>ins_llave</i> funciona correctamente.																																			

*Caso de prueba para el disparador *ins_llave**



Caso de prueba	1.09 - Disparador <i>borr_I</i>																																																								
Objetivo	Probar que el disparador <i>borr_I</i> , ejecutado tras el borrado de un llavero, lleva a cabo toda la funcionalidad para la que fue diseñado.																																																								
Actividades	Crear la población necesaria para que la eliminación de una llave implique la ejecución del disparador y podamos así comprobar su eficacia.																																																								
Técnicas y herramientas	Utilidad phpMyAdmin.																																																								
Datos de entrada	<ul style="list-style-type: none"> • Creamos un almacén con un llavero: <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: @</pre> <pre>INSERT INTO Almacen VALUES ('PRU','Almacén de prueba'); INSERT INTO Llavero VALUES (9, '2010-01-01 00:00:01', NULL, 'Llavero de prueba'); INSERT INTO Composicion_Llavero VALUES (9, '2010-01-01 00:00:01', 0, '2010-01-01 00:00:01'); INSERT INTO Composicion_Llavero VALUES (9, '2010-01-01 00:00:01', 1, '2010-01-01 00:00:01'); INSERT INTO Almacen_Llavero VALUES ('PRU', 9, '2010-01-01 00:00:01', 1);</pre> <ul style="list-style-type: none"> • Disparador <i>borr_I</i>. • Borramos las dos llaves que componen el llavero: <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: @</pre> <pre>DELETE FROM Llave WHERE (Cod_Llave,F_Alta_L) IN (SELECT Cod_Llave,F_Alta_L FROM Llavero WHERE Cod_Llavero = 9);</pre>																																																								
Salida esperada	La misma que en el caso de prueba 1.07: ni el almacén ni el llavero creado deberán existir en la base de datos.																																																								
Salida observada	La esperada: <table border="1"> <thead> <tr> <th>Cod_Alm</th> <th>Código de almacén</th> <th>Descripcion</th> <th>Descripción del almacén</th> </tr> </thead> <tbody> <tr> <td>A</td> <td></td> <td>Armarío</td> <td></td> </tr> <tr> <td>CC</td> <td></td> <td>Caja de copias</td> <td></td> </tr> <tr> <td>V</td> <td></td> <td>Varios</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Cod_Llavero</th> <th>Número de llavero</th> <th>F_Alta_Ll</th> <th>Fecha y hora de inserción del llavero</th> <th>F_Baja_Ll</th> <th>Fecha y hora de baja del llavero</th> <th>Descripcion</th> <th>Descripción del llavero</th> </tr> </thead> <tbody> <tr> <td></td> <td>0</td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Grupo Electrogeno</td> <td></td> </tr> <tr> <td></td> <td>1</td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Sala Maquinas</td> <td></td> </tr> <tr> <td></td> <td>2</td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>UPS</td> <td></td> </tr> <tr> <td></td> <td>3</td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Sala Comunicaciones</td> <td></td> </tr> </tbody> </table>	Cod_Alm	Código de almacén	Descripcion	Descripción del almacén	A		Armarío		CC		Caja de copias		V		Varios		Cod_Llavero	Número de llavero	F_Alta_Ll	Fecha y hora de inserción del llavero	F_Baja_Ll	Fecha y hora de baja del llavero	Descripcion	Descripción del llavero		0	2010-01-01 00:00:01		NULL		Grupo Electrogeno			1	2010-01-01 00:00:01		NULL		Sala Maquinas			2	2010-01-01 00:00:01		NULL		UPS			3	2010-01-01 00:00:01		NULL		Sala Comunicaciones	
Cod_Alm	Código de almacén	Descripcion	Descripción del almacén																																																						
A		Armarío																																																							
CC		Caja de copias																																																							
V		Varios																																																							
Cod_Llavero	Número de llavero	F_Alta_Ll	Fecha y hora de inserción del llavero	F_Baja_Ll	Fecha y hora de baja del llavero	Descripcion	Descripción del llavero																																																		
	0	2010-01-01 00:00:01		NULL		Grupo Electrogeno																																																			
	1	2010-01-01 00:00:01		NULL		Sala Maquinas																																																			
	2	2010-01-01 00:00:01		NULL		UPS																																																			
	3	2010-01-01 00:00:01		NULL		Sala Comunicaciones																																																			
Evaluación/ conclusión	El disparador <i>borr_I</i> funciona correctamente.																																																								

*Caso de prueba para el disparador *borr_I**



Caso de prueba	1.10 - Disparador <i>borr_al</i>																																																																								
Objetivo	Comprobar que al eliminar una llave de un almacén solo siguen almacenándose aquellas llaves que dispongan de copias por si mismas o en algún llavero y aquellos almacenes que tengan algún elemento en su interior.																																																																								
Actividades	Crear un almacén en el que almacenaremos una llave nueva. Eliminamos las copias de la llave y comprobamos que llave y almacén desaparecen.																																																																								
Técnicas y herramientas	Utilidad phpMyAdmin.																																																																								
Datos de entrada	<ul style="list-style-type: none"> Registros de prueba: <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: ?</pre> <pre>INSERT INTO Almacen VALUES ('AP','Almacén de pruebas'); INSERT INTO Llave VALUES (999,'2010-01-01 00:00:01',NULL,'CONDENS','2010-01-01 00:00:01','Descripcion de prueba',NULL); INSERT INTO Almacen_Llave VALUES ('AP',999,'2010-01-01 00:00:01',3);</pre> <ul style="list-style-type: none"> Disparador <i>borr_al</i>. Borrado de la llave del almacén: <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: ?</pre> <pre>DELETE FROM Almacen_Llave WHERE Cod_Alm = 'AP' AND Cod_Llave = 999 AND F_Alta_L = '2010-01-01 00:00:01';</pre>																																																																								
Salida esperada	La llave y almacén introducidos no constan en la base de datos.																																																																								
Salida observada	<p>La esperada:</p> <table border="1"> <thead> <tr> <th>Cod_Llave</th> <th>F_Alta_L Fecha y hora de inserción de la llave</th> <th>F_Baja_L Fecha y hora de baja de la llave</th> <th>Cod_Zona Zona a la que pertenece</th> <th>F_Alta_Z Fecha y hora de inserción de la zona</th> <th>Descripcion</th> <th>Notas Otros datos a tener en cuenta</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>GRUPELEC</td> <td>2010-01-01 00:00:01</td> <td>Puerta de acceso</td> <td>NULL</td> </tr> <tr> <td>1</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>GRUPELEC</td> <td>2010-01-01 00:00:01</td> <td>Armario técnico</td> <td>NULL</td> </tr> <tr> <td>2</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>GRUPELEC</td> <td>2010-01-01 00:00:01</td> <td>Armario principal</td> <td>NULL</td> </tr> <tr> <td>3</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>GRUPELEC</td> <td>2010-01-01 00:00:01</td> <td>Salida a patio</td> <td>NULL</td> </tr> <tr> <td colspan="7" style="text-align: center;">...</td> </tr> <tr> <td>25</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>SALACOMS</td> <td>2010-01-01 00:00:01</td> <td>Servidores</td> <td>NULL</td> </tr> <tr> <td>26</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>SALACOMS</td> <td>2010-01-01 00:00:01</td> <td>Mantenimiento</td> <td>NULL</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Cod_Alm</th> <th>Código de almacén</th> <th>Descripcion</th> <th>Descripción del almacén</th> </tr> </thead> <tbody> <tr> <td>A</td> <td></td> <td>Armario</td> <td></td> </tr> <tr> <td>CC</td> <td></td> <td>Caja de copias</td> <td></td> </tr> <tr> <td>V</td> <td></td> <td>Varios</td> <td></td> </tr> </tbody> </table>	Cod_Llave	F_Alta_L Fecha y hora de inserción de la llave	F_Baja_L Fecha y hora de baja de la llave	Cod_Zona Zona a la que pertenece	F_Alta_Z Fecha y hora de inserción de la zona	Descripcion	Notas Otros datos a tener en cuenta	0	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Puerta de acceso	NULL	1	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Armario técnico	NULL	2	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Armario principal	NULL	3	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Salida a patio	NULL	...							25	2010-01-01 00:00:01	NULL	SALACOMS	2010-01-01 00:00:01	Servidores	NULL	26	2010-01-01 00:00:01	NULL	SALACOMS	2010-01-01 00:00:01	Mantenimiento	NULL	Cod_Alm	Código de almacén	Descripcion	Descripción del almacén	A		Armario		CC		Caja de copias		V		Varios	
Cod_Llave	F_Alta_L Fecha y hora de inserción de la llave	F_Baja_L Fecha y hora de baja de la llave	Cod_Zona Zona a la que pertenece	F_Alta_Z Fecha y hora de inserción de la zona	Descripcion	Notas Otros datos a tener en cuenta																																																																			
0	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Puerta de acceso	NULL																																																																			
1	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Armario técnico	NULL																																																																			
2	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Armario principal	NULL																																																																			
3	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Salida a patio	NULL																																																																			
...																																																																									
25	2010-01-01 00:00:01	NULL	SALACOMS	2010-01-01 00:00:01	Servidores	NULL																																																																			
26	2010-01-01 00:00:01	NULL	SALACOMS	2010-01-01 00:00:01	Mantenimiento	NULL																																																																			
Cod_Alm	Código de almacén	Descripcion	Descripción del almacén																																																																						
A		Armario																																																																							
CC		Caja de copias																																																																							
V		Varios																																																																							
Evaluación/ conclusión	El disparador <i>borr_al</i> funciona correctamente.																																																																								

*Caso de prueba para el disparador *borr_al**



Caso de prueba	1.11 - Disparador <i>ins_llavero</i>																																																
Objetivo	Chequear que el alta de un llavero cuyo código ya exista en la tabla Llavero hace que el SGBD dé de baja al existente antes de introducir el nuevo.																																																
Actividades	Acceder a phpMyAdmin y probar a introducir un nuevo llavero cuyo código ya exista, pasando a verificar tras la inserción que el viejo dispone en esta ocasión de fecha de baja y el nuevo ha sido registrado correctamente.																																																
Técnicas y herramientas	Utilidad phpMyAdmin.																																																
Datos de entrada	<ul style="list-style-type: none"> Estado inicial de la tabla: <table border="1"> <thead> <tr> <th>Cod_Llavero</th> <th>Número de llavero</th> <th>F_Alta_Ll</th> <th>Fecha y hora de inserción del llavero</th> <th>F_Baja_Ll</th> <th>Fecha y hora de baja del llavero</th> <th>Descripcion</th> <th>Descripción del llavero</th> </tr> </thead> <tbody> <tr> <td></td> <td>0</td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Grupo Electrogeno</td> <td></td> </tr> <tr> <td></td> <td>1</td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Sala Maquinas</td> <td></td> </tr> <tr> <td></td> <td>2</td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>UPS</td> <td></td> </tr> <tr> <td></td> <td>3</td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Sala Comunicaciones</td> <td></td> </tr> </tbody> </table> <ul style="list-style-type: none"> Disparador <i>ins_llavero</i>. Inserción de un llavero: <p>Ejecutar la(s) consulta(s) SQL en la base de datos bd: <input type="text"/></p> <pre>INSERT INTO Llavero VALUES (0, CURDATE(), NULL, 'Llavero de prueba');</pre>	Cod_Llavero	Número de llavero	F_Alta_Ll	Fecha y hora de inserción del llavero	F_Baja_Ll	Fecha y hora de baja del llavero	Descripcion	Descripción del llavero		0	2010-01-01 00:00:01		NULL		Grupo Electrogeno			1	2010-01-01 00:00:01		NULL		Sala Maquinas			2	2010-01-01 00:00:01		NULL		UPS			3	2010-01-01 00:00:01		NULL		Sala Comunicaciones									
Cod_Llavero	Número de llavero	F_Alta_Ll	Fecha y hora de inserción del llavero	F_Baja_Ll	Fecha y hora de baja del llavero	Descripcion	Descripción del llavero																																										
	0	2010-01-01 00:00:01		NULL		Grupo Electrogeno																																											
	1	2010-01-01 00:00:01		NULL		Sala Maquinas																																											
	2	2010-01-01 00:00:01		NULL		UPS																																											
	3	2010-01-01 00:00:01		NULL		Sala Comunicaciones																																											
Salida esperada	Esperamos disponer en Llavero de dos llaveros con código 0, que tendrán los valores de '2010-06-17 00:00:00' y NULL en sus respectivas fechas de baja.																																																
Salida observada	La esperada:																																																
	<table border="1"> <thead> <tr> <th>Cod_Llavero</th> <th>Número de llavero</th> <th>F_Alta_Ll</th> <th>Fecha y hora de inserción del llavero</th> <th>F_Baja_Ll</th> <th>Fecha y hora de baja del llavero</th> <th>Descripcion</th> <th>Descripción del llavero</th> </tr> </thead> <tbody> <tr> <td></td> <td>0</td> <td>2010-01-01 00:00:01</td> <td></td> <td>2010-06-17 00:00:00</td> <td></td> <td>Grupo Electrogeno</td> <td></td> </tr> <tr> <td></td> <td>0</td> <td>2010-06-17 00:00:00</td> <td></td> <td>NULL</td> <td></td> <td>Llavero de prueba</td> <td></td> </tr> <tr> <td></td> <td>1</td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Sala Maquinas</td> <td></td> </tr> <tr> <td></td> <td>2</td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>UPS</td> <td></td> </tr> <tr> <td></td> <td>3</td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Sala Comunicaciones</td> <td></td> </tr> </tbody> </table>	Cod_Llavero	Número de llavero	F_Alta_Ll	Fecha y hora de inserción del llavero	F_Baja_Ll	Fecha y hora de baja del llavero	Descripcion	Descripción del llavero		0	2010-01-01 00:00:01		2010-06-17 00:00:00		Grupo Electrogeno			0	2010-06-17 00:00:00		NULL		Llavero de prueba			1	2010-01-01 00:00:01		NULL		Sala Maquinas			2	2010-01-01 00:00:01		NULL		UPS			3	2010-01-01 00:00:01		NULL		Sala Comunicaciones	
Cod_Llavero	Número de llavero	F_Alta_Ll	Fecha y hora de inserción del llavero	F_Baja_Ll	Fecha y hora de baja del llavero	Descripcion	Descripción del llavero																																										
	0	2010-01-01 00:00:01		2010-06-17 00:00:00		Grupo Electrogeno																																											
	0	2010-06-17 00:00:00		NULL		Llavero de prueba																																											
	1	2010-01-01 00:00:01		NULL		Sala Maquinas																																											
	2	2010-01-01 00:00:01		NULL		UPS																																											
	3	2010-01-01 00:00:01		NULL		Sala Comunicaciones																																											
Evaluación/ conclusión	El disparador <i>ins_llavero</i> funciona correctamente.																																																

Caso de prueba para el disparador ins_llavero



Caso de prueba	1.12 - Disparador <i>borr_II</i>																																																																																													
Objetivo	Comprobar que el disparador <i>borr_II</i> se ejecuta tras el borrado y realiza todas las acciones que esperamos de él, que son el borrado de todas las llaves que no pertenecen a ningún llavero ni tienen existencia propia, así como los almacenes que no atesoran ninguna llave o llavero.																																																																																													
Actividades	Añadir al sistema la población necesaria para realizar este caso de prueba, borrar el llavero y comprobar que tanto la llave como el almacén (ambos nuevos) ya no forman de la base de datos.																																																																																													
Técnicas y herramientas	Utilidad phpMyAdmin.																																																																																													
Datos de entrada	<ul style="list-style-type: none"> • Creamos un almacén con un llavero cuya única llave solo exista en éste: <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: @</pre> <pre>INSERT INTO Almacen VALUES ('PRU','Almacén de prueba'); INSERT INTO Llave VALUES (27,'2010-01-01 00:00:01',NULL,'GRUPELEC','2010-01-01 00:00:01','Llave de prueba',NULL); INSERT INTO Llavero VALUES (9,'2010-01-01 00:00:01',NULL,'Llavero de prueba'); INSERT INTO Composicion_Llavero VALUES (9,'2010-01-01 00:00:01',27,'2010-01-01 00:00:01'); INSERT INTO Almacen_Llavero VALUES ('PRU',9,'2010-01-01 00:00:01',1);</pre> <ul style="list-style-type: none"> • Disparador <i>borr_II</i>. • Borramos el llavero: <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: @</pre> <pre>DELETE FROM Llave WHERE (Cod_Llave,F_Alta_L) IN (SELECT Cod_Llave,F_Alta_L FROM Llavero WHERE Cod_Llavero = 9);</pre>																																																																																													
Salida esperada	La misma que en el caso de prueba 1.07: ni el almacén ni el llavero creado deberán existir en la base de datos.																																																																																													
Salida observada	<p>La esperada:</p> <table border="1"> <thead> <tr> <th>Cod_Alm</th> <th>Código de almacén</th> <th>Descripcion</th> <th>Descripción del almacén</th> </tr> </thead> <tbody> <tr> <td>A</td> <td></td> <td>Armarío</td> <td></td> </tr> <tr> <td>CC</td> <td></td> <td>Caja de copias</td> <td></td> </tr> <tr> <td>V</td> <td></td> <td>Varios</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Cod_Llave</th> <th>F_Alta_L</th> <th>F_Baja_L</th> <th>Cod_Zona</th> <th>F_Alta_Z</th> <th>Descripcion</th> <th>Notas</th> </tr> <tr> <th>Número de llave</th> <th>Fecha y hora de inserción de la llave</th> <th>Fecha y hora de baja de la llave</th> <th>Zona a la que pertenece</th> <th>Fecha y hora de inserción de la zona</th> <th>Descripción de la cerradura que abre</th> <th>Otros datos a tener en cuenta</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>GRUPELEC</td> <td>2010-01-01 00:00:01</td> <td>Puerta de acceso</td> <td>NULL</td> </tr> <tr> <td>1</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>GRUPELEC</td> <td>2010-01-01 00:00:01</td> <td>Armarío técnico</td> <td>NULL</td> </tr> <tr> <td>2</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>GRUPELEC</td> <td>2010-01-01 00:00:01</td> <td>Armarío principal</td> <td>NULL</td> </tr> <tr> <td>3</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>GRUPELEC</td> <td>2010-01-01 00:00:01</td> <td>Salida a patio</td> <td>NULL</td> </tr> <tr> <td colspan="7" style="text-align: center;">...</td> </tr> <tr> <td>23</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>SALACOMS</td> <td>2010-01-01 00:00:01</td> <td>Armarío 2</td> <td>NULL</td> </tr> <tr> <td>24</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>SALACOMS</td> <td>2010-01-01 00:00:01</td> <td>Técnicos</td> <td>NULL</td> </tr> <tr> <td>25</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>SALACOMS</td> <td>2010-01-01 00:00:01</td> <td>Servidores</td> <td>NULL</td> </tr> <tr> <td>26</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>SALACOMS</td> <td>2010-01-01 00:00:01</td> <td>Mantenimiento</td> <td>NULL</td> </tr> </tbody> </table>	Cod_Alm	Código de almacén	Descripcion	Descripción del almacén	A		Armarío		CC		Caja de copias		V		Varios		Cod_Llave	F_Alta_L	F_Baja_L	Cod_Zona	F_Alta_Z	Descripcion	Notas	Número de llave	Fecha y hora de inserción de la llave	Fecha y hora de baja de la llave	Zona a la que pertenece	Fecha y hora de inserción de la zona	Descripción de la cerradura que abre	Otros datos a tener en cuenta	0	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Puerta de acceso	NULL	1	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Armarío técnico	NULL	2	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Armarío principal	NULL	3	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Salida a patio	NULL	...							23	2010-01-01 00:00:01	NULL	SALACOMS	2010-01-01 00:00:01	Armarío 2	NULL	24	2010-01-01 00:00:01	NULL	SALACOMS	2010-01-01 00:00:01	Técnicos	NULL	25	2010-01-01 00:00:01	NULL	SALACOMS	2010-01-01 00:00:01	Servidores	NULL	26	2010-01-01 00:00:01	NULL	SALACOMS	2010-01-01 00:00:01	Mantenimiento	NULL
Cod_Alm	Código de almacén	Descripcion	Descripción del almacén																																																																																											
A		Armarío																																																																																												
CC		Caja de copias																																																																																												
V		Varios																																																																																												
Cod_Llave	F_Alta_L	F_Baja_L	Cod_Zona	F_Alta_Z	Descripcion	Notas																																																																																								
Número de llave	Fecha y hora de inserción de la llave	Fecha y hora de baja de la llave	Zona a la que pertenece	Fecha y hora de inserción de la zona	Descripción de la cerradura que abre	Otros datos a tener en cuenta																																																																																								
0	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Puerta de acceso	NULL																																																																																								
1	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Armarío técnico	NULL																																																																																								
2	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Armarío principal	NULL																																																																																								
3	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Salida a patio	NULL																																																																																								
...																																																																																														
23	2010-01-01 00:00:01	NULL	SALACOMS	2010-01-01 00:00:01	Armarío 2	NULL																																																																																								
24	2010-01-01 00:00:01	NULL	SALACOMS	2010-01-01 00:00:01	Técnicos	NULL																																																																																								
25	2010-01-01 00:00:01	NULL	SALACOMS	2010-01-01 00:00:01	Servidores	NULL																																																																																								
26	2010-01-01 00:00:01	NULL	SALACOMS	2010-01-01 00:00:01	Mantenimiento	NULL																																																																																								
Evaluación/ conclusión	El disparador <i>borr_II</i> funciona correctamente.																																																																																													

*Caso de prueba para el disparador *borr_II**



Caso de prueba	1.13 - Disparador <i>borr_all</i>																																																																																																																					
Objetivo	Demostrar que al borrar un llavero de un almacén solo siguen persistiendo aquellos llaveros que dispongan de alguna copia en algún almacén, aquellas llaves que dispongan de existencia propia o permanezcan a algún otro llavero y aquellos almacenes que no estén vacíos.																																																																																																																					
Actividades	Crear un almacén y un llavero (compuesto por una llave que solo exista en éste), introducirlo el segundo en el primero y acto seguido eliminarlo. Llavero, llave y almacén deberán ser borrados de la base de datos automáticamente.																																																																																																																					
Técnicas y herramientas	Utilidad phpMyAdmin.																																																																																																																					
Datos de entrada	<ul style="list-style-type: none"> Registros de prueba: <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: INSERT INTO Almacen VALUES ('PRU','Almacén de prueba'); INSERT INTO Llave VALUES (27,'2010-01-01 00:00:01',NULL,'GRUPELEC','2010-01-01 00:00:01','Llave de prueba',NULL); INSERT INTO Llavero VALUES (9,'2010-01-01 00:00:01',NULL,'Llavero de prueba'); INSERT INTO Composicion_Llavero VALUES (9,'2010-01-01 00:00:01',27,'2010-01-01 00:00:01'); INSERT INTO Almacen_Llavero VALUES ('PRU',9,'2010-01-01 00:00:01',1);</pre> <ul style="list-style-type: none"> Disparador <i>borr_all</i>. Borrado del llavero del almacén: <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: DELETE FROM Almacen_Llavero WHERE Cod_Llavero = 9;</pre>																																																																																																																					
Salida esperada	Almacen, Llave y Llavero no deberán contener registros cuyas claves primarias sean 'PRU', '27,2010-01-01 00:00:01' y '9, 2010-01-01 00:00:01' respectivamente.																																																																																																																					
Salida observada	<p>La esperada:</p> <table border="1"> <thead> <tr> <th>Cod_Alm</th> <th>Código de almacén</th> <th>Descripcion</th> <th>Descripción del almacén</th> </tr> </thead> <tbody> <tr> <td>A</td> <td></td> <td>Armario</td> <td></td> </tr> <tr> <td>CC</td> <td></td> <td>Caja de copias</td> <td></td> </tr> <tr> <td>V</td> <td></td> <td>Varios</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Cod_Llave</th> <th>F_Alta_L</th> <th>F_Baja_L</th> <th>Cod_Zona</th> <th>F_Alta_Z</th> <th>Descripcion</th> <th>Notas</th> </tr> <tr> <th>Número de llave</th> <th>Fecha y hora de inserción de la llave</th> <th>Fecha y hora de baja de la llave</th> <th>Zona a la que pertenece</th> <th>Fecha y hora de inserción de la zona</th> <th>Descripción de la cerradura que abre</th> <th>Otros datos a tener en cuenta</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>GRUPELEC</td> <td>2010-01-01 00:00:01</td> <td>Puerta de acceso</td> <td>NULL</td> </tr> <tr> <td>1</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>GRUPELEC</td> <td>2010-01-01 00:00:01</td> <td>Armario técnico</td> <td>NULL</td> </tr> <tr> <td>2</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>GRUPELEC</td> <td>2010-01-01 00:00:01</td> <td>Armario principal</td> <td>NULL</td> </tr> <tr> <td>3</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>GRUPELEC</td> <td>2010-01-01 00:00:01</td> <td>Salida a patio</td> <td>NULL</td> </tr> <tr> <td colspan="7" style="text-align: center;">...</td> </tr> <tr> <td>23</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>SALACOMS</td> <td>2010-01-01 00:00:01</td> <td>Armario 2</td> <td>NULL</td> </tr> <tr> <td>24</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>SALACOMS</td> <td>2010-01-01 00:00:01</td> <td>Técnicos</td> <td>NULL</td> </tr> <tr> <td>25</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>SALACOMS</td> <td>2010-01-01 00:00:01</td> <td>Servidores</td> <td>NULL</td> </tr> <tr> <td>26</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>SALACOMS</td> <td>2010-01-01 00:00:01</td> <td>Mantenimiento</td> <td>NULL</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Cod_Llavero</th> <th>F_Alta_LI</th> <th>F_Baja_LI</th> <th>Descripcion</th> </tr> <tr> <th>Número de llavero</th> <th>Fecha y hora de inserción del llavero</th> <th>Fecha y hora de baja del llavero</th> <th>Descripción del llavero</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>Grupo Electrogeno</td> </tr> <tr> <td>1</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>Sala Maquinas</td> </tr> <tr> <td>2</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>UPS</td> </tr> <tr> <td>3</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>Sala Comunicaciones</td> </tr> </tbody> </table>	Cod_Alm	Código de almacén	Descripcion	Descripción del almacén	A		Armario		CC		Caja de copias		V		Varios		Cod_Llave	F_Alta_L	F_Baja_L	Cod_Zona	F_Alta_Z	Descripcion	Notas	Número de llave	Fecha y hora de inserción de la llave	Fecha y hora de baja de la llave	Zona a la que pertenece	Fecha y hora de inserción de la zona	Descripción de la cerradura que abre	Otros datos a tener en cuenta	0	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Puerta de acceso	NULL	1	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Armario técnico	NULL	2	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Armario principal	NULL	3	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Salida a patio	NULL	...							23	2010-01-01 00:00:01	NULL	SALACOMS	2010-01-01 00:00:01	Armario 2	NULL	24	2010-01-01 00:00:01	NULL	SALACOMS	2010-01-01 00:00:01	Técnicos	NULL	25	2010-01-01 00:00:01	NULL	SALACOMS	2010-01-01 00:00:01	Servidores	NULL	26	2010-01-01 00:00:01	NULL	SALACOMS	2010-01-01 00:00:01	Mantenimiento	NULL	Cod_Llavero	F_Alta_LI	F_Baja_LI	Descripcion	Número de llavero	Fecha y hora de inserción del llavero	Fecha y hora de baja del llavero	Descripción del llavero	0	2010-01-01 00:00:01	NULL	Grupo Electrogeno	1	2010-01-01 00:00:01	NULL	Sala Maquinas	2	2010-01-01 00:00:01	NULL	UPS	3	2010-01-01 00:00:01	NULL	Sala Comunicaciones
Cod_Alm	Código de almacén	Descripcion	Descripción del almacén																																																																																																																			
A		Armario																																																																																																																				
CC		Caja de copias																																																																																																																				
V		Varios																																																																																																																				
Cod_Llave	F_Alta_L	F_Baja_L	Cod_Zona	F_Alta_Z	Descripcion	Notas																																																																																																																
Número de llave	Fecha y hora de inserción de la llave	Fecha y hora de baja de la llave	Zona a la que pertenece	Fecha y hora de inserción de la zona	Descripción de la cerradura que abre	Otros datos a tener en cuenta																																																																																																																
0	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Puerta de acceso	NULL																																																																																																																
1	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Armario técnico	NULL																																																																																																																
2	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Armario principal	NULL																																																																																																																
3	2010-01-01 00:00:01	NULL	GRUPELEC	2010-01-01 00:00:01	Salida a patio	NULL																																																																																																																
...																																																																																																																						
23	2010-01-01 00:00:01	NULL	SALACOMS	2010-01-01 00:00:01	Armario 2	NULL																																																																																																																
24	2010-01-01 00:00:01	NULL	SALACOMS	2010-01-01 00:00:01	Técnicos	NULL																																																																																																																
25	2010-01-01 00:00:01	NULL	SALACOMS	2010-01-01 00:00:01	Servidores	NULL																																																																																																																
26	2010-01-01 00:00:01	NULL	SALACOMS	2010-01-01 00:00:01	Mantenimiento	NULL																																																																																																																
Cod_Llavero	F_Alta_LI	F_Baja_LI	Descripcion																																																																																																																			
Número de llavero	Fecha y hora de inserción del llavero	Fecha y hora de baja del llavero	Descripción del llavero																																																																																																																			
0	2010-01-01 00:00:01	NULL	Grupo Electrogeno																																																																																																																			
1	2010-01-01 00:00:01	NULL	Sala Maquinas																																																																																																																			
2	2010-01-01 00:00:01	NULL	UPS																																																																																																																			
3	2010-01-01 00:00:01	NULL	Sala Comunicaciones																																																																																																																			
Evaluación/conclusión	El disparador <i>borr_all</i> funciona correctamente.																																																																																																																					

*Caso de prueba para el disparador *borr_all**



Caso de prueba	1.14 - Disparador <i>borr_dep</i>																																																												
Objetivo	<p>Probar que al eliminar un departamento:</p> <ul style="list-style-type: none"> • El personal que no conste en ninguna tabla especializada es eliminado. • Toda empresa que no disponga de departamentos es borrada. • Los cargos no desempeñados no tienen cabida en el sistema. 																																																												
Actividades	Crear una relación Empresa-Departamento-Contacto-Cargo, de modo que al eliminar el departamento se pueda comprobar si el resto también han sido eliminados.																																																												
Técnicas y herramientas	Utilidad phpMyAdmin.																																																												
Datos de entrada	<ul style="list-style-type: none"> • Registros de prueba: <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: INSERT INTO Empresa VALUES ('EP','CIF','NomEmp','DirEmp'); INSERT INTO Departamento VALUES ('EP','DP','NomDep','MHC',100,CURDATE(),NULL); INSERT INTO Personal VALUES ('999999999A','A1','A2','N',NULL); INSERT INTO Cargo VALUES ('C','Descripcion de prueba'); INSERT INTO Contacto VALUES ('999999999A','EP','DP','C');</pre> <ul style="list-style-type: none"> • Disparador <i>borr_dep</i>. • Borrado del departamento: <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: DELETE FROM Departamento WHERE Cod_Emp = 'EP' AND Cod_Dep = 'DP';</pre>																																																												
Salida esperada	Ninguna de las tuplas reside en la base de datos.																																																												
Salida observada	<p>La esperada:</p> <table border="1"> <thead> <tr> <th>Cod_Emp</th> <th>CIF</th> <th>Nombre</th> <th>Direccion</th> </tr> </thead> <tbody> <tr> <td>CADO</td> <td>B31168479</td> <td>Cador</td> <td>Poligono Talluntxe Calle C, Números 15167772157 31.</td> </tr> <tr> <td>CIS</td> <td>B82738212</td> <td>Compañía Integral de Seguridad</td> <td>C/Castillo de Maya, 2 31015 Pamplona</td> </tr> <tr> <td>COMF</td> <td>B37392112</td> <td>Comiflo</td> <td>R. Ruben, 12 08004 Barcelona</td> </tr> <tr> <td>CONA</td> <td>B87268920</td> <td>Conasa</td> <td>C/Santa Engracia, 43 31014 Pamplona</td> </tr> <tr> <td>CONT</td> <td>B82517261</td> <td>Conatel</td> <td>C/Serapio Huici, 22 31017 Villava</td> </tr> <tr> <td>ELEC</td> <td>B53017290</td> <td>Electrolumen</td> <td>C/Mayor, 55 31018 Burlada</td> </tr> <tr> <td>ELKO</td> <td>B43114827</td> <td>Elkor</td> <td>C/Ezkaba, 5 31014 Pamplona</td> </tr> <tr> <td>GUPO</td> <td>B23132581</td> <td>Gupost</td> <td>C/Maria Viscarret, 46 31018 Artica</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>DNI</th> <th>Cod_Emp</th> <th>Cod_Dep</th> <th>Cod_Cargo</th> </tr> </thead> <tbody> <tr> <td>00000190A</td> <td>CONA</td> <td>CAU</td> <td>RP</td> </tr> <tr> <td colspan="4" style="text-align: center;">...</td> </tr> <tr> <td>00001000A</td> <td>CIS</td> <td>VA</td> <td>RP</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Cod_Cargo</th> <th>Descripcion</th> </tr> </thead> <tbody> <tr> <td>ANAL</td> <td>Analista</td> </tr> <tr> <td>GERE</td> <td>Gerente</td> </tr> <tr> <td>RP</td> <td>Responsable de personal</td> </tr> </tbody> </table>	Cod_Emp	CIF	Nombre	Direccion	CADO	B31168479	Cador	Poligono Talluntxe Calle C, Números 15167772157 31.	CIS	B82738212	Compañía Integral de Seguridad	C/Castillo de Maya, 2 31015 Pamplona	COMF	B37392112	Comiflo	R. Ruben, 12 08004 Barcelona	CONA	B87268920	Conasa	C/Santa Engracia, 43 31014 Pamplona	CONT	B82517261	Conatel	C/Serapio Huici, 22 31017 Villava	ELEC	B53017290	Electrolumen	C/Mayor, 55 31018 Burlada	ELKO	B43114827	Elkor	C/Ezkaba, 5 31014 Pamplona	GUPO	B23132581	Gupost	C/Maria Viscarret, 46 31018 Artica	DNI	Cod_Emp	Cod_Dep	Cod_Cargo	00000190A	CONA	CAU	RP	...				00001000A	CIS	VA	RP	Cod_Cargo	Descripcion	ANAL	Analista	GERE	Gerente	RP	Responsable de personal
Cod_Emp	CIF	Nombre	Direccion																																																										
CADO	B31168479	Cador	Poligono Talluntxe Calle C, Números 15167772157 31.																																																										
CIS	B82738212	Compañía Integral de Seguridad	C/Castillo de Maya, 2 31015 Pamplona																																																										
COMF	B37392112	Comiflo	R. Ruben, 12 08004 Barcelona																																																										
CONA	B87268920	Conasa	C/Santa Engracia, 43 31014 Pamplona																																																										
CONT	B82517261	Conatel	C/Serapio Huici, 22 31017 Villava																																																										
ELEC	B53017290	Electrolumen	C/Mayor, 55 31018 Burlada																																																										
ELKO	B43114827	Elkor	C/Ezkaba, 5 31014 Pamplona																																																										
GUPO	B23132581	Gupost	C/Maria Viscarret, 46 31018 Artica																																																										
DNI	Cod_Emp	Cod_Dep	Cod_Cargo																																																										
00000190A	CONA	CAU	RP																																																										
...																																																													
00001000A	CIS	VA	RP																																																										
Cod_Cargo	Descripcion																																																												
ANAL	Analista																																																												
GERE	Gerente																																																												
RP	Responsable de personal																																																												
Evaluación/ conclusión	El disparador <i>borr_dep</i> funciona correctamente.																																																												

*Caso de prueba para el disparador *borr_dep**



Caso de prueba	1.15 - Disparador <i>borr_contacto</i>																																																																																			
Objetivo	Comprobar que <i>borr_contacto</i> funciona correctamente, es decir, que todo contacto borrado no relacionado con más departamentos es eliminado de Personal y que todo cargo no despeñado por ningún contacto será eliminado de Cargo.																																																																																			
Actividades	Crear un contacto junto con un nuevo cargo, borrarlo y ver que la persona y el cargo han dejado de existir.																																																																																			
Técnicas y herramientas	Utilidad phpMyAdmin.																																																																																			
Datos de entrada	<ul style="list-style-type: none"> Registros de prueba: <pre> Ejecutar la(s) consulta(s) SQL en la base de datos bd: INSERT INTO Empresa VALUES ('EP', 'CIF', 'NomEmp', 'DirEmp'); INSERT INTO Departamento VALUES ('EP', 'DP', 'NomDep', 'MRC', 100, CURDATE(), NULL); INSERT INTO Personal VALUES ('99999999A', 'A1', 'A2', 'N', NULL); INSERT INTO Cargo VALUES ('C', 'Descripcion de prueba'); INSERT INTO Contacto VALUES ('99999999A', 'EP', 'DP', 'C'); </pre> <ul style="list-style-type: none"> Disparador <i>borr_contacto</i>. Borrado del contacto: <pre> Ejecutar la(s) consulta(s) SQL en la base de datos bd: DELETE FROM Contacto WHERE DNI = '99999999A' AND Cod_Emp = 'EP' AND Cod_Dep = 'DP' </pre>																																																																																			
Salida esperada	La persona no existe en Personal y el cargo en Cargo.																																																																																			
Salida observada	<p>La esperada:</p> <table border="1"> <thead> <tr> <th>DNI</th> <th>Apellido1</th> <th>Apellido2</th> <th>Nombre</th> <th>DNI_Ins</th> </tr> </thead> <tbody> <tr><td>0000185A</td><td>Ruiz</td><td>Aguilar</td><td>Clara</td><td>0000018A</td></tr> <tr><td>0000186A</td><td>Reula</td><td>Gómez</td><td>Esther</td><td>0000043A</td></tr> <tr><td>0000187A</td><td>Hernández</td><td>Estiarte</td><td>Xavier</td><td>0000018A</td></tr> <tr><td>0000188A</td><td>Borde</td><td>Membibre</td><td>Irene</td><td>0000043A</td></tr> <tr><td>0000189A</td><td>Cestau</td><td>Baraibar</td><td>Begoña</td><td>0000198A</td></tr> <tr><td>0000190A</td><td>Valencia</td><td>Valle</td><td>Juana María</td><td>0000018A</td></tr> <tr><td>0000191A</td><td>Vera</td><td>Gil</td><td>Raquel</td><td>0000043A</td></tr> <tr><td>0000192A</td><td>Acosta</td><td>Moreno</td><td>Jose Manuel</td><td>0000043A</td></tr> <tr><td>0000193A</td><td>Aguilera</td><td>Frias</td><td>Jose Manuel</td><td>0000028A</td></tr> <tr><td>0000194A</td><td>Abellán</td><td>Abellán</td><td>Celia</td><td>0000043A</td></tr> <tr><td>0000195A</td><td>Baeza</td><td>Gomis</td><td>Esther</td><td>0000028A</td></tr> <tr><td>0000196A</td><td>Calderón</td><td>Ayala</td><td>Maria Olvido</td><td>0000043A</td></tr> <tr><td>0000198A</td><td>Delicado</td><td>Davo</td><td>Gaspar</td><td>0000018A</td></tr> <tr><td>0000100A</td><td>Tirado</td><td>Jimenez</td><td>Eva María</td><td>0000018A</td></tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Cod_Cargo</th> <th>Descripcion</th> </tr> </thead> <tbody> <tr><td>ANAL</td><td>Analista</td></tr> <tr><td>GERE</td><td>Gerente</td></tr> <tr><td>RP</td><td>Responsable de personal</td></tr> </tbody> </table>	DNI	Apellido1	Apellido2	Nombre	DNI_Ins	0000185A	Ruiz	Aguilar	Clara	0000018A	0000186A	Reula	Gómez	Esther	0000043A	0000187A	Hernández	Estiarte	Xavier	0000018A	0000188A	Borde	Membibre	Irene	0000043A	0000189A	Cestau	Baraibar	Begoña	0000198A	0000190A	Valencia	Valle	Juana María	0000018A	0000191A	Vera	Gil	Raquel	0000043A	0000192A	Acosta	Moreno	Jose Manuel	0000043A	0000193A	Aguilera	Frias	Jose Manuel	0000028A	0000194A	Abellán	Abellán	Celia	0000043A	0000195A	Baeza	Gomis	Esther	0000028A	0000196A	Calderón	Ayala	Maria Olvido	0000043A	0000198A	Delicado	Davo	Gaspar	0000018A	0000100A	Tirado	Jimenez	Eva María	0000018A	Cod_Cargo	Descripcion	ANAL	Analista	GERE	Gerente	RP	Responsable de personal
DNI	Apellido1	Apellido2	Nombre	DNI_Ins																																																																																
0000185A	Ruiz	Aguilar	Clara	0000018A																																																																																
0000186A	Reula	Gómez	Esther	0000043A																																																																																
0000187A	Hernández	Estiarte	Xavier	0000018A																																																																																
0000188A	Borde	Membibre	Irene	0000043A																																																																																
0000189A	Cestau	Baraibar	Begoña	0000198A																																																																																
0000190A	Valencia	Valle	Juana María	0000018A																																																																																
0000191A	Vera	Gil	Raquel	0000043A																																																																																
0000192A	Acosta	Moreno	Jose Manuel	0000043A																																																																																
0000193A	Aguilera	Frias	Jose Manuel	0000028A																																																																																
0000194A	Abellán	Abellán	Celia	0000043A																																																																																
0000195A	Baeza	Gomis	Esther	0000028A																																																																																
0000196A	Calderón	Ayala	Maria Olvido	0000043A																																																																																
0000198A	Delicado	Davo	Gaspar	0000018A																																																																																
0000100A	Tirado	Jimenez	Eva María	0000018A																																																																																
Cod_Cargo	Descripcion																																																																																			
ANAL	Analista																																																																																			
GERE	Gerente																																																																																			
RP	Responsable de personal																																																																																			
Evaluación/ conclusión	El disparador <i>borr_contacto</i> funciona correctamente.																																																																																			

Caso de prueba para el disparador *borr_contacto*



Caso de prueba	1.16 - Disparador <i>borr_cuenta</i>										
Objetivo	Verificar que si borramos una cuenta de usuario cuyo tipo es el último existente, dicho tipo desaparecerá de la tabla T_Usuario.										
Actividades	Acceder a phpMyAdmin, borrar de la tabla de cuentas todas aquellas que correspondan a administradores y consultar la tabla T_Usuario para comprobar que el tipo 'ADMINISTRADOR' ha dejado de existir.										
Técnicas y herramientas	Utilidad phpMyAdmin.										
Datos de entrada	<ul style="list-style-type: none">Estado de la tabla T_Usuario: <table border="1"><thead><tr><th>Tipo</th><th>Descripcion</th></tr></thead><tbody><tr><td>ADMINISTRADOR</td><td>Administrador del sistema</td></tr><tr><td>AUTORIZADOR</td><td>Autorizador de departamentos</td></tr><tr><td>RP</td><td>Responsable de personal de departamento(s)</td></tr><tr><td>VIGILANTE</td><td>Vigilante del centro</td></tr></tbody></table>Disparador <i>borr_cuenta</i>.Borrado de todas las cuentas de tipo 'ADMINISTRADOR': <u>Ejecutar la(s) consulta(s) SQL en la base de datos bd:</u> <pre>DELETE FROM Cuenta WHERE Tipo = 'ADMINISTRADOR';</pre>	Tipo	Descripcion	ADMINISTRADOR	Administrador del sistema	AUTORIZADOR	Autorizador de departamentos	RP	Responsable de personal de departamento(s)	VIGILANTE	Vigilante del centro
Tipo	Descripcion										
ADMINISTRADOR	Administrador del sistema										
AUTORIZADOR	Autorizador de departamentos										
RP	Responsable de personal de departamento(s)										
VIGILANTE	Vigilante del centro										
Salida esperada	Como ya hemos comentado, el borrado de todas las cuentas pertenecientes a administradores hará que dicho tipo deje de existir en la base de datos.										
Salida observada	La esperada: <table border="1"><thead><tr><th>Tipo</th><th>Descripcion</th></tr></thead><tbody><tr><td>AUTORIZADOR</td><td>Autorizador de departamentos</td></tr><tr><td>RP</td><td>Responsable de personal de departamento(s)</td></tr><tr><td>VIGILANTE</td><td>Vigilante del centro</td></tr></tbody></table>	Tipo	Descripcion	AUTORIZADOR	Autorizador de departamentos	RP	Responsable de personal de departamento(s)	VIGILANTE	Vigilante del centro		
Tipo	Descripcion										
AUTORIZADOR	Autorizador de departamentos										
RP	Responsable de personal de departamento(s)										
VIGILANTE	Vigilante del centro										
Evaluación/ conclusión	El disparador <i>borr_cuenta</i> funciona correctamente.										

*Caso de prueba para el disparador *borr_cuenta**



Caso de prueba	1.17 - Disparador <i>ins_acc</i>																																																																																																																																				
Objetivo	Demostrar que <i>ins_acc</i> realiza correctamente su función, asignando a la tarjeta en préstamo el DNI de la persona que la recibe.																																																																																																																																				
Actividades	Crear una tarjeta, una persona y un acceso, comprobando que tras insertar este último la tarjeta adquiere su DNI.																																																																																																																																				
Técnicas y herramientas	Utilidad phpMyAdmin.																																																																																																																																				
Datos de entrada	<ul style="list-style-type: none"> Registros de prueba: <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: [?]</pre> <pre>INSERT INTO Personal VALUES ('99999999A', 'A1', 'A2', 'N', NULL); INSERT INTO Tarjeta VALUES (999, '2010-01-01 00:00:01', NULL, -1, '2010-01-01 00:00:01', NULL);</pre> Disparador <i>ins_acc</i>. Inserción del acceso: <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: [?]</pre> <pre>INSERT INTO Acceso VALUES ('99999999A', CURDATE(), NULL, 'Motivo', 999, '2010-01-01 00:00:01', NULL, '99999999A');</pre> 																																																																																																																																				
Salida esperada	El DNI de la tarjeta posee el valor del DNI de la persona.																																																																																																																																				
Salida observada	<p>La esperada:</p> <table border="1"> <thead> <tr> <th>Cod_Tarj</th> <th>F_Alta_T</th> <th>F_Baja_T</th> <th>Cod_Color</th> <th>F_Alta_C</th> <th>DNI</th> </tr> <tr> <th>Número de tarjeta</th> <th>Fecha y hora de inserción de la tarjeta</th> <th>Fecha y hora de baja de la tarjeta</th> <th>Color de la tarjeta</th> <th>Fecha y hora de inserción del color</th> <th>DNI de la persona que la tiene en su posesión</th> </tr> </thead> <tbody> <tr><td>31</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-16776961</td><td>2010-01-01 00:00:01</td><td>0000002A</td></tr> <tr><td>32</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-16776961</td><td>2010-01-01 00:00:01</td><td>0000003A</td></tr> <tr><td>33</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-16776961</td><td>2010-01-01 00:00:01</td><td>NULL</td></tr> <tr><td>34</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-16776961</td><td>2010-01-01 00:00:01</td><td>00000160A</td></tr> <tr><td>35</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-16776961</td><td>2010-01-01 00:00:01</td><td>NULL</td></tr> <tr><td>36</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-16776961</td><td>2010-01-01 00:00:01</td><td>NULL</td></tr> <tr><td>37</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-16776961</td><td>2010-01-01 00:00:01</td><td>NULL</td></tr> <tr><td>38</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-16776961</td><td>2010-01-01 00:00:01</td><td>00000097A</td></tr> <tr><td>39</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-16776961</td><td>2010-01-01 00:00:01</td><td>NULL</td></tr> <tr><td>40</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-16776961</td><td>2010-01-01 00:00:01</td><td>NULL</td></tr> <tr><td>41</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-16776961</td><td>2010-01-01 00:00:01</td><td>NULL</td></tr> <tr><td>42</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-16776961</td><td>2010-01-01 00:00:01</td><td>NULL</td></tr> <tr><td>43</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-16776961</td><td>2010-01-01 00:00:01</td><td>NULL</td></tr> <tr><td>44</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-16776961</td><td>2010-01-01 00:00:01</td><td>NULL</td></tr> <tr><td>45</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-16776961</td><td>2010-01-01 00:00:01</td><td>NULL</td></tr> <tr><td>46</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-16776961</td><td>2010-01-01 00:00:01</td><td>NULL</td></tr> <tr><td>47</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-16776961</td><td>2010-01-01 00:00:01</td><td>NULL</td></tr> <tr><td>48</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-16776961</td><td>2010-01-01 00:00:01</td><td>NULL</td></tr> <tr><td>49</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-16776961</td><td>2010-01-01 00:00:01</td><td>NULL</td></tr> <tr><td>999</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-1</td><td>2010-01-01 00:00:01</td><td>99999999A</td></tr> </tbody> </table>	Cod_Tarj	F_Alta_T	F_Baja_T	Cod_Color	F_Alta_C	DNI	Número de tarjeta	Fecha y hora de inserción de la tarjeta	Fecha y hora de baja de la tarjeta	Color de la tarjeta	Fecha y hora de inserción del color	DNI de la persona que la tiene en su posesión	31	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	0000002A	32	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	0000003A	33	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL	34	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	00000160A	35	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL	36	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL	37	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL	38	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	00000097A	39	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL	40	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL	41	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL	42	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL	43	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL	44	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL	45	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL	46	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL	47	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL	48	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL	49	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL	999	2010-01-01 00:00:01	NULL	-1	2010-01-01 00:00:01	99999999A
Cod_Tarj	F_Alta_T	F_Baja_T	Cod_Color	F_Alta_C	DNI																																																																																																																																
Número de tarjeta	Fecha y hora de inserción de la tarjeta	Fecha y hora de baja de la tarjeta	Color de la tarjeta	Fecha y hora de inserción del color	DNI de la persona que la tiene en su posesión																																																																																																																																
31	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	0000002A																																																																																																																																
32	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	0000003A																																																																																																																																
33	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL																																																																																																																																
34	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	00000160A																																																																																																																																
35	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL																																																																																																																																
36	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL																																																																																																																																
37	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL																																																																																																																																
38	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	00000097A																																																																																																																																
39	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL																																																																																																																																
40	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL																																																																																																																																
41	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL																																																																																																																																
42	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL																																																																																																																																
43	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL																																																																																																																																
44	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL																																																																																																																																
45	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL																																																																																																																																
46	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL																																																																																																																																
47	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL																																																																																																																																
48	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL																																																																																																																																
49	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	NULL																																																																																																																																
999	2010-01-01 00:00:01	NULL	-1	2010-01-01 00:00:01	99999999A																																																																																																																																
Evaluación/ conclusión	El disparador <i>ins_acc</i> funciona correctamente.																																																																																																																																				

Caso de prueba para el disparador *ins_acc*



Caso de prueba	1.18 - Disparador <i>actu_nota</i>																																																																																
Objetivo	Verificar que si disponemos de una nota descartada por todos los destinatarios y el autor la descarta, dicha nota dejará de permanecer en la base de datos.																																																																																
Actividades	Crear una nota cuya única persona que falte por descartarla sea su autor, descartarla y chequear que ya no existe.																																																																																
Técnicas y herramientas	Utilidad phpMyAdmin.																																																																																
Datos de entrada	<ul style="list-style-type: none"> Registros de prueba: <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: <input type="text"/> INSERT INTO Nota VALUES ('2010-01-01 00:00:01','00000000A','Tema','Texto',false); INSERT INTO Destinatario VALUES ('2010-01-01 00:00:01','00000000A','00000001A',true,true);</pre> <ul style="list-style-type: none"> Disparador <i>actu_nota</i>. Actualización de la nota a descartar por parte de su autor: <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: <input type="text"/> UPDATE Nota SET Descartada = true WHERE Fecha ='2010-01-01 00:00:01' AND DNI_A = '00000000A';</pre>																																																																																
Salida esperada	La nota descartada dejará de existir en la tabla Nota y con ello también en cualquier registro de Destinatario.																																																																																
Salida observada	<p>La esperada:</p> <table border="1"> <thead> <tr> <th>Fecha</th> <th>DNI_A</th> <th>Tema</th> <th>Texto</th> <th>Descartada</th> </tr> </thead> <tbody> <tr> <td>2008-12-04 11:30:00</td> <td>0000018A</td> <td>Informar de entrega de llaves</td> <td>Cuando el personal de cualquier expresa externa al...</td> <td>0</td> </tr> <tr> <td>2008-12-10 11:42:00</td> <td>0000043A</td> <td>Empresas externas</td> <td>Tamara Flores Cavido se va de vacaciones, en caso...</td> <td>0</td> </tr> <tr> <td>2008-12-11 18:38:00</td> <td>0000043A</td> <td>Alta de empresas</td> <td>Antes de dar de alta a una empresa, consúltalo con...</td> <td>0</td> </tr> <tr> <td>2008-12-15 16:00:00</td> <td>0000001A</td> <td>Cambio de turno</td> <td>Hola, Estaría interesado en cambiar el turno de pa...</td> <td>0</td> </tr> <tr> <td>2008-12-16 15:14:00</td> <td>0000003A</td> <td>Actualización de personal</td> <td>Hola, Hoy ha venido la persona con DNI 00000040A.</td> <td>0</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Fecha</th> <th>DNI_A</th> <th>DNI_D</th> <th>Leída</th> <th>Descartada</th> </tr> </thead> <tbody> <tr> <td>2008-12-04 11:30:00</td> <td>0000018A</td> <td>0000001A</td> <td>0</td> <td>0</td> </tr> <tr> <td>2008-12-04 11:30:00</td> <td>0000018A</td> <td>0000002A</td> <td>0</td> <td>0</td> </tr> <tr> <td>2008-12-04 11:30:00</td> <td>0000018A</td> <td>0000003A</td> <td>0</td> <td>0</td> </tr> <tr> <td>2008-12-10 11:42:00</td> <td>0000043A</td> <td>0000001A</td> <td>0</td> <td>0</td> </tr> <tr> <td>2008-12-10 11:42:00</td> <td>0000043A</td> <td>0000002A</td> <td>0</td> <td>0</td> </tr> <tr> <td>2008-12-10 11:42:00</td> <td>0000043A</td> <td>0000003A</td> <td>0</td> <td>0</td> </tr> <tr> <td>2008-12-11 18:38:00</td> <td>0000043A</td> <td>0000028A</td> <td>1</td> <td>1</td> </tr> <tr> <td>2008-12-15 16:00:00</td> <td>0000001A</td> <td>0000002A</td> <td>0</td> <td>0</td> </tr> <tr> <td>2008-12-16 15:14:00</td> <td>0000003A</td> <td>0000000A</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	Fecha	DNI_A	Tema	Texto	Descartada	2008-12-04 11:30:00	0000018A	Informar de entrega de llaves	Cuando el personal de cualquier expresa externa al...	0	2008-12-10 11:42:00	0000043A	Empresas externas	Tamara Flores Cavido se va de vacaciones, en caso...	0	2008-12-11 18:38:00	0000043A	Alta de empresas	Antes de dar de alta a una empresa, consúltalo con...	0	2008-12-15 16:00:00	0000001A	Cambio de turno	Hola, Estaría interesado en cambiar el turno de pa...	0	2008-12-16 15:14:00	0000003A	Actualización de personal	Hola, Hoy ha venido la persona con DNI 00000040A.	0	Fecha	DNI_A	DNI_D	Leída	Descartada	2008-12-04 11:30:00	0000018A	0000001A	0	0	2008-12-04 11:30:00	0000018A	0000002A	0	0	2008-12-04 11:30:00	0000018A	0000003A	0	0	2008-12-10 11:42:00	0000043A	0000001A	0	0	2008-12-10 11:42:00	0000043A	0000002A	0	0	2008-12-10 11:42:00	0000043A	0000003A	0	0	2008-12-11 18:38:00	0000043A	0000028A	1	1	2008-12-15 16:00:00	0000001A	0000002A	0	0	2008-12-16 15:14:00	0000003A	0000000A	0	0
Fecha	DNI_A	Tema	Texto	Descartada																																																																													
2008-12-04 11:30:00	0000018A	Informar de entrega de llaves	Cuando el personal de cualquier expresa externa al...	0																																																																													
2008-12-10 11:42:00	0000043A	Empresas externas	Tamara Flores Cavido se va de vacaciones, en caso...	0																																																																													
2008-12-11 18:38:00	0000043A	Alta de empresas	Antes de dar de alta a una empresa, consúltalo con...	0																																																																													
2008-12-15 16:00:00	0000001A	Cambio de turno	Hola, Estaría interesado en cambiar el turno de pa...	0																																																																													
2008-12-16 15:14:00	0000003A	Actualización de personal	Hola, Hoy ha venido la persona con DNI 00000040A.	0																																																																													
Fecha	DNI_A	DNI_D	Leída	Descartada																																																																													
2008-12-04 11:30:00	0000018A	0000001A	0	0																																																																													
2008-12-04 11:30:00	0000018A	0000002A	0	0																																																																													
2008-12-04 11:30:00	0000018A	0000003A	0	0																																																																													
2008-12-10 11:42:00	0000043A	0000001A	0	0																																																																													
2008-12-10 11:42:00	0000043A	0000002A	0	0																																																																													
2008-12-10 11:42:00	0000043A	0000003A	0	0																																																																													
2008-12-11 18:38:00	0000043A	0000028A	1	1																																																																													
2008-12-15 16:00:00	0000001A	0000002A	0	0																																																																													
2008-12-16 15:14:00	0000003A	0000000A	0	0																																																																													
Evaluación/ conclusión	El disparador <i>actu_nota</i> funciona correctamente.																																																																																

Caso de prueba para el disparador actu_nota



Caso de prueba	1.19 - Disparador <i>actu_dest</i>																																																																																
Objetivo	Probar que la política de borrado establecida y probada en el anterior caso de prueba también se cumple si el último en descartar una nota es un destinatario.																																																																																
Actividades	Redactar una nota en la que su autor la haya descartado pero no así su único destinatario. A continuación descartarla y verificar que ha dejado de formar parte de la tabla Nota.																																																																																
Técnicas y herramientas	Utilidad phpMyAdmin.																																																																																
Datos de entrada	<ul style="list-style-type: none"> Registros de prueba: <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: @</pre> <pre>INSERT INTO Nota VALUES ('2010-01-01 00:00:01', '00000000A', 'Tema', 'Texto', true); INSERT INTO Destinatario VALUES ('2010-01-01 00:00:01', '00000000A', '00000001A', false, false);</pre> Disparador <i>actu_dep</i>. Actualización de la nota a descartar por parte del último (y único en este caso) destinatario: <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: @</pre> <pre>UPDATE Destinatario SET Descartada = true WHERE Fecha = '2010-01-01 00:00:01' AND DNI_A = '00000000A' AND DNI_D = '00000001A'</pre> 																																																																																
Salida esperada	La misma que en el caso de prueba anterior.																																																																																
Salida observada	<p>La esperada:</p> <table border="1"> <thead> <tr> <th>Fecha</th> <th>DNI_A</th> <th>Tema</th> <th>Texto</th> <th>Descartada</th> </tr> </thead> <tbody> <tr> <td>2008-12-04 11:30:00</td> <td>00000018A</td> <td>Informar de entrega de llaves</td> <td>Cuando el personal de cualquier expresa externa al...</td> <td>0</td> </tr> <tr> <td>2008-12-10 11:42:00</td> <td>00000043A</td> <td>Empresas externas</td> <td>Tamara Flores Cavido se va de vacaciones, en caso...</td> <td>0</td> </tr> <tr> <td>2008-12-11 18:38:00</td> <td>00000043A</td> <td>Alta de empresas</td> <td>Antes de dar de alta a una empresa, consúltalo con...</td> <td>0</td> </tr> <tr> <td>2008-12-15 16:00:00</td> <td>00000001A</td> <td>Cambio de turno</td> <td>Hola, Estaría interesado en cambiar el turno de pa...</td> <td>0</td> </tr> <tr> <td>2008-12-16 15:14:00</td> <td>00000003A</td> <td>Actualización de personal</td> <td>Hola, Hoy ha venido la persona con DNI 00000040A...</td> <td>0</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Fecha</th> <th>DNI_A</th> <th>DNI_D</th> <th>Leída</th> <th>Descartada</th> </tr> </thead> <tbody> <tr> <td>2008-12-04 11:30:00</td> <td>00000018A</td> <td>00000001A</td> <td>0</td> <td>0</td> </tr> <tr> <td>2008-12-04 11:30:00</td> <td>00000018A</td> <td>00000002A</td> <td>0</td> <td>0</td> </tr> <tr> <td>2008-12-04 11:30:00</td> <td>00000018A</td> <td>00000003A</td> <td>0</td> <td>0</td> </tr> <tr> <td>2008-12-10 11:42:00</td> <td>00000043A</td> <td>00000001A</td> <td>0</td> <td>0</td> </tr> <tr> <td>2008-12-10 11:42:00</td> <td>00000043A</td> <td>00000002A</td> <td>0</td> <td>0</td> </tr> <tr> <td>2008-12-10 11:42:00</td> <td>00000043A</td> <td>00000003A</td> <td>0</td> <td>0</td> </tr> <tr> <td>2008-12-11 18:38:00</td> <td>00000043A</td> <td>00000028A</td> <td>1</td> <td>1</td> </tr> <tr> <td>2008-12-15 16:00:00</td> <td>00000001A</td> <td>00000002A</td> <td>0</td> <td>0</td> </tr> <tr> <td>2008-12-16 15:14:00</td> <td>00000003A</td> <td>00000000A</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	Fecha	DNI_A	Tema	Texto	Descartada	2008-12-04 11:30:00	00000018A	Informar de entrega de llaves	Cuando el personal de cualquier expresa externa al...	0	2008-12-10 11:42:00	00000043A	Empresas externas	Tamara Flores Cavido se va de vacaciones, en caso...	0	2008-12-11 18:38:00	00000043A	Alta de empresas	Antes de dar de alta a una empresa, consúltalo con...	0	2008-12-15 16:00:00	00000001A	Cambio de turno	Hola, Estaría interesado en cambiar el turno de pa...	0	2008-12-16 15:14:00	00000003A	Actualización de personal	Hola, Hoy ha venido la persona con DNI 00000040A...	0	Fecha	DNI_A	DNI_D	Leída	Descartada	2008-12-04 11:30:00	00000018A	00000001A	0	0	2008-12-04 11:30:00	00000018A	00000002A	0	0	2008-12-04 11:30:00	00000018A	00000003A	0	0	2008-12-10 11:42:00	00000043A	00000001A	0	0	2008-12-10 11:42:00	00000043A	00000002A	0	0	2008-12-10 11:42:00	00000043A	00000003A	0	0	2008-12-11 18:38:00	00000043A	00000028A	1	1	2008-12-15 16:00:00	00000001A	00000002A	0	0	2008-12-16 15:14:00	00000003A	00000000A	0	0
Fecha	DNI_A	Tema	Texto	Descartada																																																																													
2008-12-04 11:30:00	00000018A	Informar de entrega de llaves	Cuando el personal de cualquier expresa externa al...	0																																																																													
2008-12-10 11:42:00	00000043A	Empresas externas	Tamara Flores Cavido se va de vacaciones, en caso...	0																																																																													
2008-12-11 18:38:00	00000043A	Alta de empresas	Antes de dar de alta a una empresa, consúltalo con...	0																																																																													
2008-12-15 16:00:00	00000001A	Cambio de turno	Hola, Estaría interesado en cambiar el turno de pa...	0																																																																													
2008-12-16 15:14:00	00000003A	Actualización de personal	Hola, Hoy ha venido la persona con DNI 00000040A...	0																																																																													
Fecha	DNI_A	DNI_D	Leída	Descartada																																																																													
2008-12-04 11:30:00	00000018A	00000001A	0	0																																																																													
2008-12-04 11:30:00	00000018A	00000002A	0	0																																																																													
2008-12-04 11:30:00	00000018A	00000003A	0	0																																																																													
2008-12-10 11:42:00	00000043A	00000001A	0	0																																																																													
2008-12-10 11:42:00	00000043A	00000002A	0	0																																																																													
2008-12-10 11:42:00	00000043A	00000003A	0	0																																																																													
2008-12-11 18:38:00	00000043A	00000028A	1	1																																																																													
2008-12-15 16:00:00	00000001A	00000002A	0	0																																																																													
2008-12-16 15:14:00	00000003A	00000000A	0	0																																																																													
Evaluación/ conclusión	El disparador <i>actu_dest</i> funciona correctamente.																																																																																

Caso de prueba para el disparador actu_dest



5. Conclusiones

Llegados al final de la primera iteración contamos con todo el modelo de datos analizado, diseñado e implementado. Hemos conseguido obtener una base de datos robusta y eficiente, que permitirá dar soporte a toda la información que maneje el sistema, y a su vez con un buen grado de extensibilidad, de cara a que pueda ser modificada con gran facilidad en el futuro en caso de ser necesaria. Creemos que los artefactos y componentes obtenidos son satisfactorios, de una calidad aceptable, con que vemos con buenos ojos continuar con el desarrollo del trabajo pasando a desarrollar una nueva iteración.

Pamplona, a 12 de febrero de 2010.

Héctor Napal Basterrica
Responsable del proyecto



accessManager

Desarrollo de un sistema de control de accesos a un centro

Iteración C2: API para la base de datos

Fecha de entrega: 19 de febrero de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Definiciones, acrónimos y abreviaciones
- 1.4. Estructura del documento

2. Estado del proyecto

3. Análisis

- 3.1. Planificación de la iteración
- 3.2. Gestión de riesgos

4. Diseño

- 4.1. Visión global
- 4.2. Esquema OO

5. Codificación e integración

6. Pruebas

- 6.1. De unidad
- 6.1. De integridad

7. Evaluación de la iteración



1. Introducción

1.1. Propósito

El propósito con el que hemos elaborado este documento es el de poder documentar con un nivel de detalle aceptable el análisis, diseño e implementación de todo el modelo OO que dará soporte al modelo de datos desarrollado en la iteración anterior.

1.2. Audiencia

El cliente no precisa conocer esta parte del sistema, pero se hace imprescindible el conocimiento y estudio de este documento para poder llevar a cabo un correcto mantenimiento del sistema. Por lo tanto, en un principio este documento será de carácter interno, aunque será explicado personalmente al cliente en la fase de Transición y será incluido en la documentación entregada con el producto final para su posterior consulta.

1.3. Definiciones, acrónimos y abreviaciones

- OO Object Oriented Orientado a objetos

1.4. Estructura del documento

Comenzamos el documento haciendo una recapitulación de en qué punto del proyecto nos encontramos y qué es lo siguiente que debemos hacer.

Tras acordar desarrollar en este momento la API de manejo de la base de datos, realizamos una etapa de análisis en el que planificamos la iteración y, estudiamos los riesgos que conlleva.

La siguiente sección es la más importante de este artefacto, y es donde llevamos a cabo todo el diseño de la API a raíz del esquema relacional obtenido en la anterior iteración.

El diseño da paso a la codificación e integración de las clases, realizando las pruebas necesarias para certificar que funciona correctamente, y terminamos evaluando los resultados obtenidos en la iteración.

2. Estado del proyecto

Concluida la primera iteración ya disponemos de los cimientos sobre los que erigir el proyecto: la base de datos. La siguiente etapa nos llevará a desarrollar una API con la cual interactuar con la base de datos de una forma sencilla y desde un nivel de abstracción alto.



3. Análisis

3.1. Planificación de la iteración

La mayor parte de la programación de base de datos se realiza incrustando sentencias SQL directamente en el código de la aplicación, hecho que genera en muchas ocasiones serios problemas de mantenimiento, en especial cuando la aplicación crece en alcance y cuando varias aplicaciones emplean el esquema de la base de datos. La modificación de éste podría suponer tener que realizar cambios en todas las aplicaciones que utilicen la base de datos, y éste es un aspecto que debemos evitar.

Podemos dar solución a éste problema y mejorar en buen grado el posterior mantenimiento del sistema, así como la accesibilidad a los datos, creando una API de bases de datos utilizando la tecnología Java. El objetivo consistirá en centralizar todo el código SQL (DML y DQL) en un conjunto de clases que la aplicación utilizará en un futuro para proporcionar la funcionalidad deseada.

Planifiquemos la iteración:

PLAN DE ITERACIÓN C2: API PARA LA BASE DE DATOS	
ASPECTO	DESCRIPCIÓN
Objetivos	Desarrollo de una API inicial para la base de datos.
Artefactos de entrada	<ul style="list-style-type: none">• <i>Especificación inicial de los casos de uso</i>• <i>Planificación temporal y costes detallados del proyecto</i>• Artefactos de salida de la iteración C1:<ul style="list-style-type: none">○ Esquema relacional de la base de datos, junto con los disparadores y población.○ Código SQL derivado del punto anterior.• Documentación desarrollada en la presente memoria en la que se detalla la base de datos: <i>Iteración C1 – La base de datos</i>.
Artefactos modificados o creados	<ul style="list-style-type: none">• Diagrama de clases con todas las clases que conformarán la API inicial.• Codificación de todo el diagrama.• El presente documento.

Planificación de la iteración



Calendarización detallada	Requisitos + Pruebas	15/02/10	<ul style="list-style-type: none"> Planificación de la iteración y gestión de riesgos. Planificación de las pruebas. 																																																																																																																																																										
	Análisis + Pruebas	15/02/10	<ul style="list-style-type: none"> Análisis del modelo OO. Inicio de toda la documentación. Diseño de pruebas. 																																																																																																																																																										
	Diseño + Pruebas	16/02/10	<ul style="list-style-type: none"> Diseño del modelo OO. Diseño de pruebas. 																																																																																																																																																										
	Implementación + Pruebas	17/02/10 19/02/10	<ul style="list-style-type: none"> Codificación de todas las clases Java. Ejecución y evaluación de pruebas. Finalización de toda la documentación. 																																																																																																																																																										
<table border="1"> <thead> <tr> <th>Nombre de tarea</th> <th>Duración</th> <th>Comienzo</th> <th>Fin</th> <th>dom 14 feb</th> <th>mar 16 feb</th> <th>jue 18 feb</th> </tr> </thead> <tbody> <tr> <td>Estudio de oportunidad (inception)</td> <td>7 días</td> <td>jue 07/01/10</td> <td>vie 15/01/10</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>Elaboración</td> <td>16 días</td> <td>lun 18/01/10</td> <td>lun 08/02/10</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Construcción</td> <td>41 días</td> <td>mar 09/02/10</td> <td>mar 06/04/10</td> <td></td> <td></td> <td></td> </tr> <tr> <td> + Iteración C1: La base de datos</td> <td>4 días</td> <td>mar 09/02/10</td> <td>vie 12/02/10</td> <td></td> <td></td> <td></td> </tr> <tr> <td> - Iteración C2: API para la base de datos</td> <td>5 días</td> <td>lun 15/02/10</td> <td>vie 19/02/10</td> <td></td> <td></td> <td></td> </tr> <tr> <td> Análisis</td> <td>1 día</td> <td>lun 15/02/10</td> <td>lun 15/02/10</td> <td></td> <td></td> <td></td> </tr> <tr> <td> Diseño</td> <td>1 día</td> <td>mar 16/02/10</td> <td>mar 16/02/10</td> <td></td> <td></td> <td></td> </tr> <tr> <td> Implementación</td> <td>3 días</td> <td>mié 17/02/10</td> <td>vie 19/02/10</td> <td></td> <td></td> <td></td> </tr> <tr> <td> Pruebas</td> <td>5 días</td> <td>lun 15/02/10</td> <td>vie 19/02/10</td> <td></td> <td></td> <td></td> </tr> <tr> <td> + Iteración C3: Accesos</td> <td>5 días</td> <td>lun 22/02/10</td> <td>vie 26/02/10</td> <td></td> <td></td> <td></td> </tr> <tr> <td> + Iteración C4: Empresas y personal</td> <td>5 días</td> <td>lun 01/03/10</td> <td>vie 05/03/10</td> <td></td> <td></td> <td></td> </tr> <tr> <td> + Iteración C5: Tarjetas, tipos de acceso y cargos</td> <td>3 días</td> <td>lun 08/03/10</td> <td>mié 10/03/10</td> <td></td> <td></td> <td></td> </tr> <tr> <td> + Iteración C6: Zonas, llaves y llaveros</td> <td>5 días</td> <td>jue 11/03/10</td> <td>mié 17/03/10</td> <td></td> <td></td> <td></td> </tr> <tr> <td> + Iteración C7: Préstamos</td> <td>2 días</td> <td>jue 18/03/10</td> <td>vie 19/03/10</td> <td></td> <td></td> <td></td> </tr> <tr> <td> + Iteración C8: Cuentas</td> <td>2 días</td> <td>lun 22/03/10</td> <td>mar 23/03/10</td> <td></td> <td></td> <td></td> </tr> <tr> <td> + Iteración C9: Partes de seguridad</td> <td>3 días</td> <td>mié 24/03/10</td> <td>vie 26/03/10</td> <td></td> <td></td> <td></td> </tr> <tr> <td> + Iteración C10: Notas</td> <td>3 días</td> <td>lun 29/03/10</td> <td>mié 31/03/10</td> <td></td> <td></td> <td></td> </tr> <tr> <td> + Iteración C11: Base de datos</td> <td>2 días</td> <td>jue 01/04/10</td> <td>vie 02/04/10</td> <td></td> <td></td> <td></td> </tr> <tr> <td> + Iteración C12: Diseño, personalización y configuración</td> <td>2 días</td> <td>lun 05/04/10</td> <td>mar 06/04/10</td> <td></td> <td></td> <td></td> </tr> <tr> <td> Capacidad operacional</td> <td>0 días</td> <td>mar 06/04/10</td> <td>mar 06/04/10</td> <td></td> <td></td> <td></td> </tr> <tr> <td> + Transición</td> <td>8 días</td> <td>mié 07/04/10</td> <td>vie 16/04/10</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>				Nombre de tarea	Duración	Comienzo	Fin	dom 14 feb	mar 16 feb	jue 18 feb	Estudio de oportunidad (inception)	7 días	jue 07/01/10	vie 15/01/10	0	0	0	Elaboración	16 días	lun 18/01/10	lun 08/02/10				Construcción	41 días	mar 09/02/10	mar 06/04/10				+ Iteración C1: La base de datos	4 días	mar 09/02/10	vie 12/02/10				- Iteración C2: API para la base de datos	5 días	lun 15/02/10	vie 19/02/10				Análisis	1 día	lun 15/02/10	lun 15/02/10				Diseño	1 día	mar 16/02/10	mar 16/02/10				Implementación	3 días	mié 17/02/10	vie 19/02/10				Pruebas	5 días	lun 15/02/10	vie 19/02/10				+ Iteración C3: Accesos	5 días	lun 22/02/10	vie 26/02/10				+ Iteración C4: Empresas y personal	5 días	lun 01/03/10	vie 05/03/10				+ Iteración C5: Tarjetas, tipos de acceso y cargos	3 días	lun 08/03/10	mié 10/03/10				+ Iteración C6: Zonas, llaves y llaveros	5 días	jue 11/03/10	mié 17/03/10				+ Iteración C7: Préstamos	2 días	jue 18/03/10	vie 19/03/10				+ Iteración C8: Cuentas	2 días	lun 22/03/10	mar 23/03/10				+ Iteración C9: Partes de seguridad	3 días	mié 24/03/10	vie 26/03/10				+ Iteración C10: Notas	3 días	lun 29/03/10	mié 31/03/10				+ Iteración C11: Base de datos	2 días	jue 01/04/10	vie 02/04/10				+ Iteración C12: Diseño, personalización y configuración	2 días	lun 05/04/10	mar 06/04/10				Capacidad operacional	0 días	mar 06/04/10	mar 06/04/10				+ Transición	8 días	mié 07/04/10	vie 16/04/10			
Nombre de tarea	Duración	Comienzo	Fin	dom 14 feb	mar 16 feb	jue 18 feb																																																																																																																																																							
Estudio de oportunidad (inception)	7 días	jue 07/01/10	vie 15/01/10	0	0	0																																																																																																																																																							
Elaboración	16 días	lun 18/01/10	lun 08/02/10																																																																																																																																																										
Construcción	41 días	mar 09/02/10	mar 06/04/10																																																																																																																																																										
+ Iteración C1: La base de datos	4 días	mar 09/02/10	vie 12/02/10																																																																																																																																																										
- Iteración C2: API para la base de datos	5 días	lun 15/02/10	vie 19/02/10																																																																																																																																																										
Análisis	1 día	lun 15/02/10	lun 15/02/10																																																																																																																																																										
Diseño	1 día	mar 16/02/10	mar 16/02/10																																																																																																																																																										
Implementación	3 días	mié 17/02/10	vie 19/02/10																																																																																																																																																										
Pruebas	5 días	lun 15/02/10	vie 19/02/10																																																																																																																																																										
+ Iteración C3: Accesos	5 días	lun 22/02/10	vie 26/02/10																																																																																																																																																										
+ Iteración C4: Empresas y personal	5 días	lun 01/03/10	vie 05/03/10																																																																																																																																																										
+ Iteración C5: Tarjetas, tipos de acceso y cargos	3 días	lun 08/03/10	mié 10/03/10																																																																																																																																																										
+ Iteración C6: Zonas, llaves y llaveros	5 días	jue 11/03/10	mié 17/03/10																																																																																																																																																										
+ Iteración C7: Préstamos	2 días	jue 18/03/10	vie 19/03/10																																																																																																																																																										
+ Iteración C8: Cuentas	2 días	lun 22/03/10	mar 23/03/10																																																																																																																																																										
+ Iteración C9: Partes de seguridad	3 días	mié 24/03/10	vie 26/03/10																																																																																																																																																										
+ Iteración C10: Notas	3 días	lun 29/03/10	mié 31/03/10																																																																																																																																																										
+ Iteración C11: Base de datos	2 días	jue 01/04/10	vie 02/04/10																																																																																																																																																										
+ Iteración C12: Diseño, personalización y configuración	2 días	lun 05/04/10	mar 06/04/10																																																																																																																																																										
Capacidad operacional	0 días	mar 06/04/10	mar 06/04/10																																																																																																																																																										
+ Transición	8 días	mié 07/04/10	vie 16/04/10																																																																																																																																																										
Recursos	Temporales			9 días																																																																																																																																																									
	Humanos			1 persona																																																																																																																																																									
	Software	Sistema operativo		Microsoft Windows 7																																																																																																																																																									
		Ofimática		Microsoft Word 2010 Adobe Acrobat Professional 8																																																																																																																																																									
		Diseño		Visual Paradigm 4.2																																																																																																																																																									
Codificación		NetBeans 6.8																																																																																																																																																											
Hardware			1 PC																																																																																																																																																										

Planificación de la iteración



3.2. Gestión de riesgos

La gestión de riesgos de esta iteración no dista mucho de la realizada en la anterior.

IDENTIFICACIÓN DE RIESGOS		
TIPO	EXISTENCIA	DESCRIPCIÓN
Tecnología y herramientas	No se observan	<p>El diseño y codificación de los artefactos de salida serán realizados en un ordenador con el siguiente software:</p> <ul style="list-style-type: none"> • Visual Paradigm 4.2, para el diseño del diagrama de clases. • NetBeans 6.8, con el que codificar el anterior diagrama. • Microsoft Word 2010, como editor de textos con el que documentar toda la iteración. <p>Disponemos de todo el software instalado, con licencias válidas para los productos de la gama Microsoft y Visual Paradigm, teniendo NetBeans licencia CDDL.</p>
Requerimientos	Se observan	<p>El aspecto cambiante de los requerimientos puede suponer rehacer el diseño.</p> <p>La API a desarrollar es una consecuencia directa del diagrama relacional de la base de datos. Esto implica que un cambio en los requerimientos desencadene consecuencias sobre el diseño relacional, y éste a su vez sobre la implementación de la API.</p>
Estimación	Se observan	<p>Se calcula que sean necesarios nueve días para llevar a buen puerto esta iteración. En un principio, debería ser tiempo suficiente, aunque cabe la posibilidad de que éste haya podido ser infraestimado.</p>

Identificación de riesgos

ANÁLISIS DE RIESGOS			
TIPO	RIESGOS POSIBLES	PROBABILIDAD	EFECTOS
Estimación	Infraestimación del tiempo requerido, y por tanto de los recursos económicos humanos necesarios.	Moderada	Tolerables
Requerimientos	El aspecto cambiante de los requerimientos puede suponer rehacer el diseño.	Baja	Tolerables – Serios

Análisis de riesgos



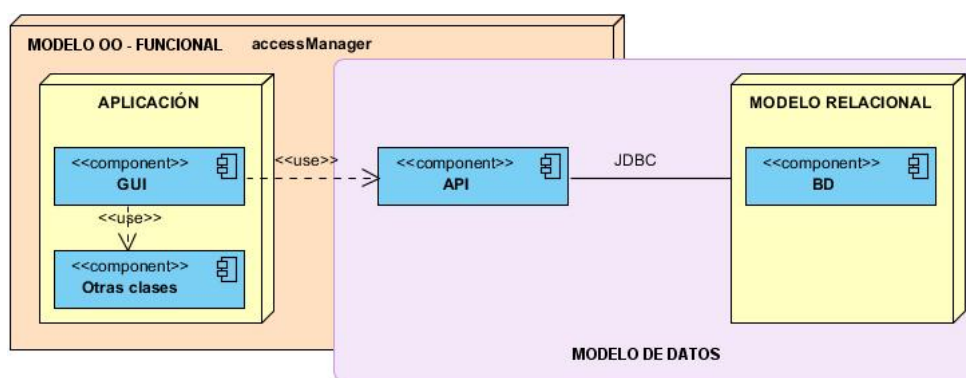
PLANES DE CONTINGENCIA	
TIPO	ESTRATEGIA
Estimación	La misma que en la iteración anterior: replanificar todo el proyecto de modo que pueda mantenerse la duración total estimada inicialmente o ésta se vea ampliada en la menor cuantía posible.
Requerimientos	Es un riesgo que debemos asumir. En caso de que éste se produzca deberemos analizar la situación actual y rediseñar y recodificar todo lo necesario.

Planes de contingencia

4. Diseño

4.1. Visión global

Como bien comentamos con anterioridad, nuestra idea es plasmar el mismo modelo de datos que estudiamos en la iteración anterior pero en esta ocasión empleando un modelo OO. De esta forma obtendremos dos modelos (relacional y OO) que relacionaremos y utilizaremos en la implementación de la aplicación. Este es el esquema que queremos obtener:



Arquitectura GUI – API – BD

La estrategia resulta sencilla y eficaz:

- El usuario interactuará con accessManager a través de su interfaz gráfica, compuesta por un JFrame principal y una serie de JDialogs que empleará para obtener y mostrar datos.
- Todo caso de uso se basará en las clases que forman la API para conseguir su cometido, siendo éstas quienes interactúen directamente con la base de datos.



Esta arquitectura presenta muchas ventajas, entre las que debemos mencionar:

- Simplificación del proceso de desarrollo (etapas de diseño y codificación) gracias a la abstracción que proporcionará la API que desarrollaremos.
- Separación entre programa principal y buena parte de la implementación del sistema, lo que facilita enormemente el futuro estudio y mantenimiento de éste. La modularidad crece así como su facilidad para ser comprendido y reprogramado, incluso para cualquier persona que no tuviera excesivos conocimientos en bases de datos.

4.2. Esquema OO

En las siguientes páginas desarrollamos la explicación de cómo hemos elaborado la API inicial. Ésta es bastante extensa, con que realizaremos un repaso global de la misma, describiendo el camino que hemos seguido y las características de las que hemos querido dotarla. Acto seguido pasaremos a estudiar una clase representativa y terminaremos haciendo una revisión rápida de cómo el modelo relacional da paso al OO. Esperemos que de este modo el lector pueda obtener una buena visión del conjunto sin que necesitemos entrar en excesivos detalles.



Para simplificar el esquema y facilitar su entendimiento, se ha omitido:

UtilidadesBD:

- Acceso
- Acceso_Sistema
- Anacion
- Cargo
- Color
- Edificio
- Empresa
- Gen_LLI
- Gen_LLI
- Nivel
- Parte_Seguridad
- Personal
- T_Acceso
- T_Acceso
- Tarjeta

UtilidadesCalendar:

- Uso por
- Anacion
- Edificio
- Empresa
- Residente

UtilFecha:

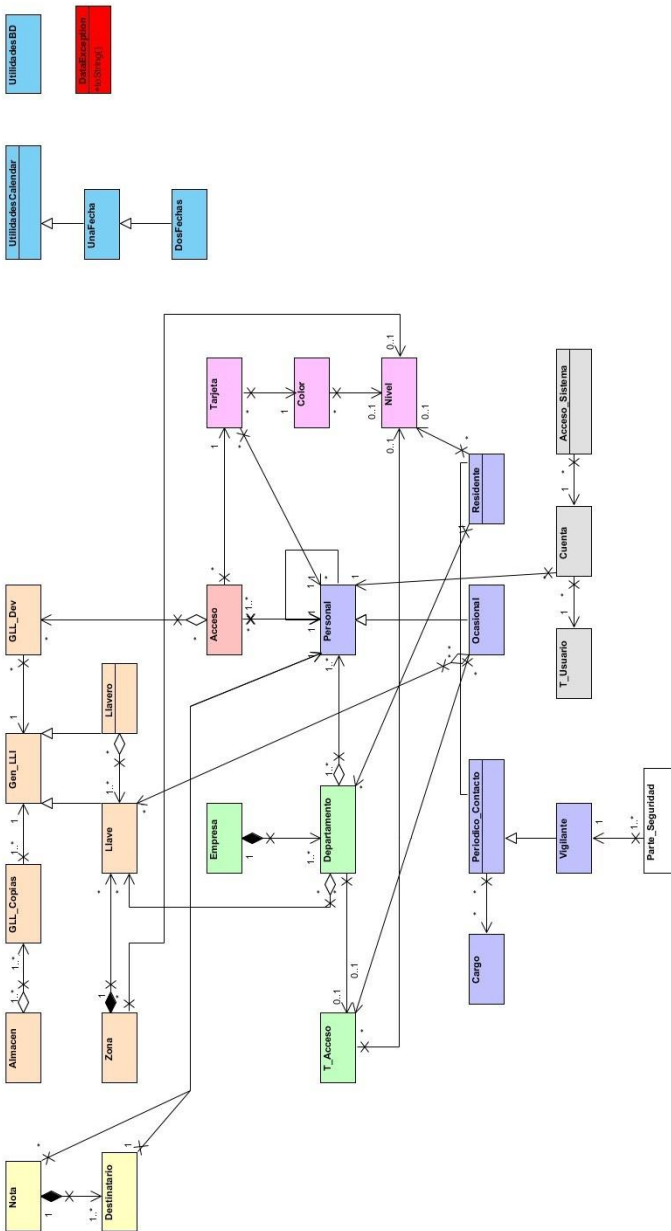
- Descendientes:
- Acceso_Sistema
- Manejo
- Manejo
- Parte_Seguridad
- Zona

Uso por:

- Permiso

DatFecha:

- Descendientes:
- Acceso
- Cargo
- Gen_LLI
- Nivel
- T_Acceso
- Tarjeta

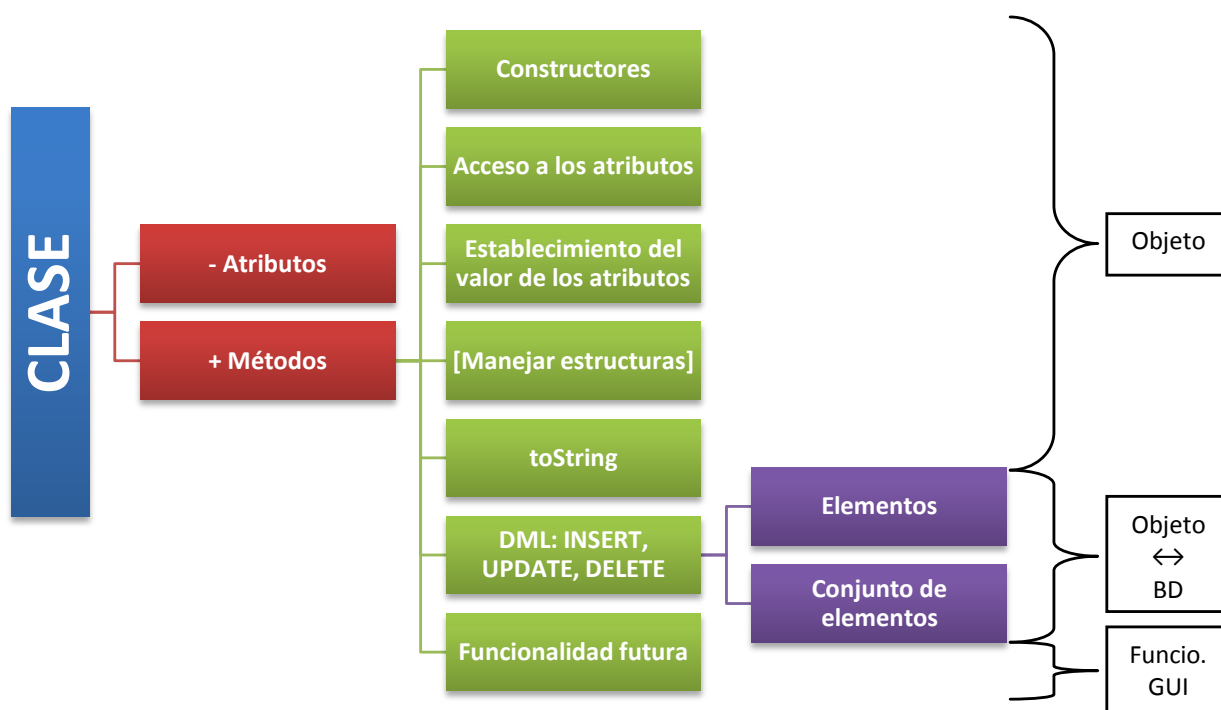


API inicial para la base de datos



Nota: Es importante atender a la parte derecha del esquema, en la que situamos objetos que proveerán todas las funcionalidades necesarias en el manejo de fechas, además de un cuadro de texto con todas las relaciones que no han sido dibujadas para simplificar en lo posible le diagrama.

Una vez disponemos del diseño inicial con todos los objetos que entrarán en juego, diseñamos las clases de modo que posean la siguiente estructura:



Comenzamos creando cada clase asignándole los atributos necesarios para que puedan representar la estructura de las tablas del modelo relacional. Éstos serán inicializados con constructores que les asignen valores a todos ellos o solo a los que no puedan ser nulos (obligados por el esquema relacional). En la codificación de éstos debemos tener muy en cuenta un aspecto: los constructores serán diseñados de tal modo que realicen siempre una comprobación previa de los datos que se les otorgan, de modo que si no cumplen con las características de los tipos de datos que componen las tablas lanzarán una *DataException* (a crear) al programa principal, evitando así conexiones y envíos de consultas a la base de datos que generen errores, y si las cumplen las instanciaciones de objetos serán exitosas.



Pongamos un ejemplo:

Si quisiésemos insertar en la base de datos un cargo con longitud de 'Codigo' mayor que 5 tendríamos:

- *Si utilizásemos constructores sin comprobación:
Crearíamos un objeto de tipo Cargo con el código erróneo, estableceríamos conexión con la base de datos y recibiríamos un error a la hora de intentar insertarlo.*
- *En caso de utilizar constructores con comprobación:
Obtendríamos el error a la hora de crear el objeto, de modo que evitaríamos conectar y enviar una consulta a la base de datos de manera innecesaria.*

A continuación diseñaremos operaciones que permitan acceder a los atributos de clase y asignarles valores, puesto que serán de carácter privado con vistas a conseguir una buena encapsulación, alta cohesión y mínimo acoplamiento de todas las clases pertenecientes a la API.

Una vez disponemos de esta base y en caso de ser preciso, se añaden las operaciones que puedan ser necesarias para manejar atributos de objetos compuestos, como puede ser el caso de ArrayLists, HashMaps,... y con operaciones nos referimos a las de inserción, modificación y borrado de elementos; con éstas bastará en un principio.

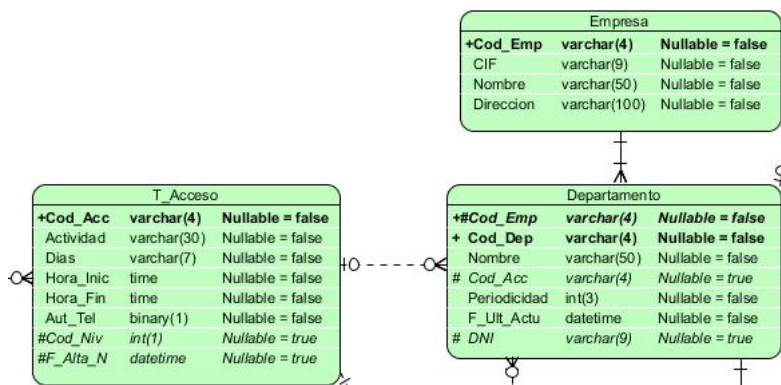
Tras sobrescribir el método *toString*, pasamos a relacionar el objeto con la base de datos, implementando las operaciones de manipulación de datos INSERT, UPDATE y DELETE en primer lugar para los objetos que puedan componer los atributos compuestos (p.e. departamentos) y a continuación las de los objetos compuestos (p.e. empresas). Es una implementación *bottom-up*, que implementa los entes más amplios a partir de las implementaciones de los entes atómicos.

Hasta aquí llegaría esta primera fase de la API. En iteraciones posteriores la ampliaremos en buen grado añadiéndole toda la funcionalidad que la aplicación (interfaz gráfica) pueda necesitar de ella.

Concretamos esta explicación en la clase Empresa para así poder disipar cualquier duda que pudiese haber quedado.

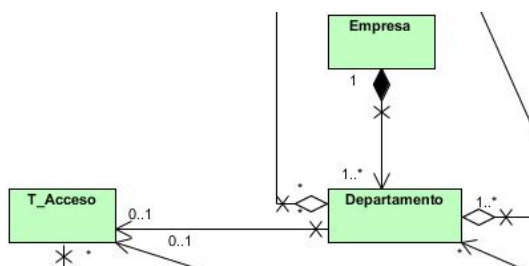


En el anterior artefacto diseñamos el siguiente esquema relacional para dar respuesta a todo el conjunto de datos relacionados con las empresas...



Esquema relacional parcial referente a las empresas

... derivando en este otro:



Esquema OO parcial referente a las empresas

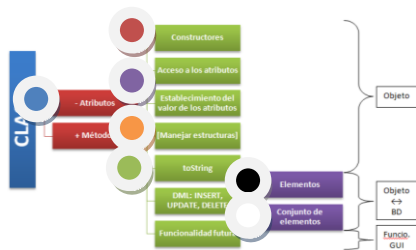
Una vez disponemos del esquema, desarrollamos cada una de las clases, obteniendo para Empresa la siguiente estructura:



Estructura del objeto Empresa



De aplicar los pasos descritos anteriormente, obtenemos la siguiente clase:



```

class Empresa {
    Cod_Emp : String
    CIF : String
    Nombre : String
    Direccion : String

    Empresa() : Empresa
    Empresa(Cod_Emp : String, CIF : String, Nombre : String, Direccion : String, Departamentos : HashMap<String, Departamento>) : Empresa
    getCodEmp() : String
    setCodEmp(Cod_Emp : String)
    getCIF() : String
    setCIF(CIF : String)
    getNombre() : String
    setNombre(Nombre : String)
    getDireccion() : String
    setDireccion(Direccion : String)
    getDepartamentos() : List<Departamento>
    setDepartamentos(Departamentos : HashMap<String, Departamento>)
    addDepartamento(Departamento : Departamento)
    updateDepartamento(Dep_old : Departamento, Dep_new : Departamento)
    removeDepartamento(Departamento : Departamento)
    getDepartamento(Cod_Dep : String) : Departamento
    toString() : String

    insertLlaveBD(Departamento : Departamento, Llave : Llave) : int
    updateLlaveBD(Departamento : Departamento, Llave_old : Llave, Llave_new : Llave) : int
    removeLlaveBD(Departamento : Departamento, Llave : Llave) : int
    insertLlavesBD(Departamento : Departamento) : int
    updateLlavesBD(Departamento : Departamento) : int
    removeLlavesBD(Departamento : Departamento) : int
    insertLlavesBD(Departamento : Departamento) : int

    insertPersonaCargoBD(Departamento : Departamento, Persona : Periodico_Contacto, Cargo : Cargo) : int
    updatePersonaCargoBD(Departamento : Departamento, Persona : Periodico_Contacto, Cargo_old : Cargo, Cargo_new : Cargo) : int
    removePersonaCargoBD(Departamento : Departamento, Persona : Periodico_Contacto, Cargo : Cargo) : int
    insertPersonaCargoBD(Departamento : Departamento, Persona : Periodico_Contacto) : int
    updatePersonaCargoBD(Departamento : Departamento, Persona : Periodico_Contacto, Cargo : Cargo) : int
    removePersonaCargoBD(Departamento : Departamento, Persona : Periodico_Contacto) : int
    insertPersonaCargoBD(Departamento : Departamento, Persona : Periodico_Contacto, Cargo : Cargo) : int

    insertPeriodicoContactoBD(Departamento : Departamento, Persona : Persona, Periodico_Contacto : Periodico_Contacto) : int
    updatePeriodicoContactoBD(Departamento : Departamento, Persona : Persona, Periodico_Contacto : Periodico_Contacto) : int
    removePeriodicoContactoBD(Departamento : Departamento, Persona : Persona, Periodico_Contacto : Periodico_Contacto) : int
    insertPeriodicoContactoBD(Departamento : Departamento, Persona : Persona) : int
    updatePeriodicoContactoBD(Departamento : Departamento, Persona : Persona) : int
    removePeriodicoContactoBD(Departamento : Departamento, Persona : Persona) : int

    insertVigilanteBD(Departamento : Departamento, Vigilante : Vigilante) : int
    updateVigilanteBD(Departamento : Departamento, Vigilante : Vigilante) : int
    removeVigilanteBD(Departamento : Departamento, Vigilante : Vigilante) : int
    insertVigilanteBD(Departamento : Departamento) : int
    updateVigilanteBD(Departamento : Departamento) : int
    removeVigilanteBD(Departamento : Departamento) : int

    insertPersonaBD(Departamento : Departamento, Persona : Persona) : int
    updatePersonaBD(Departamento : Departamento, Persona_old : Persona, Persona_new : Persona) : int
    removePersonaBD(Departamento : Departamento, Persona : Persona) : int
    insertPersonaBD(Departamento : Departamento) : int
    updatePersonaBD(Departamento : Departamento) : int
    removePersonaBD(Departamento : Departamento) : int

    getPersonaBD(Cod_Dep : String) : HashMap<String, Persona>
    insertPersonaBD(Departamento : Departamento) : int
    updatePersonaBD(Departamento : Departamento, Dep_old : Departamento, Dep_new : Departamento) : int
    removePersonaBD(Departamento : Departamento) : int
    getPersonaBD(Cod_Dep : String) : HashMap<String, Persona>
    insertPersonaBD(Departamento : Departamento) : int
    updatePersonaBD(Departamento : Departamento, Dep_old : Departamento, Dep_new : Departamento) : int
    removePersonaBD(Departamento : Departamento) : int

    insertDepartamentoBD(Departamento : Departamento) : int
    updateDepartamentoBD(Departamento : Departamento, Dep_old : Departamento, Dep_new : Departamento) : int
    removeDepartamentoBD(Departamento : Departamento) : int
    getDepartamentoBD(Cod_Dep : String) : Departamento
    insertDepartamentoBD(Departamento : Departamento) : int
    updateDepartamentoBD(Departamento : Departamento, Dep_old : Departamento, Dep_new : Departamento) : int
    removeDepartamentoBD(Departamento : Departamento) : int
    getDepartamentoBD(Cod_Dep : String) : Departamento

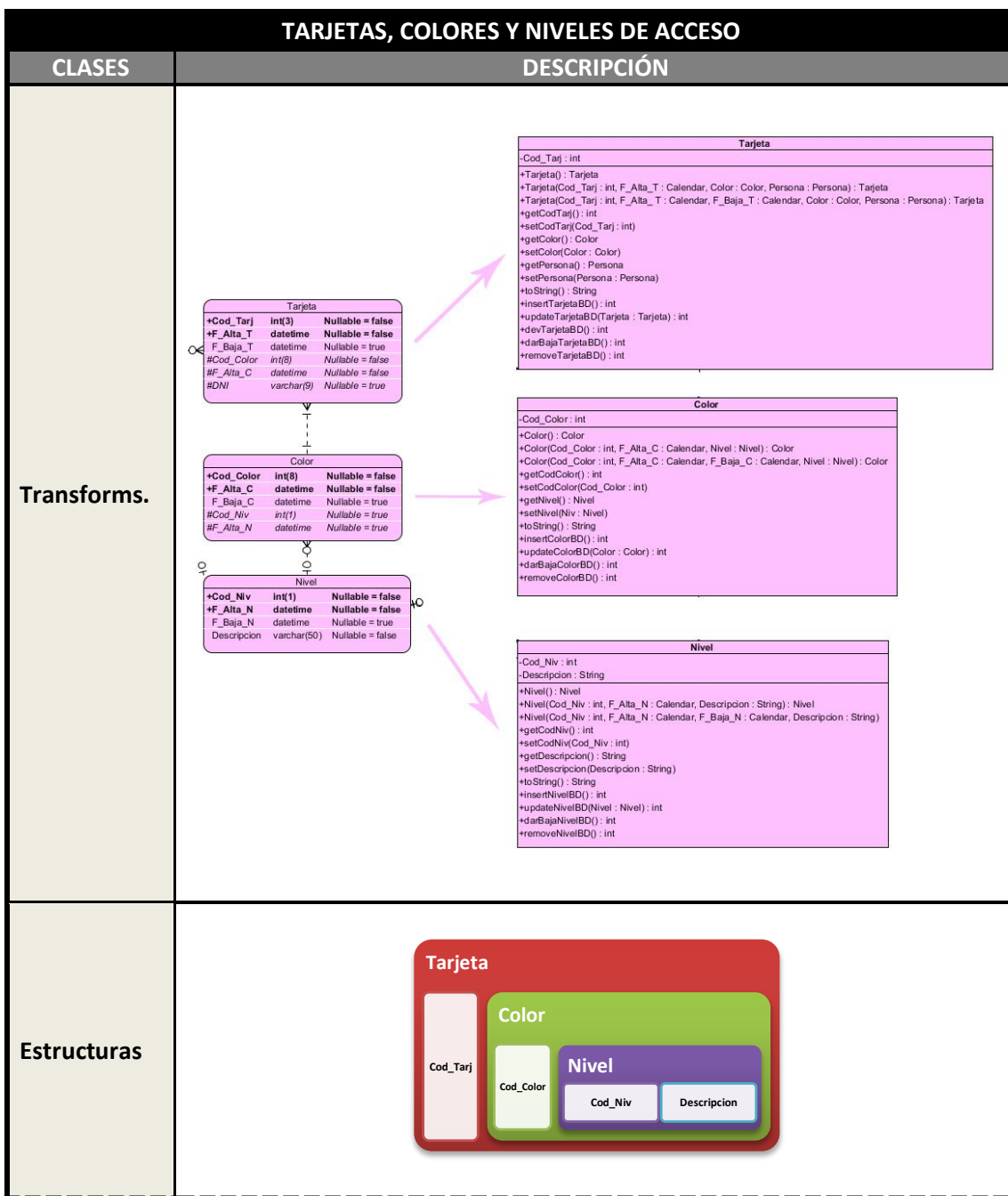
    insertDepartamentosBD(Departamentos : HashMap<String, Departamento>) : int
    updateDepartamentosBD(Departamentos : HashMap<String, Departamento>) : int
    removeDepartamentosBD(Departamentos : HashMap<String, Departamento>) : int
    getDepartamentosBD() : HashMap<String, Departamento>

    insertEmpresaBD() : int
    updateDatosBD(Empresa : Empresa) : int
    updateEmpresaBD(Empresa : Empresa) : int
    removeEmpresaBD() : int
    insertEmpresaBD(Empresa : Empresa) : int
    updateDatosBD(Empresa : Empresa) : int
    updateEmpresaBD(Empresa : Empresa) : int
    removeEmpresaBD(Empresa : Empresa) : int
    insertEmpresaBD(Empresa : Empresa) : int
  
```

Todo consiste en crear una estructura que represente el ente real y dotarla de todo lo necesario para su mantenimiento en el programa en sí y en la base de datos, desarrollando desde el detalle a lo general. En este caso desarrollamos primero operaciones para manejar cada departamento, persona y llave por separado, para después pasar a que se puedan insertar, modificar o borrar varios departamentos a la vez,... y llevamos estas mismas operaciones a la base de datos.



Llegó el momento de echar un vistazo rápido a cómo hemos transformado el resto del modelo relacional al orientado a objetos:

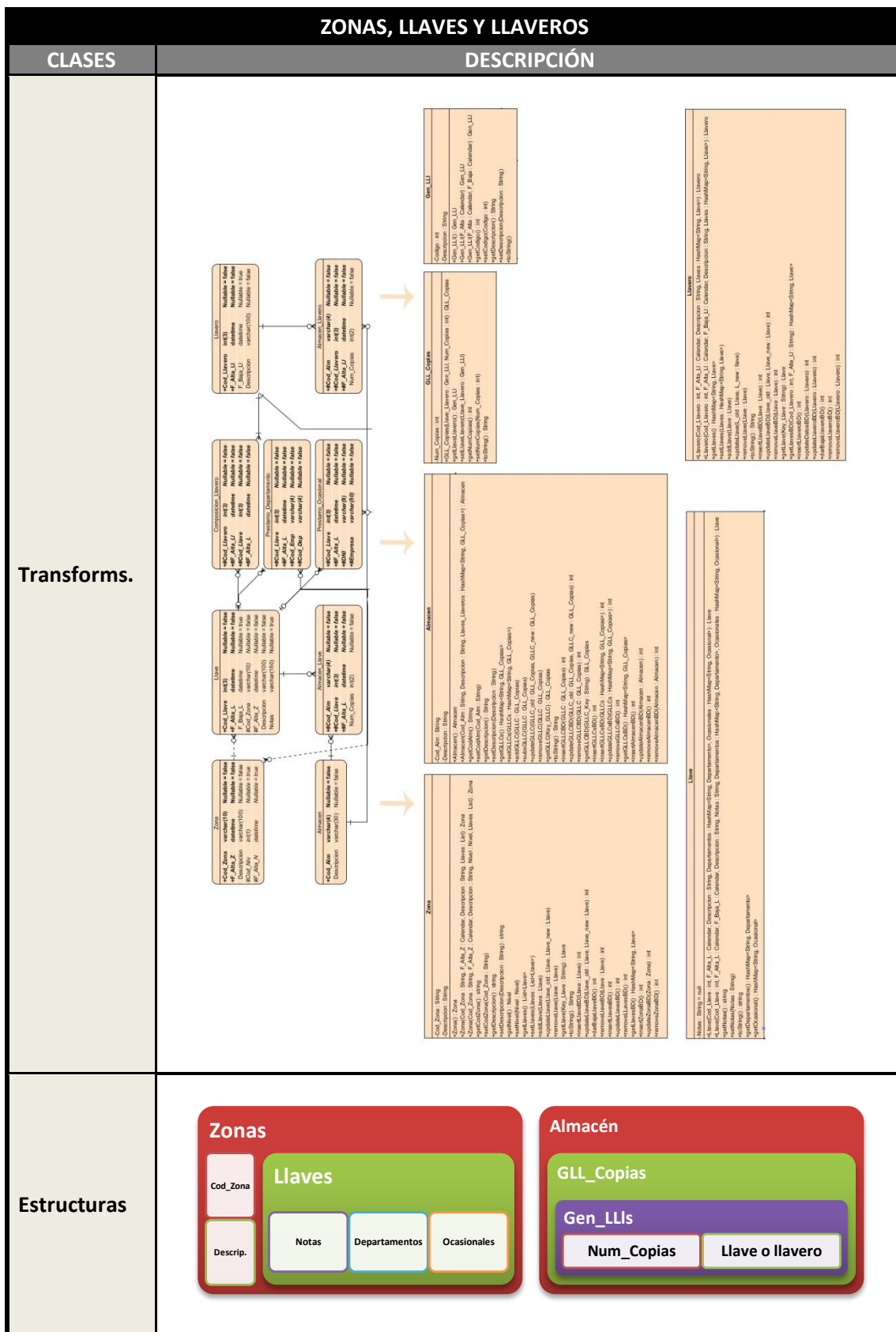


Tarjetas, colores y niveles de acceso



Tarjeta	Atribs.	<i>Cod_Tarj</i>	Código de tarjeta.
	Méts.	<i>Tarjeta getXXXX setXXXX toString insertTarjetaBD updateTarjetaBD devTarjetaBD darBajaTarjetaBD removeTarjetaBD</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. String con el estado del objeto. Insertar la tarjeta en la base de datos. Actualizar la tarjeta ... Establecer la tarjeta como devuelta ... Dar de baja la tarjeta ... Borrar la tarjeta ...
Color	Atribs.	<i>Cod_Color</i>	Código de color.
	Méts.	<i>Color getXXXX setXXXX toString insertColorBD updateColorBD darBajaColorBD removeColorBD</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. String con el estado del objeto. Insertar el color en la base de datos. Actualizar el color ... Dar de baja el color ... Borrar el color ...
Nivel	Atribs.	<i>Cod_Niv Descripción</i>	Código de nivel. Descripción del nivel.
	Méts.	<i>Nivel getXXXX setXXXX toString insertNivelBD updateNivelBD darBajaNivelBD removeNivelBD</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. String con el estado del objeto. Insertar el nivel en la base de datos. Actualizar el nivel ... Dar de baja el nivel ... Borrar el nivel ...

Tarjetas, colores y niveles de acceso



Zonas, llaves y llaveros



Zona	Atribs.	<i>Cod_Zona</i> <i>Descripción</i>	Código de zona. Descripción de la zona.
	Méts.	<i>Zona</i> <i>getXXXX</i> <i>setXXXX</i> <i>addLlave</i> <i>updateLlave</i> <i>removeLlave</i> <i>toString</i> <i>insertLlaveBD</i> <i>updateLlaveBD</i> <i>darBajaLlaveBD</i> <i>removeLlaveBD</i> <i>insertLlavesBD</i> <i>updateLlavesBD</i> <i>darBajaLlavesBD</i> <i>removeLlavesBD</i> <i>insertZonaBD</i> <i>updateZonaBD</i> <i>removeZonaBD</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. Añadir una llave a la zona. Actualizar una llave ... Borrar una llave ... String con el estado del objeto. Insertar una llave de la zona en la base de datos. Actualizar una llave ... Dar de baja una llave ... Borrar una llave ... Insertar las llaves de la zona en la base de datos. Actualizar las llaves ... Dar de baja las llaves ... Borrar las llaves ... Insertar una zona en la base de datos. Actualizar una zona ... Borrar una zona ...
Almacen	Atribs.	<i>Cod_Alm</i> <i>Descripcion</i>	Código de almacén. Descripción del almacén.
	Méts.	<i>Almacen</i> <i>getXXXX</i> <i>setXXXX</i> <i>addGLLC</i> <i>subsGLLC</i> <i>updateGLLC</i> <i>removeGLLC</i> <i>toString</i> <i>insertGLLCBD</i> <i>updateGLLCBD</i> <i>removeGLLCBD</i> <i>insertGLLCsBD</i> <i>updateGLLCsBD</i> <i>removeGLLCsBD</i> <i>insertAlmacenBD</i> <i>updateAlmacenBD</i> <i>removeAlmacenBD</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. Añadir copias de una llave al almacén. Sustraer copias ... Actualizar copias ... Borrar copias... String con el estado del objeto. Insert. copias de una llave en el alm. en la BD. Actualizar copias ... Borrar copias ... Insertar copias de las llaves de este alm. ... Actualizar copias ... Borrar copias ... Insertar almacén en la base de datos. Actualizar almacén ... Borrar almacén ...
GLL_Copias	Atribs.	<i>Num_Copias</i>	Número de copias
	Méts.	<i>GLL_Copias</i> <i>getXXXX</i> <i>setXXXX</i> <i>toString</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. String con el estado del objeto.

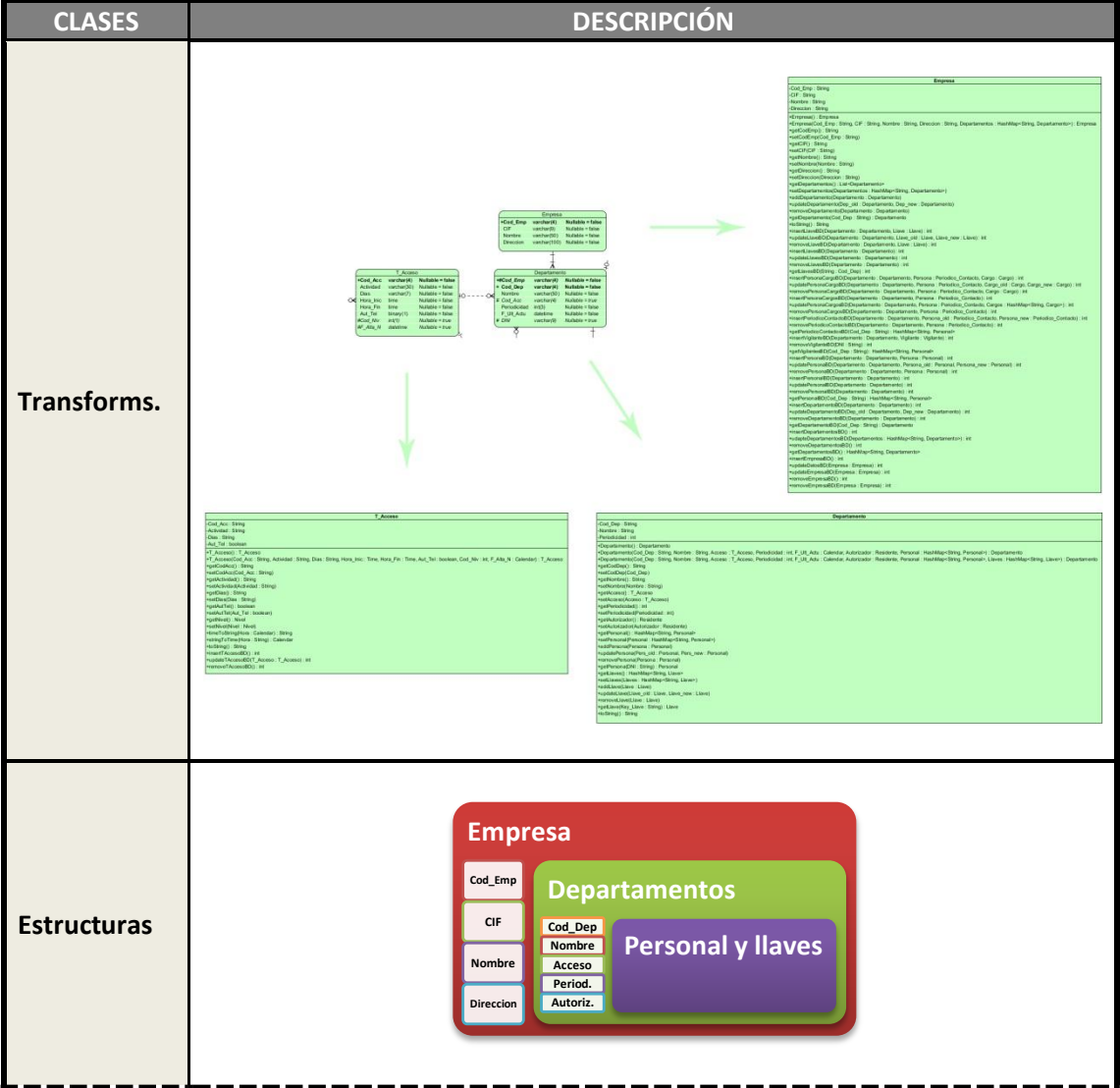
Zonas, llaves y llaveros



Gen_LLI	Atribs.	<i>Codigo</i> <i>Descripcion</i>	Código de la llave o llavero. Descripción de la llave o llavero.
	Méts.	<i>Gen_LLI</i> <i>getXXXX</i> <i>setXXXX</i> <i>toString</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. String con el estado del objeto.
Llave	Atribs.	<i>Notas</i>	Otros datos acerca de la llave.
	Méts.	<i>Llave</i> <i>getXXXX</i> <i>setXXXX</i> <i>toString</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. String con el estado del objeto.
Llavero	Atribs.	---	---
	Méts.	<i>Llavero</i> <i>getXXXX</i> <i>setXXXX</i> <i>addLlave</i> <i>updateLlave</i> <i>removeLlave</i> <i>toString</i> <i>insertLlaveBD</i> <i>updateLlaveBD</i> <i>removeLlaveBD</i> <i>insertLlaveroBD</i> <i>updateDatosBD</i> <i>updateLlaveroBD</i> <i>darBajaLlaveroBD</i> <i>removeLlaveroBD</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. Añadir una llave al llavero. Actualizar una llave ... Borrar una llave ... String con el estado del objeto. Insertar una llave en el llavero en la BD. Actualizar una llave ... Borrar una llave ... Insertar un llavero en la BD. Actualizar los datos de un llavero ... Actualizar un llavero ... Dar de baja un llavero ... Borrar un llavero ...

Zonas, llaves y llaveros

EMPRESAS, DEPARTAMENTOS Y TIPOS DE ACCESO



Empresas, departamentos y tipos de acceso



Empresa	Atribs.	<i>Cod_Emp</i> <i>CIF</i> <i>Nombre</i> <i>Direccion</i>	Código de empresa CIF. Nombre de empresa. Dirección de las oficinas de la empresa.
	Méts.	<i>Empresa</i> <i>getXXXX</i> <i>setXXXX</i> <i>addDepartamento</i> <i>updateDepartamento</i> <i>removeDepartamento</i> <i>toString</i> <i>insertLlaveBD</i> <i>updateLlaveBD</i> <i>removeLlaveBD</i> <i>insertLlavesBD</i> <i>updateLlavesBD</i> <i>removeLlavesBD</i> <i>insertPersonaCargoBD</i> <i>updatePersonaCargoBD</i> <i>removePersonaCargoBD</i> <i>insertPersonaCargosBD</i> <i>updatePersonaCargosBD</i> <i>removePersonaCargosBD</i> <i>insertPeriodicoContactoBD</i> <i>removePeriodicoContacBD</i> <i>insertVigilanteBD</i> <i>removeVigilanteBD</i> <i>insertPersonaBD</i> <i>updatePersonaBD</i> <i>removePersonaBD</i> <i>insertPersonalBD</i> <i>updatePersonalBD</i> <i>removePersonalBD</i> <i>insertDepartamentoBD</i> <i>updateDepartamentoBD</i> <i>removeDepartamentoBD</i> <i>insertDepartamentosBD</i> <i>updateDepartamentosBD</i> <i>removeDepartamentosBD</i> <i>insertEmpresaBD</i> <i>updateDatosBD</i> <i>updateEmpresaBD</i> <i>removeEmpresaBD</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. Añadir un departamento a la empresa. Modificar un departamento ... Borrar un departamento ... String con el estado del objeto. Insertar una llave como prestable en la BD. Modificar una llave ... Borrar una llave ... Insertar las llaves prestables en la BD. Modificar las llaves prestables ... Borrar las llaves prestables ... Insertar un cargo a una persona en la BD. Modificar un cargo a una persona ... Borrar un cargo a una persona ... Insertar los cargos de una persona en la BD. Modificar los cargos ... Borrar los cargos ... Insertar un periód./contac. en la BD. Borrar un periód./contac. ... Insertar un vigilante en la base de datos. Borrar un vigilante ... Insertar una persona en la base de datos. Modificar una persona ... Borrar una persona ... Insertar todo el personal en la BD. Modificar todo el personal ... Borrar todo el personal ... Insertar el departamento en la BD. Modificar el departamento ... Borrar el departamento ... Insertar todos los departamentos en la BD. Modificar todos los departamentos ... Borrar todos los departamentos ... Insertar la empresa en la base de datos. Modificar datos de la empresa ... Modificar la empresa ... Borrar la empresa ...

Empresas, departamentos y tipos de acceso



Departamento	Atribs.	<i>Cod_Dep</i> <i>Nombre</i> <i>Periodicidad</i>	Código de departamento. Nombre de departamento. Periodicidad de renovación del personal.
	Méts.	<i>Departamento</i> <i>getXXXX</i> <i>setXXXX</i> <i>addPersona</i> <i>updatePersona</i> <i>removePersona</i> <i>addLlave</i> <i>updateLlave</i> <i>removeLlave</i> <i>toString</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. Añadir una persona al departamento. Modificar una persona ... Borrar una persona ... Añadir una llave como prestable. Modificar una llave ... Borrar una llave ... String con el estado del objeto.
T_Acceso	Atribs.	<i>Cod_Acc</i> <i>Actividad</i> <i>Dias</i> <i>Aut_Tel</i>	Código de acceso. Actividad. Días de acceso. Denota si es necesaria autoriz. telefónica.
	Méts.	<i>T_Acceso</i> <i>getXXXX</i> <i>setXXXX</i> <i>timeToString</i> <i>stringToTime</i> <i>toString</i> <i>insertTAccesoBD</i> <i>updateTAccesoBD</i> <i>removeTAccesoBD</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. Pasar de un objeto Time a otro String. Pasar de un objeto String a otro Time. String con el estado del objeto. Insertar el tipo de acceso en la base de datos. Modificar el tipo de acceso ... Borrar el tipo de acceso ...

Empresas, departamentos y tipos de acceso



PERSONAL	
CLASES	DESCRIPCIÓN
Transforms.	<pre> classDiagram class Personal { DNI String Apellido1 String Apellido2 String Nombre String Insert Boolean ... } class Cargo { ... } class Periodico_Contacto { ... } class Ocasional { ... } Personal "1" -- "*" Cargo Personal "1" -- "*" Periodico_Contacto Personal "1" -- "*" Ocasional </pre>
Estructuras	<div style="display: flex; justify-content: space-between;"> <div style="border: 1px solid black; padding: 5px; background-color: #f0f0f0;"> <p>Personal</p> <ul style="list-style-type: none"> DNI Apellido1 Apellido2 Nombre Insert. <p style="text-align: center; background-color: #90ee90; padding: 5px;">Teléfonos e E-Mails</p> </div> <div style="border: 1px solid black; padding: 5px; background-color: #f0f0f0;"> <p>Periodico_Contacto</p> <p>Cargos</p> </div> <div style="border: 1px solid black; padding: 5px; background-color: #f0f0f0;"> <p>Ocasional</p> <ul style="list-style-type: none"> Emp Acc. F_Ca Not <p style="text-align: center; background-color: #90ee90; padding: 5px;">Llaves</p> </div> </div>

Personal



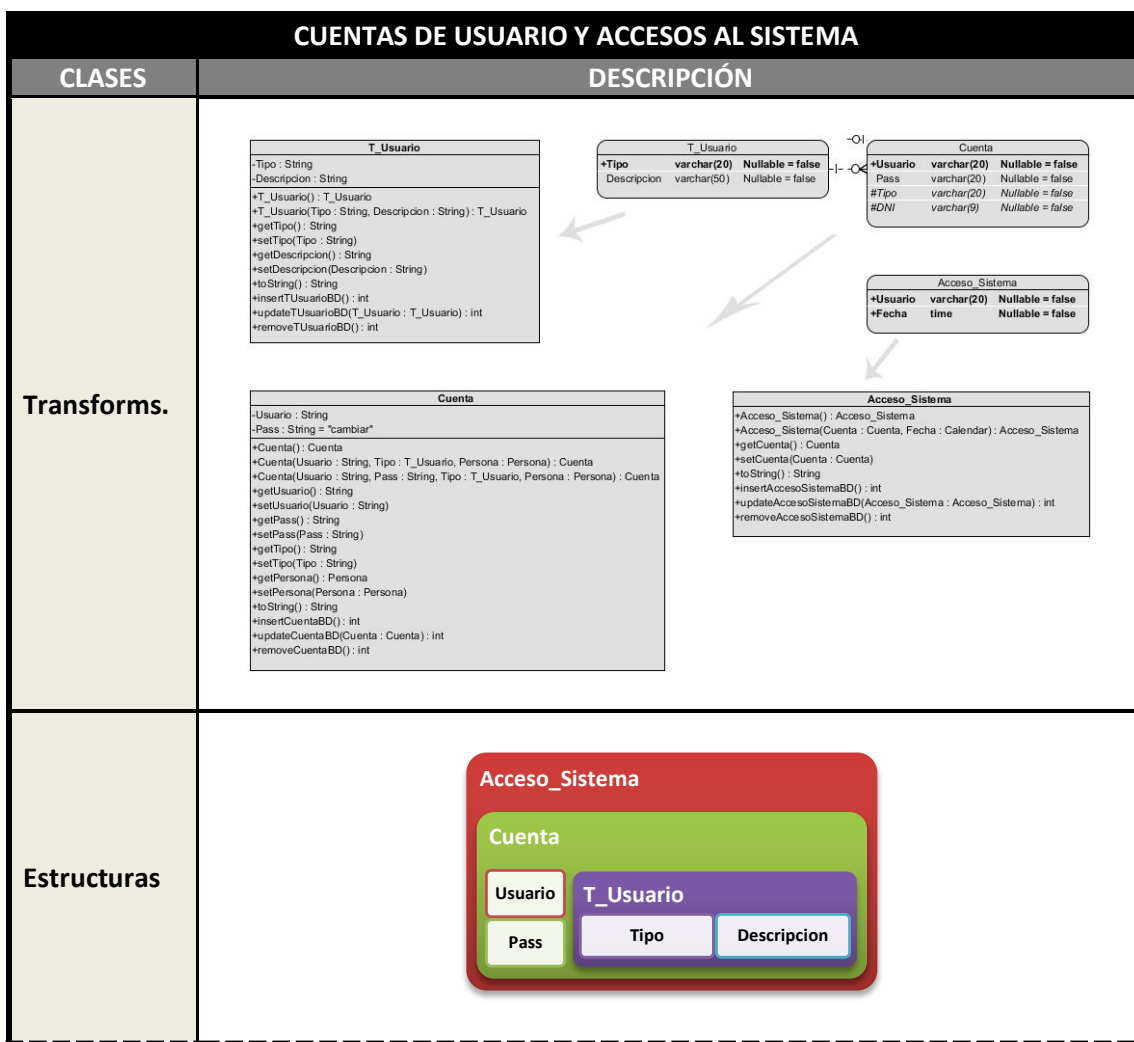
Personal	Atribs.	<i>DNI</i> <i>Apellido1</i> <i>Apellido2</i> <i>Nombre</i> <i>Telefonos</i> <i>Emails</i>	DNI. Primer apellido. Segundo apellido. Nombre. Números de teléfono. Direcciones de correo electrónico.
	Méts.	<i>Personal</i> <i>getXXXX</i> <i>setXXXX</i> <i>addTelefono</i> <i>updateTelefono</i> <i>removeTelefono</i> <i>addEmail</i> <i>updateEmail</i> <i>removeEmail</i> <i>toString()</i> <i>insertTelefonoBD</i> <i>updateTelefonoBD</i> <i>removeTelefonoBD</i> <i>insertTelefonosBD</i> <i>updateTelefonosBD</i> <i>removeTelefonosBD</i> <i>insertEmailBD</i> <i>updateEmailBD</i> <i>removeEmailBD</i> <i>insertEmailsBD</i> <i>updateEmailsBD</i> <i>removeEmailsBD</i> <i>insertPersonalBD</i> <i>updatePersonalBD</i> <i>removePersonalBD</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. Añadir número de teléfono. Modificar ... Borrar número ... Añadir dirección de correo electrónico. Modificar dirección ... Borrar dirección ... String con el estado del objeto. Insertar número de teléfono en la BD. Modificar número ... Borrar número ... Insertar los números de teléfono en la BD. Modificar los números ... Borrar los número ... Insertar dirección de E-Mail en la BD. Modificar dirección ... Borrar dirección ... Insertar las direcciones de E-Mail en la BD. Modificar las direcciones ... Borrar las direcciones ... Insertar persona en la base de datos. Modificar persona ... Borrar persona ...
Periodico _Contacto	Atribs.	---	---
	Méts.	<i>Periodico_Contacto</i> <i>getXXXX</i> <i>setXXXX</i> <i>addCargo</i> <i>updateCargo</i> <i>removeCargo</i> <i>toString</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. Añadir un cargo al contacto. Modificar un cargo ... Borrar un cargo ... String con el estado del objeto.
Cargo	Atribs.	<i>Cod_Cargo</i> <i>Descripcion</i>	Código de cargo. Descripción del cargo.
	Méts.	<i>Cargo</i> <i>getXXXX</i> <i>setXXXX</i> <i>toString</i> <i>insertcargoBD</i> <i>updateCargoBD</i> <i>removeCargoBD</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. String con el estado del objeto. Insertar un cargo en la base de datos. Modificar un cargo ... Borrar un cargo ...

Personal



Vigilante	Atribs.	<i>Cod_Vig</i>	Código de vigilante.
	Méts.	<i>Vigilante getXXXX setXXXX toString</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. String con el estado del objeto.
Ocasional	Atribs.	<i>Empresa Notas</i>	Empresa para la que trabaja. Otros datos de interés.
	Méts.	<i>Ocasional getXXXX setXXXX addLlave updateLlave removeLlave toString insertLlaveBD updateLlaveBD removeLlaveBD insertLlavesBD updateLlavesBD removeLlavesBD insertOcasionalBD updateOcasionalBD removeOcasionalBD</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. Añadir una llave a las prestables. Modificar una llave ... Borrar una llave ... String con el estado del objeto. Insertar una llave a las prestables en la BD. Modificar una llave ... Borrar una llave ... Insertar las llaves prestables en la BD. Modificar las llaves ... Borrar las llaves ... Insertar la persona ocasional en la BD. Modificar la persona ocasional ... Borrar la persona ocasional ...
Residente	Atribs.	---	---
	Méts.	<i>Residente getXXXX setXXXX toString insertResidenteBD updateResidenteBD removeResidenteBD</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. String con el estado del objeto. Insertar la persona residente en la BD. Modificar la persona residente ... Borrar la persona residente ...

Personal

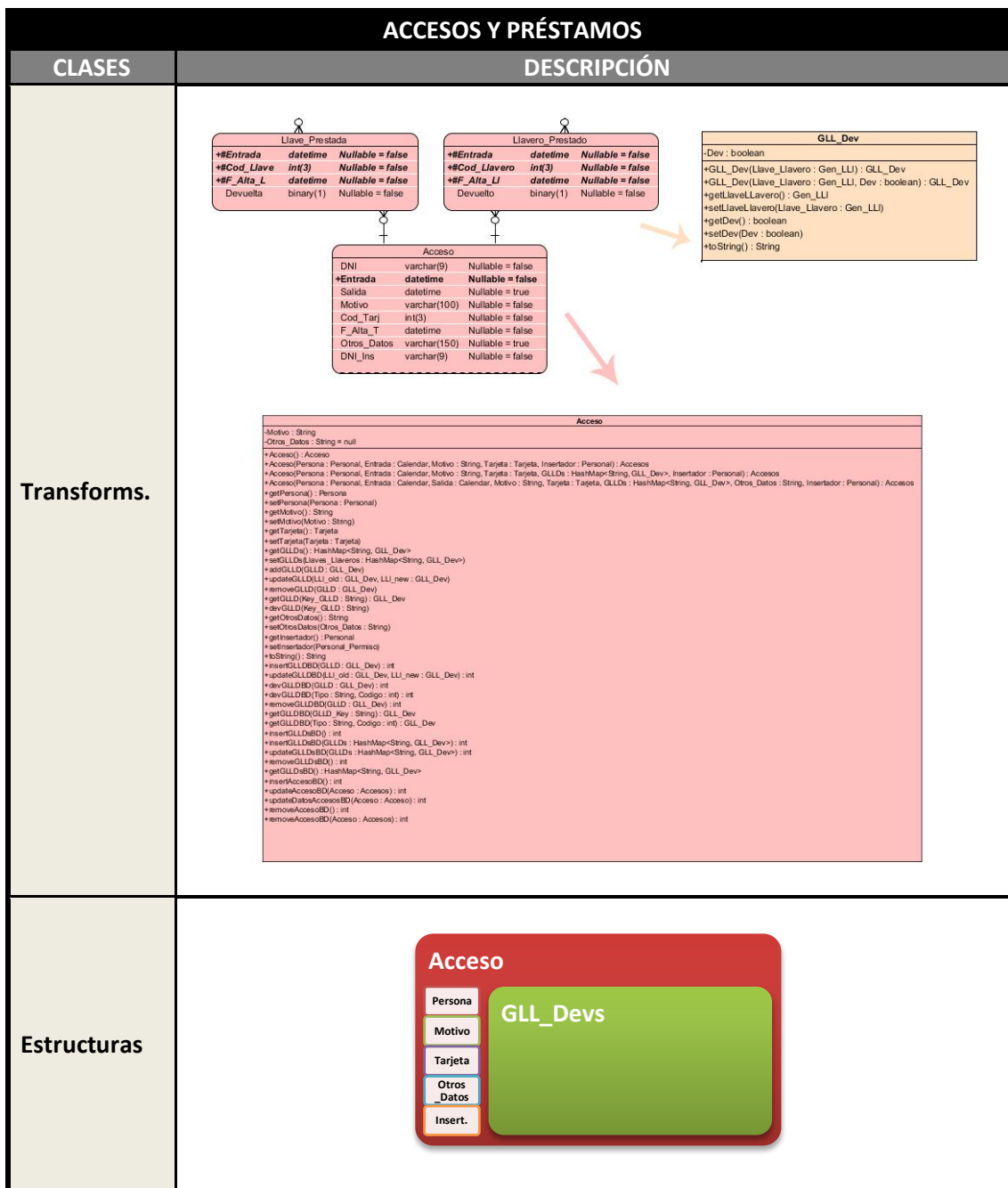


Cuentas de usuario y accesos al sistema



Acceso _Sistema	Atribs.	---	---
	Méts.	<i>Acceso_Sistema</i> <i>getXXXX</i> <i>setXXXX</i> <i>toString</i> <i>insertAccesoSistemaBD</i> <i>updateAccesoSistemaBD</i> <i>removeAccesoSistemaBD</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. String con el estado del objeto. Insertar un acceso en el sistema en la BD. Modificar un acceso en el sistema ... Borrar un acceso en el sistema...
Cuenta	Atribs.	<i>Usuario</i> <i>Pass</i>	Nombre de usuario. Contraseña
	Méts.	<i>Cuenta</i> <i>getXXXX</i> <i>setXXXX</i> <i>toString</i> <i>insertCuentaBD</i> <i>updateCuentaBD</i> <i>removeCuentaBD</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. String con el estado del objeto. Insertar la cuenta en la base de datos. Actualizar la cuenta ... Borrar la cuenta ...
T_Usuario	Atribs.	<i>Cod_Niv</i> <i>Descripción</i>	Código de nivel. Descripción del nivel.
	Méts.	<i>T_Usuario</i> <i>getXXXX</i> <i>setXXXX</i> <i>toString</i> <i>insertTUsuarioBD</i> <i>updateTUsuarioBD</i> <i>removeTUsuarioBD</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. String con el estado del objeto. Insertar el tipo de usuario en la base de datos. Modificar el tipo de usuario ... Dar de baja el tipo de usuario ... Borrar el tipo de usuario ...

Cuentas de usuario y accesos al sistema



Accesos y préstamos



Acceso	Atribs.	<i>Motivo</i> <i>Otros datos</i>	Motivo del acceso. Otros datos de interés
	Méts.	<i>Acceso</i> <i>getXXXX</i> <i>setXXXX</i> <i>addGLLD</i> <i>updateGLLD</i> <i>removeGLLD</i> <i>devGLLD</i> <i>toString</i> <i>insertGLLDBD</i> <i>updateGLLDBD</i> <i>devGLLDBD</i> <i>removeGLLDBD</i> <i>insertGLLDsBD</i> <i>updateGLLDsBD</i> <i>removeGLLDsBD</i> <i>insertAccesoBD</i> <i>updateAccesoBD</i> <i>updateDatosAccesoBD</i> <i>removeAccesoGLLDBD</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. Añadir un préstamo al acceso. Modificar un préstamo... Devolver un préstamo ... Borrar un préstamo... String con el estado del objeto. Insertar un préstamo al acceso en la BD. Modificar un préstamo ... Devolver un préstamo ... Borrar un préstamo ... Insertar los préstamos del acceso en la BD. Modificar los préstamos ... Borrar los préstamos ... Insertar el acceso en la base de datos. Modificar el acceso ... Modificar los datos del acceso ... Borrar el acceso ...
GLL_Dev	Atribs.	<i>Dev</i>	Denota si la llave/llavero ha sido devuelta.
	Méts.	<i>GLL_Dev</i> <i>getXXXX</i> <i>setXXXX</i> <i>toString</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. String con el estado del objeto.

Accesos y préstamos



NOTAS	
CLASES	DESCRIPCIÓN
Transforms.	<div style="display: flex; justify-content: space-between;"> <div style="width: 60%;"> <pre> Nota -Tema : String -Texto : String -Descartada : boolean = False +Nota() : Nota +Nota(Fecha : Calendar, Redactor : Persona, Tema : String, Texto : String, Destinatarios : HashMap<String, Destinatario>) : Nota +getRedactor() : Persona +setRedactor(Redactor : Persona) +getTema() : String +setTema(Tema : String) +getText() : String +setTexto(Texto : String) +getDescartada() : boolean +setDescartada(Descartada : boolean) +getDestinatarios() : HashMap<String, Destinatario> +setDestinatarios(Destinatarios : HashMap<String, Destinatario>) +addDestinatario(Destinatario : Destinatario) +updateDestinatario(Dest_old : Destinatario, Dest_new : Destinatario) +removeDestinatario(Destinatario : Destinatario) +getDestinatario(DNI : String) : Destinatario +toString() +insertDestinatarioBD(Destinatario : Destinatario) : int +updateDestinatarioBD(Dest_old : Destinatario, Dest_new : Destinatario) : int +removeDestinatarioBD(Destinatario : Destinatario) : int +getDestinatarioBD(DNI : String) : Destinatario +insertDestinatariosBD(Destinatarios : HashMap<String, Destinatario>) : int +updateDestinatariosBD(Destinatarios : HashMap<String, Destinatario>) : int +removeDestinatariosBD() : int +getDestinatariosBD() : HashMap<String, Destinatario> +insertNotaBD() : int +updateNotaBD(Nota : Nota) : int +removeNotaBD() : int +removeNotaBD(Nota : Nota) : int </pre> </div> <div style="width: 35%;"> <pre> classDiagram class Nota { -Tema : String -Texto : String -Descartada : boolean = False +Nota() +Nota(Fecha, Redactor, Tema, Texto, Destinatarios) +getRedactor() +setRedactor(Redactor) +getTema() +setTema(Tema) +getText() +setTexto(Texto) +getDescartada() +setDescartada(Descartada) +getDestinatarios() +setDestinatarios(Destinatarios) +addDestinatario(Destinatario) +updateDestinatario(Dest_old, Dest_new) +removeDestinatario(Destinatario) +getDestinatario(DNI) +toString() +insertDestinatarioBD(Destinatario) +updateDestinatarioBD(Dest_old, Dest_new) +removeDestinatarioBD(Destinatario) +getDestinatarioBD(DNI) +insertDestinatariosBD(Destinatarios) +updateDestinatariosBD(Destinatarios) +removeDestinatariosBD() +getDestinatariosBD() +insertNotaBD() +updateNotaBD(Nota) +removeNotaBD() +removeNotaBD(Nota) } class Destinatario { -Leida : boolean = False -Descartada : boolean = False +Destinatario(Receptor : Persona) +Destinatario(Receptor : Persona, Leida : boolean, Descartada : boolean) +getReceptor() +setReceptor(Receptor : Persona) +getLeida() +setLeida(Leida : boolean) +getDescartada() +setDescartada(Descartada : boolean) +toString() } Nota "1" -- "1" Destinatario Destinatario "1" -- "0..*" Destinatario </pre> </div> </div>
Estructuras	

Notas



Nota	Atribs.	<i>Tema</i> <i>Texto</i> <i>Descartada</i>	Tema de la nota. Texto de la nota. Denota si ha sido descartada por su autor.
	Méts.	<i>Nota</i> <i>getXXXX</i> <i>setXXXX</i> <i>addDestinatario</i> <i>updateDestinatario</i> <i>removeDestinatario</i> <i>toString</i> <i>insertDestinatarioBD</i> <i>updateDestinatarioBD</i> <i>removeDestinatarioBD</i> <i>insertDestinatariosBD</i> <i>updateDestinatariosBD</i> <i>removeDestinatariosBD</i> <i>insertNotaBD</i> <i>updateNotaBD</i> <i>removeNotaBD</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. Añadir un destinatario a la nota. Modificar un destinatario ... Borrar un destinatario ... String con el estado del objeto. Insertar un destinatario en la nota en la BD. Modificar un destinatario ... Borrar un destinatario ... Insertar los destinatarios de la nota en la BD. Modificar los destinatarios ... Borrar los destinatarios ... Insertar la nota en la base de datos. Modificar la nota ... Borrar la nota ...
Destinatario	Atribs.	<i>Leida</i> <i>Descartada</i>	Denota si la nota ha sido leída. Denota si la nota ha sido descartada.
	Méts.	<i>GLL_Dev</i> <i>getXXXX</i> <i>setXXXX</i> <i>toString</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. String con el estado del objeto.

Notas

PARTES DE SEGURIDAD																																					
CLASES	DESCRIPCIÓN																																				
Transforms.	<div style="text-align: center;"> <table border="1" style="margin-bottom: 10px;"> <thead> <tr> <th colspan="3">Parte_Seguridad</th> </tr> </thead> <tbody> <tr> <td>+Fecha</td> <td>datetime</td> <td>Nullable = false</td> </tr> <tr> <td>Cod_Vig</td> <td>varchar(6)</td> <td>Nullable = false</td> </tr> <tr> <td>Incidencias</td> <td>varchar(1000)</td> <td>Nullable = false</td> </tr> </tbody> </table> <p style="text-align: center;">↓</p> <table border="1"> <thead> <tr> <th colspan="2">Parte_Seguridad</th> </tr> </thead> <tbody> <tr> <td>-Incidencias</td> <td>: String</td> </tr> <tr> <td>+Parte_Seguridad</td> <td>(Fecha : Calendar, Vigilante : Vigilante, Incidencias : String) : Parte_Seguridad</td> </tr> <tr> <td>+getVigilante()</td> <td>: Vigilante</td> </tr> <tr> <td>+setVigilante</td> <td>(Vigilante : Vigilante)</td> </tr> <tr> <td>+getIncidencias()</td> <td>: String</td> </tr> <tr> <td>+setIncidencias</td> <td>(Incidencias : String)</td> </tr> <tr> <td>+toString()</td> <td>: String</td> </tr> <tr> <td>+insertParteSeguridadBD()</td> <td>: int</td> </tr> <tr> <td>+updateParteSeguridadBD</td> <td>(Parte_Seguridad : Parte_Seguridad) : int</td> </tr> <tr> <td>+removeParteSeguridadBD()</td> <td>: int</td> </tr> </tbody> </table> </div>			Parte_Seguridad			+Fecha	datetime	Nullable = false	Cod_Vig	varchar(6)	Nullable = false	Incidencias	varchar(1000)	Nullable = false	Parte_Seguridad		-Incidencias	: String	+Parte_Seguridad	(Fecha : Calendar, Vigilante : Vigilante, Incidencias : String) : Parte_Seguridad	+getVigilante()	: Vigilante	+setVigilante	(Vigilante : Vigilante)	+getIncidencias()	: String	+setIncidencias	(Incidencias : String)	+toString()	: String	+insertParteSeguridadBD()	: int	+updateParteSeguridadBD	(Parte_Seguridad : Parte_Seguridad) : int	+removeParteSeguridadBD()	: int
Parte_Seguridad																																					
+Fecha	datetime	Nullable = false																																			
Cod_Vig	varchar(6)	Nullable = false																																			
Incidencias	varchar(1000)	Nullable = false																																			
Parte_Seguridad																																					
-Incidencias	: String																																				
+Parte_Seguridad	(Fecha : Calendar, Vigilante : Vigilante, Incidencias : String) : Parte_Seguridad																																				
+getVigilante()	: Vigilante																																				
+setVigilante	(Vigilante : Vigilante)																																				
+getIncidencias()	: String																																				
+setIncidencias	(Incidencias : String)																																				
+toString()	: String																																				
+insertParteSeguridadBD()	: int																																				
+updateParteSeguridadBD	(Parte_Seguridad : Parte_Seguridad) : int																																				
+removeParteSeguridadBD()	: int																																				
Estructuras																																					
Parte_Seguridad	Atribs.	<i>Incidencias</i>	Incidencias acontecidas durante el servicio.																																		
	Méts.	<i>Parte_Seguridad</i> <i>getXXXX</i> <i>setXXXX</i> <i>toString</i> <i>insertParteSeguridadBD</i> <i>updateParteSeguridadBD</i> <i>removeParteSeguridadBD</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. String con el estado del objeto. Insertar un parte de seguridad en la BD. Modificar un parte de seguridad ... Borrar un parte de seguridad ...																																		

Partes de seguridad



FECHAS			
CLASES	DESCRIPCIÓN		
Clases	<pre> Utilidades Calendar +calendarToString(Fecha : Calendar) : String +stringToCalendar(Fecha : String) : Calendar UnaFecha #Fecha_i = Now() +UnaFecha() : UnaFecha +UnaFecha(Fecha_i : Calendar) : UnaFecha +getFecha() : Calendar +setFecha(Fecha_i : Calendar) +toString() : String DosFechas -Fecha_j = null +DosFechas() : DosFechas +DosFechas(Fecha_i : Calendar) : DosFechas +DosFechas(Fecha_i : Calendar, Fecha_j : Calendar) : DosFechas +getFecha_i() : Calendar +setFecha_i(Fecha_i : Calendar) +toString() : String </pre>		<pre> UtilidadesBD -DireccionURL : String -Usuario : String -Pass : String #Con : Connection +abrirConexion() : Statement +cerrarConexion() +ejecutarQuery(Query : String) : ResultSet +ejecutarQuery(DireccionURL : String, Usuario : String, Pass : String, Query : String) : ResultSet +ejecutarUpdate(Query : String) +ejecutarUpdate(DireccionURL : String, Usuario : String, Pass : String, Query : String) </pre>
	Utilidades Calendar	Atribs.	---
	Méts.	<i>calendarToString</i> <i>stringToCalendar</i>	Pasar de un objeto Calendar a otro String. Pasar de un objeto String a otra Calendar.
UnaFecha	Atribs.	<i>Fecha_i</i>	Fecha.
	Méts.	<i>UnaFecha</i> <i>getXXXX</i> <i>setXXXX</i> <i>toString</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. String con el estado del objeto.
DosFechas	Atribs.	<i>Fecha_j</i>	Fecha.
	Méts.	<i>DosFechas</i> <i>getXXXX</i> <i>setXXXX</i> <i>toString</i>	Constructores. Obtener valores de los atributos. Establecer valores de los atributos. String con el estado del objeto.

Fechas



BASE DE DATOS			
CLASES	DESCRIPCIÓN		
Clases	<pre> UtilidadesBD -DireccionURL : String -Usuario : String -Pass : String #Con : Connection +abrirConexion() : Statement +cerrarConexion() +ejecutarQuery(Query : String) : ResultSet +ejecutarQuery(DireccionURL : String, Usuario : String, Pass : String, Query : String) : ResultSet +ejecutarUpdate(Query : String) +ejecutarUpdate(DireccionURL : String, Usuario : String, Pass : String, Query : String) </pre>		
UtilidadesBD	Atribs.	<i>DireccionURL</i> <i>Usuario</i> <i>Pass</i> <i>Con</i>	Dirección URL. Nombre de usuario. Contraseña. Conexión con la base de datos.
	Méts.	<i>abrirConexion</i> <i>cerrarConexion</i> <i>ejecutarQuery</i> <i>ejecutarUpdate</i>	Abrir conexión con la base de datos. Cerrar conexión con la base de datos. Ejecutar una sentencia (SELECT). Ejecutar una actualización (INSERT,...).

Base de datos

EXCEPCIONES			
CLASES	DESCRIPCIÓN		
Clases	<pre> DataException +toString() </pre>		
Data Exception	Atribs.	---	---
	Méts.	<i>toString</i>	String con el estado del objeto (tipo de excepción).

Excepciones

5. Codificación e integración

Una vez se dispone de un buen diseño detallado, la programación no implica más dificultad que la de saber manejar e iterar todas y cada una de las estructuras que manejamos, sin olvidar la correcta codificación de todas las sentencias DML. No creemos conveniente explicar cómo se ha implementado todo el diagrama, pues es una tarea muy extensa que creemos inservible en este apartado. Por ello, hemos documentado todo el código proporcionado de una forma muy pormenorizada, de modo que si el lector desea consultar cualquier detalle no tendrá más que acudir a éste.

En cuanto a la integración no cabe decir nada, ya que la API, aun estando relacionada muy estrechamente con la base de datos, no puede ser integrada en ningún componente en estos momentos. A partir de las siguientes iteraciones será cuando comencemos a integrarla con la GUI y el trinomio GUI-API-BD comenzará a tener un verdadero sentido, será algo tangible.



6. Pruebas

6.1. De unidad

Disponemos de muchas clases, cada una de ellas con muchos métodos, dando lugar a un conjunto bastante abultado de elementos a probar. Tenemos:

- Constructores, métodos de obtención y asignación de valores a los atributos y operaciones para los objetos compuestos:
Podrían ser probados a través de la herramienta *JUnit*, pero resultan bastante sencillos con que resultaría un trabajo muy poco provechoso diseñar todas las clases de pruebas para encontrar escasos errores. Evitamos esta actividad y solucionaremos cualquier error que pudiera surgir en este aspecto en el futuro.
- Operaciones DML sobre el objeto:
Deberíamos probar todas y cada una de las operaciones DML efectuadas en cada una de las clases. Si nos damos cuenta, a partir de la siguiente iteración desarrollaremos todos los casos de uso que hacen uso de éstas, con que optamos por no realizarlas en este momento, trazar todas las consultas por el prompt de NetBeans y modificar en un futuro todas las consultas desarrolladas que no sean correctas en este momento.

6.2. De integridad

Hasta el momento solo disponemos de la base de datos y la API, con que lo único que debemos probar es que ambas se comuniquen correctamente. Para ello se desarrolla el caso de prueba de la siguiente página.



Caso de prueba	2.01 - Comunicación API - BD
Objetivo	Comprobar que la comunicación entre la API desarrollada y la base de datos es correcta.
Actividades	<p>Crear una clase que conecte con la base de datos y cree un objeto de tipo Statement. Llamaremos al método <i>getMetaData</i> y recogemos el objeto DatabaseMetaData resultante, imprimiéndolo por el prompt de NetBeans. Si no salta ninguna excepción y el mensaje es imprimido podemos dar por buena la prueba.</p> <p>Nota: Debemos establecer el atributo UtilidadesBD.Con a público solo durante el desarrollo de esta prueba.</p>
Técnicas y herramientas	Aplicación NetBeans.
Datos de entrada	<ul style="list-style-type: none">• Servidor de bases de datos corriendo en la máquina.• Clase <i>PruebaComunAPIBD</i>: <pre>/** CLASE PRUEBACOMUNAPIBD: Clase que prueba que la comunicacion entre la API y la BD es correcta * * @author Napal Basterrica, Hector * @version 1.0 */ public class PruebaComunAPIBD { // ATRIBUTOS ----- /** UBD: Objeto de tipo UtilidadesBD */ public static UtilidadesBD UBD = new UtilidadesBD(); // METODOS ----- // MAIN public static void main (String [] args) { try { UBD.abrirConexion(); System.out.println(UBD.Con.getMetaData().toString()); } catch (SQLException sqle) { System.err.println("ERROR: No ha sido posible establecer la conexión con la BD."); } } }</pre>
Salida esperada	Mensaje por el prompt de NetBeans indicando los metadatos de la base de datos
Salida observada	La esperada: <pre>run: com.mysql.jdbc.JDBC4DatabaseMetaData@94949a BUILD SUCCESSFUL (total time: 1 second)</pre>
Evaluación/ conclusión	La comunicación entre la API y la base de datos funciona correctamente.

Caso de prueba para la comunicación API - BD



7. Evaluación de la iteración

Al llegar a este punto podemos decir que disponemos de buena parte de la implementación de la API con la que trabajar con la base de datos. Puede parecer que hemos dedicado un gran esfuerzo para terminar no teniendo apenas nada, pero una vez nos dispongamos a programar la interfaz gráfica veremos lo rápido y cómodo que resulta, no olvidando las futuras ventajas que habremos obtenido en su mantenimiento.

Se trata de una etapa de mucha importancia, y tras los artefactos de salida y los resultados de las pruebas obtenidos, podemos decir que la hemos completado con éxito.

Pamplona, a 12 de febrero de 2010.

Héctor Napal Basterrica
Responsable del proyecto



accessManager
Desarrollo de un sistema de control de accesos a un centro

Iteración C3: Accesos

Fecha de entrega: 26 de febrero de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Definiciones, acrónimos y abreviaciones
- 1.4. Estructura del documento

2. Estado del proyecto

3. Análisis

- 3.1. Planificación de la iteración
- 3.2. Análisis de los casos de uso y escenarios

4. Diseño

- 4.1. Caso de uso 1: Logueo
- 4.2. Caso de uso 2: Visualizar accesos
- 4.3. Caso de uso 3: Visualizar acceso
- 4.4. Caso de uso 4: Dar entrada
- 4.5. Caso de uso 5: Prestar llaves y/o llaveros
- 4.6. Caso de uso 6: Borrar acceso
- 4.7. Caso de uso 7: Modificar acceso
- 4.8. Caso de uso 8: Borrar préstamo
- 4.9. Caso de uso 9: Dar salida
- 4.10. Caso de uso 10: Devolver llaves y llaveros
- 4.11. Caso de uso 11: Listar accesos
- 4.12. Caso de uso 12: Dar salida automática

5. Codificación e integración

- 5.1. Caso de uso 1: Login
- 5.2. Casos de uso 2 a 11: Visualizar accesos a Listar accesos
- 5.3. Casos de uso 12: Dar salida automática

6. Pruebas

- 6.1. De unidad
- 6.2. De integración
- 6.3. De sistema

7. Evaluación de la iteración

8. Despliegue



1. Introducción

1.1. Propósito

El propósito con el que se confecciona este documento es el de documentar todo el desarrollo de la funcionalidad que ofrecerá la aplicación para llevar a cabo una correcta gestión de los accesos.

1.2. Audiencia

Este documento tendrá acompañará a su editor en el momento en que los resultados obtenidos en esta iteración sean presentados al cliente, si bien en un principio no pensamos entregárselo hasta el final del desarrollo, junto con toda la demás documentación.

1.3. Definiciones, acrónimos y abreviaciones

- | | | |
|------|-------------|----------------|
| • BD | | Base de datos |
| • PK | Primary Key | Clave primaria |

1.4. Estructura del documento

Tras estas líneas comenzamos haciendo una pequeña recapitulación, visionando en qué punto del proyecto nos encontramos, qué hemos hecho y qué nos queda por hacer.

Una vez situados y sabiendo cuáles serán nuestros próximos pasos, planificamos la iteración y qué casos de uso resolveremos en la sección 3.

En la siguiente procedemos con el diseño de los casos analizados, explicando cómo resolveremos todos y cada uno de ellos y acompañando estas explicaciones con sus correspondientes diagramas.

Las etapas de análisis y diseño dan lugar a la de codificación e integración, donde llevaremos a la realidad todo el trabajo realizado en las secciones anteriores. Todas las explicaciones necesarias se encuentran aquí, en el quinto apartado de este documento.

Probaremos que todo el trabajo realizado cumple con nuestras expectativas iniciales en el capítulo 6, concluyendo este escrito con una evaluación del trabajo realizado y las conclusiones extraídas de la etapa de despliegue.



2. Estado del proyecto

Transcurridas las dos primeras iteraciones podemos decir que ya disponemos de una buena base sobre la que sustentar toda la aplicación: la base de datos y una API que la maneja. A partir de ahora y a lo largo de las siguientes iteraciones (hasta la C21) nos dedicaremos a desarrollar toda la aplicación que manejará el usuario, en lo que a la parte de interfaz gráfica se refiere y la que en realidad será la encargada de ofrecer toda la funcionalidad. La forma de operar es sencilla: implementaremos los casos de uso descritos en la *Especificación inicial de los casos de uso*, en orden de mayor a menor importancia; orden que determinó la organización del desarrollo en iteraciones y que puede ser consultado en la documentación generada en la fase de Elaboración.

3. Análisis

3.1. Planificación de la iteración

En primer lugar nos centramos en desarrollar toda la funcionalidad relativa al tema de accesos, realizando previamente el trivial caso de uso de logueo en el sistema. Para todo ello, disponemos de la siguiente planificación:

PLAN DE ITERACIÓN C3: ACCESOS	
ASPECTO	DESCRIPCIÓN
Objetivos	Desarrollar la funcionalidad de la aplicación correspondiente a los accesos, así como el logueo en el sistema.
Artefactos de entrada	<ul style="list-style-type: none">• <i>Especificación inicial de los casos de uso</i>• <i>Planificación temporal y costes detallados del proyecto</i>• Base de datos y API desarrolladas en las iteraciones anteriores.
Artefactos modificados o creados	<ul style="list-style-type: none">• Diagrama de clases de la API.• Clases Java de la API y de la aplicación.• El presente documento.

Planificación de la iteración



Calendarización detallada	Análisis + Pruebas	22/02/10	<ul style="list-style-type: none"> Planificación de la iteración y gestión de riesgos. Análisis de los casos de uso y escenarios. Planificación de las pruebas. Inicio de la documentación. 																																																																																																																																																																																																																																						
	Diseño + Pruebas	23/02/10	<ul style="list-style-type: none"> Diseño de opciones arquitectónicas. 																																																																																																																																																																																																																																						
		24/02/10	<ul style="list-style-type: none"> Diseño de pruebas. 																																																																																																																																																																																																																																						
	Codificación + Pruebas	24/02/10	<ul style="list-style-type: none"> Codificación e integración de la solución. Ejecución y evaluación de pruebas. 																																																																																																																																																																																																																																						
		26/02/10	<ul style="list-style-type: none"> Evaluación del prototipo. Preparación de la entrega ejecutable. 																																																																																																																																																																																																																																						
Despliegue	26/02/10	<ul style="list-style-type: none"> Muestra, prueba y evaluación conjunta con el cliente. Concreción de los cambios necesarios. Finalización de toda la documentación. 																																																																																																																																																																																																																																							
<table border="1"> <thead> <tr> <th>Nombre de tarea</th> <th>Duración</th> <th>Comienzo</th> <th>Fin</th> <th>0</th> <th>lun 22 feb</th> <th>0</th> <th>mié 24 feb</th> <th>0</th> <th>vie 26 feb</th> </tr> </thead> <tbody> <tr> <td>Estudio de oportunidad (inception)</td> <td>7 días</td> <td>jue 07/01/10</td> <td>vie 15/01/10</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Elaboración</td> <td>16 días</td> <td>lun 18/01/10</td> <td>lun 08/02/10</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Construcción</td> <td>41 días</td> <td>mar 09/02/10</td> <td>mar 06/04/10</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td> Iteración C1: La base de datos</td> <td>4 días</td> <td>mar 09/02/10</td> <td>vie 12/02/10</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td> Iteración C2: API para la base de datos</td> <td>5 días</td> <td>lun 15/02/10</td> <td>vie 19/02/10</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td> Iteración C3: Accesos</td> <td>5 días</td> <td>lun 22/02/10</td> <td>vie 26/02/10</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td> Análisis</td> <td>1 día</td> <td>lun 22/02/10</td> <td>lun 22/02/10</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td> Diseño</td> <td>1,5 días</td> <td>mar 23/02/10</td> <td>mié 24/02/10</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td> Implementación</td> <td>2 días</td> <td>mié 24/02/10</td> <td>vie 26/02/10</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td> Pruebas</td> <td>5 días</td> <td>lun 22/02/10</td> <td>vie 26/02/10</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td> Despliegue</td> <td>0,5 días</td> <td>vie 26/02/10</td> <td>vie 26/02/10</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td> Iteración C4: Empresas y personal</td> <td>5 días</td> <td>lun 01/03/10</td> <td>vie 05/03/10</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td> Iteración C5: Tarjetas, tipos de acceso y cargos</td> <td>3 días</td> <td>lun 08/03/10</td> <td>mié 10/03/10</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td> Iteración C6: Zonas, llaves y llaveros</td> <td>5 días</td> <td>jue 11/03/10</td> <td>mié 17/03/10</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td> Iteración C7: Préstamos</td> <td>2 días</td> <td>jue 18/03/10</td> <td>vie 19/03/10</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td> Iteración C8: Cuentas</td> <td>2 días</td> <td>lun 22/03/10</td> <td>mar 23/03/10</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td> Iteración C9: Partes de seguridad</td> <td>3 días</td> <td>mié 24/03/10</td> <td>vie 26/03/10</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td> Iteración C10: Notas</td> <td>3 días</td> <td>lun 29/03/10</td> <td>mié 31/03/10</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td> Iteración C11: Base de datos</td> <td>2 días</td> <td>jue 01/04/10</td> <td>vie 02/04/10</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td> Iteración C12: Diseño, personalización y configuración</td> <td>2 días</td> <td>lun 05/04/10</td> <td>mar 06/04/10</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td> Capacidad operacional</td> <td>0 días</td> <td>mar 06/04/10</td> <td>mar 06/04/10</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Transición</td> <td>8 días</td> <td>mié 07/04/10</td> <td>vie 16/04/10</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>				Nombre de tarea	Duración	Comienzo	Fin	0	lun 22 feb	0	mié 24 feb	0	vie 26 feb	Estudio de oportunidad (inception)	7 días	jue 07/01/10	vie 15/01/10							Elaboración	16 días	lun 18/01/10	lun 08/02/10							Construcción	41 días	mar 09/02/10	mar 06/04/10							Iteración C1: La base de datos	4 días	mar 09/02/10	vie 12/02/10							Iteración C2: API para la base de datos	5 días	lun 15/02/10	vie 19/02/10							Iteración C3: Accesos	5 días	lun 22/02/10	vie 26/02/10							Análisis	1 día	lun 22/02/10	lun 22/02/10							Diseño	1,5 días	mar 23/02/10	mié 24/02/10							Implementación	2 días	mié 24/02/10	vie 26/02/10							Pruebas	5 días	lun 22/02/10	vie 26/02/10							Despliegue	0,5 días	vie 26/02/10	vie 26/02/10							Iteración C4: Empresas y personal	5 días	lun 01/03/10	vie 05/03/10							Iteración C5: Tarjetas, tipos de acceso y cargos	3 días	lun 08/03/10	mié 10/03/10							Iteración C6: Zonas, llaves y llaveros	5 días	jue 11/03/10	mié 17/03/10							Iteración C7: Préstamos	2 días	jue 18/03/10	vie 19/03/10							Iteración C8: Cuentas	2 días	lun 22/03/10	mar 23/03/10							Iteración C9: Partes de seguridad	3 días	mié 24/03/10	vie 26/03/10							Iteración C10: Notas	3 días	lun 29/03/10	mié 31/03/10							Iteración C11: Base de datos	2 días	jue 01/04/10	vie 02/04/10							Iteración C12: Diseño, personalización y configuración	2 días	lun 05/04/10	mar 06/04/10							Capacidad operacional	0 días	mar 06/04/10	mar 06/04/10							Transición	8 días	mié 07/04/10	vie 16/04/10						
Nombre de tarea	Duración	Comienzo	Fin	0	lun 22 feb	0	mié 24 feb	0	vie 26 feb																																																																																																																																																																																																																																
Estudio de oportunidad (inception)	7 días	jue 07/01/10	vie 15/01/10																																																																																																																																																																																																																																						
Elaboración	16 días	lun 18/01/10	lun 08/02/10																																																																																																																																																																																																																																						
Construcción	41 días	mar 09/02/10	mar 06/04/10																																																																																																																																																																																																																																						
Iteración C1: La base de datos	4 días	mar 09/02/10	vie 12/02/10																																																																																																																																																																																																																																						
Iteración C2: API para la base de datos	5 días	lun 15/02/10	vie 19/02/10																																																																																																																																																																																																																																						
Iteración C3: Accesos	5 días	lun 22/02/10	vie 26/02/10																																																																																																																																																																																																																																						
Análisis	1 día	lun 22/02/10	lun 22/02/10																																																																																																																																																																																																																																						
Diseño	1,5 días	mar 23/02/10	mié 24/02/10																																																																																																																																																																																																																																						
Implementación	2 días	mié 24/02/10	vie 26/02/10																																																																																																																																																																																																																																						
Pruebas	5 días	lun 22/02/10	vie 26/02/10																																																																																																																																																																																																																																						
Despliegue	0,5 días	vie 26/02/10	vie 26/02/10																																																																																																																																																																																																																																						
Iteración C4: Empresas y personal	5 días	lun 01/03/10	vie 05/03/10																																																																																																																																																																																																																																						
Iteración C5: Tarjetas, tipos de acceso y cargos	3 días	lun 08/03/10	mié 10/03/10																																																																																																																																																																																																																																						
Iteración C6: Zonas, llaves y llaveros	5 días	jue 11/03/10	mié 17/03/10																																																																																																																																																																																																																																						
Iteración C7: Préstamos	2 días	jue 18/03/10	vie 19/03/10																																																																																																																																																																																																																																						
Iteración C8: Cuentas	2 días	lun 22/03/10	mar 23/03/10																																																																																																																																																																																																																																						
Iteración C9: Partes de seguridad	3 días	mié 24/03/10	vie 26/03/10																																																																																																																																																																																																																																						
Iteración C10: Notas	3 días	lun 29/03/10	mié 31/03/10																																																																																																																																																																																																																																						
Iteración C11: Base de datos	2 días	jue 01/04/10	vie 02/04/10																																																																																																																																																																																																																																						
Iteración C12: Diseño, personalización y configuración	2 días	lun 05/04/10	mar 06/04/10																																																																																																																																																																																																																																						
Capacidad operacional	0 días	mar 06/04/10	mar 06/04/10																																																																																																																																																																																																																																						
Transición	8 días	mié 07/04/10	vie 16/04/10																																																																																																																																																																																																																																						
Recursos	Temporales	5 días																																																																																																																																																																																																																																							
	Humanos	1 persona																																																																																																																																																																																																																																							
	Software	Sistema operativo	Microsoft Windows 7																																																																																																																																																																																																																																						
		Ofimática	Microsoft Word 2010 Adobe Acrobat Professional 8																																																																																																																																																																																																																																						
		Diseño	Visual Paradigm 4.2																																																																																																																																																																																																																																						
Codificación		NetBeans 6.8																																																																																																																																																																																																																																							
Hardware	1 PC																																																																																																																																																																																																																																								
Casos de uso y escenarios implementados	<table border="0"> <tr> <td>1 - Login</td> <td>7 - Modificar acceso</td> </tr> <tr> <td>2 - Visualizar accesos</td> <td>8 - Borrar préstamo</td> </tr> <tr> <td>3 - Visualizar acceso</td> <td>9 - Dar salida</td> </tr> <tr> <td>4 - Dar entrada</td> <td>10 - Devolver llaves y/o llaveros</td> </tr> <tr> <td>5 - Prestar llaves y/o llaveros</td> <td>11 - Listar accesos</td> </tr> <tr> <td>6 - Borrar acceso</td> <td>12 - Dar salida automática</td> </tr> </table>			1 - Login	7 - Modificar acceso	2 - Visualizar accesos	8 - Borrar préstamo	3 - Visualizar acceso	9 - Dar salida	4 - Dar entrada	10 - Devolver llaves y/o llaveros	5 - Prestar llaves y/o llaveros	11 - Listar accesos	6 - Borrar acceso	12 - Dar salida automática																																																																																																																																																																																																																										
1 - Login	7 - Modificar acceso																																																																																																																																																																																																																																								
2 - Visualizar accesos	8 - Borrar préstamo																																																																																																																																																																																																																																								
3 - Visualizar acceso	9 - Dar salida																																																																																																																																																																																																																																								
4 - Dar entrada	10 - Devolver llaves y/o llaveros																																																																																																																																																																																																																																								
5 - Prestar llaves y/o llaveros	11 - Listar accesos																																																																																																																																																																																																																																								
6 - Borrar acceso	12 - Dar salida automática																																																																																																																																																																																																																																								

Planificación de la iteración



3.2. Análisis de los casos de uso y escenarios

Estas son las descripciones de trazo fino generadas para los casos de uso que acometeremos en esta iteración:

Caso de uso 1: Login		
Descripción	<p>El primer paso para acceder al sistema es realizar un logueo, que conducirá a la pantalla principal de la aplicación bajo las condiciones impuestas por el tipo de usuario que realice el acceso al sistema.</p> <p>Realizar el logueo es sencillo, bastará con introducir el nombre de usuario y contraseña en la pantalla de inicio y presionar el botón de 'Entrar'.</p>	
Actores principal	Administrador, Vigilante, Autorizador y RP.	
Pre-condiciones	El usuario desea acceder al sistema.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario introduce su nombre de usuario y contraseña y pulsa el botón 'Entrar'. 2. El sistema realiza la comprobación de la existencia de la cuenta, pasando a cerrar la ventana 'Identificación' y a abrir la de 'Aplicación' con las funcionalidades que le estén permitidas. (*)
	Alternativos (*)	<p>El usuario y/o contraseña no existen o no casan con los de la base de datos:</p> <ul style="list-style-type: none"> • El sistema indica al usuario del error acontecido y se pasa al punto de partida, permaneciendo inalterados los datos introducidos anteriormente por el usuario.
Post-condiciones	El sistema ha desplegado la pantalla desde la que el usuario podrá realizar las funciones que estime oportunas.	

Caso de uso 1: Login



Caso de uso 2: Visualizar accesos	
Descripción	Uno de los requisitos que impone el cliente es que se puedan observar el conjunto de accesos de forma sencilla y rápida, instantánea. Habitualmente se trabaja con los accesos al centro que se han realizado en el día, con que será conveniente aportar una simple forma de visualizarlos y hacer accesible al usuario el resto de accesos a través de algún paso adicional.
Actores principal	Administrador y Vigilante.
Pre-condiciones	El usuario se encuentra logueado en el sistema y desea consultar los accesos realizados al centro.
Flujos	Básico <ol style="list-style-type: none">1. El administrador o vigilante desea consultar los accesos realizados en el centro en el día en que se encuentra. (*)2. El sistema comprueba que las fechas proporcionadas son correctas y realiza la búsqueda de accesos según los parámetros establecidos, mostrándole al usuario el resultado a través de una tabla.
	Alternativos (*) <p>El usuario desea consultar los accesos realizados entre días y horas específicas:</p> <ul style="list-style-type: none">• Redefine los parámetros que delimitan la búsqueda y se pasa al punto 2.
Post-condiciones	El usuario obtiene por pantalla los accesos realizados al centro entre los días y horas indicados.

Caso de uso 2: Visualizar accesos

Caso de uso 3: Visualizar acceso	
Descripción	Así como el usuario puede ver los datos característicos de todos aquellos accesos que cumplan con los parámetros establecidos, también debe poder visualizar todos los datos del acceso que desee.
Actores principal	Administrador y Vigilante.
Pre-condiciones	El usuario se encuentra logueado en el sistema y desea consultar todos los datos de uno de los accesos que visualiza en pantalla.
Flujos	Básico <ol style="list-style-type: none">1. El usuario selecciona un acceso en pantalla.2. El sistema se encarga de recoger todos los datos referentes a ese acceso y mostrarlos.
	Alternativos (*)
Post-condiciones	El usuario obtiene por pantalla todos los datos pertenecientes al acceso seleccionado.

Caso de uso 3: Visualizar acceso



Caso de uso 4: Dar entrada	
Descripción	Todo acceso al centro queda registrado en estos momentos en hojas manejadas por los vigilantes. Aquí lo que haremos es evitar este trámite, siendo almacenados todos los accesos mediante la aplicación.
Actores principal	Vigilante.
Pre-condiciones	Una persona desea acceder al centro y el vigilante, logueado en el sistema, desea registrar el acceso.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario indica al sistema que desea registrar un nuevo acceso. 2. El sistema le proporciona un formulario en el que definir todos los datos. 3. El usuario rellena el formulario, respondiéndole el sistema con la tarjeta que deberá emplear en el acceso y si es necesario que avise a alguna persona del centro. 4. Acto seguido, el vigilante hace constar el motivo y otros datos del acceso, además de las llaves y llaveros que prestará a la persona, y termina pidiendo al sistema que registre el acceso. 5. El sistema, tras una previa aprobación, hace caso al vigilante y le indica que todo el proceso ha sido resuelto satisfactoriamente. (*)
	<p>Alternativos (*)</p> <p>El usuario introduce datos con un formato erróneo:</p> <ul style="list-style-type: none"> • El sistema avisa al usuario del error acontecido y vuelve al punto de partida. <p>El usuario no da su aprobación para que el acceso quede registrado en el sistema:</p> <ul style="list-style-type: none"> • Éste no realiza ninguna inserción y da por concluido el caso de uso.
Post-condiciones	El acceso (y préstamos) ha sido registrado con éxito y el vigilante ya puede permitir la entrada del trabajador al centro.

Caso de uso 4: Dar entrada



Caso de uso 5: Prestar llaves y/o llaveros	
Descripción	Tras haber realizado un acceso, la persona puede que haya olvidado pedir una llave o llavero, con que deberá existir la posibilidad de poder modificar los préstamos realizados en un acceso.
Actores principal	Vigilante.
Pre-condiciones	Una persona que ya se encuentra en el centro desea que le sea prestada una nueva llave o llavero. El vigilante, que se encuentra logueado en el sistema, se dispone a registrar el préstamo.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El vigilante selecciona el acceso que desea modificar. 2. El sistema le muestra toda la información acerca de éste y le permite añadir préstamos si es que el acceso no posee salida. Le muestra la lista de llaves y llaveros que la persona no posee y que le pueden ser prestados. (*) 3. El usuario selecciona las llaves y/o llaveros del nuevo préstamo. (*) 4. El caso de uso termina registrando el sistema los préstamos en la base de datos.
	<p>Alternativos (*)</p> <p>El acceso consta de salida:</p> <ul style="list-style-type: none"> • El sistema no permitirá modificarlo, dando por terminado el caso de uso. <p>No existen llaves o llaveros que poder prestar:</p> <ul style="list-style-type: none"> • El sistema no mostrará ninguna llave ni llavero y el usuario no tendrá más opción que cerrar la ventana y no poder realizar ningún préstamo.
Post-condiciones	El vigilante registra los préstamos solicitados y puede ofrecer a la persona las llaves y llaveros que necesita.

Caso de uso 5: Prestar llaves y/o llaveros



Caso de uso 6: Borrar acceso	
Descripción	No es lo normal, pero bien por error en la introducción de los datos o bien por necesidad, puede ser preciso eliminar algunos de los accesos que constan en la base de datos.
Actores principal	Autorizador y Vigilante.
Pre-condiciones	El vigilante ha introducido algún acceso de forma errónea o el autorizador desea eliminar algunos de los accesos por las razones que estime oportunas. Sea el usuario que sea, se encuentra logueado en el sistema y con intención de borrar algunos de los accesos existentes.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario selecciona de la tabla de accesos los accesos que desee eliminar y pide al sistema que los borre. 2. Éste pide confirmación al primero, por si los accesos seleccionados no fueran los que realmente desea eliminar. 3. Tras confirmar el usuario, el borrado tiene lugar, los accesos dejan de existir y el sistema refresca la tabla de accesos. (*)
	<p>Alternativos (*)</p> <p>El usuario ha seleccionado accesos que no debía y cancela la operación de borrado:</p> <ul style="list-style-type: none"> • El sistema no realiza ningún borrado y permanece en el punto previo a que el usuario pidiese el borrado.
Post-condiciones	El usuario consigue lo que desea: los accesos dejan de existir en el sistema y la tabla de accesos está actualizada con los existentes.

Caso de uso 6: Borrar acceso



Caso de uso 7: Modificar acceso		
Descripción	En el caso de uso 5 hablábamos de cómo modificar un acceso en cuanto a añadir más préstamos. Ahora tocaremos la funcionalidad de poder modificar el asunto y otros datos y posteriormente remataremos el trabajo con el borrado y devolución de préstamos.	
Actores principal	Vigilante.	
Pre-condiciones	El usuario se encuentra logueado en el sistema y desea modificar los atributos de 'Motivo' y 'Otros datos' de un acceso que no dispone de salida.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El vigilante selecciona el acceso a modificar. 2. El sistema ofrece toda la información en el formulario y hace los campos editables. 3. El usuario los modifica de forma que queden a su gusto y le pide al sistema que los grabe.
	Alternativos (*)	Ninguno
Post-condiciones	El acceso posee en sus atributos 'Motivo' y 'Otros datos' valores correctos.	

Caso de uso 7: Modificar acceso

Caso de uso 8: Borrar préstamo		
Descripción	Otra forma de poder modificar un acceso es la de eliminar uno o varios de los préstamos que se hayan realizado en éste.	
Actores principal	Vigilante.	
Pre-condiciones	El vigilante se encuentra logueado en el sistema, está visualizando un acceso y desea eliminar uno o varios de los préstamos que lo conforman.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El vigilante selecciona todos los préstamos que desea eliminar y pulsa el botón de eliminado. 2. El sistema le pide confirmación de que los accesos que ha seleccionado son los que realmente desea eliminar. (*) 3. El vigilante responde afirmativamente, lo que hace que los préstamos dejen de existir en la base de datos y la tabla de préstamos queda actualizada.
	Alternativos (*)	<p>El usuario decide que no desea eliminar alguno de los préstamos seleccionados:</p> <ul style="list-style-type: none"> • Pulsa 'No' cuando el sistema le pide la confirmación y el sistema permanece en el estado que se llegó al visualizar el acceso.
Post-condiciones	Las descritas en el tercer punto del flujo básico.	

Caso de uso 8: Borrar préstamo



Caso de uso 9: Dar salida					
Descripción	Complementariamente a poder dar entradas, también se debe poder dar salidas a todo el personal que abandone el centro, existiendo dos vías: <ol style="list-style-type: none"> 1. Salida indicando que prestamos se devuelven. 2. Salida en la que se devuelven todos los préstamos pendientes en el acceso. 				
Actores principal	Vigilante.				
Pre-condiciones	El vigilante, con la aplicación corriendo, observa que una persona desea abandonar el centro. Ante esta situación, se dispone a registrar esta salida.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> <ol style="list-style-type: none"> 1. El usuario selecciona el acceso al que desea otorgar salida. 2. El sistema le muestra todos los datos del acceso. 3. El usuario da salida al acceso seleccionando los préstamos devueltos. 4. El sistema pide confirmación para llevar a cabo la modificación del acceso. 5. El usuario la da. (*) </td> </tr> <tr> <td>Alternativos (*)</td> <td> El usuario no da el visto bueno para que la salida se lleve a efecto: <ul style="list-style-type: none"> • El sistema no realiza ninguna acción y da por finalizado el caso de uso. </td> </tr> </table>	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona el acceso al que desea otorgar salida. 2. El sistema le muestra todos los datos del acceso. 3. El usuario da salida al acceso seleccionando los préstamos devueltos. 4. El sistema pide confirmación para llevar a cabo la modificación del acceso. 5. El usuario la da. (*) 	Alternativos (*)	El usuario no da el visto bueno para que la salida se lleve a efecto: <ul style="list-style-type: none"> • El sistema no realiza ninguna acción y da por finalizado el caso de uso.
	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona el acceso al que desea otorgar salida. 2. El sistema le muestra todos los datos del acceso. 3. El usuario da salida al acceso seleccionando los préstamos devueltos. 4. El sistema pide confirmación para llevar a cabo la modificación del acceso. 5. El usuario la da. (*) 			
Alternativos (*)	El usuario no da el visto bueno para que la salida se lleve a efecto: <ul style="list-style-type: none"> • El sistema no realiza ninguna acción y da por finalizado el caso de uso. 				
Post-condiciones	El acceso pasa a disponer de salida y todos los préstamos indicados han sido marcados como devueltos.				

Caso de uso 9: Dar salida

Caso de uso 10: Devolver llaves y/o llaveros					
Descripción	Existen casos en los que los prestatarios querrán devolver parte o todos sus préstamos antes de terminar su jornada.				
Actores principal	Vigilante.				
Pre-condiciones	El vigilante, que se encuentra en uso de accessManager, desea marcar como devueltos alguno de los préstamos visualizados en algún acceso.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> <ol style="list-style-type: none"> 1. El usuario selecciona los préstamos que desea marcar como devueltos e indica al sistema que los devuelva. 2. El sistema pide una previa confirmación, tras la cual procede a grabar los préstamos indicados como devueltos. (*) </td> </tr> <tr> <td>Alternativos (*)</td> <td> El usuario cancela la devolución de los préstamos: <ul style="list-style-type: none"> • El sistema no realiza ninguna operación. </td> </tr> </table>	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona los préstamos que desea marcar como devueltos e indica al sistema que los devuelva. 2. El sistema pide una previa confirmación, tras la cual procede a grabar los préstamos indicados como devueltos. (*) 	Alternativos (*)	El usuario cancela la devolución de los préstamos: <ul style="list-style-type: none"> • El sistema no realiza ninguna operación.
	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona los préstamos que desea marcar como devueltos e indica al sistema que los devuelva. 2. El sistema pide una previa confirmación, tras la cual procede a grabar los préstamos indicados como devueltos. (*) 			
Alternativos (*)	El usuario cancela la devolución de los préstamos: <ul style="list-style-type: none"> • El sistema no realiza ninguna operación. 				
Post-condiciones	El vigilante de seguridad ha marcado como devueltos los préstamos que indicaba.				

Caso de uso 10: Devolver llaves y/o llaveros



Caso de uso 11: Listar accesos		
Descripción	El usuario puede consultar cuando desee la lista de los accesos que quiera, pero además de esta funcionalidad debemos ofrecer al usuario la posibilidad de que pueda guardar los resultados en un reporte en PDF.	
Actores principal	Administrador y Vigilante.	
Pre-condiciones	El usuario se encuentra logueado en el sistema y desea obtener un listado de todos los accesos que se encuentra visualizando.	
Flujos	Básico	<ol style="list-style-type: none">1. El usuario presiona el botón de 'Generar reporte' e indica el destino (ruta) en el que desea almacenarlo.2. El sistema genera el reporte con los datos principales de todos los accesos y lo almacena donde el usuario ha indicado.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario obtiene el reporte con todos los accesos que se encuentra visualizando.	

Caso de uso 11: Listar accesos

Caso de uso 12: Dar salida automática		
Descripción	El vigilante dará salida a la gran mayoría de los accesos, pero siempre puede quedar la posibilidad que el vigilante se deje alguno. Para evitar que los accesos queden sin salida y por tanto las personas que constan en ellos no puedan volver a acceder al centro, programamos la aplicación de modo que el sistema de salida a estos accesos automáticamente.	
Actores principal	Sistema.	
Pre-condiciones	El sistema se inicia por primera vez en todo el día.	
Flujos	Básico	El sistema busca los accesos que no disponen de salida y los recorre uno a uno proporcionándoles la máxima hora de salida que pudieran tener según el tipo de acceso de cada persona.
	Alternativos (*)	Ninguno.
Post-condiciones	Se consigue que todos los accesos del día anterior dispongan de entrada y salida, permaneciendo la tabla Acceso coherente.	

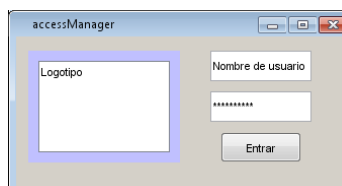
Caso de uso 12: Dar salida automática



4. Diseño

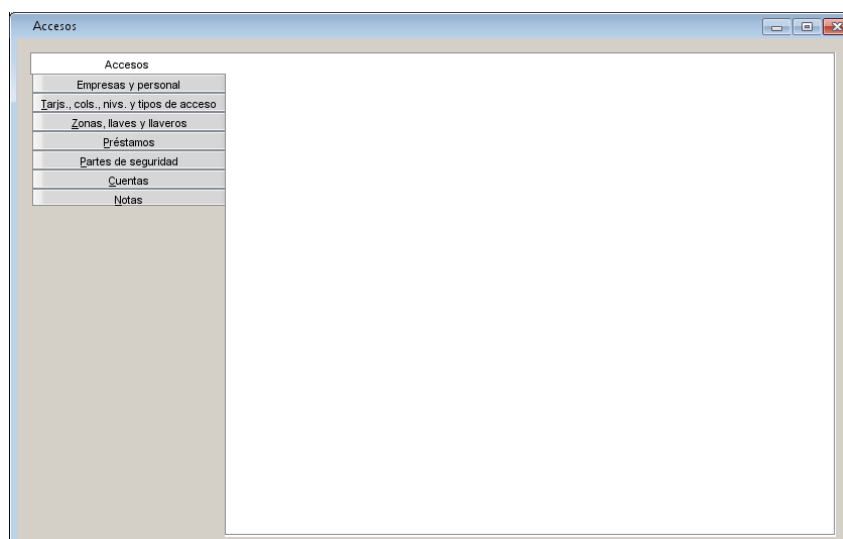
4.1. Caso de uso 1: Logueo

Para este primer caso de uso se diseñarán dos interfaces gráficas: una que permita realizar el logueo en el sistema...



Interfaz gráfica propuesta

... y otra a la que redirija la anterior en caso de realizarse con éxito el registro en el sistema:



Interfaz gráfica propuesta

De momento se diseñará solo el JTabbedPane que proporcionará toda la funcionalidad descrita en *Especificación inicial de los casos de uso* para así implementarlo en ésta y las siguientes iteraciones.



El flujo de sucesos que definen el proceso de logueo es:

1. Crear un objeto de tipo Cuenta con el nombre de usuario proporcionado.
2. Chequear que la contraseña introducida coincide con la del objeto creado.
 - a. En caso de coincidir: Se cierra la ventana de 'Identificación' y se abre la de 'Aplicación' con la funcionalidad permitida al tipo de usuario que accede.
 - b. En caso de no coincidir: Se avisa al usuario que el usuario y/o contraseña no son válidos, volviéndose al punto de partida.

De este procedimiento deducimos que debemos añadir a la API los siguientes métodos:

MÉTODO	FUNCIONALIDAD
<i>Cuenta</i>	Constructor que cree un objeto de tipo Cuenta a partir de los datos obtenidos de la base de datos según el parámetro Usuario pasado.
<i>checkPass</i>	Método que comprueba si la contraseña del objeto (atributo Pass) es la suministrada como argumento. En caso afirmativo devolverá el tipo de usuario que resulte ser, y en caso de no casar devolverá "Inexistente".

Métodos a implementar y a añadir a la API

Con éstos métodos ya estamos en disposición de crear el siguiente flujo de actividades...

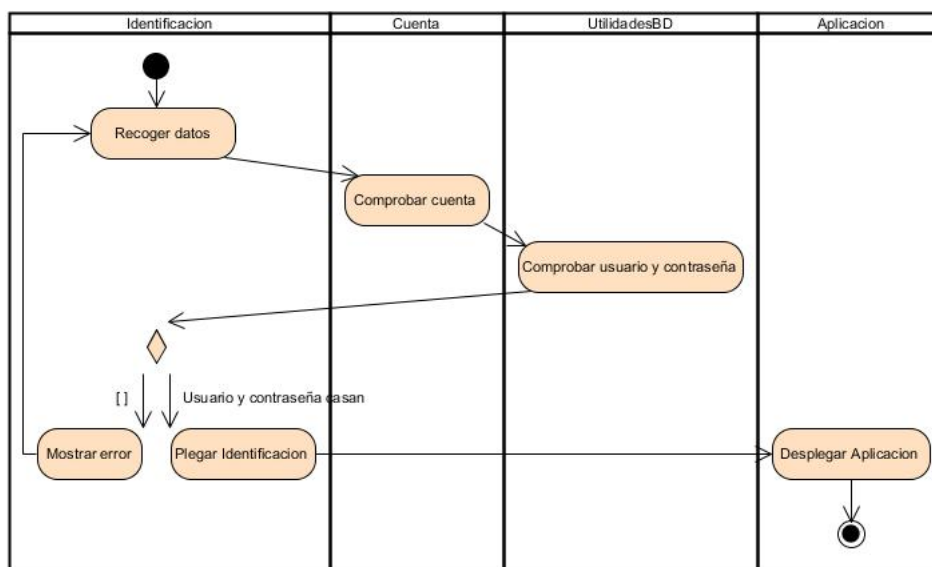


Diagrama de actividades



... y concretarlo en el siguiente diagrama de secuencia:

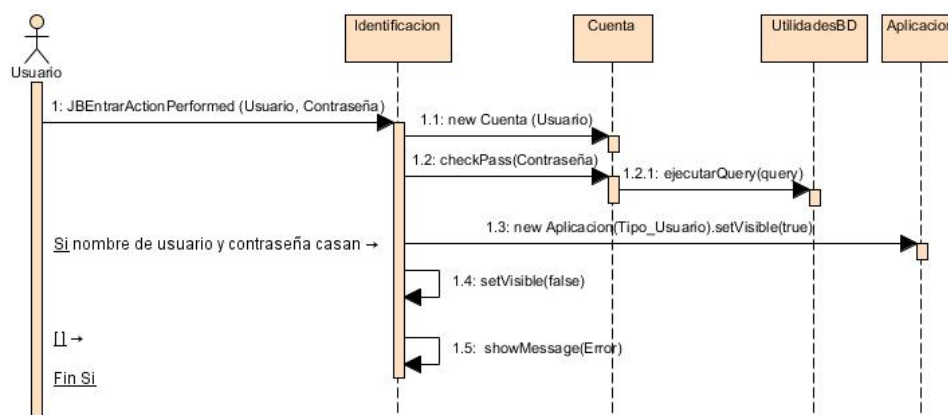
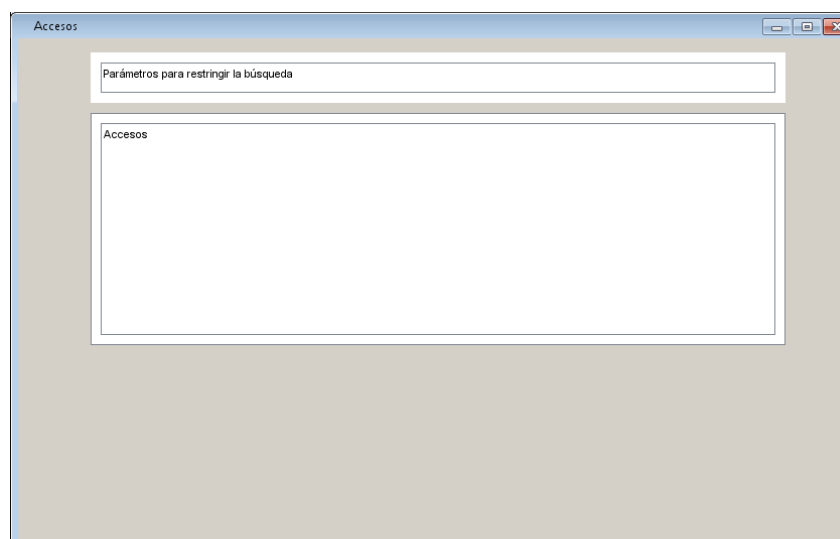


Diagrama de secuencia

4.2. Caso de uso 2: Visualizar accesos

Comenzamos a diseñar la pestaña de accesos colocando en la zona alta una tabla que mostrará los accesos y que permitirá realizar una búsqueda restringida sobre éstos:



Interfaz gráfica propuesta

Cada vez que se refina la búsqueda se efectuará el siguiente flujo de sucesos:

1. Crear un objeto de tipo Acceso con el que realizar la búsqueda.
2. Comprobar que las fechas y horas introducidas son correctas (en caso contrario, haremos que lo sean).
3. Recoger todos los parámetros necesarios para definir la búsqueda.
4. Rellenar la tabla de accesos con los resultados obtenidos.



Para todo este proceso necesitaremos un método que compruebe la corrección de las fechas y horas, otro que realice la búsqueda de accesos según los parámetros recogidos y otro que rellene la tabla con los resultados obtenidos:

MÉTODO	FUNCIONALIDAD
<i>comprobarFechasHorasAccesos</i>	Método que comprueba y corrige las fechas introducidas.
<i>obtenerAccesos</i>	Método que obtiene los accesos que cumplen las condiciones impuestas en la búsqueda.
<i>rellenarTabla</i>	Método que rellena una JTable a partir del ResultSet extraído de la consulta.

Métodos a implementar y a añadir a la API

Una vez que disponemos de todo lo necesario, podemos diseñar los siguientes diagramas que determinan el flujo de actividades y las llamadas que darán respuesta al caso de uso:

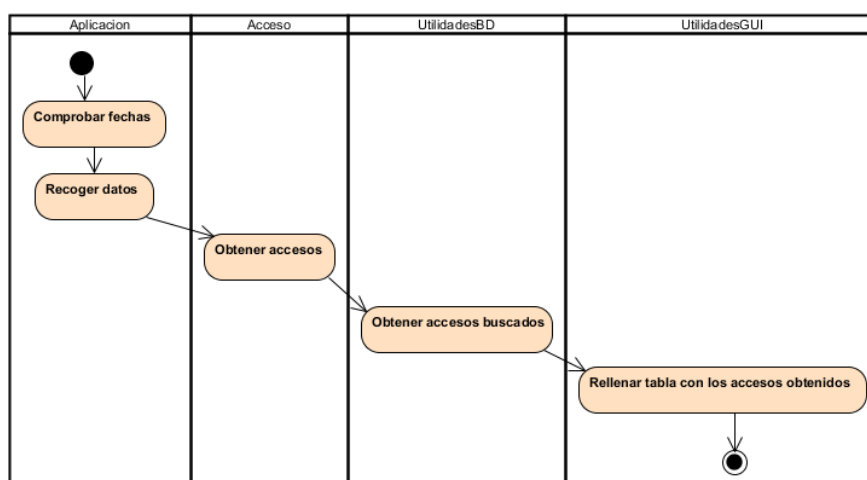


Diagrama de actividades

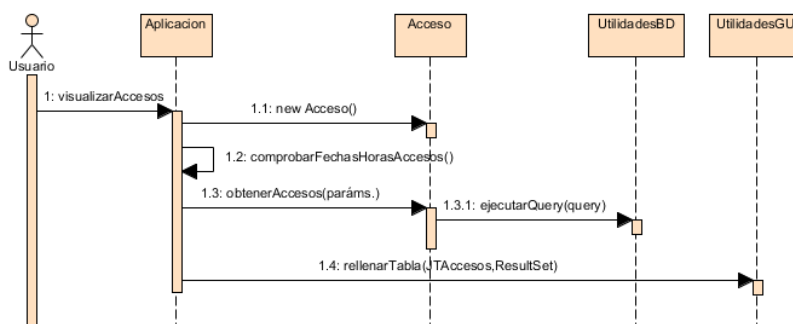
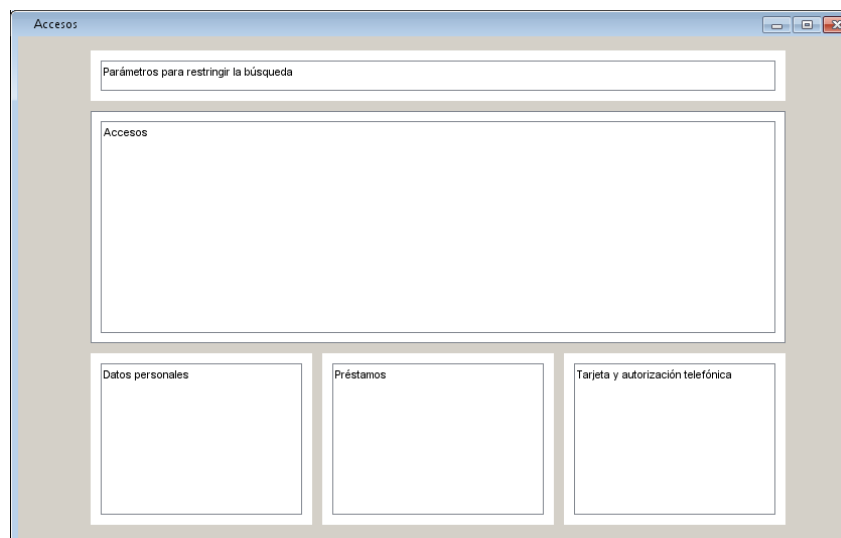


Diagrama de secuencia

4.3. Caso de uso 3: Visualizar acceso

A la interfaz diseñada hasta el momento se le añaden componentes en la parte inferior para que puedan ser mostrados en ellos todos los datos que componen cada acceso:



Interfaz gráfica propuesta

La secuencia de pasos que seguiremos es:

- a. Si disponemos de un acceso clicado:
 - Recuperamos el acceso de la BD.
 - b. Si no disponemos de ningún acceso clicado:
 - No recuperamos ningún acceso, siendo éste 'null'.
1. Mostramos los datos personales.
 2. Mostramos los datos de la tarjeta
 3. Mostramos los datos de los préstamos.
 4. Mostramos los datos de la autorización telefónica.

Como consecuencia, necesitamos crear los siguientes métodos:

MÉTODO	FUNCIONALIDAD
<i>Acceso</i>	Constructor que crea un objeto de tipo Acceso a partir de su PK (hora de entrada).
<i>obtenerLsLlsAcceso</i>	Método que obtiene todas las llaves y llaveros en préstamo del acceso indicado.
<i>Personal</i>	Constructor que recupera un Personal a partir de su DNI.
<i>obtenerTelefonos</i>	Método que obtiene los números de teléfono de una persona dada.

Métodos a implementar y a añadir a la API

accessManager · Iteración C3: Accesos · Página 18 de 55



La secuencia de actividades y de intercooperación de objetos que tendrán lugar para resolver el caso de uso son:

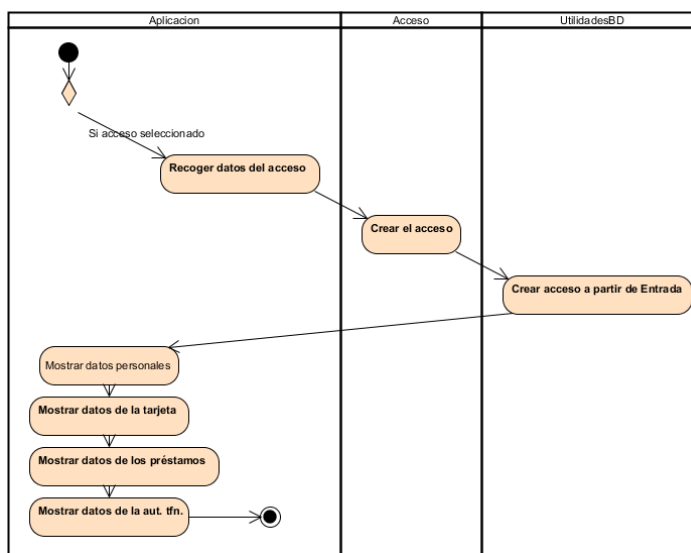


Diagrama de actividades

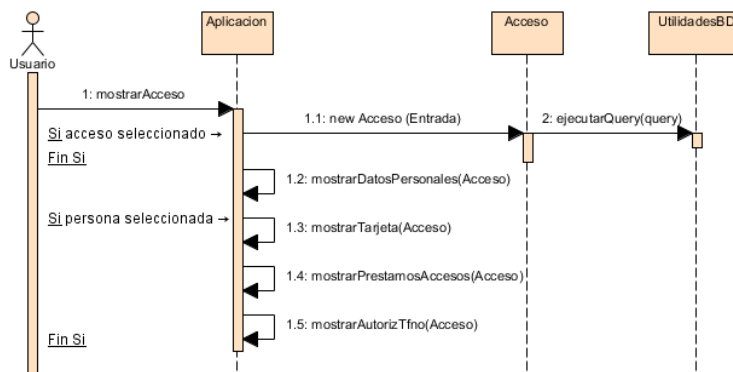


Diagrama de secuencia



4.4. Caso de uso 4: Dar entrada

Utilizaremos la interfaz creada hasta el momento para realizar el registro de accesos. Solo nos faltará ubicar un panel de botones con el que realizar todas las acciones relacionadas con los accesos, préstamos,...

Interfaces gráficas propuestas

Tenemos muy presente que dado el notable número de personas, llaves y llaveros que pueden existir y el buen número de accesos que hay que registrar a lo largo del día, este proceso debe resultar lo más sencillo posible y debe causar la menor posibilidad de incoherencias, fallos,... Por ello haremos que los dos únicos elementos que el vigilante deba indicar (persona y préstamos) sean elementos a seleccionar de listas/tablas, encargándose el sistema de “rellenar” toda la información necesaria. El vigilante seleccionará la persona y préstamos a grabar y el sistema rellenará todo el formulario de datos con los resultados obtenidos de la base de datos.



En esta ocasión seguiremos el siguiente proceso:

1. Realizar la muestra de datos de la persona que va a acceder mediante la selección de la misma y el mismo proceso llevado a cabo en 'Visualizar acceso', obteniéndose todos los datos salvo los de préstamos.
2. Redactar los campos 'Motivo' y 'Otros datos' del acceso.
3. Seleccionar los préstamos de forma análoga a como se seleccionó la persona: en un JDialog.
4. Instar al sistema a que recoja todos los datos del formulario y lleve a la base de datos todos los necesarios.
5. Refrescar los tabs de accesos y préstamos.
6. Avisar al usuario de que el acceso se ha realizado correctamente.

Algunas de las funciones más importantes que surgen son:

MÉTODO	FUNCIONALIDAD
<i>getDNI</i>	Método que obtiene el DNI seleccionado del JDialog de selección de la persona que va a acceder.
<i>getPulsarExit</i>	Para saber cómo se ha salido del JDialog: seleccionando una persona (lo que implica el visionado de sus datos, tarjeta a prestar,...) o pulsando 'Salir' (que no debe realizar acción alguna sobre la GUI).
<i>colorPorDefecto</i>	Método que proporciona el color de la tarjeta que debe ser prestada a una persona.
<i>obtenerResponsable</i>	Método que obtiene el DNI de la persona responsable de otra. En caso de no tener, devuelve la cadena vacía "".
<i>getLsLls</i>	Método que obtiene las llaves y llaveros seleccionados para ser prestados.

Métodos a implementar y a añadir a la API



Este ya empieza a ser un proceso bastante complejo, que recogemos con mucho más detalle en los siguientes diagramas:

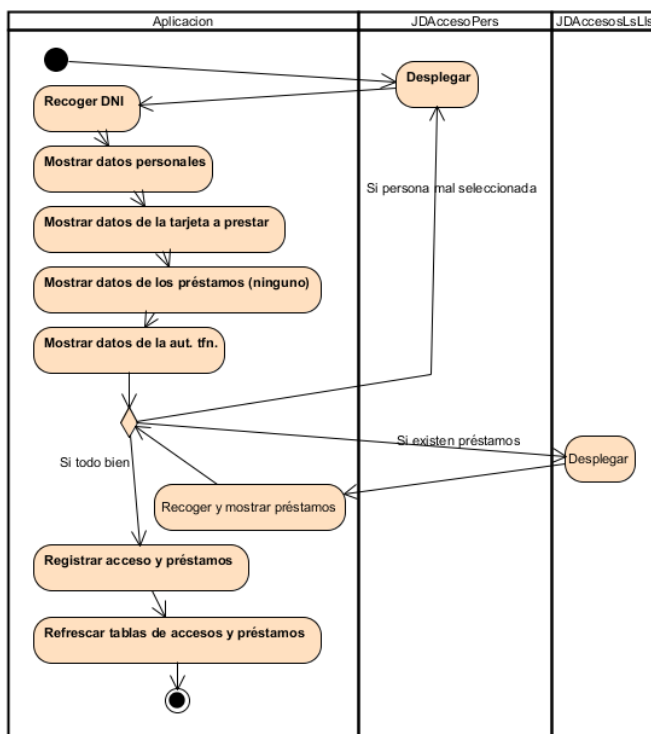


Diagrama de actividades

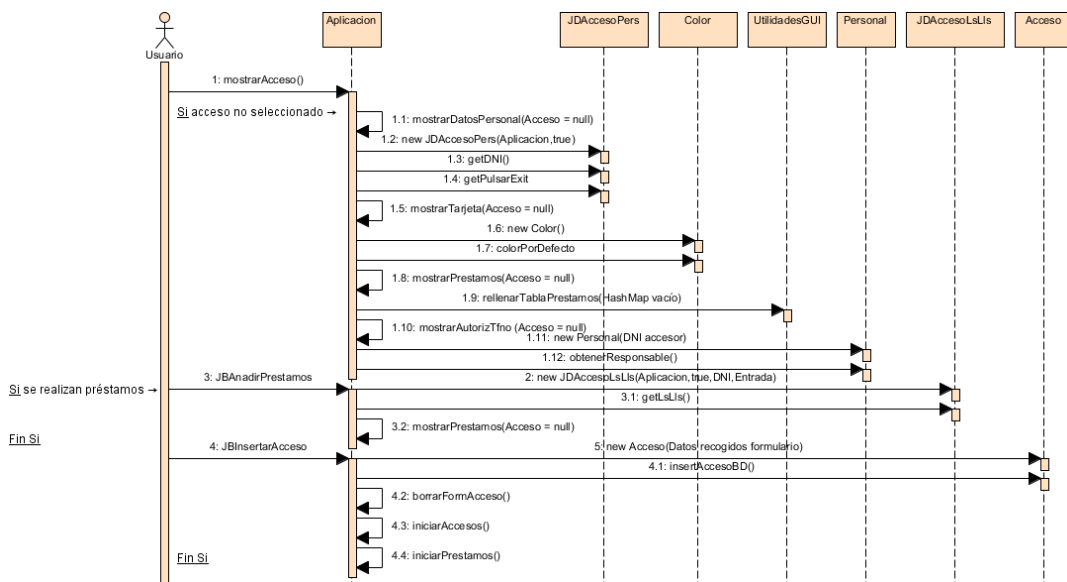


Diagrama de secuencia



4.5. Caso de uso 5: Prestar llaves y/o llaveros

No es necesario añadir ningún componente gráfico a los empleados en el caso de uso anterior.

Lo que debemos hacer es:

1. Seleccionar y obtener el acceso a modificar.
2. Desplegar el JDialog de selección de llaves y llaveros.
3. Añadir las llaves y llaveros seleccionados a los prestados.
4. Refrescar el JTable de préstamos y el tab de préstamos pendientes.

En esta ocasión surge este método:

MÉTODO	FUNCIONALIDAD
<i>insertGLLDsBD</i>	Método que inserta en el acceso los préstamos existentes en el HashMap destinado a tal efecto.

Método a implementar y a añadir a la API

La secuencia de actividades y los mensajes enviados entre los objetos son:

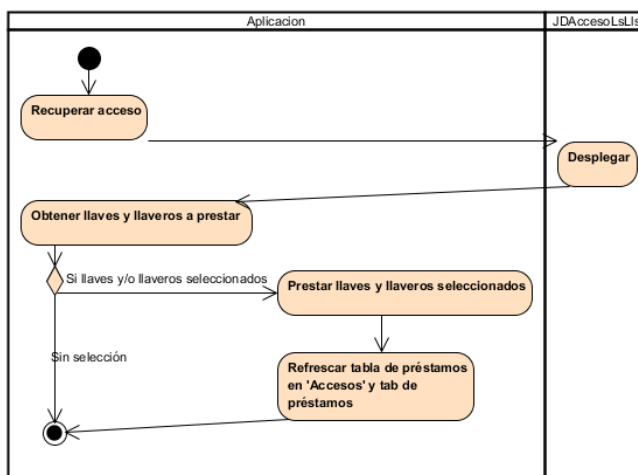


Diagrama de actividades

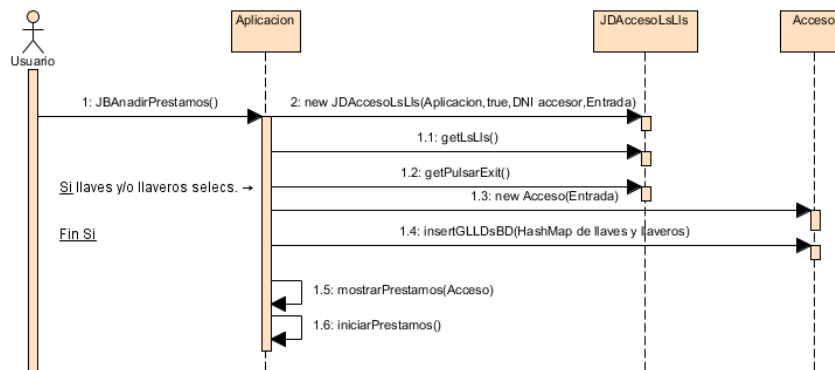


Diagrama de secuencia



4.6. Caso de uso 6: Borrar acceso

En este caso tampoco es necesario añadir ningún componente gráfico ya que todo lo que necesitaremos es la tabla de accesos y un botón de eliminación que añadamos en el panel de botones.

Los pasos a seguir son:

1. Crear un Acceso con su constructor por defecto.
2. Recorrer todos los accesos seleccionados asignando a su atributo Entrada el leído de la tabla y ejecutando el método de borrado.
3. Refrescar los tabs de accesos y préstamos.

El flujo de actividades y mensajes que describen el caso de uso son:

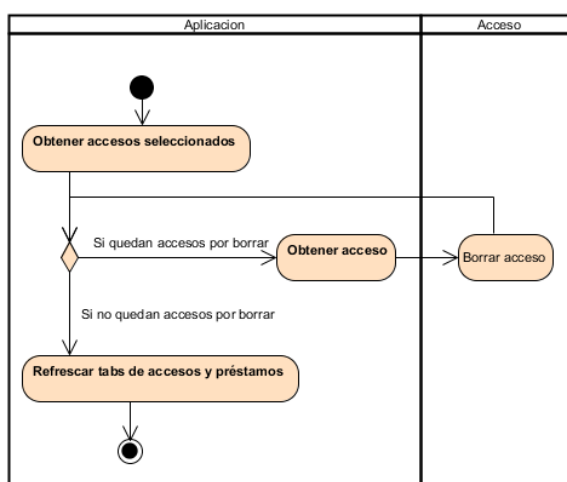


Diagrama de actividades

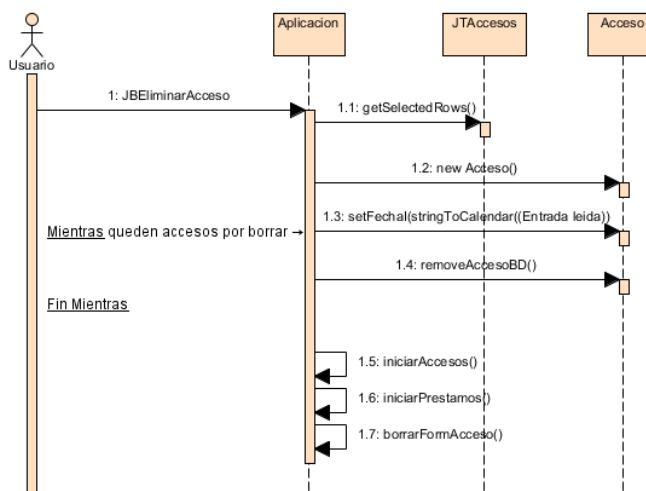


Diagrama de secuencia



4.7. Caso de uso 7: Modificar acceso

La modificación de acceso se realizará sobre el mismo formulario de muestra y el botón para grabar el acceso modificado será el mismo que el de dar acceso, por tanto no necesitamos de nada nuevo.

El proceso seguirá el siguiente flujo de sucesos:

1. Recuperar el acceso seleccionado.
2. Crear el nuevo acceso con los datos existentes en el formulario (ya modificados).
3. Modificar los datos del primer acceso (el nuevo) con los del segundo (el viejo) en la base de datos.
4. Refrescar la tabla de accesos.

Estos son los diagramas de actividades y de secuencia a seguir en la posterior codificación:

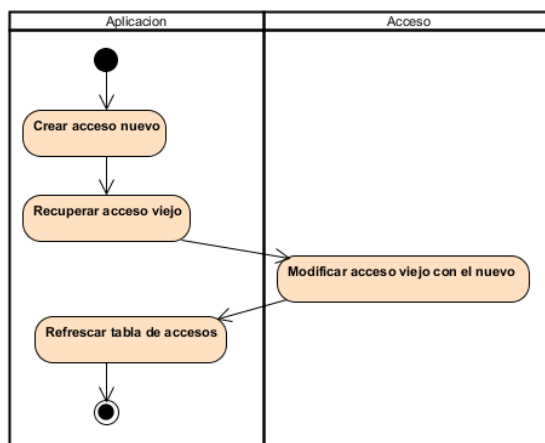


Diagrama de actividades

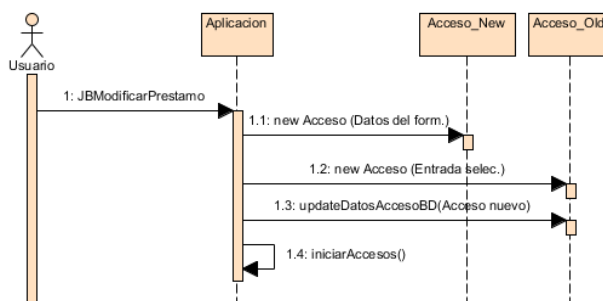


Diagrama de secuencia



4.8. Caso de uso 8: Borrar préstamo

Aquí tampoco añadimos ningún componente a la interfaz gráfica, siendo el proceso de ejecución el siguiente:

1. Recuperar el acceso seleccionado.
2. Obtener cada uno de los préstamos seleccionados en el objeto Acceso, eliminándolos uno a uno.
3. Refrescar la tabla de préstamos con los préstamos existentes para ese acceso en la base de datos y refrescar el tab de préstamos pendientes.

Las actividades a realizar son...

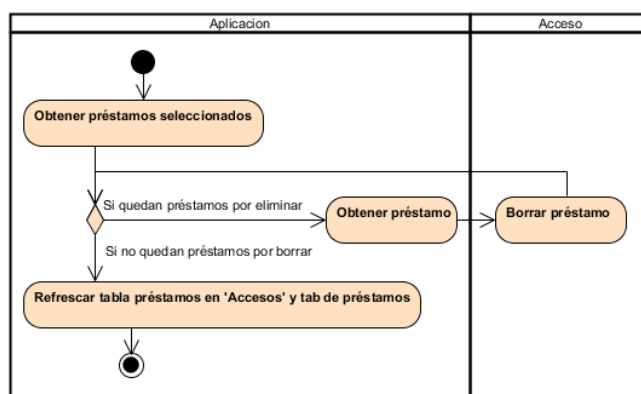


Diagrama de actividades

... y las llamadas que resolverán el caso de uso:

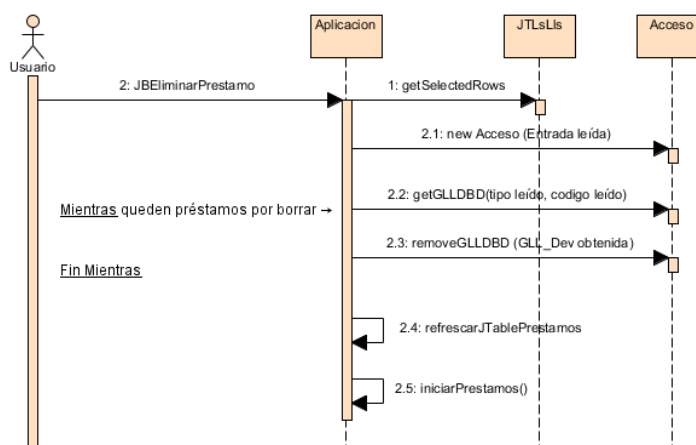


Diagrama de secuencia



4.9. Caso de uso 9: Dar salida

Añadimos dos botones al panel de botones del panel de accesos: uno para dar salida según los préstamos seleccionados y otro que dará salida devolviendo todos los préstamos que la persona posee pendientes en ese acceso.

El caso de uso será guiado de la siguiente forma:

1. Rescatamos el acceso seleccionado.
2. Devolvemos tarjeta, llaves y llaveros que han sido dados de vuelta.
3. Refrescamos la tabla de préstamos.
4. Damos salida al acceso.
5. Refrescamos las tabs de accesos y préstamos.

Surgen:

MÉTODO	FUNCIONALIDAD
<i>devolverLlavesLlaveros</i>	Método que devuelve todas las llaves y llaveros prestados.
<i>devolverLlaves</i>	Método que devuelve todas las llaves prestadas.
<i>devolverLlaveros</i>	Método que devuelve todos los llaveros prestados.

Métodos a implementar y a añadir a la API

El diseño de los diagramas es el que sigue:

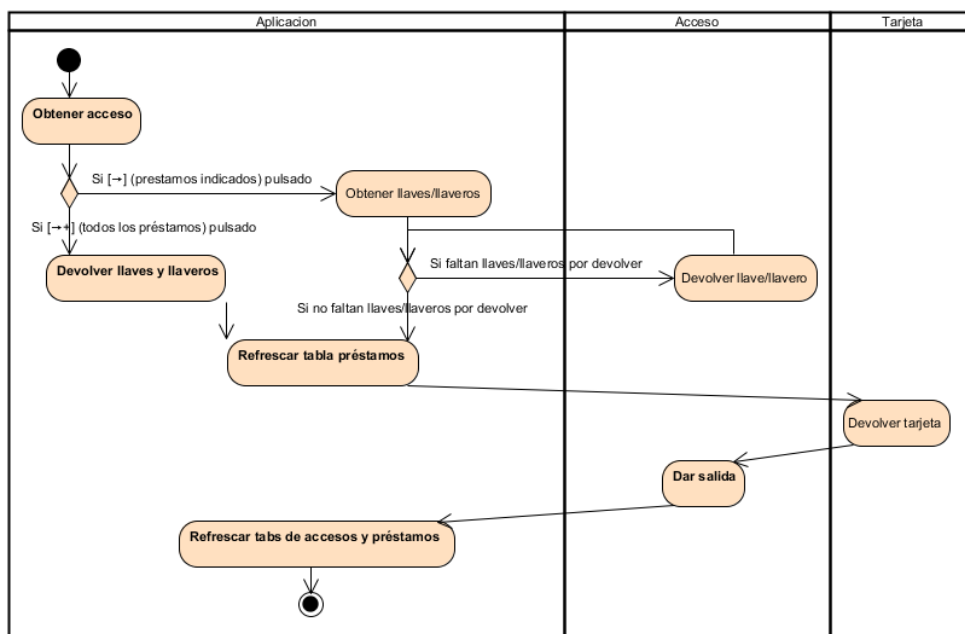


Diagrama de actividades

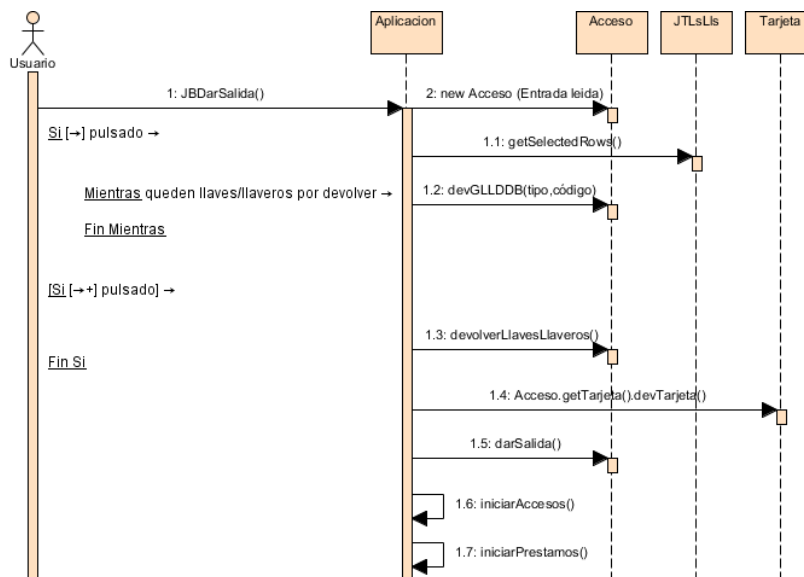


Diagrama de secuencia

4.10. Caso de uso 10: Devolver llaves y llaveros

No tenemos nada que añadir salvo el botón de ‘Devolver préstamos’ y diseñar toda su funcionalidad:

1. Rescatar el acceso seleccionado.
2. Obtener los préstamos seleccionados y marcarlos uno a uno como devueltos.
3. Refrescar el jTable de préstamos y el tab de préstamos pendientes.

No surge nada nuevo que debemos mencionar, ya que este caso de uso puede tomarse como parte del caso de uso anterior. Eso sí, veamos los diagramas creados:

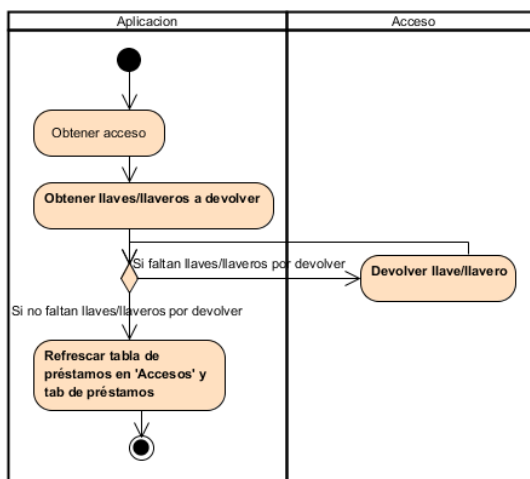


Diagrama de actividades

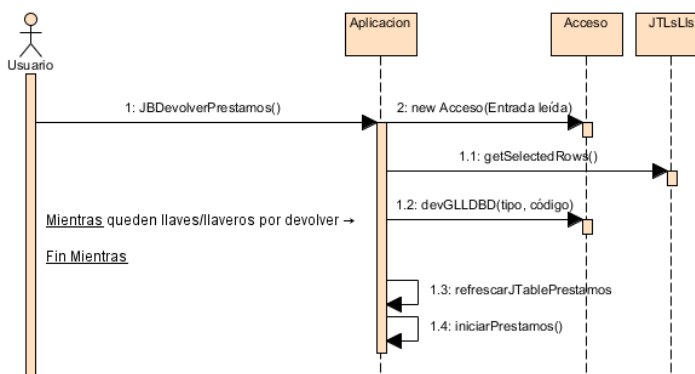
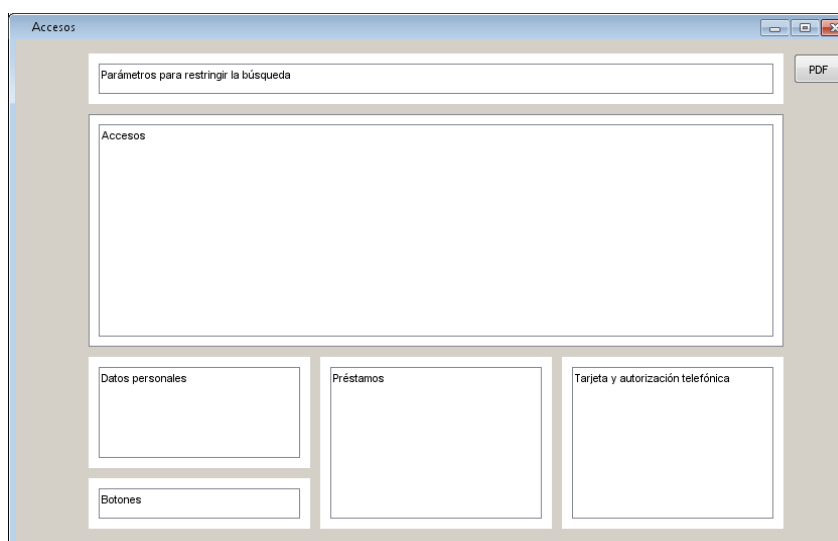


Diagrama de secuencia

4.1.1. Caso de uso 11: Listar accesos

Añadimos un botón a la interfaz generada hasta el momento, de forma que su pulsación inicie todo el caso de uso.



Interfaz gráfica propuesta

El proceso es trivial:

1. Se genera un JFileChooser que permita indicar al usuario la ruta donde crear el reporte.
 - a. Si el usuario elige una ruta
 - Creamos un objeto `GeneradorInformes` con la ruta en la que debe dejar el informe. Le pedimos que lo genere con el título y texto indicados y que extraiga toda la información de la tabla suministrada.
 - El objeto `GeneradorInformes` realiza su tarea de forma satisfactoria y el caso de uso finaliza.



Crearemos un nuevo objeto llamado `GeneradorInformes` que cree informes a partir de los datos que le suministremos mediante los métodos siguientes:

MÉTODO	FUNCIONALIDAD
<i>GeneradorInformes</i>	Constructor que crea un objeto <code>GeneradorInformes</code> tomando como parámetro de entrada la ruta en la que queremos que los almacene.
<i>generarInforme</i>	Método que genera un informe tomando como datos de entrada el título, texto y tabla a reportar.

Métodos a implementar y a añadir a la API

`GeneradorInformes` será implementado mediante la herramienta *iText*. Aportamos una descripción de la clase junto con la de la API final y de la herramienta en el capítulo 4 de la presente memoria, dedicado a las herramientas utilizadas en el desarrollo de esta aplicación.

Como en todos los caso de uso anteriores, formalizamos la secuencia de actividades que se sucederán y el intercambio de mensajes que tendrá lugar entre los objetos a través de sus respectivos diagramas:

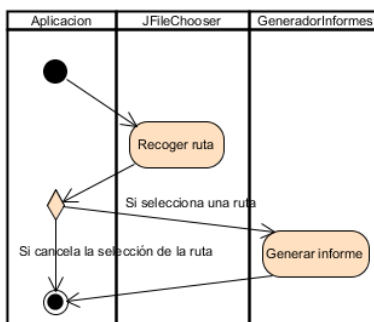


Diagrama de actividades

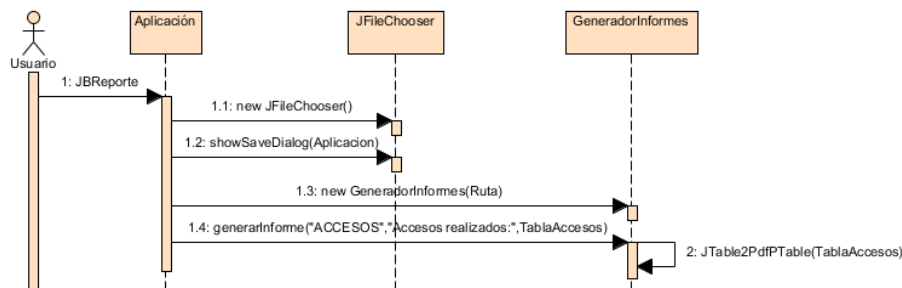


Diagrama de secuencia



4.12. Caso de uso 12: Dar salida automática

La funcionalidad a implementar es de carácter interno, no tiene ninguna relación con la interfaz gráfica. Se trata una tarea periódica programada que debería correr de parte del servidor y no estar implementada en accessManager. De todas formas, para disponer de toda la funcionalidad en un mismo sitio, vamos a desarrollar esta funcionalidad de la siguiente forma:

1. Comprobaremos si se trata de la primera vez que se accede al sistema en el día.
 - a. Si se trata de la primera vez, ejecutaremos la tarea de actualización de la tabla Acceso, que consistirá en:
 - Obtener los accesos que no dispongan de fecha de salida.
 - Recorrerlos uno a uno estableciendo cada salida con la hora máxima hasta la que pudiera haber estado la persona en el centro, dependiendo del tipo de acceso que posea.

Para conseguir realizar este proceso se crean estos cuatro métodos:

MÉTODO	FUNCIONALIDAD
<i>primerVigilante</i>	Método que denota si el usuario que accede a la aplicación es el primer vigilante del día.
<i>mantenerAcceso</i>	Método que realiza la tarea de mantenimiento de la tabla Acceso, dotando de salida a todos los accesos del día anterior que no la tuvieran.
<i>obtenerAccesosSinSalida</i>	Método que obtiene un ArrayList con todos los accesos del día anterior que no disponen de salida.
<i>darSalidaAutomatica</i>	Método que otorga a un acceso de salida según el tipo de acceso del que disponga la persona.

Métodos a implementar y a añadir a la API

La secuencia de ejecución queda de la siguiente forma:

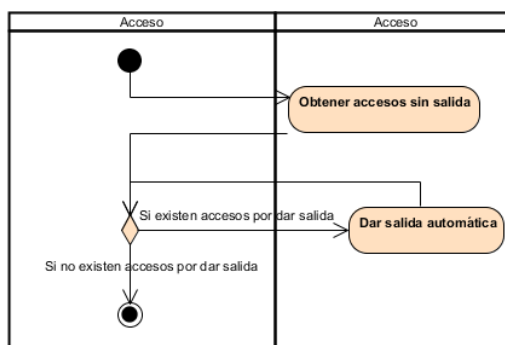


Diagrama de actividades

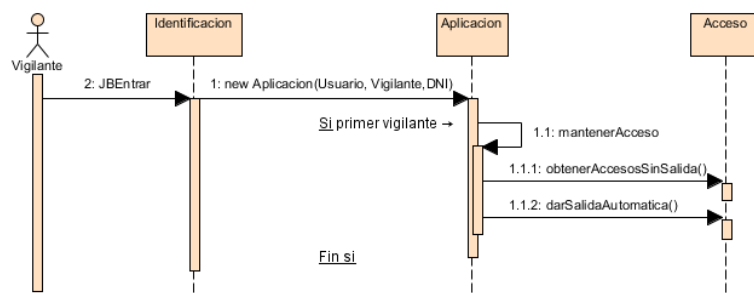


Diagrama de secuencia

5. Codificación e integración

Tras disponer de todos los casos de uso analizados y diseñados, toca armarse de paciencia y codificar todo en las clases que correspondan, así como diseñar las interfaces gráficas necesarias, que hasta donde llevamos desarrollado serán las de logueo y la de accesos al centro.

En cuanto a la integración no podemos comentar muchas cosas de momento, ya que se trata de la única pestaña desarrollada y sus componentes se interrelacionan de manera correcta, como veremos posteriormente. Una vez desarrollemos la pestaña de préstamos sí que analizaremos que ambas pestañas se complementen y sean coherentes entre sí.

5.1. Caso de uso 1: Login



Interfaz de logueo en el sistema



5.2. Casos de uso 2 a 11: Visualizar accesos a Listar accesos

accessManager

accessManager

Accesos

Empresas y personal

Tarjs., cols., nivs. y tipos de acc.

Zonas, llaves y llaveros

Préstamos

Cuentas

Partes de seguridad

Notas

Datos

DNI

Primer apellido

Segundo apellido

Nombre

Motivo

Otros datos

LLaves y llaveros

Tarjeta

Aviso telefónico

Devuelta

Teléfonos

Interfaz de la pestaña 'Accesos'

accessManager

accessManager

Accesos

Empresas y personal

Tarjs., cols., nivs. y tipos de acc.

Zonas, llaves y llaveros

Préstamos

Cuentas

Partes de seguridad

Notas

Datos

DNI

Primer apellido

Segundo apellido

Nombre

Motivo

Otros datos

LLaves y llaveros

Tarjeta

Aviso telefónico

Devuelta

Teléfonos

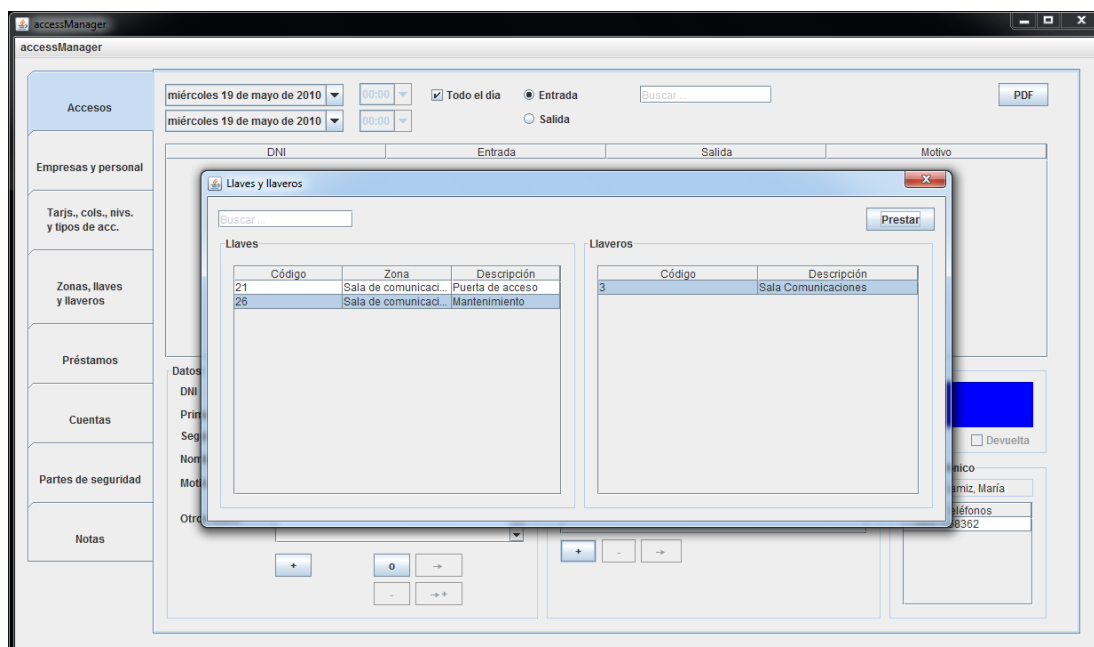
Personal

Buscar

Aceptar

DNI	Primer apellido	Segundo apellido	Nombre
00000157A	Emilia	Durán	María
00000158A	Durán	Quesada	Miriam
00000159A	Reyes	Rios	Patricia
00000161A	Romero	Riveroll	María Teresa
00000162A	Alejandra	Cortez	María
00000163A	García	Pérez	Javier
00000164A	Aldasoro	Ortega	Eduarne
00000165A	Beguristain	Amatriain	Verónica
00000188A	Borde	Membibre	Irene
00000189A	Cestau	Baralbar	Begoña

Interfaz de selección de persona



Interfaz de selección de préstamos

5.3. Casos de uso 12: Dar salida automática

Este caso de uso corre en *background* y no necesita de interfaz gráfica.

6. Pruebas

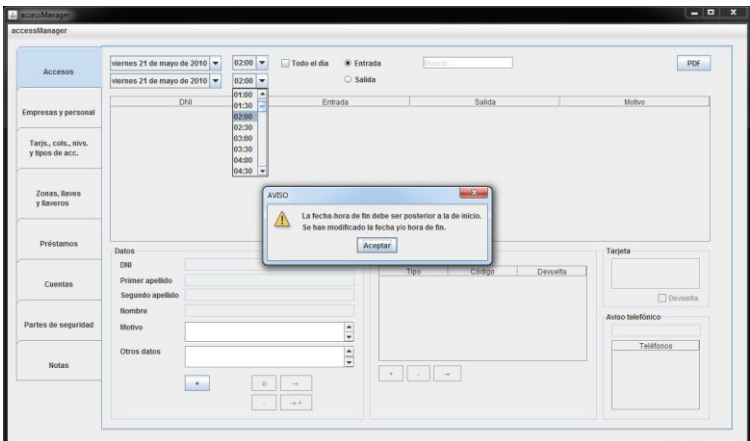
Entramos en una fase fundamental de la iteración: probar que el trabajo realizado es satisfactorio y cumple con nuestras expectativas. A continuación mostramos las pruebas de unidad realizadas, para después pasar a una breve mención de las de integración y sistema.

6.1. De unidad

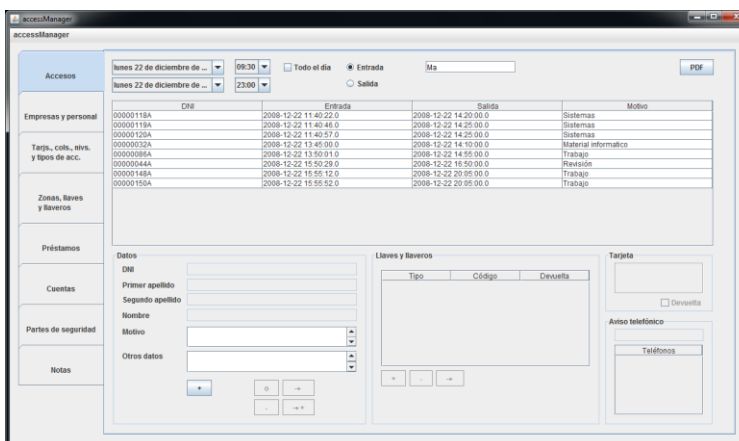
Caso de prueba	3.01 - Caso de uso 1: Login										
Objetivo	Comprobar que el logueo se desarrolla según lo establecido. Arrancar la aplicación y probar las siguientes combinaciones:										
Actividades	<table border="1"> <thead> <tr> <th>Usuario</th> <th>Contraseña</th> </tr> </thead> <tbody> <tr> <td>Válido</td> <td>No válida</td> </tr> <tr> <td>No válido</td> <td>Válida</td> </tr> <tr> <td>No válido</td> <td>No válida</td> </tr> <tr> <td>Válido</td> <td>Válida</td> </tr> </tbody> </table>	Usuario	Contraseña	Válido	No válida	No válido	Válida	No válido	No válida	Válido	Válida
Usuario	Contraseña										
Válido	No válida										
No válido	Válida										
No válido	No válida										
Válido	Válida										

Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. 										
Datos de entrada	<table border="1" data-bbox="746 304 1091 501"> <thead> <tr> <th>Usuario</th> <th>Contraseña</th> </tr> </thead> <tbody> <tr> <td>a</td> <td>b</td> </tr> <tr> <td>b</td> <td>a</td> </tr> <tr> <td>b</td> <td>b</td> </tr> <tr> <td>a</td> <td>a</td> </tr> </tbody> </table>	Usuario	Contraseña	a	b	b	a	b	b	a	a
Usuario	Contraseña										
a	b										
b	a										
b	b										
a	a										
Salida esperada	<p>En los tres primeros casos se espera que la aplicación muestre el mensaje de error y no despliegue el frame principal, mientras que el último sí que realizará esta transición.</p>										
Salida observada	<p>La esperada:</p> <ul style="list-style-type: none"> • En los tres primeros casos: <div data-bbox="564 792 887 909" data-label="Image"> </div> • En el caso de introducir un usuario y contraseña existente se despliega la pantalla principal: <div data-bbox="564 1048 1308 1487" data-label="Image"> </div> 										
Evaluación/ conclusión	<p>El caso de uso '1 - Login' ha sido realizado satisfactoriamente.</p>										

Caso de prueba para el caso de uso '1 - Login'

Caso de prueba	3.02 - Caso de uso 2: Visualizar accesos
Objetivo	Probar que la muestra de accesos al usuario siempre es posible, sean cuales sean los parámetros de búsqueda.
Actividades	<p>Arrancar la aplicación y variar todo lo posible los parámetros de búsqueda:</p> <ul style="list-style-type: none"> • Fechas de entrada y de salida, válidas y erróneas. • Horas de entrada y de salida, válidas y erróneas. • Buscar entradas, buscar salidas. • Buscar con el filtro de texto.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba.
Datos de entrada	Variar todos los parámetros anteriormente apuntados.
Salida esperada	<p>En caso de insertar incorrectamente las fechas y/u horas, éstas deberán restablecerse/reconfigurarse correctamente.</p> <p>En todas y cada una de las posibles situaciones nunca deberán surgir errores, visualizándose en todo momento los accesos que constan en la base de datos y cumplen con los parámetros.</p>
Salida observada	<p>La esperada:</p> <ul style="list-style-type: none"> • En caso de introducirse una fecha y/u hora incorrecta, se avisa del error y se restablecen: 

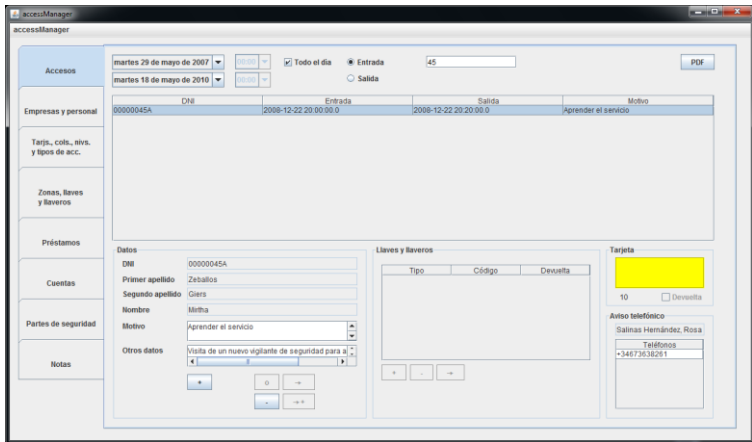
- En el caso de introducir cualquier combinación de parámetros, los resultados mostrados se corresponden con los que deben mostrarse según la información recogida en la base de datos:



**Evaluación/
conclusión**

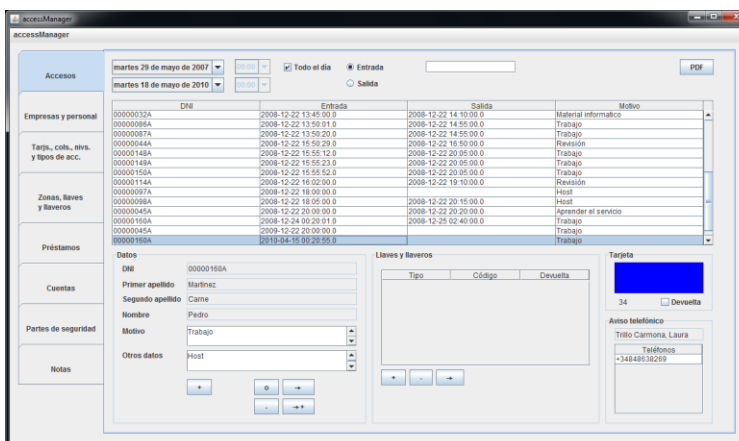
El caso de uso '2 - Visualizar accesos' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso 2: Visualizar accesos

Caso de prueba	3.03 - Caso de uso 3: Visualizar acceso														
Objetivo	Probar que cualquier acceso de cualquier tipo de persona es visionado correctamente por pantalla tras su selección.														
Actividades	<p>Arrancar la aplicación y seleccionar accesos que cumplan las siguientes características:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Tipo de persona</th> <th>Ha salido del centro</th> </tr> </thead> <tbody> <tr> <td>Periódica</td> <td>Sí</td> </tr> <tr> <td>Periódica</td> <td>No</td> </tr> <tr> <td>Ocasional</td> <td>Sí</td> </tr> <tr> <td>Ocasional</td> <td>No</td> </tr> <tr> <td>Residente</td> <td>Sí</td> </tr> <tr> <td>Residente</td> <td>No</td> </tr> </tbody> </table>	Tipo de persona	Ha salido del centro	Periódica	Sí	Periódica	No	Ocasional	Sí	Ocasional	No	Residente	Sí	Residente	No
Tipo de persona	Ha salido del centro														
Periódica	Sí														
Periódica	No														
Ocasional	Sí														
Ocasional	No														
Residente	Sí														
Residente	No														
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. 														
Datos de entrada	Variar todos los parámetros consiguiendo todos los casos anteriormente apuntados.														
Salida esperada	Se deben mostrar los datos de todos los accesos, permaneciendo activados siempre los botones de 'Dar entrada' y 'Eliminar acceso'. Además, en caso de tratarse de un acceso sin salida, los botones de 'Dar salida' y todos los relativos a los préstamos tendrán que estar activados.														
Salida observada	<p>La esperada:</p> <ul style="list-style-type: none"> • En caso de seleccionarse cualquier tipo de persona que ya haya salido del centro:  <p>The screenshot shows the 'accessManager' application interface. At the top, there are filters for dates (May 29, 2007 and May 18, 2010) and a search for 'Todo el día' with 'Entrada' set to 45. Below this is a table with columns for 'Empresas y personal', 'Tarjeta, coils, miv. y tipos de acc.', 'Zonas, Bares y Boveros', 'Préstamos', 'Cuentas', 'Partes de seguridad', and 'Notas'. The main table shows one record with DNI 00000045A, entry date 2008-12-22 20:00:00, and exit date 2008-12-22 20:20:00, with the reason 'Aprender el servicio'. Below the table, there are sections for 'Datos' (DNI, Primer apellido, Segundo apellido, Nombre, Motivo), 'Llaves y Boveros', 'Tarjeta', and 'Aviso telefónico'.</p>														



- En el caso de que la persona aún se encuentre en el centro:

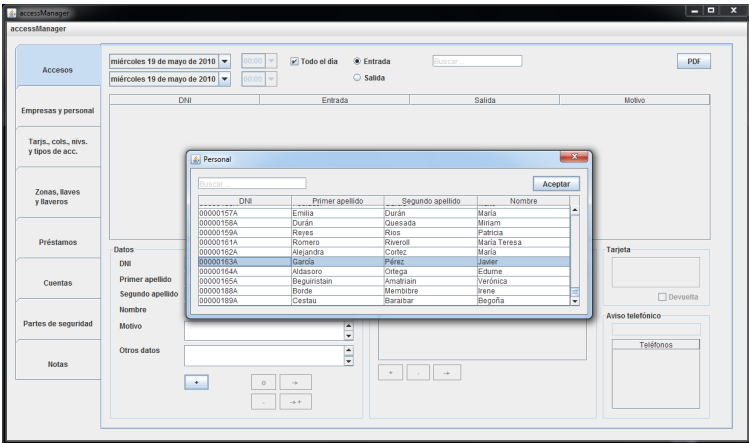


**Evaluación/
conclusión**

El caso de uso '3 - Visualizar acceso' ha sido realizado satisfactoriamente.

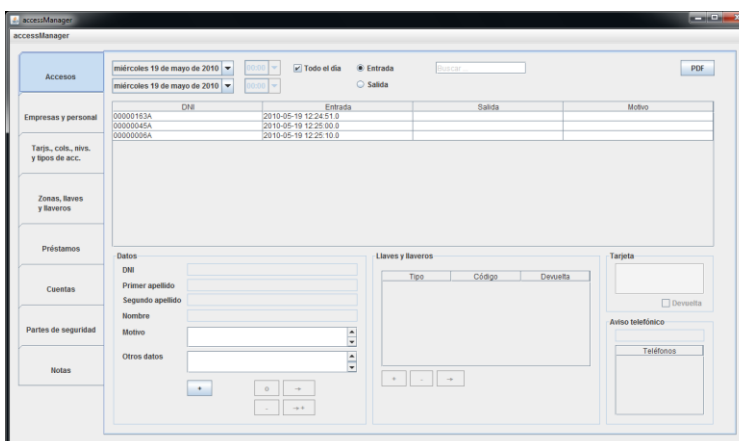
Caso de prueba para el caso de uso '3 - Visualizar acceso'



Caso de prueba	3.04 - Caso de uso 4: Dar entrada														
Objetivo	Probar que el sistema es capaz de registrar entradas de cualquier tipo de persona, así como los préstamos que se le otorguen.														
Actividades	<p>Arrancar la aplicación y probar las siguientes combinaciones:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Tipo de persona</th> <th>Llaves y llaveros prestados</th> </tr> </thead> <tbody> <tr> <td>Periódica</td> <td>Sí</td> </tr> <tr> <td>Periódica</td> <td>No</td> </tr> <tr> <td>Ocasional</td> <td>Sí</td> </tr> <tr> <td>Ocasional</td> <td>No</td> </tr> <tr> <td>Residente</td> <td>Sí</td> </tr> <tr> <td>Residente</td> <td>No</td> </tr> </tbody> </table>	Tipo de persona	Llaves y llaveros prestados	Periódica	Sí	Periódica	No	Ocasional	Sí	Ocasional	No	Residente	Sí	Residente	No
Tipo de persona	Llaves y llaveros prestados														
Periódica	Sí														
Periódica	No														
Ocasional	Sí														
Ocasional	No														
Residente	Sí														
Residente	No														
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. 														
Datos de entrada	<p>Haremos las pruebas con las personas:</p> <ul style="list-style-type: none"> • Periódico: 00000163A • Ocasional: 00000045A • Residente: 00000006A <p>Se probarán a realizar accesos sin préstamos y accesos con préstamos de llaves y llaveros.</p>														
Salida esperada	<p>Todo préstamo deberá quedar registrado en la tabla de Acceso y las tarjetas prestadas deberán adquirir el DNI del prestatario en la tabla Tarjeta.</p> <p>A los accesos con préstamos habrá que añadir el registro en Llave_Prestada y Llaverero_Prestado de todos los préstamos realizados.</p> 														

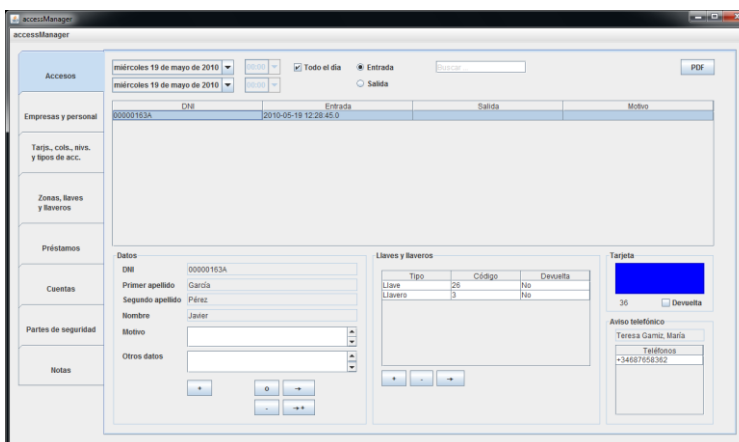
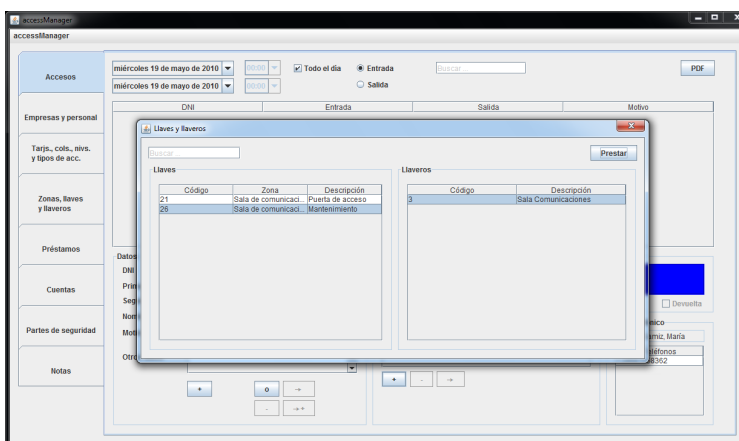
La espera:

- En los accesos sin préstamos:



- Se pueden realizar accesos con préstamos para el personal periódico y ocasional, no mostrándose nunca llaves para el residente:

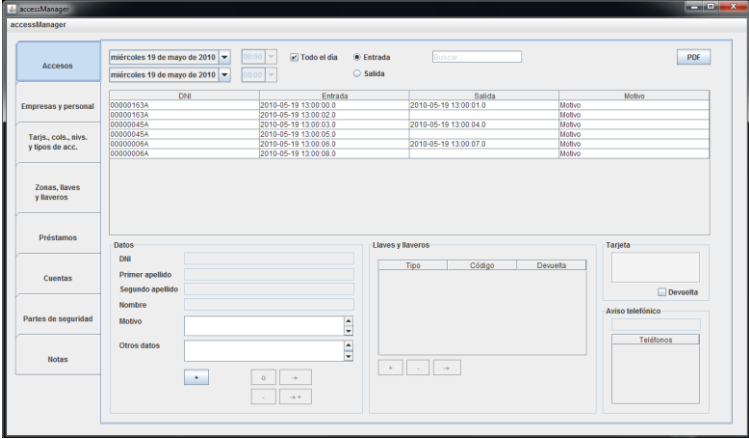
Salida observada



Evaluación/
conclusión

El caso de uso '4 - Dar entrada' ha sido realizado satisfactoriamente.

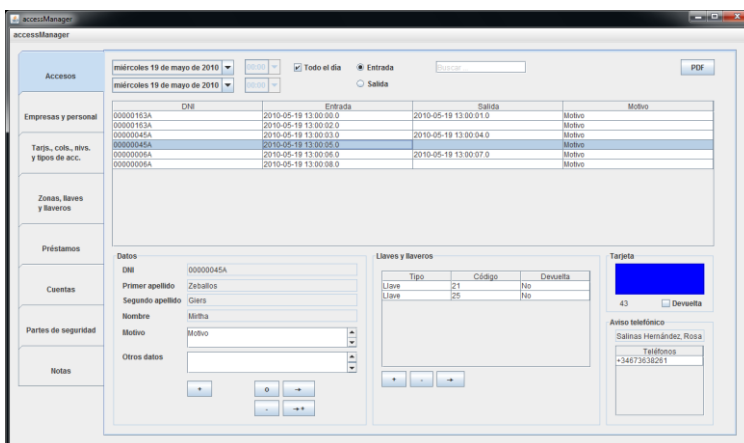
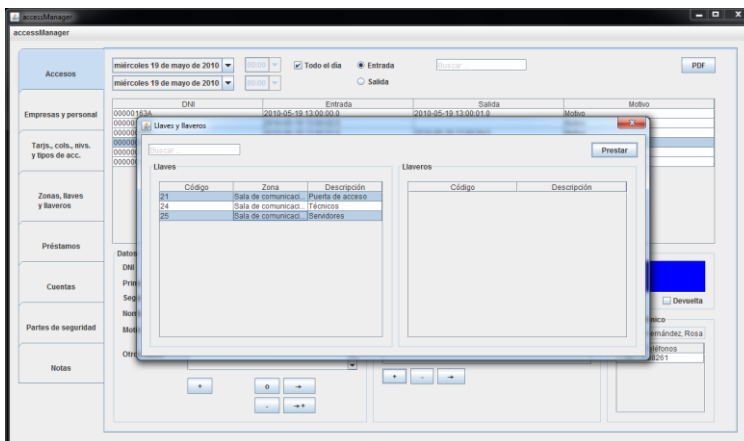
Caso de prueba para el caso de uso '4 - Dar entrada'

Caso de prueba	3.05 - Caso de uso 5: Prestar llaves y/o llaveros														
Objetivo	Chequear que se puede realizar el préstamo de llaves y/o llaveros a cualquier acceso sin salida, y que resulta imposible realizarlo en los que ya la tienen.														
Actividades	<p>Arrancar la aplicación y probar las siguientes combinaciones:</p> <table border="1" data-bbox="746 450 1091 757"> <thead> <tr> <th>Tipo de persona</th> <th>Con salida</th> </tr> </thead> <tbody> <tr> <td>Periódica</td> <td>Sí</td> </tr> <tr> <td>Periódica</td> <td>No</td> </tr> <tr> <td>Ocasional</td> <td>Sí</td> </tr> <tr> <td>Ocasional</td> <td>No</td> </tr> <tr> <td>Residente</td> <td>Sí</td> </tr> <tr> <td>Residente</td> <td>No</td> </tr> </tbody> </table>	Tipo de persona	Con salida	Periódica	Sí	Periódica	No	Ocasional	Sí	Ocasional	No	Residente	Sí	Residente	No
Tipo de persona	Con salida														
Periódica	Sí														
Periódica	No														
Ocasional	Sí														
Ocasional	No														
Residente	Sí														
Residente	No														
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. • Utilidad phpMyAdmin. 														
Datos de entrada	<p>Emplearemos las mismas personas que en el caso de prueba 3.06., creando seis acceso que describen las combinaciones anteriormente citadas. Una vez insertadas en la base de datos las probaremos una a una.</p> <pre data-bbox="564 1088 1246 1205"> Ejecutar la(s) consulta(s) SQL en la base de datos bd: @ INSERT INTO Acceso VALUES ('00000163A','2010-05-19 13:00:00','2010-05-19 13:00:01','Motivo',42,'2010-01-01 00:00:01',null,'00000001A'), ('00000163A','2010-05-19 13:00:02',null,'Motivo',42,'2010-01-01 00:00:01',null,'00000001A'), ('00000045A','2010-05-19 13:00:03','2010-05-19 13:00:04','Motivo',43,'2010-01-01 00:00:01',null,'00000001A'), ('00000045A','2010-05-19 13:00:05',null,'Motivo',43,'2010-01-01 00:00:01',null,'00000001A'), ('00000006A','2010-05-19 13:00:06','2010-05-19 13:00:07','Motivo',44,'2010-01-01 00:00:01',null,'00000001A'), ('00000006A','2010-05-19 13:00:08',null,'Motivo',44,'2010-01-01 00:00:01',null,'00000001A') </pre> 														
Salida esperada	En los accesos que ya disponen de salida no debe ser posible poder prestarles llaves y/o llaveros. En el caso de que la persona aún permanezca en el centro sí que se podrá asignarle nuevas llaves y/o llaveros.														

La espera:

- Si se selecciona un acceso con salida no se le puede añadir ningún préstamo ya que los botones destinados para ello permanecen desactivados.
- Si no poseen salida, los botones están activados y sea cual sea el tipo de persona (periódica u ocasional) se le puede añadir los préstamos que estimemos oportunos:

Salida observada



Evaluación/
conclusión

El caso de uso '5 - Prestar llaves y/o llaveros' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '5 - Prestar llaves y/o llaveros'



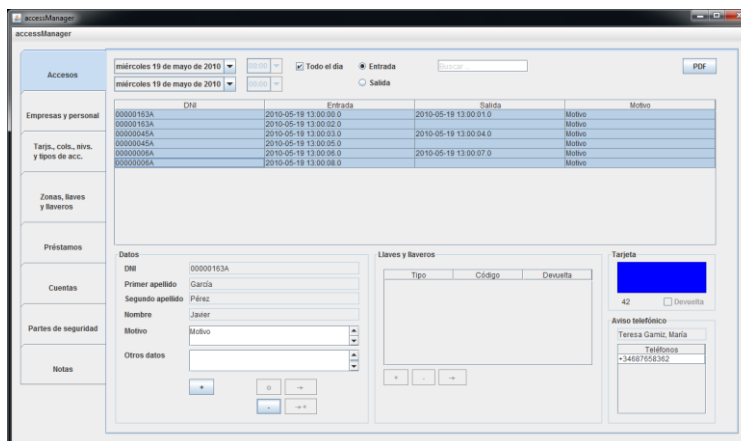
Caso de prueba **3.06 - Caso de uso 6: Borrar acceso**

Objetivo Demostrar que sea cual sea los accesos seleccionados, el usuario podrá eliminarlos si lo estima oportuno.

Actividades Arrancar la aplicación y eliminar los accesos introducidos en el caso de prueba anterior.

- Técnicas y herramientas**
- Aplicación accessManager.
 - Base de datos con población de prueba.
 - Utilidad phpMyAdmin.

Eliminamos los siguientes accesos...



Datos de entrada

... que poseen estos préstamos:

Entrada	Cod_Llave	F_Alta_L	Deuelta
Fecha y hora de acceso	Número de llave prestada	Fecha y hora de inserción de la llave.	Denota si la llave ha sido devuelta
2008-12-22 18:00:00	17	2010-01-01 00:00:01	0
2008-12-22 18:00:00	18	2010-01-01 00:00:01	0
2008-12-22 18:00:00	19	2010-01-01 00:00:01	0
2008-12-22 18:00:00	20	2010-01-01 00:00:01	0
2010-05-19 13:00:02	21	2010-01-01 00:00:01	0
2010-05-19 13:00:02	26	2010-01-01 00:00:01	0
2010-05-19 13:00:05	21	2010-01-01 00:00:01	0
2010-05-19 13:00:05	25	2010-01-01 00:00:01	0

Entrada	Cod_Llavero	F_Alta_LI	Deuelto
Fecha y hora de acceso	Número de llavero prestado	Fecha y hora de inserción del llavero	Denota si el llavero ha sido devuelto
2008-12-22 16:02:00	0	2010-01-01 00:00:01	1
2010-05-19 13:00:02	3	2010-01-01 00:00:01	0

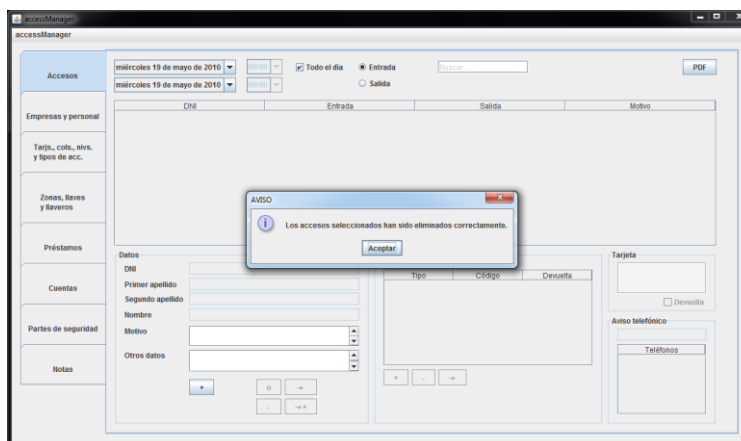
Salida esperada Los accesos seleccionados dejarán de existir, así como los préstamos que se realizaron en ellos.

La esperada: los accesos y sus correspondientes préstamos desaparecen:

Salida observada

Entrada	Cod_Llave	F_Alta_L	Deuelta
Fecha y hora de acceso	Número de llave prestada	Fecha y hora de inserción de la llave.	Denota si la llave ha sido devuelta
2008-12-22 18:00:00	17	2010-01-01 00:00:01	0
2008-12-22 18:00:00	18	2010-01-01 00:00:01	0
2008-12-22 18:00:00	19	2010-01-01 00:00:01	0
2008-12-22 18:00:00	20	2010-01-01 00:00:01	0

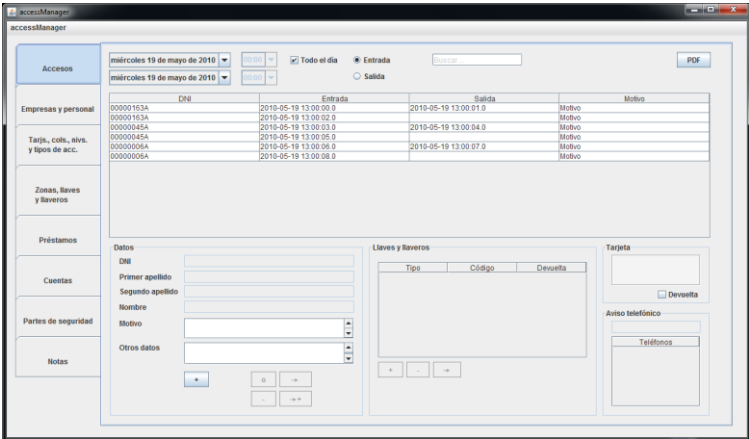
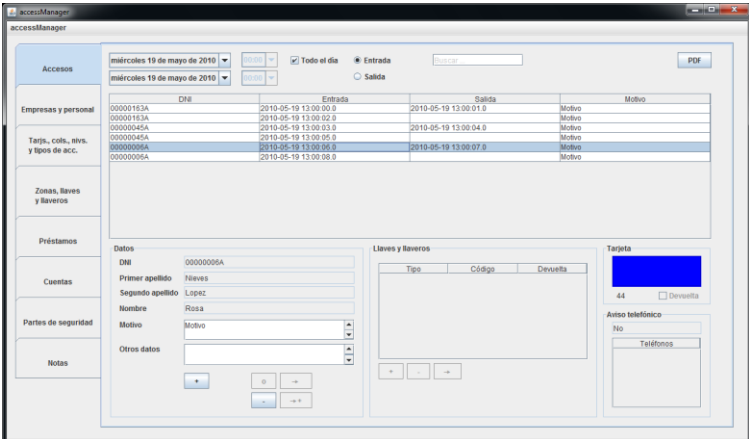
Entrada	Cod_Llavero	F_Alta_LI	Deuelto
Fecha y hora de acceso	Número de llavero prestado	Fecha y hora de inserción del llavero	Denota si el llavero ha sido devuelto
2008-12-22 16:02:00	0	2010-01-01 00:00:01	1



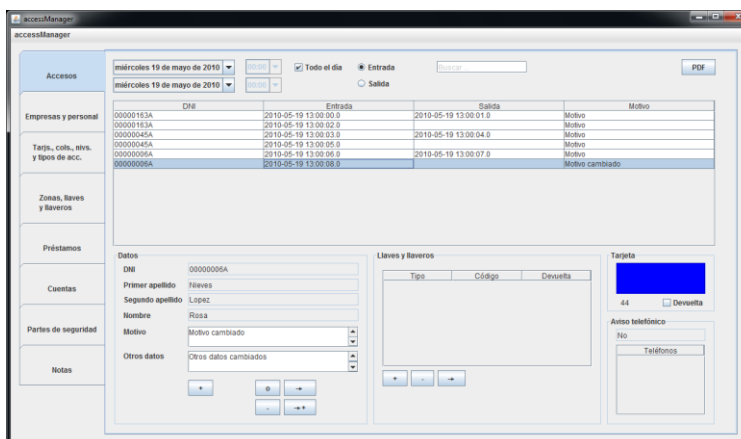
**Evaluación/
conclusión**

El caso de uso '6 - Borrar acceso' ha sido realizado satisfactoriamente.

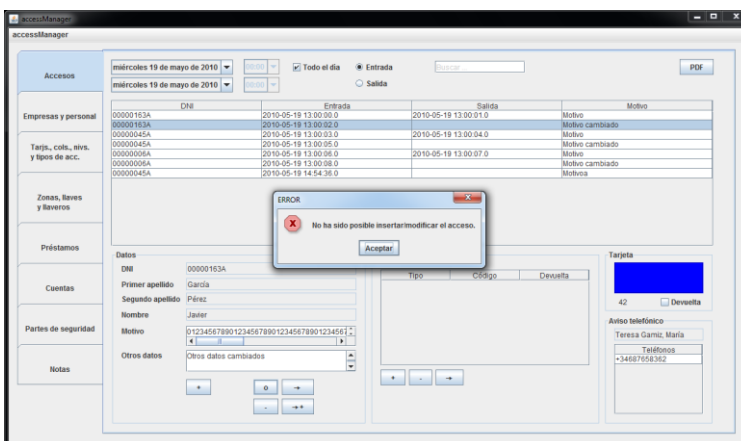
Caso de prueba para el caso de uso '6 - Borrar acceso'

Caso de prueba	3.07 - Caso de uso 7: Modificar acceso
Objetivo	Comprobar que los atributos Motivo y Otros datos de todo acceso sin salida pueden ser modificados.
Actividades	Arrancar la aplicación, introducir los datos de entrada del caso de prueba 5 y modificar los campos indicados con datos correctos y erróneos (formato no permitido).
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba.
Datos de entrada	<p>Los mismos que el quinto caso de prueba:</p> 
Salida esperada	<p>En el caso de introducir valores válidos las modificaciones tendrán lugar; en caso contrario saltará la excepción <i>DataException</i>, se avisará del error y no se modificará ningún dato.</p>
Salida observada	<p>La esperada:</p> <ul style="list-style-type: none"> • En caso de querer modificar un acceso con salida el sistema no lo permite: 

- Es posible modificar un acceso sin salida:
En caso de introducir valores válidos se lleva a cabo la modificación:



En caso de que sean erróneos, no se realiza ninguna acción:



**Evaluación/
conclusión**

El caso de uso '7 - Modificar acceso' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '7 - Modificar acceso'



Caso de prueba	3.08 - Caso de uso 8: Borrar préstamo
Objetivo	Probar que los accesos que no disponen de fecha de salida pueden verse modificados mediante la eliminación de parte de sus préstamos.
Actividades	Arrancar la aplicación, cargar los accesos de los casos de prueba del 5 al 7, añadirles préstamos y eliminarlos comprobando que el sistema reacciona según lo establecido.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos con población de prueba.

Los mismos que en cualquier de los tres últimos casos de prueba ejecutados, junto con los siguientes préstamos:

Datos de entrada

Empresas y personal	DNI	Entrada	Salida	Motivo
00000163A	00000163A	2010-05-19 13:00:00.0	2010-05-19 13:00:01.0	Motivo
00000163A	00000163A	2010-05-19 13:00:02.0		Motivo
00000045A	00000045A	2010-05-19 13:00:03.0	2010-05-19 13:00:04.0	Motivo
00000045A	00000045A	2010-05-19 13:00:05.0		Motivo
0000006A	0000006A	2010-05-19 13:00:06.0	2010-05-19 13:00:07.0	Motivo
0000006A	0000006A	2010-05-19 13:00:08.0		Motivo

Form fields for adding loans:

- DNI: 00000163A
- Primer apellido: Garcia
- Segundo apellido: Pérez
- Nombre: Javier
- Motivo: Motivo
- Tipos de llaves: Llave 21 (No), Llave 25 (No), Llave 26 (No)
- Tarjeta: 42, Teresa Garcia, María, Teléfono: +3458765312

Empresas y personal	DNI	Entrada	Salida	Motivo
00000163A	00000163A	2010-05-19 13:00:09.0	2010-05-19 13:00:10.0	Motivo
00000163A	00000163A	2010-05-19 13:00:12.0		Motivo
00000045A	00000045A	2010-05-19 13:00:13.0	2010-05-19 13:00:14.0	Motivo
00000045A	00000045A	2010-05-19 13:00:15.0		Motivo
0000006A	0000006A	2010-05-19 13:00:16.0	2010-05-19 13:00:17.0	Motivo
0000006A	0000006A	2010-05-19 13:00:18.0		Motivo

Form fields for adding loans:

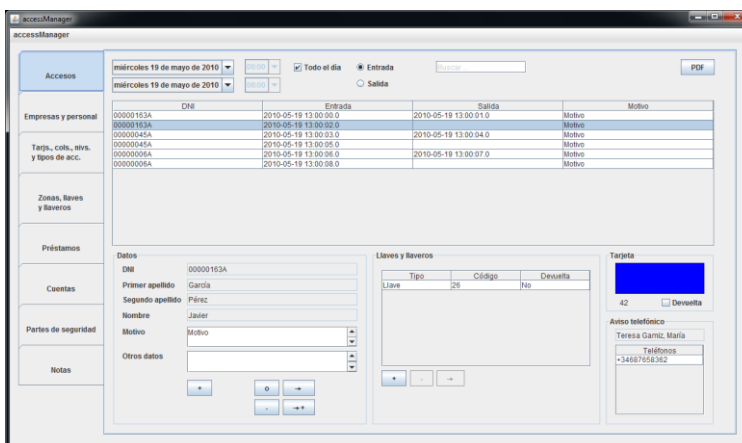
- DNI: 00000045A
- Primer apellido: Zaballos
- Segundo apellido: Giers
- Nombre: Mitza
- Motivo: Motivo
- Tipos de llaves: Llave 24 (No), Llave 25 (No), Llave 26 (No)
- Tarjeta: 43, Salinas Hernández, Rosa, Teléfono: +3457353251

Salida esperada

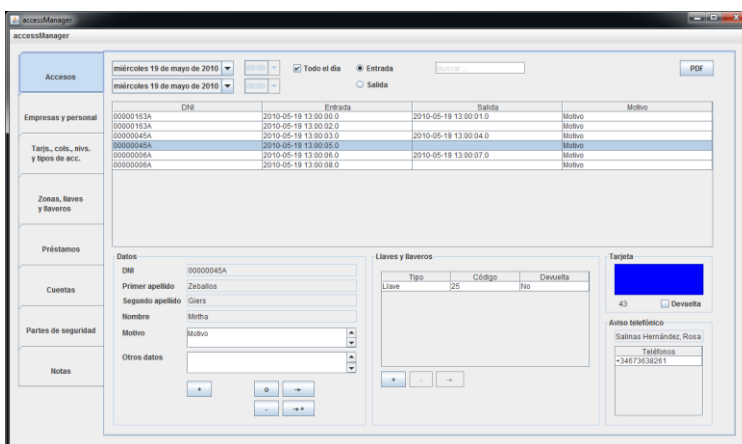
Tras borrar el primer y tercer préstamo del primer acceso y el primer préstamo del segundo, ambos accesos deberían poseer en préstamo únicamente las llaves 26 y 25 respectivamente.



La esperada:



Salida observada



Evaluación/
conclusión

El caso de uso '8 - Borrar préstamo' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '8 - Borrar préstamo'



Caso de prueba	3.09 - Caso de uso 9: Dar salida
Objetivo	Verificar que la aplicación permite dar salida a cualquier tipo de acceso devolviendo los préstamos que este posea de forma parcial o total.
Actividades	Arrancar la aplicación, cargar los accesos que se indican y dotarlos de préstamos. Acto seguido, dar salida a todos ellos probando que la salida con devolución parcial y total se realiza correctamente.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. • Utilidad phpMyAdmin.

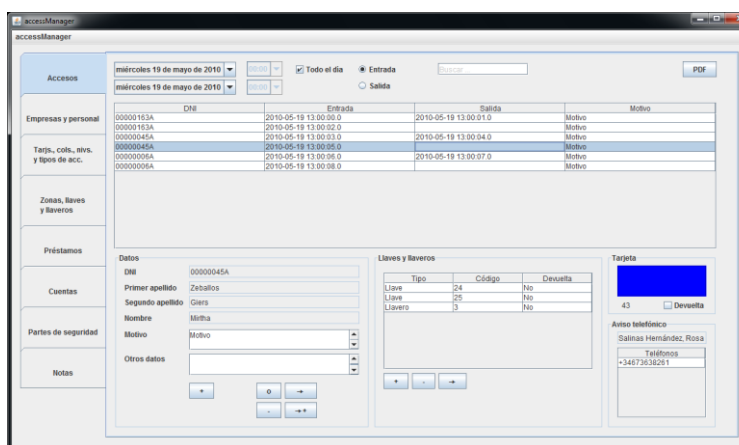
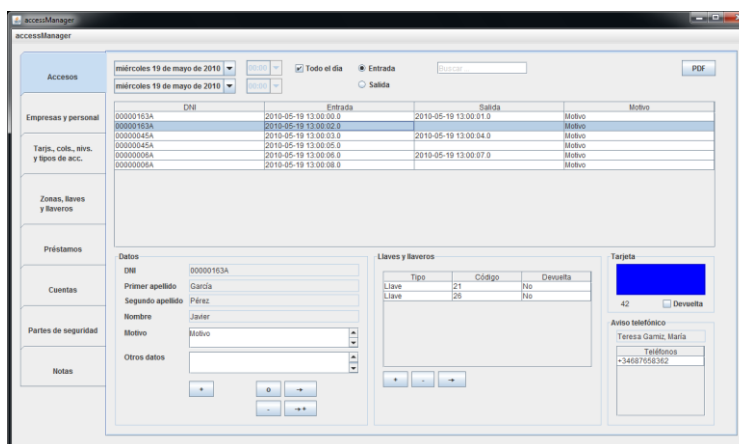
Los siguientes accesos y préstamos:

```

Ejecutar la(s) consulta(s) SQL en la base de datos bd: @
INSERT INTO Acceso VALUES
('00000163A','2010-05-19 13:00:00','2010-05-19 13:00:01','Motivo',42,'2010-01-01
00:00:01',null,'00000001A'),
('00000163A','2010-05-19 13:00:02',null,'Motivo',42,'2010-01-01 00:00:01',null,'00000001A'),
('0000045A','2010-05-19 13:00:03','2010-05-19 13:00:04','Motivo',43,'2010-01-01
00:00:01',null,'00000001A'),
('0000045A','2010-05-19 13:00:05',null,'Motivo',43,'2010-01-01 00:00:01',null,'00000001A')

```

Datos de entrada



Daremos salida a todos ellos, dotando de salida parcial al segundo y de total al cuarto.

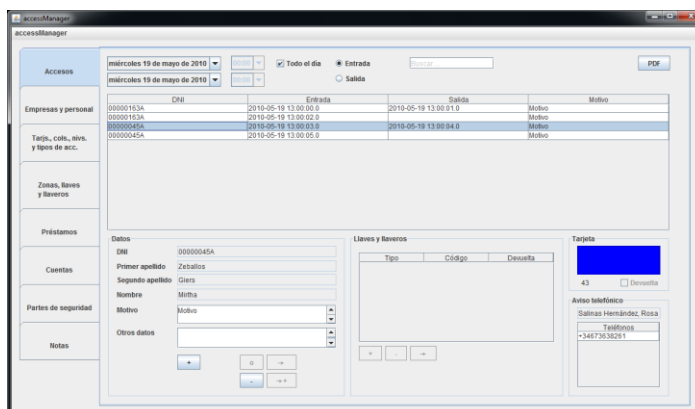
Salida esperada

El acceso con devolución parcial tendrá como devueltas las llaves y llaveros que escogimos a la hora de proporcionar salida, mientras que el que realizamos de forma total tendrá todos sus préstamos marcados como devueltos.



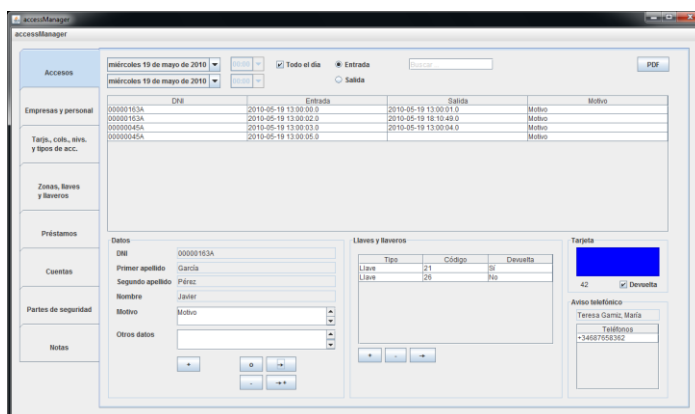
La espera:

- A aquellos que disponen de salida no les podemos volver a dársela:

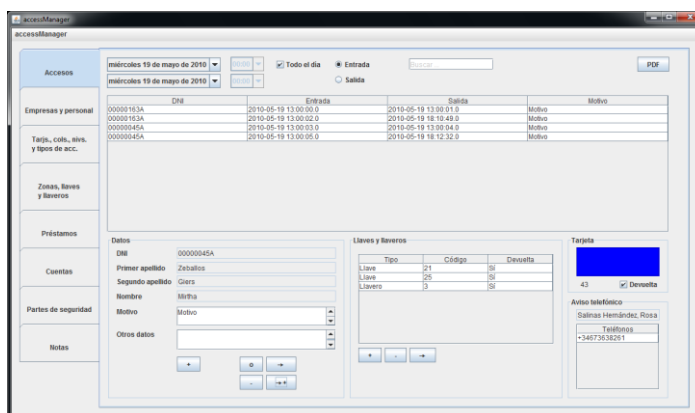


- Aquellos que no disponen de ella podemos dársela devolviendo parcial...

Salida observada



... o totalmente los préstamos:

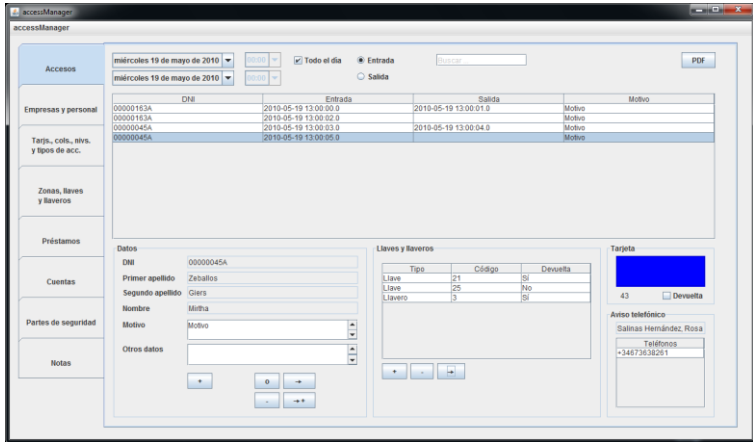
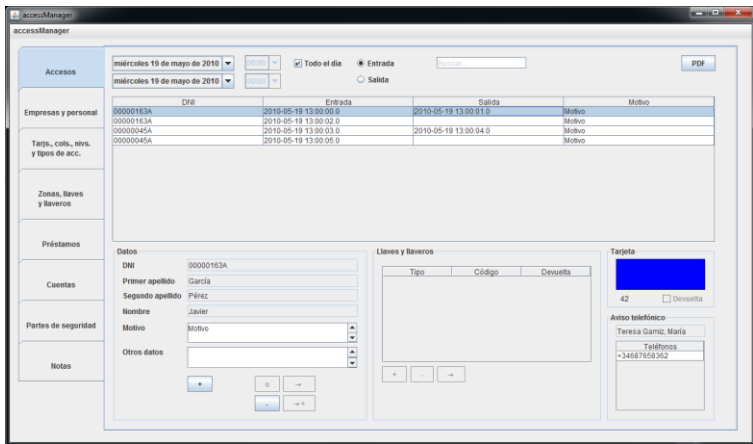


Evaluación/
conclusión

El caso de uso '9 - Dar salida' ha sido realizado satisfactoriamente.


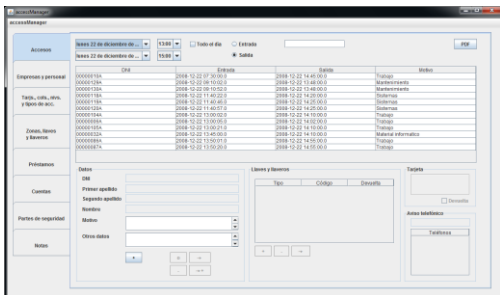
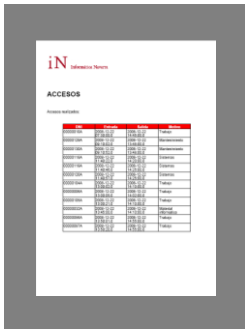
Caso de prueba para el caso de uso '9 - Dar salida'



Caso de prueba	3.10 - Caso de uso 10: Devolver llaves y/o llaveros
Objetivo	Probar que el usuario puede devolver las llaves y llaveros de cualquier préstamo que no posea salida.
Actividades	Arrancar la aplicación, cargar los accesos y dotarlos de préstamos. Después probaremos a intentar dar de baja préstamos de accesos que constan de salida, obteniendo una respuesta negativa, y comprobaremos que sí podemos marcar como devueltos los préstamos de accesos que no tienen salida.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba.
Datos de entrada	Los accesos y préstamos del caso de uso anterior.
Salida esperada	Debe resultar imposible marcar préstamos como devueltos en accesos con salida, y al contrario, debe ser posible marcar préstamos como devueltos en accesos sin salida. Probaremos las mismas tuplas que en el caso de uso '8 - Borrar préstamo', solo que en vez de eliminar los préstamos deberán quedar marcados como devueltos.
Salida observada	<p>La esperada:</p>  
Evaluación/ conclusión	El caso de uso '10 - Devolver llaves y/o llaveros' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '10 - Devolver llaves y/o llaveros'

accessManager · Iteración C3: Accesos · Página 52 de 55

Caso de prueba	3.11 - Caso de uso 11: Listar accesos
Objetivo	Verificar que la aplicación produce listados de los accesos visualizados sea cual sea el estado de la tabla de accesos.
Actividades	Arrancar la aplicación y probar que se generan listados con la tabla de accesos vacía y llena.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba.
Datos de entrada	<ul style="list-style-type: none"> • Tabla de accesos vacía. • Tabla de accesos en la que constan accesos.
Salida esperada	En el primer caso un informe que indique que no hay datos y en el segundo un informe con todos los datos de la tabla.
Salida observada	<p>La esperada:</p> <ul style="list-style-type: none"> • En el primer caso: <div style="display: flex; justify-content: space-around;">   </div> <ul style="list-style-type: none"> • En el caso segundo caso: <div style="display: flex; justify-content: space-around;">   </div>
Evaluación/ conclusión	El caso de uso '11 - Listar accesos' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '11 - Listar accesos'



Caso de prueba	3.12 - Caso de uso 12: Dar acceso automático																																																																
Objetivo	Comprobar que el sistema realiza el mantenimiento de la tabla Acceso una vez al día, otorgando salida a todos aquellos accesos realizados en el día anterior y que no disponen de ella.																																																																
Actividades	Introducir tres accesos del día anterior sin fecha de salida, borrar la población de la tabla que recoge el historial de accesos al sistema (Acceso_Sistema) y loguear como vigilante, verificando que se ha asignado salida a los accesos de prueba.																																																																
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. • Utilidad phpMyAdmin. 																																																																
Datos de entrada	<ul style="list-style-type: none"> • Accesos de prueba, uno por cada tipo de persona: <pre> Ejecutar la(s) consulta(s) SQL en la base de datos bd: @ _____ INSERT INTO Acceso VALUES ('00000160A', '2010-05-18 16:00:00', null, 'Motivo', 32, '2010-01-01 00:00:01', null, '00000001A'); INSERT INTO Acceso VALUES ('00000045A', '2010-05-18 16:00:01', null, 'Motivo', 33, '2010-01-01 00:00:01', null, '00000001A'); INSERT INTO Acceso VALUES ('00000006A', '2010-05-18 16:00:02', null, 'Motivo', 34, '2010-01-01 00:00:01', null, '00000001A'); </pre>																																																																
Salida esperada	<p>Los accesos insertados deben tener las siguientes horas de salida:</p> <ul style="list-style-type: none"> • 23:59:59 • 21:00:00 • 23:59:59 																																																																
Salida observada	<p>La esperada:</p> <table border="1"> <thead> <tr> <th>DNI DNI de la persona que accede</th> <th>Entrada Fecha y hora de entrada</th> <th>Salida Fecha y hora de salida</th> <th>Motivo Motivo de acceso</th> <th>Cod_Tarj Número de tarjeta empleada</th> <th>F_Alta_T Fecha y hora de inserción de la tarjeta</th> <th>Otros_Datos Otros datos de interés</th> <th>DNI_Ins DNI del usuario que realiza la inserción del acceso (por lo general el vigilante de seguridad)</th> </tr> </thead> <tbody> <tr> <td>00000045A</td> <td>2010-05-19 13:00:05</td> <td>NULL</td> <td>Motivo</td> <td>43</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>00000001A</td> </tr> <tr> <td>00000045A</td> <td>2010-05-19 13:00:03</td> <td>2010-05-19 13:00:04</td> <td>Motivo</td> <td>43</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>00000001A</td> </tr> <tr> <td>00000163A</td> <td>2010-05-19 13:00:02</td> <td>NULL</td> <td>Motivo</td> <td>42</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>00000001A</td> </tr> <tr> <td>00000163A</td> <td>2010-05-19 13:00:00</td> <td>2010-05-19 13:00:01</td> <td>Motivo</td> <td>42</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>00000001A</td> </tr> <tr> <td>00000006A</td> <td>2010-05-18 16:00:02</td> <td>2010-05-18 23:59:59</td> <td>Motivo</td> <td>34</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>00000001A</td> </tr> <tr> <td>00000045A</td> <td>2010-05-18 16:00:01</td> <td>2010-05-18 21:00:00</td> <td>Motivo</td> <td>33</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>00000001A</td> </tr> <tr> <td>00000160A</td> <td>2010-05-18 16:00:00</td> <td>2010-05-18 23:59:59</td> <td>Motivo</td> <td>32</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>00000001A</td> </tr> </tbody> </table>	DNI DNI de la persona que accede	Entrada Fecha y hora de entrada	Salida Fecha y hora de salida	Motivo Motivo de acceso	Cod_Tarj Número de tarjeta empleada	F_Alta_T Fecha y hora de inserción de la tarjeta	Otros_Datos Otros datos de interés	DNI_Ins DNI del usuario que realiza la inserción del acceso (por lo general el vigilante de seguridad)	00000045A	2010-05-19 13:00:05	NULL	Motivo	43	2010-01-01 00:00:01	NULL	00000001A	00000045A	2010-05-19 13:00:03	2010-05-19 13:00:04	Motivo	43	2010-01-01 00:00:01	NULL	00000001A	00000163A	2010-05-19 13:00:02	NULL	Motivo	42	2010-01-01 00:00:01	NULL	00000001A	00000163A	2010-05-19 13:00:00	2010-05-19 13:00:01	Motivo	42	2010-01-01 00:00:01	NULL	00000001A	00000006A	2010-05-18 16:00:02	2010-05-18 23:59:59	Motivo	34	2010-01-01 00:00:01	NULL	00000001A	00000045A	2010-05-18 16:00:01	2010-05-18 21:00:00	Motivo	33	2010-01-01 00:00:01	NULL	00000001A	00000160A	2010-05-18 16:00:00	2010-05-18 23:59:59	Motivo	32	2010-01-01 00:00:01	NULL	00000001A
DNI DNI de la persona que accede	Entrada Fecha y hora de entrada	Salida Fecha y hora de salida	Motivo Motivo de acceso	Cod_Tarj Número de tarjeta empleada	F_Alta_T Fecha y hora de inserción de la tarjeta	Otros_Datos Otros datos de interés	DNI_Ins DNI del usuario que realiza la inserción del acceso (por lo general el vigilante de seguridad)																																																										
00000045A	2010-05-19 13:00:05	NULL	Motivo	43	2010-01-01 00:00:01	NULL	00000001A																																																										
00000045A	2010-05-19 13:00:03	2010-05-19 13:00:04	Motivo	43	2010-01-01 00:00:01	NULL	00000001A																																																										
00000163A	2010-05-19 13:00:02	NULL	Motivo	42	2010-01-01 00:00:01	NULL	00000001A																																																										
00000163A	2010-05-19 13:00:00	2010-05-19 13:00:01	Motivo	42	2010-01-01 00:00:01	NULL	00000001A																																																										
00000006A	2010-05-18 16:00:02	2010-05-18 23:59:59	Motivo	34	2010-01-01 00:00:01	NULL	00000001A																																																										
00000045A	2010-05-18 16:00:01	2010-05-18 21:00:00	Motivo	33	2010-01-01 00:00:01	NULL	00000001A																																																										
00000160A	2010-05-18 16:00:00	2010-05-18 23:59:59	Motivo	32	2010-01-01 00:00:01	NULL	00000001A																																																										
Evaluación/ conclusión	El caso de uso '12 - Dar acceso automático' ha sido realizado satisfactoriamente.																																																																

Caso de prueba para el caso de uso '12 - Dar acceso automático'

6.2. De integración

Como ya mencionamos en la anterior sección, al encontrarnos en una fase tan temprana del desarrollo no tenemos elementos suficientes tales que puedan dar lugar a cualquier problema de integración. Todos los componentes de la pestaña 'Accesos' se relacionan de forma coherente, y las capas GUI-API-BD interactúan a la perfección ya que, como hemos podido observar, los casos de uso son ejecutados de forma correcta, según lo que teníamos previsto, con que no observamos fallo alguno.



6.3. De sistema

Llegados al final de la iteración debemos hacer una breve recapitulación, comprobando que hemos implementado todos los casos de uso que nos propusimos al inicio de la misma, que todo funciona correctamente y que la documentación que hemos elaborado es la apropiada.

Los dos primeros puntos surgidos quedan respondidos por la sección 5 de este artefacto. En cuanto al último, hemos desarrollado este documento que creemos que documenta toda la parte esencial del análisis, diseño, codificación y pruebas que hemos realizado. No creemos conveniente ahondar más en todo lo comentado, pues lo creemos suficiente para que cualquier persona ajena al proyecto pueda llegar a comprender todo el trabajo realizado en esta iteración. De todos modos, ante cualquier duda que pueda surgir, el lector siempre podrá acudir al código fuente; código que nos hemos preocupado de documentar de forma muy pormenorizada y explicativa.

7. Evaluación de la iteración

Concluimos la parte de desarrollo correspondiente a la iteración 'C3 – Accesos', dando por desarrollados de forma preliminar los casos de uso en lo que al manejo de accesos se refiere. Creemos que hemos realizado un buen trabajo, creando una interfaz usable, que recoge toda la información que desea el cliente y proporciona una correcta funcionalidad a tenor de los casos de prueba ejecutados.

8. Despliegue

Con este documento y el software desarrollado hasta el momento nos reunimos con el cliente para mostrarle el trabajo realizado.

Tras explicarle el funcionamiento del prototipo y probarlo con varios ejemplos, Pedro Martínez se muestra bastante satisfecho con el diseño, usabilidad y funcionalidad que el software proporciona, dándonos vía libre para proseguir con nuestro trabajo.

Pamplona, a 26 de febrero de 2010.

Héctor Napal Basterrica
Responsable del proyecto

Pedro Martínez Pérez
*Jefe de área de producción de
Informática Navarra S.A.*

accessManager · Iteración C3: Accesos · Página 55 de 55



accessManager
Desarrollo de un sistema de control de accesos a un centro

Iteración C4: Empresas y personal

Fecha de entrega: 5 de marzo de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Estructura del documento

2. Estado del proyecto

3. Análisis

- 3.1. Planificación de la iteración
- 3.2. Análisis de los casos de uso y escenarios

4. Diseño

- 4.1. Casos de uso 13, 19 y 25: Visualizar empresas, departamentos y personal
- 4.2. Caso de uso 14, 20, 26: Visualizar empresa, departamento y persona
- 4.3. Caso de uso 15: Añadir empresa
- 4.4. Caso de uso 16: Borrar empresa
- 4.5. Caso de uso 17: Modificar empresa
- 4.6. Caso de uso 21: Añadir departamento
- 4.7. Caso de uso 22: Borrar departamento
- 4.8. Caso de uso 23: Modificar departamento
- 4.9. Caso de uso 27: Añadir persona
- 4.10. Caso de uso 28: Borrar persona
- 4.11. Caso de uso 29: Modificar persona
- 4.12. Casos de uso 30 y 32: Añadir teléfono o E-Mail
- 4.13. Casos de uso 31 y 33: Borrar teléfono o E-Mail
- 4.14. Caso de uso 34: Añadir cargo
- 4.15. Caso de uso 35: Borrar cargo
- 4.16. Casos de uso 18, 24 y 36: Listar empresas, departamentos y personal

5. Codificación e integración

6. Pruebas

- 6.1. De unidad
- 6.2. De integración
- 6.3. De sistema

7. Evaluación de la iteración

8. Despliegue



1. Introducción

1.1. Propósito

El propósito de este documento no es otro que el de recoger todo el trabajo que realizaremos a lo largo de los próximos cinco días, en los que buscaremos implementar dentro de la pestaña 'Empresas y personal' todos los casos de uso relacionados con empresas, departamentos y personal que accede al centro o con el que se relaciona Informática Navarra S.A..

1.2. Audiencia

Hasta el día 9 de abril éste será un artefacto que guardaremos junto con el resto de documentación asociada al proyecto. A partir de esa fecha el cliente será quien disponga de este documento y pueda así consultarlo él y el personal encargado del futuro mantenimiento.

1.3. Estructura del documento

Iniciamos la redacción de este documento con una breve descripción del estado en el que se encuentra el proceso de desarrollo en el que estamos involucrados, estableciendo qué parte del producto software tenemos implementado y qué pasos deben seguir a partir de ahora.

En la sección 3 planificamos la iteración, concretando los objetivos de la misma, artefactos de entrada y salida que utilizaremos y crearemos y una calendarización detallada de todas las actividades que tendrán lugar. Una vez decididos los casos de uso a implementar redactamos las descripciones de trazo fino que nos permitan profundizar en éstos de cara a su mejor entendimiento.

En el siguiente apartado, el cuarto, abordamos toda la fase de diseño que ha sido necesaria para obtener unas estrategias que codificaremos y que proporcionarán la lógica necesaria para que el usuario pueda obtener los beneficios de la ejecución de los casos de uso. Acto seguido pasamos a la siguiente sección y mencionamos algunos detalles de codificación e integración que hay que tener en cuenta a la hora de llevar a código todo el diseño previo y mostramos los resultados gráficos obtenidos en esta etapa.

En el capítulo 6 acometemos toda la fase de diseño y ejecución de pruebas, recopilando las especificaciones y resultados obtenidos en todas y cada una de ellas.

Para concluir realizamos en las dos últimas secciones una evaluación del trabajo y resultados obtenidos con el desarrollo de esta iteración y mencionamos la información más relevante extraída de la reunión de despliegue realizada con el cliente de cara a evaluar y validar el prototipo software y documentación generada en estos cinco días.



2. Estado del proyecto

Completadas las tres primeras iteraciones de la fase de Construcción ya disponemos de un buen soporte sobre el que continuar desarrollando la aplicación. Tenemos completado el desarrollo de la primera pestaña y en estos momentos nos disponemos a iniciar la segunda, que será una de las de mayor complejidad de todo el desarrollo.

3. Análisis

3.1. Planificación de la iteración

Abrimos la fase de análisis creando un plan de iteración que especifique los objetivos, artefactos, calendarización y casos de uso con los que trabajaremos:

PLAN DE ITERACIÓN C4: EMPRESAS Y PERSONAL			
ASPECTO	DESCRIPCIÓN		
Objetivos	Implementar todos los casos de uso que permitan interactuar con la aplicación y cuyo fin sea el de manipular todos los datos pertenecientes a las empresas, departamentos y personal relacionado con la vida diaria del centro.		
Artefactos de entrada	<ul style="list-style-type: none"> • <i>Especificación inicial de los casos de uso</i> • <i>Planificación temporal y costes detallados del proyecto</i> • Base de datos, API y aplicación desarrolladas en las iteraciones anteriores. 		
Artefactos modificados o creados	<ul style="list-style-type: none"> • Diagrama de clases de la API inicial. • Clases Java de la API inicial y de la aplicación. • El presente documento. 		
Calendarización detallada	Análisis + Pruebas	01/03/10	<ul style="list-style-type: none"> • Planificación de la iteración y gestión de riesgos. • Análisis de los casos de uso y escenarios. • Planificación de las pruebas. • Inicio de la documentación.
	Diseño + Pruebas	02/03/10	<ul style="list-style-type: none"> • Diseño de opciones arquitectónicas. • Diseño de pruebas.
		03/03/10	
	Codificación + Pruebas	03/03/10	<ul style="list-style-type: none"> • Codificación e integración de la solución. • Ejecución y evaluación de pruebas. • Evaluación del prototipo. • Preparación de la entrega ejecutable.
05/03/10			
Despliegue	05/03/10	<ul style="list-style-type: none"> • Muestra, prueba y evaluación conjunta con el cliente. • Concreción de los cambios necesarios. • Finalización de toda la documentación. 	

Planificación de la iteración



Nombre de tarea	Duración	Comienzo	Fin	dom 28 feb	0	0	mar 02 mar	0	0	jue 04 mar	0	0
Estudio de oportunidad (inception)	7 días	jue 07/01/10	vie 15/01/10									
Elaboración	16 días	lun 18/01/10	lun 08/02/10									
Construcción	41 días	mar 09/02/10	mar 06/04/10									
Iteración C1: La base de datos	4 días	mar 09/02/10	vie 12/02/10									
Iteración C2: API para la base de datos	5 días	lun 15/02/10	vie 19/02/10									
Iteración C3: Accesos	5 días	lun 22/02/10	vie 26/02/10									
Iteración C4: Empresas y personal	5 días	lun 01/03/10	vie 05/03/10									
Análisis	1 día	lun 01/03/10	lun 01/03/10									
Diseño	1,5 días	mar 02/03/10	mié 03/03/10									
Implementación	2 días	mié 03/03/10	vie 05/03/10									
Pruebas	5 días	lun 01/03/10	vie 05/03/10									
Despliegue	0,5 días	vie 05/03/10	vie 05/03/10									
Iteración C5: Tarjetas, tipos de acceso y cargos	3 días	lun 08/03/10	mié 10/03/10									
Iteración C6: Zonas, llaves y llaveros	5 días	jue 11/03/10	mié 17/03/10									
Iteración C7: Préstamos	2 días	jue 18/03/10	vie 19/03/10									
Iteración C8: Cuentas	2 días	lun 22/03/10	mar 23/03/10									
Iteración C9: Partes de seguridad	3 días	mié 24/03/10	vie 26/03/10									
Iteración C10: Notas	3 días	lun 29/03/10	mié 31/03/10									
Iteración C11: Base de datos	2 días	jue 01/04/10	vie 02/04/10									
Iteración C12: Diseño, personalización y configuración	2 días	lun 05/04/10	mar 06/04/10									
Capacidad operacional	0 días	mar 06/04/10	mar 06/04/10									
Transición	8 días	mié 07/04/10	vie 16/04/10									
Iteración T1: Preparación de la release final	2 días	mié 07/04/10	jue 08/04/10									
Iteración T2: Entrega	3 días	vie 09/04/10	mar 13/04/10									
Iteración T3: Entrenamiento, soporte y mantenimiento	3 días	mié 14/04/10	vie 16/04/10									
Liberación del producto	0 días	vie 16/04/10	vie 16/04/10									

Recursos	Temporales		5 días
	Humanos		1 persona
	Software	<i>Sistema operativo</i>	Microsoft Windows 7
		<i>Ofimática</i>	Microsoft Word 2010 Adobe Acrobat Professional 8
		<i>Diseño</i>	Visual Paradigm 4.2
		<i>Codificación</i>	NetBeans 6.8
Hardware		1 PC	

Casos de uso y escenarios implementados	13 - Visualizar empresas	25 - Visualizar personal
	14 - Visualizar empresa	26 - Visualizar persona
	15 - Añadir empresa	27 - Añadir persona
	16 - Borrar empresa	28 - Borrar persona
	17 - Modificar empresa	29 - Modificar persona
	18 - Listar empresas	30 - Añadir teléfono
	19 - Visualizar departamentos	31 - Borrar teléfono
	20 - Visualizar departamento	32 - Añadir E-Mail
	21 - Añadir departamento	33 - Borrar E-Mail
	22 - Borrar departamento	34 - Añadir cargo
	23 - Modificar departamento	35 - Borrar cargo
	24 - Listar departamentos	36 - Listar personal

Planificación de la iteración



3.2. Análisis de los casos de uso y escenarios

En un principio disponemos de las descripciones de trazo grueso de los 24 casos de uso seleccionados, pero antes de ponernos a diseñar nada es necesario ahondar en ellas y crear las siguientes de trazo fino:

Caso de uso 13: Visualizar empresas	
Descripción	Lo primero que debemos conseguir para poder manejar todas las empresas y personal que accede al centro es poder visualizar las primeras.
Actores principal	Administrador, Autorizador y RP.
Pre-condiciones	El usuario se encuentra logueado en el sistema y desea visualizar qué empresas forman parte de la vida diaria del centro.
Flujos	Básico 1. El usuario accede a la pestaña de 'Empresas y personal' y define la búsqueda de empresa. 2. El sistema se encarga de mostrarle todas las empresas existentes que satisfacen dichos filtros.
	Alternativos (*) Ninguno.
Post-condiciones	El usuario visualiza en pantalla todas las empresas registradas en el sistema que cumplen con sus criterios de búsqueda.

Caso de uso 13: Visualizar empresas

Caso de uso 14: Visualizar empresa	
Descripción	Una empresa no solo está compuesta de su nombre, sino de otros datos, departamentos y personal que la conforman.
Actores principal	Administrador, Autorizador y RP.
Pre-condiciones	El usuario se encuentra visualizando el conjunto de empresas registradas en el sistema y desea acceder a todos los datos de una de ellas.
Flujos	Básico 1. El usuario selecciona la empresa que quiere consultar. 2. El sistema se encarga de mostrarle todos los datos secundarios, departamentos y personal que la conforman.
	Alternativos (*) Ninguno.
Post-condiciones	El usuario consigue su objetivo: poder consultar a través del monitor todos los datos que componen la empresa seleccionada.

Caso de uso 14: Visualizar empresa



Caso de uso 15: Añadir empresa	
Descripción	Como ya comentamos con anterioridad, el conjunto de empresas que trabaja en el centro es muy variable, lo que hace que debamos ofrecer al usuario una vía para insertar nuevas.
Actores principal	Administrador y Autorizador.
Pre-condiciones	El usuario se encuentra en el sistema, más concretamente en la tab de empresas y personal, y desea registrar una nueva empresa.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario pulsa el botón de 'Añadir empresa'. 2. El sistema le proporciona un formulario con todos los campos que es necesario rellenar para dar de alta una nueva empresa. 3. El usuario los rellena y pide al sistema que realice el alta. 4. Éste le contesta mostrándole un mensaje de aviso con la operación que se encuentra a punto de realizar y solicitándole una confirmación previa para ello. 5. El usuario la da. (*) 6. El sistema registra la nueva empresa en el sistema. (*)
	<p>Alternativos (*)</p> <p>El usuario no da el visto bueno a la inserción:</p> <ul style="list-style-type: none"> • El sistema no realiza ninguna acción y vuelve a la mitad del tercer punto sin haber modificado ningún campo. <p>El usuario introduce algún campo con formato erróneo:</p> <ul style="list-style-type: none"> • El sistema le avisa de su error y queda a la espera de que continúe con el caso de uso o inicie otro.
Post-condiciones	El usuario consigue registrar la empresa en el sistema, pasando a ser ésta operativa a partir de ahora.

Caso de uso 15: Añadir empresa



Caso de uso 16: Borrar empresa		
Descripción	Así como se debe poder añadir empresas, también debe ser posible eliminarlas ya que no formarán parte del centro de por vida.	
Actores principal	Administrador y Autorizador.	
Pre-condiciones	Una o varias empresas dejan de formar parte del centro, lo que hace que el usuario se logué en el sistema con la intención de eliminarlas.	
Flujos	Básico	<ol style="list-style-type: none">1. El usuario selecciona del conjunto de empresas que el sistema muestra aquellas que desea eliminar y traslada al sistema su intención.2. Éste le pide que confirme la operación para que el borrado tenga lugar.3. El usuario responde 'Sí' y el caso de uso cumple su objetivo. (*)
	Alternativos (*)	El usuario aborta el borrado: <ul style="list-style-type: none">• El sistema no realiza ninguna acción y el caso de uso concluye.
Post-condiciones	El sistema ya no almacena el conjunto de empresas que el usuario mandó eliminar.	

Caso de uso 16: Borrar empresa



Caso de uso 17: Modificar empresa	
Descripción	No es extraño que una empresa pueda cambiar de dirección, o que veamos oportuno modificar su código, nombre,...
Actores principal	Administrador y Autorizador.
Pre-condiciones	El usuario se encuentra visualizando el conjunto de empresas existentes en el sistema y tiene intención de modificar los datos de alguna de ellas.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario selecciona la empresa a modificar. 2. El sistema le proporciona un formulario con los datos que posee actualmente. 3. El usuario edita todos los que estime erróneos y pide al sistema que lleve a cabo la actualización. 4. El sistema envía al usuario un diálogo indicándole que necesita una autorización previa para poder realizar la acción que demanda. 5. El usuario da su visto bueno. (*) 6. El sistema actualiza la empresa con los datos proporcionados. (*)
	<p>Alternativos (*)</p> <p>El usuario responde 'No' ante la advertencia del sistema:</p> <ul style="list-style-type: none"> • El caso de uso concluye sin lograr el fin para el que fue ideado. <p>Alguno de los datos introducidos por el usuario es erróneo:</p> <ul style="list-style-type: none"> • El sistema le avisa del error acontecido y permanece a la espera de nuevas acciones sin haber realizado ningún tipo de actualización.
Post-condiciones	La empresa que al comienzo de este caso de uso constaba con datos erróneos ahora lo hace con válidos.

Caso de uso 17: Modificar empresa



Caso de uso 18: Listar empresas		
Descripción	El usuario nos indica explícitamente que desea disponer de una vía con la que poder obtener un listado de todas las empresas participantes en la vida del centro.	
Actores principal	Administrador y Autorizador.	
Pre-condiciones	El usuario ha desplegado la aplicación y desea obtener un listado de las empresas que se encuentra visualizando.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario pide al sistema que genere el listado de empresas que visualiza en pantalla. 2. El sistema le insta a que le indique una ruta en la que depositar el reporte, a lo que el usuario responde ofreciéndosela. (*) 3. El sistema elabora el reporte solicitado y lo deposita donde el usuario indicó.
	Alternativos (*)	<p>El usuario no ofrece una ruta en la que depositar el listado:</p> <ul style="list-style-type: none"> • El sistema da por concluido el caso de uso sin elaborar ningún tipo de informe.
Post-condiciones	El usuario dispone del reporte que solicitó.	

Caso de uso 18: Listar empresas

Caso de uso 19: Visualizar departamentos		
Descripción	Se pueden realizar dos tipos de visualización de departamento: todos o los pertenecientes a una empresa indicada.	
Actores principal	Administrador, Autorizador y RP.	
Pre-condiciones	El usuario accede a accessManager con la intención de visualizar los departamentos existentes y se sitúa en la pestaña de empresas y personal.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario se sitúa en la pestaña de empresas y personal e indica al sistema que le muestre todos los departamentos. (*) 2. El sistema accede a la base de datos, recoge los que necesite y se los muestra por pantalla.
	Alternativos (*)	<p>El usuario selecciona una empresa concreta y/o redefine la búsqueda:</p> <ul style="list-style-type: none"> • El sistema continúa en el punto 2 pero obteniendo solo aquellos departamentos pertenecientes a la empresa seleccionada y/o que cumplan con los filtros de búsqueda indicados.
Post-condiciones	El usuario obtiene por pantalla todos los departamentos y personas que solicitó a través de los diversos campos de búsqueda.	

Caso de uso 19: Visualizar departamentos



Caso de uso 20: Visualizar departamento		
Descripción	Si alguien desea obtener toda la información de que consta un departamento, nosotros le ofreceremos una vía para ello.	
Actores principal	Administrador, Autorizador y RP.	
Pre-condiciones	El usuario loguea en el sistema y se sitúa en la tab 'Empresas y personal'.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario restringe la visualización de departamentos hasta conseguir visualizar el que desee. Lo selecciona e indica al sistema que muestre toda la información que disponga de éste. 2. El sistema accede a la base de datos y muestra por pantalla todos los datos de los que consta el departamento, así como el personal que forma parte de él.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario visualiza por pantalla toda la información que buscaba.	

Caso de uso 20: Visualizar departamento

Caso de uso 21: Añadir departamento		
Descripción	Al igual que las empresas varían, los departamentos también lo harán dentro de ellas.	
Actores principal	Administrador y Autorizador.	
Pre-condiciones	El usuario accede al sistema para dar de alta un nuevo departamento.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona la empresa a la que pertenecerá el departamento y pide al sistema añadir uno nuevo. 2. El sistema le ofrece un formulario que deberá rellenar. 3. El usuario obedece y tras concluir de editarlo pide al sistema que realice el alta. 4. El sistema responde con un mensaje de aviso, donde informa al usuario la operación que va a realizar y pidiéndole que certifique si es la deseada. 5. El usuario responde afirmativamente y el departamento es registrado en el sistema. (*)
	Alternativos (*)	<p>El usuario responde negativamente:</p> <ul style="list-style-type: none"> • El sistema no realiza ningún registro y el caso de uso finaliza. <p>El usuario introduce algún dato erróneo:</p> <ul style="list-style-type: none"> • El sistema le comunica el error que ha realizado y permanece a la espera de una nueva inserción o el inicio de otro caso de uso.
Post-condiciones	El departamento suministrado pasa a formar parte del sistema.	

Caso de uso 21: Añadir departamento



Caso de uso 22: Borrar departamento	
Descripción	Así como se pueden añadir departamentos, también deberá ser posible poder eliminarlos.
Actores principal	Administrador y Autorizador.
Pre-condiciones	El usuario visualiza en pantalla algunos departamentos que no debieran estar registrados, lo que hace que desee eliminarlos.
Flujos	Básico <ol style="list-style-type: none">1. El usuario selecciona los departamentos que desea eliminar y pide al sistema que los borre.2. El sistema insta al usuario a que confirme que desea realizar el borrado de los departamentos seleccionados.3. El usuario da el visto bueno ya que ve correcta la operación que se va a realizar. (*)4. El sistema elimina de la base de datos todos los departamentos indicados.
	Alternativos (*) <p>El usuario decide abortar el borrado por no estimarlo oportuno o haber seleccionado algún departamento que no debía:</p> <ul style="list-style-type: none">• El sistema da por concluido el caso de uso sin realizar ninguna operación sobre la base de datos.
Post-condiciones	Los departamentos que el usuario seleccionó ya no forman parte del sistema.

Caso de uso 22: Borrar departamento



Caso de uso 23: Modificar departamento		
Descripción	Acompañamos las operaciones de adición y borrado con la de modificación.	
Actores principal	Administrador y Autorizador.	
Pre-condiciones	El usuario se encuentra dentro del sistema visualizando los departamentos de una empresa y estima oportuno modificar uno de ellos ya que los datos que contiene no son correctos.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona la empresa y departamento a modificar y pide al sistema poder actualizarlo. 2. El sistema le proporciona todos los datos que lo conforman actualmente. 3. El usuario modifica los que no cree correctos y pasa el testigo al sistema. 4. El sistema realiza la actualización siempre y cuando el usuario dé su visto bueno. (*)
	Alternativos (*)	<p>El usuario no lo da:</p> <ul style="list-style-type: none"> • El sistema no realiza la actualización y queda a la espera de nuevas órdenes. <p>Alguno de los campos que el usuario ha editado consta con datos con formato erróneo:</p> <ul style="list-style-type: none"> • El sistema avisa al usuario del error que acaba de cometer y permanece al final del tercer punto de este caso de uso.
Post-condiciones	El usuario consigue que el departamento conste con los datos deseados.	

Caso de uso 23: Modificar departamento

Caso de uso 24: Listar departamentos		
Descripción	Caso de uso idéntico al 18, pero tomando como referencia los departamentos.	
Actores principal	Administrador y Autorizador.	
Pre-condiciones	El usuario despliega accessManager con la intención de obtener un listado de departamentos.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario se sitúa en la segunda pestaña, define la búsqueda de departamentos y pide al sistema que genere un reporte con los visualizados por pantalla. 2. El sistema obedece, pero previamente el usuario deberá ofrecerle una ruta donde depositarlo. (*)
	Alternativos (*)	<p>El usuario no ofrece ninguna ruta:</p> <ul style="list-style-type: none"> • El sistema salta al final del caso de uso sin elaborar ningún informe.
Post-condiciones	El usuario puede disfrutar del reporte solicitado en la ubicación que indicó durante el transcurso del caso de uso.	

Caso de uso 24: Listar departamentos



Caso de uso 25: Visualizar personal		
Descripción	Los últimos entes a tratar son las personas que forman parte de las empresas y departamentos anteriormente citados. En primer lugar facilitaremos la visualización de éste en función de los parámetros de búsqueda proporcionados.	
Actores principal	Administrador, Autorizador y RP.	
Pre-condiciones	El usuario de la aplicación accede a ella con la intención de visualizar el conjunto de personas que cumplen unas características concretas.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario se sitúa en la pestaña de empresas y personal y define el conjunto de personas que desea visualizar (totales, pertenecientes a una empresa y/o pertenecientes a un departamento,...). 2. El sistema accede a la base de datos para mostrar por pantalla todos los datos que obtiene en función de los parámetros de búsqueda proporcionados.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario puede disfrutar en pantalla del conjunto de personas que quería.	

Caso de uso 25: Visualizar personal

Caso de uso 26: Visualizar persona		
Descripción	La muestra de los principales datos del personal puede resultar escasa para las pretensiones del usuario. Con este caso de uso le permitiremos que visualice toda la información de la persona que desee.	
Actores principal	Administrador.	
Pre-condiciones	El usuario se encuentra logueado en el sistema y con la necesidad de visualizar todos los datos correspondientes a una persona mostrada en pantalla.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona la persona que desea consultar y pide al sistema que le proporcione todos los datos que disponga de ella. 2. El sistema obedece, recuperando toda la información oportuna de la base de datos y proporcionándole un formulario elaborado con ésta.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario consigue su objetivo: visualizar por pantalla todos los datos pertenecientes a la persona seleccionada.	

Caso de uso 26: Visualizar persona



Caso de uso 27: Añadir persona	
Descripción	Cuando una nueva persona entra a formar parte de algún tipo de empresa-departamento, ésta deberá ser registrada en el sistema si deseamos poder manejarla en las diferentes funcionalidades que ofrecerá accessManager.
Actores principal	Administrador, Autorizador y RP.
Pre-condiciones	Una nueva persona entra a formar parte de la vida diaria del centro, lo que lleva al usuario a querer registrarla.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario accede al sistema y se sitúa en la pestaña 'Empresas y personal'. A continuación selecciona la empresa (y departamento) que corresponda e indica al sistema que desea añadirle una nueva persona. 2. El sistema le ofrece un formulario diseñado en función del tipo de persona a insertar; formulario que el usuario rellena y devuelve. 3. El sistema pide al usuario que dé su aprobación a la operación que se encuentra a punto de ejecutarse. 4. El usuario la da. (*) 5. El sistema registra la nueva persona en la empresa (y departamento) indicada. (*)
	<p>Alternativos (*)</p> <p>El usuario no ve oportuno realizar el alta:</p> <ul style="list-style-type: none"> • El sistema no la lleva a cabo y permanece a la espera de nuevas órdenes. <p>El usuario proporciona algún dato erróneo:</p> <ul style="list-style-type: none"> • El sistema le avisa del error que acaba de cometer y pasa al final del punto 2.
Post-condiciones	El usuario consigue realizar el alta en el sistema de la nueva persona.

Caso de uso 27: Añadir persona



Caso de uso 28: Borrar persona	
Descripción	El borrado de personas es tan necesario como la adición, ya que durante la relación que mantengan Informática Navarra S.A. y cualquier empresa-departamento, dentro de ésta variará el flujo de personal destinado al centro.
Actores principal	Administrador, Autorizador y RP.
Pre-condiciones	Un conjunto de personas deja de acudir de manera definitiva al centro, lo que provoca que cualquiera de los tres actores quiera darlas de baja en el sistema.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario despliega accessManager y se dirige a la tab de empresas y personal. Una vez allí selecciona la empresa (y departamento) y el conjunto de personas que desea eliminar y pulsa el botón de borrado. 2. El sistema genera un mensaje para avisar al usuario de la operación que está a punto de ejecutar y le pide que la ratifique. 3. El usuario responde afirmativamente y el borrado de las personas seleccionadas es llevado a cabo. (*)
	<p>Alternativos (*)</p> <ul style="list-style-type: none"> • El usuario responde negativamente: <ul style="list-style-type: none"> • El sistema no realiza ninguna acción y da el caso de uso por finalizado.
Post-condiciones	El conjunto de personas que el usuario mandó borrar ya no forman parte del sistema.

Caso de uso 28: Borrar persona



Caso de uso 29: Modificar persona	
Descripción	Éste sí que será un caso de uso menos frecuente ya que el cambio de alguno de los datos que conforman una persona será un hecho aislado. Aun así, como puede ser necesario y el cliente nos los exige, implementaremos un caso de uso que dé solución a esta situación.
Actores principal	Administrador, Autorizador y RP.
Pre-condiciones	El usuario se encuentra en el sistema, más concretamente en la tab de empresas y personal, y visualiza una persona que consta con datos erróneos.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario selecciona la persona e indica al sistema que desea actualizar sus datos. 2. El sistema le proporciona un formulario con todos los datos que dispone de ésta actualmente. 3. El usuario modifica los datos que crea convenientes y pide al sistema que lleve a cabo la actualización. 4. Éste obedece siempre y cuando el usuario dé su visto bueno. (*)
	<p>Alternativos (*)</p> <p>El usuario decide abortar la actualización:</p> <ul style="list-style-type: none"> • El sistema no modifica ningún registro de la base de datos y el caso de uso queda estancado a mitad del punto 3. <p>El usuario proporciona algún dato erróneo:</p> <ul style="list-style-type: none"> • El sistema no puede llevar a cabo la actualización solicitada, lo que hace que muestre un mensaje con el error cometido y quede a la espera de nuevas órdenes.
Post-condiciones	La actualización que el usuario buscaba ha sido llevada a buen puerto.

Caso de uso 29: Modificar persona



Caso de uso 30: Añadir teléfono		
Descripción	accessManager permitirá llevar a cabo un registro de todos los números de teléfono que una persona disponga.	
Actores principal	Administrador, Autorizador y RP.	
Pre-condiciones	El usuario se encuentra en la pestaña 'Empresas y personal' y desea añadir un nuevo número de teléfono a una de las personas que visualiza.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona la persona e introduce el nuevo número de teléfono, indicando al sistema que lo registre en la base de datos. 2. El sistema obedece. (*)
	Alternativos (*)	<p>El sistema no puede realizar la inserción ya que el número de teléfono proporcionado no dispone del formato deseado:</p> <ul style="list-style-type: none"> • El sistema avisa al usuario del error que ha cometido y le pide que corrija el número de teléfono que desea introducir, permaneciendo a la espera.
Post-condiciones	La persona seleccionada pasa a disponer de un nuevo número de teléfono registrado en el sistema.	

Caso de uso 30: Añadir teléfono

Caso de uso 31: Borrar teléfono		
Descripción	En el momento de que una persona ya no disponga de un teléfono que fue registrado en el sistema será conveniente proceder con su eliminación, pues no tiene sentido almacenar información innecesaria.	
Actores principal	Administrador, Autorizador y RP.	
Pre-condiciones	Una persona deja de poseer un número de teléfono, disponiéndose el usuario a eliminarlo.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario loguea en el sistema, se sitúa en la pestaña de empresas y personal y define la búsqueda hasta dar con la persona deseada. Acto seguido selecciona de entre los existentes el conjunto de números de teléfono que desea borrar y clicla en el botón de borrado. 2. El sistema elimina los teléfonos del sistema previa autorización por parte del usuario. (*)
	Alternativos (*)	<p>El usuario no da su autorización:</p> <ul style="list-style-type: none"> • No tiene lugar ningún borrado y el caso de uso concluye.
Post-condiciones	La persona seleccionada ya no dispone de los teléfonos eliminados entre los registrados.	

Caso de uso 31: Borrar teléfono



Caso de uso 32: Añadir E-Mail	
Descripción	Nos encontramos en la misma situación que con los teléfonos, pero en este caso lo que trataremos serán las direcciones de correo electrónico del personal.
Actores principal	Administrador, Autorizador y RP.
Pre-condiciones	Cualquiera de los tres actores se encuentra con la aplicación accessManager desplegada y se dispone a asignar una nueva dirección de correo electrónico a una de las personas ya registradas.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario accede a la segunda tab y selecciona la empresa (y departamento) que le lleve a visualizar a la persona deseada. La selecciona e indica el nuevo correo electrónico a registrar. 2. El sistema registra el E-Mail y se lo asigna a la persona seleccionada. (*)
	<p>Alternativos (*)</p> <p>No puede realizarse la adición ya que el E-Mail proporcionado posee un formato no válido:</p> <ul style="list-style-type: none"> • El sistema avisa al usuario de su error y permanece a la espera de que corrija la dirección o pase a ejecutar otro caso de uso.
Post-condiciones	La persona seleccionada dispone de una nueva dirección de correo electrónico registrada.

Caso de uso 32: Añadir E-Mail

Caso de uso 33: Borrar E-Mail	
Descripción	El borrado de cualquier dirección de correo electrónico es semejante al realizado con los números de teléfono.
Actores principal	Administrador, Autorizador y RP.
Pre-condiciones	El usuario se encuentra visualizando una persona y observa que posee direcciones E-Mail que ha dejado de utilizar y se dispone a borrarlos.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario selecciona el conjunto de direcciones que desea eliminar y clic en el botón de borrado. 2. El sistema se encarga de hacerlas desaparecer de la base de datos si el usuario da su visto bueno. (*)
	<p>Alternativos (*)</p> <p>El usuario no lo da:</p> <ul style="list-style-type: none"> • El sistema no realiza ningún borrado en la base de datos y da el caso de uso por finalizado, permaneciendo a la espera de nuevas órdenes.
Post-condiciones	El borrado de direcciones E-Mail ha sido realizado correctamente.

Caso de uso 33: Borrar E-Mail



Caso de uso 34: Añadir cargo	
Descripción	En este caso haremos alusión a las personas con las que Informática Navarra S.A. mantiene contacto.
Actores principal	Administrador, Autorizador y RP.
Pre-condiciones	El usuario se encuentra logueado en accessManager en disposición de asignar un nuevo conjunto de cargos a uno de los contactos existentes.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario selecciona la empresa y departamento en la que se encuentra encuadrado el contacto para así visualizarlo en la tabla destinada a éstos. Lo selecciona y pide al sistema poder añadirle un nuevo cargo. 2. El sistema le ofrece la lista de cargos existentes, seleccionando el usuario los que desee añadir. 3. El sistema pide al usuario que confirme la operación de adición que va a ser realizada, tras la cual el sistema añade los cargos indicados al contacto seleccionado y da el caso de uso por finalizado. (*)
	<p>Alternativos (*)</p> <p>El usuario aborta la operación de adición:</p> <ul style="list-style-type: none"> • El sistema no realiza ninguna acción y pasa al final del segundo punto del flujo básico.
Post-condiciones	El contacto pasa a disponer de los cargos que poseía junto con los nuevos asignados.

Caso de uso 34: Añadir cargo

Caso de uso 35: Borrar cargo	
Descripción	No es muy común, pero sí que es verdad que las responsabilidades de cualquier contacto con respecto a Informática Navarra S.A. pueden variar. De ahí el origen de este caso de uso.
Actores principal	Administrador, Autorizador y RP.
Pre-condiciones	El usuario, logueado en el sistema y visualizando el conjunto de contactos de una empresa-departamento percibe que un contacto dispone de algún tipo de cargo que ya no desempeña y se dispone a eliminarlo.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario selecciona el conjunto de cargos desempeñados por un contacto que no deberían constar en el sistema y pulsa sobre el botón de borrado. 2. El sistema se encarga de eliminarlos y dar por finalizado exitosamente el caso de uso solo sí el usuario a dado su autorización previa. (*)
	<p>Alternativos (*)</p> <p>El usuario contesta 'No' ante el diálogo de aviso:</p> <ul style="list-style-type: none"> • El sistema no borra ningún cargo, dando por concluido el caso de uso en ejecución.
Post-condiciones	El contacto ya no dispone de cargos que no desempeña en la actualidad.

Caso de uso 35: Borrar cargo

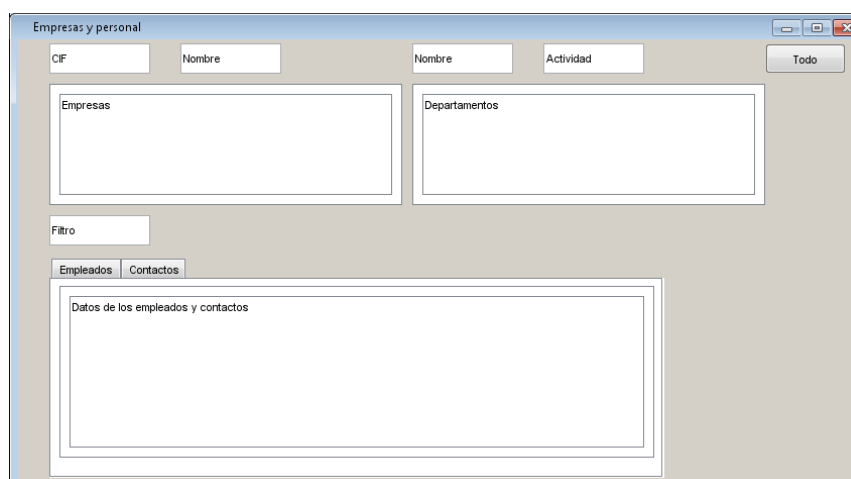
Caso de uso 36: Listar personal	
Descripción	Para terminar los casos de uso que implementaremos en esta pestaña, el cliente nos pide que le proporcionemos una vía para generar un listado con el personal que se encuentre visualizando en pantalla.
Actores principal	Administrador, Autorizador y RP.
Pre-condiciones	El usuario se encuentra situado en la pestaña 'Empresas y personal' y ve interesante generar un informe con las personas que visualiza.
Flujos	Básico <ol style="list-style-type: none"> 1. El usuario pide al sistema que elabore un reporte con los usuarios que muestra en la tabla de personal. 2. Éste le responde pidiéndole que indique una ruta en la que depositar el informe solicitado. 3. El usuario la ofrece y el sistema se encarga de crear y depositar el reporte en ésta. (*)
	Alternativos (*) <p>El usuario no proporciona ninguna ruta, cancelando el cuadro de selección:</p> <ul style="list-style-type: none"> • El sistema no elabora ningún informe y el caso de uso llega a su fin.
Post-condiciones	El usuario dispone del informe solicitado en la ruta indicada.

Caso de uso 36: Listar personal

4. Diseño

4.1. Casos de uso 13, 19 y 25: Visualizar empresas, departamentos y personal

Vamos a comenzar diseñando la estructura inicial de la pestaña, que estará compuesta por cuatro tablas, dos de las cuales estarán agrupadas en un JTabbedPane. Además se crean campos que permitirán acotar el resultado de la búsqueda sobre las tablas y un botón que devuelva el estado de éstas al inicial:



Interfaz gráfica propuesta



Con toda esta disposición de componentes gráficos nos encontramos en disposición de comenzar. Partiremos creando una serie de funciones que permitan mostrar el contenido de las diversas tablas en función de los parámetros indicados, que en nuestro caso serán las filas seleccionadas y los valores de los campos de búsqueda. En concreto tendremos que:

1. Rellenar la tabla de empresas:
 - 1.1. Crear un objeto de tipo Empresa.
 - 1.2. Rellenar la tabla de empresas con los resultados obtenidos de llamar a *datosEmpresas*.
2. Rellena la tabla de departamentos:
 - 2.1. Crear un objeto de tipo Empresa.
 - 2.2. Rellenar la tabla de departamentos con los resultados proporcionados por *datosDepartamentos*.
3. Rellenar la tabla de empleados:
 - 3.1. Crear un objeto de tipo Empresa.
 - 3.2. Rellenar la tabla de empleados con el ResultSet que devuelve *datosEmpleados*.
4. Rellenar la tabla de contactos:
 - 4.1. Crear un objeto de tipo Empresa.
 - 4.2. Rellenar la tabla de contactos con los resultados extraídos de *datosContactos*.

Cada tabla poseerá un método que la refresque tomando como parámetros de entrada las variables mencionadas (uno por cada punto del listado anterior):

MÉTODO	FUNCIONALIDAD
<i>refrescarJTEmpresas</i>	Método que refresca la tabla de empresas a través de la llamada al método <i>datosEmpresas</i> .
<i>datosEmpresas</i>	Método que obtiene los datos de todas las empresas que cumplen con los filtros pasados como parámetros de entrada.
<i>refrescarJTDepartamentos</i>	Método que refresca el contenido de JTDepartamentos con los resultados obtenidos de <i>datosDepartamentos</i> .
<i>datosDepartamentos</i>	Método que obtiene los datos de todos aquellos departamentos que cumplen con los filtros obtenidos paramétricamente.
<i>refrescarJTEmpleados</i>	Método que rellena la tabla de empleados con los registros extraídos de <i>datosEmpleados</i> .
<i>datosEmpleados</i>	Método que obtiene los datos de todos aquellos empleados que cumplen con los filtros leídos como parámetros de entrada.



<i>refrescarJTContactos</i>	Método que se encarga de refrescar la tabla de contactos con aquellos obtenidos de la llamada al método <i>datosContactos</i> .
<i>datosContactos</i>	Método que recupera de la base de datos todos los datos de aquellos contactos que superan los filtros leídos como parámetros de entrada.

Métodos utilizados para inicializar la pestaña 'Empresas y personal'

El primer, tercer, quinto y séptimo métodos serán métodos pertenecientes al *frame* principal cada uno de los cuales llamará a su método de obtención de datos. Estos métodos (*datosXXXX*) deberán ser diseñados y codificados en nuestra API. Una vez que dispongamos de ellos podremos agruparlos en un método que los llame según la String pasada como entrada y que denotará las tablas a actualizar:

refrescarTablasEmpresasPersonal(String tablas):

```
{  
    Lectura de parámetros.  
  
    Si (tablas contiene "E")    refrescarJTEmpresas(Parámetros);  
    Si (tablas contiene "D")    refrescarJTDepartamentos(Parámetros);  
    Si (tablas contiene "G")    refrescarJTEmpleados(Parámetros);  
    Si (tablas contiene "C")    refrescarJTContactos(Parámetros);  
}
```

Flujo de control del método refrescarTablasEmpresasPersonal

Este método será muy conveniente de cara a poder actualizar tablas en un futuro y será refinado a medida que surjan nuevas.

En un principio, al inicializar el contenido de la pestaña, los parámetros leídos serán inexistentes, pero cada vez que introduzcamos alguno o que pinchemos sobre alguna empresa (y departamento) volveremos a refrescar el contenido de las tablas supeditadas (según el orden empresas > departamentos > personal) obteniendo resultados distintos al ser los parámetros de búsqueda también distintos.



Este es el proceso de refresco de tablas inicial que utilizaremos para visualizar los datos:

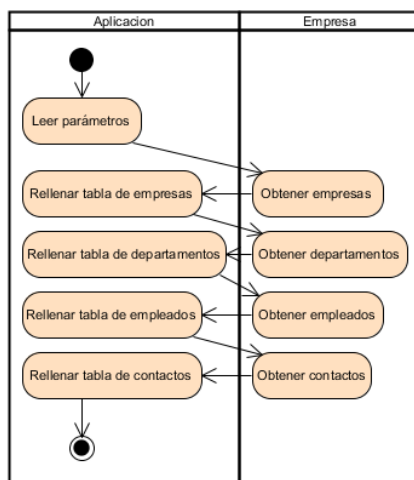


Diagrama de actividades

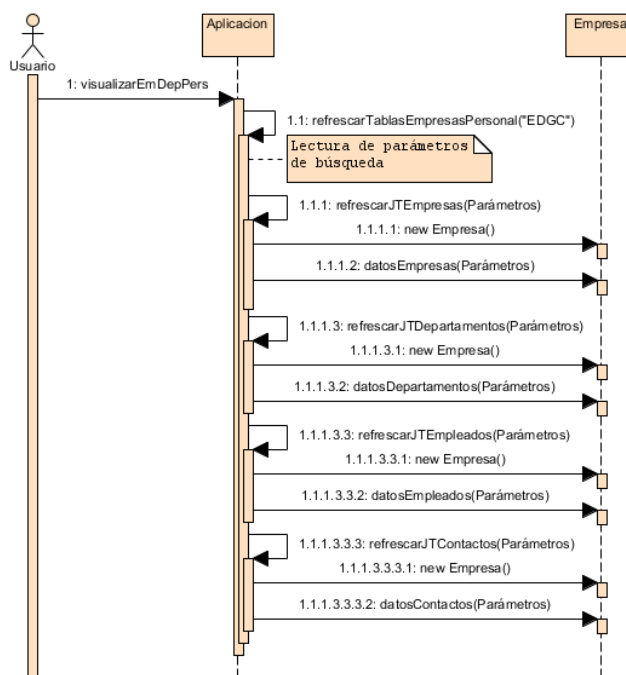


Diagrama de secuencia

Los datos extraídos en cada caso son:

- Tabla 'Empresas': Nombre.
- Tabla 'Departamentos': Nombre y actividad.
- Tablas 'Empleados' y 'Contactos': DNI, primer apellido, segundo apellido y nombre.

La visualización del resto de atributos será realizada en los casos de uso destinados para ello (14, 20 y 26).

4.2. Caso de uso 14, 20, 26: Visualizar empresa, departamento y persona

Para el resto de datos que no tratamos en los casos de uso número 13, 19 y 25 debemos crear una vía que permita al usuario acceder a ellos. Para tal fin se añade a la pestaña una serie de JPanels que muestren la información de cada empresa, departamento y persona seleccionada y que en el futuro emplearemos también para realizar las operaciones de adición y modificación:

- Empresas: panel de datos.
- Departamentos: panel de datos y de llaves prestables.
- Personal: panel de datos comunes (datos personales, teléfonos e E-mails) y paneles de datos específicos (llaves, cargos,...).

Interfaz gráfica propuesta

Dependiendo de la tabla que manejemos el sistema deberá seguir el siguiente proceso:

Si clicamos sobre 'Empresas':

1. Obtenemos el nombre de la empresa seleccionada.
2. Si es 'Ocasional' → No mostraremos datos en el formulario salvo el nombre, que será "Ocasional".
3. Si es 'Informática Navarra S.A.' → Haremos lo mismo que en el anterior punto.
4. En cualquier otro caso (resto de empresas) → Obtenemos los datos de la empresa y rellenamos el formulario.

Si clicamos sobre 'Departamentos':

1. Leer los nombres de empresa y departamento seleccionados.
2. Obtener el objeto Departamento que tenga el nombre leído.
3. Rellenar el formulario con los datos del Departamento.



Si clicamos sobre 'Empleados':

1. Leer el DNI de la persona seleccionada.
2. Mostrar sus datos personales (comunes).
3. Obtenemos el tipo de persona seleccionada.
4. Si es de tipo 'Ocasional' → Crearemos un objeto Ocasional con el DNI y la primera empresa (por orden alfabético) a la que pertenezca con el que rellenar sus datos específicos y la tabla de llaves que pueden serle prestadas.
5. Si es 'Residente' → Obtenemos el objeto Residente a partir del DNI leído y mediante el cual obtendremos el nivel a indicar en el formulario.
6. Si es 'Vigilante' → Crearemos un objeto Vigilante pasándole como entrada el DNI leído y obteniendo a partir de él el código de vigilante.

Nota: En cualquier otro caso (resto de personal) no se realizará ninguna opción ya que con los datos comunes será suficiente (no existen más).

Si clicamos sobre 'Contactos' → Mostraremos los datos personales de la persona leída del mismo modo que hicimos anteriormente.

Con el fin de poder implementar estos cuatro procesos es necesario añadir a la API todos los constructores y métodos que emplearemos y que resumimos en la siguiente tabla:

MÉTODO	FUNCIONALIDAD
<i>restablecerDatos</i>	Método que da valor a los atributos de un objeto Empresa en función del código que posea y con la ayuda de la base de datos.
<i>obtenerCodEmp</i>	Método que obtiene el código de empresa a partir de su nombre.
<i>obtenerCodDep</i>	Método que obtiene el código de departamento a partir de su nombre y empresa a la que pertenece (ejecutado por ésta).
<i>Personal</i>	Constructor que crea un objeto Personal a partir del DNI asignado. Se empleará para obtener los datos comunes a todo el personal.
<i>Ocasional</i>	Constructor que crea un objeto de tipo Ocasional a partir de su clave primaria (DNI y empresa).
<i>Residente</i>	Constructor que devuelve un objeto Residente tomando como parámetro de entrada el DNI de la persona y obteniendo el resto de atributos a partir de los datos obtenidos de la base de datos.
<i>Vigilante</i>	Constructor que crea un objeto de tipo Vigilante a partir del DNI que éste posee.

Métodos a implementar y a añadir a la API



Ahora que tratamos un mayor número de datos en comparación con los tres primeros casos de uso diseñados, debemos ampliar el método que construimos con el fin de refrescar las nuevas tablas de la pestaña, incluyendo llamadas dentro de éste a nuevos métodos:

```
refrescarTablasEmpresasPersonal(String tablas):  
{  
    Lectura de parámetros.  
  
    Si (tablas contiene "E")    refrescarJTEmpresas(Parámetros);  
    Si (tablas contiene "D")    refrescarJTDepartamentos(Parámetros);  
    Si (tablas contiene "G")    refrescarJTEmpleados(Parámetros);  
    Si (tablas contiene "C")    refrescarJTContactos(Parámetros);  
    Si (tablas contiene "T")    refrescarJTTFns(Parámetros);  
    Si (tablas contiene "M")    refrescarJTEmails(Parámetros);  
    Si (tablas contiene "X")    refrescarJTCargos(Parámetros);  
    Si (tablas contiene "L")    refrescarJTLlavesDep(Parámetros);  
    Si (tablas contiene "L2")   refrescarJTLlavesOcas(Parámetros);  
}  
  
Flujo de control del método refrescarTablasEmpresasPersonal
```

A groso modo ya tendríamos toda la estrategia diseñada. Para acabar de determinar todos los detalles elaboramos los siguientes diagramas que permitirán establecer el flujo de actividades y de llamadas a codificar:

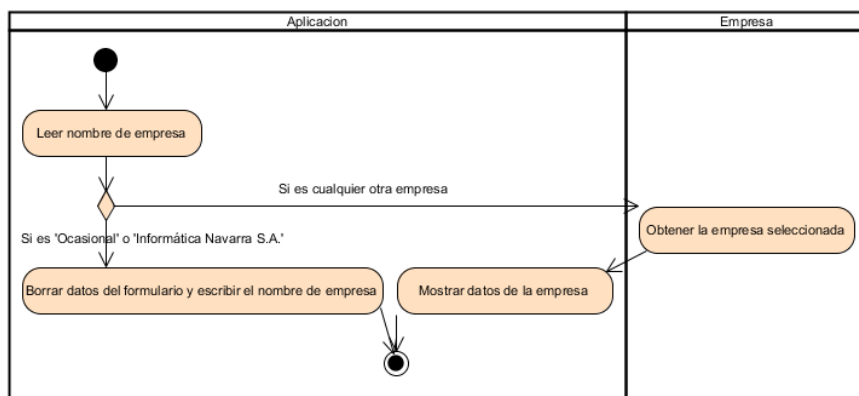


Diagrama de actividades - Visualizar empresa

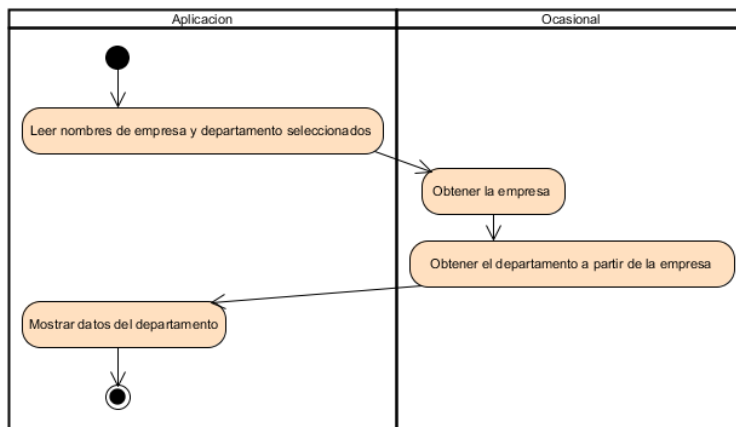


Diagrama de actividades - Visualizar departamento

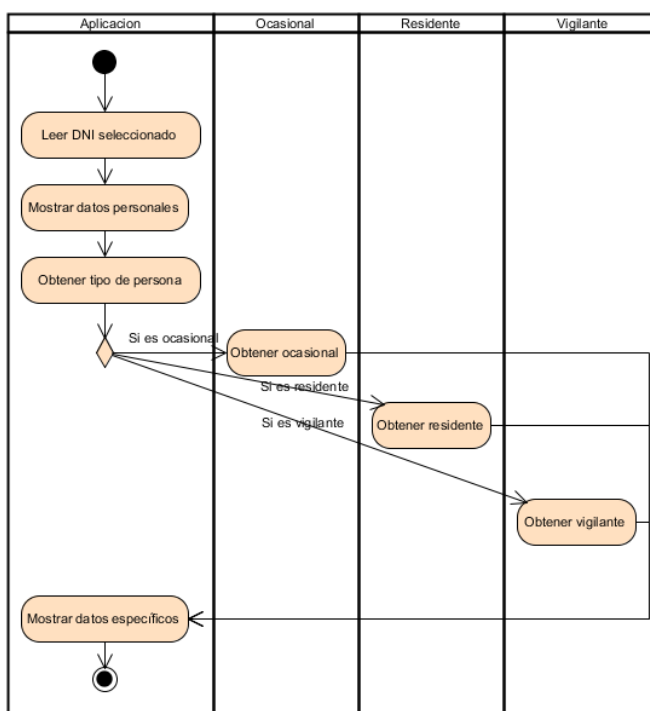


Diagrama de actividades - Visualizar personal (empleados)

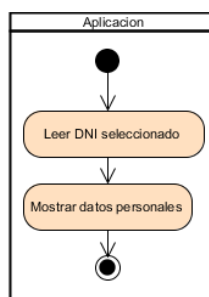


Diagrama de actividades - Visualizar personal (contactos)

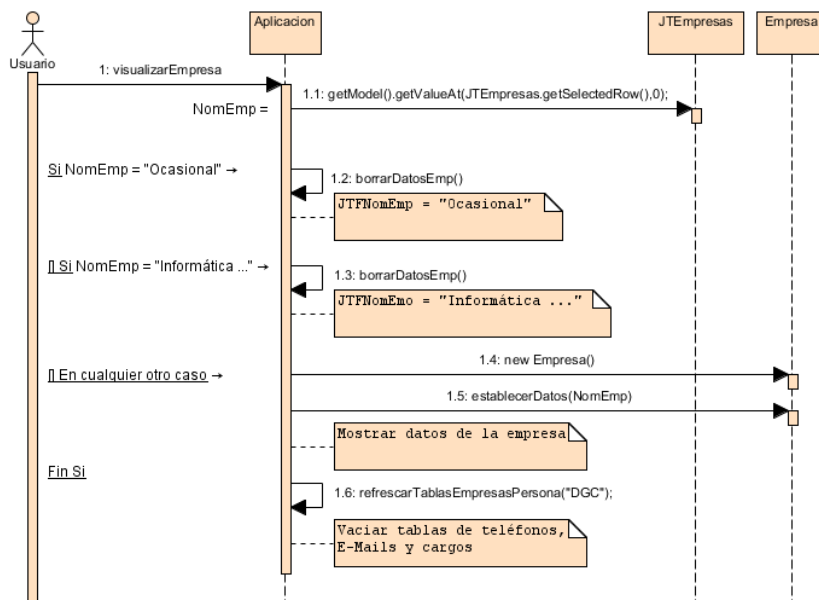


Diagrama de secuencia - Visualizar empresa

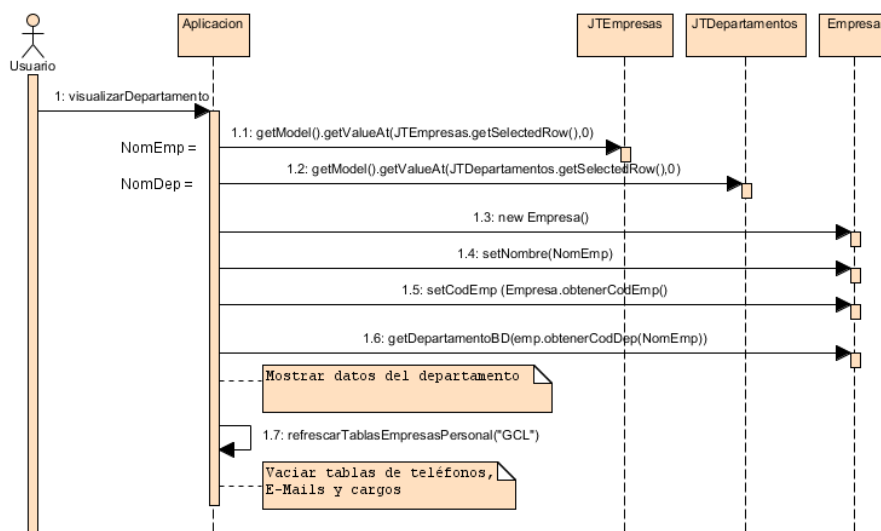


Diagrama de secuencia - Visualizar departamento

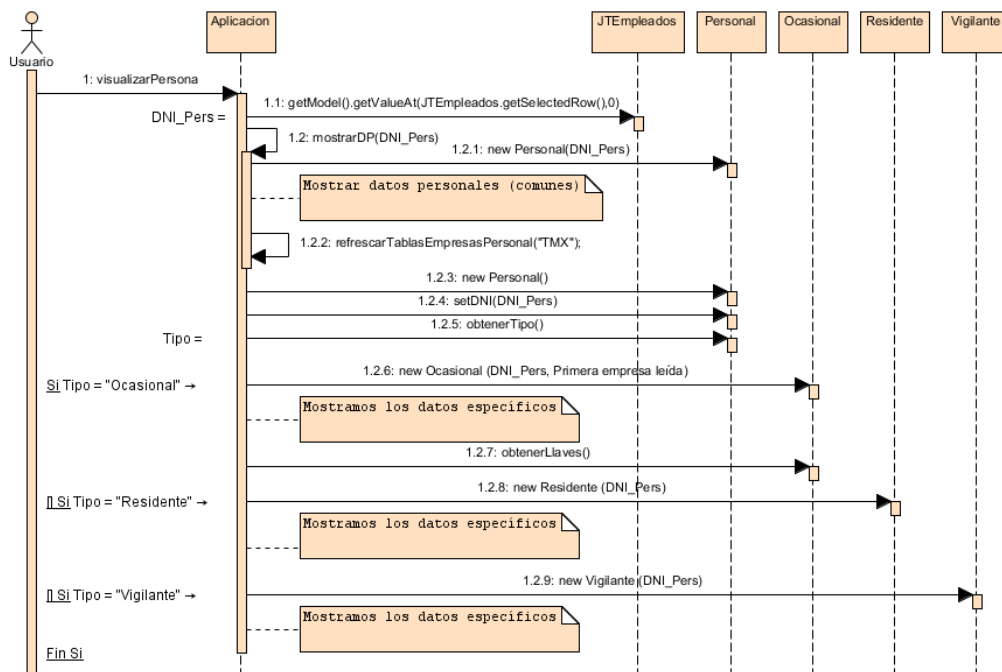


Diagrama de secuencia - Visualizar personal (empleados)

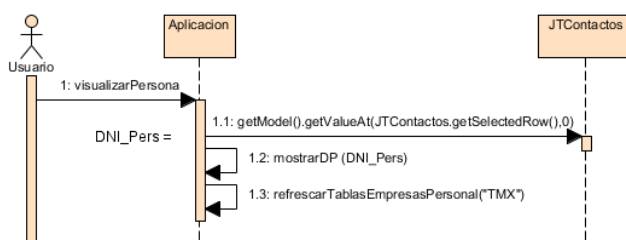


Diagrama de secuencia - Visualizar personal (contactos)

Esperamos que el lector quede satisfecho con las explicaciones proporcionadas. En caso de no ser así le aconsejamos que acuda y consulte el código fuente de la aplicación, que ha sido documentado de manera minuciosa.



4.3. Caso de uso 15: Añadir empresa

El haber comenzado la fase de diseño con todas las visualizaciones implica que a lo largo de los siguientes casos de uso sean mínimos los cambios que debamos realizar en la interfaz gráfica. Para empezar, añadiremos una botonera en la parte inferior de la tabla de empresas que nos permitirá realizar todas las operaciones oportunas sobre éstas. Añadimos a la botonera el botón de adición...

La imagen muestra una propuesta de interfaz gráfica para una aplicación web titulada 'Empresas y personal'. El diseño incluye:

- Una barra superior con campos de texto para 'CF', 'Nombre', 'Nombre', 'Actividad' y un botón 'Todo'.
- Una sección central con un formulario 'Empresas' y un formulario 'Departamentos'.
- Una sección inferior con un formulario 'Datos de la empresa', un formulario 'Datos del departamento' y un botón 'Llaves'.
- Una sección de 'Filtro' con un campo de texto.
- Una sección de 'Empleados' y 'Contactos' con un formulario 'Datos de los empleados y contactos', un botón 'Teléfonos' y un botón 'E-Mails'.
- Una sección inferior con un formulario 'Datos comunes', un formulario 'Datos del residente', un formulario 'Datos del ocasional', un formulario 'Datos del vigilante' y un botón 'Llaves'.

Interfaz gráfica propuesta

... que desencadenará el siguiente proceso:

1. Crear un objeto de tipo Empresa a partir de los parámetros leídos.
2. Insertar la nueva empresa.
3. Refrescar toda la pestaña para así dejarla en el estado inicial (consistente) ya que si refrescásemos solo éstas si en las demás disponemos del resultado de algún tipo de búsqueda el estado de las mismas podría inducir a error.

No disponemos de ningún método nuevo a añadir a la API, ya que el constructor y los métodos de inserción y refresco fueron creados con la API inicial (los dos primeros) y el primer caso de uso de esta iteración (el restante).



Se trata de un proceso muy sencillo que puede resumirse en este diagrama de actividades...

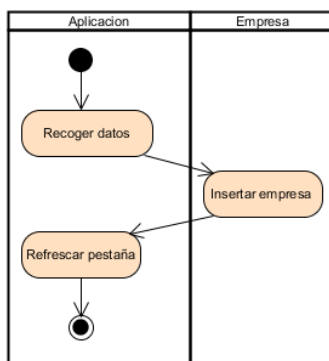


Diagrama de actividades

... y este otro de secuencia:

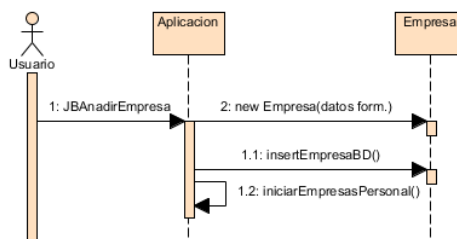


Diagrama de secuencia

4.4. Caso de uso 16: Borrar empresa

Para eliminar un conjunto de empresas sólo necesitaremos de un botón de borrado que será añadido a la botonera creada en el anterior caso de uso.

La estrategia de borrado es sencilla y consistirá en recorrer todas las empresas seleccionadas, borrarlas una a una y terminar refrescando toda la pestaña para así eliminar de todas las tablas todos los elementos que resulten eliminados por el borrado en cascada que se desencadenará:

1. Obtener las empresas seleccionadas.
2. Recorrerlas eliminándolas una a una.
3. Refrescar todas las tablas que conforman la pestaña para que los cambios (borrado) se hagan visibles.



La obtención de filas seleccionadas es proporcionada por la clase `JTable`, la iteración entre el conjunto de empresas será realizada con la ayuda de la clase `Iterator` y llevaremos a cabo el refresco con el método creado en el primer caso de uso. Solo nos queda la forma de obtener una a una las empresas para ir borrándolas. Lo más sencillo e interesante es crear un constructor que nos permita ir recuperándolas a través de su clave primaria, que será obtenida a partir del nombre leído y el método `obtenerCodEmp` ya diseñado. Con todo esto nos quedaría añadir a la API el constructor mencionado:

MÉTODO	FUNCIONALIDAD
<i>Empresa</i>	Constructor que crea una Empresa a partir del código que obtiene como parámetro de entrada.

Método a implementar y a añadir a la API

El flujo de actividades y la secuencia de mensajes a enviar entre objetos quedan definidos en los siguientes diagramas:

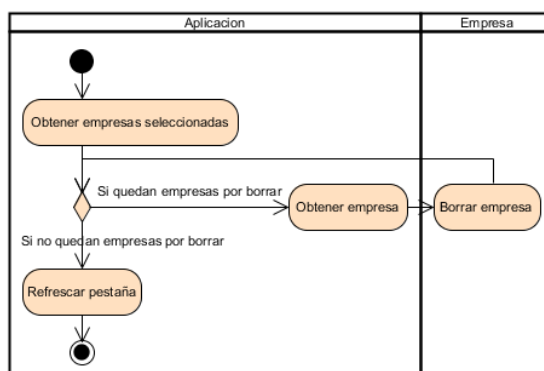


Diagrama de actividades

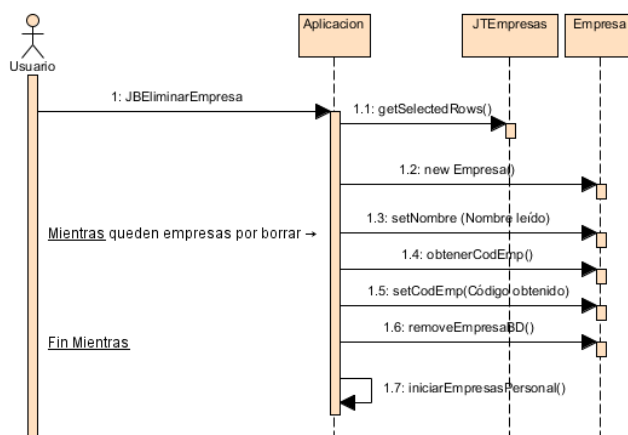


Diagrama de secuencia



4.5. Caso de uso 17: Modificar empresa

La modificación de una empresa, como todas las acciones que en accessManager tienen lugar, debe ser iniciada por un estímulo, y este estímulo se materializa en un botón que será añadido a la botonera de la tabla empresas.

Para modificar una empresa nos basaremos en el método *updateEmpresaBD* creado en la API inicial, que modifica en la base de datos una empresa con otra que posee los datos finales:

1. Crear la nueva empresa con los datos introducidos por el usuario en el formulario.
2. Recuperar la vieja empresa de la base de datos a partir de su nombre.
3. Actualizar la empresa vieja con la nueva.
4. Refrescar todas las tablas que conforman la pestaña.

Todos son acciones muy simples que emplean métodos ya creados y que conforman la siguiente secuencia de actividades y de paso de mensajes:

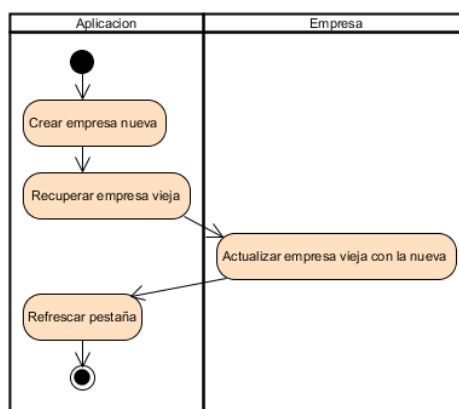


Diagrama de actividades

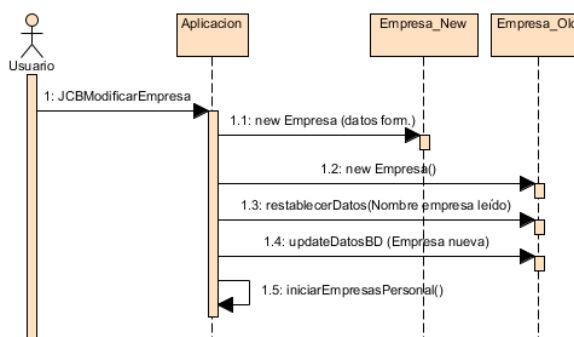


Diagrama de secuencia



4.6. Caso de uso 21: Añadir departamento

Cambiamos de elementos y pasamos de las empresas a los entes que las forman: los departamentos.

Antes que nada atenderemos a la interfaz gráfica. Disponemos de una forma de visualizar los departamentos y sus datos, pero nos faltaría por añadir un componente gráfico que indique al sistema que dispone de un nuevo departamento a registrar. Como siempre ese componente será un botón que estará encuadrado en una botonera bajo la tabla de departamentos:

Empresas y personal

CIF Nombre Nombre Actividad Todo

Empresas Departamentos

Botones Botones

Datos de la empresa Datos del departamento Llaves

Filtro

Empleados Contactos

Datos de los empleados y contactos

Teléfonos

E-Mails

Datos comunes Datos del residente Datos del ocasional Llaves

Datos del vigilante

Interfaz gráfica propuesta

La adición de nuevos departamentos sigue las mismas pautas que las empleadas con las empresas:

1. Obtener la empresa seleccionada.
2. Crear un Departamento nuevo a partir de los datos recogidos del formulario.
3. Insertarlo asignándolo a la empresa en la base de datos.
4. Refrescar las tablas de departamentos, empleados, contactos y llaves prestables a un departamento (en un principio vacía).



No observamos que sea necesario crear ningún nuevo método a añadir a la API. Únicamente deberemos implementar en el evento que genera la pulsación del botón en la API una secuencia de sentencias que lleven a cabo el flujo de actividades descrito en los siguientes diagramas UML:

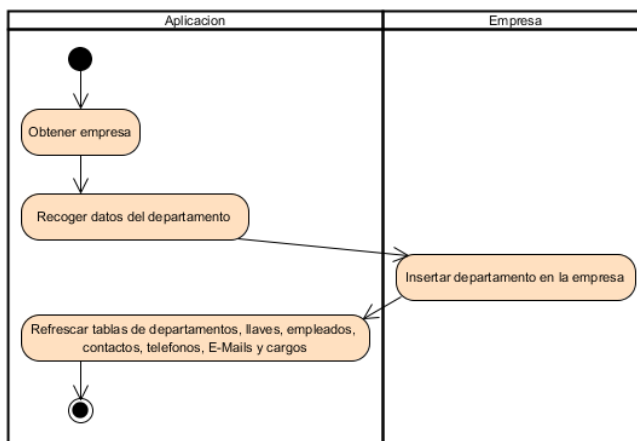


Diagrama de actividades

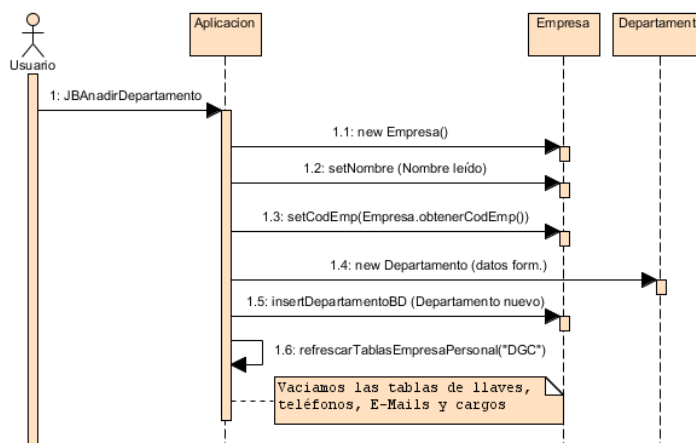


Diagrama de secuencia



4.7. Caso de uso 22: Borrar departamento

Proseguimos con el caso de uso 22, que nos lleva a desarrollar una estrategia con la que poder borrar departamentos cuyo almacenamiento carezca de sentido. Este borrado puede ser planteado de la misma forma que planteamos el de empresas salvo por la novedad de que el borrado se realiza desde la empresa y no desde el propio objeto Departamento:

1. Obtener la empresa seleccionada.
2. Obtener los departamentos seleccionados.
3. Recorrerlos uno a uno procediendo con su borrado desde el objeto Empresa creado en el primer paso.
4. Refrescar toda la pestaña ya que el borrado de departamentos afectará a su personal pero también podrá hacerlo a su propia empresa si es el último departamento que ésta posee.

Ni que decir tiene que los componentes gráficos necesarios son las tablas en las que seleccionar la empresa y los departamentos (ya creadas) y un botón que inicie el proceso de borrado (éste sí que deberemos crearlo y añadirlo a la botonera de departamentos).

El flujo de actividades descrito anteriormente podría quedar plasmado en estos diagramas:

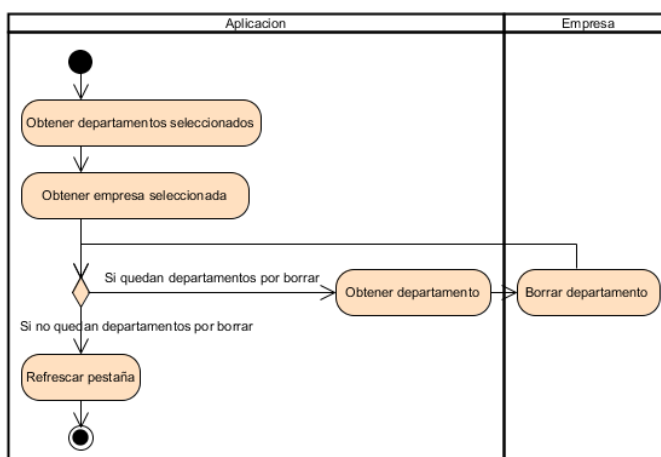
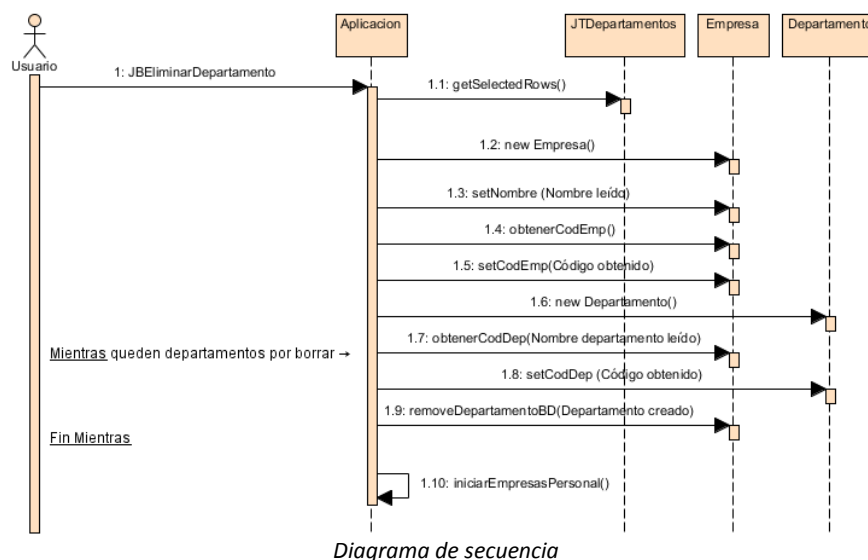


Diagrama de actividades



4.8. Caso de uso 23: Modificar departamento

Desdoblaremos este caso de uso en modificar los datos de un departamento y el conjunto de llaves al que su personal puede optar en el centro.

En primer lugar nos centramos en modificar los datos que componen cada departamento. En lo que respecta a la interfaz gráfica solo nos queda añadir un botón de modificación, pues el formulario y la tabla ya las tenemos diseñadas.

Añadido el botón a la botonera pasamos a crear la estructura que guíe todo el caso de uso:

1. Obtener la empresa cuyo departamento vamos a modificar.
2. Crear el departamento nuevo con los datos leídos del formulario.
3. Recuperar el departamento viejo a través del objeto Empresa creado en el punto 1 y el nombre leído de la tabla JTDepartamentos.
4. Actualizar el departamento viejo con el nuevo.
5. Refrescar la tabla de departamentos (el resto es innecesario).



Como siempre, acompañamos esta explicación del correspondiente diagrama de actividades...

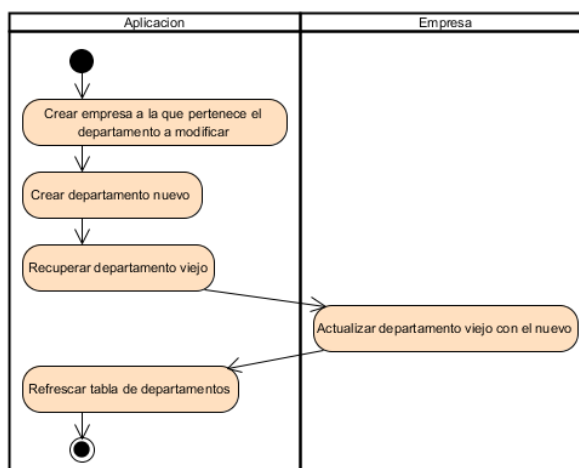


Diagrama de actividades

... y de secuencia:

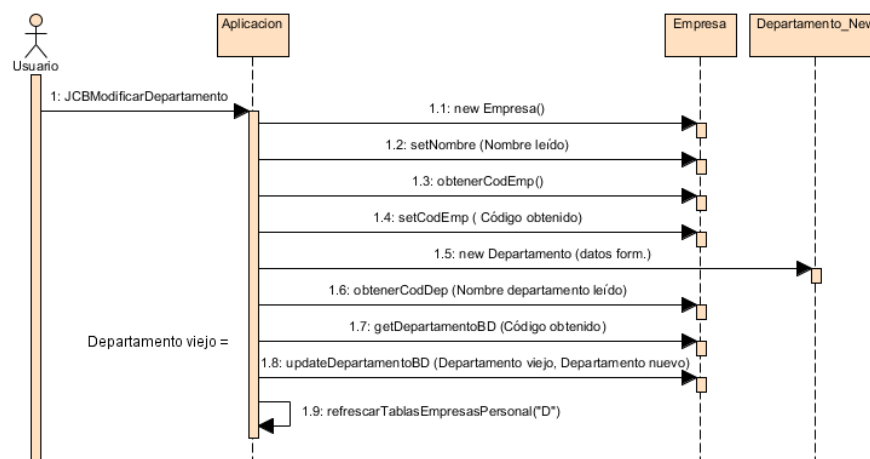


Diagrama de secuencia



Tras tener definido la modificación de los datos que detallan cada departamento debemos desviar nuestra atención a las llaves que su personal puede pedir prestadas en el acceso al centro. Tenemos creada la tabla de muestra, pero no los botones que permitan añadir y eliminar llaves prestables. La botonera que acoja estos botones será colocada bajo la tabla de llaves:

Interfaz gráfica propuesta

Ahora tenemos que pensar en cómo añadir y eliminar llaves. La segunda opción es sencilla: con seleccionarlas y pulsar el botón de borrado nos bastará. El problema viene con la primera. Para añadir llaves a la tabla necesitamos poder visualizarlas, seleccionarlas y cargarlas. Una muestra en la misma pestaña resultaría descabellada con lo cargada que ya la tenemos, con que optamos por crear un diálogo que se encargue de la adición de llaves:

Interfaz gráfica propuesta



De esta forma, cada vez que el usuario quiera añadir llaves a un departamento lo que hará será pulsar el botón de adición de llaves (hecho que desplegará este diálogo), a continuación seleccionará las llaves que desee asignar y realizará la asignación pulsando el botón diseñado para ello.

¿Quién se encargará de hacer posible todo este proceso? El *frame* principal de la aplicación se encargará de desplegar el diálogo y refrescar la tabla de llaves tras realizar la asignación y el diálogo se encargará de recoger las llaves a insertar y asignárselas al departamento en la base de datos. Paso a paso tendremos:

1. Obtener los códigos de empresa y departamento señalados en la interfaz.
2. Desplegar el diálogo de llaves.
3. Recoger las llaves seleccionadas.
4. Asociarlas una a una al departamento en la base de datos.
5. Esconder el diálogo.
6. Refrescar la tabla de llaves para que ésta muestre las que antes poblaban más las seleccionadas por el usuario (recientemente insertadas).
7. Eliminar el diálogo.

La inicialización del diálogo de llaves requerirá del siguiente método para rellenar la tabla con las llaves debidas:

MÉTODO	FUNCIONALIDAD
<i>obtenerLsPorPrestarDep</i>	Método ejecutado por una Empresa y que obtiene el conjunto de llaves que existen y aún no han sido asignadas como prestables al departamento que recibe como parámetro de entrada y que además pasan el filtro especificado por el usuario.

Método a implementar y a añadir a la API



Quizá pueda parecer un proceso algo más complicado que los anteriores debido a que se realiza entre varios componentes gráficos (*frame* principal y diálogo de llaves), pero no es así y buena fe de ello dan los siguientes diagramas:

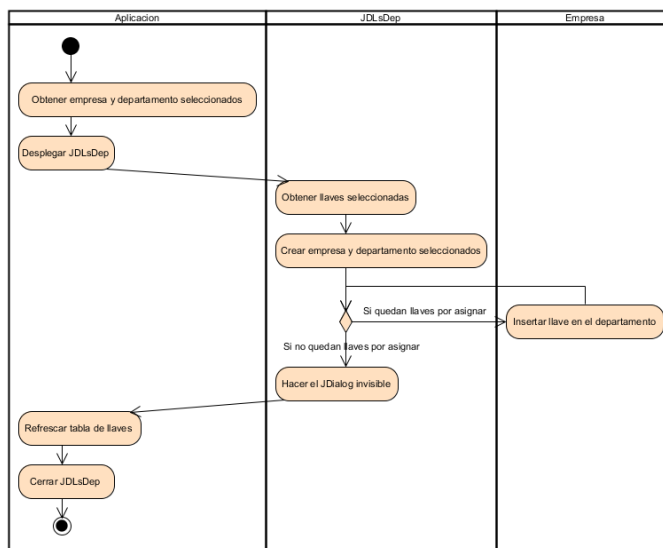


Diagrama de actividades

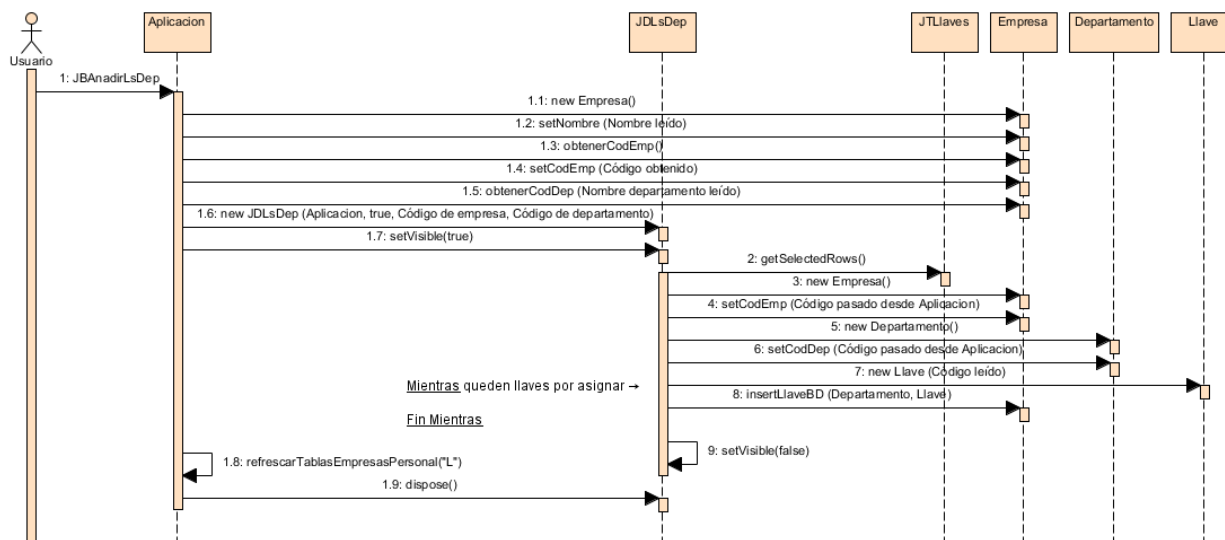


Diagrama de secuencia



Restaría solo el borrado de llaves, que puede realizarse desde la misma pestaña seleccionando las llaves que deseemos borrar y clicando el botón establecido para tal fin, que desencadenará el siguiente flujo de actividades:

1. Obtener las llaves seleccionadas.
2. Obtener la empresa y departamento en el que borrar/desasignar.
3. Eliminar una a una todas las llaves.
4. Refrescar la tabla de llaves para que los cambios surtan efecto visible.

Esta estrategia también puede ser definida con más detalle mediante diagramas:

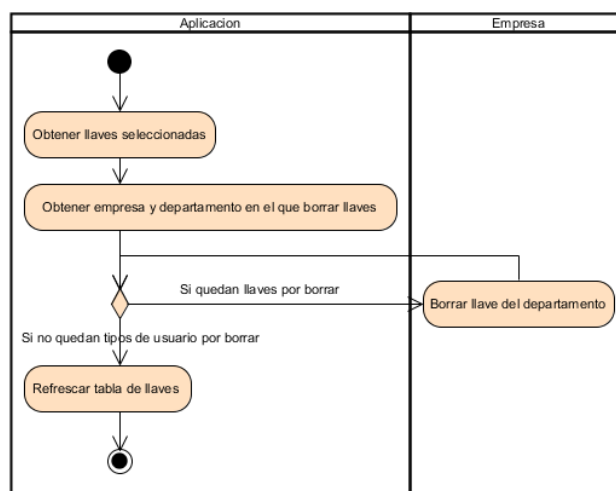


Diagrama de actividades

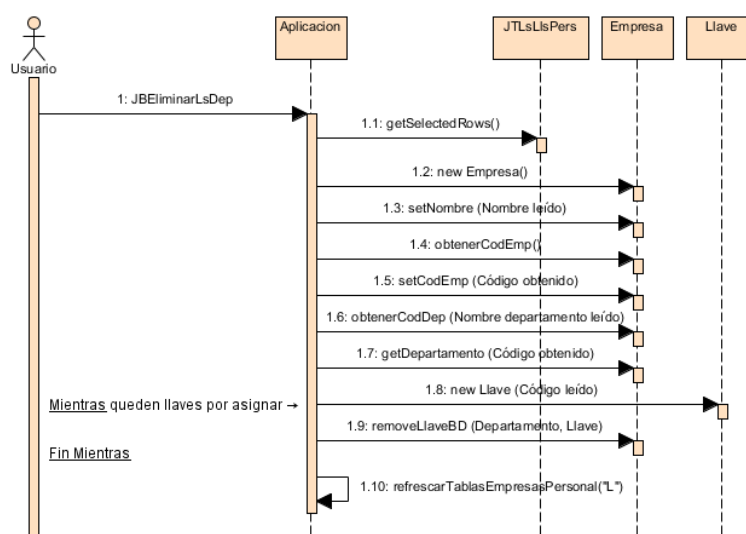


Diagrama de secuencia



4.9. Caso de uso 27: Añadir persona

Con el grupo de casos de uso relativos a personas concluiremos la fase de diseño prevista. La implementación que a continuación veremos deberá ser meticulosa debido al variable número de tipos de usuarios existentes, pero con calma y atención conseguiremos alcanzar nuestro objetivo.

Para empezar nos centramos en proponer una vía para añadir personas al sistema. Desde el diseño del caso de uso '26 - Visualizar persona' disponemos de un conjunto de paneles que permiten visualizar cualquier tipo de persona. Con la ayuda de éstos y un botón de adición podremos conseguir el objetivo que nos proponemos:

Empresas y personal

Formulario de búsqueda y gestión de empresas y personal. Incluye campos para 'CIF', 'Nombre', 'Actividad' y un botón 'Todo'. Sección de 'Empresas' con un campo de texto y un botón 'Botones'. Sección de 'Departamentos' con un campo de texto y un botón 'Botones'. Sección de 'Datos de la empresa' y 'Datos del departamento'. Sección de 'Filtro'. Sección de 'Empleados' y 'Contactos' con un campo de texto 'Datos de los empleados y contactos', un botón 'Teléfonos', un botón 'Botones', un botón 'E-Mails'. Sección de 'Datos comunes', 'Datos del residente', 'Datos del ocasional', 'Datos del vigilante' y un botón 'Llaves'.

Interfaz gráfica propuesta



El modo de actuar será el siguiente:

1. Recoger los datos personales, teléfonos, E-Mails y cargos (en caso de ser necesario) proporcionados por el usuario.
2. Obtener el nombre de empresa seleccionado y en función de ella terminar de recoger los datos específicos del tipo de persona que se va a insertar, crear el objeto que corresponda e insertarlo.
En concreto tendríamos las siguientes bifurcaciones:
 - a. Si es "Ocasional" → Crear objeto Ocasional e insertarlo.
 - b. Si es "Informática Navarra S.A." → Crear el objeto Residente e insertarlo.
 - c. Si es "Compañía Integral de Seguridad" → Obtener la empresa y departamento, crear un objeto Vigilante o Periodico_Contacto (en función de si el panel que se muestra es el de empleados o contactos) e insertarlo. Si lo que insertamos es un contacto ('RP' por defecto) se le creará una cuenta de sistema.
 - d. En cualquier otro caso → Crear el objeto Periodico_Contacto, obtener la empresa y departamento al que corresponde e insertarlo en ésta, creando una cuenta de sistema si lo que introducimos es un nuevo contacto, que nuevamente será 'RP' por defecto.
3. Para terminar refrescaremos las tablas de empleados y contactos para que así aparezca la nueva persona insertada en el sistema y vaciaremos las tablas de teléfonos, E-Mails y cargos para dejarlas en su estado inicial.

Complicado a primera vista, el proceso no entraña más dificultad que la de poder obtener correctamente los valores de los componentes gráficos que componen los diversos formularios. Utilizaremos en todo momento métodos proporcionados por la API de Swing o ya creados en anteriores casos de uso salvo uno, *obtenerCodCargo*, que deberemos crearlo y añadirlo a la nuestra API en estos momentos:

MÉTODO	FUNCIONALIDAD
<i>obtenerCodCargo</i>	Método que obtiene el código de un cargo recogiendo como parámetro de entrada la descripción del mismo.

Método a implementar y a añadir a la API



En esta ocasión sí que va a resultar de gran utilidad diseñar los diagramas de actividades y de secuencia correspondientes ya que nos ayudarán a aclarar y profundizar en toda la secuencia de actividades y bifurcaciones que deberemos implementar:

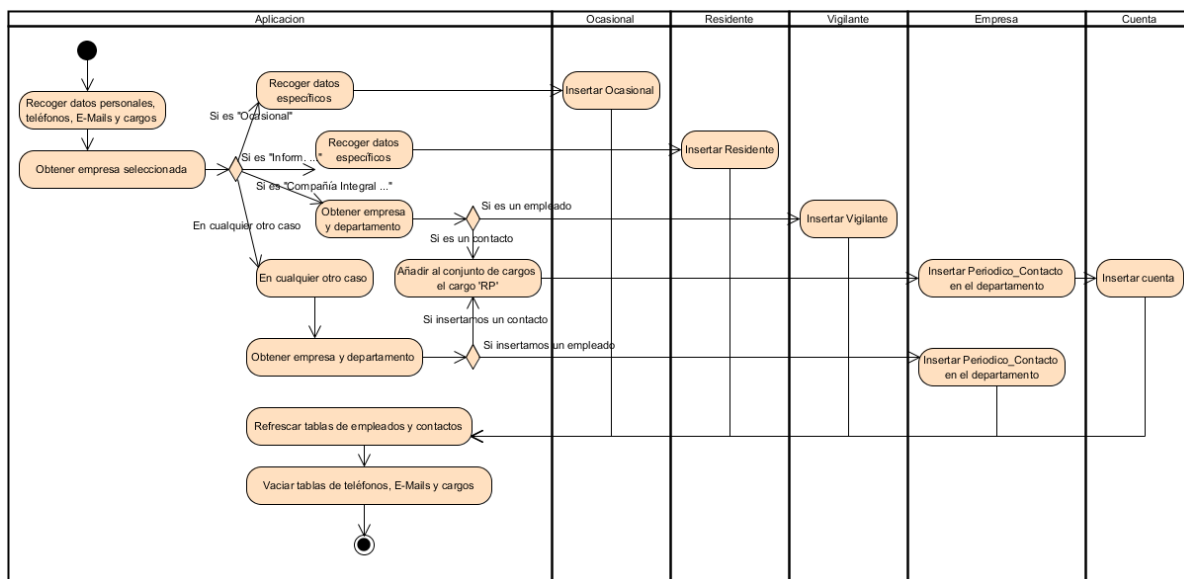


Diagrama de actividades

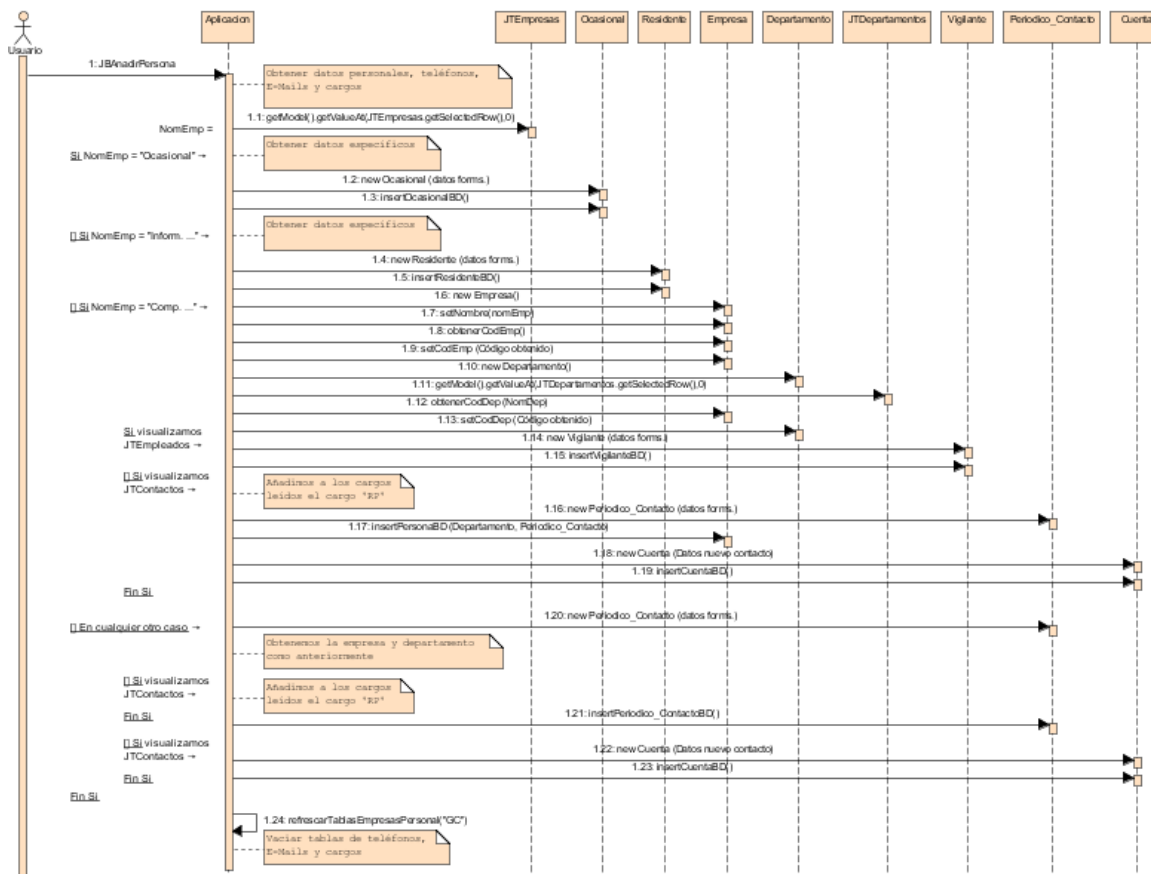


Diagrama de actividades



4.10. Caso de uso 28: Borrar persona

El borrado de personas también dependerá del tipo de persona a eliminar. Para ejecutarlo bastará con seleccionar de la interfaz gráfica el conjunto de personas que queremos que dejen de formar parte del sistema y pulsar un botón que añadiremos a la botonera creada en el anterior caso de uso. La pulsación de éste desencadenará el siguiente proceso:

1. Obtener personas seleccionadas en función del panel visible: empleados o contactos.
2. Recoger el nombre de la empresa seleccionada y actuar en función de ella:
 - a. Si es “Ocasional” → Crear un objeto Ocasional y recorrer todo el conjunto de personas seleccionadas obteniendo su DNI, asignándolo al objeto creado y eliminándolo.
 - b. Si es “Informática Navarra S.A.” o “Compañía Intergal de Seguridad” → Crear un objeto de tipo Personal y recorrer las personas seleccionadas obteniendo su DNI, asignandoselo al objeto Personal e invocando el método de borrado sobre éste.
 - c. En cualquier otro caso → Obtener la empresa y departamento clicados, recorrer el conjunto de personas seleccionadas obteniendo los objetos Periodico_Contacto que las representan e ir eliminándolas del departamento una a una.
3. Al final refrescar toda la pestaña para que se hagan visibles todos los cambios que han tenido lugar en la base de datos.

En esta secuencia de acciones surge un constructor que no podemos olvidar añadir a nuestra API:

Con el fin de poder recuperar a todo el personal debemos proveer a nuestra API de métodos constructores que permitan crear objetos a partir de las claves primarias de éstos:

MÉTODO	FUNCIONALIDAD
<i>Ocasional</i>	Constructores que crear un objeto de la clase a la que representan tomando como parámetros de entradas las claves primarias de las tablas a las que representan en la base de datos y apoyándose en la información contenida por ésta.
<i>Residente</i>	
<i>Vigilante</i>	
<i>Periodico_Contacto</i>	

Métodos a implementar y a añadir a la API



Disponemos de todo lo necesario para poder codificar en la siguiente fase estos diagramas de actividades y de secuencia:

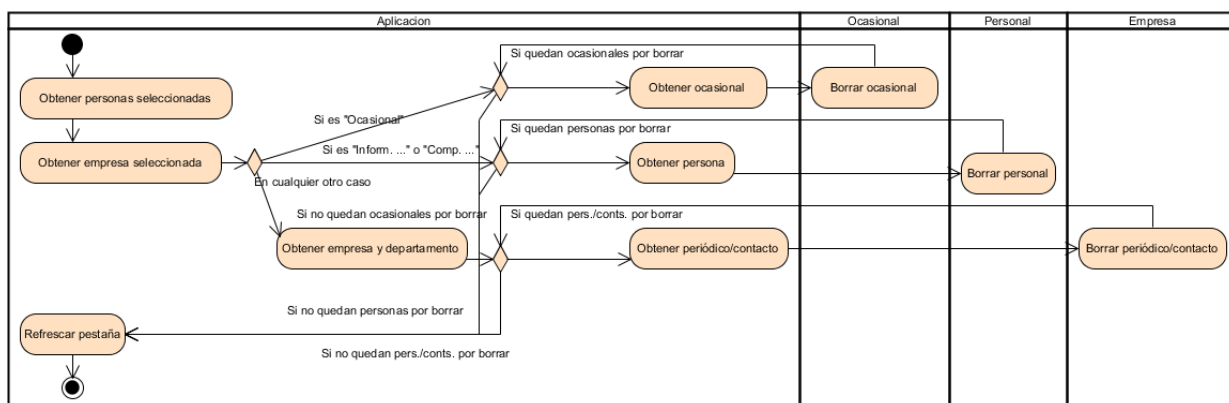


Diagrama de actividades

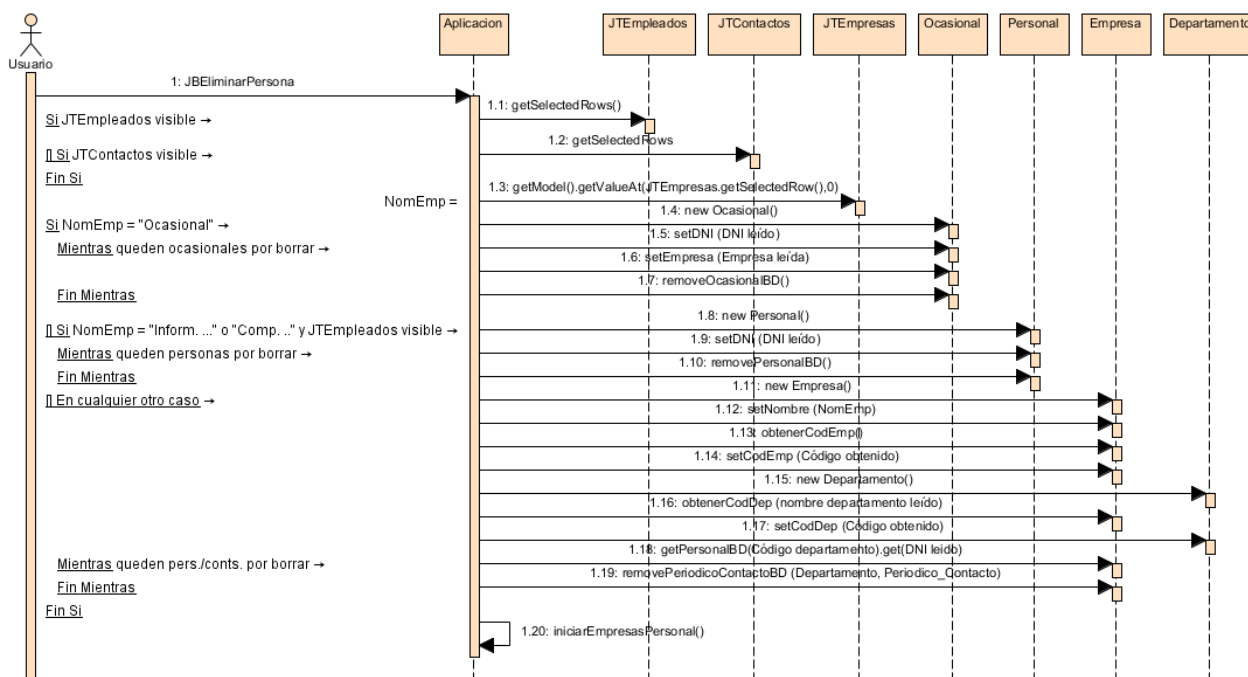


Diagrama de secuencia

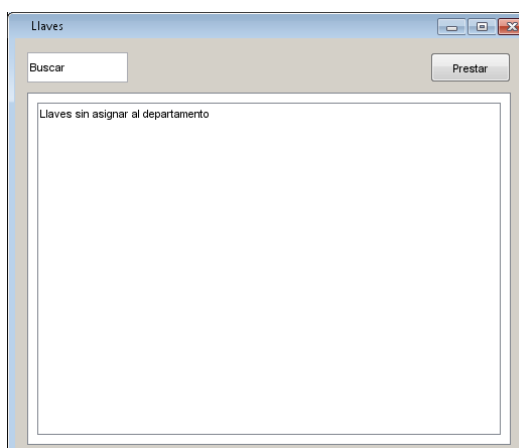


4.1.1. Caso de uso 29: Modificar persona

Nos encontramos ante un caso de uso muy similar al de añadir una persona, pues deberemos crear las mismas estructuras de control que las creadas entonces, solo que en lugar de insertar los objetos creados deberemos recuperar aquellos con los datos existentes en la base de datos y actualizarlos con los nuevos creados.

Dado que el proceso es casi idéntico al de añadir persona optamos por no re-explicarlo, haciendo hincapié en que debemos intercambiar las operaciones de inserción por las de recuperación de los objetos viejos y actualización de éstos con los nuevos.

Hasta aquí tendríamos cubierta la modificación de los datos personales y específicos de cualquier tipo de persona. Nos quedaría por tratar la adición y eliminación de las llaves que pueden ser prestadas al personal ocasional. Al igual que pasaba con los departamentos, la visualización y eliminación se realizará mediante la tabla destinada a tal fin en el panel de datos específicos del personal ocasional, pero para la adición nos basaremos en un diálogo idéntico al empleado para los departamentos pero que será inicializado de distinta forma (mediante el paso del DNI y empresa a la que representa la persona ocasional, no los códigos de empresa y departamento) ya que la estructura de la base de datos lo requiere:



Interfaz gráfica propuesta

El método que debemos añadir a la API y que nos permitirá rellenar la tabla de llaves en la inicialización es el siguiente:

MÉTODO	FUNCIONALIDAD
<i>obtenerLsPorPrestarOcas</i>	Método ejecutado por un Ocasional que obtiene el conjunto de llaves que existen y aún no le han sido asignadas como prestables y que además pasan el filtro de búsqueda especificado por el usuario.

Método a implementar y a añadir a la API



A la hora de inicializar ambos diálogos de llaves (el de departamentos y el de personal ocasional) se empleará *obtenerLsPorPrestarXXX* con la cadena vacía como filtro, pero el usuario podrá redefinir el contenido de la tabla de llaves a través del campo de búsqueda ubicado en la parte superior izquierda.

El flujo de actividades y la estructura de control que empleemos serán los mismos que los empleados para añadir llaves a los departamentos, solo que en lugar de trabajar con registros "*Cod_Llave,F_Alta_L,Cod_Emp,Cod_Dep*" lo haremos con registros "*Cod_Llave,F_Alta_L,DNI,Empresa*".

Si el lector desea conocer cómo realizar el borrado de llaves de una persona ocasional no tiene más que acudir a la sección 4.10 e imaginar el mismo proceso pero sobre el objeto Ocasional en lugar del objeto Empresa.

Para terminar nos quedaría poder modificar los teléfonos e E-mails de cualquier tipo de persona, así como los cargos que ejerce cualquier contacto. Acometemos esta tarea a lo largo de los siguientes tres apartados.

4.12. Casos de uso 30 y 32: Añadir teléfono o E-Mail

Un teléfono o E-Mail está compuesto por una serie de letras y números que pueden ser recogidos con un simple campo de texto. Añadiremos éste al panel de datos comunes, ya que cualquier persona puede disponer de teléfonos o E-Mails, y lo acompañaremos de dos botones de adición, uno para cada tipo de elemento.

Atendiendo a la modificación de datos personales, teníamos que al comienzo del caso de uso recogíamos los teléfonos e E-Mails para crear el objeto que representaba a la persona con los datos actuales. Si somos un poco inteligentes podremos ayudarnos del proceso de modificación anteriormente creado para añadir los teléfonos o E-Mails que estimemos oportunos a cualquier persona.

Al ser los teléfonos e E-Mails recogidos de las tablas a la hora de realizar la actualización, si conseguimos crear un método que introduzca los datos escritos por el usuario en la interfaz gráfica (tablas), luego solo restará que el proceso de actualización los recoja.



Para introducir datos en las tablas no necesitamos añadir nada a la API ya que con la funcionalidad proporcionada por el objeto *DefaultTableModel*, con el que podemos obtener y manejar el contenido de las JTables, será más que suficiente. Esto será lo que haremos para añadir teléfonos:

1. Obtener el modelo actual de la tabla.
2. Si la String con el teléfono tiene un formato válido crearemos un objeto *Vector* con ésta y lo añadiremos al modelo con el método *addRow*.
3. Asignar a la tabla el modelo que acabamos de crear, lo que provocará su actualización.
4. Borrar el campo de texto para la próxima inserción.
5. Invocar el proceso de actualización definido con anterioridad.

Sencillo, ¿verdad? Con los E-Mails realizaremos exactamente lo mismo, cambiando únicamente la sentencia de validación de la String y la tabla de la que obtener y restaurar el modelo.

El proceso de modificación resulta tan sumamente sencillo que la realización de sus respectivos diagramas resulta ridículo:

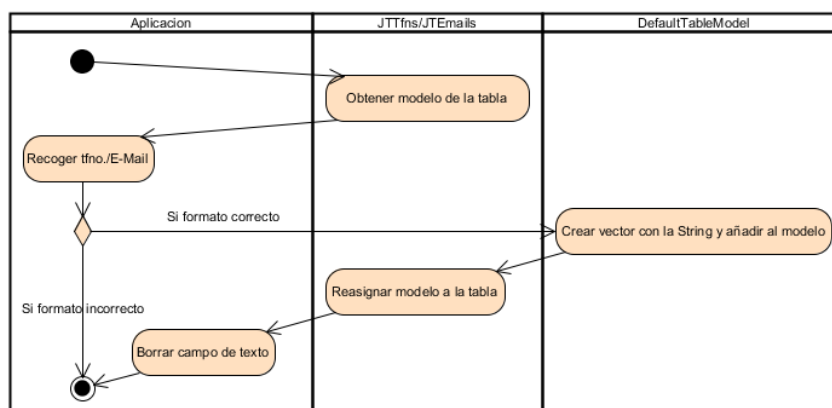


Diagrama de actividades

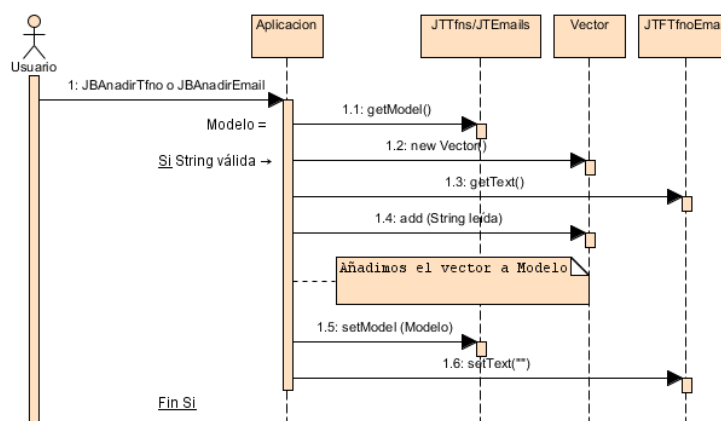


Diagrama de secuencia



4.13. Casos de uso 31 y 33: Borrar teléfono o E-Mail

Para borrar teléfonos e E-Mails también utilizaremos la misma estrategia y nos volveremos a basar en el proceso de modificación. Si antes necesitábamos encontrar una vía con la que añadir filas a una tabla, lo que ahora tendremos que hacer es eliminarlas. Para ello añadiremos un botón de borrado en cada tabla:

Interfaz gráfica propuesta

Si para añadir elementos nos basamos en un campo de texto y el método *addRow*, para eliminarlos únicamente nos basaremos en la selección que el usuario realice y el método *removeRow* para:

1. Obtener las filas seleccionadas y el modelo que representa la tabla actualmente.
2. Eliminar una a una todas las filas, comenzando de la de mayor índice para terminar en la de menor, manteniendo así en todo momento los índices correctos.
3. Restablecer la tabla con el modelo resultante tras la eliminación de las filas seleccionadas.



La complejidad brilla por su ausencia y más aún si realizamos los diagramas de actividades y de secuencia que describan todo este proceso:

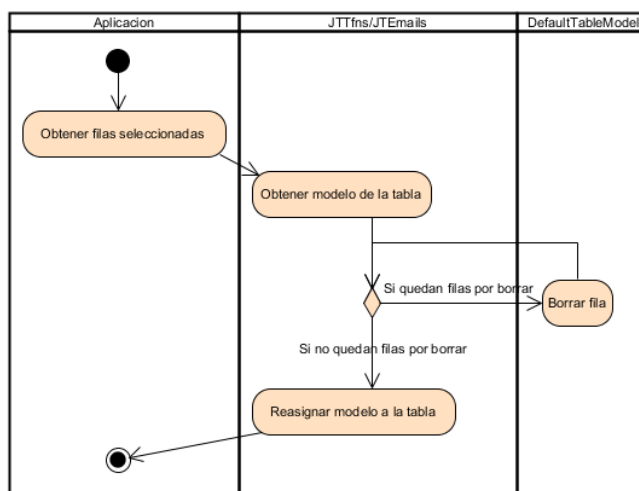


Diagrama de actividades

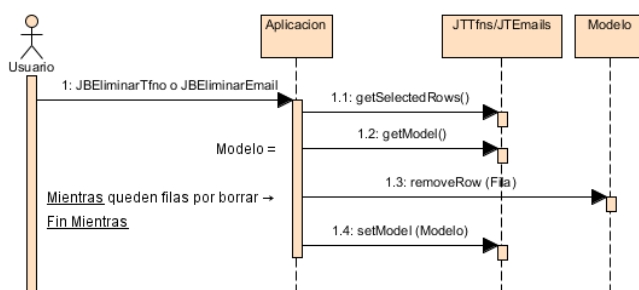


Diagrama de secuencia



4.14. Caso de uso 34: Añadir cargo

Este caso de uso también estará encuadrado dentro del proceso de modificación pero solo podrá ser ejecutado con un contacto, no con cualquier tipo de persona.

A la tabla de muestra de cargos que ya disponíamos le sumamos una lista desplegable:

Interfaz gráfica propuesta

La idea a seguir es la misma que la ejecutada con los teléfonos e E-Mails: añadir filas a las tablas que serán leídas a la hora de actualizar las personas. En este caso trabajaremos sobre la de cargos.

En los casos anteriores el contenido de las filas a introducir era aleatorio, con que no había forma de preestablecerlo en un componente gráfico. En este caso en el que los cargos deben estar previamente registrados, optaremos por rellenar el JComboBox añadido con los cargos existentes y hacer que se añadan a la tabla cada vez que uno de ellos sea seleccionado.

El flujo de actividades resulta idéntico al utilizado con los teléfonos e E-Mails variando la tabla sobre la que se ejecuta y el método de obtención del nuevo elemento, que pasa de `getText` a `JCB.getModel().getSelectedItem()`. Los diagramas de actividades y de secuencia son calcados, con que nos ahorramos exponerlos.



4.15. Caso de uso 35: Borrar cargo

Al igual que los cinco últimos casos de uso éste también se encuentra encuadrado en la modificación de un contacto, pero le dedicaremos una breve reseña aparte al ser ejecutado solo con los contactos y por requerir de un nuevo botón que colocaremos a la derecha del JComboBox de cargos:

Interfaz gráfica propuesta

Podemos decir que casi tenemos la interfaz gráfica de esta pestaña perfilada. Para implementar el proceso de borrado seguiremos el mismo diseñado para los teléfonos e E-Mails pero sobre la tabla JTCargos. Por lo demás no creemos oportuno adentrarnos de nuevo en la explicación del mismo.

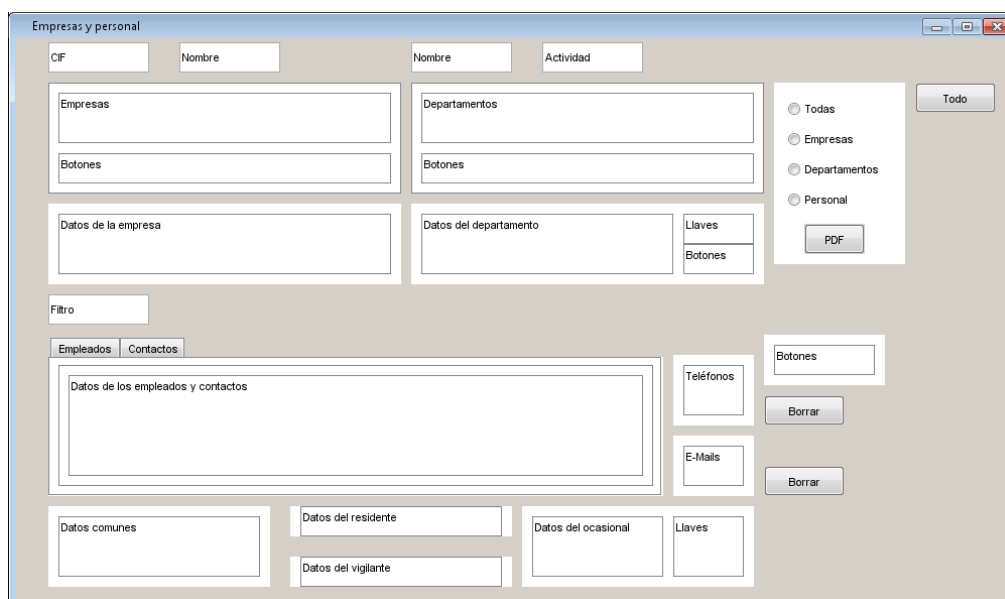
4.16. Casos de uso 18, 24 y 36: Listar empresas, departamentos y personal

Finalizamos el diseño de los casos de uso a implementar en la iteración con el diseño de una vía que permita al usuario generar reportes de las empresas, departamentos y personal que visualiza en pantalla.

Al comienzo de esta iteración elaboramos una serie de estrategias que nos permitirán visualizar el conjunto de empresas, departamentos y personal atendiendo a las filas seleccionadas y los campos de búsqueda existentes. Bien pues, éstos son mecanismo suficientes para delimitar las búsquedas que el cliente desea poder realizar en las tablas y cuya información quisiera obtener en forma de listados, con lo cual lo que debemos hacer es buscar una forma de poder llevarlas a un archivo PDF.



Será buen comienzo empezar diseñando un apartado dentro de la pestaña que permita indicar las tablas que debe contener el informe solicitado al sistema. Teniendo en cuenta que éste puede contener cualquier combinación de las tablas JTEmpresas, JTDepartamentos y JTEmpleados/JTContactos, la experiencia nos hace optar por crear un JButtomGroup como el que sigue:



Interfaz gráfica propuesta

Son muchas las librerías/paquetes Java que permiten crear archivos en formato PDF, entre las que debemos destacar *JasperReports* (junto con *iReport*), *JFreeText* y *iText*. Siendo la primera una herramienta muy reconocida pero excesivamente compleja para los fines que buscamos y las dos segundas de igual calidad pero la tercera más sencilla de manejar, optamos por esta última.¹

La API de la que disponemos carece de cualquier clase o método que genere algún tipo de listado, con que lo primero que tenemos que hacer es crearla y pensar en desarrollar un método que permita generar un reporte en función de las tablas seleccionadas en el JButtomGroup y mediante el empleo de la librería *iText*.

¹ Para una explicación más detallada de la herramienta *iText* acuda al capítulo 4 de la presente memoria.
accessManager · Iteración C4: Empresas y personal · Página 56 de 114



En nuestro caso la clase a añadir recibirá el nombre de *GeneradorInformes* e implementará el siguiente método:

MÉTODO	FUNCIONALIDAD
<i>generarInformeAvanzado</i>	Método que genera un reporte mediante los siguientes parámetros pasados como entrada: <ul style="list-style-type: none">• Título del informe.• Descripción.• Subtítulos.• Tablas reportables (máximo tres tablas).• Tablas a reportar.

Método a implementar y a añadir a la API

Una vez dispongamos de *generarInformeAvanzado* implementado el proceso a ejecutar al pulsar el botón 'PDF' resulta muy sencillo:

1. Desplegar un JFileChooser con el que el usuario determine en qué ubicación del sistema desea recibir el informe.
2. Crear un objeto de tipo GeneradorInformes.
3. Ejecutar *generarInformeAvanzado* pasándole los títulos y subtítulos que deberá contener el reporte, así como las tablas y cuáles de ellas deberán ser incluidas.

Podemos resumir la estrategia anteriormente citada con este diagrama de actividades...

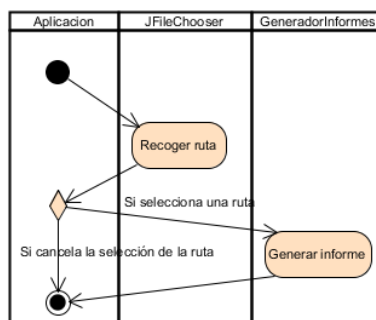


Diagrama de actividades



... y este otro de secuencia:

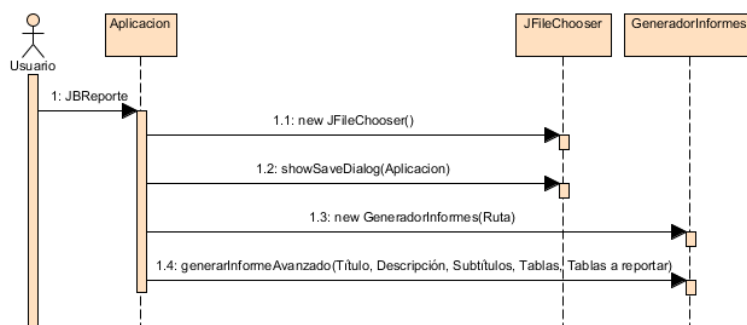
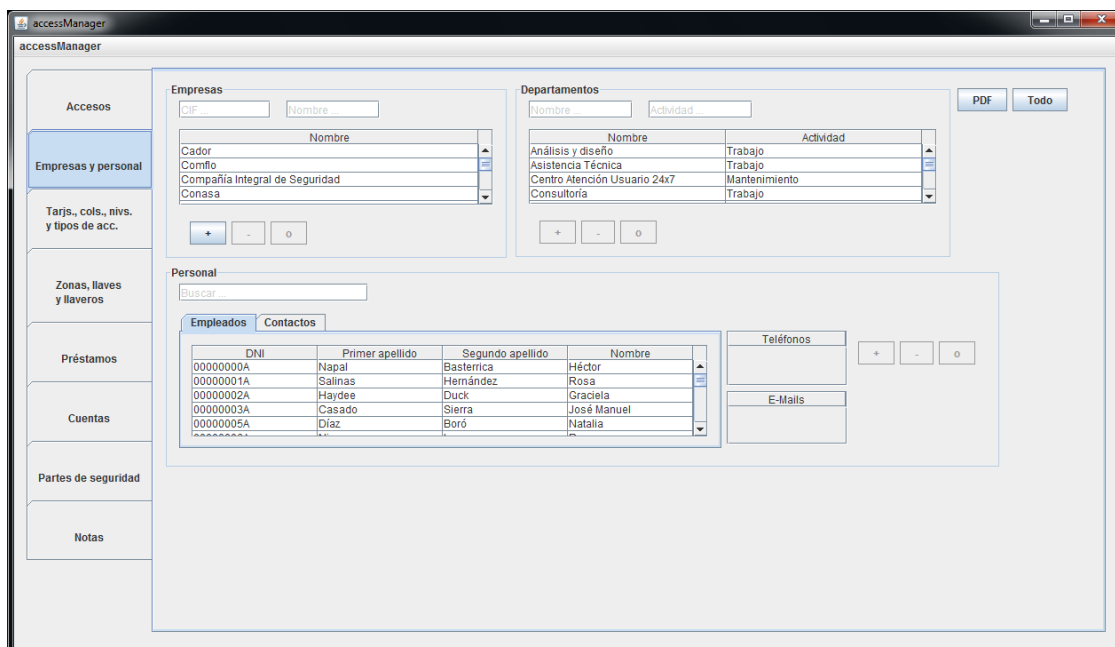


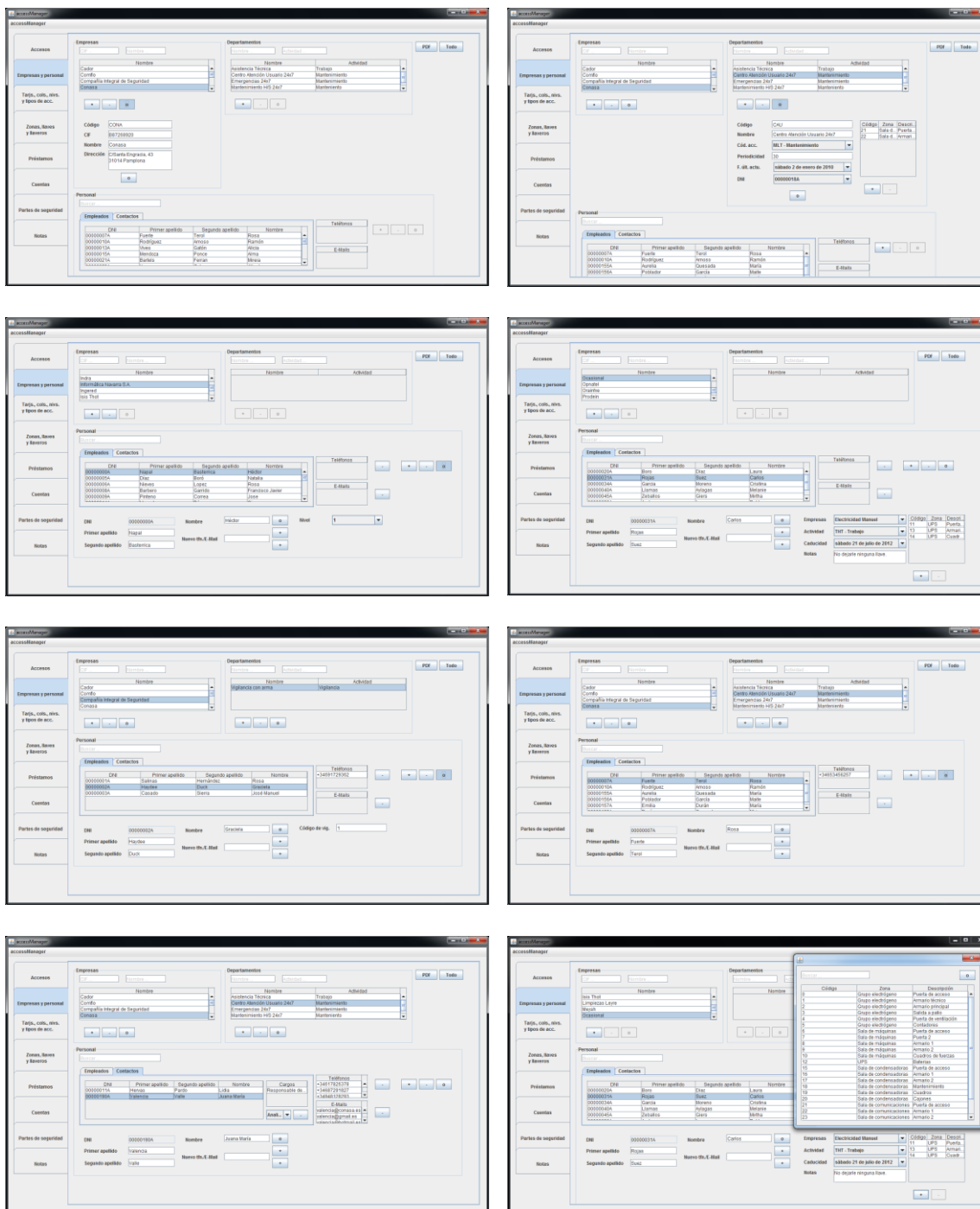
Diagrama de secuencia

5. Codificación e integración

Una vez hemos concluido toda la fase de diseño podemos ponernos manos a la obra con la codificación. La verdad es que no podemos decir que esta iteración esté encuadrada entre las “sencillas” de codificar, pero con el análisis y diseño que hemos realizado, el estudio de la API de *iText* y un buen saco de paciencia no tuvimos serias dificultades en realizar toda la codificación y acabar obteniendo la siguiente interfaz:



Interfaz de la pestaña 'Empresas y personal'



Interfaces parciales de la pestaña 'Empresas y personal'

Con respecto a la integración del trabajo elaborado dentro del sistema no cabe mencionar absolutamente nada ya que toda la codificación ha sido realizada en la API e interfaz gráfica ya disponibles al comienzo de esta iteración. Los únicos componentes nuevos que podríamos mencionar son los diálogos de inserción de nuevas llaves prestables a departamentos y personal ocasional, pero éstos son sencillos JDialogs subordinados a *frame* principal que no representan ningún aspecto por el que debamos preocuparnos.



6. Pruebas

6.1. De unidad

Llega el momento de la verdad: la fase de pruebas. No podemos dar por finalizada la iteración sin antes ejecutar un buen abanico de pruebas que nos permitan tener la certeza de que todos los casos de uso han sido implementados correctamente. A continuación exponemos las que hemos realizado y los resultados que hemos obtenido en ellas:

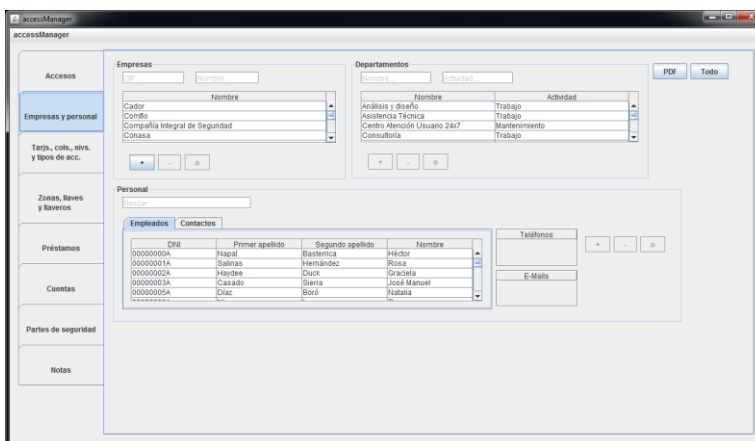
Caso de prueba	4.01 - Casos de uso 13, 19 y 25: Visualizaciones																																														
Objetivo	Comprobar que el usuario visualizará en las tablas de empresas, departamentos, personal y contactos la información que la base de datos alberga.																																														
Actividades	Loguear en el sistema, acceder a la pestaña de empresas y personal y comprobar que los datos iniciales y los obtenidos mediante la pulsación del botón 'Todo' son los esperados.																																														
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. • Utilidad phpMyAdmin. 																																														
Datos de entrada	La población existente en la base de datos, que deberá ser mostrada correctamente en la inicialización de la aplicación y la pulsación del botón 'Todo'.																																														
Salida esperada	<p>Deberíamos visualizar el siguiente contenido en las tablas de empresas, departamentos, empleados y contactos respectivamente:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <p>Nombre Nombre de la empresa</p> <table border="1"> <tr><td>Cador</td></tr> <tr><td>Cornflo</td></tr> <tr><td>Compañía Integral de Seguridad</td></tr> <tr><td>Conasa</td></tr> <tr><td>Conatel</td></tr> <tr><td>Electrolumen</td></tr> <tr><td>Elkor</td></tr> </table> <p style="text-align: center;">...</p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <p>Nombre Nombre del departamento</p> <table border="1"> <tr><td>Análisis y diseño</td></tr> <tr><td>Asistencia Técnica</td></tr> <tr><td>Centro Atención Usuario 24x7</td></tr> <tr><td>Consultoría</td></tr> <tr><td>Consultoría legal y organizativa</td></tr> <tr><td>Consultoría técnica y comunicaciones</td></tr> <tr><td>Dep. Almacenamiento y datos</td></tr> </table> <p style="text-align: center;">...</p> </div> <div style="border: 1px solid #ccc; padding: 5px;"> <table border="1"> <thead> <tr> <th>DNI DNI</th> <th>Primer apellido</th> <th>Segundo apellido</th> <th>Nombre Nombre</th> </tr> </thead> <tbody> <tr><td>00000000A</td><td>Napal</td><td>Basterrica</td><td>Héctor</td></tr> <tr><td>00000001A</td><td>Salinas</td><td>Hernández</td><td>Rosa</td></tr> <tr><td>00000002A</td><td>Haydee</td><td>Duck</td><td>Graciela</td></tr> <tr><td>00000003A</td><td>Casado</td><td>Sierra</td><td>José Manuel</td></tr> <tr><td>00000005A</td><td>Díaz</td><td>Boró</td><td>Natalia</td></tr> <tr><td>00000006A</td><td>Nieves</td><td>Lopez</td><td>Rosa</td></tr> <tr><td>00000007A</td><td>Fuerte</td><td>Terol</td><td>Rosa</td></tr> </tbody> </table> <p style="text-align: center;">...</p> </div>	Cador	Cornflo	Compañía Integral de Seguridad	Conasa	Conatel	Electrolumen	Elkor	Análisis y diseño	Asistencia Técnica	Centro Atención Usuario 24x7	Consultoría	Consultoría legal y organizativa	Consultoría técnica y comunicaciones	Dep. Almacenamiento y datos	DNI DNI	Primer apellido	Segundo apellido	Nombre Nombre	00000000A	Napal	Basterrica	Héctor	00000001A	Salinas	Hernández	Rosa	00000002A	Haydee	Duck	Graciela	00000003A	Casado	Sierra	José Manuel	00000005A	Díaz	Boró	Natalia	00000006A	Nieves	Lopez	Rosa	00000007A	Fuerte	Terol	Rosa
Cador																																															
Cornflo																																															
Compañía Integral de Seguridad																																															
Conasa																																															
Conatel																																															
Electrolumen																																															
Elkor																																															
Análisis y diseño																																															
Asistencia Técnica																																															
Centro Atención Usuario 24x7																																															
Consultoría																																															
Consultoría legal y organizativa																																															
Consultoría técnica y comunicaciones																																															
Dep. Almacenamiento y datos																																															
DNI DNI	Primer apellido	Segundo apellido	Nombre Nombre																																												
00000000A	Napal	Basterrica	Héctor																																												
00000001A	Salinas	Hernández	Rosa																																												
00000002A	Haydee	Duck	Graciela																																												
00000003A	Casado	Sierra	José Manuel																																												
00000005A	Díaz	Boró	Natalia																																												
00000006A	Nieves	Lopez	Rosa																																												
00000007A	Fuerte	Terol	Rosa																																												



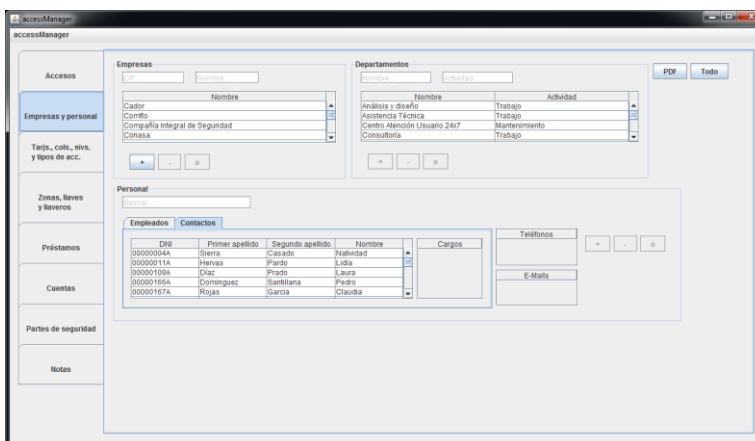
DNI	DNI	Primer apellido	Segundo apellido	Nombre	Nombre
0000004A		Sierra	Casado	Natividad	
0000011A		Hervas	Pardo	Lidia	
0000109A		Diaz	Prado	Laura	
0000166A		Dominguez	Santilana	Pedro	
0000167A		Rojas	Garcia	Claudia	
0000168A		Gonzalez	Cuadrado	Maribel	
0000169A		Van	Elsen	Eric	
0000170A		Hyun	Kim	Yong	
0000171A		Gonzalez	Quiroga	Paloma	
0000172A		Oteo	Latorre	Maria Elena	

...

La esperada. Tenemos:

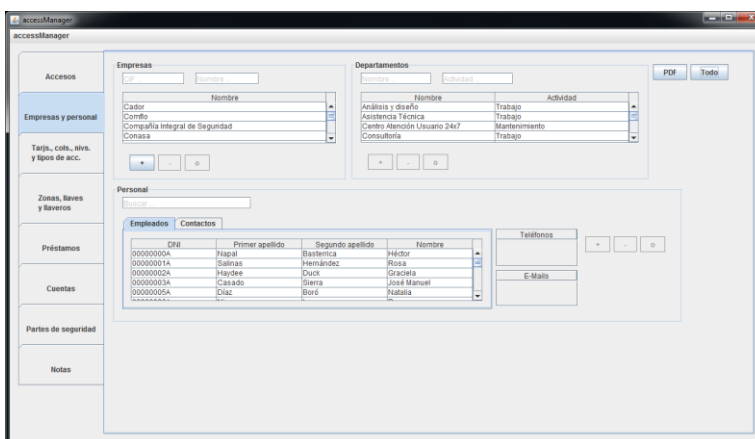


Salida observada

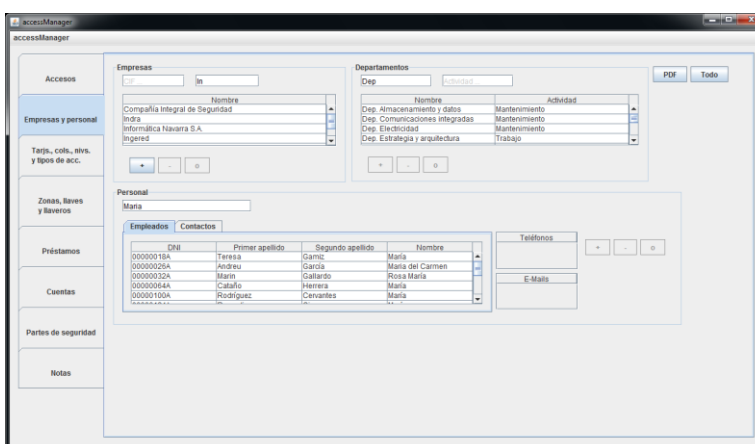




Para finalizar mencionar que el botón 'Todo' realiza su función a la perfección, dejando la pestaña en el estado inicial...



... y que los campos de búsqueda redefinen la muestra de información en las tablas:



**Evaluación/
conclusión**

Los casos de uso '13, 19 y 25 - Visualizaciones' han sido realizados satisfactoriamente.

Caso de prueba para los casos de uso '13, 19 y 25 - Visualizaciones'



Caso de prueba	4.02 - Caso de uso 14: Visualizar empresa																																																																																																									
Objetivo	Chequear que el usuario podrá visualizar toda la información concerniente a una empresa si lo estima oportuno y sin obtener ningún tipo de error.																																																																																																									
Actividades	Desplegar la aplicación, acceder a la tab 'Empresas y personal' y pulsar sobre una empresa, comprobando que toda la información que es desplegada en la pestaña se corresponde con la real (existente en la base de datos).																																																																																																									
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. • Utilidad phpMyAdmin. 																																																																																																									
Datos de entrada	<p>Los grupos de empresas que debemos probar son:</p> <ul style="list-style-type: none"> • Empresa 'Informática Navarra S.A'. • Empresa 'Ocasional'. • Departamentos y personal pertenecientes a cualquier empresa de las restantes (p.e.: 'Conasa'). 																																																																																																									
Salida esperada	<p>Los datos que esperamos obtener son:</p> <ul style="list-style-type: none"> • Empresa 'Informática Navarra S.A': <table border="1" style="margin-left: 40px;"> <thead> <tr> <th colspan="2">Datos</th> <th colspan="3">Ninguno.</th> </tr> <tr> <th colspan="2">Departamentos</th> <th colspan="3">Ninguno.</th> </tr> <tr> <th>Empleados</th> <th>DNI <small>DNI</small></th> <th>Primer apellido</th> <th>Segundo apellido</th> <th>Nombre <small>Nombre</small></th> </tr> </thead> <tbody> <tr><td></td><td>00000000A</td><td>Napal</td><td>Basterrica</td><td>Héctor</td></tr> <tr><td></td><td>00000005A</td><td>Diaz</td><td>Boró</td><td>Natalia</td></tr> <tr><td></td><td>00000006A</td><td>Nieves</td><td>Lopez</td><td>Rosa</td></tr> <tr><td></td><td>00000008A</td><td>Barbero</td><td>Garrido</td><td>Francisco Javier</td></tr> <tr><td></td><td>00000009A</td><td>Pinteno</td><td>Correa</td><td>Jose</td></tr> <tr><td></td><td>00000014A</td><td>Llorente</td><td>Pérez</td><td>Siro</td></tr> <tr><td></td><td>00000018A</td><td>Teresa</td><td>Gamiz</td><td>María</td></tr> <tr><td></td><td>00000019A</td><td>Bescos</td><td>Garcia</td><td>Luis</td></tr> <tr><td></td><td>00000028A</td><td>Trillo</td><td>Carmona</td><td>Laura</td></tr> </tbody> </table> <p style="margin-left: 40px;">Contactos Ninguno.</p> <ul style="list-style-type: none"> • Empresa 'Ocasional': <table border="1" style="margin-left: 40px;"> <thead> <tr> <th colspan="2">Departamentos</th> <th colspan="3">Ninguno.</th> </tr> <tr> <th>Empleados</th> <th>DNI <small>DNI</small></th> <th>Primer apellido</th> <th>Segundo apellido</th> <th>Nombre <small>Nombre</small></th> </tr> </thead> <tbody> <tr><td></td><td>00000020A</td><td>Boro</td><td>Diaz</td><td>Laura</td></tr> <tr><td></td><td>00000031A</td><td>Rojas</td><td>Suez</td><td>Carlos</td></tr> <tr><td></td><td>00000034A</td><td>Garcia</td><td>Moreno</td><td>Cristina</td></tr> <tr><td></td><td>00000040A</td><td>Llamas</td><td>Ayagas</td><td>Melanie</td></tr> <tr><td></td><td>00000045A</td><td>Zeballos</td><td>Giers</td><td>Mirtha</td></tr> <tr><td></td><td>00000056A</td><td>Andres</td><td>Lozano</td><td>Pablo</td></tr> <tr><td></td><td>00000067A</td><td>Teresa</td><td>Cáceres</td><td>Claudia</td></tr> </tbody> </table> <p style="margin-left: 40px;">Contactos Ninguno.</p>	Datos		Ninguno.			Departamentos		Ninguno.			Empleados	DNI <small>DNI</small>	Primer apellido	Segundo apellido	Nombre <small>Nombre</small>		00000000A	Napal	Basterrica	Héctor		00000005A	Diaz	Boró	Natalia		00000006A	Nieves	Lopez	Rosa		00000008A	Barbero	Garrido	Francisco Javier		00000009A	Pinteno	Correa	Jose		00000014A	Llorente	Pérez	Siro		00000018A	Teresa	Gamiz	María		00000019A	Bescos	Garcia	Luis		00000028A	Trillo	Carmona	Laura	Departamentos		Ninguno.			Empleados	DNI <small>DNI</small>	Primer apellido	Segundo apellido	Nombre <small>Nombre</small>		00000020A	Boro	Diaz	Laura		00000031A	Rojas	Suez	Carlos		00000034A	Garcia	Moreno	Cristina		00000040A	Llamas	Ayagas	Melanie		00000045A	Zeballos	Giers	Mirtha		00000056A	Andres	Lozano	Pablo		00000067A	Teresa	Cáceres	Claudia
Datos		Ninguno.																																																																																																								
Departamentos		Ninguno.																																																																																																								
Empleados	DNI <small>DNI</small>	Primer apellido	Segundo apellido	Nombre <small>Nombre</small>																																																																																																						
	00000000A	Napal	Basterrica	Héctor																																																																																																						
	00000005A	Diaz	Boró	Natalia																																																																																																						
	00000006A	Nieves	Lopez	Rosa																																																																																																						
	00000008A	Barbero	Garrido	Francisco Javier																																																																																																						
	00000009A	Pinteno	Correa	Jose																																																																																																						
	00000014A	Llorente	Pérez	Siro																																																																																																						
	00000018A	Teresa	Gamiz	María																																																																																																						
	00000019A	Bescos	Garcia	Luis																																																																																																						
	00000028A	Trillo	Carmona	Laura																																																																																																						
Departamentos		Ninguno.																																																																																																								
Empleados	DNI <small>DNI</small>	Primer apellido	Segundo apellido	Nombre <small>Nombre</small>																																																																																																						
	00000020A	Boro	Diaz	Laura																																																																																																						
	00000031A	Rojas	Suez	Carlos																																																																																																						
	00000034A	Garcia	Moreno	Cristina																																																																																																						
	00000040A	Llamas	Ayagas	Melanie																																																																																																						
	00000045A	Zeballos	Giers	Mirtha																																																																																																						
	00000056A	Andres	Lozano	Pablo																																																																																																						
	00000067A	Teresa	Cáceres	Claudia																																																																																																						



- Departamentos y personal pertenecientes a una empresa (p.e.: 'Conasa'):

Datos	
Cod_Emp Código de la empresa	CIF Código de identificación fiscal de la empresa
CONA	B87268920
Nombre Nombre de la empresa	Direccion Dirección de las oficinas de la empresa
Conasa	C/Santa Engracia, 43 31014 Pamplona

Departamentos	
Nombre Nombre del departamento	
Asistencia Técnica	
Centro Atención Usuario 24x7	
Emergencias 24x7	
Mantenimiento H/S 24x7	
Sistemas	

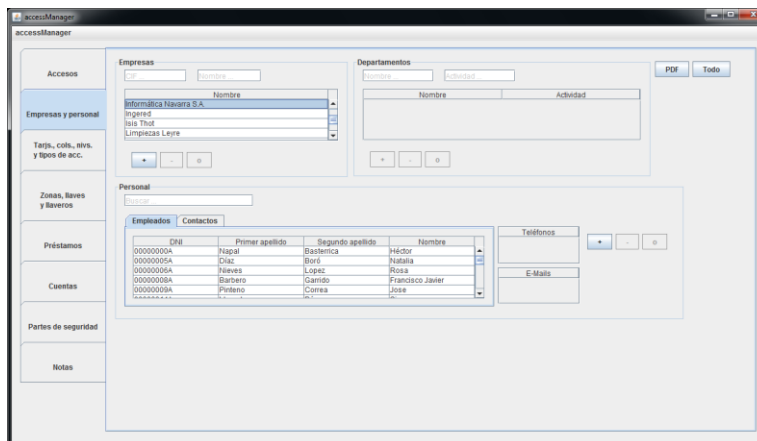
Empleados				
DNI	Primer apellido	Segundo apellido	Nombre	
00000007A	Fuerte	Terol	Rosa	
00000010A	Rodriguez	Amoso	Ramón	
00000013A	Vives	Gatón	Alicia	
00000015A	Mendoza	Ponce	Alma	
00000021A	Bartels	Ferran	Mireia	
00000025A	Vergara	Cela	Alberto	
...				

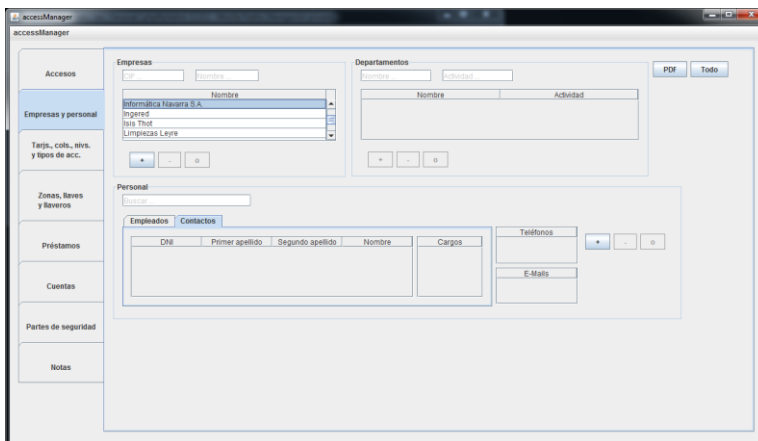
Contactos				
DNI	Primer apellido	Segundo apellido	Nombre	
00000004A	Sierra	Casado	Natividad	
00000011A	Hervas	Pardo	Lidia	
00000190A	Valencia	Vale	Juana María	
00000191A	Vera	Gil	Raquel	

La esperada:

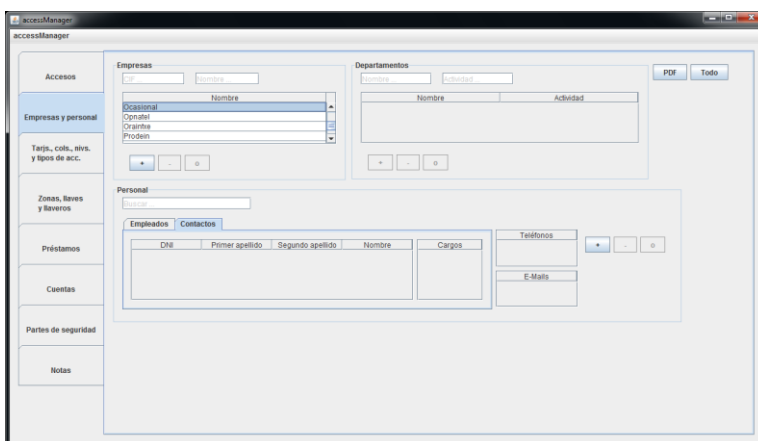
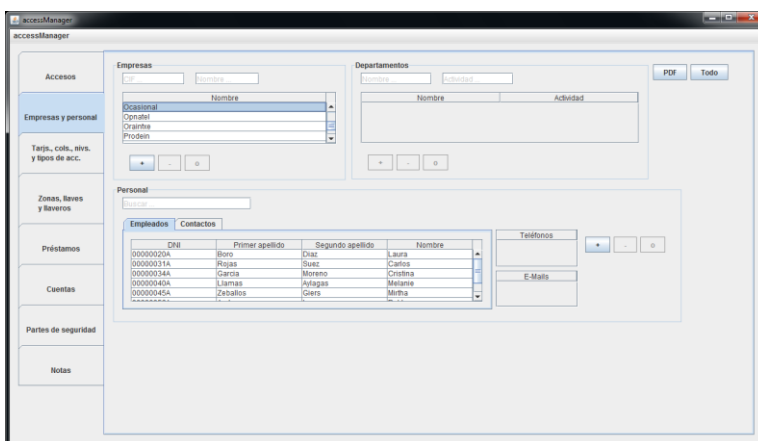
- Para la empresa 'Informática Navarra S.A.':

Salida observada





- En el caso de la empresa que agrupa a todo el personal ocasional:





- Por último, todos los datos que el sistema muestra de la empresa 'Conasa':

The screenshot shows the 'accessManager' application window. The 'Empresas' section is active, displaying the following data for 'Conasa':

Código	Nombre
CONA	Conasa

The 'Departamentos' section shows a list of departments:

Nombre	Actividad
Asistencia Técnica	Trabajo
Centro Atención Usuario 24x7	Mantenimiento
Emergencias 24x7	Mantenimiento
Mantenimiento HS 24x7	Mantenimiento

Other fields include: Cif: 887268920, Dirección: C/Santa Engracia, 43 31014 Pamplona.

The screenshot shows the 'accessManager' application window with the 'Personal' section active. The 'Empleados' tab is selected, displaying a list of employees:

DNI	Primer apellido	Segundo apellido	Nombre
00000007A	Puente	Tercero	Rosa
00000019A	Rodriguez	Arenas	Ramón
00000013A	Irujo	Gallón	Alicia
00000015A	Blondoa	Ponce	Alma
00000021A	Bartals	Ferran	Maria

The 'Contactos' tab is also visible, showing a list of contacts with columns for DNI, Primer apellido, Segundo apellido, Nombre, and Cargos.

Evaluación/
conclusión

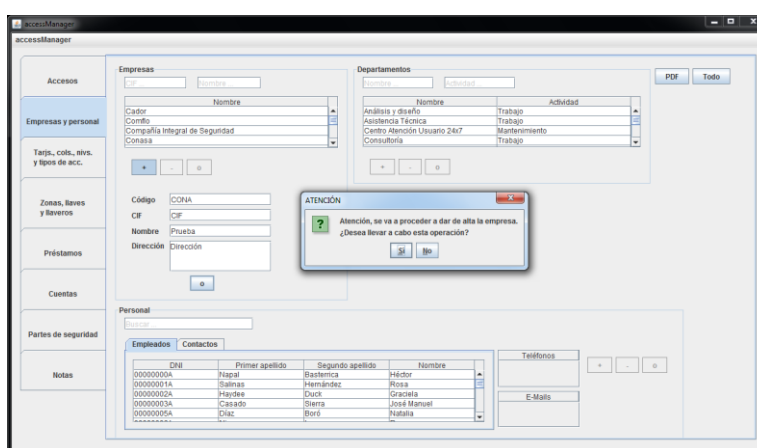
El caso de uso '14 - Visualizar empresa' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '14 - Visualizar empresa'

Caso de prueba	4.03 - Caso de uso 15: Añadir empresa
Objetivo	Asegurar que el proceso de inserción de empresas se realiza de forma correcta.
Actividades	Situarnos en la pestaña de empresas y personal y probar a introducir varias empresas con datos válidos y erróneos, comprobando que el sistema siempre responde cómo esperamos.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. • Utilidad phpMyAdmin.

Generamos el siguiente conjunto de empresas que probaremos a insertar:

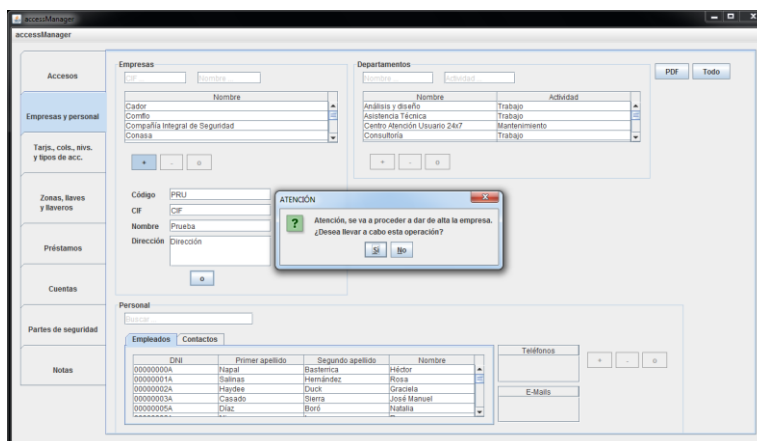
- Erróneas:
‘CONA, CIF, Prueba, Dirección’. (Código de empresa existente)



‘CODIGODEMASIADOLARGO, CIF, PRUEBA, Dirección’. (Formato de datos erróneo)

Datos de entrada

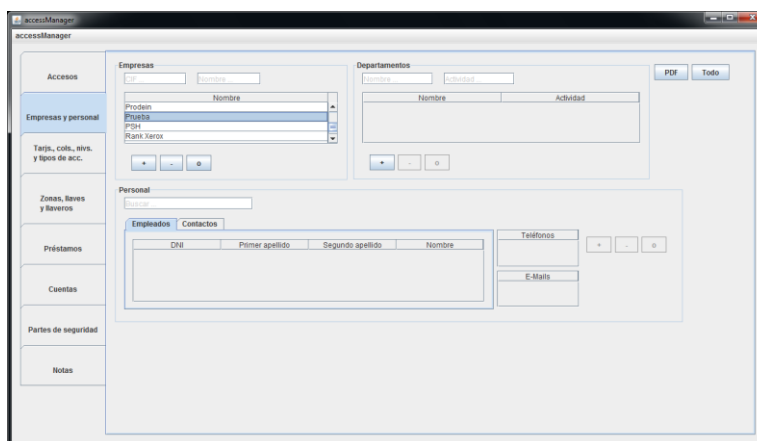
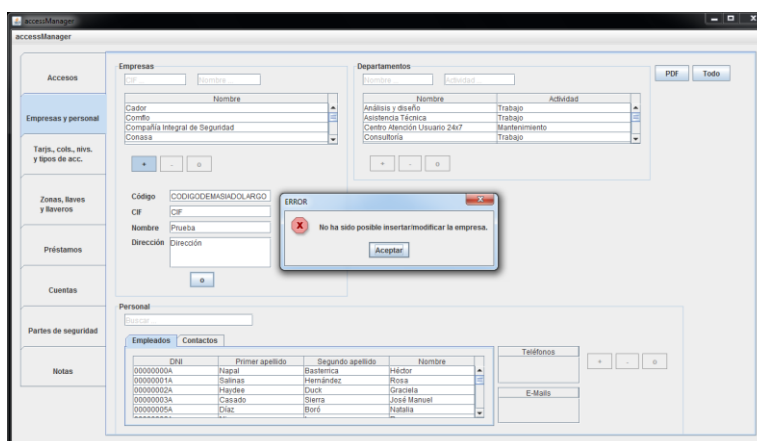
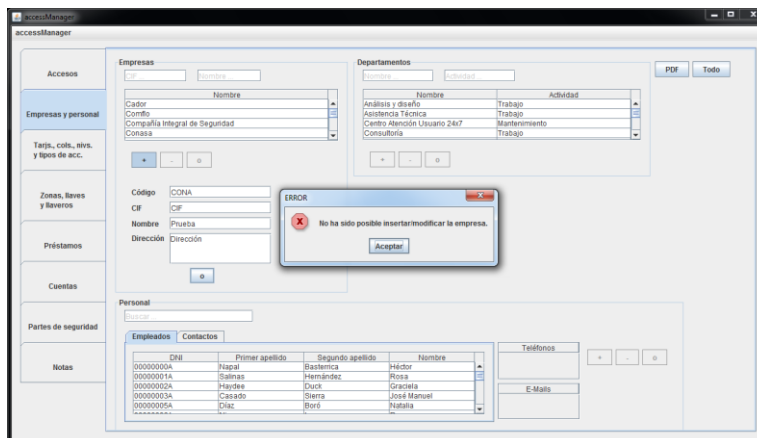
- Válida
‘PRU, CIF, Prueba, Direccion’



Salida esperada Los dos primeros intentos deberían ser en balde, obteniendo del sistema únicamente mensajes de error, mientras que el tercero sí que debería producir la inserción de la empresa en el sistema.

Salida observada

La esperada:



Evaluación/ conclusión El caso de uso '15 - Añadir empresa' ha sido realizado satisfactoriamente.

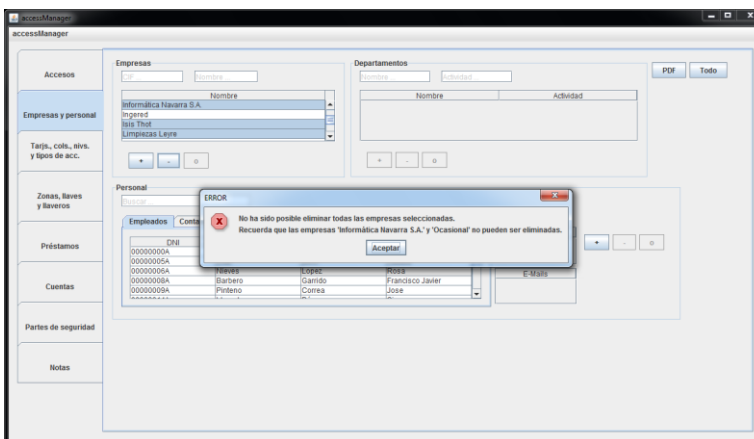
Caso de prueba para el caso de uso '15 - Añadir empresa'

Caso de prueba	4.04 - Caso de uso 16: Borrar empresa
Objetivo	Verificar que el usuario puede eliminar el conjunto de empresas que estime oportuno sin obtener como respuesta algún tipo de situación indeseada.
Actividades	Arrancar el sistema, acceder a la pestaña de empresas y personal y una vez allí probar a eliminar cualquier conjunto de empresas y observar los resultados que se producen.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior. • Utilidad phpMyAdmin.
Datos de entrada	<p>No es posible eliminar cero empresas ya que el botón de borrado se encontrará desactivado en caso de no tener ninguna empresa seleccionada. Nos quedaría pues probar a borrar de 1 a N empresas, lo que nos deja dos posibles pruebas:</p> <ul style="list-style-type: none"> • Conjunto de empresas en las que está incluida 'Informática Navarra S.A' u 'Ocasional':
	<div data-bbox="563 853 1331 1294" data-label="Image"> </div> <ul style="list-style-type: none"> • Conjunto de empresas en las que no está incluida 'Informática Navarra S.A' u 'Ocasional': <div data-bbox="563 1435 1331 1877" data-label="Image"> </div>

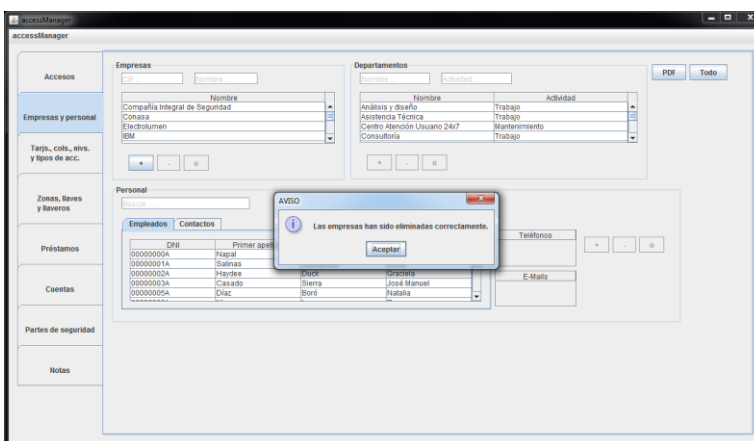
Salida esperada

No deberá ser posible realizar el primer borrado pero sí el segundo.

La espera:



Salida observada



Evaluación/
conclusión

El caso de uso '16 - Borrar empresa' ha sido realizado satisfactoriamente.

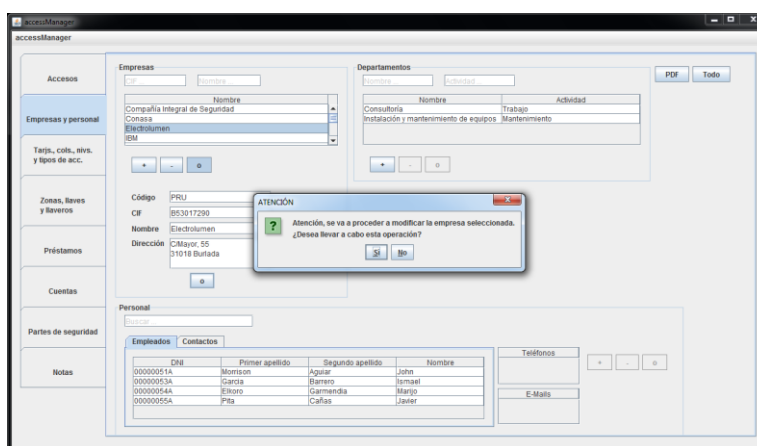
Caso de prueba para el caso de uso '16 - Borrar empresa'

Caso de prueba	4.05 - Caso de uso 17: Modificar empresa
Objetivo	Comprobar que se ha implementado satisfactoriamente el proceso que se encarga de modificar una empresa con los datos proporcionados por el usuario.
Actividades	Loguear en el sistema, situarnos en 'Empresas y personal' y probar a modificar varias empresas de forma correcta e incorrecta verificando que el sistema actúa correctamente en todo momento.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior. • Utilidad phpMyAdmin.

Las empresas a probar son 'Informática Navarra S.A.', 'Ocasional' y cualquier otra de las restantes. Para este ejemplo tomaremos la de nombre 'Electrolumen' y probaremos a modificarla de manera:

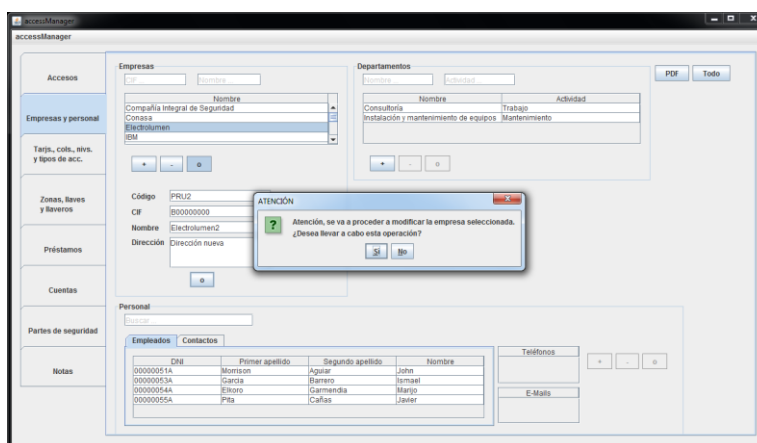
- Incorrecta:
 - 'PRU, B53017290, Electrolumen, C/Mayor, 55\n31018 Burlada'. (Código existente).

Datos de entrada



'CODIGODEMASIADOLARGO, B53017290, Electrolumen, C/Mayor, 55\n31018 Burlada'. (Código existente).

- Correcta:
 - 'PRU2, B00000000, Electrolumen2, Dirección nueva'.

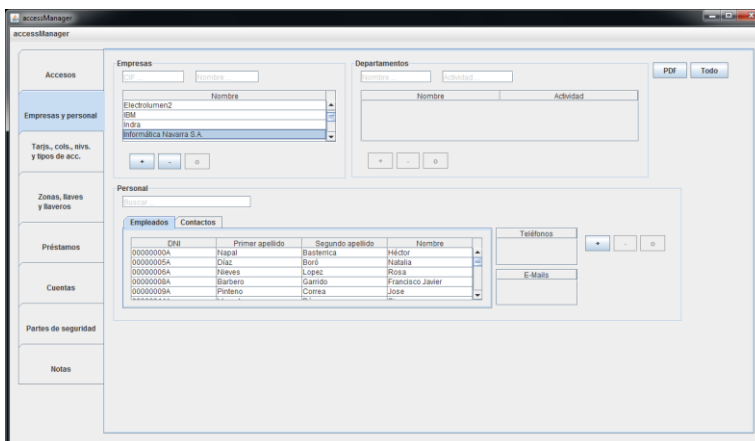




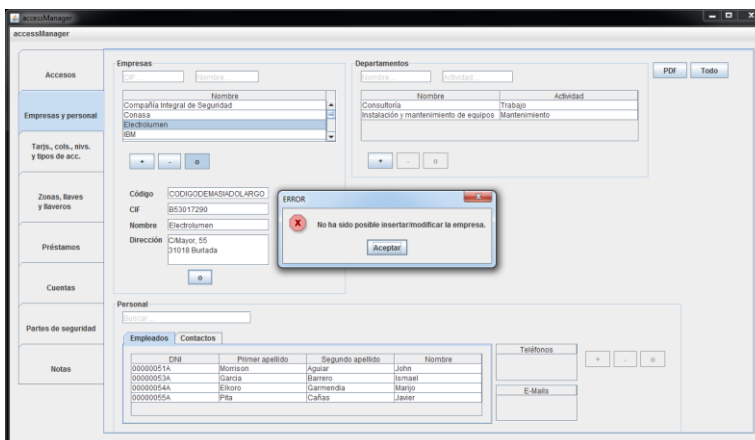
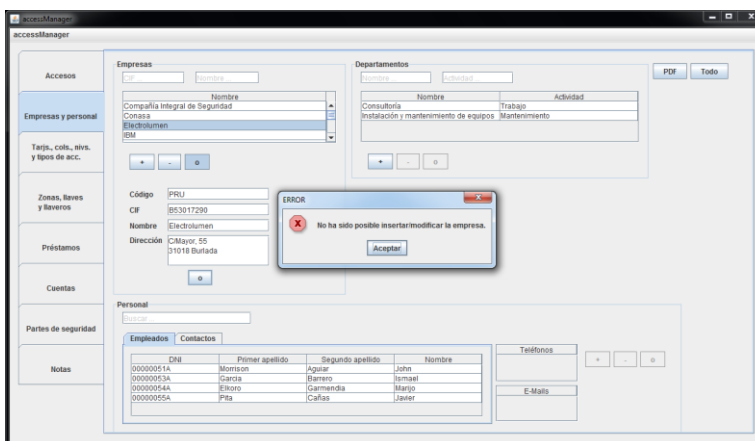
Si nos fijamos son los mismos casos de prueba que lo empleados para la inserción, solo que aplicados a la actualización.

Salida esperada Las dos primeras empresas (botón de modificación desactivado) no deberían ser modificables pero sí la tercera en su segundo intento.

La esperada. En orden obtenemos:



Salida observada





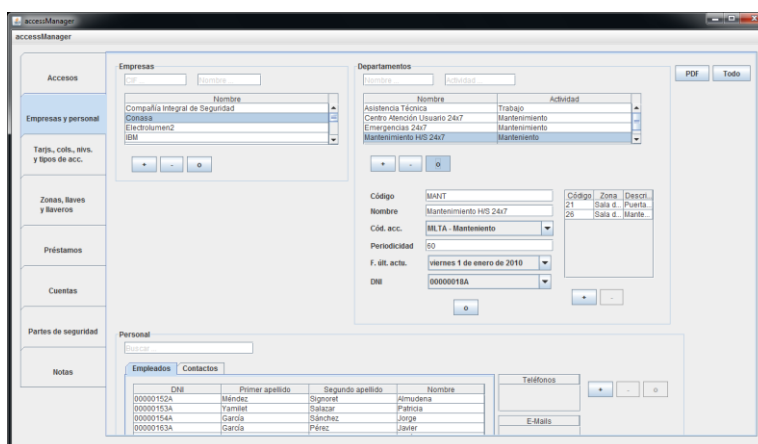
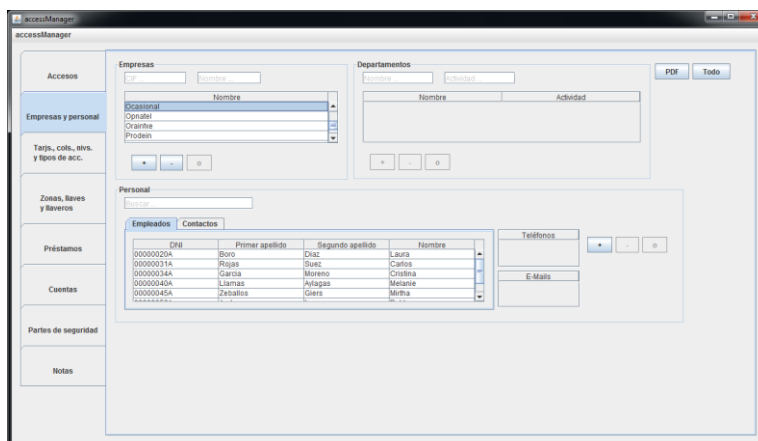
**Evaluación/
conclusión**

El caso de uso '17 - Modificar empresa' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '17 - Modificar empresa'



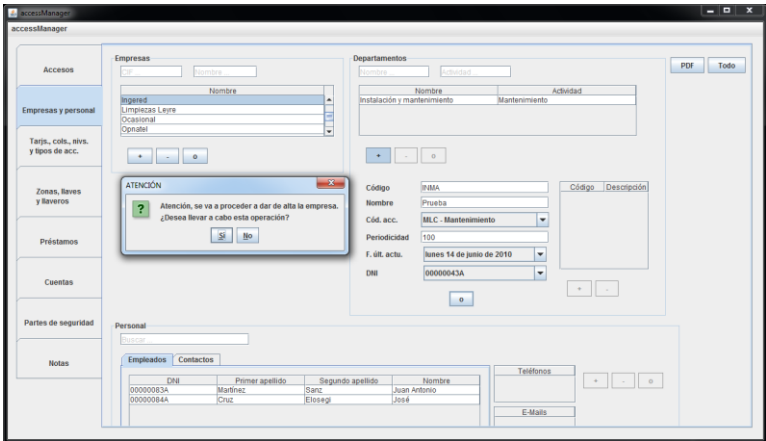
Caso de prueba	4.06 - Caso de uso 20: Visualizar departamento																																																																																																			
Objetivo	Verificar que podemos visualizar todos los datos que conforman cualquier departamento.																																																																																																			
Actividades	Acceder a la tab 'Empresas y personal' de accessManager y realizar el visionado de varios departamentos comprobando que es correcto en todo momento.																																																																																																			
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior. • Utilidad phpMyAdmin. 																																																																																																			
Datos de entrada	Vamos a seleccionar las empresas 'Informática Navarra S.A.' y 'Ocasional' y el departamento 'Mantenimiento H/S 24x7' de 'Conasa'.																																																																																																			
Salida esperada	<p>Las dos primeras empresas no deben disponer de ningún tipo de departamento, mientras que el departamento 'Mantenimiento H/S 24x7' deberá ofrecernos los siguientes datos:</p> <table border="1"> <thead> <tr> <th colspan="4">Datos</th> </tr> <tr> <th>Cod_Emp</th> <th>Cod_Dep</th> <th>Nombre</th> <th>Cod_Acc</th> </tr> <tr> <td><small>Código de empresa</small></td> <td><small>Código de departamento</small></td> <td><small>Nombre del departamento</small></td> <td><small>Código de acceso que posee</small></td> </tr> </thead> <tbody> <tr> <td>CONA</td> <td>MANT</td> <td>Mantenimiento H/S 24x7</td> <td>MLTA</td> </tr> <tr> <th>Periodicidad</th> <th>F_Ult_Actu</th> <th>DNI</th> <td></td> </tr> <tr> <td><small>Periodicidad con la que renueva la lista de personal que trabaja en el centro</small></td> <td><small>Fecha y hora de la última actualización de personal realizada</small></td> <td><small>DNI del autorizador responsable del departamento</small></td> <td></td> </tr> <tr> <td>60</td> <td>2010-01-01 00:00:00</td> <td>0000018A</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="5">Empleados</th> </tr> <tr> <th>DNI</th> <th>Primer apellido</th> <th>Segundo apellido</th> <th>Nombre</th> <td></td> </tr> <tr> <td><small>DNI</small></td> <td><small>Primer apellido</small></td> <td><small>Segundo apellido</small></td> <td><small>Nombre</small></td> <td></td> </tr> </thead> <tbody> <tr> <td>00000152A</td> <td>Méndez</td> <td>Signoret</td> <td>Almudena</td> <td></td> </tr> <tr> <td>00000153A</td> <td>Yamilet</td> <td>Salazar</td> <td>Patricia</td> <td></td> </tr> <tr> <td>00000154A</td> <td>García</td> <td>Sánchez</td> <td>Jorge</td> <td></td> </tr> <tr> <td>00000163A</td> <td>García</td> <td>Pérez</td> <td>Javier</td> <td></td> </tr> <tr> <td>00000165A</td> <td>Beguiristain</td> <td>Amatriain</td> <td>Verónica</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="4">Contactos</th> </tr> <tr> <th>DNI</th> <th>Primer apellido</th> <th>Segundo apellido</th> <th>Nombre</th> </tr> <tr> <td><small>DNI</small></td> <td><small>Primer apellido</small></td> <td><small>Segundo apellido</small></td> <td><small>Nombre</small></td> </tr> </thead> <tbody> <tr> <td>00000004A</td> <td>Sierra</td> <td>Casado</td> <td>Natividad</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="3">Llaves</th> </tr> <tr> <th>Código</th> <th>Zona</th> <th>Descripcion</th> </tr> <tr> <td><small>Código</small></td> <td><small>Zona</small></td> <td><small>Descripcion</small></td> </tr> </thead> <tbody> <tr> <td>21</td> <td>Sala de comunicaciones</td> <td>Puerta de acceso</td> </tr> <tr> <td>26</td> <td>Sala de comunicaciones</td> <td>Mantenimiento</td> </tr> </tbody> </table>	Datos				Cod_Emp	Cod_Dep	Nombre	Cod_Acc	<small>Código de empresa</small>	<small>Código de departamento</small>	<small>Nombre del departamento</small>	<small>Código de acceso que posee</small>	CONA	MANT	Mantenimiento H/S 24x7	MLTA	Periodicidad	F_Ult_Actu	DNI		<small>Periodicidad con la que renueva la lista de personal que trabaja en el centro</small>	<small>Fecha y hora de la última actualización de personal realizada</small>	<small>DNI del autorizador responsable del departamento</small>		60	2010-01-01 00:00:00	0000018A		Empleados					DNI	Primer apellido	Segundo apellido	Nombre		<small>DNI</small>	<small>Primer apellido</small>	<small>Segundo apellido</small>	<small>Nombre</small>		00000152A	Méndez	Signoret	Almudena		00000153A	Yamilet	Salazar	Patricia		00000154A	García	Sánchez	Jorge		00000163A	García	Pérez	Javier		00000165A	Beguiristain	Amatriain	Verónica		Contactos				DNI	Primer apellido	Segundo apellido	Nombre	<small>DNI</small>	<small>Primer apellido</small>	<small>Segundo apellido</small>	<small>Nombre</small>	00000004A	Sierra	Casado	Natividad	Llaves			Código	Zona	Descripcion	<small>Código</small>	<small>Zona</small>	<small>Descripcion</small>	21	Sala de comunicaciones	Puerta de acceso	26	Sala de comunicaciones	Mantenimiento
Datos																																																																																																				
Cod_Emp	Cod_Dep	Nombre	Cod_Acc																																																																																																	
<small>Código de empresa</small>	<small>Código de departamento</small>	<small>Nombre del departamento</small>	<small>Código de acceso que posee</small>																																																																																																	
CONA	MANT	Mantenimiento H/S 24x7	MLTA																																																																																																	
Periodicidad	F_Ult_Actu	DNI																																																																																																		
<small>Periodicidad con la que renueva la lista de personal que trabaja en el centro</small>	<small>Fecha y hora de la última actualización de personal realizada</small>	<small>DNI del autorizador responsable del departamento</small>																																																																																																		
60	2010-01-01 00:00:00	0000018A																																																																																																		
Empleados																																																																																																				
DNI	Primer apellido	Segundo apellido	Nombre																																																																																																	
<small>DNI</small>	<small>Primer apellido</small>	<small>Segundo apellido</small>	<small>Nombre</small>																																																																																																	
00000152A	Méndez	Signoret	Almudena																																																																																																	
00000153A	Yamilet	Salazar	Patricia																																																																																																	
00000154A	García	Sánchez	Jorge																																																																																																	
00000163A	García	Pérez	Javier																																																																																																	
00000165A	Beguiristain	Amatriain	Verónica																																																																																																	
Contactos																																																																																																				
DNI	Primer apellido	Segundo apellido	Nombre																																																																																																	
<small>DNI</small>	<small>Primer apellido</small>	<small>Segundo apellido</small>	<small>Nombre</small>																																																																																																	
00000004A	Sierra	Casado	Natividad																																																																																																	
Llaves																																																																																																				
Código	Zona	Descripcion																																																																																																		
<small>Código</small>	<small>Zona</small>	<small>Descripcion</small>																																																																																																		
21	Sala de comunicaciones	Puerta de acceso																																																																																																		
26	Sala de comunicaciones	Mantenimiento																																																																																																		
Salida observada	<p>La esperada:</p>																																																																																																			



Evaluación/
conclusión

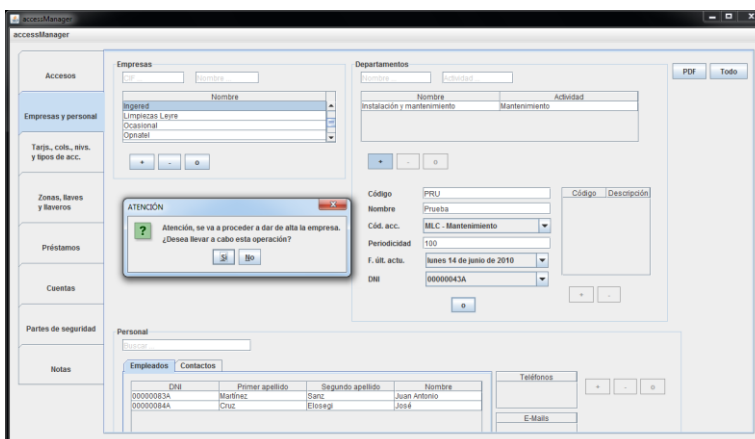
El caso de uso '20 - Visualizar departamento' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '20 - Visualizar departamento'

Caso de prueba	4.07 - Caso de uso 21: Añadir departamento
Objetivo	Chequear que podemos añadir nuevos departamentos a cualquier empresa (salvo 'Informática Navarra S.A.' y 'Ocasional') siempre y cuando proporcionemos datos con un formato válido.
Actividades	Desplegar la aplicación, situarnos en la segunda pestaña y probar a insertar departamentos válidos y erróneos en varias empresas, comprobando que el sistema siempre responde según lo previsto.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior.
Datos de entrada	<p>Como bien hemos venido comentando, 'Informática Navarra S.A.' y 'Ocasional' son empresas llamémoslas "especiales", que albergan al personal residente y ocasional y no pueden contener departamentos. Éstas serán las primeras empresas a probar.</p> <p>Para el resto de empresas debería ser posible insertar nuevos departamentos, con que eligiéremos una cualquiera (en este caso 'Ingered') y probaremos a añadirle departamentos erróneos...</p> <ul style="list-style-type: none"> • 'INMA, Prueba, MLC, 100, 14 de junio de 2010, 00000043A'. (Código de departamento existente)  <ul style="list-style-type: none"> • 'CODIGODEMASIADOLARGO, Prueba, MLC, 100, 14 de junio de 2010, 00000043a'. (Código con formato no válido)

... y válidos:

- 'PRU, Nombre de prueba, MLC, 100, 14 de junio de 2010, 0000043A'.

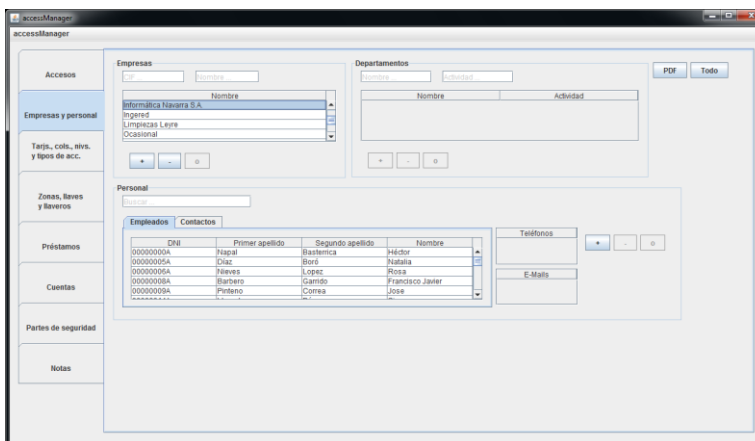


Salida esperada

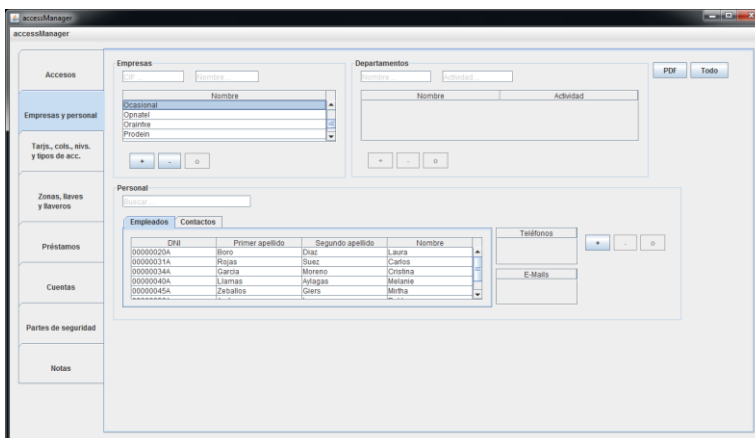
Debe ser imposible agregar nuevos departamentos a 'Informática Navarra S.A.' y 'Ocasional', pero no así a 'Ingered' en caso de proporcionar datos válidos (tercera prueba sobre esta empresa).

La esperada:

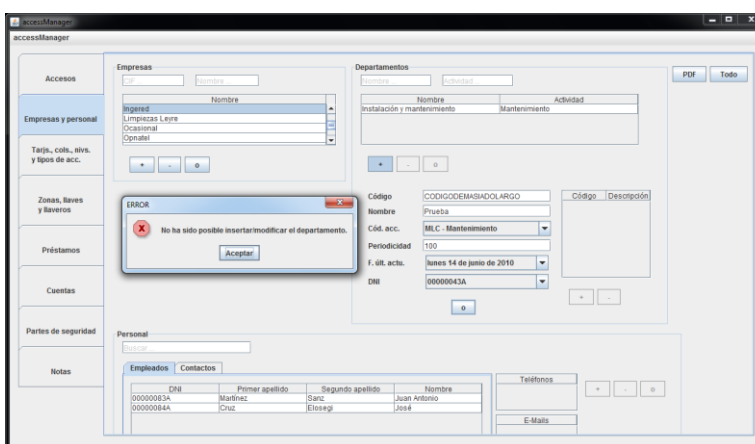
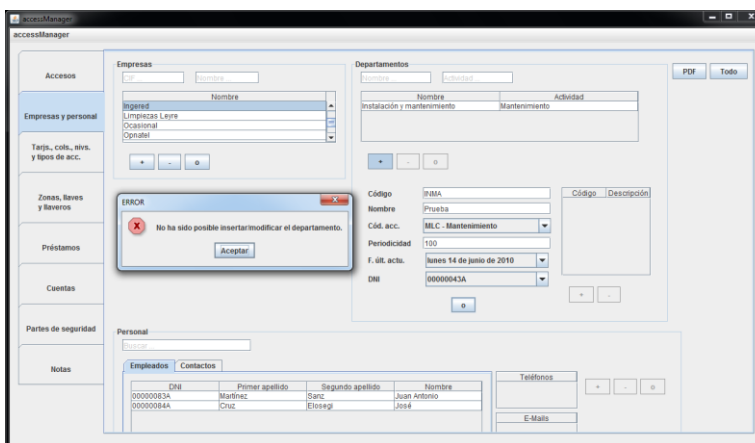
- Para las empresas 'Informática Navarra S.A.' y 'Ocasional' no disponemos del botón de adición de departamentos activado:



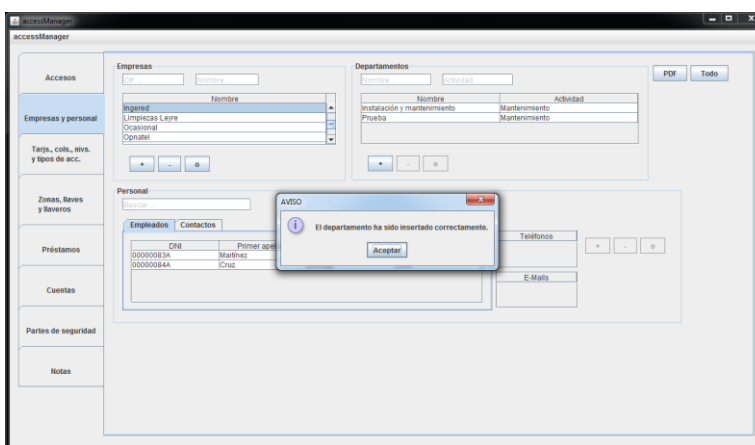
Salida observada



- En los dos primeros intentos con 'Ingered':



- La última prueba se realiza con éxito:



**Evaluación/
conclusión**

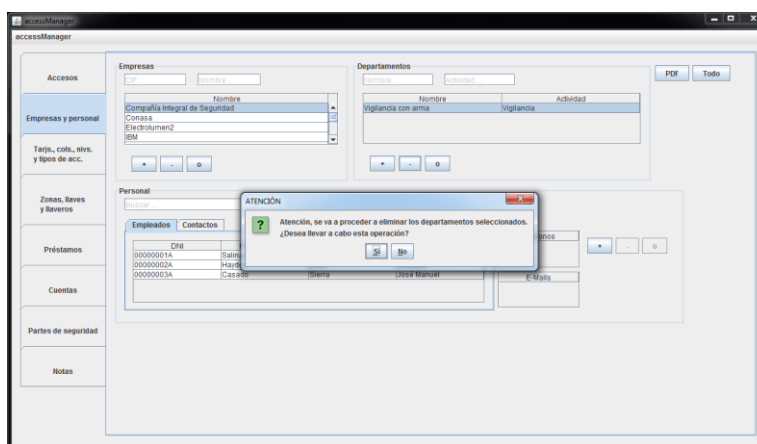
El caso de uso '21 - Añadir departamento' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '21 - Añadir departamento'

Caso de prueba	4.08 - Caso de uso 22: Borrar departamento
Objetivo	Chequear que el usuario puede realizar el borrado de cualquier conjunto de departamentos sin preocuparse de que el sistema falle.
Actividades	Iniciar accessManager, ir a la pestaña de empresas y personal, seleccionar varias empresas y probar a realizar borrados sobre ellas comprobando que el caso de uso se ejecuta con éxito.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior.

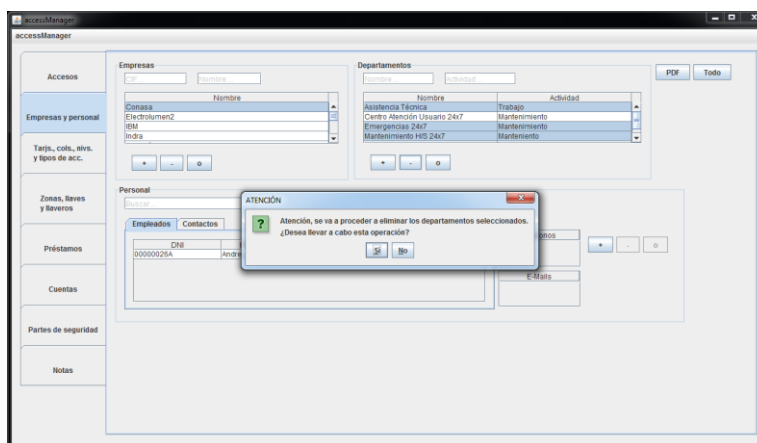
Hemos visto que 'Informática Navarra S.A' y 'Ocasional' no disponen de departamentos, lo que nos lleva a tener que probar solo el resto de empresas. Tenemos la empresa encargada de la vigilancia del centro y el conjunto de empresas que restan, con que probaremos a:

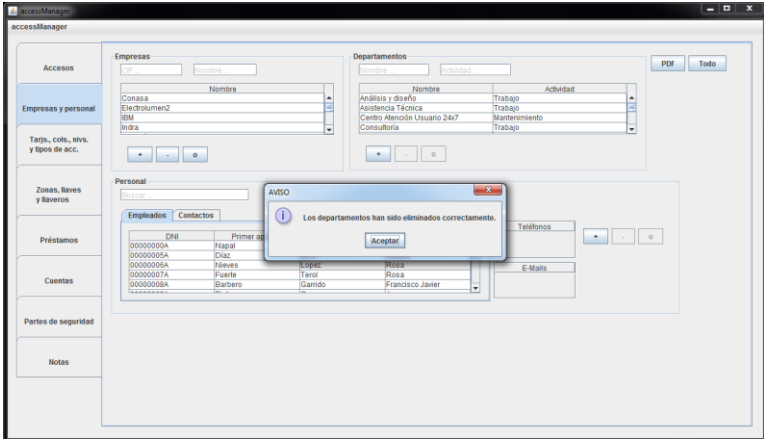
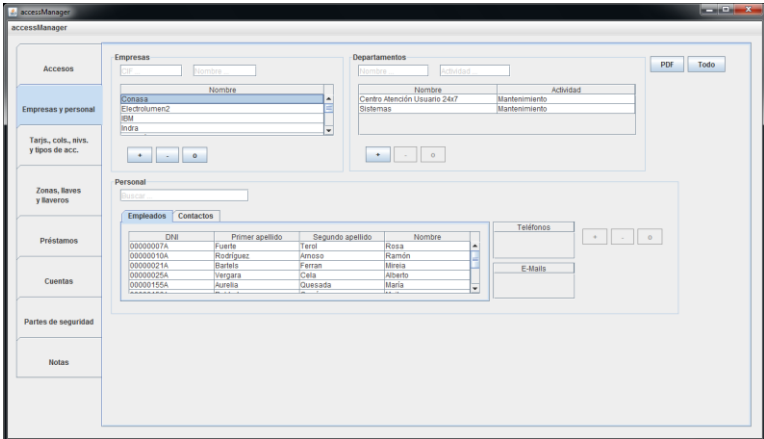
- Borrar el departamento 'Vigilancia con arma' de 'Compañía Integral de Seguridad', dejando al centro sin departamento de vigilancia:



Datos de entrada

- Borrado múltiple de departamentos en la empresa 'Conasa' (por ejemplo):



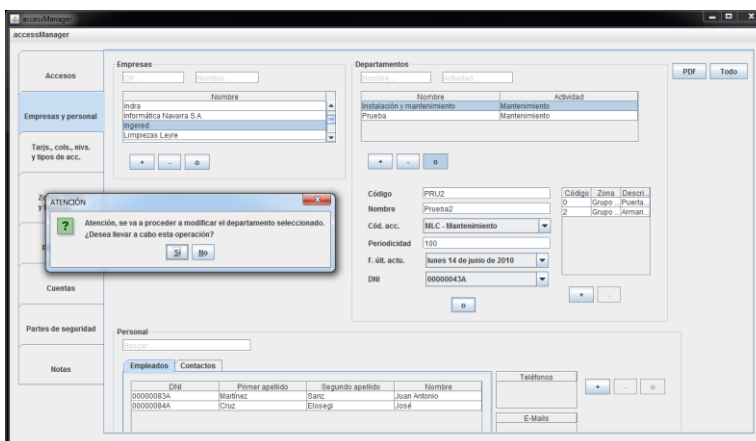
<p>Salida esperada</p>	<p>El primer borrado causará el borrado de la empresa ‘Compañía Integral de Seguridad’ provocado por el disparador pertinente diseñado en la iteración C1 (borrado de empresas que no disponen de departamentos en el centro) mientras que el segundo borrado sólo causará borrados en cascada sobre ‘Conasa’.</p>
<p>Salida observada</p>	<p>La esperada, para la primera prueba...</p>  <p>... y para la segunda:</p> 
<p>Evaluación/ conclusión</p>	<p>El caso de uso ‘22 - Borrar departamento’ ha sido realizado satisfactoriamente.</p>

Caso de prueba para el caso de uso ‘22 - Borrar departamento’

Caso de prueba	4.09 - Caso de uso 23: Modificar departamento
Objetivo	Comprobar que el proceso de modificación de departamentos ha sido desarrollado correctamente, ofreciendo al usuario la posibilidad de editar todos los datos de cualquier departamento para establecerle el estado deseado del mismo.
Actividades	Acceder a la pestaña de empresas y personal, seleccionar un departamento cualquiera y probar a modificarlo de varias formas (correctas e incorrectas) verificando en todo momento que el sistema actúa como lo teníamos previsto.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior.
Datos de entrada	<p>La verdad es que podría servirnos cualquier departamento para realizar este caso de prueba. Optaremos por 'Ingered - Instalación y mantenimiento', que también posee llaves y así podremos comprobar la inicialización del diálogo que alberga a las restantes.</p> <p>En primer lugar nos centramos en los datos del departamento, que podemos modificarlos de forma incorrecta...</p> <ul style="list-style-type: none"> • 'PRU, Prueba, MLC, 100, 14 de junio de 2010, 00000043A'. (Código de departamento existente) <div data-bbox="563 994 1331 1435" data-label="Image"> </div> <ul style="list-style-type: none"> • 'CODIGODEMASIADOLARGO, Prueba, MLC, 100, 14 de junio de 2010, 00000043A'. (Código con formato no válido)

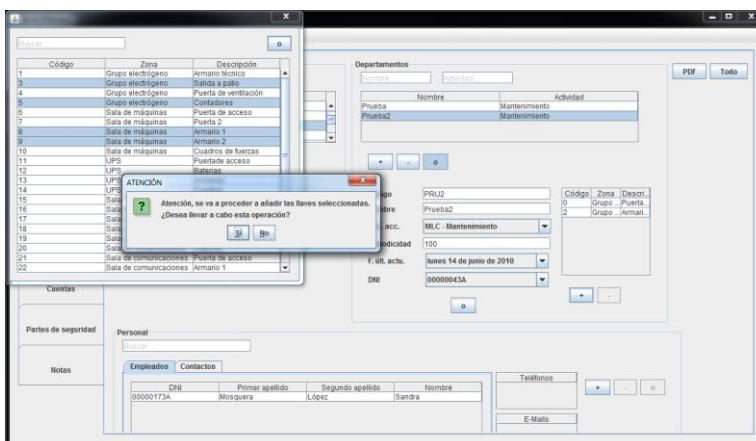
... o correcta:

- 'PRU2, Prueba2, MLC, 100, 14 de junio de 2010, 0000043A'.

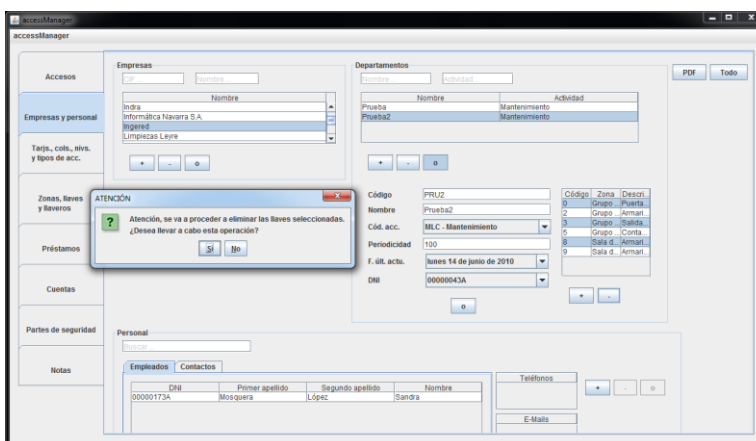


En segundo lugar nos centramos en la asignación de llaves al departamento, donde tenemos que probar que:

- El diálogo de llaves se inicia correctamente.
- Se pueden añadir las llaves deseadas al departamento:



- Se pueden eliminar las llaves prestables que estimemos oportuno:



Salida esperada

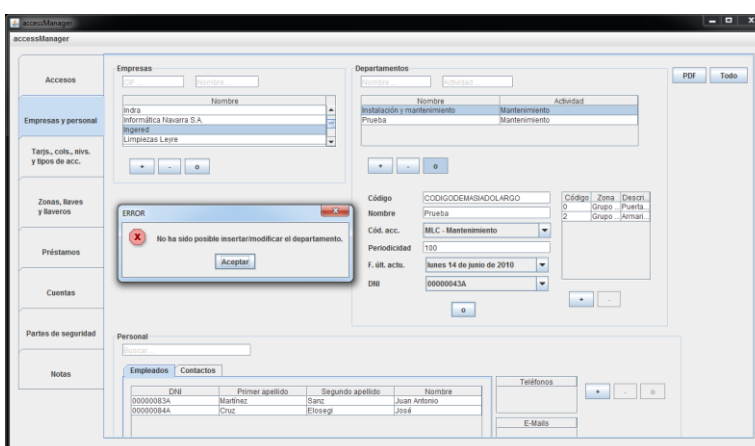
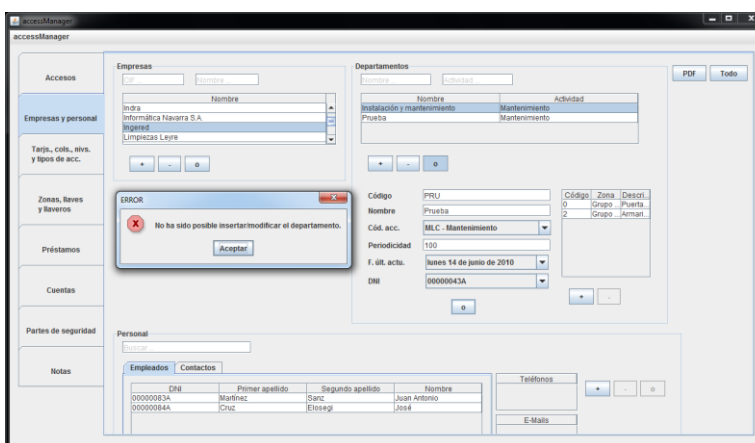
Los dos primeros intentos de modificación de los datos del departamento deberían obtener como única respuesta mensajes de error avisando al usuario de que ha sido imposible realizar las modificaciones que propone. En cambio, en el tercero, donde el usuario sí que proporciona datos con un formato válido, no se producirá ningún error y la actualización tendrá lugar en la base de datos.

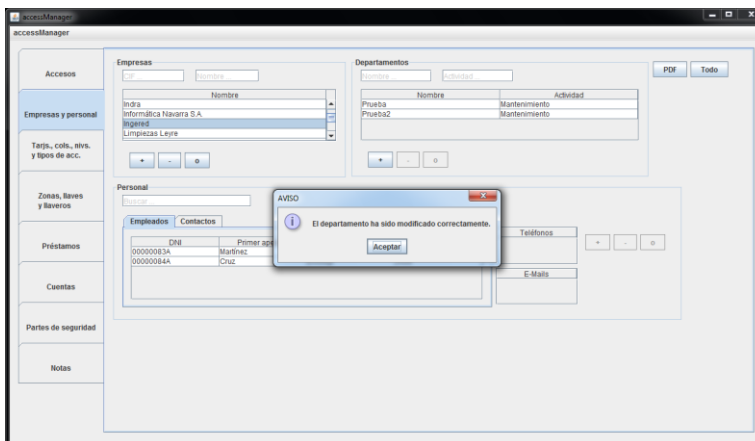
Para la segunda fase de pruebas deberíamos obtener con cada paso:

- Un diálogo de llaves que no contengan las llaves 0 y 2, ya que éstas ya figuran como prestables al departamento.
- La tabla de llaves deberá mostrar las llaves existentes (0 y 2) más las introducidas como nuevas (3, 5, 8 y 9).
- La tabla de llaves compuesta por las llaves 2, 5 y 9, que son las únicas que no se seleccionarán para el borrado.

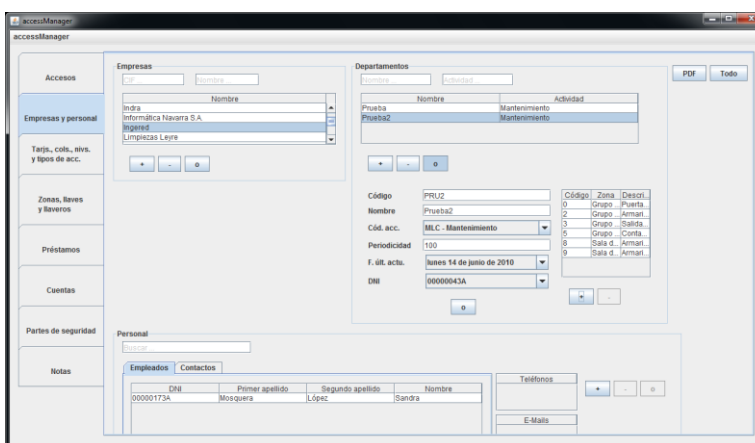
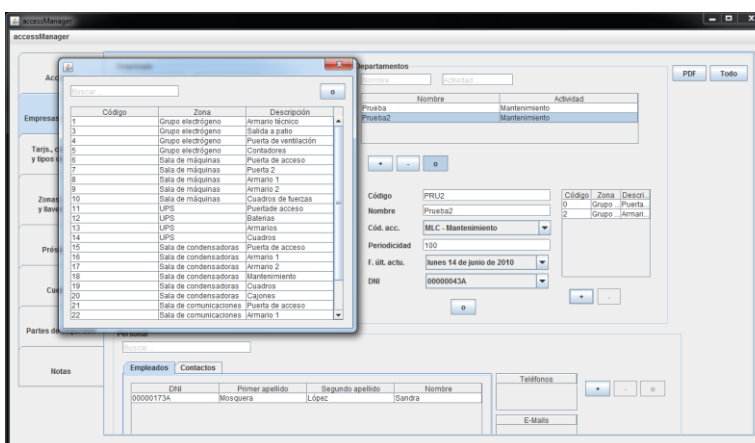
La esperada. Tendríamos para los intentos de modificación de datos los siguientes resultados respectivamente...

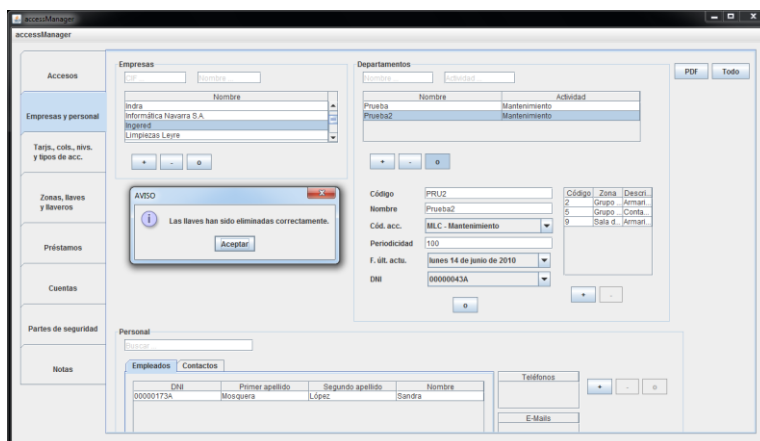
Salida observada





... y para la modificación de las llaves que pueden ser prestadas al departamento éstos otros:





**Evaluación/
conclusión**

El caso de uso '23 - Modificar departamento' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '23 - Modificar departamento'



Caso de prueba	4.10 - Caso de uso 26: Visualizar persona																																																																																	
Objetivo	Probar que los diversos tipos de personas que coexisten en el sistema pueden ser visualizados correctamente y en función del tipo al que representen.																																																																																	
Actividades	Loguear en el sistema, situarnos en la pestaña 'Empresas y personal' y probar a visualizar una persona de cada tipo, comprobando que tanto los datos generales como los específicos son mostrados correctamente.																																																																																	
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. • Utilidad phpMyAdmin. 																																																																																	
Datos de entrada	<p>Disponemos de los siguientes tipos de personas que probaremos a visualizar (escogidas aleatoriamente pero de modo que posean teléfonos, E-Mails y cargos para así poder probar todas las tablas):</p> <ul style="list-style-type: none"> • Residente: '00000006A' • Ocasional: '00000020A' • Vigilante: '00000002A' • Periódico: '00000163A' • Contacto: '00000168A' 																																																																																	
Salida esperada	<p>Deberemos obtener los siguientes datos respectivamente:</p> <ul style="list-style-type: none"> • Residente: <table border="1"> <thead> <tr> <th>DNI</th> <th>Primer apellido</th> <th>Segundo apellido</th> <th>Nombre</th> <th>Nivel</th> </tr> </thead> <tbody> <tr> <td>00000006A</td> <td>Nieves</td> <td>Lopez</td> <td>Rosa</td> <td>1</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Tfno</th> <th>Numero de telefono de contacto</th> <th>Email</th> <th>Direccion de correo electrónico de contacto</th> </tr> </thead> <tbody> <tr> <td>+34682028009</td> <td></td> <td>nieves@terra.es</td> <td></td> </tr> </tbody> </table> • Ocasional: <table border="1"> <thead> <tr> <th>DNI</th> <th>Primer apellido</th> <th>Segundo apellido</th> <th>Nombre</th> <th>Empresa</th> <th>Actividad</th> <th>Caducidad</th> <th>Notas</th> </tr> </thead> <tbody> <tr> <td>00000020A</td> <td>Boro</td> <td>Diaz</td> <td>Laura</td> <td>Endesa</td> <td>THTA</td> <td>2013-07-07 00:00:00</td> <td>Otros datos</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Tfno</th> <th>Numero de telefono de contacto</th> <th>Email</th> <th>Direccion de correo electrónico de contacto</th> </tr> </thead> <tbody> <tr> <td>+34948251726</td> <td></td> <td>laurab@gmail.com</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Código</th> <th>Zona</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>17</td> <td>Sala de condensadoras</td> <td>Armario 2</td> </tr> <tr> <td>19</td> <td>Sala de condensadoras</td> <td>Cuadros</td> </tr> </tbody> </table> • Vigilante: <table border="1"> <thead> <tr> <th>DNI</th> <th>Primer apellido</th> <th>Segundo apellido</th> <th>Nombre</th> <th>Código de vig.</th> </tr> </thead> <tbody> <tr> <td>00000002A</td> <td>Haydee</td> <td>Duck</td> <td>Graciela</td> <td>1</td> </tr> </tbody> </table> <p><i>Sin teléfonos o E-Mails.</i></p> • Periódico: <table border="1"> <thead> <tr> <th>DNI</th> <th>Primer apellido</th> <th>Segundo apellido</th> <th>Nombre</th> </tr> </thead> <tbody> <tr> <td>00000163A</td> <td>García</td> <td>Pérez</td> <td>Javier</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Tfno</th> <th>Numero de telefono de contacto</th> <th>Email</th> <th>Direccion de correo electrónico de contacto</th> </tr> </thead> <tbody> <tr> <td>+34603135809</td> <td></td> <td>garcia@conasa.es</td> <td></td> </tr> <tr> <td>+34948739172</td> <td></td> <td></td> <td></td> </tr> </tbody> </table> 	DNI	Primer apellido	Segundo apellido	Nombre	Nivel	00000006A	Nieves	Lopez	Rosa	1	Tfno	Numero de telefono de contacto	Email	Direccion de correo electrónico de contacto	+34682028009		nieves@terra.es		DNI	Primer apellido	Segundo apellido	Nombre	Empresa	Actividad	Caducidad	Notas	00000020A	Boro	Diaz	Laura	Endesa	THTA	2013-07-07 00:00:00	Otros datos	Tfno	Numero de telefono de contacto	Email	Direccion de correo electrónico de contacto	+34948251726		laurab@gmail.com		Código	Zona	Descripción	17	Sala de condensadoras	Armario 2	19	Sala de condensadoras	Cuadros	DNI	Primer apellido	Segundo apellido	Nombre	Código de vig.	00000002A	Haydee	Duck	Graciela	1	DNI	Primer apellido	Segundo apellido	Nombre	00000163A	García	Pérez	Javier	Tfno	Numero de telefono de contacto	Email	Direccion de correo electrónico de contacto	+34603135809		garcia@conasa.es		+34948739172			
DNI	Primer apellido	Segundo apellido	Nombre	Nivel																																																																														
00000006A	Nieves	Lopez	Rosa	1																																																																														
Tfno	Numero de telefono de contacto	Email	Direccion de correo electrónico de contacto																																																																															
+34682028009		nieves@terra.es																																																																																
DNI	Primer apellido	Segundo apellido	Nombre	Empresa	Actividad	Caducidad	Notas																																																																											
00000020A	Boro	Diaz	Laura	Endesa	THTA	2013-07-07 00:00:00	Otros datos																																																																											
Tfno	Numero de telefono de contacto	Email	Direccion de correo electrónico de contacto																																																																															
+34948251726		laurab@gmail.com																																																																																
Código	Zona	Descripción																																																																																
17	Sala de condensadoras	Armario 2																																																																																
19	Sala de condensadoras	Cuadros																																																																																
DNI	Primer apellido	Segundo apellido	Nombre	Código de vig.																																																																														
00000002A	Haydee	Duck	Graciela	1																																																																														
DNI	Primer apellido	Segundo apellido	Nombre																																																																															
00000163A	García	Pérez	Javier																																																																															
Tfno	Numero de telefono de contacto	Email	Direccion de correo electrónico de contacto																																																																															
+34603135809		garcia@conasa.es																																																																																
+34948739172																																																																																		



• Contacto:

DNI	Primer apellido	Segundo apellido	Nombre	Nombre
00000168A	Gonzalez	Cuadrado	Maribel	

Tfno Número de teléfono de contacto
+34691628315

Email Dirección de correo electrónico de contacto
gonzalez@comfio.es

Cargo
Responsable de personal

La espera:

accessManager

Empresas: Informática Navarra S.A.

Departamentos: [Empty]

Personal: [Empty]

Empleados:

DNI	Primer apellido	Segundo apellido	Nombre	Telefonos
00000000A	Navas	Hástor	Hástor	+3468202909
00000005A	Díaz	Boró	Isatila	
00000006A	Navas	Lopez	Rosa	
00000008A	Barbero	Carrido	Francisco Javier	
00000009A	Pinteno	Comas	Jose	

Partes de seguridad: DNI: 00000006A, Nombre: Rosa, Nivel: 1

Primer apellido: Lopez, Segundo apellido: Nuevo th.E-Mail

Salida observada

accessManager

Empresas: Occidental

Departamentos: [Empty]

Personal: [Empty]

Empleados:

DNI	Primer apellido	Segundo apellido	Nombre	Telefonos
00000020A	Boró	Díaz	Laura	+3494521726
00000031A	Riquas	Suero	Carolina	
00000034A	García	Moreno	Cristina	
00000040A	Llamas	Ayala	Mariano	
00000045A	Zaballos	García	Marta	

Partes de seguridad: DNI: 00000020A, Nombre: Laura, Empresa: Eadesa, Actividad: THTA - Trabajo, Caducidad: domingo 7 de julio de 20...

Primer apellido: Boró, Segundo apellido: Díaz

accessManager

Empresas: Cañor

Departamentos: Vigilancia con arma

Personal: [Empty]

Empleados:

DNI	Primer apellido	Segundo apellido	Nombre	Telefonos
00000001A	Sainza	Hernández	Rosa	+3491729262
00000002A	Haydee	Duck	Graciela	
00000003A	Casado	Sierra	José Manuel	

Partes de seguridad: DNI: 00000002A, Nombre: Graciela, Código de vig.: 1

Primer apellido: Haydee, Segundo apellido: Duck



Empresas	Departamentos
Cador	Asistencia Técnica
Condo	Credito Atención Usuario 24x7
Compañía Integral de Seguridad	Emergencias 24x7
Comasa	Mantenimiento

Empresas y personal	Personal
Empresas y personal	Empleado
Tarj., coll., niv. y tipos de acc.	
Zonas, Raves y Raveros	
Préstamos	Empleados
Cuentas	Contactos
Partes de seguridad	
Notas	

DNI	Primer apellido	Segundo apellido	Nombre	Telefonos
00000152A	Méndez	Signoret	Almudena	+3490315059
00000153A	Tamiet	Gallego	Patria	+34948739172
00000154A	García	Sanchez	Jorge	
00000163A	García	Pérez	Javier	
00000165A	Seguristain	Arribas	Verónica	

DNI	Primer apellido	Segundo apellido	Nombre	Telefonos
00000163A	García	Pérez	Javier	

Empresas	Departamentos
Eitor	Dep. Marketing
Gupost	Trabajo
GIM	
Indra	

Empresas y personal	Personal
Empresas y personal	Empleado
Tarj., coll., niv. y tipos de acc.	
Zonas, Raves y Raveros	
Préstamos	Empleados
Cuentas	Contactos
Partes de seguridad	
Notas	

DNI	Primer apellido	Segundo apellido	Nombre	Cargos	Telefonos
00000187A	Hernández	Estarke	Javier	Responsable de	+34948126792

DNI	Primer apellido	Segundo apellido	Nombre	Telefonos
00000187A	Hernández	Estarke	Javier	

Evaluación/
conclusión

El caso de uso '26 - Visualizar persona' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '26 - Visualizar persona'



Caso de prueba	4.11 - Caso de uso 27: Añadir persona
Objetivo	Verificar que el usuario podrá añadir las personas que desee, sean del tipo que sean, siempre y cuando cumpla con los formatos de datos exigidos para cada una.
Actividades	Acceder a la pestaña de empresas y personal y realizar varios intentos de inserción de personal comprobando los resultados que se obtienen.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos resultante del caso de prueba anterior.
Datos de entrada	<p>Atendiendo a la estructura del modelo relacional empleado, tenemos que:</p> <ul style="list-style-type: none">• Dos personas no podrán tener el mismo DNI.• Dos empleados no podrán desempeñar sus labores en varias empresas y departamentos.• Un contacto podrá pertenecer a varias empresas y departamentos.• Una persona ocasional podrá estar relacionada con varias empresas.• Todos los datos deben disponer del formato esperado. <p>Teniendo ésto presente, generaremos el siguiente conjunto de casos de prueba:</p> <ul style="list-style-type: none">• Residente:<ul style="list-style-type: none">‘00000000A, Apellido1, Apellido2, Nombre, 0’. (DNI existente)‘00000000A, Apellido1, Apellido2, Nombre, 0’. (DNI con formato erróneo)‘99999999A, Apellido1, Apellido2, Nombre, 0’. (Válido)• Ocasional:<ul style="list-style-type: none">‘00000020A, Apellido1, Apellido2, Nombre, Endesa, THTA, 14 de junio de 2010, ’. (DNI con formato erróneo)‘00000020A, Boro, Diaz, Laura, Iberdrola, THTA, 14 de junio de 2010, ’. (Ocasional existente pero con una nueva empresa)‘99999998A, Apellido1, Apellido2, Nombre, Empresa, THTA, 14 de junio de 2010, ’. (Válido)• Vigilante:<ul style="list-style-type: none">‘999999997A, Apellido1, Apellido2, Nombre, 9’. (DNI con formato erróneo)‘99999997A, Apellido1, Apellido2, Nombre, 0’. (Código de vigilante existente)‘99999997A, Apellido1, Apellido2, Nombre, 9’. (Válido)• Periódico (departamento ‘Emergencias 24x7’ de ‘Conasa’):<ul style="list-style-type: none">‘000000001A, Apellido1, Apellido2, Nombre’. (DNI con formato erróneo)‘00000163A, García, Pérez, Javier’. (Persona existente en otra empresa-departamento)‘99999996A, Apellido1, Apellido2, Nombre’. (Válido)



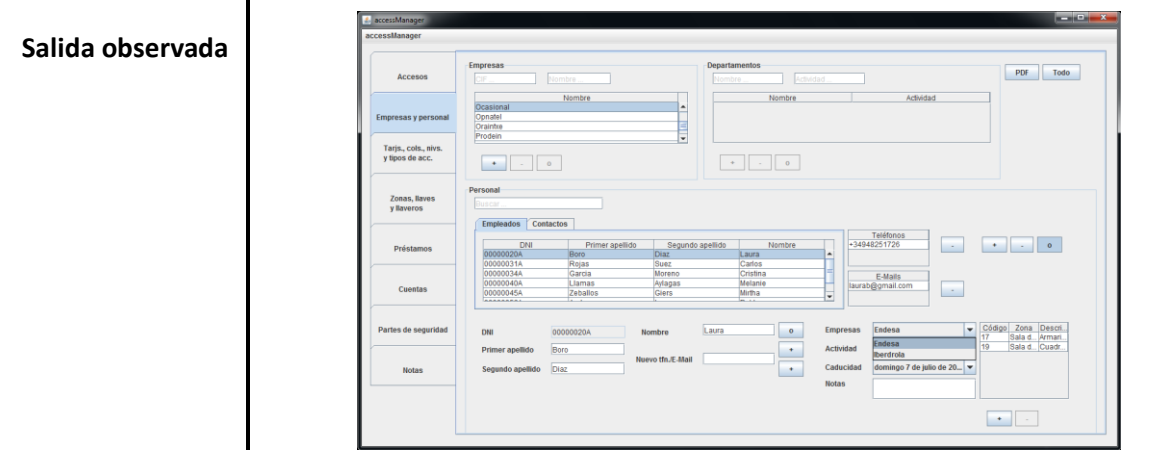
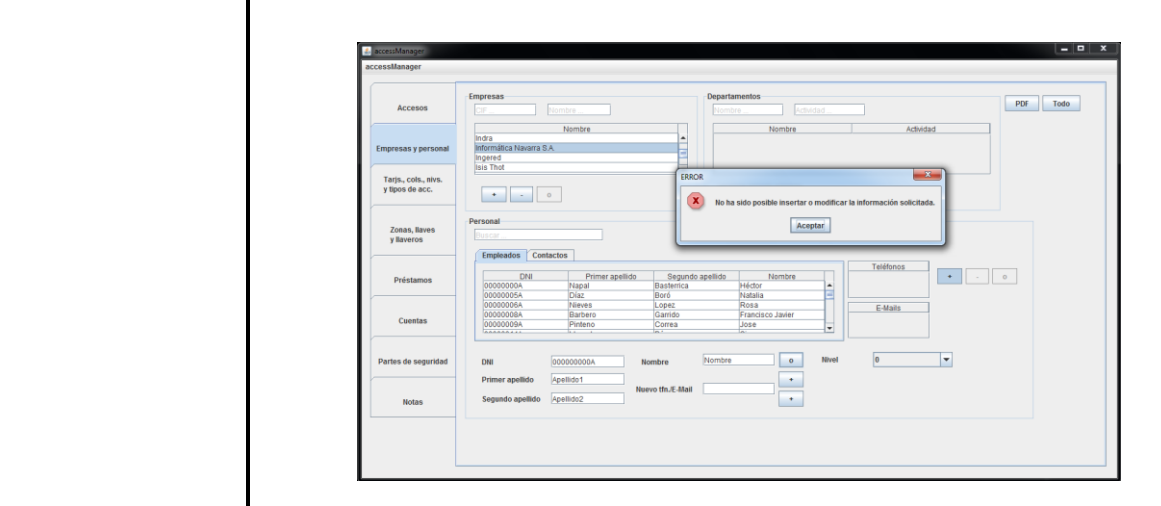
- Contactos (mismo departamento y empresa):
'00000001A, Apellido1, Apellido2, Nombre'. (DNI con formato erróneo)
'0000011A, Hervas, Pardo, Lidia'. (Persona existente en otra empresa-departamento)
'99999995A, Apellido1, Apellido2, Nombre'. (Válido)

Salida esperada

Podemos resumir los resultados esperados en:

- Residente: Error, error, inserción.
- Ocasional: Error, inserción, inserción.
- Vigilante: Error, inserción.
- Periódico (departamento 'Emergencias 24x7' de 'Conasa'): Error, error, inserción.
- Contactos (mismo departamento y empresa): Error, inserción, inserción.

La esperada. He aquí a modo de muestra algunos de los resultados obtenidos:



Salida observada

Aprovechamos en este punto para comprobar y certificar que el JComboBox de relleno de formulario a partir de las personas ocasionales existentes funciona correctamente, así como la modificación del valor de los campos específicos al personal una vez se selecciona una empresa distinta en el JComboBox de empresas.

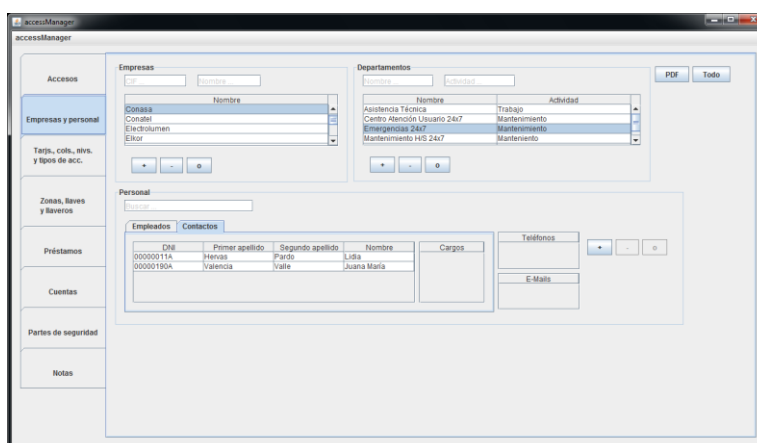
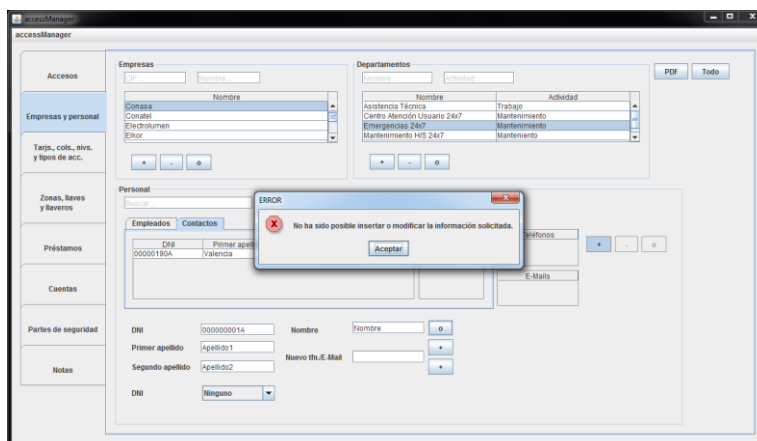


The screenshot shows the 'accessManager' application interface. A modal dialog box with a red 'X' icon and the title 'ERROR' is displayed in the center. The message inside the dialog reads: 'No ha sido posible insertar o modificar la información solicitada.' (It was not possible to insert or modify the requested information.) Below the message is an 'Aceptar' (Accept) button. The background interface is partially obscured but shows sections for 'Empresas', 'Departamentos', and 'Personal'.

The screenshot shows the 'accessManager' application interface. The 'Personal' section is active, displaying a table of employees. The table has columns for 'DNI', 'Primer apellido', 'Segundo apellido', 'Nombre', and 'E-mails'. The data in the table is as follows:

DNI	Primer apellido	Segundo apellido	Nombre	E-mails
0000001A	Sainas	Hernández	Rosa	
0000002A	Haydee	Duck	Graciela	
0000003A	Lasado	Sierra	José Manuel	
9999997A	Apellidos1	Apellidos2	Nombre	

The screenshot shows the 'accessManager' application interface. A modal dialog box with a red 'X' icon and the title 'ERROR' is displayed in the center. The message inside the dialog reads: 'No ha sido posible insertar o modificar la información solicitada.' (It was not possible to insert or modify the requested information.) Below the message is an 'Aceptar' (Accept) button. The background interface is partially obscured but shows sections for 'Empresas', 'Departamentos', and 'Personal'.



Aquí probamos de nuevo el JComboBox que rellena los campos específicos de cada persona, pero esta vez desplegando sólo los DNIs de los contactos ya existentes en el sistema.

**Evaluación/
conclusión**

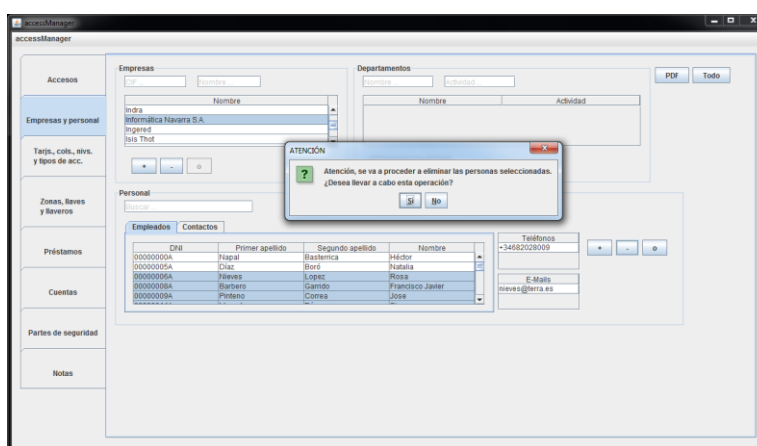
El caso de uso '27 - Añadir persona' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '27 - Añadir persona'

Caso de prueba	4.12 - Caso de uso 28: Borrar persona
Objetivo	Chequear que el usuario podrá realizar el borrado de cualquier conjunto de personas sin que el sistema termine en una situación de error.
Actividades	Acceder a la segunda pestaña de accessManager y realizar el borrado múltiple en cada grupo de personas existentes, comprobando que dejan de formar parte del sistema.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior.

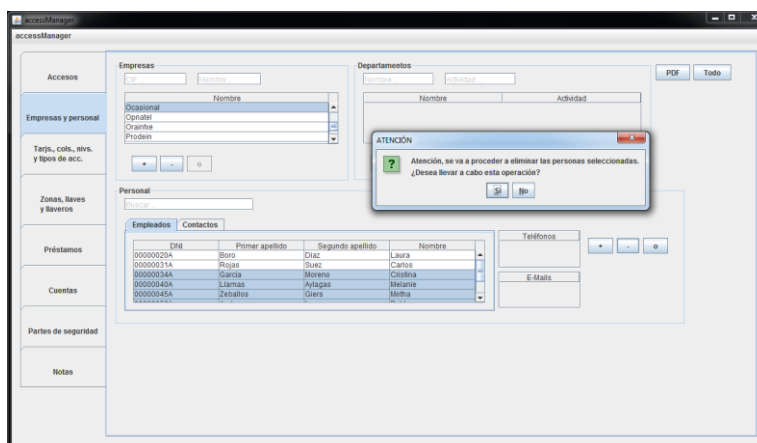
Realizaremos los siguientes borrados (todas las personas salvo las dos primeras en todos los casos) abarcando todos los grupos de personas de los que disponemos:

- Residente:



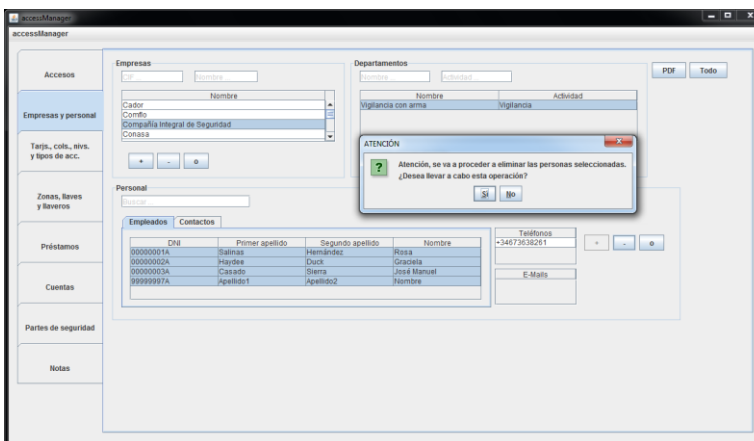
- Ocasional:

Datos de entrada

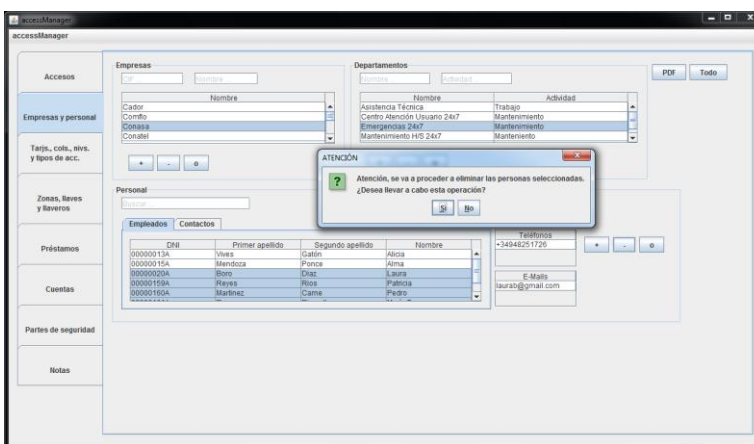




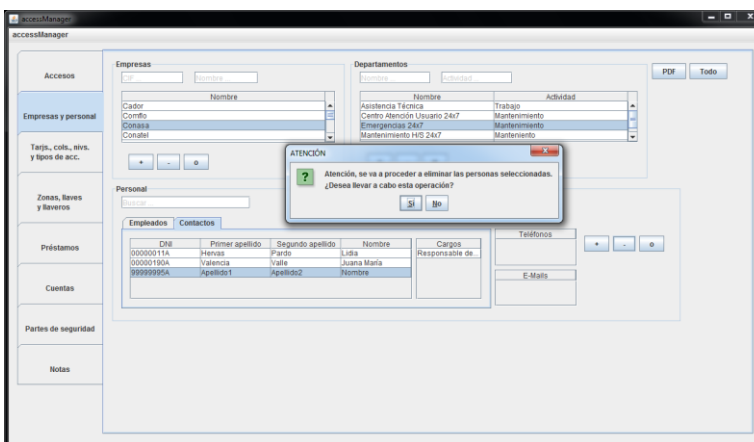
- Vigilante:



- Periódico ('Conasa - Emergencias 24x7'):



- Contactos ('Conasa - Emergencias 24x7'):

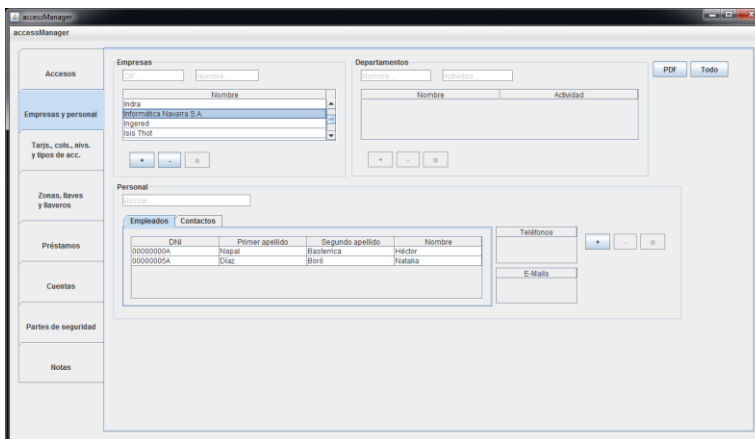




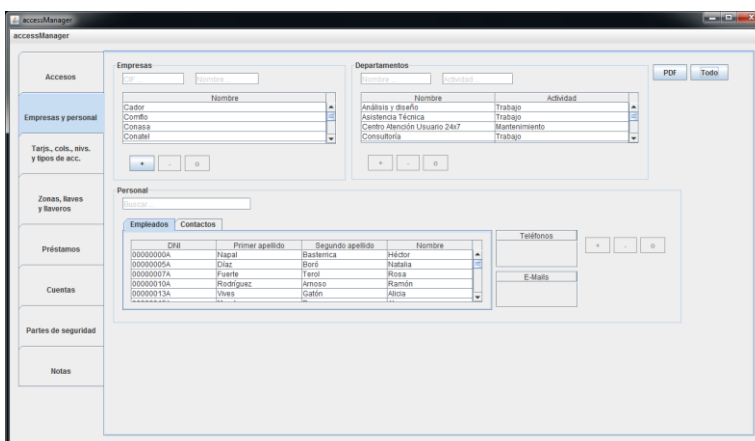
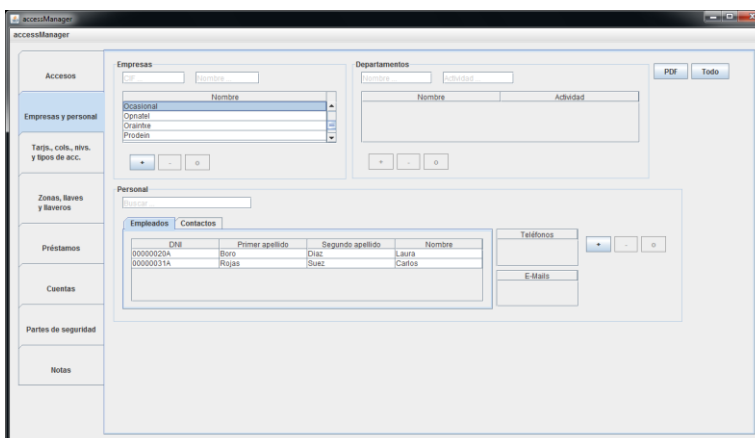
Salida esperada

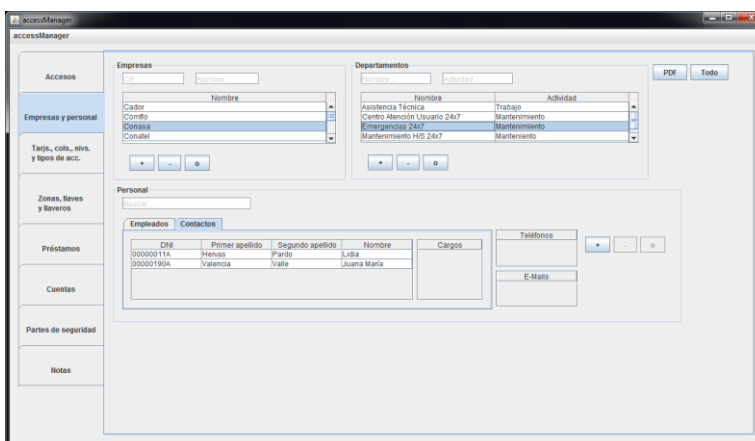
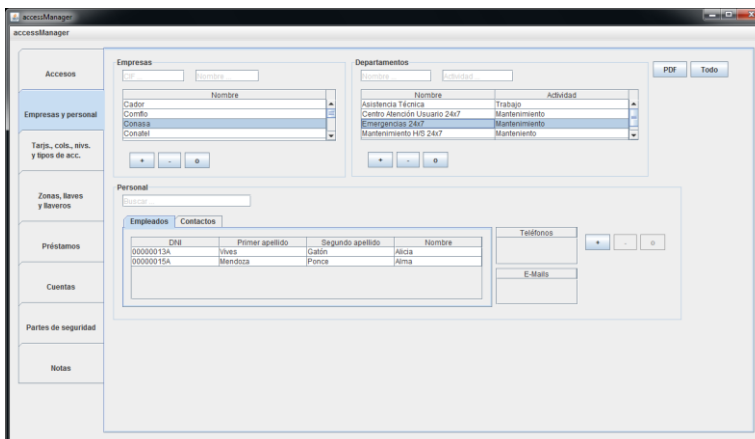
En cada borrado deberán quedar las dos primeras personas de cada tabla, que se visualizan en la sección 'Datos de entrada', salvo para el caso del departamento 'Vigilancia con arma' de 'Compañía Integral de Seguridad', donde borraremos todo el personal y contactos, lo que deberá hacer que tanto la empresa como el departamento dejen de formar parte del sistema al haberse quedado sin personal.

La esperada. Por orden obtenemos:



Salida observada





**Evaluación/
conclusión**

El caso de uso '28 - Borrar persona' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '28 - Borrar persona'

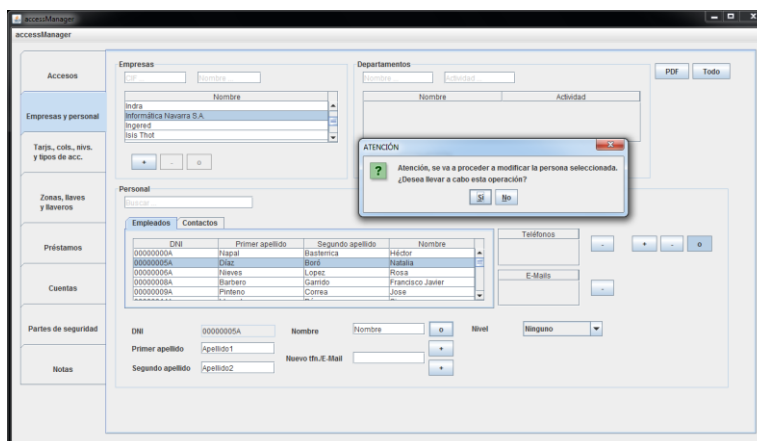
Caso de prueba	4.13 - Caso de uso 29: Modificar persona
Objetivo	Asegurar que el proceso de modificación de los datos que componen una persona se desarrolla correctamente sea cual sea el tipo de persona.
Actividades	Desplegar accessManager, situarnos en la pestaña de empresas y personal y probar a modificar todos los tipos de persona, comprobando que nunca se produce ningún tipo de error si se introducen datos válidos.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba.

Hagamos un breve inciso para aligerar esta prueba. Los casos a probar son exactamente los mismos que los empleados en la adición de personal al sistema. En ese caso comprobamos que el sistema siempre recogía bien los datos y mostraba error en caso de que éstos tuvieran un formato erróneo. Bien, como el proceso de modificación es igual que el de adición solo que en vez de insertar el objeto se recuperará el seleccionado en la tabla y se actualizará con el nuevo, si nos damos cuenta no es necesario probar los casos que induzcan a errores, ya que son los mismos que los tratados en el caso de prueba 27. Teniendo en cuenta ésto, probaremos a actualizar con datos correctos una persona aleatoria de cada tipo, certificando así que la recuperación de las personas a modificar (constructores) y el proceso de modificación junto con sus métodos (*updateXXXXBD*) han sido implementados correctamente.

A continuación las actualizaciones que probaremos a realizar:

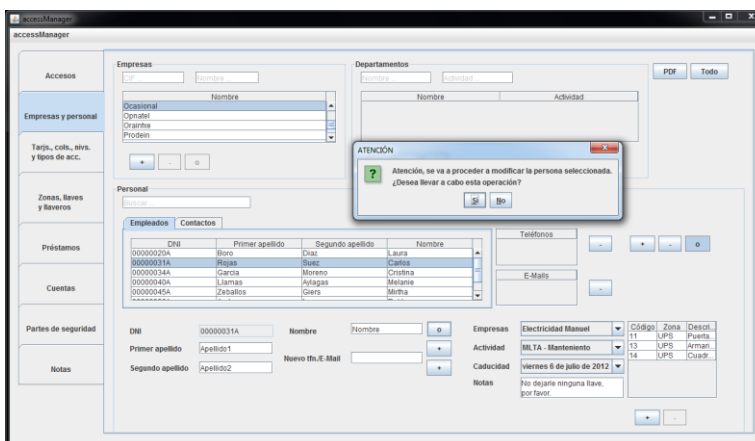
- Residente:

Datos de entrada

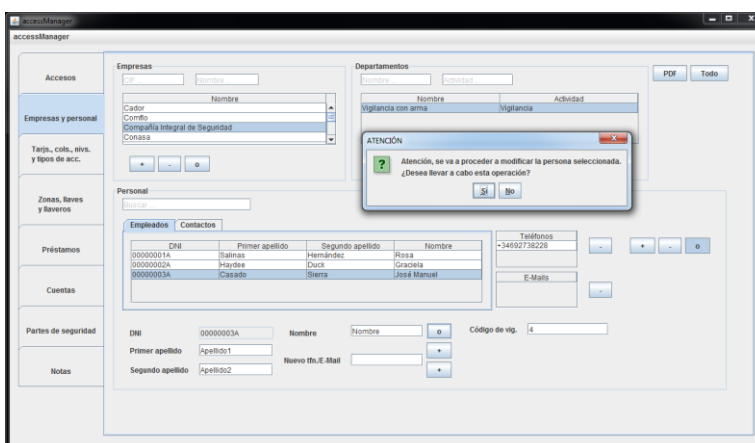




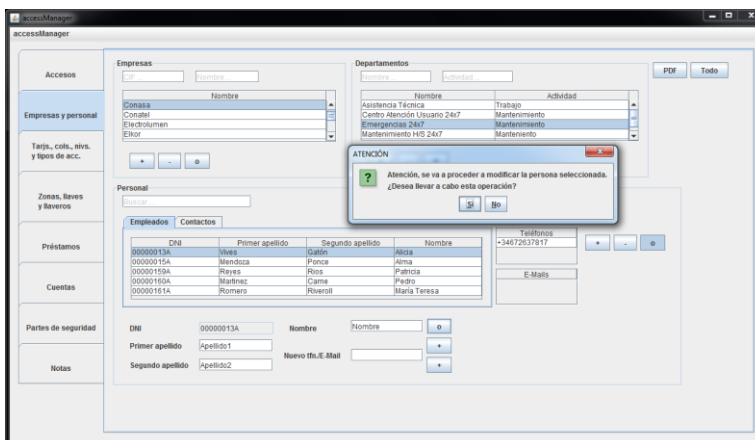
- Ocasional:



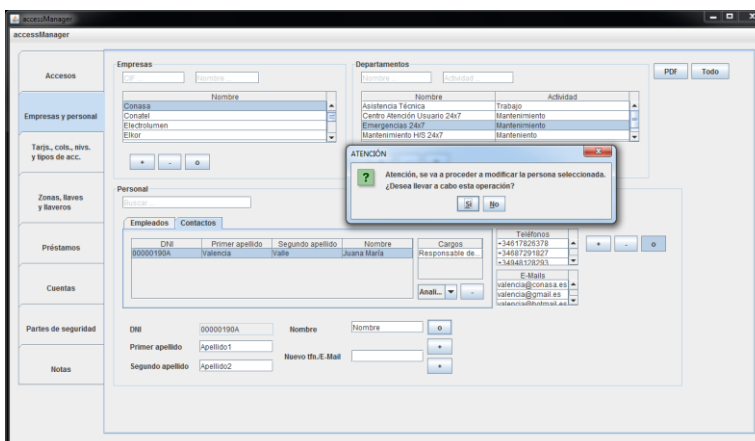
- Vigilante:



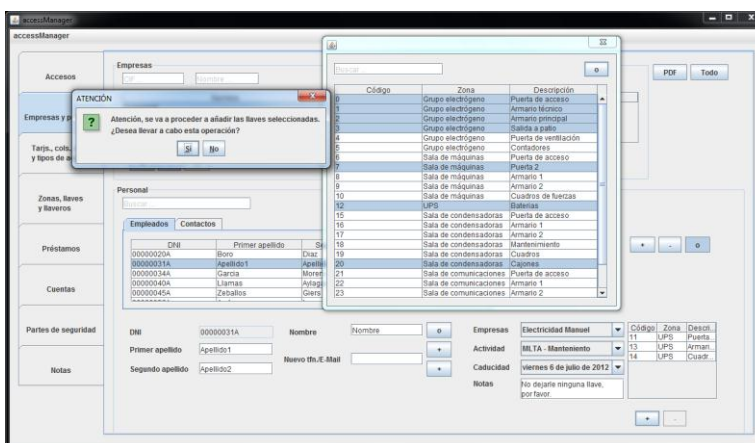
- Periódico:



- Contactos:



Una vez que hemos asegurado que la actualización de los datos tanto generales como los específicos a cada tipo de persona es siempre correcta, nos quedaría por probar la adición y substracción de llaves prestables al personal ocasional. Cualquier persona nos valdrá, aunque optaremos por una que disponga ya de alguna llave asignada para así comprobar que el diálogo de llaves se inicializa como debe. Optando por usar la persona con DNI '00000031A' probamos a añadir las siguientes llaves...



... y posteriormente a eliminar todas menos la última.

Salida esperada

Todas las actualizaciones propuestas deberán llevarse a cabo sin ningún tipo de problema, actualizando los datos existentes con los proporcionados. En la segunda parte de la prueba las llaves seleccionadas pasarán a formar parte de la tabla Prestamo_Ocasional en la base de datos, así como de la de llaves de la interfaz gráfica. El caso de prueba deberá finalizar única y exclusivamente con la llave 20 en el modelo de dicha tabla al haber mandado eliminar todas las existentes menos ésta.



La espera. Según el orden establecido en 'Datos de entrada' obtenemos las siguientes salidas por pantalla:

accessManager

Empresas: Informática Navarra S.A.

Departamentos: (Empty)

Personal:

DNI	Primer apellido	Segundo apellido	Nombre	Telefonos
0000000A	Blasco	Héctor		
0000000A	Apellido1	Apellido2	Nombre	
0000000A	Rivas	Rosa		
0000000A	Barbero	Francisco Javier		
0000000A	Pirreno	Jose		

Salida observada

accessManager

Empresas: Occidental

Departamentos: (Empty)

Personal:

DNI	Primer apellido	Segundo apellido	Nombre	Telefonos
0000002A	Boro	Dia	Laura	
0000003A	Apellido1	Apellido2	Nombre	
0000004A	Garcia	Moreno	Christina	
0000004A	Llamos	Arlapas	Marta	
0000004A	Zabaltos	Olga		

Partes de seguridad: DNI: 0000031A, Nombre: Nombre, Empresa: Electricidad Manual, Código: T1, Zona: LPS, Descripción: Puerta, Primer apellido: Apellido1, Actividad: MLTA - Mantenimiento, T3, LPS, Arman, Segundo apellido: Apellido2, Nuevo th.E-Mail: Nuevo th.E-Mail, Caducidad: Viernes 6 de julio de 2012, T4, LPS, Cuarte, Notas: No dejarte ninguna llave, por favor

accessManager

Empresas: Cador

Departamentos: Vigilancia con arma, Vigilancia

Personal:

DNI	Primer apellido	Segundo apellido	Nombre	Telefonos
0000001A	Balmes	Hernández	Rosa	+3492738228
0000002A	Haydee	Duck	Gracela	
0000003A	Apellido1	Apellido2	Nombre	

Partes de seguridad: DNI: 0000003A, Nombre: Nombre, Código de vig: 4, Primer apellido: Apellido1, Segundo apellido: Apellido2, Nuevo th.E-Mail: Nuevo th.E-Mail



accessManager

Empresas

Departamentos

Personal

Empleados	Contactos		
DNI	Primer apellido	Segundo apellido	Nombre
0000013A	Apellido1	Apellido2	Nombre
0000015A	Alencija	Ponce	Alma
0000016A	Rivas	Rios	Patricia
0000016A	Martinez	Carne	Padro
0000016A	Romero	Riverol	Maria Teresa

Partes de seguridad

Notas

accessManager

Empresas

Departamentos

Personal

Empleados	Contactos					
DNI	Primer apellido	Segundo apellido	Nombre	Cargos	Responsable de	Telefonos
0000019A	Apellido1	Apellido2	Nombre			+34917629378
						+34987211627
						+34948132931

Partes de seguridad

Notas

accessManager

Empresas

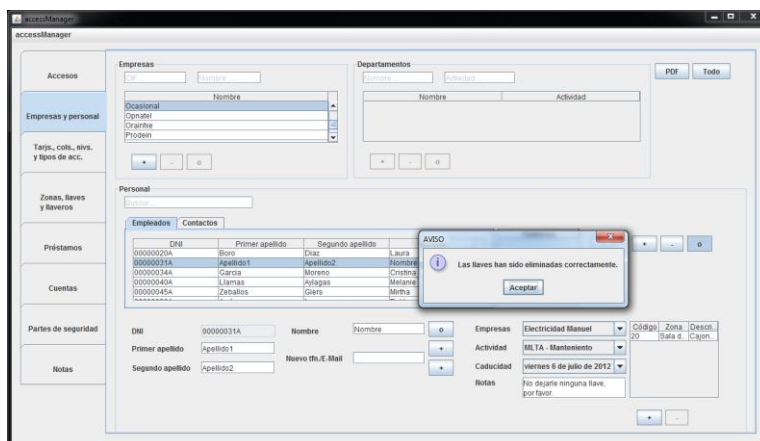
Departamentos

Personal

Empleados	Contactos						
DNI	Primer apellido	Segundo apellido	Nombre	Empresas	Código	Zona	Descr.
0000020A	Buro	Diaz	Laura	Electricidad Manual	0	Grup	Parti.
0000031A	Aguilera	Apelido	Norma	MLTA - Mantenimiento	1	Grup	Arma.
0000034A	Garcia	Moreno	Cristina		2	Grup	Arma.
0000040A	Llamas	Ayagas	Mariana		3	Grup	Salid.
0000045A	Zabaltos	Gera	María		7	Grup	Parti.
					11	LPS	Parti.
					12	LPS	Subst.

Partes de seguridad

Notas



Evaluación/
conclusión

El caso de uso '29 - Modificar persona' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '29 - Modificar persona'



Caso de prueba	4.14 - Casos de uso 30, 31, 32 y 33: Añadir o borrar teléfono o E-Mail
Objetivo	Verificar que además de poder modificar los datos generales y específicos de cada persona, también podremos añadirles y/o eliminarles teléfonos y/o E-Mails.
Actividades	Loguear en el sistema, situarnos en la tab 'Empresas y personal' y probar a añadir teléfonos e E-Mails a todos los tipos de personas existentes, comprobando que el proceso siempre se realiza correctamente si proporcionamos datos con un formato válido.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos resultante del caso de prueba anterior.
Datos de entrada	<p>Recordemos que la modificación del listado de teléfonos e E-Mails se encuadra en el proceso de modificación de una persona y está compuesta de:</p> <ul style="list-style-type: none">• Introducción y eliminación de datos en las correspondientes tablas.• Actualización de la información en la base de datos. <p>En la primera parte del proceso podremos ofrecer al sistema teléfonos e E-Mails con formato válido y no válido. Éste será el primer punto a probar. En la segunda parte la actualización se produce recogiendo los datos validados de las tablas, con que el único punto de fallo podría estar en el proceso de lectura de tablas o métodos que se encargan de las actualizaciones.</p> <p>Realizado este análisis previo probaremos:</p> <ul style="list-style-type: none">• En primer lugar, que el paso de los datos (con la correspondiente validación) del formulario a las tablas de teléfonos e E-Mails se realiza correctamente, para lo que probamos a introducir: Teléfonos: Vacíos Excesivamente largos: '01234567890123' E-Mails: Vacíos Sin el carácter '@': 'a' Excesivamente largos: 'direcciondecorreoelectronicodeprueba@servidordeprueba.com'



- A continuación, que podemos añadir y eliminar teléfonos e E-Mails a cualquier tipo de persona. Escogeremos para la ocasión la siguiente combinación de teléfonos e E-Mails a añadir y eliminar:

	DNI		Teléfonos	
	DNI	Existentes	A añadir	A eliminar
Resid.	...6A	+34682028009	+340 <i>+341</i>	+34682028009 <i>+340</i>
Ocas.	...34A	Ninguno	+340 <i>+341</i>	+341
Vig.	...1A	+34673638261	+340 <i>+341</i>	+340 <i>+341</i>
Per.	...163A	+34603135809 <i>+34948739172</i>	+340	+340 <i>+34603135809</i>
Cont.	..167A	+34673826356	+340 <i>+341</i>	+34673826356

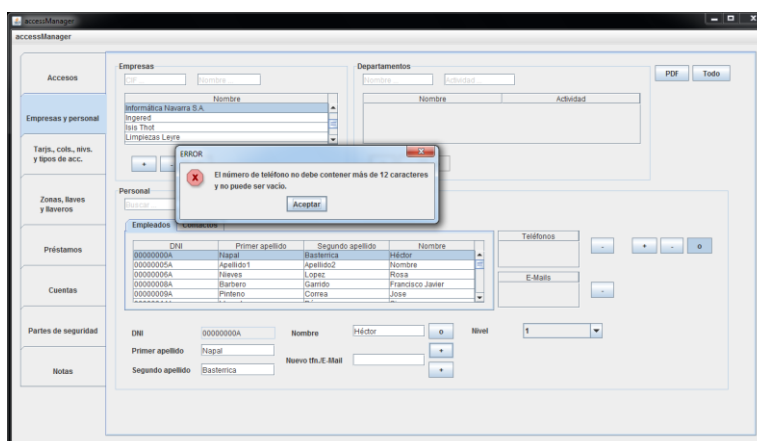
	DNI		E-Mails	
	DNI	Existentes	A añadir	A eliminar
Resid.	...6A	nieves@terra.es	a@a	a@a nieves@terra.es
Ocas.	...34A	Ninguno	a@a <i>b@b</i>	a@a
Vig.	...1A	Ninguno	a@a <i>b@b</i>	b@b
Per.	...163A	<i>garcia@conasa.es</i>	a@a <i>b@b</i>	b@b
Cont.	..167A	rojas@cador.com	a@a <i>b@b</i>	rojas@cador.com

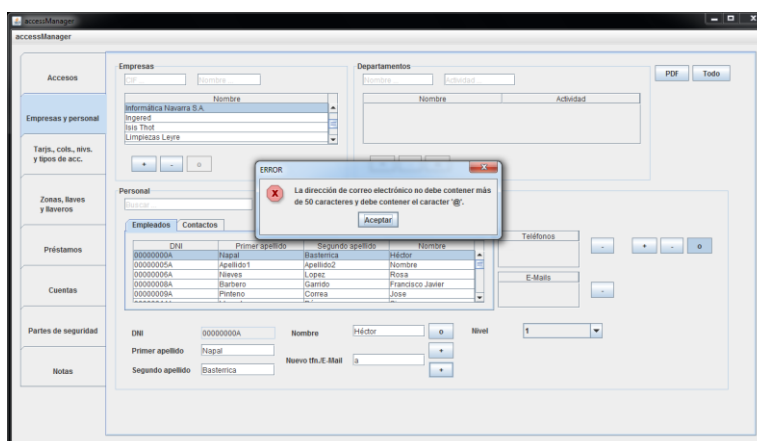
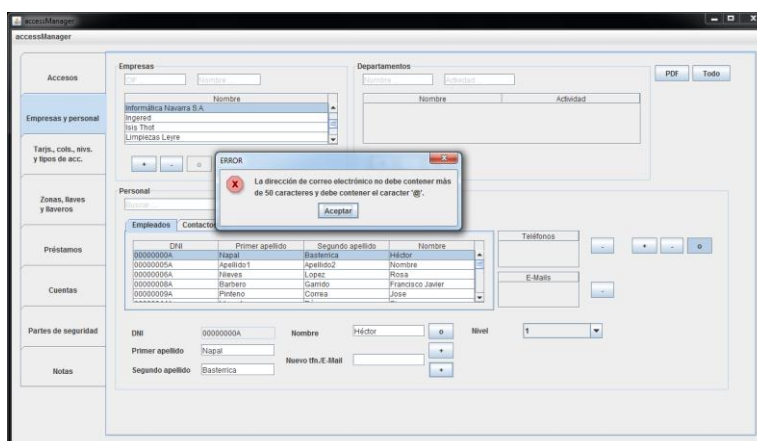
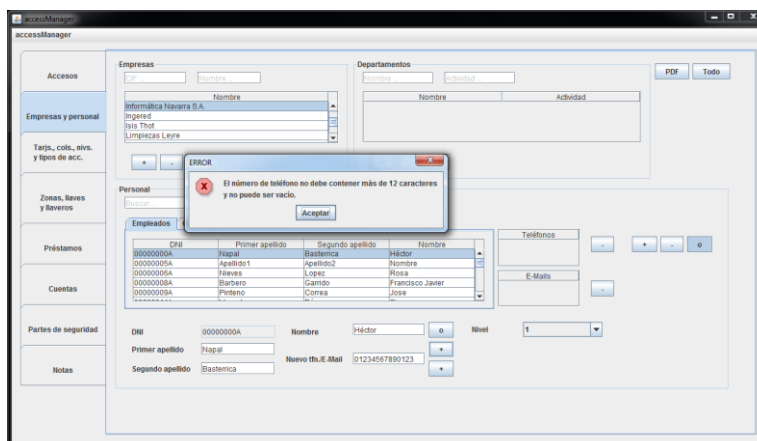
Salida esperada

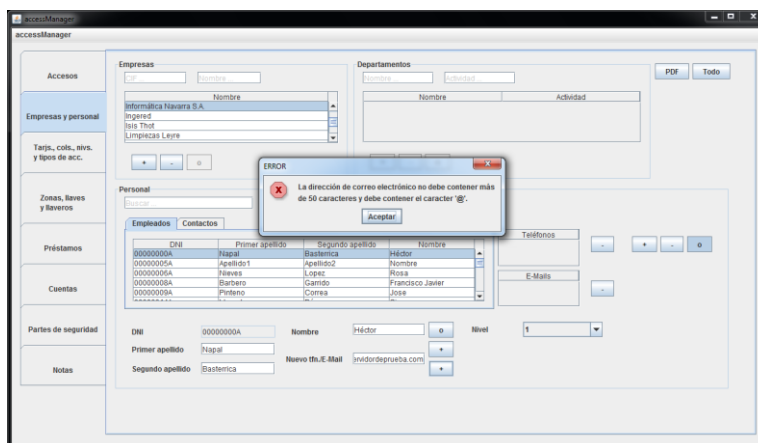
Las tablas de teléfonos e E-Mails deberán contener los elementos marcados con letra cursiva en la tabla superior.

Salida observada

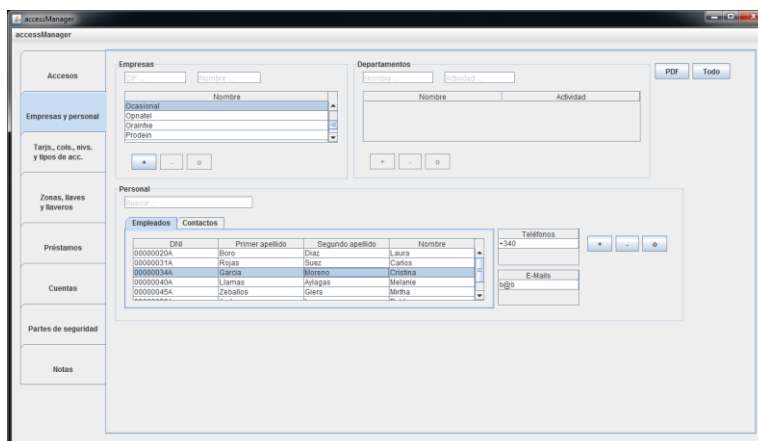
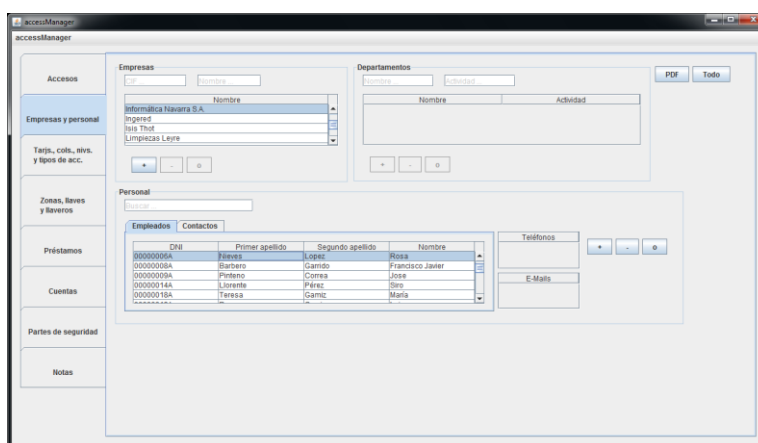
La esperada. Todos los formatos erróneos de teléfonos e E-Mails son capturados...







... y el contenido final de las tablas es el esperado:





Empresas	Departamentos
Compañía Integral de Seguridad	Vigilancia con arma
Conasa	Vigilancia
Conatel	
Electrolumen	

Personal				
Empleados	Contactos			
DNI	Primer apellido	Segundo apellido	Nombre	Teléfonos
0000001A	Ruinas	Hernández	Rosa	+3497353621
0000002A	Pedres	Dock	Graciela	
0000003A	Casado	Sierra	José Manuel	

Empresas	Departamentos
Conasa	Centro Atención Usuario 24x7
Conatel	Emergencias 24x7
Electrolumen	Mantenimiento
Elkor	Mantenimiento NIS 24x7
	Sistemas

Personal				
Empleados	Contactos			
DNI	Primer apellido	Segundo apellido	Nombre	Teléfonos
0000152A	Méndez	Sigüenza	Almudena	+34948739172
0000153A	Namiet	Salazar	Patricia	
0000154A	García	Sanchez	Jorge	
0000155A	García	Perez	Javier	
0000156A	Beguirain	Amatain	Verónica	

Empresas	Departamentos
Cañor	Diseño y construcción
Compañía Integral de Seguridad	Trabajo
Conasa	

Personal					
Empleados	Contactos				
DNI	Primer apellido	Segundo apellido	Nombre	Cargos	Teléfonos
0000167A	Rojas	García	Claudia	Responsable de	+340 +341

**Evaluación/
conclusión**

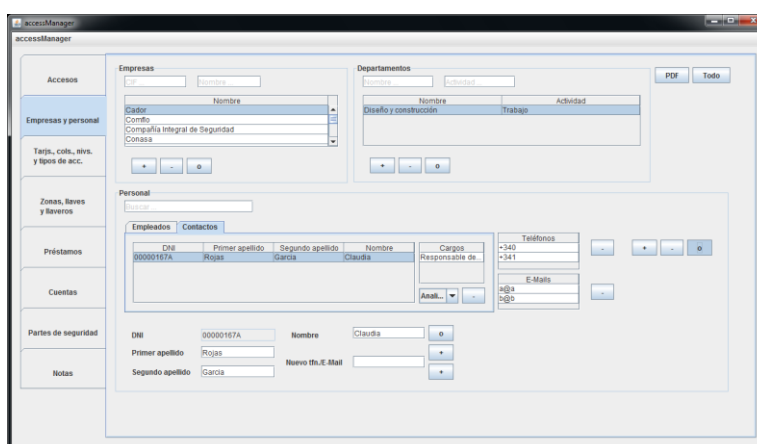
Los casos de uso '30, 31, 32 y 33 - Añadir o borrar teléfono o E-Mail' han sido realizados satisfactoriamente.

Caso de prueba para los casos de uso '30, 31, 32 y 33 - Añadir o borrar teléfono o E-Mail'

Caso de prueba	4.15 - Casos de uso 34 y 35: Añadir o borrar cargo
Objetivo	Asegurar que es posible añadir o eliminar cualquier conjunto de cargos a un contacto registrado en el sistema.
Actividades	Situarnos en la segunda pestaña de accessManager y modificar un cargo al azar, comprobando que podemos añadirle y eliminarle los cargos que creamos convenientes.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior.

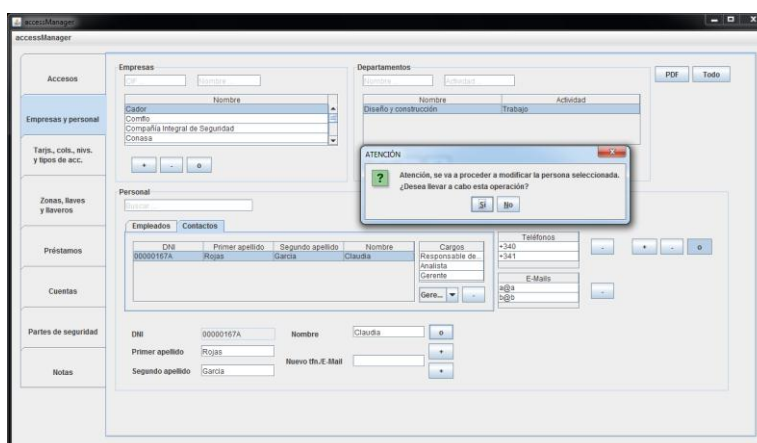
Las empresas 'Informática Navarra S.A.' y 'Ocasional' no pueden disponer de ningún contacto y las empresas-departamentos restantes son tratadas por igual, con que probaremos a seleccionar cualquier contacto (p.e. el de DNI '00000167A') y modificar los cargos que desempeña en res fases:

- Comprobando su estado inicial:

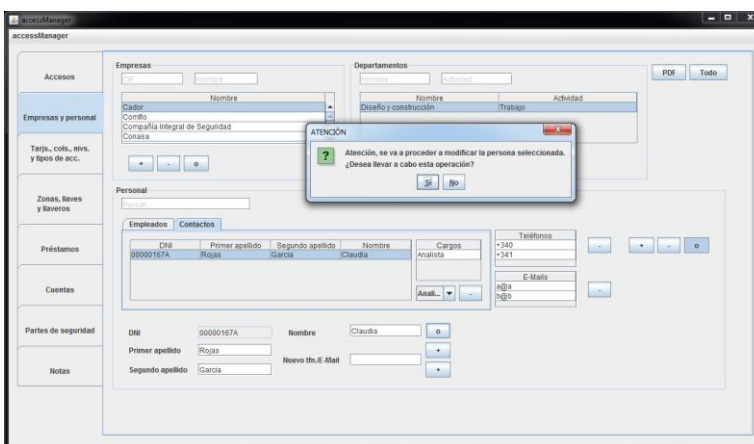


Datos de entrada

- Añadiéndole nuevos cargos:



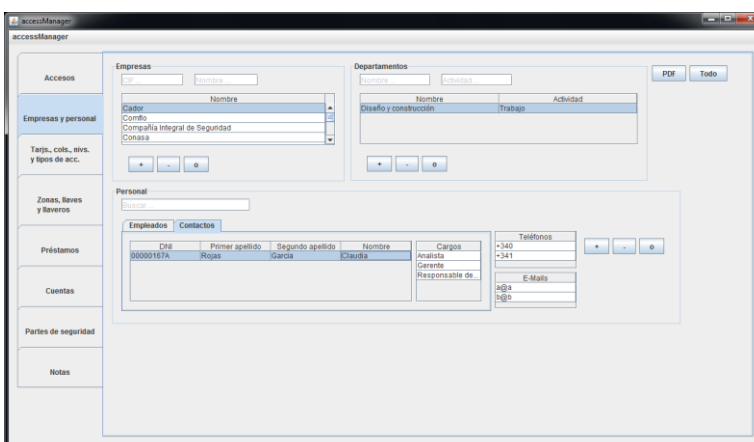
- Eliminándole todos los cargos menos el primero:



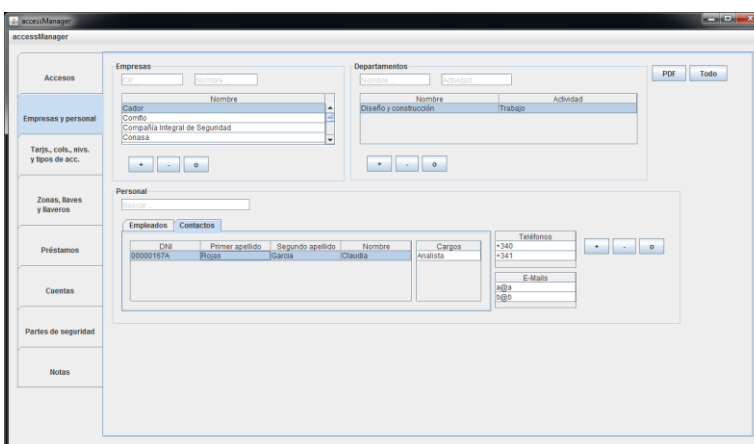
Salida esperada

Tras la adición de cargos el contacto pasará a poseer todos los cargos asignados (existentes). Posteriormente el borrado lo dejará desempeñando únicamente el papel de 'Analista'.

La esperada:



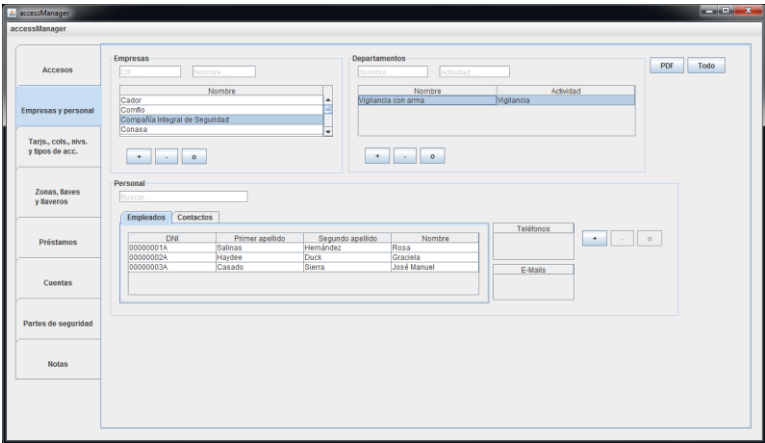
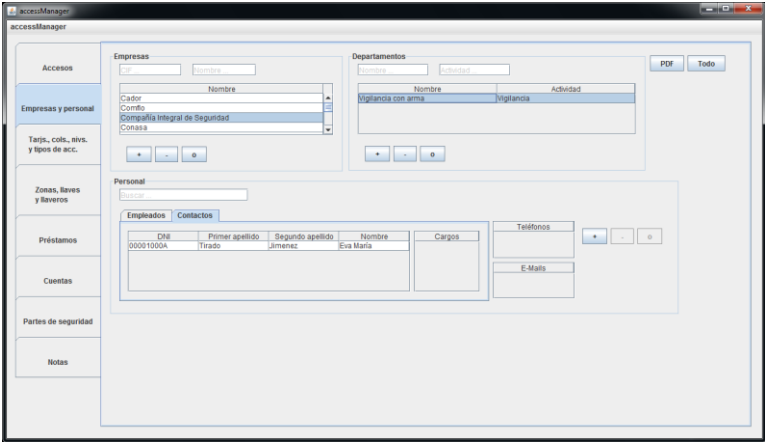
Salida observada



**Evaluación/
conclusión**

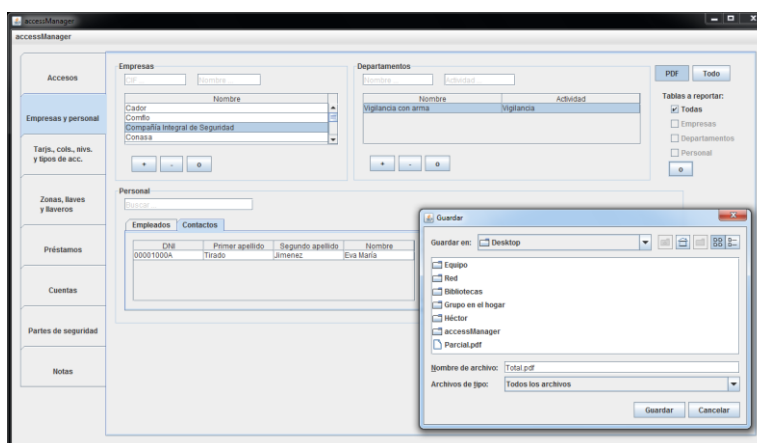
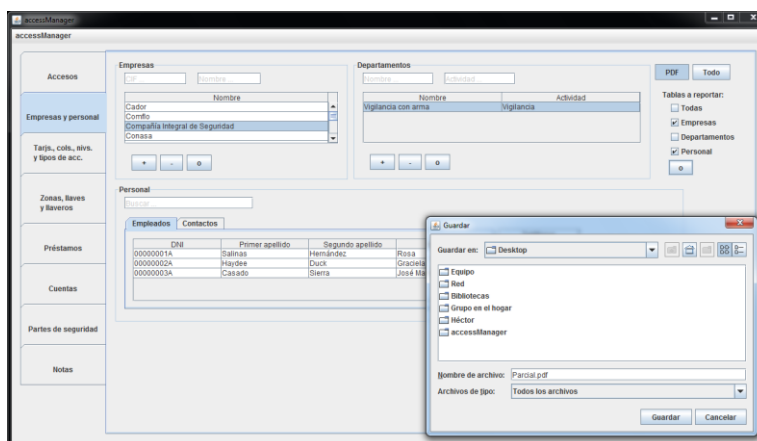
Los casos de uso '34 y 35 - Añadir o borrar cargo' han sido realizados satisfactoriamente.

Caso de prueba para los casos de uso '34 y 35 - Añadir o borrar cargo'
accessManager · Iteración C4: Empresas y personal · Página 109 de 114

Caso de prueba	4.16 - Casos de uso 18, 24 y 36: Listar empresas, departamentos y personal
Objetivo	Comprobar que accessManager es capaz de elaborar los informes que podamos pedirle en la pestaña de empresas y personal y que podrán contener cualquier combinación de las tablas de empresas, departamentos y empleados/contactos.
Actividades	Loguear en el sistema, acudir a la tab 'Empresas y personal', definir el contenido de las tablas a nuestro gusto, pedir al sistema que genere dos reportes (uno parcial y otro con todas ellas) y verificar que el contenido de éstos se corresponde con los registros de la base de datos.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior. • Aplicación Adobe Reader.
Datos de entrada	<p>Vamos a generar una búsqueda que nos muestre la empresa de vigilancia del centro (por ejemplo):</p>  



Con el resultado deseado poblando las tablas requerimos a accessManager que nos genere un listado parcial de empresas y empleados y otro total de empresas, departamentos y contactos:



Salida esperada

- Esperamos que el sistema deposite en 'Desktop' dos archivos PDF:
- Parcial.pdf, conteniendo todas las empresas y el personal trabajador de 'Compañía Integral de Seguridad – Vigilancia con arma' (ver primera imagen de 'Datos de entrada').
 - Total.pdf, que recogerá todas las empresas, departamentos y contactos pertenecientes a 'Compañía Integral de Seguridad – Vigilancia con arma' (ver segunda imagen de 'Datos de entrada').



6.3. De sistema

Ha quedado más que demostrado con el desarrollo de este valioso documento que hemos incorporado al sistema todos los casos de uso propuestos al inicio de la iteración y que éste sigue funcionando de manera correcta.

7. Evaluación de la iteración

Llegados casi al final de la iteración disponemos de la pestaña 'Empresas y personal' implementada a falta de rediseñarla en la futura iteración 'C12 - Diseño, personalización y configuración'.

La verdad es que no nos hemos enfrentado a una iteración sencilla, sino todo lo contrario. Ésta recoge tres tipos de entes (empresas, departamentos y personal) los cuales representan a nada más y nada menos que 11 tablas del modelo relacional diseñado tres iteraciones atrás, que si las combinamos con los tres tipos de operaciones existentes (inserción, borrado y actualización) y relacionamos con todos los elementos gráficos necesarios dan lugar a un buen número de combinaciones de situaciones que debemos tener en cuenta e implementar con sumo cuidado.

Todos los casos de uso han sido incorporados satisfactoriamente a tenor de los resultados obtenidos en la fase de pruebas y tras probar toda la funcionalidad de la pestaña una vez más la creemos lista para darle el visto bueno y acudir a la reunión de despliegue que tenemos concertada con el cliente.

8. Despliegue

El quinto día del mes de marzo nos dirigimos a la sede de Informática Navarra S.A. con este documento y el prototipo desarrollado hasta el momento con el fin de evaluarlo conjuntamente con el cliente. Pedro Martínez nos recibe y acudimos juntos a la sala acondicionada para dicha reunión.

Comenzamos la reunión explicándole qué hemos desarrollado y ofreciéndole una breve perspectiva de todas las actividades y resultados que hemos obtenido a lo largo de la iteración. Acto seguido procedemos con la parte fundamental de la reunión: la prueba de todos y cada uno de los casos de uso codificados. Todas las pruebas dan como resultado los obtenidos por nosotros en nuestra fase interna de pruebas y tras realizarlas solicitamos al cliente su opinión. Éste nos transmite que ve con buenos ojos continuar con el desarrollo del proyecto pues toda la funcionalidad que la nueva pestaña ofrece le parece suficiente para poder manejar eficientemente a todas las empresas, departamentos y personal que acceden al centro o tienen relación con Informática Navarra S.A..



Pamplona, a 5 de marzo de 2010.

Héctor Napal Basterrica
Responsable del proyecto

Pedro Martínez Pérez
*Jefe de área de producción de
Informática Navarra S.A.*



accessManager

Desarrollo de un sistema de control de accesos a un centro

Iteración C5: Tarjetas, tipos de acceso y cargos

Fecha de entrega: 10 de marzo de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Estructura del documento

2. Estado del proyecto

3. Análisis

- 3.1. Planificación de la iteración
- 3.2. Análisis de los casos de uso y escenarios

4. Diseño

- 4.1. Casos de uso 37 y 38: Visualizar niveles y visualizar nivel
- 4.2. Caso de uso 39: Añadir nivel
- 4.3. Caso de uso 40: Borrar nivel
- 4.4. Caso de uso 41: Modificar nivel
- 4.5. Caso de uso 42: Dar de baja nivel
- 4.6. Casos de uso 43 y 44: Visualizar colores y visualizar color
- 4.7. Caso de uso 45: Añadir color
- 4.8. Caso de uso 46: Borrar color
- 4.9. Caso de uso 47: Modificar color
- 4.10. Caso de uso 48: Dar de baja color
- 4.11. Casos de uso 49 y 50: Visualizar tarjetas y visualizar tarjeta
- 4.12. Caso de uso 51: Añadir tarjeta
- 4.13. Caso de uso 52: Borrar tarjeta
- 4.14. Caso de uso 53: Modificar tarjeta
- 4.15. Caso de uso 54: Dar de baja tarjeta
- 4.16. Caso de uso 55: Visualizar tipos de acceso
- 4.17. Caso de uso 56: Visualizar tipo de acceso
- 4.18. Caso de uso 57: Añadir tipo de acceso
- 4.19. Caso de uso 58: Borrar tipo de acceso
- 4.20. Caso de uso 59: Modificar tipo de acceso
- 4.21. Casos de uso 60 y 61: Visualizar cargos y visualizar cargo
- 4.22. Caso de uso 62: Añadir cargo
- 4.23. Caso de uso 63: Borrar cargo
- 4.24. Caso de uso 64: Modificar cargo

5. Codificación e integración

6. Pruebas

- 6.1. De unidad
- 6.2. De integración
- 6.3. De sistema



7. Evaluación de la iteración

8. Despliegue



1. Introducción

1.1. Propósito

A lo largo de las siguientes páginas guiaremos al lector en todo el desarrollo de la funcionalidad relativa a tarjetas, colores, niveles, tipos de acceso y cargos. El propósito de este documento es pues servir de material de consulta en caso de que el lector necesite conocer los detalles de cómo será implementada esta parte de la aplicación.

1.2. Audiencia

Nos encontramos ante la misma situación que en las dos iteraciones anteriores. No podemos negar de la gran importancia de este artefacto de cara a poder realizar futuras funciones de mantenimiento de la aplicación, pero hasta que ésta no esté completamente desarrollada será tratado como un documento interno.

1.3. Estructura del documento

Continuamos empleando la estructura con la que dotamos a los artefactos descriptores de las iteraciones de construcción. En primer lugar hacemos un breve paréntesis para evaluar en qué punto del proyecto nos encontramos y hacia donde debemos encaminar nuestros pasos. En la sección 3 planificamos la iteración y analizamos todos los casos de uso que implementaremos en ésta, elaborando para ello todas las descripciones de trazo fino de éstos. Esta sección da paso a la de diseño, donde mostramos todo el diseño realizado de cara a poder llevar a cabo la etapa de codificación, vista después. En el sexto apartado exponemos todas las pruebas realizadas que nos han llevado a certificar que el software desarrollado es correcto y finalizamos este escrito con una evaluación de la iteración y las conclusiones extraídas de la etapa de despliegue.

2. Estado del proyecto

Vistas las cuatro primeras iteraciones de la fase de Construcción, nos toca continuar con la planificación establecida en la fase de Elaboración, tratando de implementar los casos de uso más importantes que nos queden por tratar, es decir, todos aquellos relativos a tarjetas, colores, niveles, tipos de acceso y cargos.



3. Análisis

3.1. Planificación de la iteración

Nos ponemos manos a la obra:

PLAN DE ITERACIÓN C5: TARJETAS, TIPOS DE ACCESO Y CARGOS			
ASPECTO	DESCRIPCIÓN		
Objetivos	Analizar, diseñar, codificar y probar todos los casos de uso relacionados con tarjetas, colores, niveles, tipos de acceso y cargos, es decir, los casos de uso del 37 al 64.		
Artefactos de entrada	<ul style="list-style-type: none"> • <i>Especificación inicial de los casos de uso</i> • <i>Planificación temporal y costes detallados del proyecto</i> • Base de datos, API y aplicación desarrolladas en las iteraciones anteriores. 		
Artefactos modificados o creados	<ul style="list-style-type: none"> • Diagrama de clases de la API inicial. • Clases Java de la API inicial y de la aplicación. • El presente documento. 		
Calendarización detallada	Análisis + Pruebas	08/03/10	<ul style="list-style-type: none"> • Planificación de la iteración y gestión de riesgos. • Análisis de los casos de uso y escenarios. • Planificación de las pruebas. • Inicio de la documentación.
	Diseño + Pruebas	08/03/10	<ul style="list-style-type: none"> • Diseño de opciones arquitectónicas. • Diseño de pruebas.
	Codificación + Pruebas	09/03/10	<ul style="list-style-type: none"> • Codificación e integración de la solución. • Ejecución y evaluación de pruebas.
		10/03/10	<ul style="list-style-type: none"> • Evaluación del prototipo. • Preparación de la entrega ejecutable.
Despliegue	10/03/10	<ul style="list-style-type: none"> • Muestra, prueba y evaluación conjunta con el cliente. • Concreción de los cambios necesarios. • Finalización de toda la documentación. 	

Planificación de la iteración



Nombre de tarea	Duración	Comienzo	Fin	0	0	0	0
Estudio de oportunidad (inception)	7 días	jue 07/01/10	vie 15/01/10				
Elaboración	16 días	lun 18/01/10	lun 08/02/10				
Construcción	41 días	mar 09/02/10	mar 06/04/10				
Iteración C1: La base de datos	4 días	mar 09/02/10	vie 12/02/10				
Iteración C2: API para la base de datos	5 días	lun 15/02/10	vie 19/02/10				
Iteración C3: Accesos	5 días	lun 22/02/10	vie 26/02/10				
Iteración C4: Empresas y personal	5 días	lun 01/03/10	vie 05/03/10				
Iteración C5: Tarjetas, tipos de acceso y cargos	3 días	lun 08/03/10	mié 10/03/10				
Análisis	0,5 días	lun 08/03/10	lun 08/03/10				
Diseño	0,5 días	lun 08/03/10	lun 08/03/10				
Implementación	1,5 días	mar 09/03/10	mié 10/03/10				
Pruebas	3 días	lun 08/03/10	mié 10/03/10				
Despliegue	0,5 días	mié 10/03/10	mié 10/03/10				
Iteración C6: Zonas, llaves y llaveros	5 días	jue 11/03/10	mié 17/03/10				
Iteración C7: Préstamos	2 días	jue 18/03/10	vie 19/03/10				
Iteración C8: Cuentas	2 días	lun 22/03/10	mar 23/03/10				
Iteración C9: Partes de seguridad	3 días	mié 24/03/10	vie 26/03/10				
Iteración C10: Notas	3 días	lun 29/03/10	mié 31/03/10				
Iteración C11: Base de datos	2 días	jue 01/04/10	vie 02/04/10				
Iteración C12: Diseño, personalización y configuración	2 días	lun 05/04/10	mar 06/04/10				
Capacidad operacional	0 días	mar 06/04/10	mar 06/04/10				
Transición	8 días	mié 07/04/10	vie 16/04/10				

Recursos	Temporales		3 días
	Humanos		1 persona
	Software	Sistema operativo	Microsoft Windows 7
		Ofimática	Microsoft Word 2010 Adobe Acrobat Professional 8
		Diseño	Visual Paradigm 4.2
		Codificación	NetBeans 6.8
Hardware		1 PC	

Casos de uso y escenarios implementados	37 - Visualizar niveles	51 - Añadir tarjeta
	38 - Visualizar nivel	52 - Borrar tarjeta
	39 - Añadir nivel	53 - Modificar tarjeta
	40 - Borrar nivel	54 - Dar de baja tarjeta
	41 - Modificar nivel	55 - Visualizar tipos de acceso
	42 - Dar de baja nivel	56 - Visualizar tipo de acceso
	43 - Visualizar colores	57 - Añadir tipo de acceso
	44 - Visualizar color	58 - Borrar tipo de acceso
	45 - Añadir color	59 - Modificar tipo de acceso
	46 - Borrar color	60 - Visualizar cargos
	47 - Modificar color	61 - Visualizar cargo
	48 - Dar de baja color	62 - Añadir cargo
	49 - Visualizar tarjetas	63 - Borrar cargo
	50 - Visualizar tarjeta	64 - Modificar cargo

Planificación de la iteración



3.2. Análisis de los casos de uso y escenarios

Tras seleccionar y enumerar los casos de uso a implementar, procede realizar las descripciones de trazo fino que permitan analizar y comprender más profundamente todos y cada uno de los casos:

Caso de uso 37: Visualizar niveles		
Descripción	Todos los niveles deben poder ser visualizados de forma directa, de un solo vistazo.	
Actores principal	Administrador.	
Pre-condiciones	El administrador se encuentra logueado en el sistema y desea consultar los niveles existentes en el sistema.	
Flujos	Básico	1. El usuario accede al apartado de 'Niveles'. 2. El sistema muestra todos los niveles existentes, además de otros datos.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario conoce todos los niveles existentes.	

Caso de uso 37: Visualizar niveles

Caso de uso 38: Visualizar nivel		
Descripción	El caso de uso anterior nos daba la posibilidad de visualizar los principales datos de todos los niveles existentes de un vistazo. En esta ocasión posibilitaremos que el usuario pueda observar todos los detalles de un nivel seleccionado.	
Actores principal	Administrador.	
Pre-condiciones	El administrador desea consultar todos los detalles de uno de los niveles que observa en la tabla de niveles.	
Flujos	Básico	1. El usuario pulsa el nivel del que desea consultar sus datos. 2. El sistema rescata todos los datos que posee del nivel y se los muestra al usuario.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario disfruta de todos los detalles que buscaba.	

Caso de uso 38: Visualizar nivel



Caso de uso 39: Añadir nivel	
Descripción	Si surge un nuevo nivel en el centro, debemos proporcionar la posibilidad de poder registrarlo en el sistema.
Actores principal	Administrador.
Pre-condiciones	El administrador desea almacenar un nuevo nivel en el sistema.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario indica al sistema que quiere registrar un nuevo nivel. 2. El sistema proporciona al usuario un formulario con el que dar de alta el nuevo nivel. 3. El usuario rellena el formulario, pide al sistema que realice el alta y aprueba el aviso previo. (*) 4. Si todo es correcto se lleva a cabo el alta del nuevo nivel. (*)
	<p>Alternativos (*)</p> <p>El usuario no da su conformidad a que se lleve a cabo el alta del nivel:</p> <ul style="list-style-type: none"> • El sistema aborta el proceso y finaliza el caso de uso. <p>Los datos introducidos ya existen o son incorrectos:</p> <ul style="list-style-type: none"> • El sistema avisa al usuario de esta situación y retorna al punto 2, sin haber modificado los campos redactados.
Post-condiciones	El administrador consigue registrar el nuevo nivel en el sistema.

Caso de uso 39: Añadir nivel

Caso de uso 40: Borrar nivel	
Descripción	Si existe un nivel que no debiera constar en la base de datos, el usuario podrá seleccionarlo y pedir al sistema que lo elimine.
Actores principal	Administrador.
Pre-condiciones	El administrador se encuentra logueado en el sistema y en la pestaña de niveles observa que existe alguno que no debiera, con que se dispone a eliminarlo.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario selecciona los niveles que no debieran existir y pide al sistema que los elimine. 2. El sistema pide al usuario que confirme que los niveles seleccionados son los que realmente quiere eliminar. 3. Los niveles seleccionados son eliminados si el usuario da su conformidad. (*)
	<p>Alternativos (*)</p> <p>El usuario no da su conformidad al borrado:</p> <ul style="list-style-type: none"> • El sistema no realiza acción alguna.
Post-condiciones	El administrador ha conseguido eliminar los niveles que no debían constar en el sistema.

Caso de uso 40: Borrar nivel



Caso de uso 41: Modificar nivel	
Descripción	Un nivel puede variar a lo largo de su existencia, con que debe ser posible modificar los valores de los atributos que lo componen.
Actores principal	Administrador.
Pre-condiciones	El usuario desea modificar los datos de un nivel existente.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El administrador selecciona el nivel a modificar y el sistema despliega los datos de éste. 2. Se modifican los datos oportunos y se pide al sistema que lleve a cabo la actualización. 3. El sistema realiza la actualización del nivel en la base de datos si el administrador da su conformidad. (*)
	<p>Alternativos (*)</p> <p>El administrador no da el visto bueno o introduce los datos con un formato no válido:</p> <ul style="list-style-type: none"> • El sistema avisa del error y no realiza modificación alguna.
Post-condiciones	El usuario consigue actualizar el nivel indicado y que conste como deseaba.

Caso de uso 41: Modificar nivel

Caso de uso 42: Dar de baja nivel	
Descripción	Si un nivel deja de existir y se quiere seguir manteniendo la trazabilidad, debe existir una forma de darlo de baja de modo que permanezca inoperativo pero que siga constando en el sistema.
Actores principal	Administrador.
Pre-condiciones	El administrador se encuentra logueado en el sistema, en la tab de niveles y dispuesto a dar de baja a alguno de ellos.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario selecciona los niveles que desea dar de baja y traslada dicha petición al sistema. 2. Éste pide al usuario que confirme que los niveles seleccionados son los que realmente quiere dar de baja. (*) 3. El sistema da de baja de todos los niveles indicados tras recibir la confirmación.
	<p>Alternativos (*)</p> <p>El administrador no da su conformidad:</p> <ul style="list-style-type: none"> • El sistema no realiza ninguna baja.
Post-condiciones	El sistema ha dado de baja a todos los niveles seleccionados por el administrador.

Caso de uso 42: Dar de baja nivel



Caso de uso 43: Visualizar colores		
Descripción	Pasamos de los niveles a los colores. Las condiciones en las que deberán visualizarse todos los colores existentes son análogas a las de los niveles: todos de un vistazo, fácilmente.	
Actores principal	Administrador.	
Pre-condiciones	El usuario desea observar qué colores se encuentran registrados en el sistema.	
Flujos	Básico	1. El administrador accede al apartado de 'Colores'. 2. El sistema muestra todos los colores existentes.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario visualiza todos los colores activos.	

Caso de uso 43: Visualizar colores

Caso de uso 44: Visualizar color		
Descripción	Quizá al usuario no le baste con observar los datos principales de cada color, sino que desee observar todos los datos de uno concreto.	
Actores principal	Administrador.	
Pre-condiciones	El administrador se encuentra en la tab de 'Colores' y desea conocer todos los datos de uno de ellos.	
Flujos	Básico	1. El usuario selecciona un color e insta al sistema a que le muestre todos los detalles del mismo. 2. El sistema rescata todos los datos que posee del color y se los muestra al usuario por pantalla.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario puede consultar los datos que desee.	

Caso de uso 44: Visualizar color



Caso de uso 45: Añadir color	
Descripción	No es lo usual, pero siempre puede darse el caso de que llegue al centro un nuevo juego de tarjetas, con que tenemos que tener la posibilidad de poder añadir nuevos colores para así luego registrar las tarjetas.
Actores principal	Administrador.
Pre-condiciones	El administrador se encuentra logueado y desea introducir un nuevo color.
Flujos	Básico <ol style="list-style-type: none">1. El usuario pide al sistema introducir un nuevo color.2. El sistema le proporciona un formulario con el que registrarlo.3. El usuario rellena todos los campos e indica al sistema que realice el alta.4. El caso de uso termina registrando el sistema el nuevo color en el caso de que el usuario haya dado su visto bueno. (*)
	Alternativos (*) <p>El usuario no da su visto bueno:</p> <ul style="list-style-type: none">• El sistema no realiza ninguna acción y el caso de uso termina.
Post-condiciones	El administrador consigue dar de alta el color que deseaba.

Caso de uso 45: Añadir color

Caso de uso 46: Borrar color	
Descripción	Nos ocupa el caso contrario: eliminar algún color debido a que ya no tiene sentido seguir almacenándolo (perdiendo la consecuente trazabilidad).
Actores principal	Administrador.
Pre-condiciones	El usuario se encuentra en la pestaña de tarjetas y se percata de que existen colores que no debieran, con lo que se dispone a eliminarlos.
Flujos	Básico <ol style="list-style-type: none">1. El administrador selecciona los colores que desea eliminar de la tabla de colores, pidiendo al sistema que los elimine.2. El sistema solicita al usuario que dé su visto bueno para proceder con el borrado.3. El usuario acepta y los colores dejan de existir. (*)
	Alternativos (*) <p>El usuario no da su visto bueno:</p> <ul style="list-style-type: none">• El sistema no borra ningún color y el caso de uso finaliza.
Post-condiciones	El usuario consigue borrar del sistema los colores que se proponía.

Caso de uso 46: Borrar color



Caso de uso 47: Modificar color					
Descripción	Uno de los casos de uso detallados en la <i>Especificación inicial de los casos de uso</i> era que todo color pudiera ser modificado. Esto se debe a que puede darse el caso de que un color (juego de tarjetas) vea modificado su nivel, proporcionando así un menor o mayor grado de acceso, más que el color cambie de coloración.				
Actores principal	Administrador.				
Pre-condiciones	El usuario necesita modificar los datos de un color, con que abre la aplicación y se dirige a la pestaña de tarjetas.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> <ol style="list-style-type: none"> 1. El administrador selecciona el color que desea modificar y pide al sistema poder modificarlo. 2. Éste le proporciona todos los datos del color, editando el usuario todo lo que estime oportuno. 3. El administrador traslada al sistema su intención de que haga permanentes los cambios y ratifica el aviso enviado por el sistema. (*) </td> </tr> <tr> <td>Alternativos (*)</td> <td> <p>El usuario no da su visto bueno:</p> <ul style="list-style-type: none"> • El sistema no borra ningún color y el caso de uso concluye. </td> </tr> </table>	Básico	<ol style="list-style-type: none"> 1. El administrador selecciona el color que desea modificar y pide al sistema poder modificarlo. 2. Éste le proporciona todos los datos del color, editando el usuario todo lo que estime oportuno. 3. El administrador traslada al sistema su intención de que haga permanentes los cambios y ratifica el aviso enviado por el sistema. (*) 	Alternativos (*)	<p>El usuario no da su visto bueno:</p> <ul style="list-style-type: none"> • El sistema no borra ningún color y el caso de uso concluye.
	Básico	<ol style="list-style-type: none"> 1. El administrador selecciona el color que desea modificar y pide al sistema poder modificarlo. 2. Éste le proporciona todos los datos del color, editando el usuario todo lo que estime oportuno. 3. El administrador traslada al sistema su intención de que haga permanentes los cambios y ratifica el aviso enviado por el sistema. (*) 			
Alternativos (*)	<p>El usuario no da su visto bueno:</p> <ul style="list-style-type: none"> • El sistema no borra ningún color y el caso de uso concluye. 				
Post-condiciones	El sistema lleva a cabo la modificación indicada y el usuario puede dar por cumplido su objetivo.				

Caso de uso 47: Modificar color

Caso de uso 48: Dar de baja color					
Descripción	Es el turno de desarrollar las bajas de colores.				
Actores principal	Administrador.				
Pre-condiciones	El administrador desea dar de baja algún color activo en el sistema.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> <ol style="list-style-type: none"> 1. El usuario selecciona el conjunto de colores que desea dar de baja, permaneciendo almacenados en este pero dejando de forma parte del día a día en el centro. Clica en el botón 'Dar de baja'. 2. El sistema le envía un mensaje avisándole de que va a proceder con la baja de los colores indicados, a lo que el usuario responde afirmativamente (*). </td> </tr> <tr> <td>Alternativos (*)</td> <td> <p>El usuario no responde afirmativamente:</p> <ul style="list-style-type: none"> • El sistema no procede con la baja de ningún color y vuelve al punto de partida. </td> </tr> </table>	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona el conjunto de colores que desea dar de baja, permaneciendo almacenados en este pero dejando de forma parte del día a día en el centro. Clica en el botón 'Dar de baja'. 2. El sistema le envía un mensaje avisándole de que va a proceder con la baja de los colores indicados, a lo que el usuario responde afirmativamente (*). 	Alternativos (*)	<p>El usuario no responde afirmativamente:</p> <ul style="list-style-type: none"> • El sistema no procede con la baja de ningún color y vuelve al punto de partida.
	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona el conjunto de colores que desea dar de baja, permaneciendo almacenados en este pero dejando de forma parte del día a día en el centro. Clica en el botón 'Dar de baja'. 2. El sistema le envía un mensaje avisándole de que va a proceder con la baja de los colores indicados, a lo que el usuario responde afirmativamente (*). 			
Alternativos (*)	<p>El usuario no responde afirmativamente:</p> <ul style="list-style-type: none"> • El sistema no procede con la baja de ningún color y vuelve al punto de partida. 				
Post-condiciones	El administrador logra dar de baja los colores que desea.				

Caso de uso 48: Dar de baja color



Caso de uso 49: Visualizar tarjetas		
Descripción	La visualización de las tarjetas sigue la misma lógica que los niveles y los colores. Procedamos con ella.	
Actores principal	Administrador.	
Pre-condiciones	El administrador se encuentra logueado en el sistema y desea visualizar los juegos de tarjetas de los que disponen los vigilantes para llevar a cabo el proceso de accesos.	
Flujos	Básico	<ol style="list-style-type: none">1. El administrador accede al apartado 'Tarjetas'.2. El sistema es el encargado de mostrarle todas las tarjetas existentes en el sistema y que están activas, sin fecha de baja.
	Alternativos (*)	Ninguno.
Post-condiciones	El administrador observa todo el rango de tarjetas existentes.	

Caso de uso 49: Visualizar tarjetas

Caso de uso 50: Visualizar tarjeta		
Descripción	No sólo debe bastar con ver los datos más representativos de cada tarjeta, sino que el administrador debe poder consultar todos los datos de los que esté compuesta.	
Actores principal	Administrador.	
Pre-condiciones	El usuario se encuentra en el tab de tarjetas y se dispone a obtener los datos de una concreta.	
Flujos	Básico	<ol style="list-style-type: none">1. El administrador clics sobre la tarjeta que desea consultar y pide al sistema que le proporcione todos los datos que disponga de ésta.2. El sistema recupera todos los datos posibles y se los muestra al usuario.
	Alternativos (*)	Ninguno.
Post-condiciones	El administrador obtiene por pantalla todos los datos de la tarjeta que deseaba consultar.	

Caso de uso 50: Visualizar tarjeta



Caso de uso 51: Añadir tarjeta	
Descripción	Este hecho sí que puede llegar a ser más habitual, debido a que el constante préstamo de tarjetas puede derivar en la pérdida de éstas, y por tanto en que sea necesario ir introduciendo nuevas.
Actores principal	Administrador.
Pre-condiciones	El administrador dispone de una nueva tarjeta a dar de alta en el sistema para que pueda utilizarse.
Flujos	Básico <ol style="list-style-type: none"> 1. El usuario clica sobre el botón de 'Dar de alta tarjeta', lo que hace que el sistema muestre un formulario con el que poder llevar a cabo la tarea. 2. Tras disponer del formulario completado y la petición y aprobación del cliente, el sistema puede proceder con el alta de la tarjeta. (*)
	Alternativos (*) <p>El usuario no da su conformidad con el alta de la tarjeta:</p> <ul style="list-style-type: none"> • Se interrumpe el proceso y da por concluido el caso de uso. <p>Los datos introducidos no son válidos (el código de tarjeta no está comprendido entre 0 y 999):</p> <ul style="list-style-type: none"> • El sistema avisa al usuario del error ocurrido y se retorna al punto de partida.
Post-condiciones	El administrador ha conseguido dar de alta la tarjeta que deseaba.

Caso de uso 51: Añadir tarjeta

Caso de uso 52: Borrar tarjeta	
Descripción	Es cierto que lo normal será que el administrador dé de baja las tarjetas y no las borre, pero puede llegar a darse el primer caso.
Actores principal	Administrador.
Pre-condiciones	El usuario desea eliminar alguna de las tarjetas registradas en el sistema.
Flujos	Básico <ol style="list-style-type: none"> 1. El administrador selecciona de la tabla de tarjetas aquellas que desea eliminar y pide al sistema que las elimine. 2. Éste procede a su borrado, pero previamente debe de haber recibo una confirmación por parte del administrador. (*)
	Alternativos (*) <p>El administrador opta por abortar la eliminación:</p> <ul style="list-style-type: none"> • El sistema no borra ningún dato y el caso de uso finaliza sin más consecuencias.
Post-condiciones	El sistema no dispone de las tarjetas eliminadas por el usuario.

Caso de uso 52: Borrar tarjeta



Caso de uso 53: Modificar tarjeta	
Descripción	No es lo normal, pero también debemos tratar el caso de que una tarjeta pueda cambiar de numeración o color.
Actores principal	Administrador.
Pre-condiciones	El usuario quiere modificar la numeración y/o color de una de las tarjetas.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El administrador selecciona la tarjeta que desea modificar y traslada al sistema su petición. 2. El sistema le muestra todos los datos de ésta; datos que edita el administrador. 3. Se produce un paso previo de confirmación y el sistema procede con la actualización de la tarjeta. (*)
	<p>Alternativos (*)</p> <p>El administrador ha introducido datos con el formato equivocado:</p> <ul style="list-style-type: none"> • El sistema avisa del error y no actualiza ningún registro. <p>El administrador niega que se realice la modificación:</p> <ul style="list-style-type: none"> • El sistema no lleva a cabo ninguna acción y el caso de uso finaliza sin producir cambios.
Post-condiciones	La tarjeta que el usuario seleccionó y modificó queda alterada en la base de datos.

Caso de uso 53: Modificar tarjeta

Caso de uso 54: Dar de baja tarjeta	
Descripción	Ésta utilidad será bastante más utilizada que sus dos predecesoras: todas y cada una de las veces que se extravíe una tarjeta.
Actores principal	Administrador.
Pre-condiciones	El administrador desea dar de baja una tarjeta ya que ésta se ha extraviado.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario selecciona un rango de tarjetas y pide al sistema que las dé de baja. 2. El sistema le pregunta al usuario si de verdad quiere dar de baja las tarjetas seleccionadas. 3. El usuario responde que sí. (*) 4. El sistema procede con la baja de todas las tarjetas proporcionadas.
	<p>Alternativos (*)</p> <p>El usuario responde que no:</p> <ul style="list-style-type: none"> • El sistema no modifica ningún dato y da por concluido el caso de uso.
Post-condiciones	El administrador ha conseguido dar de baja las tarjetas seleccionadas.

Caso de uso 54: Dar de baja tarjeta



Caso de uso 55: Visualizar tipos de acceso		
Descripción	El cliente requiere que puedan visualizarse los principales datos de todos los tipos de acceso y todos los datos que conforman cada tipo (caso de uso siguiente).	
Actores principal	Administrador.	
Pre-condiciones	El usuario desea visualizar los principales datos de todos los tipos de acceso existentes.	
Flujos	Básico	<ol style="list-style-type: none">1. El usuario se dirige al tab de tarjetas.2. El sistema muestra al usuario los datos que busca en una tabla.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario disfruta de los principales datos de todos los tipos de acceso residentes en la base de datos.	

Caso de uso 55: Visualizar tipos de acceso

Caso de uso 56: Visualizar tipo de acceso		
Descripción	Completamos el caso de uso anterior posibilitando al administrador que pueda visualizar todos los datos que componen un tipo de acceso.	
Actores principal	Administrador.	
Pre-condiciones	El administrador se encuentra logueado en el sistema y acude a la pestaña de tipos de acceso en búsqueda de todos los datos de un tipo de acceso.	
Flujos	Básico	<ol style="list-style-type: none">1. El administrador selecciona un tipo de acceso y pide al sistema que le ofrezca todos los datos de los que disponga.2. El sistema acude a la base de datos, rescata todos los datos del tipo de acceso y se los hace llegar al usuario a través de un formulario.
	Alternativos (*)	Ninguno.
Post-condiciones	El administrador obtiene todos los datos que estaba buscando.	

Caso de uso 56: Visualizar tipo de acceso



Caso de uso 57: Añadir tipo de acceso	
Descripción	No podemos limitar los tipos de acceso a que sólo existan los iniciales, con que a partir de ahora realizaremos los casos de uso que permitan su manejo.
Actores principal	Administrador.
Pre-condiciones	Surge un nuevo tipo de acceso, con lo que el administrador arranca la aplicación y se dirige a la pestaña de tarjetas.
Flujos	Básico <ol style="list-style-type: none">1. El usuario pide al sistema introducir un nuevo tipo de acceso.2. El sistema le proporciona un formulario mediante el cual obtiene todos los datos de éste. Acto seguido envía al usuario una petición de confirmación de alta de tipo de acceso.3. El usuario responde afirmativamente y el sistema se encarga de registrar el nuevo tipo de acceso. (*)
	Alternativos (*) <p>El usuario introduce algún dato con un formato erróneo:</p> <ul style="list-style-type: none">• El sistema avisa del error acontecido y no realiza ninguna inserción. <p>El usuario responde negativamente ante la petición de confirmación:</p> <ul style="list-style-type: none">• El sistema no realiza ninguna acción más y da por concluido en caso de uso.
Post-condiciones	El administrador consigue que el nuevo tipo de acceso pase a formar parte del sistema.

Caso de uso 57: Añadir tipo de acceso



Caso de uso 58: Borrar tipo de acceso		
Descripción	Borrar un tipo de acceso es tan fácil como puede ser borrar un nivel o una tarjeta.	
Actores principal	Administrador.	
Pre-condiciones	El usuario no desea seguir almacenando alguno de los tipos de acceso existentes en el sistema.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El administrador selecciona los tipos de acceso que quiere eliminar e indica al sistema que proceda. 2. El sistema pide al usuario que confirme que los tipos de acceso seleccionados son realmente los que desea eliminar. 3. El administrador responde afirmativamente ante la advertencia. (*) 4. El sistema procede con el borrado de los tipos de acceso indicados.
	Alternativos (*)	<p>El administrador responde con un no:</p> <ul style="list-style-type: none"> • El sistema aborta cualquier tipo de operación y el caso de uso finaliza.
Post-condiciones	El usuario consigue lo que se proponía: eliminar los tipos de acceso no deseados.	

Caso de uso 58: Borrar tipo de acceso

Caso de uso 59: Modificar tipo de acceso		
Descripción	Es bastante probable que los tipos de acceso vayan variando a lo largo de la vida del centro para adaptarse a las nuevas necesidades que surjan.	
Actores principal	Administrador.	
Pre-condiciones	El administrador quiere cambiar alguna de las características de un tipo de acceso.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona el tipo de acceso a modificar y transmite al sistema su intención. 2. El sistema le muestra todos los datos del tipo de acceso. 3. El usuario edita lo que estima oportuno y solicita grabar los cambios. 4. El sistema indica al usuario que va a modificar el acceso y que necesita previamente una autorización. 5. El usuario la da y el sistema procede a actualizar el tipo de acceso. (*)
	Alternativos (*)	<p>El usuario no da su aprobación:</p> <ul style="list-style-type: none"> • El sistema no realiza ninguna acción y el caso de uso finaliza.
Post-condiciones	El tipo de acceso pasa a tener el estado deseado.	

Caso de uso 59: Modificar tipo de acceso



Caso de uso 60: Visualizar cargos		
Descripción	Llegamos al último conjunto de elementos que trataremos en esta pestaña: los cargos.	
Actores principal	Administrador.	
Pre-condiciones	El usuario desea poder consultar todos los cargos existentes en el sistema.	
Flujos	Básico	<ol style="list-style-type: none">1. El usuario pide al sistema poder visualizar todos los cargos existentes.2. El sistema accede a la base de datos y obtiene los datos requeridos, mostrándoselos al usuario.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario visualiza todos los datos que buscaba.	

Caso de uso 60: Visualizar cargos

Caso de uso 61: Visualizar cargo		
Descripción	Es hora de permitir al usuario que visualice todos los datos del cargo que desee.	
Actores principal	Administrador.	
Pre-condiciones	El administrador quiere consultar todos los datos de un cargo. Inicia la aplicación y se sitúa en la pestaña de cargos.	
Flujos	Básico	<ol style="list-style-type: none">1. El administrador selecciona un cargo y pide al sistema que le muestre todos los datos que posea de éste.2. El sistema acude a la base de datos para rescatarlos y se los proporciona al administrador.
	Alternativos (*)	Ninguno.
Post-condiciones	El administrador visualiza todos los datos del cargo.	

Caso de uso 61: Visualizar cargo



Caso de uso 62: Añadir cargo	
Descripción	Si Informática Navarra S.A. decide guardar contactos que desempeñen un nuevo cargo, previamente deberá ser dado de alta éste en la base de datos.
Actores principal	Administrador.
Pre-condiciones	El administrador necesita añadir un nuevo cargo a los existentes. Arranca la aplicación, loguea y se dirige al tab de cargos.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario traslada al sistema su intención de dar de alta un nuevo cargo. 2. El sistema le proporciona un formulario que debe rellenar. 3. Tras completarlo, el usuario le pide al sistema que registre el nuevo cargo. 4. El sistema envía al usuario un mensaje que debe aceptar para proceder con el alta. 5. El usuario acepta el mensaje y el cargo pasa a formar parte de la base de datos. (*)
	<p>Alternativos (*)</p> <p>El usuario introduce datos que no poseen el formato adecuado:</p> <ul style="list-style-type: none"> • El sistema avisa del error y no realiza ninguna inserción. <p>El usuario no acepta el mensaje de confirmación:</p> <ul style="list-style-type: none"> • El sistema finaliza el caso de uso sin realizar ninguna acción.
Post-condiciones	El administrador dispone del nuevo cargo registrado en el sistema.

Caso de uso 62: Añadir cargo



Caso de uso 63: Borrar cargo		
Descripción	Puede llegar el momento en el que el administrador opte por prescindir de todos los contactos que desempeñen algún tipo de cargo.	
Actores principal	Administrador.	
Pre-condiciones	El usuario se encuentra en el tab de cargos dispuesto a eliminar uno o varios de ellos.	
Flujos	Básico	<ol style="list-style-type: none">1. El administrador selecciona los cargos que desea borrar e indica al sistema que proceda.2. Éste le avisa que va a eliminar los cargos seleccionados y le pregunta si de está seguro de querer realizar tal acción.3. El administrador le contesta afirmativamente y todos los cargos seleccionados dejan de existir. (*)
	Alternativos (*)	<p>El administrador decide abortar el borrado respondiendo negativamente al aviso:</p> <ul style="list-style-type: none">• El sistema también aborta y el caso de uso no llega a completarse.
Post-condiciones	El administrador consigue eliminar los cargos que pretendía.	

Caso de uso 63: Borrar cargo



Caso de uso 64: Modificar cargo	
Descripción	Este caso de uso resulta menos frecuente, pero no por ello podemos olvidarlo.
Actores principal	Administrador.
Pre-condiciones	El administrador desea modificar un cargo por constar éste con datos erróneos.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario se sitúa en la pestaña de cargos y selecciona el que desea modificar. Pulsa el botón de 'Modificar cargo' y el sistema le proporciona todos los datos actuales del cargo. 2. Una vez que el usuario ha modificado lo que estima oportuno, pide al sistema que actualice el cargo. 3. El sistema genera un aviso que el usuario debe aceptar como paso previo a la modificación. 4. El usuario da el visto bueno y la actualización tiene lugar. (*)
	<p>Alternativos (*)</p> <p>El usuario introduce algún nuevo dato con un formato indeseado:</p> <ul style="list-style-type: none"> • El sistema avisa del error y no realiza ninguna acción, volviendo al punto de partida. <p>El usuario responde negativamente ante la petición de confirmación:</p> <ul style="list-style-type: none"> • El sistema no envía ningún mensaje ni actualiza ningún registro, pero sí que da por concluido el caso de uso.
Post-condiciones	El administrador consigue modificar el cargo que se proponía con los datos proporcionados al sistema.

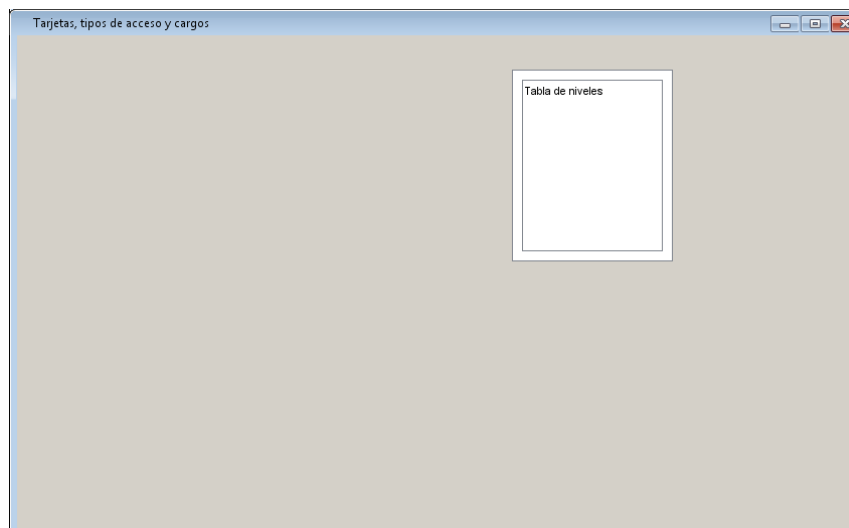
Caso de uso 64: Modificar cargo



4. Diseño

4.1. Casos de uso 37 y 38: Visualizar niveles y visualizar nivel

Destinamos a los siguientes 25 casos de uso una pestaña dentro de la aplicación, en la que comenzaremos colocando un JTable donde visualizar todos los niveles existentes:



Interfaz gráfica propuesta

La estrategia a seguir es sencilla:

1. Crear un objeto de tipo Nivel con el que realizar la búsqueda.
2. Rellenar la tabla con los niveles obtenidos.

Necesitaremos un método con el que realizar la búsqueda...

MÉTODO	FUNCIONALIDAD
<i>obtenerNiveles</i>	Método que rescata de la base de datos todos los niveles existentes.

Método a implementar y a añadir a la API



... y seguir los siguientes pasos:

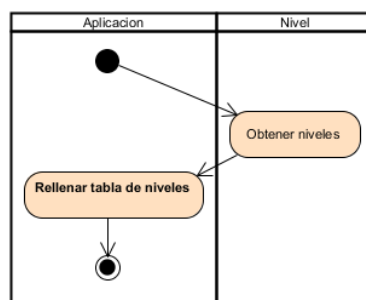


Diagrama de actividades

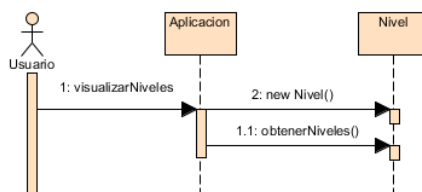
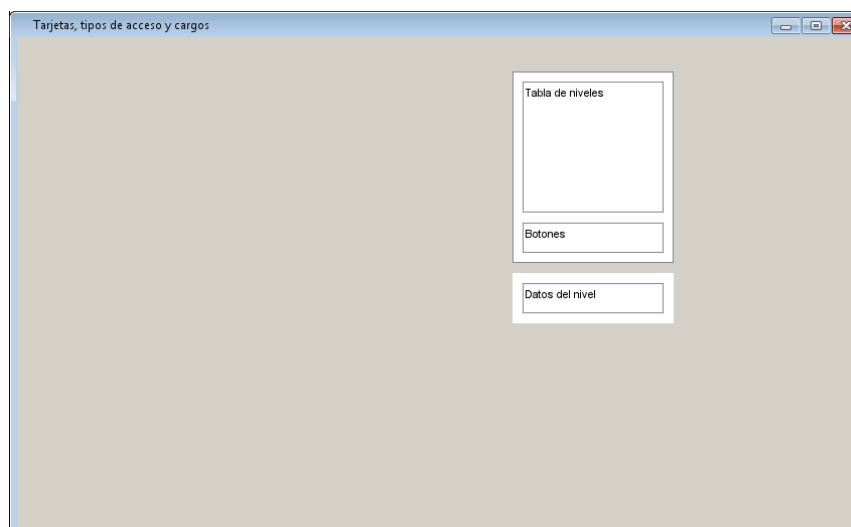


Diagrama de secuencia

Los niveles constan de dos campos de importancia, representativos, con que debería bastar al con mostrarlos al usuario en la misma tabla y evitar que deba realizar operaciones adicionales.

4.2. Caso de uso 39: Añadir nivel

Añadimos a la tabla de niveles existente un panel con el que poder mostrar el formulario de alta de nivel y otro con los botones de acción:



Interfaz gráfica propuesta



Se indicará el nivel por defecto con el siguiente método:

MÉTODO	FUNCIONALIDAD
<i>nivelPorDefecto</i>	Método que devuelve el número de nivel mínimo no creado.

Método a implementar y a añadir a la API

Tras rellenar el usuario el formulario y pedir el alta, tendremos que:

1. Crear un objeto de tipo Nivel con los parámetros introducidos.
2. Insertar el nuevo nivel.
3. Refrescar la tabla de niveles.

Estos son los diagramas de actividades y secuencia que describen el caso de uso:

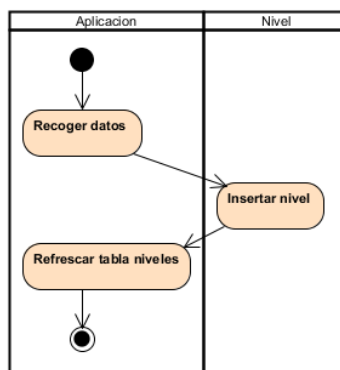


Diagrama de actividades

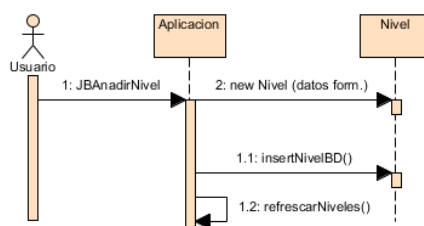


Diagrama de secuencia



4.3. Caso de uso 40: Borrar nivel

Modificamos el panel de botones añadiendo uno que permita el borrado:

1. Obtener los niveles seleccionados.
2. Recorrer los niveles seleccionados eliminándolos uno a uno.
3. Refrescar todas las tablas del tab y la futura de préstamos pendientes (comentamos esta línea de código que activaremos posteriormente).

Para obtener el nivel a través de su código debemos implementar el siguiente constructor:

MÉTODO	FUNCIONALIDAD
<i>Nivel</i>	Constructor que crea un Nivel a partir del código proporcionado.

Método a implementar y a añadir a la API

El flujo de actividades sería...

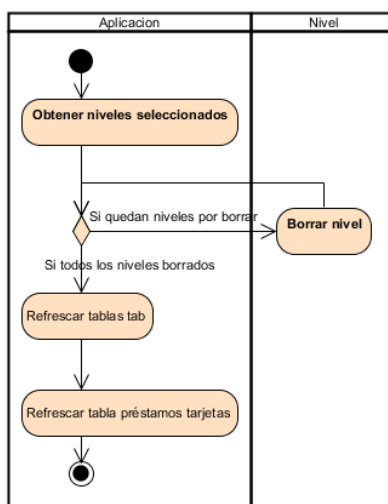


Diagrama de actividades

... y el de mensajes:

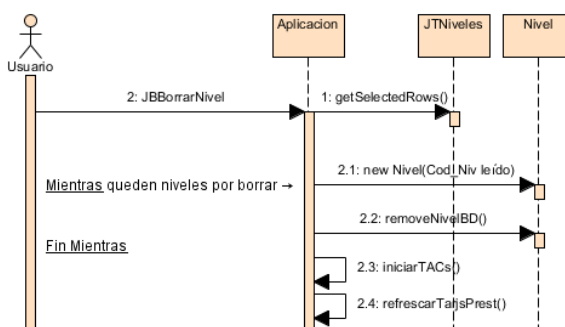


Diagrama de secuencia



4.4. Caso de uso 41: Modificar nivel

Añadimos el botón de modificación y proponemos la siguiente secuencia de actividades:

1. Crear el nuevo nivel con los datos del formulario.
2. Recuperar el viejo nivel de la base de datos.
3. Actualizar el nivel viejo con el nuevo.
4. Refrescar las tablas de colores y niveles.

Detallemos más en profundidad todo este proceso:

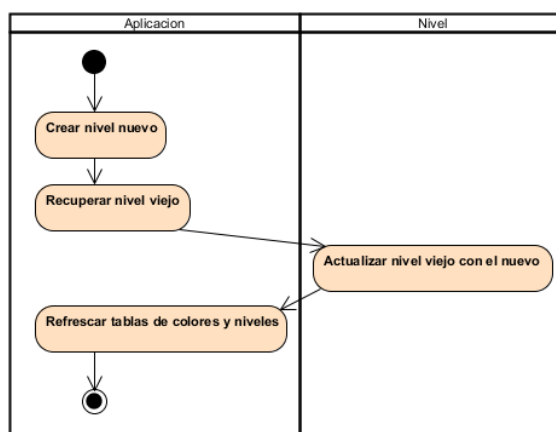


Diagrama de actividades

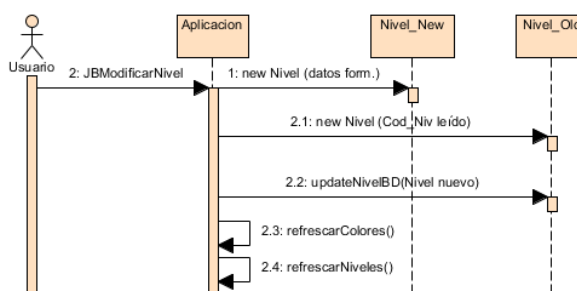


Diagrama de secuencia



4.5. Caso de uso 42: Dar de baja nivel

Culminamos el panel de botones de niveles añadiendo un botón que sirva para realizar las bajas de éstos.

Diseñamos la siguiente estrategia:

1. Obtener los niveles seleccionados.
2. Recorrer los niveles seleccionados dándolos de baja uno a uno.
3. Refrescamos las tablas del tab y la de préstamos de tarjetas.

En esta ocasión no es necesario añadir ningún método a la API para conseguir implementar en un futuro el siguiente proceso:

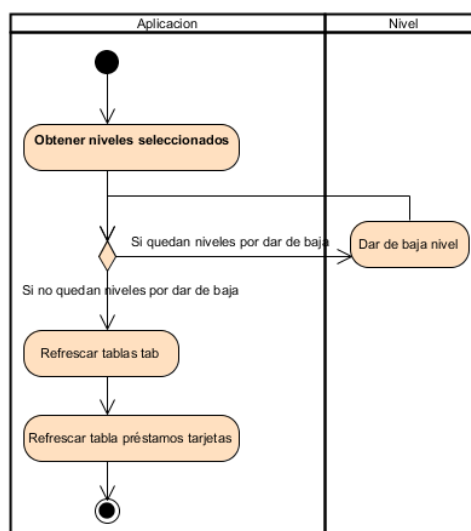


Diagrama de actividades

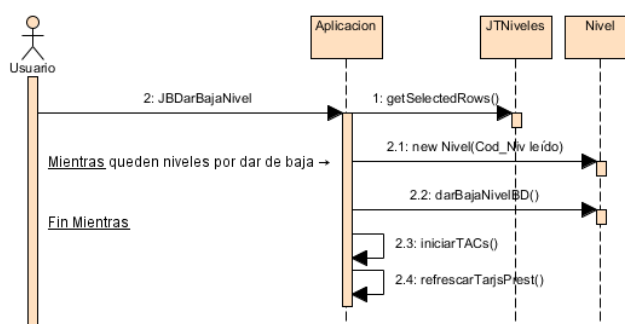
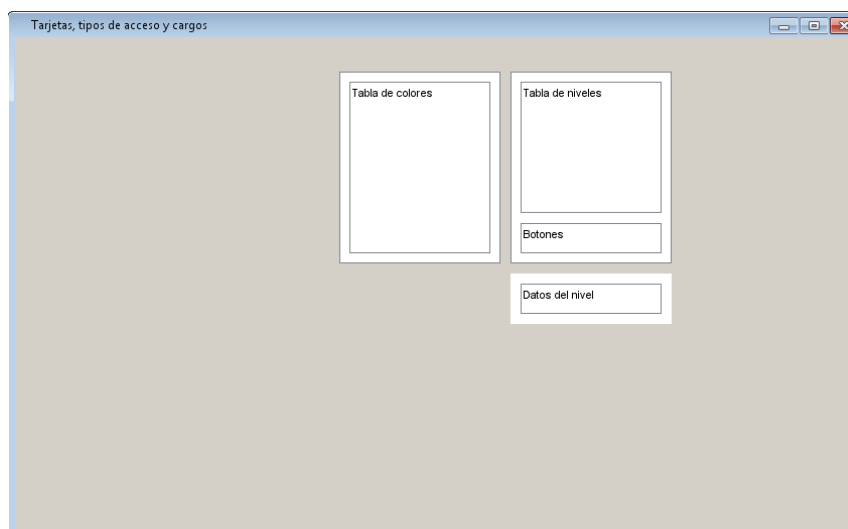


Diagrama de secuencia



4.6. Casos de uso 43 y 44: Visualizar colores y visualizar color

A la interfaz desarrollada hasta el momento añadimos una tabla para visualizar colores similar a la utilizada con los niveles:



Interfaz gráfica propuesta

Como los datos a almacenar acerca de los colores son sólo el código y nivel que proporcionan, optamos por implementar ambos casos de uso mostrando en la tabla ambos atributos, pudiendo ser relacionado el segundo con su descripción a través de la tabla de niveles desarrollada con anterioridad, y que queda a la derecha.

Los pasos a seguir son:

1. Crear un objeto de tipo Color con el que realizar la búsqueda.
2. Rellenar la tabla con los datos de los niveles obtenidos.
3. Pintar la tabla con los colores obtenidos.

En este caso sí que serán necesarios nuevos métodos...

MÉTODO	FUNCIONALIDAD
<i>obtenerCeldasColores</i>	Método que rescata de la base de datos todos los colores existentes.
<i>pintarTabla</i>	Método que pinta las celdas de una tabla con los colores proporcionados.

Métodos a implementar y a añadir a la API



... y optaremos por la siguiente estrategia:

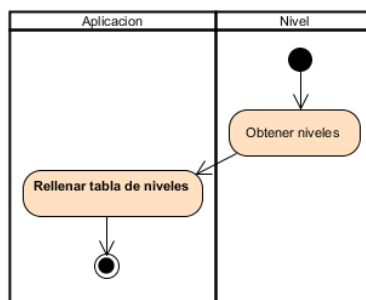


Diagrama de actividades

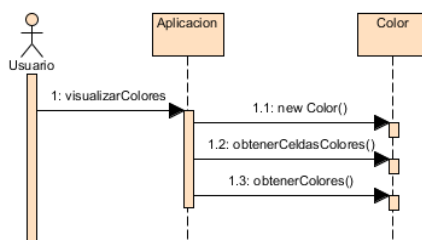
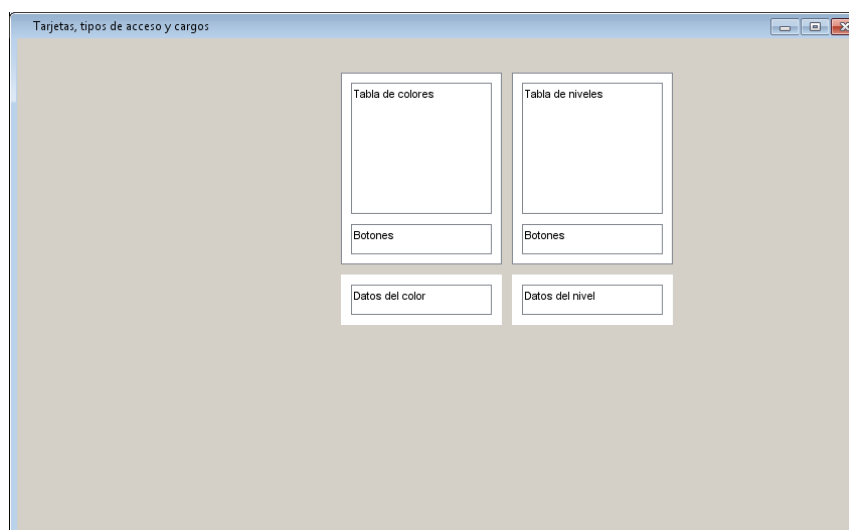


Diagrama de secuencia

4.7. Caso de uso 45: Añadir color

Al igual que ocurría con los niveles, necesitamos añadir un panel de datos y otro de botones:



Interfaz gráfica propuesta



Añadimos los siguientes métodos a la API, que nos permitirán dar al usuario por defecto el menor color de menor nivel existente. Por supuesto, estos datos deberán ser variados en la adición del nuevo nivel (mínimo el color):

MÉTODO	FUNCIONALIDAD
<i>colorPorDefecto</i>	Método que devuelve el color de menor numeración correspondiente al menor nivel existente.
<i>nivelMinimo</i>	Método que devuelve el número de nivel mínimo existente.

Métodos a implementar y a añadir a la API

Procederemos de la siguiente forma:

1. Crear un objeto de tipo Color con los datos proporcionados por el usuario.
2. Insertar el nuevo color.
3. Refrescar la tabla de colores.

Todo queda reflejado en los estos diagramas:

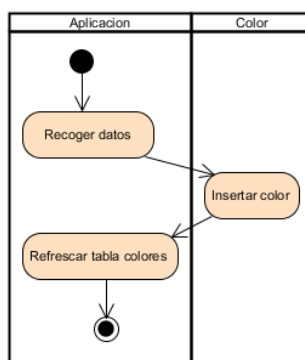


Diagrama de actividades

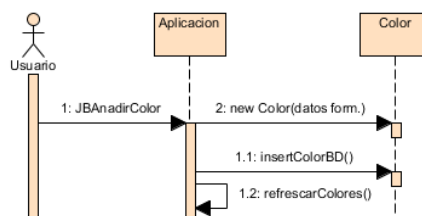


Diagrama de secuencia



4.8. Caso de uso 46: Borrar color

Añadimos el botón de borrado al JPanel de botones de colores y nos proponemos diseñar la siguiente secuencia de actividades, análoga a la de niveles:

1. Obtener los colores seleccionados.
2. Recorrer los colores seleccionados borrándolos uno a uno.
3. Refrescar todas las tablas del tab y la futura de préstamos pendientes.

Añadimos este constructor a la API:

MÉTODO	FUNCIONALIDAD
<i>Color</i>	Constructor que crea un objeto de tipo Color a partir del código proporcionado.

Método a implementar y a añadir a la API

Los diagramas a seguir serían:

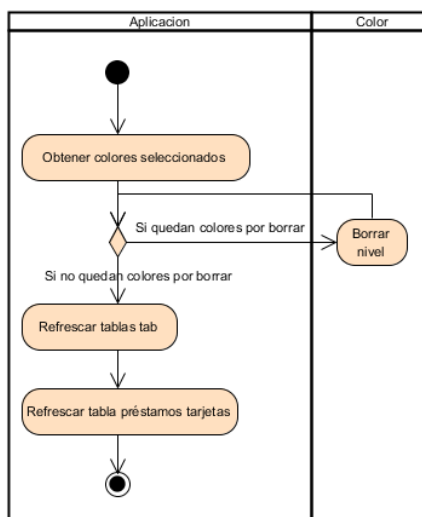


Diagrama de actividades

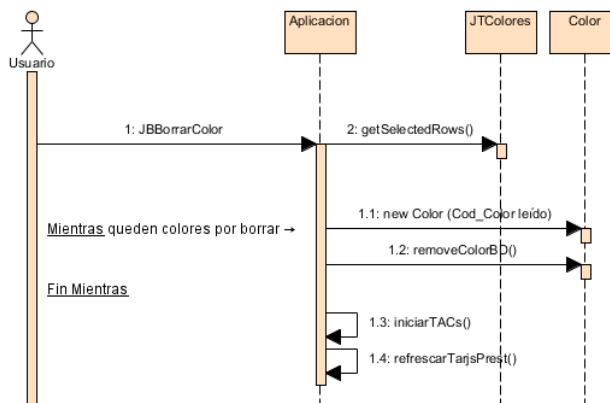


Diagrama de secuencia



4.9. Caso de uso 47: Modificar color

Caso de uso homólogo al del nivel. Basta con añadir un botón que permita modificar el color seleccionado y seguir la misma línea marcada con los colores, que resumimos en:

1. Crear el nuevo color con los datos del formulario.
2. Recuperar el viejo color de la base de datos.
3. Actualizar el color viejo con el nuevo.
4. Refrescar las tablas del tab y la de tarjetas en préstamo.

Añadimos un nuevo constructor a la API, en este caso en la clase Color:

MÉTODO	FUNCIONALIDAD
<i>Color</i>	Constructor que crea un objeto de tipo Color teniendo como parámetro de entrada su código. En caso de existir varios con el mismo código proporciona el más actual.

Método a implementar y a añadir a la API

A continuación mostramos los diagramas que guiarán la implementación de este caso de uso:

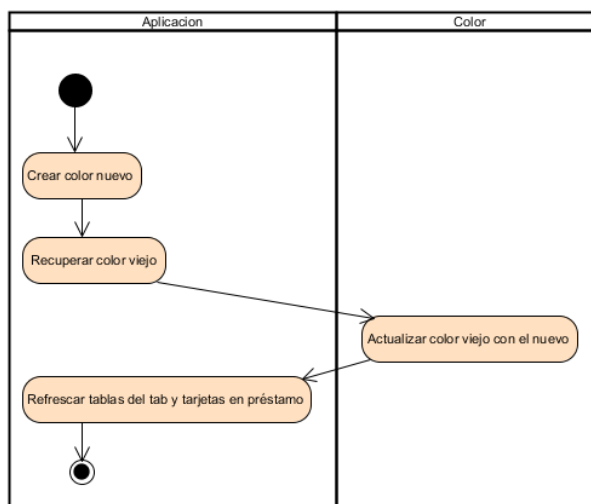


Diagrama de actividades

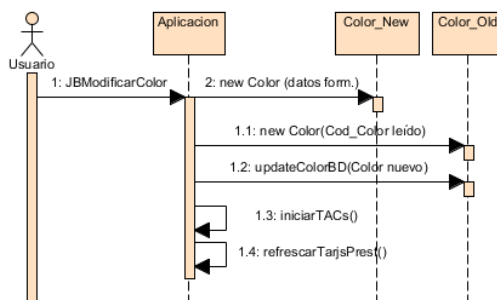


Diagrama de secuencia

4.10. Caso de uso 48: Dar de baja color

Cerramos el capítulo de colores añadiendo un nuevo botón al panel destinado a albergarlos. La secuencia de pasos a seguir es análoga a la que se da cuando damos de baja un nivel o eliminamos un color: obtener los colores seleccionados en la tabla, darlos de baja y refrescar todas las tablas del tab y la futura de tarjetas en préstamo.

Profundizando con el diseño, creamos estos diagramas que darán solución a cómo debemos implementar este caso de uso:

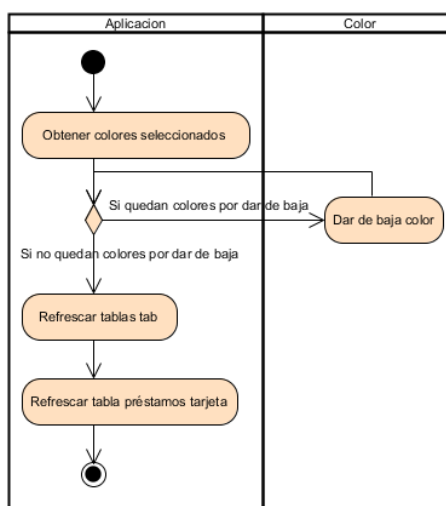


Diagrama de actividades

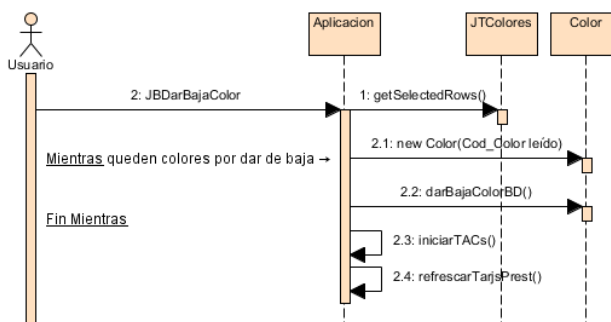
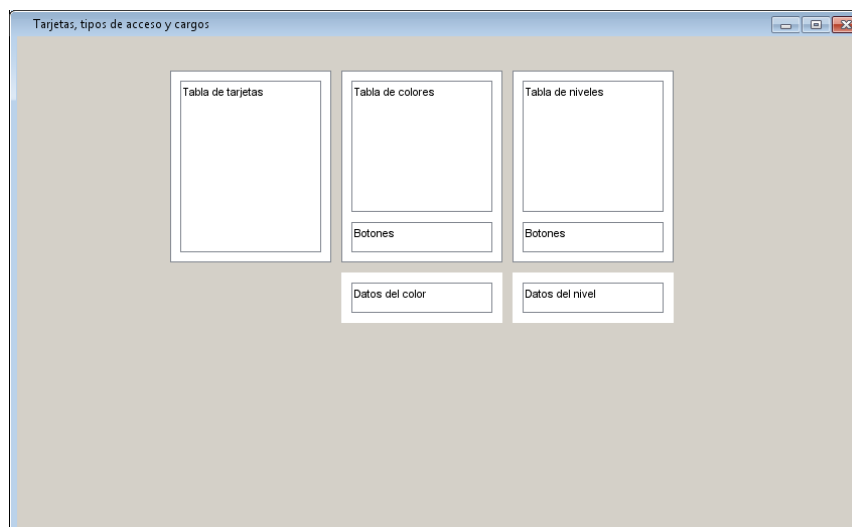


Diagrama de secuencia



4.11. Casos de uso 49 y 50: Visualizar tarjetas y visualizar tarjeta

Añadamos a la interfaz gráfica existente otra tabla más con la que poder visualizar los únicos dos datos que conforman la información útil de una tarjeta: su código y su color:



Interfaz gráfica propuesta

No variaremos la estrategia que utilizamos con la visualización de los colores...

1. Crear un objeto de tipo Tarjeta con el que realizar la búsqueda.
2. Rellenar la tabla con las tarjetas obtenidas.
3. Pintar la tabla con los colores de cada tarjeta.

... para la cual se necesita...

MÉTODO	FUNCIONALIDAD
<i>obtenerTarjetas</i>	Método que obtiene la numeración de todas las tarjetas registradas.
<i>obtenerColoresTarjs</i>	Método que obtiene la coloración de todas y cada una de las tarjetas activas.

Métodos a implementar y a añadir a la API



... y los pasos a seguir son:

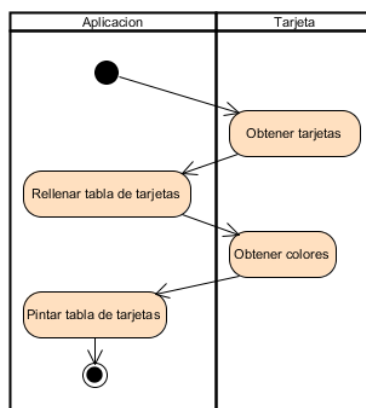


Diagrama de actividades

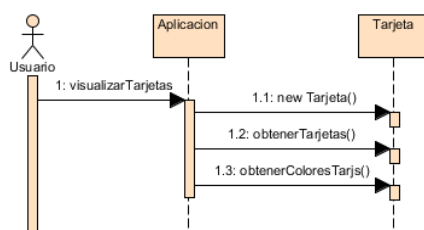
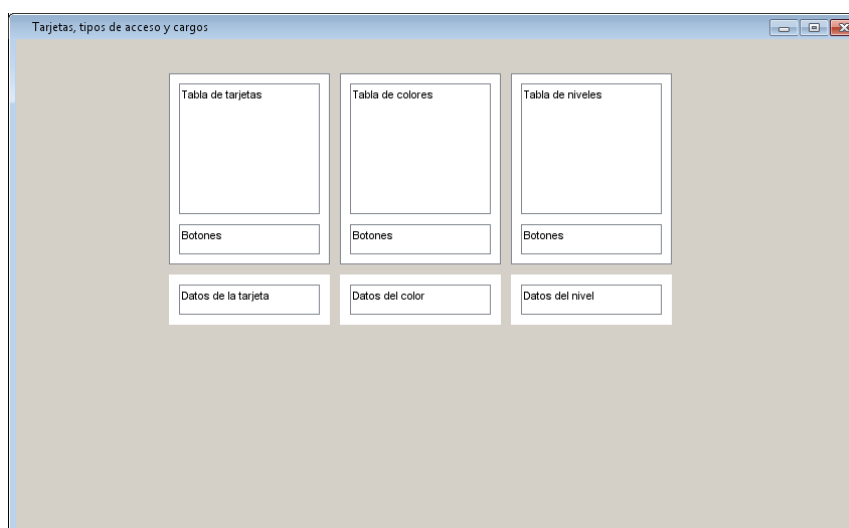


Diagrama de secuencia

4.12. Caso de uso 51: Añadir tarjeta

Con las tarjetas la cosa no varía. Añadimos un panel para los botones que proporcionarán toda la funcionalidad sobre éstas y otro con el que poder introducir los datos de cada nueva tarjeta:



Interfaz gráfica propuesta



Proporcionaremos un número de tarjeta por defecto gracias al método *numTarjPorDefecto*:

MÉTODO	FUNCIONALIDAD
<i>numTarjPorDefecto</i>	Método que devuelve el número de tarjeta por defecto.

Método a implementar y a añadir a la API

La estrategia es sencilla ...

1. Crear un objeto de tipo Tarjeta con los parámetros introducidos.
2. Insertar la nueva tarjeta.
3. Refrescar la tabla de tarjetas.

... o visto de otra manera:

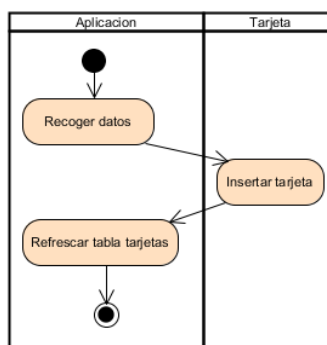


Diagrama de actividades

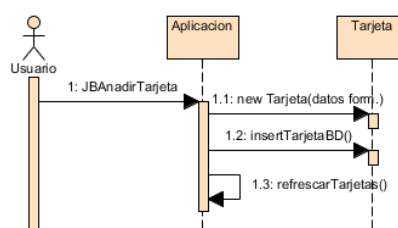


Diagrama de secuencia



4.13. Caso de uso 52: Borrar tarjeta

Creamos un botón de eliminación en el panel de botones de tarjetas y diseñamos el siguiente plan de acción:

1. Obtener las tarjetas seleccionadas.
2. Recorrerlas eliminándolas una a una.
3. Refrescar todas las tablas del tab y la futura de tarjetas en préstamo.

Además, necesitamos crear un constructor que obtenga la tarjeta a partir del código que le proporcionemos:

MÉTODO	FUNCIONALIDAD
<i>Tarjeta</i>	Constructor que crea un objeto Tarjeta a partir del código proporcionado.

Método a implementar y a añadir a la API

Resumimos el proceso en los siguientes diagramas:

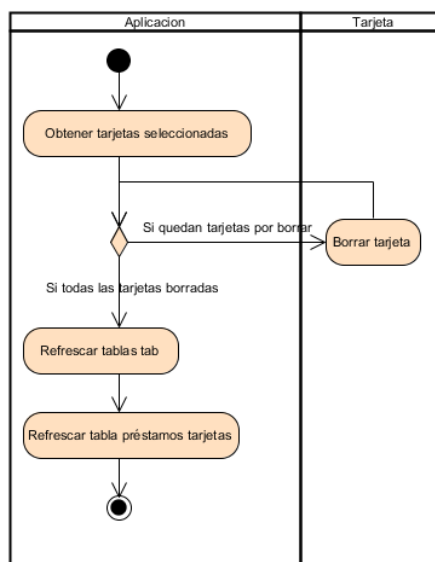


Diagrama de actividades

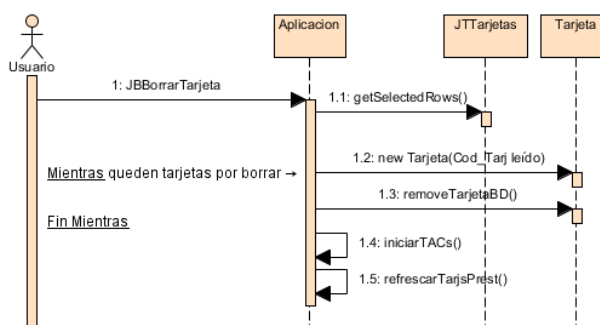


Diagrama de secuencia



4.14. Caso de uso 53: Modificar tarjeta

Añadiendo un simple botón al panel de botones dispondremos de todos los componentes gráficos necesarios para llevar a cabo el siguiente proceso:

1. Crear la nueva tarjeta con los datos recogidos del formulario.
2. Recuperar la tarjeta vieja de la base de datos.
3. Actualizar la tarjeta vieja con la nueva.
4. Refrescar las tablas del tab y la de tarjetas en préstamo (recordemos, aún sin desarrollar).

Necesitamos otro constructor similar a los ya realizados para niveles y colores:

MÉTODO	FUNCIONALIDAD
<i>Tarjeta</i>	Constructor que crea un objeto de Tarjeta con los datos rescatados de la base de datos mediante el código proporcionado.

Método a implementar y a añadir a la API

El diseño da lugar a los siguientes diagramas de actividades y de secuencia:

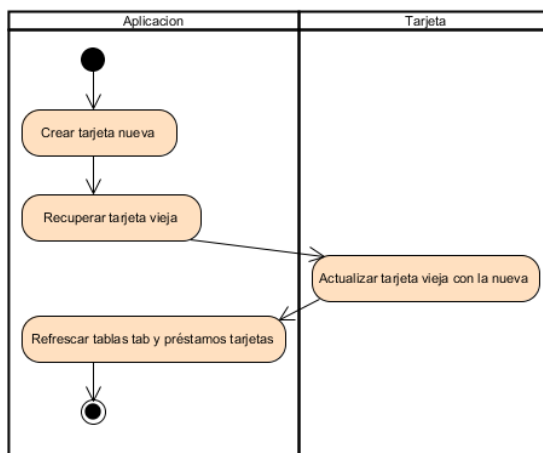


Diagrama de actividades

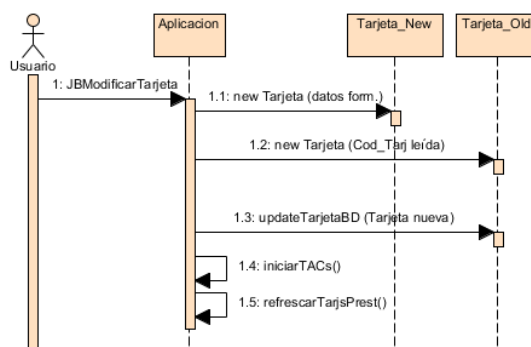


Diagrama de secuencia



4.15. Caso de uso 54: Dar de baja tarjeta

Añadimos el último botón al panel de botones de tarjetas y diseñamos la misma estrategia de baja que la llevada a cabo anteriormente: obtener tarjetas, dar de baja tarjetas y refrescar todas las tablas necesarias.

Estos diagramas nos serán de gran utilidad en la implementación:

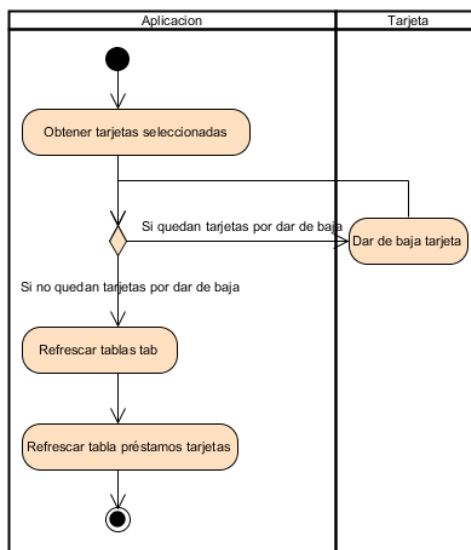


Diagrama de actividades

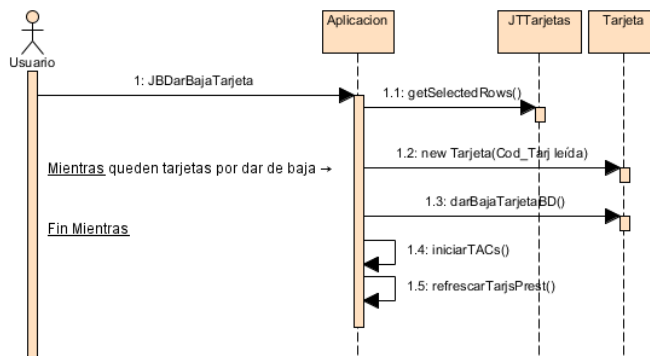
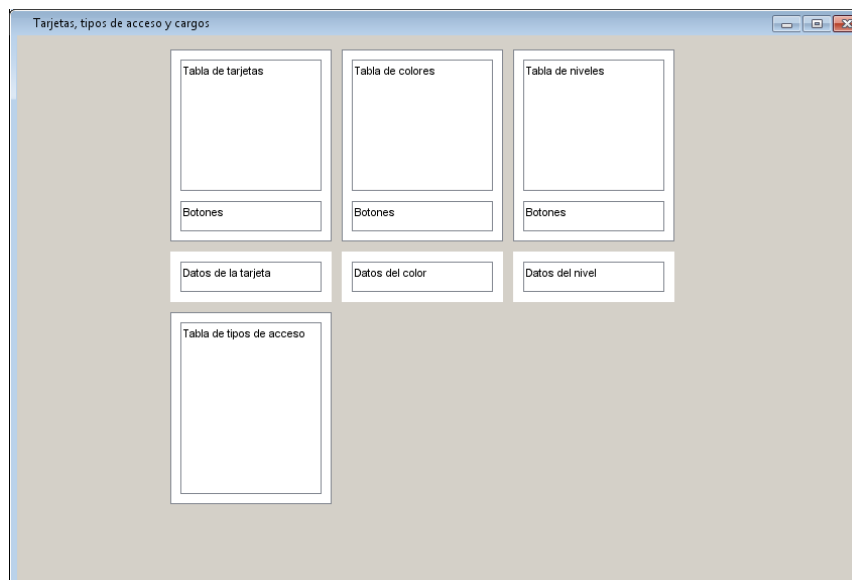


Diagrama de secuencia



4.16. Caso de uso 55: Visualizar tipos de acceso

Así como para los niveles, colores y tarjetas dispusimos de la misma tabla para mostrar los datos principales y totales, en el caso que nos ocupa no podremos realizar esto ya que no coinciden en esta ocasión. Optamos por mostrar el código de acceso y actividad en la tabla y el resto de datos a través de un formulario (véase el siguiente caso de uso):



Interfaz gráfica propuesta

Lo que haremos será:

1. Crear un objeto de tipo T_Acceso que nos permita realizar la búsqueda.
2. Rellenar la tabla de tipos de acceso con los resultados obtenidos.

Antes de nada, debemos crear un método que realice la búsqueda ...

MÉTODO	FUNCIONALIDAD
<i>obtenerTAccesos</i>	Método que recupera todos los códigos y actividades de los tipos de acceso existentes.

Método a implementar y a añadir a la API



... y después implementar el siguiente proceso:

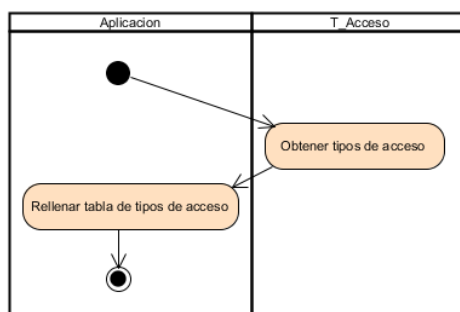


Diagrama de actividades

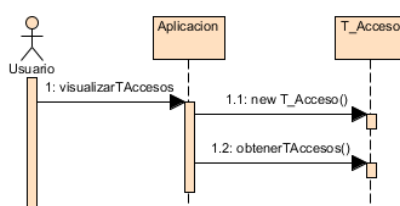
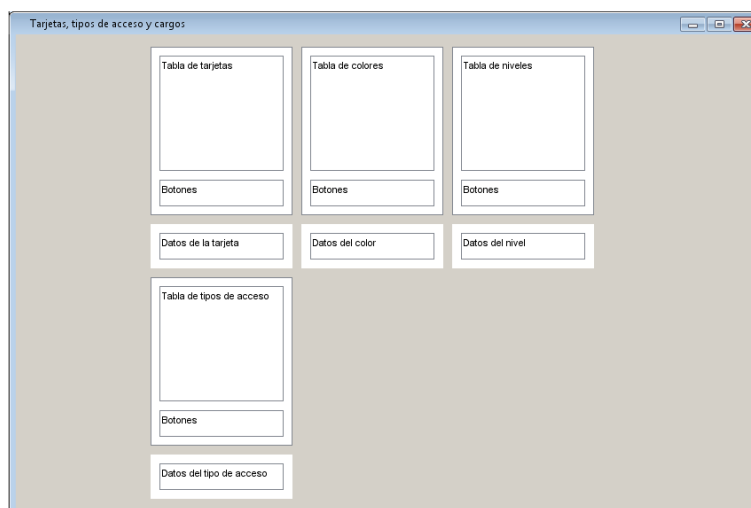


Diagrama de secuencia

4.17. Caso de uso 56: Visualizar tipo de acceso

En esta ocasión añadiremos ahora el panel de datos con el que mostrar toda la información de cada tipo de acceso y el de botones, a diferencia de los anteriores casos de uso, que comenzábamos a necesitar estos dos paneles a raíz del caso de uso de adición de elementos a la tabla:



Interfaz gráfica propuesta



Nos guiaremos de la forma que sigue:

1. Obtener el tipo de acceso seleccionado.
2. Rellenar todos los campos del formulario con éste.

Dado que con el formulario de muestra de datos también realizaremos el de inserción de tipos de acceso, el nivel de acceso será mostrado y seleccionado a través de un JComboBox. ¿Esto que implica? Los niveles serán variantes a lo largo de toda la vida del sistema, con lo que el modelo de este JComboBox también, lo que nos lleva a inicializarlo en el momento en que se realice la búsqueda de datos generales/parciales de los tipos de accesos.

Resumiendo, debemos añadir al anterior caso de uso el método *obtenerNivelesJCB*, que será ejecutado tras refrescar la tabla de tipos de acceso:

MÉTODO	FUNCIONALIDAD
<i>obtenerNivelesJCB</i>	Método que obtiene una lista de todos los niveles que existen actualmente en el sistema.
<i>T_Acceso</i>	Constructor que crea un objeto de tipo <i>T_Acceso</i> a partir del código proporcionado y los datos extraídos de la base de datos.

Métodos a implementar y a añadir a la API

Hemos añadido el segundo método a nuestra API con la intención de poder crear objetos *T_Acceso* a partir de su código, como su descripción bien indica.

No disponemos de ninguna estrategia previa que maneje este tipo de caso de uso, pero resulta muy sencillo crear una:

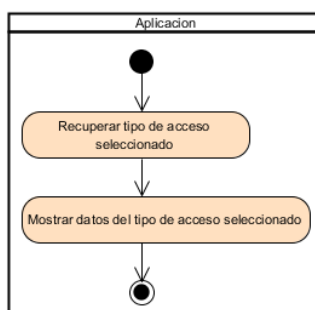


Diagrama de actividades

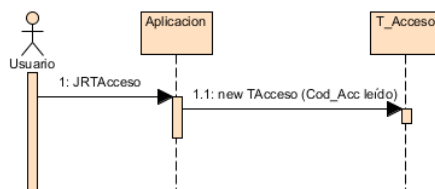


Diagrama de secuencia

Nota: El formulario con los datos será mostrado al presionar el botón de 'Modificar tipo de acceso'.

4.18. Caso de uso 57: Añadir tipo de acceso

Ya disponemos de todos los componentes gráficos necesarios para llevar a buen puerto este nuevo caso de uso, con que pasamos a diseñar los pasos a ejecutar:

1. Comprobar que las horas introducidas son correctas.
2. Crear un nuevo T_Acceso con los datos recogidos del formulario.
3. Insertar el nuevo tipo de acceso.
4. Refrescar la tabla de tipos de acceso.

Concretamos el diseño:

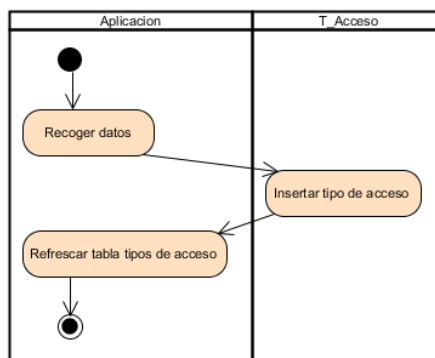


Diagrama de actividades

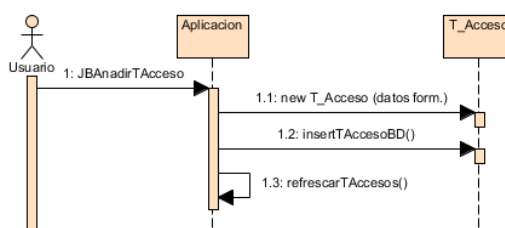


Diagrama de secuencia



4.19. Caso de uso 58: Borrar tipo de acceso

Añadir un botón de borrado al panel de botones de tipos de acceso debería bastar para poder llevar a cabo el siguiente proceso:

1. Obtener los tipos de accesos seleccionados.
2. Recorrer los tipos de accesos seleccionados eliminándolos uno a uno.
3. Refrescar todas las tablas de niveles y tipos de accesos.

No es necesario añadir ningún nuevo método a la API para poder implementar estos diagramas:

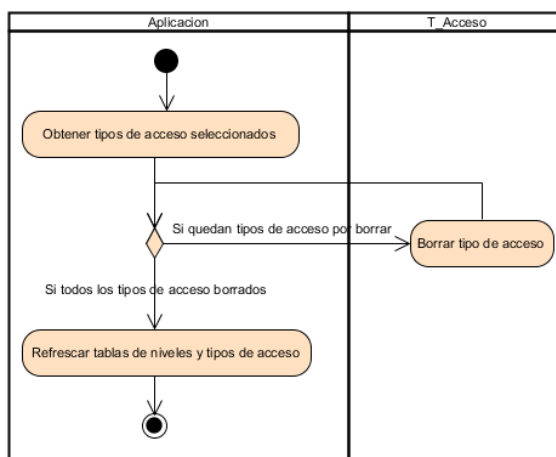


Diagrama de actividades

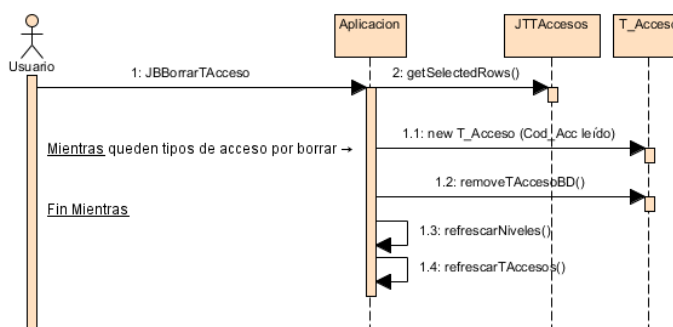


Diagrama de secuencia



4.20. Caso de uso 59: Modificar tipo de acceso

Damos por terminada la interfaz destinada a manejar los tipos de acceso rematándola con un botón que desencadene todo este proceso:

1. Crear el nuevo tipo de acceso con los datos recogidos del formulario.
2. Recuperar el viejo tipo de acceso de la base de datos.
3. Modificar el tipo de acceso viejo con el nuevo.
4. Refrescar la tabla de tipos de acceso.

Estos son los diagramas que hemos diseñado de cara al posterior proceso de implementación:

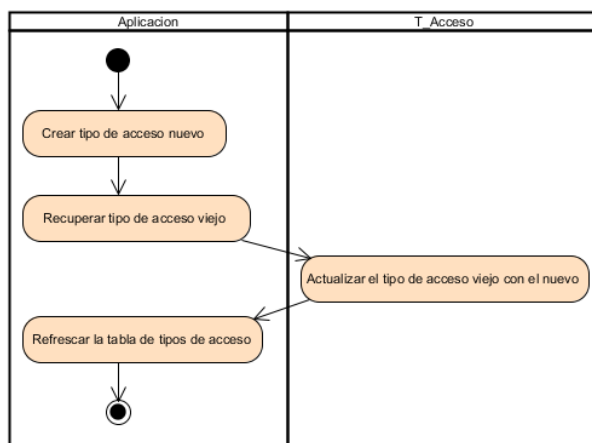


Diagrama de actividades

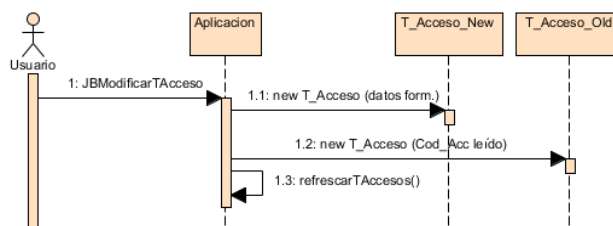


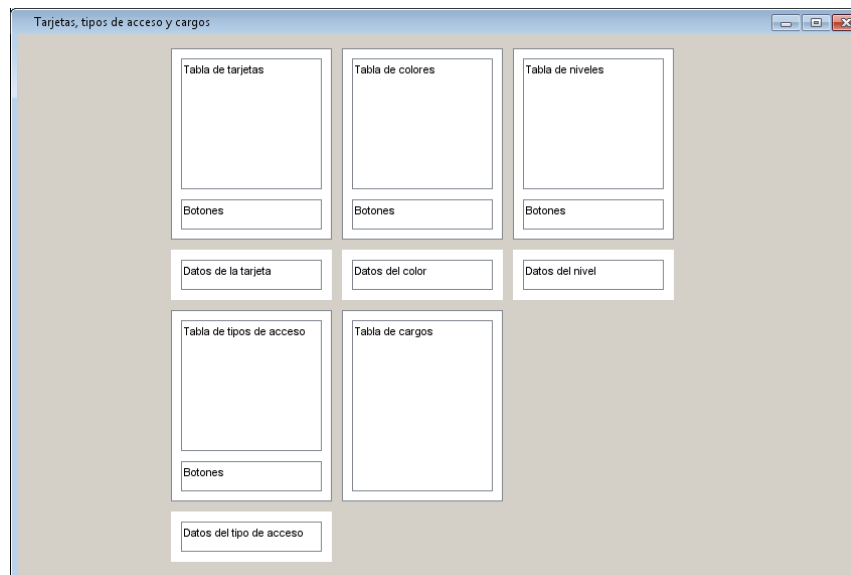
Diagrama de secuencia



4.2.1. Casos de uso 60 y 61: Visualizar cargos y visualizar cargo

Nos encontramos ante la misma situación que con los niveles, colores y tarjetas, cambiándolos por el ente 'cargo'.

En un principio diseñamos una tabla de cargos que añadiremos a las ya existentes y que tendrá la funcionalidad de visualizar todos los datos de todos los cargos existentes:



Interfaz gráfica propuesta

Podemos mostrar los datos en la tabla de la siguiente forma:

1. Crear un objeto de tipo Cargo con el que realizar la búsqueda.
2. Rellenar la tabla con todos los datos obtenidos.

Como ocurría anteriormente, necesitaremos añadir a nuestra API un método que realice la búsqueda de los cargos:

MÉTODO	FUNCIONALIDAD
<i>obtenerCargos</i>	Método que recupera de la base de datos todos los códigos y descripciones de los cargos que ésta almacena.

Método a implementar y a añadir a la API



Seguiremos los siguientes diagramas:

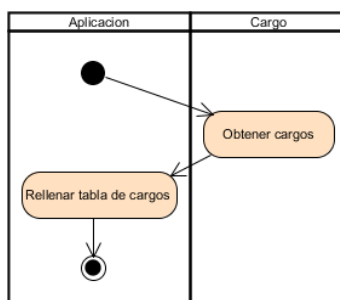


Diagrama de actividades

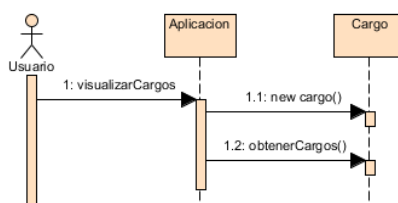
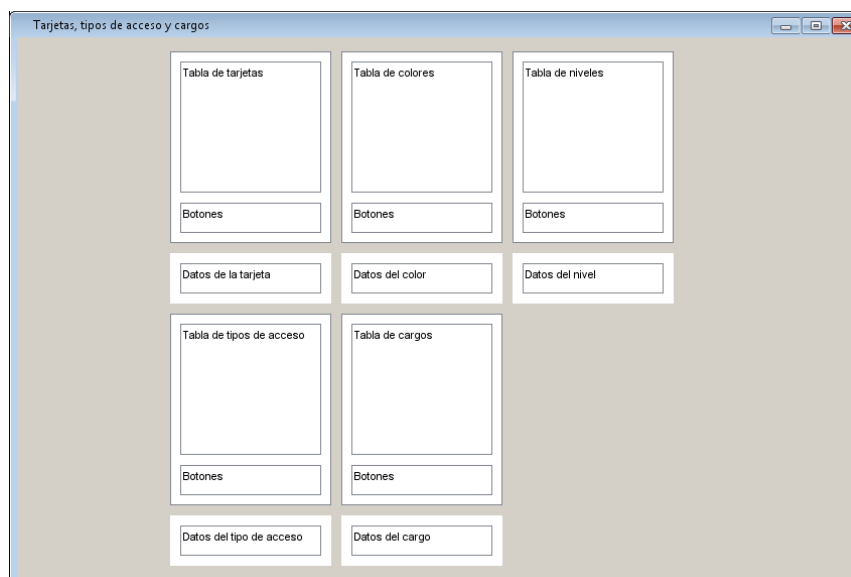


Diagrama de secuencia

4.2.2. Caso de uso 62: Añadir cargo

A la JTable del caso de uso anterior le añadimos un panel de botones y otro en el que mostrar e introducir datos al sistema:



Interfaz gráfica propuesta



El caso de uso puede ser implementado de la siguiente forma:

1. Crear un objeto Cargo con los datos proporcionados.
2. Insertar el nuevo cargo.
3. Refrescar la tabla de cargos.

Llevamos este proceso a un nivel de mayor detalle a través de los siguientes diagramas de actividades y de secuencia:

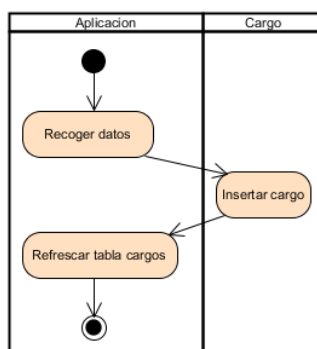


Diagrama de actividades

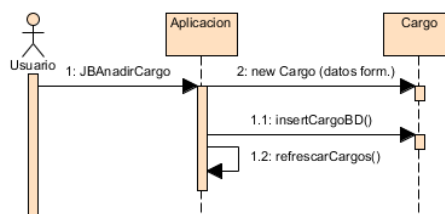


Diagrama de secuencia

4.23. Caso de uso 63: Borrar cargo

En el punto opuesto a añadir un cargo tenemos el de borrarlo. Nos basaremos en la tabla y un nuevo botón que añadamos al JPanel de botones para:

1. Obtener los cargos seleccionados.
2. Recorrer todos los cargos obtenidos eliminándolos uno a uno.
3. Refrescar el contenido de la tabla de cargos.

En el recorrido de cargos no sería posible recuperarlos uno a uno si no es gracias a este nuevo constructor que debemos añadir a la API:

MÉTODO	FUNCIONALIDAD
<i>Cargo</i>	Constructor que crea un Cargo con la ayuda de la base de datos y su código asociado.

Método a implementar y a añadir a la API



Aquí tenemos los diagramas que utilizaremos en la etapa de codificación:

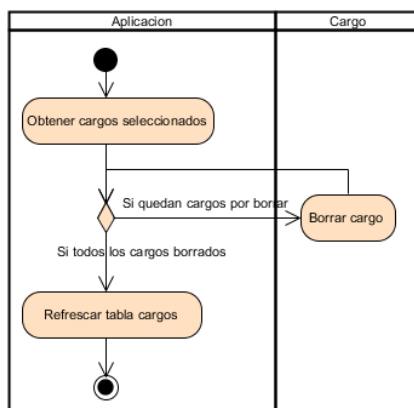


Diagrama de actividades

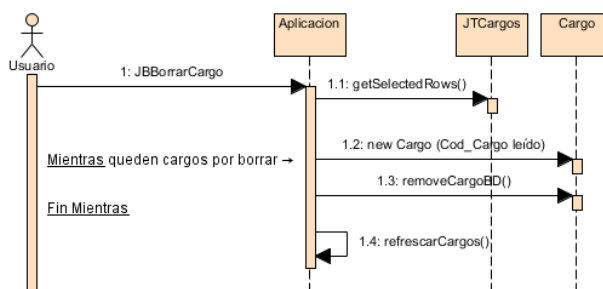


Diagrama de secuencia



4.24. Caso de uso 64: Modificar cargo

Último caso de uso de la iteración. Modificaremos el cargo siguiendo la siguiente estrategia:

1. Crear el nuevo cargo con los datos obtenidos del formulario.
2. Recuperar de la base de datos el viejo cargo a actualizar.
3. Actualizar el cargo viejo con el nuevo.
4. Refrescar las tablas de cargos.

No es necesario añadir ningún nuevo método a la API, pero sí crear un diagrama de actividades y otro de secuencia que nos ayuden en la etapa de codificación que estamos a punto de comenzar:

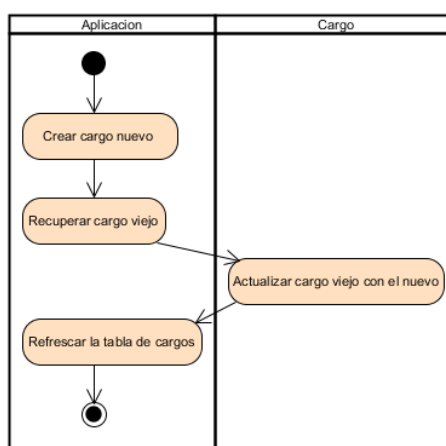


Diagrama de actividades

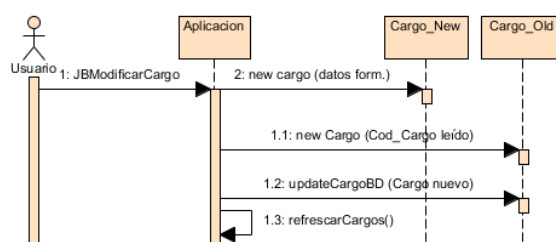


Diagrama de secuencia



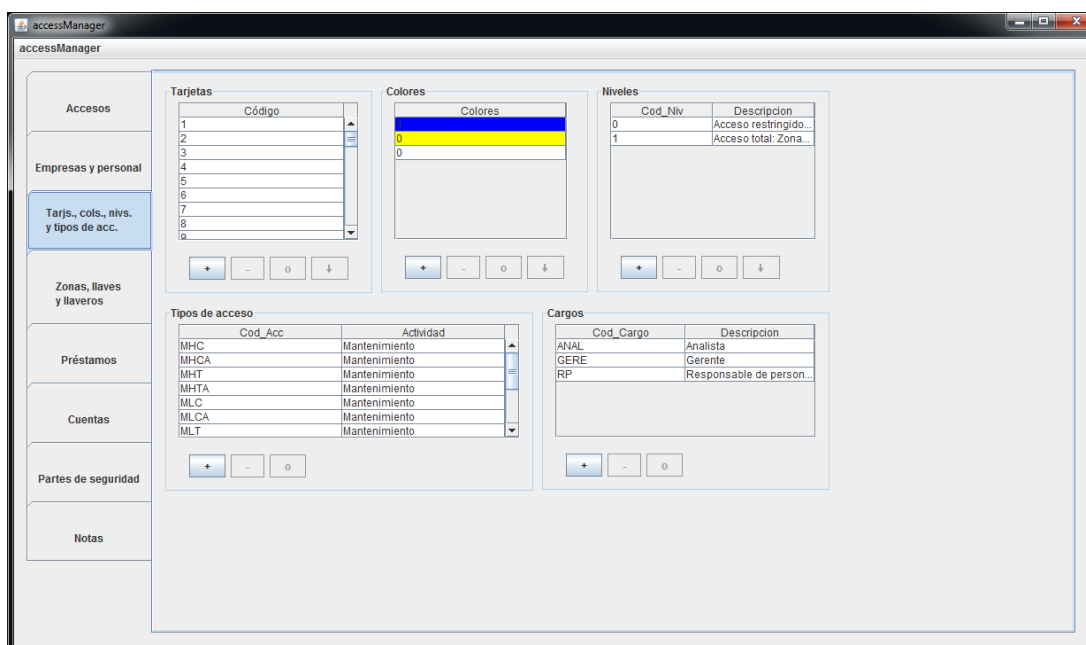
5. Codificación e integración

Una vez tenemos todo el diseño hecho, pasamos a codificarlo. Esta fase no resulta complicada, pues los datos a manejar son triviales en comparación con los manejados en la iteración anterior.

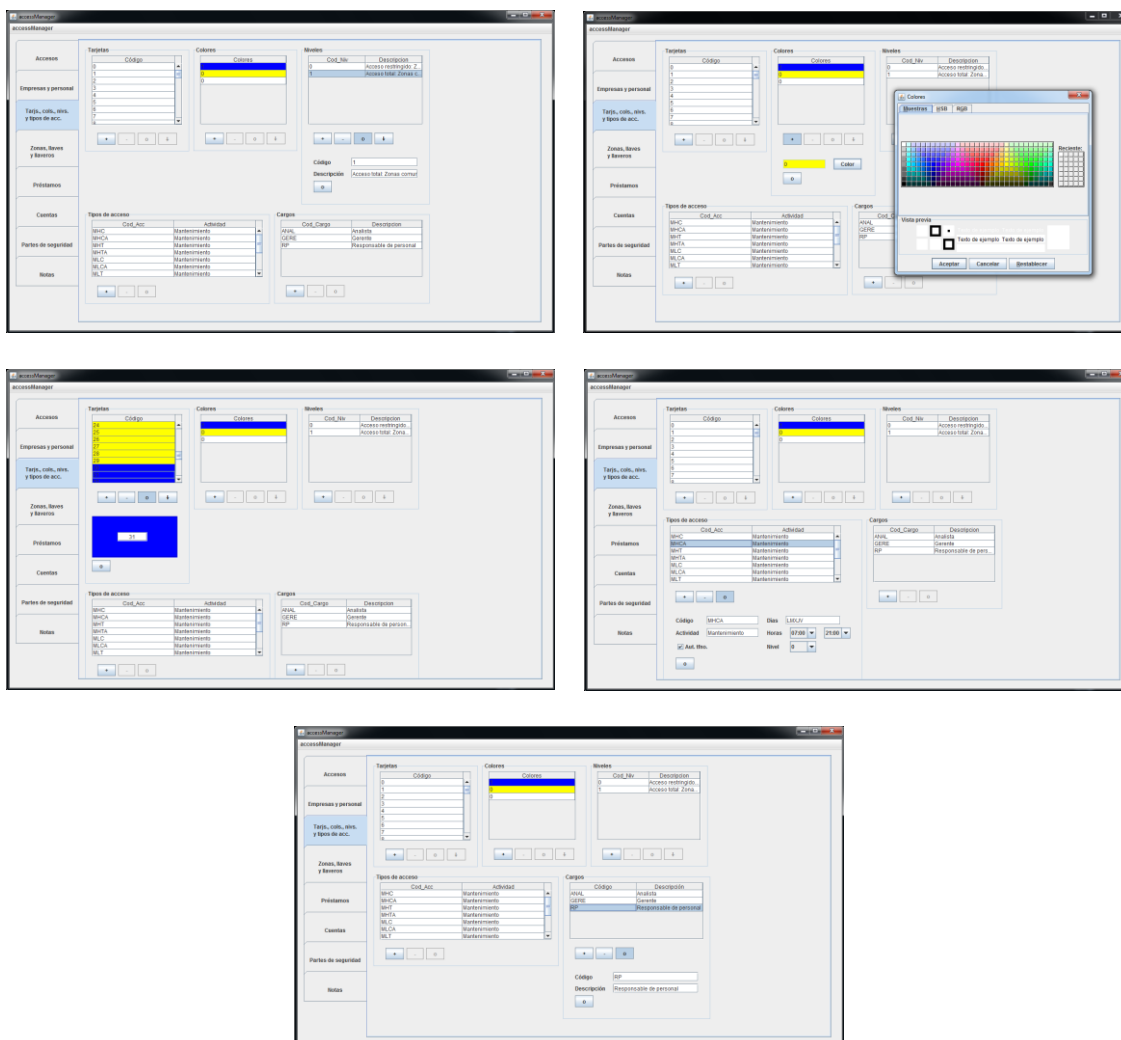
Empleamos los diseños de interfaz desarrollados teniendo en cuenta que todos los paneles serán auto excluyentes, de forma que sólo se pueda estar manejando uno en cada momento. Esto se debe que al estar íntimamente relacionados unos elementos con otros podremos combinarlos en la creación y modificación de todos ellos. Además, al manejar varios elementos a la vez, la autoexclusión ayudará a mantener la interfaz mucho más ordenada, menos caótica, hecho que simplificará la tarea del administrador.

De cara a integrar los componentes software, toda la iteración se codifica sobre el JFrame principal a través de la API desarrollada, lo que seguirá sin provocarnos ningún posible problema. Si atendemos a la integración/interrelación de las diversas pestañas, ésta solo lo deberá estar con la de préstamos, y es una que aún no hemos desarrollado. Al igual que pasó con la de accesos, esperaremos a más adelante para comprobar que todas se relacionan de forma correcta y coherente.

Proponemos en un principio las siguientes interfaces para el desarrollo de todos los casos de uso:



Interfaz de la pestaña 'Tarjs., cols., nivs. y tipos de acc.'



Interfaces parciales de la pestaña 'Tarjs., cols., nivs. y tipos de acc.'



6. Pruebas

6.1. De unidad

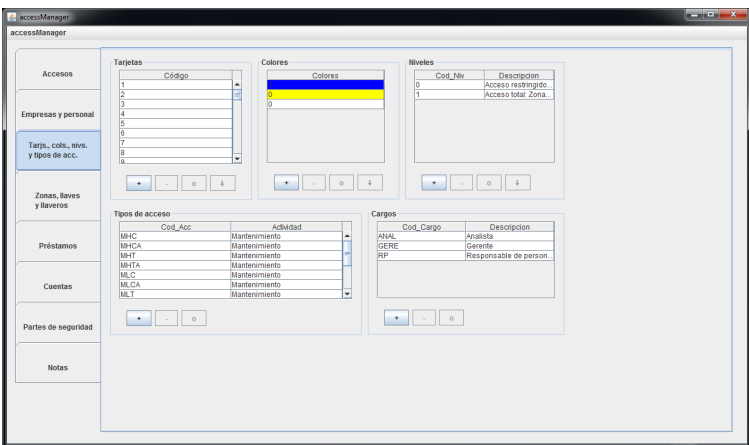
Éstas son todas las pruebas que hemos estimado oportuno realizar para asegurar la óptima calidad de nuestro trabajo realizado:

Caso de prueba	5.01 - Casos de uso 37, 38, 43, 44, 49, 50, 55, 60 y 61: Visualizaciones
Objetivo	Comprobar que el usuario puede visualizar todos los datos de niveles, colores, tarjetas y cargos y los principales atributos de los tipos diversos tipos de acceso existentes.
Actividades	Arrancar la aplicación e ir a la pestaña de tarjetas, comprobando que las tablas reflejen la realidad (contenido de la base de datos).
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. • Aplicación Notepad++.
Datos de entrada	La población de prueba existente en la base de datos.
Salida esperada	<p>Según la población existente, las tablas de niveles, colores, tarjetas, tipos de acceso y cargos deberían contener las siguientes tuplas:</p> <pre> -- TABLA 'Nivel' INSERT INTO Nivel (Cod_Niv,F_Alta_N,Descripcion) VALUES (0,'2010-01-01 00:00:01','Acceso restringido: Zonas comunes'), (1,'2010-01-01 00:00:01','Acceso total: Zonas comunes + Host'); -- TABLA 'Color' INSERT INTO Color (Cod_Color,F_Alta_C,Cod_Niv,F_Alta_N) VALUES (-1,'2010-01-01 00:00:01',0,'2010-01-01 00:00:01'), (-256,'2010-01-01 00:00:01',0,'2010-01-01 00:00:01'), (-16776961,'2010-01-01 00:00:01',1,'2010-01-01 00:00:01'); -- TABLA 'Tarjeta' INSERT INTO Tarjeta (Cod_Tarj,F_Alta_T,Cod_Color,F_Alta_C,DNI) VALUES ('000','2010-01-01 00:00:01',-1,'2010-01-01 00:00:01','00000027A'), ('001','2010-01-01 00:00:01',-1,'2010-01-01 00:00:01','00000032A'), ('002','2010-01-01 00:00:01',-1,'2010-01-01 00:00:01','00000035A'), ('003','2010-01-01 00:00:01',-1,'2010-01-01 00:00:01','00000037A'), ('004','2010-01-01 00:00:01',-1,'2010-01-01 00:00:01',null), ... -- TABLA 'T_Acceso' INSERT INTO T_Acceso VALUES ('VIG','Vigilancia','LMXJVSD','00:00','23:59:59',False,1,'2010-01-01 00:00:01'), ('MHT','Mantenimiento','LMXJV','07:00','21:00',False,1,'2010-01-01 00:00:01'), ('MHC','Mantenimiento','LMXJV','07:00','21:00',False,0,'2010-01-01 00:00:01'), ('MHTA','Mantenimiento','LMXJV','07:00','21:00',True,1,'2010-01-01 00:00:01'), ('MHCA','Mantenimiento','LMXJV','07:00','21:00',True,0,'2010-01-01 00:00:01'), ('MLT','Mantenimiento','LMXJVSD','00:00','23:59:59',False,1,'2010-01-01 00:00:01'), ('MLC','Mantenimiento','LMXJVSD','00:00','23:59:59',False,0,'2010-01-01 00:00:01'), ('MLTA','Mantenimiento','LMXJVSD','00:00','23:59:59',True,1,'2010-01-01 00:00:01'), ('MLCA','Mantenimiento','LMXJVSD','00:00','23:59:59',True,0,'2010-01-01 00:00:01'), ('TRT','Trabajo','LMXJV','09:00','21:00',False,1,'2010-01-01 00:00:01'), ('TRC','Trabajo','LMXJV','09:00','21:00',False,0,'2010-01-01 00:00:01'), ('TRTA','Trabajo','LMXJV','09:00','21:00',True,1,'2010-01-01 00:00:01'), ('TRCA','Trabajo','LMXJV','09:00','21:00',True,0,'2010-01-01 00:00:01'); -- TABLA 'Cargo' INSERT INTO Cargo VALUES ('RP','Responsable de personal'), ('ANAL','Analista'), ('GERE','Gerente'); </pre>



Salida observada

La esperada:

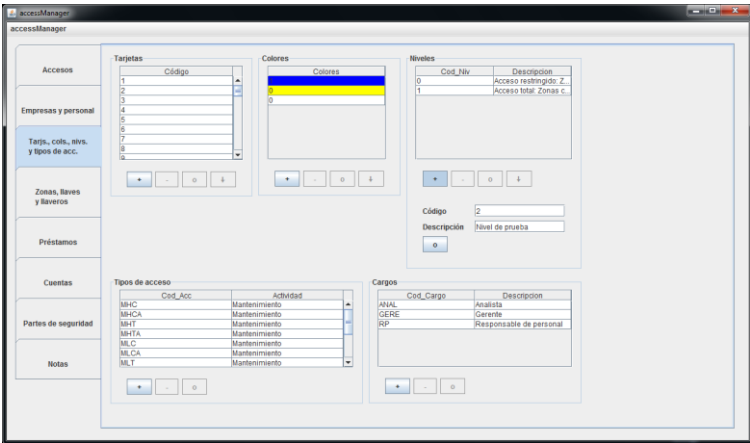
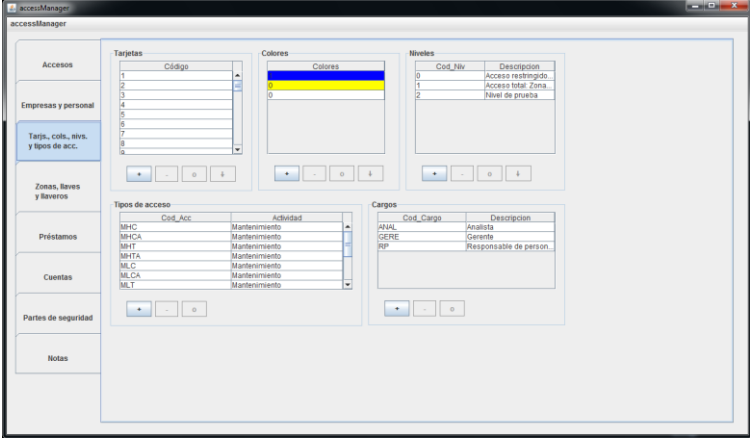


**Evaluación/
conclusión**

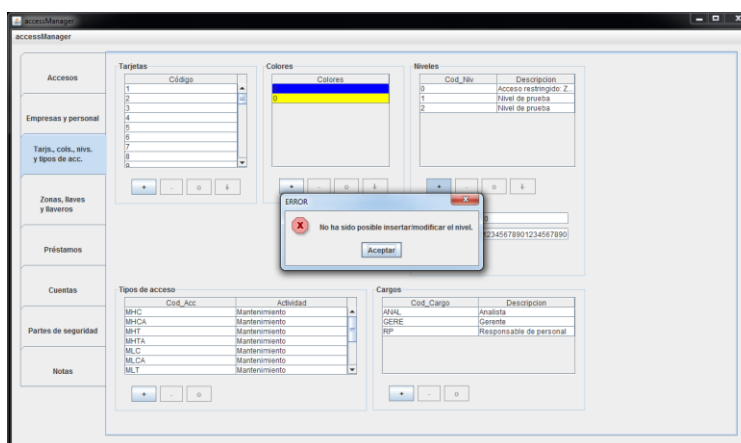
Los casos de uso '37, 38, 43, 44, 49, 50, 55, 60 y 61 - Visualizaciones' han sido realizados satisfactoriamente.

Caso de prueba para los casos de uso '37, 38, 43, 44, 49, 50, 55, 60 y 61 - Visualizaciones'



Caso de prueba	5.02 - Caso de uso 39: Añadir nivel
Objetivo	Probar que el usuario puede realizar el alta de niveles si introduce los datos correctamente y que la inserción no tiene lugar si el nivel existe o el formato de los datos es incorrecto.
Actividades	Ir a la pestaña de tarjetas donde introduciremos un nuevo nivel e intentaremos introducir otro que no disponga de un formato adecuado.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos con población de prueba.
Datos de entrada	<p>Probaremos a insertar los siguientes niveles:</p> <ul style="list-style-type: none">• '2 – Nivel de prueba'• '10 – 0123456789012345678901234567890123456789012345678901234567890' 
Salida esperada	El primer nivel deberá insertarse, pero no el segundo.
Salida observada	<p>La esperada:</p> <ul style="list-style-type: none">• En el primer caso: 

- En el otro:

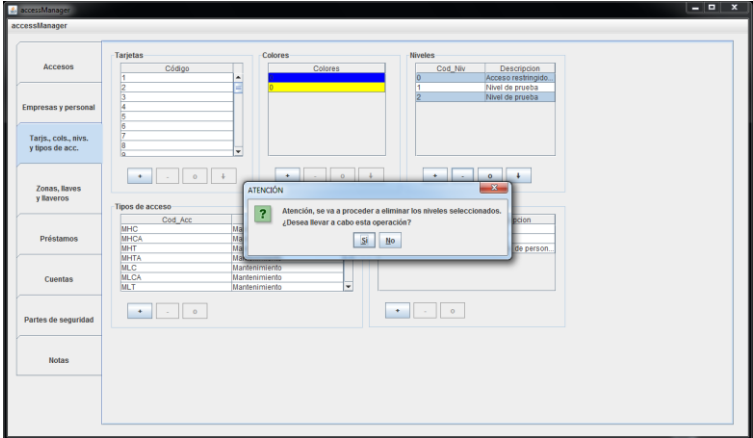
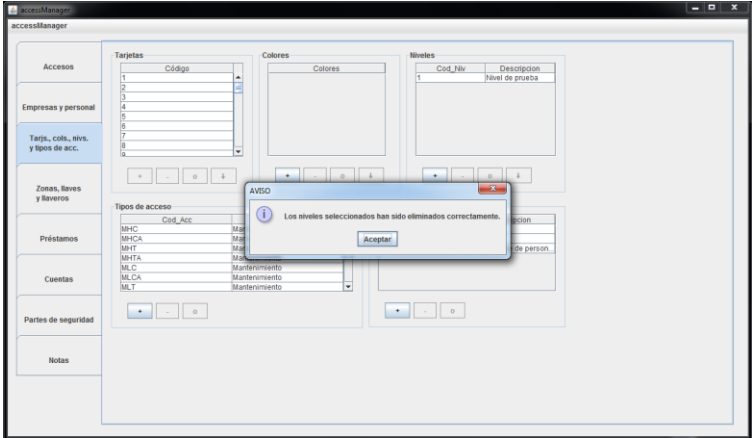


Añadir que la generación del número de nivel por defecto es correcta.

**Evaluación/
conclusión**

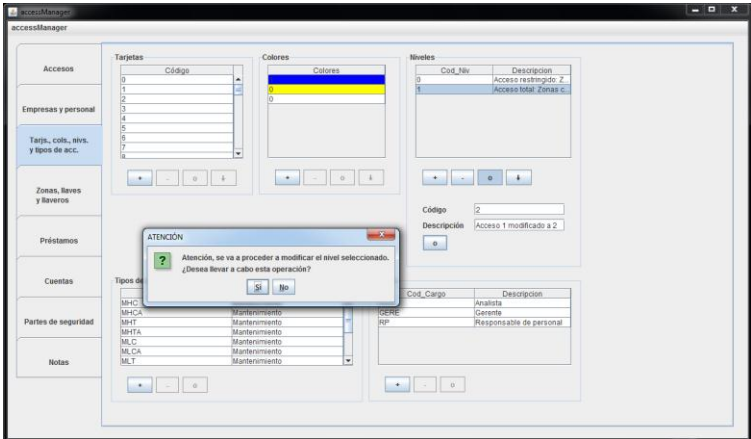
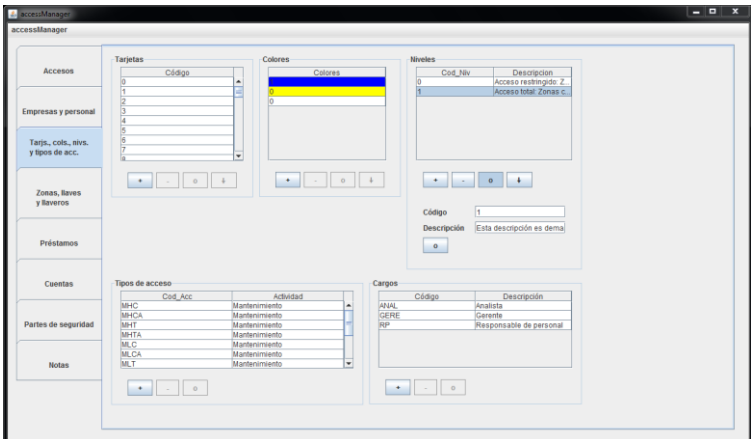
El caso de uso '39 - Añadir nivel' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '39 - Añadir nivel'

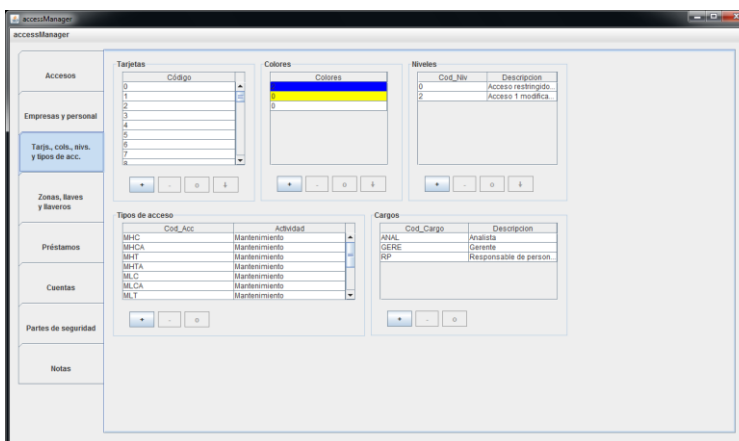
Caso de prueba	5.03 - Caso de uso 40: Borrar nivel
Objetivo	Verificar que el usuario puede borrar todos los niveles que desee mediante su selección y petición al sistema.
Actividades	Arrancar la aplicación e ir a la pestaña de tarjetas donde seleccionaremos varios niveles indistintamente y probaremos a eliminarlos.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba.
Datos de entrada	<p>Borraremos los niveles 0 y 3:</p> 
Salida esperada	Todas las tablas serán refrescadas, constando sólo los elementos que debieran. En este caso desaparecerán los niveles seleccionados junto con todos los colores y tarjetas que tengan relacionados.
Salida observada	<p>La esperada:</p> 
Evaluación/ conclusión	El caso de uso '40 - Borrar nivel' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '40 - Borrar nivel'

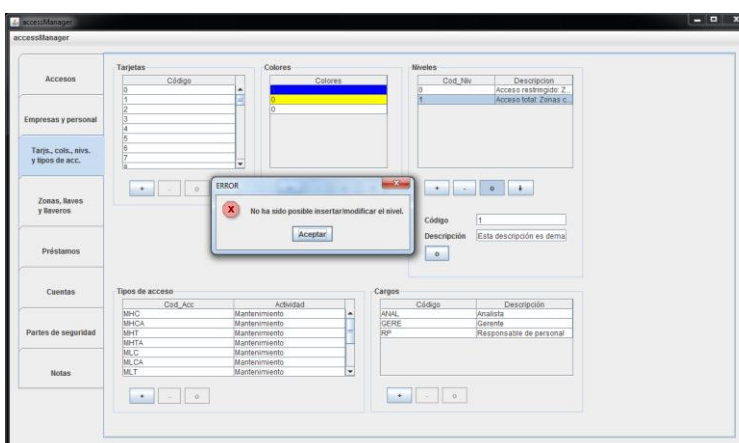


Caso de prueba	5.04 - Caso de uso 41: Modificar nivel
Objetivo	Probar que se puede modificar cualquier nivel existente en el sistema.
Actividades	Seleccionar un nivel en la tab de tarjetas, modificarlo y comprobar que los cambios se almacenan en la base de datos y tienen efecto en el resto de las tablas.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos con población de prueba.
Datos de entrada	<p>Modificamos el nivel 1 editando ambos campos dándoles un formato correcto y otro incorrecto:</p>  
Salida esperada	<p>En el primer caso deberían surtir efecto los cambios en las tablas de niveles y colores, actualizando la segunda tupla de la primera a "2 – Acceso 1 modificado a 2" y la primera tupla de colores a "2".</p> <p>En el segundo la aplicación nos mostrará un error avisándonos de que no se puede modificar el nivel.</p>

La esperada:



Salida observada

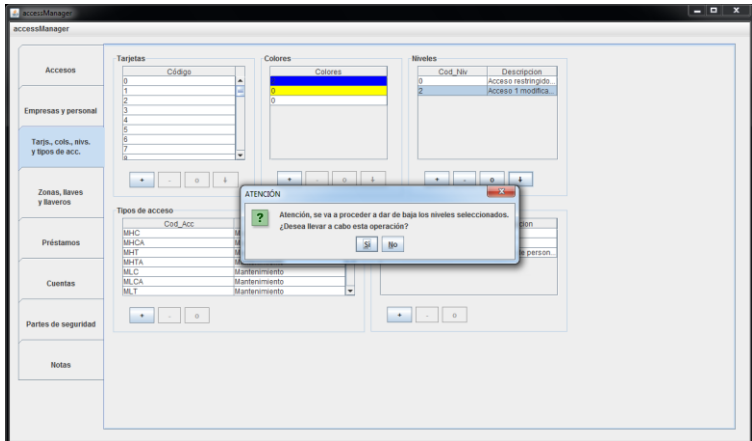


Evaluación/
conclusión

El caso de uso '41 - Modificar nivel' ha sido realizado satisfactoriamente.

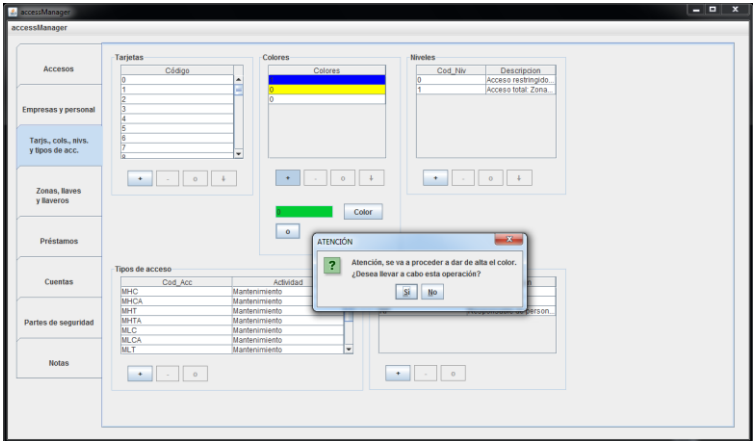
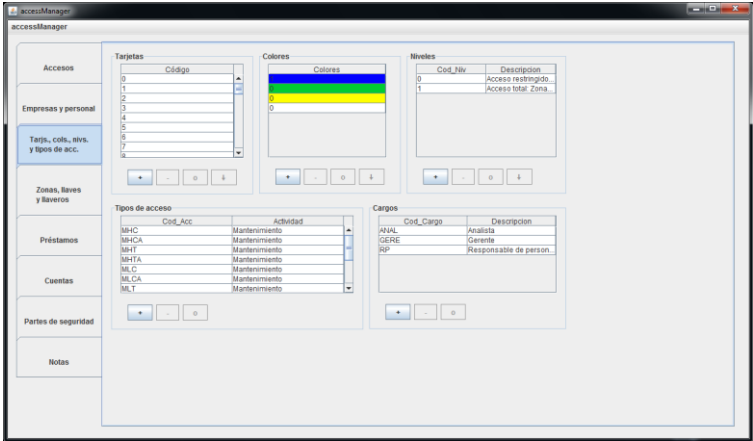
Caso de prueba para el caso de uso '41 - Modificar nivel'



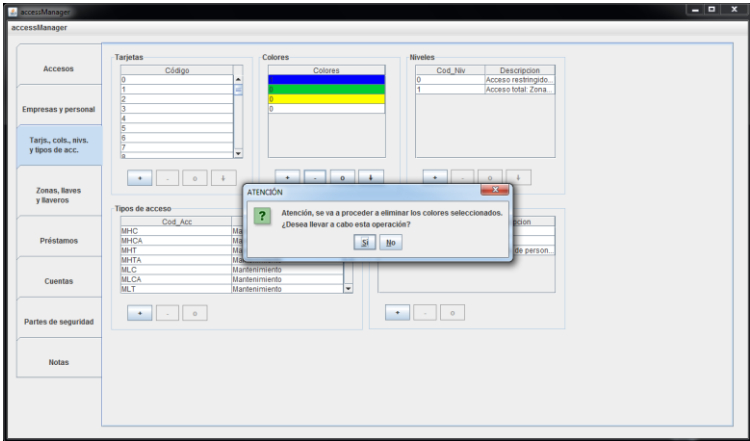
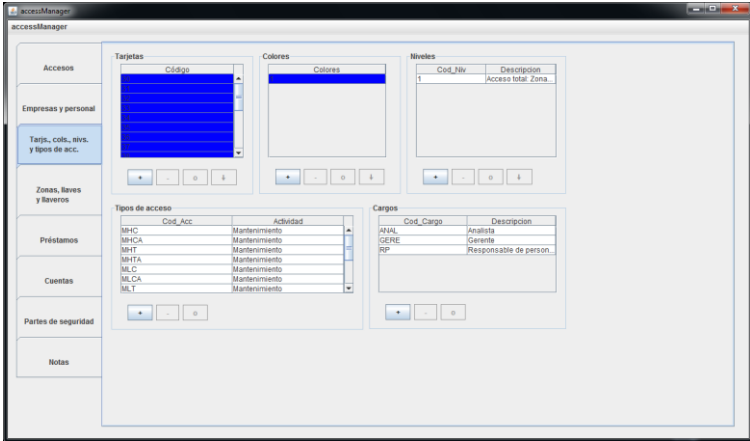
Caso de prueba	5.05 - Caso de uso 42: Dar de baja nivel																																																																																																																												
Objetivo	Demostrar que cuando el usuario da de baja los niveles que estime oportunos, el sistema se comporta tal y como esperamos.																																																																																																																												
Actividades	Arrancar la aplicación e ir a la pestaña de tarjetas, donde seleccionaremos un nivel, lo daremos de baja y verificaremos que esta se ha llevado a cabo.																																																																																																																												
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos en tras ejecutar el anterior caso de prueba. • Utilidad phpMyAdmin. 																																																																																																																												
Datos de entrada	<p>Damos de baja el nivel 2:</p> 																																																																																																																												
Salida esperada	La apariencia de la aplicación será similar a la que obtenemos del borrado, pero en este caso el nivel 2.																																																																																																																												
Salida observada	<p>La esperada:</p> <table border="1"> <thead> <tr> <th>Cod_Niv</th> <th>Nivel de acceso</th> <th>F_Alta_N</th> <th>Fecha y hora de inserción del nivel</th> <th>F_Baja_N</th> <th>Fecha y hora de baja del nivel</th> <th>Descripcion</th> <th>Descripción del nivel de acceso</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Acceso restringido: Zonas comunes</td> <td></td> </tr> <tr> <td>2</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>2010-05-20 13:43:43</td> <td></td> <td>Acceso 1 modificado a 2</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Cod_Color</th> <th>Color de la tarjeta</th> <th>F_Alta_C</th> <th>Fecha y hora de inserción del color</th> <th>F_Baja_C</th> <th>Fecha y hora de baja del color</th> <th>Cod_Niv</th> <th>Nivel de acceso</th> <th>F_Alta_N</th> <th>Fecha y hora de inserción del nivel</th> </tr> </thead> <tbody> <tr> <td>-16776961</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>2010-05-20 13:43:43</td> <td></td> <td>2</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> </tr> <tr> <td>-256</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>0</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> </tr> <tr> <td>-1</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>0</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Cod_Tarj</th> <th>Número de tarjeta</th> <th>F_Alta_T</th> <th>Fecha y hora de inserción de la tarjeta</th> <th>F_Baja_T</th> <th>Fecha y hora de baja de la tarjeta</th> <th>Cod_Color</th> <th>Color de la tarjeta</th> <th>F_Alta_C</th> <th>Fecha y hora de inserción del color</th> <th>DNI</th> <th>DNI de la persona que la tiene en su posesión</th> </tr> </thead> <tbody> <tr> <td>30</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>2010-05-20 13:43:43</td> <td></td> <td>-16776961</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>00000001A</td> <td></td> </tr> <tr> <td>31</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>2010-05-20 13:43:43</td> <td></td> <td>-16776961</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>00000002A</td> <td></td> </tr> <tr> <td>32</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>2010-05-20 13:43:43</td> <td></td> <td>-16776961</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>00000003A</td> <td></td> </tr> <tr> <td>33</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>2010-05-20 13:43:43</td> <td></td> <td>-16776961</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> </tr> </tbody> </table> <p>...</p>	Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y hora de inserción del nivel	F_Baja_N	Fecha y hora de baja del nivel	Descripcion	Descripción del nivel de acceso	0		2010-01-01 00:00:01		NULL		Acceso restringido: Zonas comunes		2		2010-01-01 00:00:01		2010-05-20 13:43:43		Acceso 1 modificado a 2		Cod_Color	Color de la tarjeta	F_Alta_C	Fecha y hora de inserción del color	F_Baja_C	Fecha y hora de baja del color	Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y hora de inserción del nivel	-16776961		2010-01-01 00:00:01		2010-05-20 13:43:43		2		2010-01-01 00:00:01		-256		2010-01-01 00:00:01		NULL		0		2010-01-01 00:00:01		-1		2010-01-01 00:00:01		NULL		0		2010-01-01 00:00:01		Cod_Tarj	Número de tarjeta	F_Alta_T	Fecha y hora de inserción de la tarjeta	F_Baja_T	Fecha y hora de baja de la tarjeta	Cod_Color	Color de la tarjeta	F_Alta_C	Fecha y hora de inserción del color	DNI	DNI de la persona que la tiene en su posesión	30		2010-01-01 00:00:01		2010-05-20 13:43:43		-16776961		2010-01-01 00:00:01		00000001A		31		2010-01-01 00:00:01		2010-05-20 13:43:43		-16776961		2010-01-01 00:00:01		00000002A		32		2010-01-01 00:00:01		2010-05-20 13:43:43		-16776961		2010-01-01 00:00:01		00000003A		33		2010-01-01 00:00:01		2010-05-20 13:43:43		-16776961		2010-01-01 00:00:01		NULL	
Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y hora de inserción del nivel	F_Baja_N	Fecha y hora de baja del nivel	Descripcion	Descripción del nivel de acceso																																																																																																																						
0		2010-01-01 00:00:01		NULL		Acceso restringido: Zonas comunes																																																																																																																							
2		2010-01-01 00:00:01		2010-05-20 13:43:43		Acceso 1 modificado a 2																																																																																																																							
Cod_Color	Color de la tarjeta	F_Alta_C	Fecha y hora de inserción del color	F_Baja_C	Fecha y hora de baja del color	Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y hora de inserción del nivel																																																																																																																				
-16776961		2010-01-01 00:00:01		2010-05-20 13:43:43		2		2010-01-01 00:00:01																																																																																																																					
-256		2010-01-01 00:00:01		NULL		0		2010-01-01 00:00:01																																																																																																																					
-1		2010-01-01 00:00:01		NULL		0		2010-01-01 00:00:01																																																																																																																					
Cod_Tarj	Número de tarjeta	F_Alta_T	Fecha y hora de inserción de la tarjeta	F_Baja_T	Fecha y hora de baja de la tarjeta	Cod_Color	Color de la tarjeta	F_Alta_C	Fecha y hora de inserción del color	DNI	DNI de la persona que la tiene en su posesión																																																																																																																		
30		2010-01-01 00:00:01		2010-05-20 13:43:43		-16776961		2010-01-01 00:00:01		00000001A																																																																																																																			
31		2010-01-01 00:00:01		2010-05-20 13:43:43		-16776961		2010-01-01 00:00:01		00000002A																																																																																																																			
32		2010-01-01 00:00:01		2010-05-20 13:43:43		-16776961		2010-01-01 00:00:01		00000003A																																																																																																																			
33		2010-01-01 00:00:01		2010-05-20 13:43:43		-16776961		2010-01-01 00:00:01		NULL																																																																																																																			
Evaluación/conclusión	El caso de uso '42 - Dar de baja nivel' ha sido realizado satisfactoriamente.																																																																																																																												

Caso de prueba para el caso de uso '42 - Dar de baja nivel'

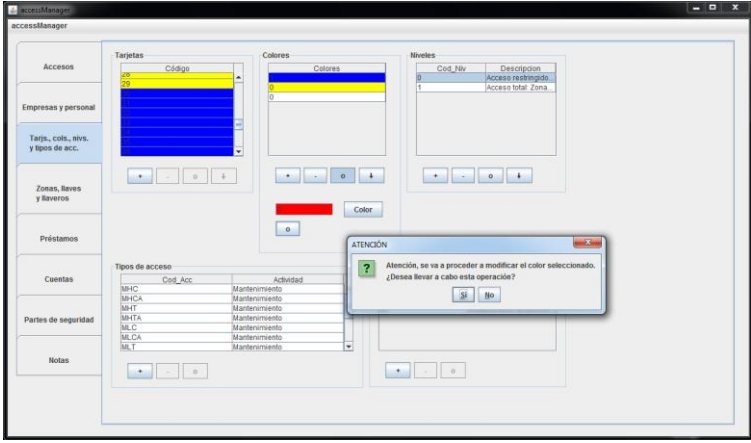
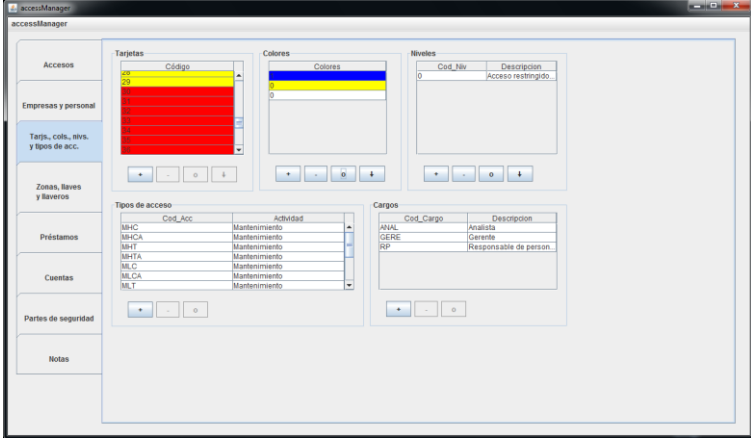


Caso de prueba	5.06 - Caso de uso 45: Añadir color
Objetivo	Comprobar que el usuario puede introducir un nuevo color.
Actividades	Introducir un nuevo color y verificar que la inserción tiene lugar de forma correcta.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba.
Datos de entrada	<p>Probaremos a insertar el color verde con nivel 0:</p> 
Salida esperada	La tabla de colores se actualizará y mostrará el nuevo color.
Salida observada	<p>La esperada:</p> 
Evaluación/conclusión	El caso de uso '45 - Añadir color' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '45 - Añadir color'

Caso de prueba	5.07 - Caso de uso 46: Borrar color
Objetivo	Chequear que el usuario puede realizar el borrado de colores, comprobando que se producen los borrados en cascada establecidos.
Actividades	Arrancar la aplicación e ir a la pestaña de tarjetas, donde borraremos los colores acordados y comprobaremos que se hayan actualizado todas las tablas como debieran.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de resultante del caso de prueba anterior.
Datos de entrada	<p>Eliminaremos todos los colores que determinan el nivel de acceso 0, es decir, el verde, amarillo y blanco:</p> 
Salida esperada	Las tablas del tab habrán sido actualizadas, siendo eliminadas todas las tarjetas correspondientes a colores borrados y dejando de existir el nivel 0 (ya que no existen colores que lo proporcionen), probando así también uno de los disparadores creados al comienzo del proyecto.
Salida observada	<p>La esperada:</p> 
Evaluación/conclusión	El caso de uso '46 - Borrar color' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '46 - Borrar color'

Caso de prueba	5.08 - Caso de uso 47: Modificar color
Objetivo	Comprobar que un color puede ser modificado, y que la modificación de éste implica la modificación de todos los entes con los que se relaciona (tarjetas y niveles).
Actividades	Al igual que siempre, arrancaremos accessManager con la población que hemos realizado de prueba, modificaremos un color y veremos si todo se desarrolla tal y como hemos previsto o surge algún problema.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba.
Datos de entrada	<p>Cambiamos el color 'Azul – 1', asignándole el 0 y un nuevo color inexistente hasta el momento en la base de datos, por ejemplo el rojo:</p> 
Salida esperada	Si todo se desarrolla según lo previsto, en la tabla de colores el color 'Azul – 1' debería pasar a ser 'Rojo – 0', provocando que las tarjetas de color azul pasasen a ser rojas y el nivel 1 desaparecerá por no existir colores que lo representen.
Salida observada	<p>La esperada:</p> 
Evaluación/ conclusión	El caso de uso '47 - Modificar color' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '47 - Modificar color'



Caso de prueba 5.09 - Caso de uso 48: Dar de baja color

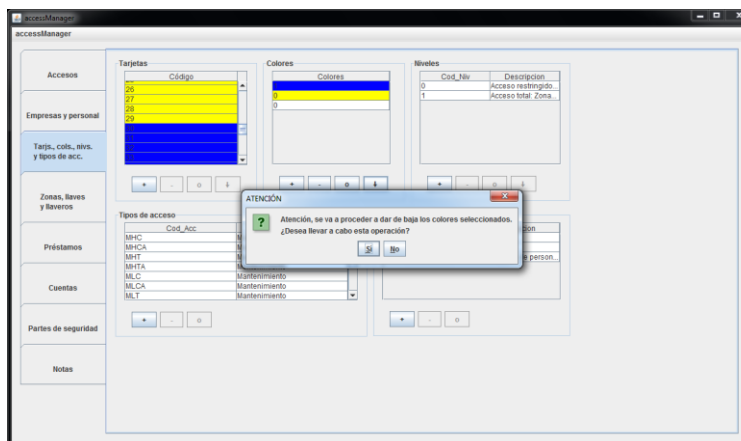
Objetivo Demostrar que el proceso de dar baja colores se realiza sin ningún error.

Actividades Acceder a la tab de tarjetas, seleccionar varios colores y comprobar que se produce el mismo resultado que en el borrado pero permaneciendo en el sistema todos los datos dados de baja.

Técnicas y herramientas

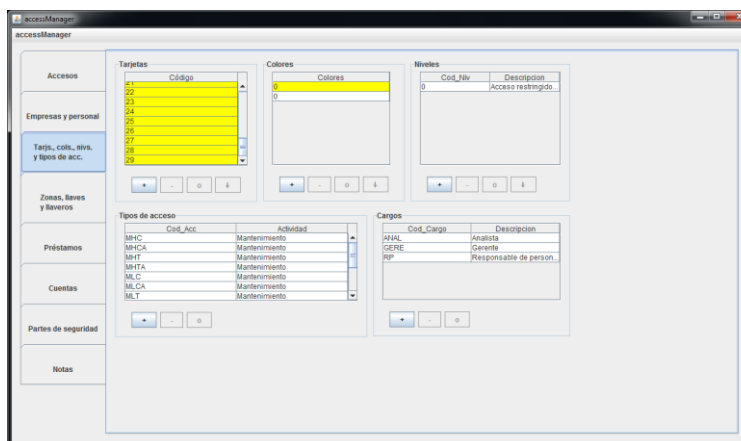
- Aplicación accessManager.
- Base de datos con población de prueba.
- Utilidad phpMyAdmin.

Datos de entrada Damos de baja el nivel 1:



Salida esperada Todas las tarjetas azules deberán desaparecer de su tabla, permaneciendo en la base de datos con la misma fecha de baja que el color. Lo mismo sucederá con el nivel 1.

Salida observada La esperada:



Cod_Tarj	F_Alta_T	F_Baja_T	Cod_Color	F_Alta_C	DNI
Número de tarjeta	Fecha y hora de inserción de la tarjeta	Fecha y hora de baja de la tarjeta	Color de la tarjeta	Fecha y hora de inserción del color	DNI de la persona que la tiene en su posesión
30	2010-01-01 00:00:01	2010-05-20 19:24:48	-16776961	2010-01-01 00:00:01	0000001A
31	2010-01-01 00:00:01	2010-05-20 19:24:48	-16776961	2010-01-01 00:00:01	00000002A
32	2010-01-01 00:00:01	2010-05-20 19:24:48	-16776961	2010-01-01 00:00:01	00000003A

Cod_Color	F_Alta_C	F_Baja_C	Cod_Niv	F_Alta_N
Color de la tarjeta	Fecha y hora de inserción del color	Fecha y hora de baja del color	Nivel de acceso	Fecha y hora de inserción del nivel
-16776961	2010-01-01 00:00:01	2010-05-20 19:24:48	1	2010-01-01 00:00:01
-256	2010-01-01 00:00:01	NULL	0	2010-01-01 00:00:01
-1	2010-01-01 00:00:01	NULL	0	2010-01-01 00:00:01



Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y hora de inserción del nivel	F_Baja_N	Fecha y hora de baja del nivel	Descripcion	Descripción del nivel de acceso
0		2010-01-01 00:00:01		NULL		Acceso restringido: Zonas comunes	
1		2010-01-01 00:00:01		2010-05-20 19:24:48		Acceso total: Zonas comunes + Host	

**Evaluación/
conclusión**

El caso de uso '48 - Dar de baja color' ha sido realizado satisfactoriamente.

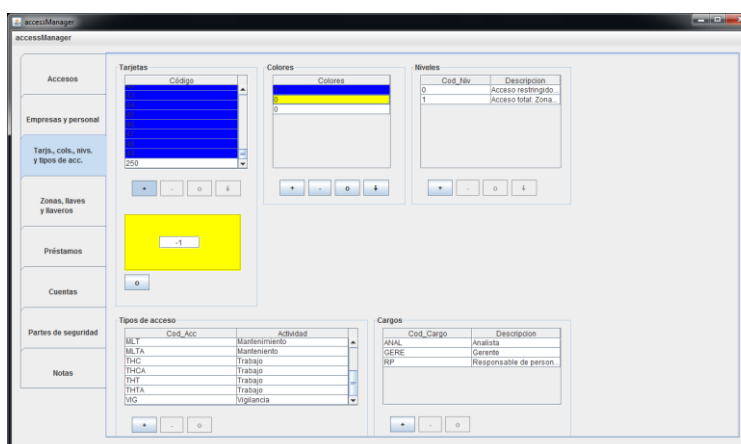
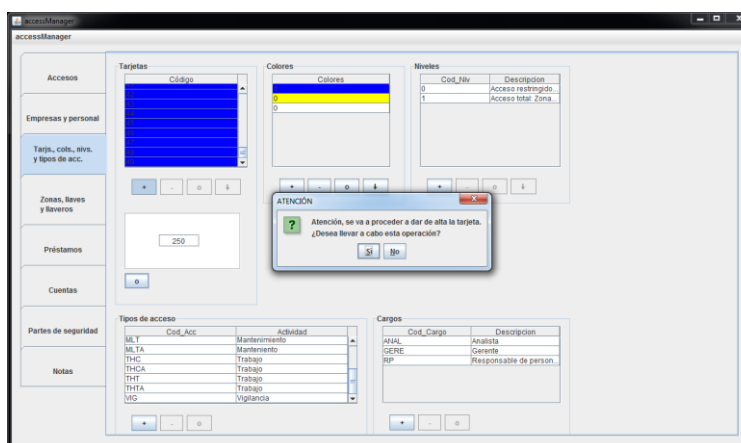
Caso de prueba para el caso de uso '48 - Dar de baja color'



Caso de prueba	5.10 - Caso de uso 51: Añadir tarjeta
Objetivo	Verificar que es el administrador puede introducir sin problemas cualquier tarjeta nueva que se proponga, a menos que su código no cumpla con el formato establecido.
Actividades	Dirigirnos a la sección de tarjetas y probar a añadir una tarjeta válida y otra errónea, comprobando que el primer caso es posible pero que el segundo es rechazado por el sistema.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos con población de prueba.

Probaremos a insertar las tarjetas '250 – Blanca' y '-1 – Amarilla':

Datos de entrada

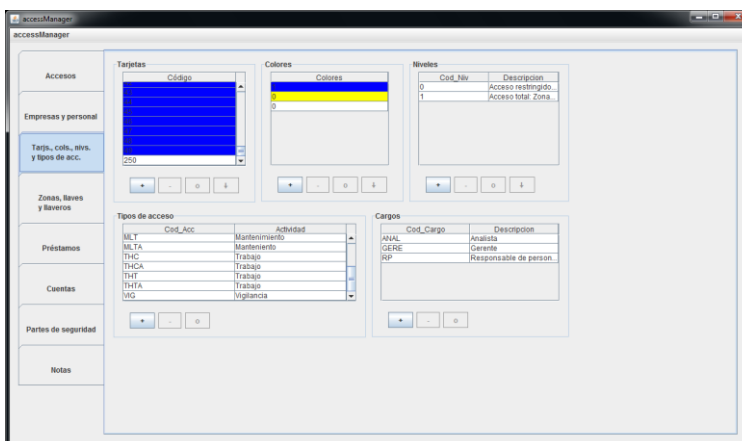


Salida esperada

La primera quedará registrada en el sistema, desencadenando la actualización de la tabla de tarjetas, mientras que al intentar introducir la segunda obtendremos un mensaje de error y el sistema no realizará acción alguna.

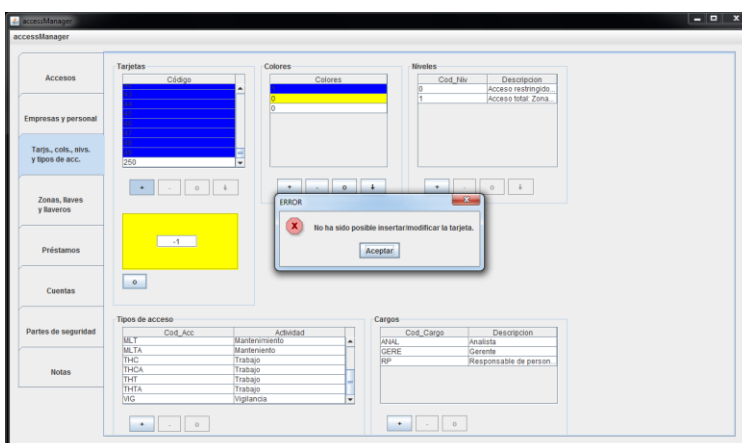
La esperada:

- En el primer caso:



Salida observada

- En el segundo:

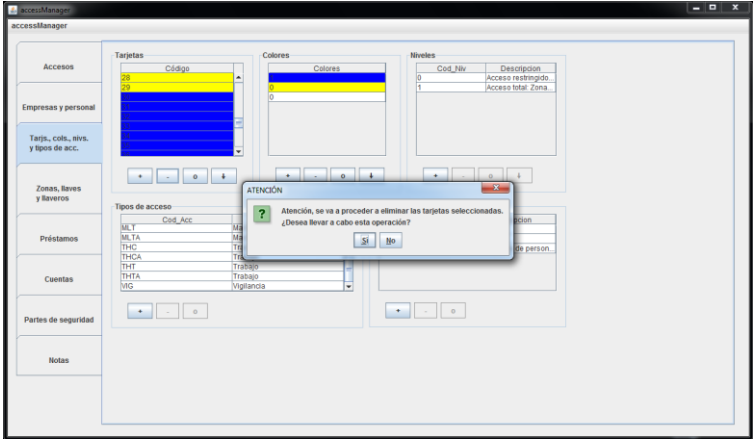
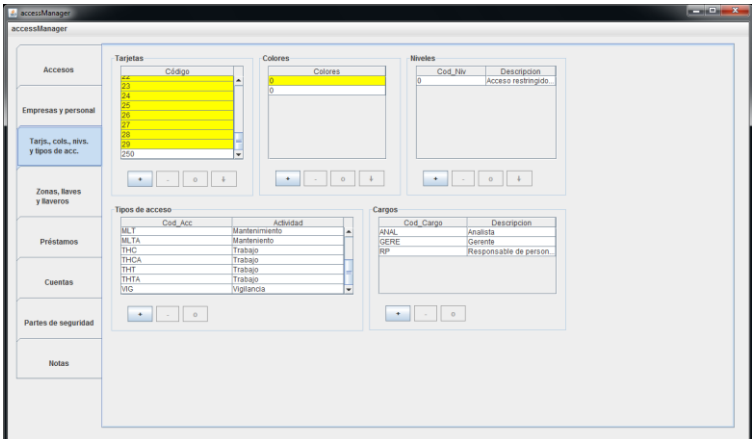


Agregar que la generación del número de tarjeta por defecto es correcta.

Evaluación/
conclusión

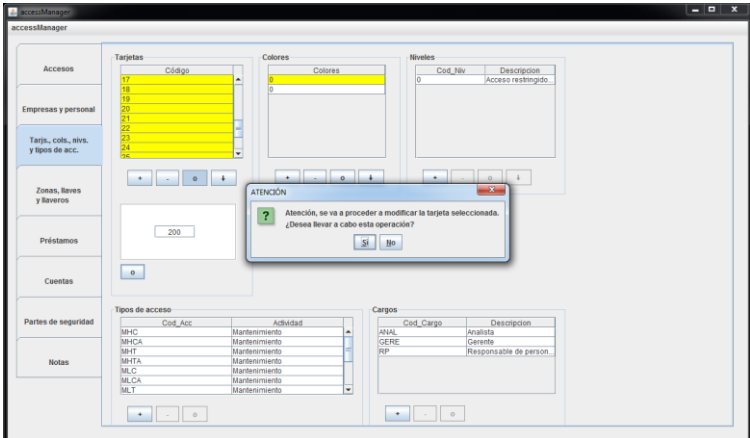
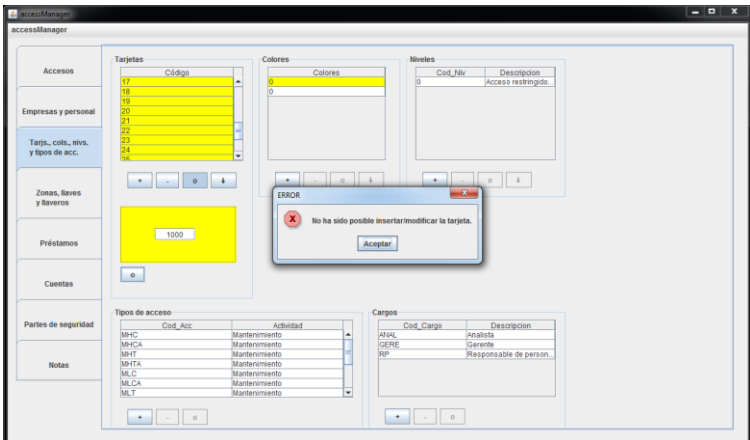
El caso de uso '51 - Añadir tarjeta' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '51 - Añadir tarjeta'

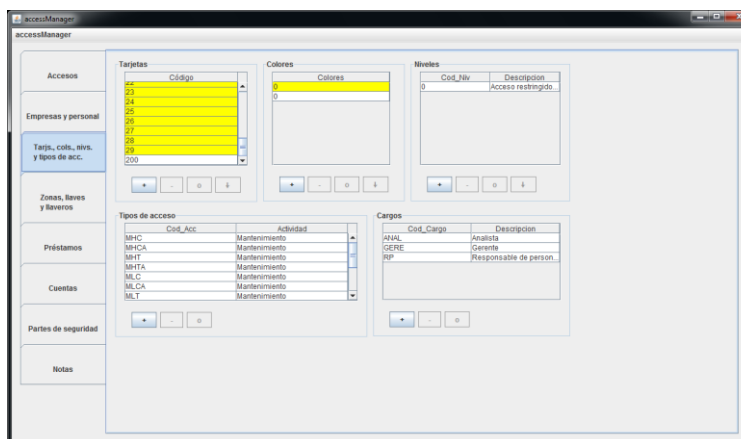
Caso de prueba	5.11 - Caso de uso 52: Borrar tarjeta
Objetivo	Probar que no se producirá ninguna situación inesperada durante el proceso de borrado de tarjetas.
Actividades	Arrancar la aplicación, ir a la pestaña de tarjetas y eliminar algunas de ellas, comprobando que el sistema siempre responde como debe.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior.
Datos de entrada	<p>Eliminaremos todas las tarjetas azules:</p> 
Salida esperada	El borrado de todas las tarjetas azules hará que el color azul se quede sin tarjetas y que el nivel 1 se quede sin colores, con lo cual deberían actualizarse las tres tablas y no existir ni tarjetas azules, ni el color azul, ni el nivel 1.
Salida observada	<p>La esperada:</p> 
Evaluación/conclusión	El caso de uso '52 - Borrar tarjeta' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '52 - Borrar tarjeta'



Caso de prueba	5.12 - Caso de uso 53: Modificar tarjeta
Objetivo	Comprobar que toda la implementación que lleva a cabo la actualización de tarjetas es correcta.
Actividades	Situarnos en la pestaña de tarjetas, seleccionar una y modificar sus datos, comprobando que todo sucede conforme a lo previsto.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos con población de prueba.
Datos de entrada	<p>Modificamos la tarjeta '20 – Amarilla', primero asignándole valores que provoquen el fallo ('1000 – Blanco') y después otros correctos ('200 – Blanco'):</p> 
Salida esperada	La primera inserción será imposible, pero la segunda sí que debería tener lugar.
Salida observada	<p>La esperada:</p> <ul style="list-style-type: none">• En el primer caso: 

- En el segundo:

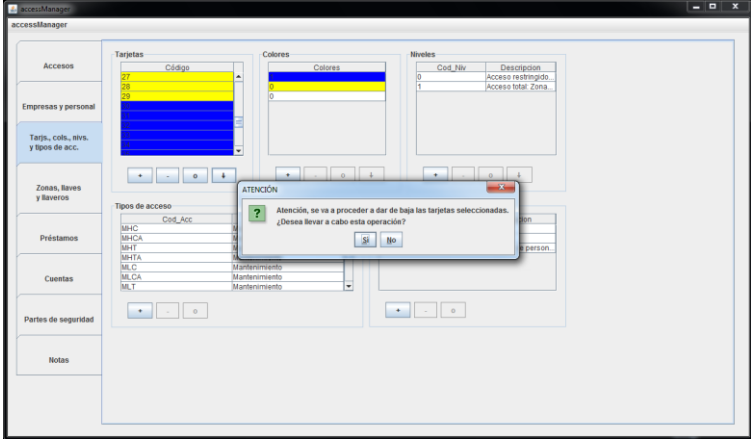
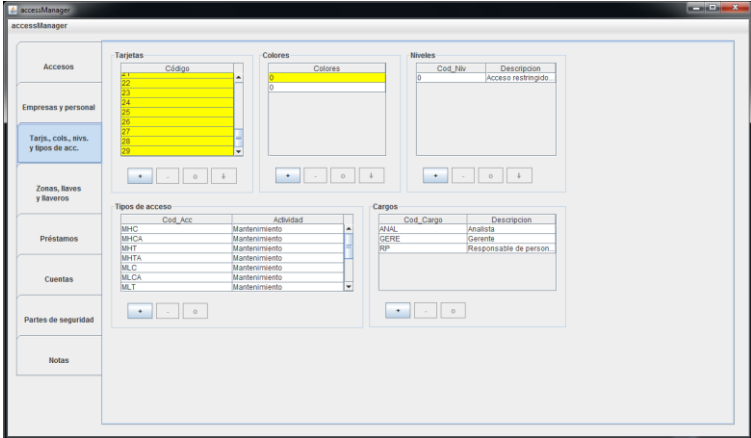


**Evaluación/
conclusión**

El caso de uso '53 - Modificar tarjeta' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '53 - Modificar tarjeta'



Caso de prueba	5.13 - Caso de uso 54: Dar de baja tarjeta																																																																
Objetivo	Demostrar que el caso de uso 54 es llevado a cabo por el sistema correctamente.																																																																
Actividades	Arrancar la aplicación e ir a la pestaña de tarjetas, donde daremos de baja un conjunto de tarjetas y observaremos si el sistema responde como se espera de él.																																																																
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. • Utilidad phpMyAdmin. 																																																																
Datos de entrada	<p>Damos de baja todas las tarjetas azules:</p> 																																																																
Salida esperada	La baja de todas las tarjetas azules debería provocar la baja del color azul, y ésta la del nivel 1.																																																																
Salida observada	<p>La esperada:</p>  <table border="1" data-bbox="563 1731 1284 1798"> <thead> <tr> <th>Cod_Niv</th> <th>Nivel de acceso</th> <th>F_Alta_N</th> <th>Fecha y hora de inserción del nivel</th> <th>F_Baja_N</th> <th>Fecha y hora de baja del nivel</th> <th>Descripcion</th> <th>Descripción del nivel de acceso</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Acceso restringido: Zonas comunes</td> <td></td> </tr> <tr> <td>1</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>2010-05-21 09:16:26</td> <td></td> <td>Acceso total: Zonas comunes + Host</td> <td></td> </tr> </tbody> </table> <table border="1" data-bbox="563 1827 1284 1921"> <thead> <tr> <th>Cod_Color</th> <th>Color de la tarjeta</th> <th>F_Alta_C</th> <th>Fecha y hora de inserción del color</th> <th>F_Baja_C</th> <th>Fecha y hora de baja del color</th> <th>Cod_Niv</th> <th>Nivel de acceso</th> <th>F_Alta_N</th> <th>Fecha y hora de inserción del nivel</th> </tr> </thead> <tbody> <tr> <td>-16776961</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>2010-05-21 09:16:26</td> <td></td> <td>1</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> </tr> <tr> <td>-256</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>0</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> </tr> <tr> <td>-1</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>0</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> </tr> </tbody> </table>	Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y hora de inserción del nivel	F_Baja_N	Fecha y hora de baja del nivel	Descripcion	Descripción del nivel de acceso	0		2010-01-01 00:00:01		NULL		Acceso restringido: Zonas comunes		1		2010-01-01 00:00:01		2010-05-21 09:16:26		Acceso total: Zonas comunes + Host		Cod_Color	Color de la tarjeta	F_Alta_C	Fecha y hora de inserción del color	F_Baja_C	Fecha y hora de baja del color	Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y hora de inserción del nivel	-16776961		2010-01-01 00:00:01		2010-05-21 09:16:26		1		2010-01-01 00:00:01		-256		2010-01-01 00:00:01		NULL		0		2010-01-01 00:00:01		-1		2010-01-01 00:00:01		NULL		0		2010-01-01 00:00:01	
Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y hora de inserción del nivel	F_Baja_N	Fecha y hora de baja del nivel	Descripcion	Descripción del nivel de acceso																																																										
0		2010-01-01 00:00:01		NULL		Acceso restringido: Zonas comunes																																																											
1		2010-01-01 00:00:01		2010-05-21 09:16:26		Acceso total: Zonas comunes + Host																																																											
Cod_Color	Color de la tarjeta	F_Alta_C	Fecha y hora de inserción del color	F_Baja_C	Fecha y hora de baja del color	Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y hora de inserción del nivel																																																								
-16776961		2010-01-01 00:00:01		2010-05-21 09:16:26		1		2010-01-01 00:00:01																																																									
-256		2010-01-01 00:00:01		NULL		0		2010-01-01 00:00:01																																																									
-1		2010-01-01 00:00:01		NULL		0		2010-01-01 00:00:01																																																									



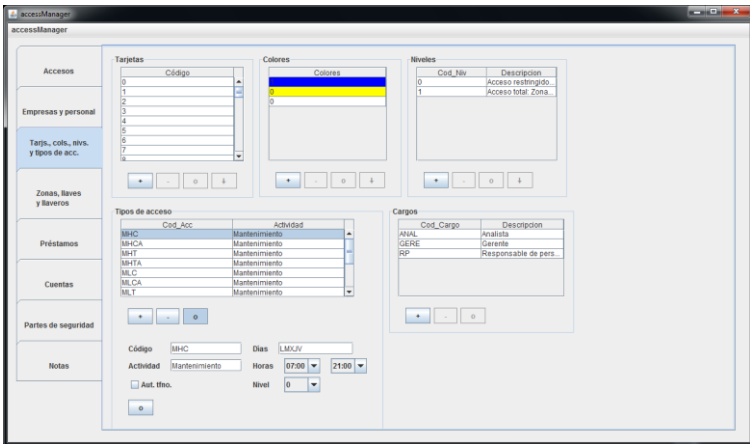
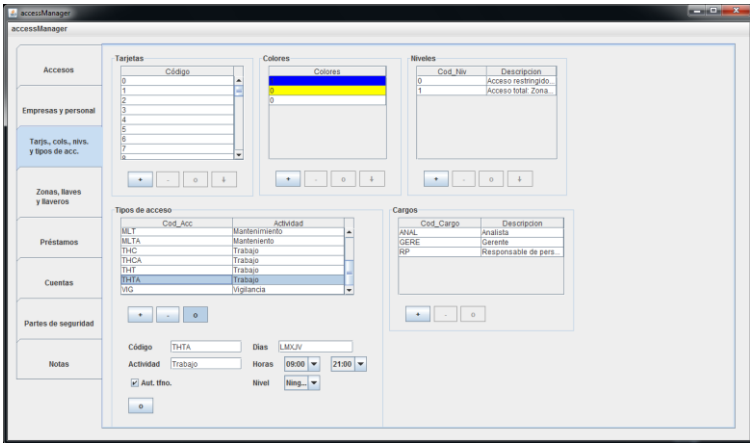
Cod_Tarj	F_Alta_T	F_Baja_T	Cod_Color	F_Alta_C	DNI
Número de tarjeta	Fecha y hora de inserción de la tarjeta	Fecha y hora de baja de la tarjeta	Color de la tarjeta	Fecha y hora de inserción del color	DNI de la persona que la tiene en su posesión
30	2010-01-01 00:00:01	2010-05-21 09:16:26	-16776961	2010-01-01 00:00:01	0000001A
31	2010-01-01 00:00:01	2010-05-21 09:16:26	-16776961	2010-01-01 00:00:01	0000002A
32	2010-01-01 00:00:01	2010-05-21 09:16:26	-16776961	2010-01-01 00:00:01	0000003A
33	2010-01-01 00:00:01	2010-05-21 09:16:26	-16776961	2010-01-01 00:00:01	NULL

...

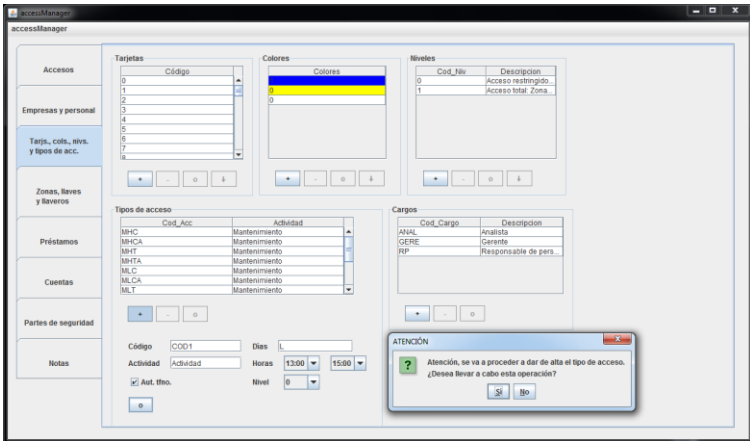
**Evaluación/
conclusión** **El caso de uso '53 - Dar de baja tarjeta' ha sido realizado satisfactoriamente.**

Caso de prueba para el caso de uso '53 - Dar de baja tarjeta'



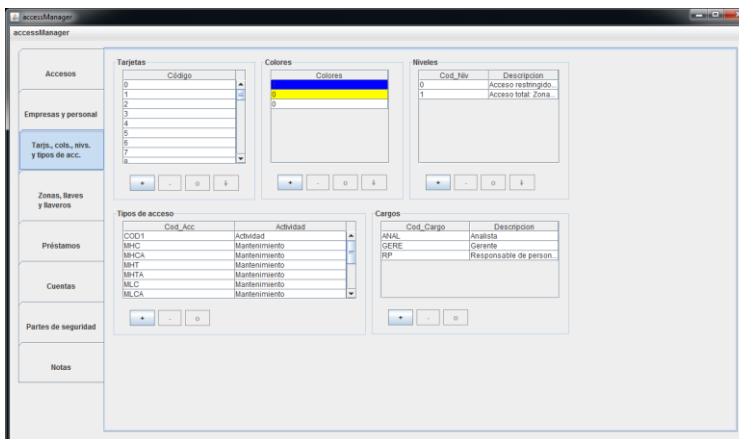
Caso de prueba	5.14 - Caso de uso 56: Visualizar tipo de acceso																
Objetivo	Chequear que el usuario puede seleccionar un tipo de acceso de la tabla y visualizar todos los datos que lo componen.																
Actividades	Ir a la tab de tarjetas, seleccionar varios tipos de acceso y comprobar que se visualizan correctamente.																
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. • Utilidad phpMyAdmin. 																
Datos de entrada	Probaremos a visualizar un tipo de acceso usual y otro que tenga un nivel de acceso nulo, por ejemplo, el 'MHC' y el 'THTA'.																
Salida esperada	<p>Se espera visualizar:</p> <ul style="list-style-type: none"> • 'MHC, Mantenimiento, LMXJV, 07:00, 21:00, No, 0' <table border="1"> <tr> <td>MHC</td> <td>Mantenimiento</td> <td>LMXJV</td> <td>07:00:00</td> <td>21:00:00</td> <td>0</td> <td>0</td> <td>2010-01-01 00:00:01</td> </tr> </table> <ul style="list-style-type: none"> • 'THTA, Trabajo, LMXJV, 09:00, 21:00, Sí, Ninguno' <table border="1"> <tr> <td>THTA</td> <td>Trabajo</td> <td>LMXJV</td> <td>09:00:00</td> <td>21:00:00</td> <td>1</td> <td>NULL</td> <td>NULL</td> </tr> </table>	MHC	Mantenimiento	LMXJV	07:00:00	21:00:00	0	0	2010-01-01 00:00:01	THTA	Trabajo	LMXJV	09:00:00	21:00:00	1	NULL	NULL
MHC	Mantenimiento	LMXJV	07:00:00	21:00:00	0	0	2010-01-01 00:00:01										
THTA	Trabajo	LMXJV	09:00:00	21:00:00	1	NULL	NULL										
Salida observada	<p>La esperada:</p>  																
Evaluación/ conclusión	<p>El caso de uso '56 - Visualizar tipo de acceso' ha sido realizado satisfactoriamente.</p> <p><i>Caso de prueba para el caso de uso '56 - Visualizar tipo de acceso'</i></p>																



Caso de prueba	5.15 - Caso de uso 57: Añadir tipo de acceso
Objetivo	Comprobar que el usuario puede añadir tipos de acceso en caso de ser necesario y sin que ello conlleve alguna situación indeseada.
Actividades	Arrancar la aplicación, acceder al tab de tarjetas e introducir tipos de acceso válidos y erróneos, verificando que el sistema reacciona correctamente.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos con población de prueba.
Datos de entrada	<p>Diseñamos un acceso válido y dos erróneos:</p> <ul style="list-style-type: none">• 'COD1, Actividad, L, 13:00, 15:00, Sí, 1'  <ul style="list-style-type: none">• 'MHC, Actividad, L, 13:00, 15:00, Sí, 1'. (Código existente, además probamos que a introducir en orden inverso las horas y ver que el sistema las corrige)• 'COD2, Actividad, N, 13:00, 15:00, Sí,1'. (Día no contenido en LMXJVSD)• 'COD2, Actividad demasiado larga para caber en este campo, L, 13:00, 15:00, Sí, 1'. (Actividad con longitud excesiva)
Salida esperada	El primer nivel deberá insertarse, pero el otro no.

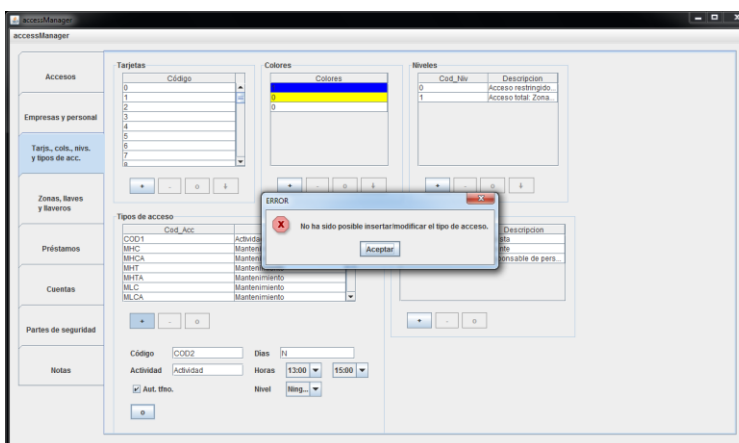
La esperada:

- En el primer caso tendríamos...



Salida observada

- ... mientras que en el resto:



Añadir que el comprobador de fechas funciona de forma correcta.

Evaluación/
conclusión

El caso de uso '57 - Añadir tipo de acceso' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '57 - Añadir tipo de acceso'



Caso de prueba 5.16 - Caso de uso 58: Borrar tipo de acceso

Objetivo Verificar que es posible realizar el borrado de tipos de acceso sin que el sistema llegue a una situación de error.

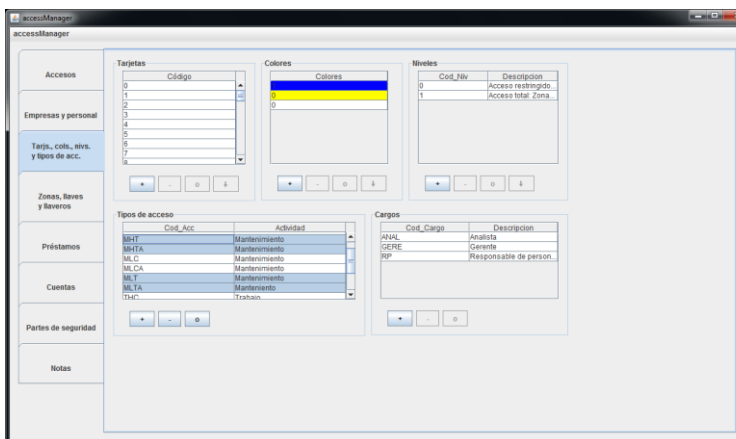
Actividades Arrancar la aplicación e ir a la pestaña de tarjetas, donde seleccionaremos varios tipos de acceso y probaremos a eliminarlos.

- Técnicas y herramientas**
- Aplicación accessManager.
 - Base de datos con población de prueba.
 - Utilidad phpMyAdmin.

Borramos los siguientes tipos de acceso:

Cod_Acc	Actividad	Dias	Hora_inicio	Hora_fin	Aut_Tel	Cod_Niv	F_Alta_N
VIG	Vigilancia	LMXJVS	00:00:00	23:59:59		0	2010-01-01 00:00:01
THT	Trabajo	LMXJV	09:00:00	21:00:00		0	2010-01-01 00:00:01
MLTA	Mantenimiento	LMXJVS	00:00:00	23:59:59		1	2010-01-01 00:00:01
MLT	Mantenimiento	LMXJVS	00:00:00	23:59:59		0	2010-01-01 00:00:01
MHTA	Mantenimiento	LMXJV	07:00:00	21:00:00		1	2010-01-01 00:00:01
MHT	Mantenimiento	LMXJV	07:00:00	21:00:00		0	2010-01-01 00:00:01

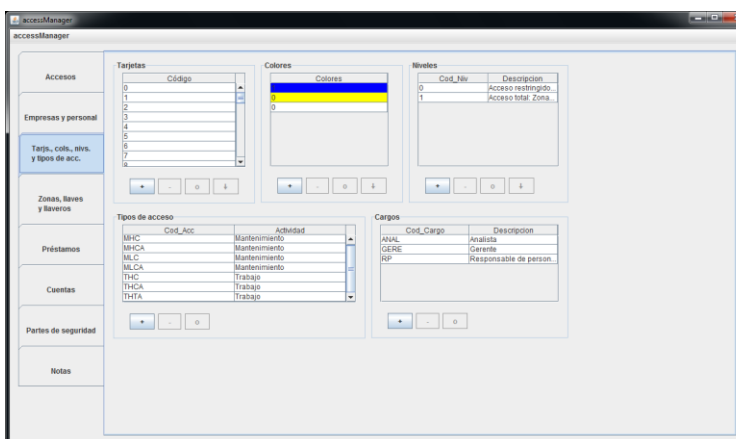
Datos de entrada



Salida esperada La tabla de accesos se actualizará automáticamente y no dispondremos de ningún acceso con nivel 1.

La esperada:

Salida observada

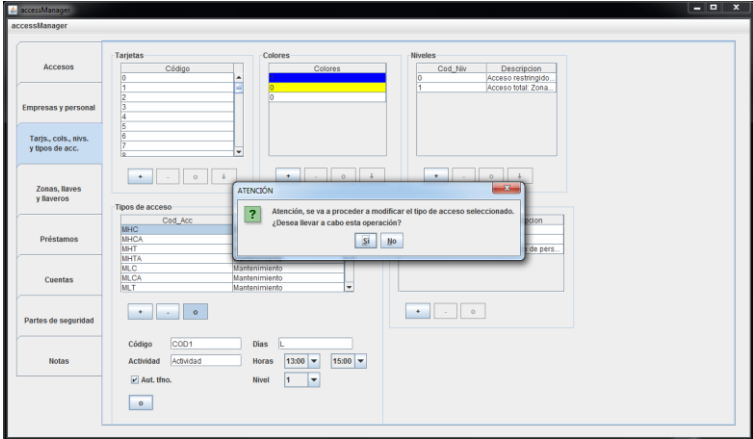
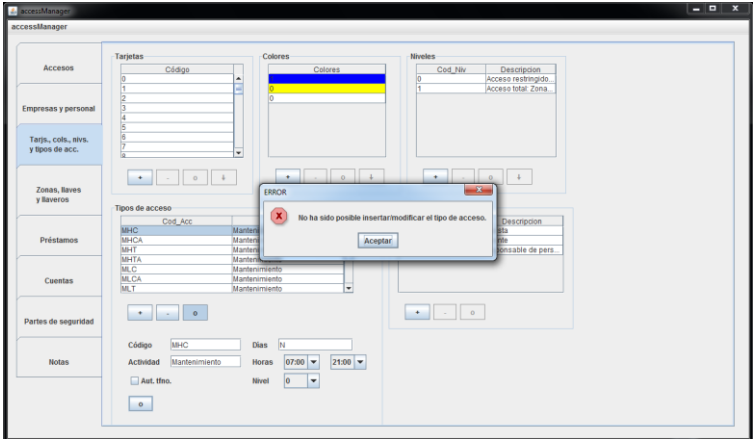


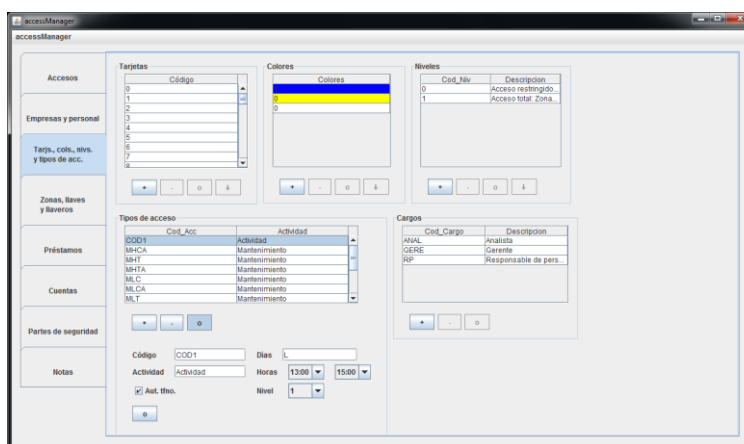
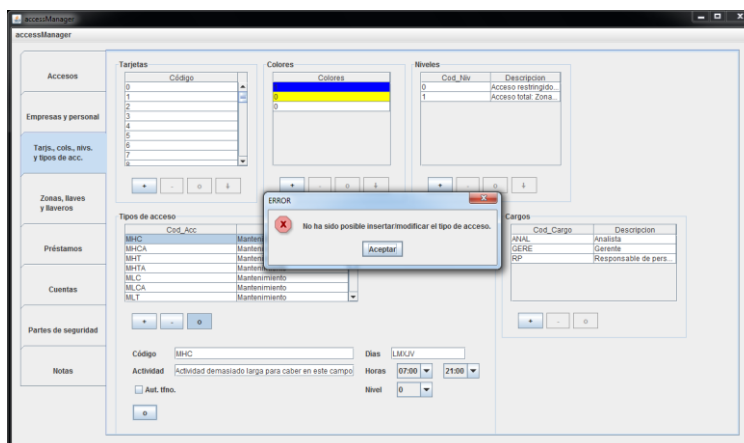


Cod_Acc	Actividad	Dias	Hora_Inic	Hora_Fin	Aut_Tel	Cod_Niv	F_Alta_N
<small>▲ Código de acceso</small>	<small>Actividad para la cual se concede el acceso</small>	<small>Días permitidos de acceso</small>	<small>Hora de inicio a partir de la cual está permitido el acceso</small>	<small>hora hasta la que está permitido el acceso al centro</small>	<small>Denota si es necesario el aviso telefónico a la persona responsable del departamento cada vez que un empleado de ésta accede</small>	<small>Nivel de acceso al centro</small>	<small>Fecha y hora de inserción del nivel</small>
MHC	Mantenimiento	LMXJV	07:00:00	21:00:00	0	0	2010-01-01 00:00:01
MHCA	Mantenimiento	LMXJV	07:00:00	21:00:00	1	0	2010-01-01 00:00:01
MLC	Mantenimiento	LMXJVSD	00:00:00	23:59:59	0	0	2010-01-01 00:00:01
MLCA	Mantenimiento	LMXJVSD	00:00:00	23:59:59	1	0	2010-01-01 00:00:01
THC	Trabajo	LMXJV	09:00:00	21:00:00	0	0	2010-01-01 00:00:01
THCA	Trabajo	LMXJV	09:00:00	21:00:00	1	0	2010-01-01 00:00:01
THTA	Trabajo	LMXJV	09:00:00	21:00:00	1	NULL	NULL

Evaluación/ conclusión **El caso de uso '58 - Borrar tipo de acceso' ha sido realizado satisfactoriamente.**

Caso de prueba para el caso de uso '58 - Borrar tipo de acceso'

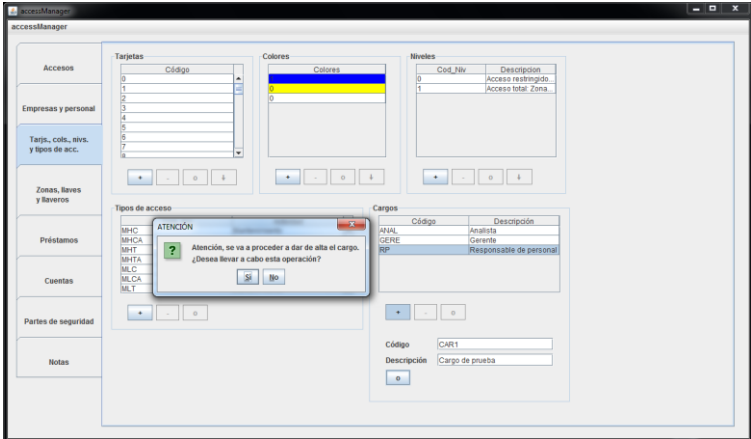
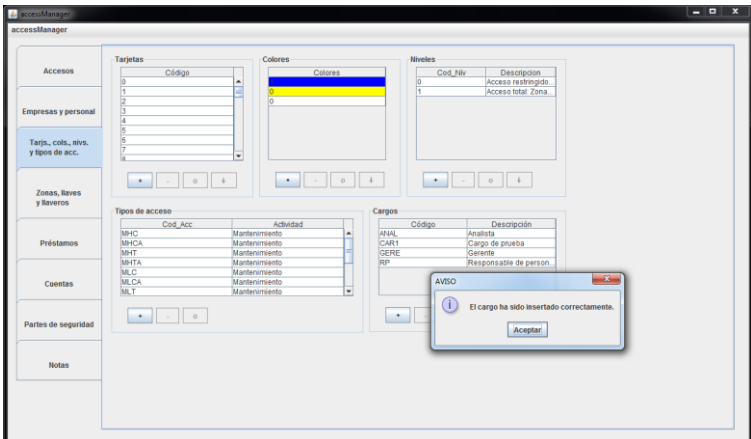
Caso de prueba	5.17 - Caso de uso 59: Modificar tipo de acceso
Objetivo	Probar que puede llevarse a cabo la modificación de accesos según lo establecido, no llegando nunca a ninguna situación del sistema inconsistente.
Actividades	Ir a la pestaña de tarjetas, seleccionar un tipo de acceso, modificarlo de diversas formas (permitidas y prohibidas) y comprobar que el sistema responde como se espera.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba.
Datos de entrada	<p>Modificaremos 'MHC' cambiando:</p> <ul style="list-style-type: none"> • Día: 'LMXJV' parará a ser 'N'. (Día no contenido en LMXJVSD) • Actividad: 'Mantenimiento' pasará a ser 'Actividad demasiado larga para caber en este campo'. (Actividad con longitud excesiva) • Todos los campos: 'COD1, Actividad, L, 13:00, 15:00, Sí, 1'. 
Salida esperada	Los dos primeros casos no deberían surtir efecto, mientras que en el tercero el tipo de acceso 'MHC' debería dar paso al 'COD1'.
Salida observada	<p>La esperada en los tres casos:</p> 



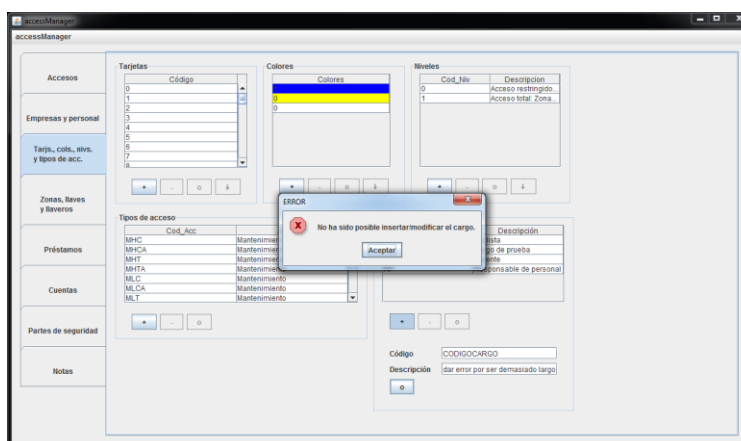
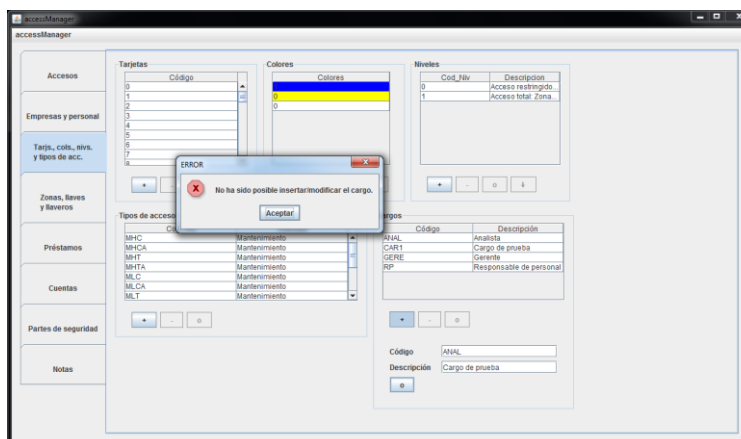
Evaluación/
conclusión

El caso de uso '58 - Borrar tipo de acceso' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '58 - Borrar tipo de acceso'

Caso de prueba	5.18 - Caso de uso 62: Añadir cargo
Objetivo	Demostrar que el administrador puede insertar cargos siempre que éste lo haga conforme a las condiciones establecidas.
Actividades	Poner en marcha la aplicación, acceder a la pestaña de tarjetas y probar a insertar diversos cargos, válidos y erróneos, comprobando que los primeros se insertan pero no los segundos.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba.
Datos de entrada	<p>Probaremos a insertar:</p> <ul style="list-style-type: none"> • Cargo válido: 'CAR1', 'Cargo de prueba'.  <ul style="list-style-type: none"> • Cargos erróneos: 'ANAL', 'Cargo de prueba' (código existente) y 'CODIGOCARGO', 'Descripción del cargo que debe dar error por ser demasiado largo' (formato de los campos erróneo).
Salida esperada	El primer cargo pasará a formar parte del sistema, mientras que los dos siguientes provocarán un mensaje de error y no serán incorporados a la base de datos.
Salida observada	<p>La esperada:</p> <ul style="list-style-type: none"> • En el primer caso: 

- En los dos siguientes:

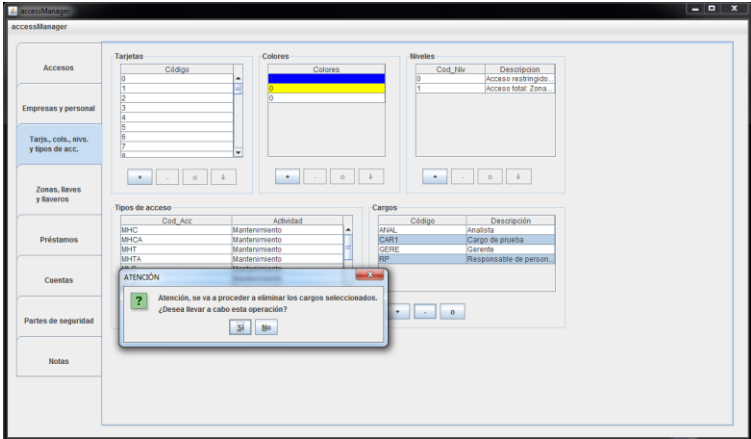
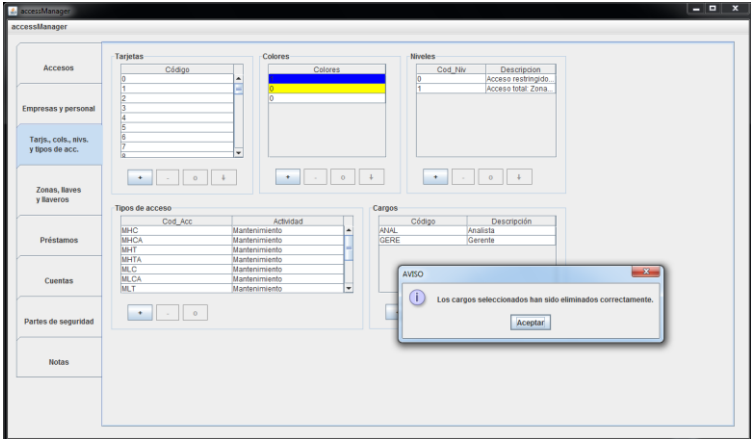


**Evaluación/
conclusión**

El caso de uso '62 - Añadir cargo' ha sido realizado satisfactoriamente.

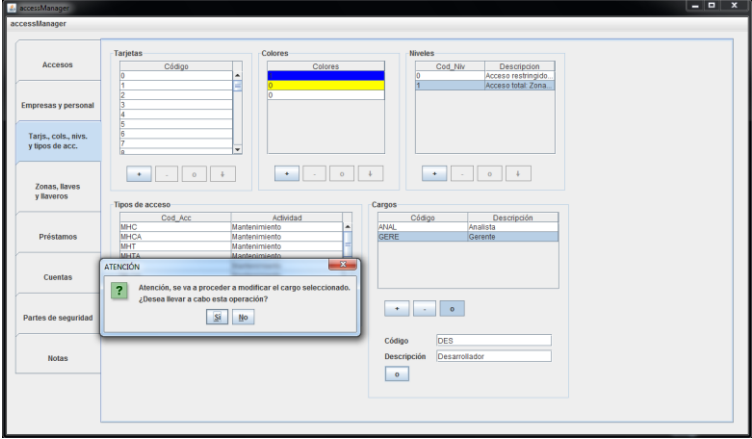
Caso de prueba para el caso de uso '62 - Añadir cargo'



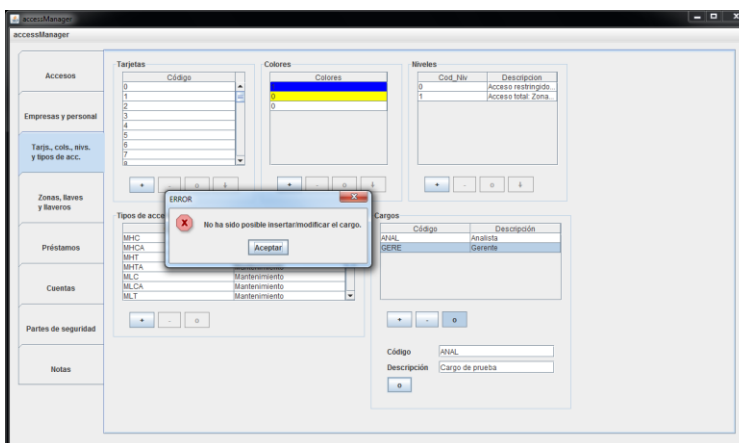
Caso de prueba	5.19 - Caso de uso 63: Borrar cargo																																																								
Objetivo	Verificar que el borrado de cargos se ejecuta de acuerdo a como tenemos planeado.																																																								
Actividades	Acceder a la tab de tarjetas, seleccionar varios cargos y borrarlos comprobando que dejan de formar parte del sistema, así como todo el personal que lo desempeña.																																																								
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior. • Utilidad phpMyAdmin. 																																																								
Datos de entrada	<p>Eliminaremos los cargos 'CAR1' y 'RP':</p>  <table border="1" data-bbox="563 1115 1281 1261"> <thead> <tr> <th>DNI</th> <th>DNI</th> <th>Cod_Emp</th> <th>Código de empresa para la que trabaja</th> <th>Cod_Dep</th> <th>Código de departamento para el que trabaja</th> <th>Cod_Cargo</th> <th>Cargo desempeñado</th> </tr> </thead> <tbody> <tr> <td>00001000A</td> <td>CIS</td> <td></td> <td></td> <td>VA</td> <td></td> <td></td> <td>RP</td> </tr> <tr> <td>00000198A</td> <td>IBM</td> <td></td> <td></td> <td>SEPR</td> <td></td> <td></td> <td>RP</td> </tr> <tr> <td>00000198A</td> <td>IBM</td> <td></td> <td></td> <td>COIN</td> <td></td> <td></td> <td>RP</td> </tr> <tr> <td>00000198A</td> <td>IBM</td> <td></td> <td></td> <td>ALDA</td> <td></td> <td></td> <td>RP</td> </tr> <tr> <td>00000196A</td> <td>IBM</td> <td></td> <td></td> <td>MIDD</td> <td></td> <td></td> <td>RP</td> </tr> <tr> <td>00000195A</td> <td>ELKO</td> <td></td> <td></td> <td>EMER</td> <td></td> <td></td> <td>RP</td> </tr> </tbody> </table>	DNI	DNI	Cod_Emp	Código de empresa para la que trabaja	Cod_Dep	Código de departamento para el que trabaja	Cod_Cargo	Cargo desempeñado	00001000A	CIS			VA			RP	00000198A	IBM			SEPR			RP	00000198A	IBM			COIN			RP	00000198A	IBM			ALDA			RP	00000196A	IBM			MIDD			RP	00000195A	ELKO			EMER			RP
DNI	DNI	Cod_Emp	Código de empresa para la que trabaja	Cod_Dep	Código de departamento para el que trabaja	Cod_Cargo	Cargo desempeñado																																																		
00001000A	CIS			VA			RP																																																		
00000198A	IBM			SEPR			RP																																																		
00000198A	IBM			COIN			RP																																																		
00000198A	IBM			ALDA			RP																																																		
00000196A	IBM			MIDD			RP																																																		
00000195A	ELKO			EMER			RP																																																		
Salida esperada	Ambos códigos dejarán de existir en el sistema, así como todas aquellas personas que desempeñen el cargo de responsable de personal.																																																								
Salida observada	<p>La esperada:</p> 																																																								



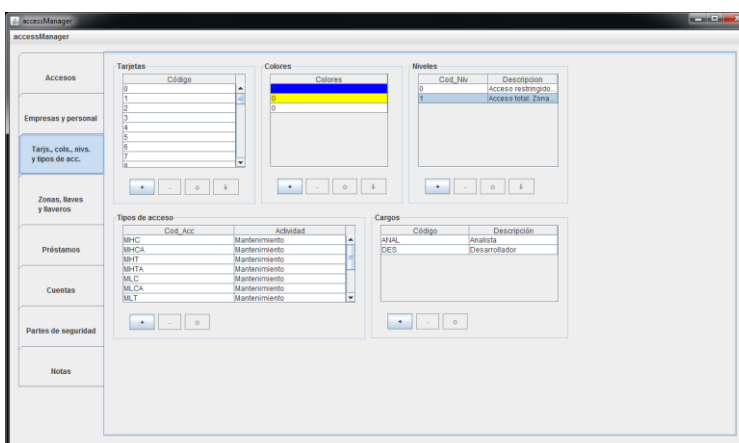
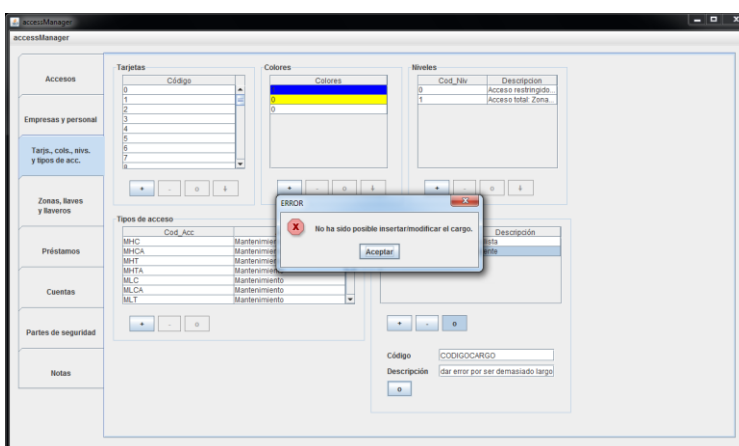
	<table border="1"><thead><tr><th>DNI</th><th>Cod_Emp</th><th>Cod_Dep</th><th>Cod_Cargo</th></tr></thead><tbody><tr><td>00000170A</td><td>ELEC</td><td>CONS</td><td>ANAL</td></tr><tr><td>00000011A</td><td>CONA</td><td>CAU</td><td>GERE</td></tr></tbody></table>	DNI	Cod_Emp	Cod_Dep	Cod_Cargo	00000170A	ELEC	CONS	ANAL	00000011A	CONA	CAU	GERE
DNI	Cod_Emp	Cod_Dep	Cod_Cargo										
00000170A	ELEC	CONS	ANAL										
00000011A	CONA	CAU	GERE										
Evaluación/ conclusión	El caso de uso '63 - Borrar cargo' ha sido realizado satisfactoriamente. <i>Caso de prueba para el caso de uso '63 - Borrar cargo'</i>												

Caso de prueba	5.20 - Caso de uso 64: Modificar cargo
Objetivo	Chequear que el usuario puede modificar un cargo a su gusto siempre que introduzca datos con un formato correcto y que el nuevo cargo no exista previamente.
Actividades	Arrancar la aplicación, ir al tab de tarjetas y modificar un cargo probando casos similares a los probados con la inserción, comprobando que todo es correcto.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del anterior caso de prueba.
Datos de entrada	<p>Probaremos a actualizar el cargo 'GERE' con los siguientes:</p> <ul style="list-style-type: none"> • Cargos erróneos: <ul style="list-style-type: none"> 'ANAL, Cargo de prueba'. (Código existente) 'CODIGOCARGO, Descripción del cargo que debe dar error por ser demasiado largo'. (Formato de los campos erróneo) • Cargo válido: <ul style="list-style-type: none"> 'DES, Desarrollador' 
Salida esperada	Con los dos primeros casos únicamente deberíamos obtener mensajes de error, mientras que con el último se actualizará la tabla de cargos actualizando el cargo 'GERE' por 'DES' y todas las personas que desempeñen el cargo de gerente.

La esperada:



Salida observada



DNI	CNI	Cod_Emp	Código de empresa para la que trabaja	Cod_Dep	Código de departamento para el que trabaja	Cod_Cargo	Cargo desempeñado
00000170A	ELEC			CONS		ANAL	
00000011A	CONA			CAU		DES	

Evaluación/
conclusión

El caso de uso '64 - Modificar cargo' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '64 - Modificar cargo'



6.2. De integración

La integración del sistema sigue siendo la misma, ya que no desarrollamos componentes nuevos, sino que desarrollamos los ya existentes. Los casos de uso han sido probados con éxito, con que podemos decir que la integración de GUI, API y base de datos sigue siendo correcta.

6.3. De sistema

Veintiocho son los casos de uso que nos propusimos realizar en esta iteración y veintiocho son los que hemos implementado. Todos los casos de uso han sido incorporados, consiguiendo un funcionamiento robusto y una más que útil documentación, con que en este aspecto podemos estar muy contentos.

7. Evaluación de la iteración

Tercera pestaña desarrolla que dará la posibilidad al administrador de mantener actualizados la mitad de los elementos que entran en juego en el proceso de accesos (las tarjetas), así como los cargos de los contactos con los que Informática Navarra mantendrá relación.

Estamos satisfechos con el trabajo desarrollado, ya que hemos conseguido ofrecer al usuario mucha funcionalidad de una manera fácil, sencilla y al alcance de la mano, pero antes de nada nos espera la reunión con el cliente y ver qué le parece la aplicación, si propone mejoras,...

8. Despliegue

Este documento junto con el prototipo desarrollado hasta el momento son piezas únicas e indispensables a llevar a la reunión con el cliente. Ésta tiene lugar el 10 de marzo, tal y como se planeó en un principio, y el cliente nos recibe ansioso por ver los resultados obtenidos.

La actividad de despliegue comienza explicándole qué trabajo hemos realizado y probando con él todos los casos de uso nuevos que incorpora este nuevo prototipo. Se muestra conforme con el software desarrollado pero no ve coherente colocar el panel de edición de cargos en la pestaña recientemente desarrollada, ya que mezcla elementos que forman parte de los accesos con los cargos que desempeñan los contactos. Le reconocemos que sus quejas son razonables, pero que incorporar este panel a la pestaña 'Empresas y personal' acabaría por sobrecargarla en exceso. Le proponemos esperar al final del desarrollo para ver dónde colocarlo, lo cual acepta.



Pamplona, a 10 de marzo de 2010.

Héctor Napal Basterrica
Responsable del proyecto

Pedro Martínez Pérez
*Jefe de área de producción de
Informática Navarra S.A.*



accessManager
Desarrollo de un sistema de control de accesos a un centro

Iteración C6: Zonas, llaves y llaveros

Fecha de entrega: 17 de marzo de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Estructura del documento

2. Estado del proyecto

3. Análisis

- 3.1. Planificación de la iteración
- 3.2. Análisis de los casos de uso y escenarios

4. Diseño

- 4.1. Caso de uso 65 y 66: Visualizar zonas del centro y visualizar zona del centro
- 4.2. Caso de uso 67: Añadir zona del centro
- 4.3. Caso de uso 68: Borrar zona del centro
- 4.4. Caso de uso 69: Modificar zona del centro
- 4.5. Caso de uso 70: Visualizar llaves
- 4.6. Caso de uso 71: Visualizar llave
- 4.7. Caso de uso 72: Añadir llave
- 4.8. Caso de uso 73: Borrar llave
- 4.9. Caso de uso 74: Dar de baja llave
- 4.10. Caso de uso 75: Modificar llave
- 4.11. Caso de uso 77: Visualizar llaveros
- 4.12. Caso de uso 78: Visualizar llavero
- 4.13. Caso de uso 79: Añadir llavero
- 4.14. Caso de uso 80: Borrar llavero
- 4.15. Caso de uso 81: Dar de baja llavero
- 4.16. Caso de uso 82: Modificar llavero
- 4.17. Caso de uso 83: Añadir llave
- 4.18. Caso de uso 84: Borrar llave
- 4.19. Caso de uso 86: Visualizar almacenes de llaves
- 4.20. Caso de uso 87: Visualizar almacén de llaves
- 4.21. Caso de uso 88: Añadir almacén de llaves
- 4.22. Caso de uso 89: Borrar almacén de llaves
- 4.23. Caso de uso 90: Modificar almacén de llaves
- 4.24. Caso de uso 91: Añadir llave
- 4.25. Caso de uso 92: Borrar llave
- 4.26. Caso de uso 93: Modificar copias llave
- 4.27. Caso de uso 94: Añadir llavero
- 4.28. Caso de uso 95: Borrar llavero
- 4.29. Caso de uso 96: Modificar copias llavero
- 4.30. Casos de uso 76, 85 y 97: Listar llaves, llaveros y almacenes



5. Codificación e integración

6. Pruebas

- 6.1. De unidad
- 6.2. De integración
- 6.3. De sistema

7. Evaluación de la iteración

8. Despliegue



1. Introducción

1.1. Propósito

Recoger toda la información de interés que surja a lo largo del desarrollo de toda la funcionalidad encerrada en la pestaña 'Zonas, llaves y llaveros', de modo que pueda servir de documento de consulta para cualquier persona ajena al desarrollo que desee entender el sistema para poder realizar un correcto mantenimiento futuro.

1.2. Audiencia

En un principio se trata de un documento interno a Software Solutions que será portado en la actividad de despliegue por si fuese necesario, pero el cliente no dispondrá de él hasta que le sea transferido el producto software finalizado junto con toda su documentación (iteración T2).

1.3. Estructura del documento

Comenzamos la redacción de este documento exponiendo brevemente en qué punto del proceso de desarrollo nos encontramos, ya que saber qué hemos realizado determinará qué debemos realizar en esta iteración. Conociendo los pasos que deben seguir realizamos la planificación de la misma y analizamos los casos de uso a implementar en ésta a lo largo de la siguiente sección, la 3. Dedicamos todo el cuarto capítulo a la fase de diseño, donde describiremos con detalle el diseño realizado para cada uno de los casos de uso que deberemos implementar en esta iteración. Esta fase da paso a la de codificación e integración; para ella va dirigido el apartado 5.

Con todos los casos de uso codificados procedemos a probarlos uno a uno para asegurar que el funcionamiento del sistema es el deseado. Si el lector desea observar los casos de prueba diseñados y ejecutados puede dirigirse a la sección 6 de este artefacto, donde además dispondrá de una breve mención a las pruebas de integración y de sistema llevadas a cabo.

Acto seguido realizamos una breve evaluación de la iteración, comparando objetivos propuestos y conseguidos, sacando así conclusiones de lo satisfechos que podemos llegar a estar con el trabajo realizado o si puede ser necesario rehacer alguna parte del mismo.

Para finalizar, mencionamos toda la información de interés que hemos extraído de la actividad de despliegue realizada junto con el cliente al final de la iteración.



2. Estado del proyecto

El comienzo de la sexta iteración deja detrás de sí las referentes a accesos, empresas, persona y elementos que entran en juego en el proceso de acceso, como puede ser los tipos de acceso, tarjetas empleadas,... Con esta iteración abordaremos el resto de elementos: zonas, llaves, llaveros y almacenes que los guardan.

Se trata de la última iteración de real complicación, cuya superación dejará el camino allanado hasta el final.

3. Análisis

3.1. Planificación de la iteración

Planificaremos la iteración atendiendo a los criterios empleados en las anteriores y que tan buenos resultados nos han proporcionado:

PLAN DE ITERACIÓN C6: ZONAS, LLAVES Y LLAVEROS			
ASPECTO	DESCRIPCIÓN		
Objetivos	Implementar los casos de uso del 65 al 97, que definen toda la funcionalidad que la aplicación deberá proporcionar en cuanto a lo que zonas, llaves, llaveros y almacenes se refiere.		
Artefactos de entrada	<ul style="list-style-type: none"> • <i>Especificación inicial de los casos de uso</i> • <i>Planificación temporal y costes detallados del proyecto</i> • Base de datos, API y aplicación desarrolladas en las iteraciones anteriores. 		
Artefactos modificados o creados	<ul style="list-style-type: none"> • Diagrama de clases de la API inicial. • Clases Java de la API inicial y de la aplicación. • El presente documento. 		
Calendarización detallada	Análisis + Pruebas	11/03/10	<ul style="list-style-type: none"> • Planificación de la iteración y gestión de riesgos. • Análisis de los casos de uso y escenarios. • Planificación de las pruebas. • Inicio de la documentación.
	Diseño + Pruebas	12/03/10	<ul style="list-style-type: none"> • Diseño de opciones arquitectónicas. • Diseño de pruebas.
		15/03/10	
	Codificación + Pruebas	15/03/10	<ul style="list-style-type: none"> • Codificación e integración de la solución. • Ejecución y evaluación de pruebas. • Evaluación del prototipo. • Preparación de la entrega ejecutable.
17/03/10			
Despliegue	17/03/10	<ul style="list-style-type: none"> • Muestra, prueba y evaluación conjunta con el cliente. • Concreción de los cambios necesarios. • Finalización de toda la documentación. 	

Planificación de la iteración



Nombre de tarea	Duración	Comienzo	Fin	mié 10 mar	vie 12 mar	dom 14 mar	mar 16 mar
Estudio de oportunidad (inception)	7 días	jue 07/01/10	vie 15/01/10	0	0	0	0
Elaboración	16 días	lun 18/01/10	lun 08/02/10				
Construcción	41 días	mar 09/02/10	mar 06/04/10				
Iteración C1: La base de datos	4 días	mar 09/02/10	vie 12/02/10				
Iteración C2: API para la base de datos	5 días	lun 15/02/10	vie 19/02/10				
Iteración C3: Accesos	5 días	lun 22/02/10	vie 26/02/10				
Iteración C4: Empresas y personal	5 días	lun 01/03/10	vie 05/03/10				
Iteración C5: Tarjetas, tipos de acceso y cargos	3 días	lun 08/03/10	mié 10/03/10				
Iteración C6: Zonas, llaves y llaveros	5 días	jue 11/03/10	mié 17/03/10				
Análisis	1 día	jue 11/03/10	jue 11/03/10				
Diseño	1,5 días	vie 12/03/10	lun 15/03/10				
Implementación	2 días	lun 15/03/10	mié 17/03/10				
Pruebas	5 días	jue 11/03/10	mié 17/03/10				
Despliegue	0,5 días	mié 17/03/10	mié 17/03/10				
Iteración C7: Préstamos	2 días	jue 18/03/10	vie 19/03/10				
Iteración C8: Cuentas	2 días	lun 22/03/10	mar 23/03/10				
Iteración C9: Partes de seguridad	3 días	mié 24/03/10	vie 26/03/10				
Iteración C10: Notas	3 días	lun 29/03/10	mié 31/03/10				
Iteración C11: Base de datos	2 días	jue 01/04/10	vie 02/04/10				
Iteración C12: Diseño, personalización y configuración	2 días	lun 05/04/10	mar 06/04/10				
Capacidad operacional	0 días	mar 06/04/10	mar 06/04/10				
Transición	8 días	mié 07/04/10	vie 16/04/10				

Recursos	Temporales	5 días	
	Humanos	1 persona	
	Software	<i>Sistema operativo</i>	Microsoft Windows 7
		<i>Ofimática</i>	Microsoft Word 2010 Adobe Acrobat Professional 8
		<i>Diseño</i>	Visual Paradigm 4.2
		<i>Codificación</i>	NetBeans 6.8
Hardware	1 PC		

Casos de uso y escenarios implementados	65 - Visuali. zonas del centro	82 - Modificar llavero
	66 - Visualizar zona del centro	83 - Añadir llave
	67 - Añadir zona del centro	84 - Borrar llave
	68 - Borrar zona del centro	85 - Listar llaveros
	69 - Modificar zona del centro	86 - Visuali. almacenes de llaves
	70 - Visualizar llaves	87 - Visualizar almacén de llaves
	71 - Visualizar llave	88 - Añadir almacén de llaves
	72 - Añadir llave	89 - Borrar almacén de llaves
	73 - Borrar llave	90 - Modificar almacén de llaves
	74 - Dar de baja llave	91 - Añadir llave
	75 - Modificar llave	92 - Borrar llave
	76 - Listar llaves	93 - Modificar copias llave
	77 - Visualizar llaveros	94 - Añadir llavero
78 - Visualizar llavero	95 - Borrar llavero	
79 - Añadir llavero	96 - Modificar copias llavero	
80 - Borrar llavero	97 - Listar almacenes de llaves	
81 - Dar de baja llavero		

Planificación de la iteración



3.2. Análisis de los casos de uso y escenarios

A continuación mostramos todas las descripciones de trazo fino elaboradas para profundizar en el estudio de todos los casos de uso que implementaremos en esta iteración:

Caso de uso 65: Visualizar zonas del centro		
Descripción	Comenzaremos tratando en primer lugar las zonas del centro, que son a las que están asociadas las llaves y llaveros. Al igual que ocurría con otros elementos, se nos presentan casos de uso que requieren poder observar datos globales y concretos. En este caso trataremos el poder observar los datos principales de todas las zonas del centro, mientras que en el siguiente lo haremos con todos los datos de una zona.	
Actores principal	Administrador.	
Pre-condiciones	El usuario quiere visualizar los principales datos de todas las zonas que componen la sede de Informática Navarra S.A.	
Flujos	Básico	<ol style="list-style-type: none">1. El administrador loguea en el sistema y se dirige a la sección de zonas del centro.2. El sistema recopila todos los datos necesarios y se los muestra al administrador.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario visualiza en pantalla todos los datos que buscaba.	

Caso de uso 65: Visualizar zonas del centro

Caso de uso 66: Visualizar zona del centro		
Descripción	Una vez que se pueden visualizar todas las zonas, es el momento de ofrecer una vía para conocer todos los datos de cualquiera de ellas.	
Actores principal	Administrador.	
Pre-condiciones	El administrador desea observar todos los datos que componen una de las zonas que se encuentra visualizando.	
Flujos	Básico	<ol style="list-style-type: none">1. El usuario selecciona la zona a consultar y pide al sistema que le muestre todos sus datos.2. El sistema recupera todos los datos de los que dispone y se los ofrece al usuario.
	Alternativos (*)	Ninguno.
Post-condiciones	El administrador observa en un formulario ofrecido por el sistema todos los datos que componen la zona seleccionada.	

Caso de uso 66: Visualizar zona del centro



Caso de uso 67: Añadir zona del centro		
Descripción	No es normal la redistribución de zonas, pero debemos tener en cuenta que el centro no tiene por qué permanecer inalterado durante toda su vida. Puede que nazcan nuevas zonas con intención de particionar en mayor grado la sede, puede que sea necesario modificar algunas de ellas,...	
Actores principal	Administrador.	
Pre-condiciones	El administrador se encuentra logueado en el sistema y se dispone a dar de alta una nueva zona.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El administrador traslada al sistema sus pretensiones. 2. El sistema le ofrece un formulario con el que poder recoger todos los datos de la nueva zona. 3. El administrador lo rellena y devuelve al sistema, quien le pide al usuario una confirmación de que todos los datos introducidos son correctos y que está seguro de querer realizar la inserción. 4. El administrador da su visto bueno. (*) 5. El sistema procede con la adición de la zona. (*)
	Alternativos (*)	<p>El administrador no da su visto bueno:</p> <ul style="list-style-type: none"> • El sistema no realiza ninguna operación y el caso de uso da por concluido. <p>El administrador introduce algún dato con formato erróneo:</p> <ul style="list-style-type: none"> • El sistema avisa al usuario de su error y vuelve al final del punto 2 sin haber modificado ningún campo.
Post-condiciones	El administrador consigue su objetivo y la nueva zona entra a formar parte del sistema.	

Caso de uso 67: Añadir zona del centro



Caso de uso 68: Borrar zona del centro	
Descripción	Una zona puede dejar de existir en el centro ya sea porque otra la embeba o porque no se desee controlar con la aplicación (cosa muy poco probable). Éste es un hecho que controlaremos con accessManager.
Actores principal	Administrador.
Pre-condiciones	El usuario necesita borrar una zona del sistema ya que ésta ha dejado de existir.
Flujos	Básico <ol style="list-style-type: none">1. El usuario loguea en el sistema, se dirige a la pestaña de zonas, selecciona aquellas que desea borrar y pulsa el botón de borrado.2. El sistema envía al usuario un mensaje de advertencia avisándole de la operación que va a realizar.3. El usuario aprueba el borrado y éste tiene lugar en el sistema. (*)
	Alternativos (*) <p>El usuario no aprueba el borrado:</p> <ul style="list-style-type: none">• El sistema no elimina ningún registro de la base de datos y el caso de uso concluye.
Post-condiciones	El sistema ya no dispone de las zonas que el usuario mandó borrar.

Caso de uso 68: Borrar zona del centro



Caso de uso 69: Modificar zona del centro	
Descripción	Ahora trataremos el registro de los cambios que pueden darse en los datos que componen una zona, como son su código, descripción o nivel necesario para acceder a ésta.
Actores principal	Administrador.
Pre-condiciones	El administrador loguea en el sistema con la pretensión de modificar los datos de una de las zonas que se encuentra observando.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El administrador selecciona la zona que desea modificar y el sistema le proporciona un formulario con todos los datos que la componen. 2. El administrador edita todos los datos que considera erróneos y pide al sistema que actualice la zona con los datos que le proporciona. 3. El sistema avisa al usuario de que va a modificar la zona seleccionada con los datos indicados y le pregunta si está seguro de querer llevar a cabo esta operación. 4. El administrador asiente. (*) 5. El sistema actualiza todos los registros necesarios en la base de datos. (*)
	<p>Alternativos (*)</p> <p>El usuario no ve con buenos ojos realizar la actualización:</p> <ul style="list-style-type: none"> • El sistema da por concluido el caso de uso sin realizar ningún tipo de operación. <p>La actualización no puede llevarse a cabo ya que el usuario ha introducido algún dato con formato erróneo:</p> <ul style="list-style-type: none"> • El sistema le avisa de su error y pasa al final del primer punto del flujo básico.
Post-condiciones	Los cambios indicados surten efecto y la zona seleccionada dispone en estos momentos de datos válidos.

Caso de uso 69: Modificar zona del centro



Caso de uso 70: Visualizar llaves	
Descripción	Tras disponer de toda la funcionalidad necesaria para llevar a cabo una correcta gestión de las zonas, podemos pasar a tratar todo lo que ellas suponen: llaves, llaveros y almacenes. En primer lugar nos ocupamos de las llaves, ofreciendo una vía sencilla de poder observar los principales datos de todas las existentes.
Actores principal	Administrador.
Pre-condiciones	El usuario precisa conocer qué llaves están relacionadas con una zona.
Flujos	Básico 1. El administrador selecciona la zona de la que desea conocer sus llaves. 2. El sistema accede a la base de datos y recupera los datos más representativos de éstas, mostrándolos por pantalla.
	Alternativos (*) Ninguno.
Post-condiciones	El usuario puede visualizar los principales datos de todas las llaves pertenecientes a una zona.

Caso de uso 70: Visualizar llaves

Caso de uso 71: Visualizar llave	
Descripción	La visualización de los principales datos de las llaves puede que no deje satisfecho al usuario, con que no estaría de más el poder visualizar todos los atributos al completo.
Actores principal	Administrador.
Pre-condiciones	El usuario se encuentra en la pestaña de zonas y desea que el sistema le proporcione todos los datos que componen una llave de una zona dada.
Flujos	Básico 1. El administrador selecciona la llave a consultar. 2. El sistema se encarga de recuperar de la base de datos todos los datos que guarda de ella. A continuación genera un formulario con éstos y se lo muestra al administrador.
	Alternativos (*) Ninguno.
Post-condiciones	El usuario visualiza en pantalla todos los datos de la llave seleccionada.

Caso de uso 71: Visualizar llave



Caso de uso 72: Añadir llave	
Descripción	Está claro que el conjunto de llaves que permiten el acceso total a las zonas no permanecerá inmutable. Por ello los cuatro siguientes casos de uso.
Actores principal	Administrador.
Pre-condiciones	Una nueva llave entra en juego en la sede y es obligación del administrador registrarla en el sistema.
Flujos	Básico <ol style="list-style-type: none">1. El usuario indica al sistema que desea dar de alta una nueva llave.2. El sistema le proporciona un formulario con todos los campos necesarios.3. El usuario rellena el formulario y pide que se realice el registro.4. El sistema indica que necesita de una aprobación previa para poder realizar el alta solicitada.5. El usuario da su visto bueno y la adición de la llave al conjunto de llaves existentes en la zona es una realidad. (*)
	Alternativos (*) <p>El usuario ha introducido algún dato con un formato no válido:</p> <ul style="list-style-type: none">• El sistema se percató del error, lo exterioriza y no realiza la inserción solicitada. <p>El usuario cambia de idea y decide abortar el alta de la llave:</p> <ul style="list-style-type: none">• El sistema interrumpe el flujo básico y salta al final del caso de uso sin haber sido éste realizado.
Post-condiciones	El administrador ha conseguido dar de alta una nueva llave asociándola a una de las zonas existentes.

Caso de uso 72: Añadir llave



Caso de uso 73: Borrar llave					
Descripción	Si alguna de las cerraduras que pueden llegar a poblar una de las zonas desaparece (alguna puerta deja de necesitar llave, algún armario que deja de existir,...) también deberá hacerlo del sistema. No tiene sentido almacenar algo que no existe, es incoherente, innecesario.				
Actores principal	Administrador.				
Pre-condiciones	El administrador loguea en el sistema persiguiendo borrar alguna de las llaves existentes en una zona dada.				
Flujos	<table border="1"> <tr> <td style="text-align: center;">Básico</td> <td> <ol style="list-style-type: none"> 1. El usuario selecciona la zona en la que desea eliminar llaves. 2. El sistema le muestra las llaves existentes. 3. El usuario selecciona todas las llaves de las que desea prescindir y pide al sistema que proceda con la operación de borrado. 4. El sistema despliega un dialogo de aviso en el que informa al usuario del borrado y pide que lo apruebe si lo estima oportuno. 5. El usuario responde 'Sí' ante esta pregunta. (*) 6. El sistema elimina las llaves seleccionadas de la base de datos. </td> </tr> <tr> <td style="text-align: center;">Alternativos (*)</td> <td> <p>El usuario selecciona 'No':</p> <ul style="list-style-type: none"> • El sistema finaliza el caso de uso sorteando el sexto paso. </td> </tr> </table>	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona la zona en la que desea eliminar llaves. 2. El sistema le muestra las llaves existentes. 3. El usuario selecciona todas las llaves de las que desea prescindir y pide al sistema que proceda con la operación de borrado. 4. El sistema despliega un dialogo de aviso en el que informa al usuario del borrado y pide que lo apruebe si lo estima oportuno. 5. El usuario responde 'Sí' ante esta pregunta. (*) 6. El sistema elimina las llaves seleccionadas de la base de datos. 	Alternativos (*)	<p>El usuario selecciona 'No':</p> <ul style="list-style-type: none"> • El sistema finaliza el caso de uso sorteando el sexto paso.
	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona la zona en la que desea eliminar llaves. 2. El sistema le muestra las llaves existentes. 3. El usuario selecciona todas las llaves de las que desea prescindir y pide al sistema que proceda con la operación de borrado. 4. El sistema despliega un dialogo de aviso en el que informa al usuario del borrado y pide que lo apruebe si lo estima oportuno. 5. El usuario responde 'Sí' ante esta pregunta. (*) 6. El sistema elimina las llaves seleccionadas de la base de datos. 			
Alternativos (*)	<p>El usuario selecciona 'No':</p> <ul style="list-style-type: none"> • El sistema finaliza el caso de uso sorteando el sexto paso. 				
Post-condiciones	El sistema ya no dispone de las llaves eliminadas por el administrador.				

Caso de uso 73: Borrar llave



Caso de uso 74: Dar de baja llave	
Descripción	El borrado anteriormente citado implicaría la pérdida de la trazabilidad entre accesos y llaves empleadas. Para evitarlo se diseñará un sistema por el cual la llave deje de formar parte de la aplicación (muestra en pestaña) pero no del sistema.
Actores principal	Administrador.
Pre-condiciones	El administrador quiere dar de baja una llave ya que esta no será necesaria de aquí en adelante.
Flujos	Básico <ol style="list-style-type: none">1. El administrador se sitúa en la pestaña de zonas, selecciona aquella donde desea dar de baja llaves y selecciona las que estime oportuno. Acto seguido, traslada al sistema su intención de darlas de baja.2. El sistema procede con la baja de éstas siempre y cuando el administrador acepte el mensaje de aviso que le es mostrado por pantalla. (*)
	Alternativos (*) <p>El administrador finalmente decide no dar de baja las llaves que ha seleccionado, respondiendo negativamente ante el diálogo en pantalla:</p> <ul style="list-style-type: none">• El sistema no ejecuta ninguna acción y da por finalizado el caso de uso.
Post-condiciones	El administrador consigue dar de baja las llaves que seleccionó.

Caso de uso 74: Dar de baja llave



Caso de uso 75: Modificar llave		
Descripción	Las llaves están compuestas de código, descripción y nota, que son atributos que no tienen por qué permanecer inalterados. Puede ser necesario cambiar el código de una llave, editar la descripción de ésta por ser errónea,...	
Actores principal	Administrador.	
Pre-condiciones	El usuario se encuentra en el sistema, más concretamente en la tab de zonas visualizando el conjunto de llaves que forman parte de una zona, y desea modificar los datos que conforman una llave al no ser éstos válidos.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona la llave que desea modificar. 2. El sistema le ofrece todos los datos que la componen en un formulario. 3. El usuario edita los que estima conveniente y pide al sistema que proceda con la actualización. 4. El sistema insta al usuario a que dé su visto bueno para que la actualización tenga lugar. 5. El usuario obedece, la da y el caso de uso finaliza con la consecución del fin para el que fue ideado. (*)
	Alternativos (*)	<p>El usuario ha proporcionado datos que no puede aceptar el sistema por ser erróneos en su formato:</p> <ul style="list-style-type: none"> • El sistema avisa al usuario del error ocurrido y retorna al final del punto 2 sin haber modificado dato alguno. <p>El usuario no da su visto bueno a la actualización:</p> <ul style="list-style-type: none"> • El sistema no actualiza ningún registro de la base de datos y da por concluido el caso de uso.
Post-condiciones	El usuario consigue que la llave conste con datos válidos y no erróneos, como los que tenía antes de iniciarse este caso de uso.	

Caso de uso 75: Modificar llave



Caso de uso 76: Listar llaves		
Descripción	El cliente pide expresamente al equipo de desarrollo que le ofrezca una vía para poder obtener reportes de las llaves que existen en el sistema.	
Actores principal	Administrador.	
Pre-condiciones	El usuario se encuentra observando las llaves que forman parte del sistema y se dispone a obtener un reporte de éstas.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario pide al sistema que genere un informe con las llaves que se encuentra visualizando en pantalla. 2. El sistema pide al usuario la ubicación donde deberá depositar el reporte a generar. 3. El usuario proporciona la ruta. (*) 4. El sistema elabora el informe y lo deposita en la ubicación indicada.
	Alternativos (*)	<p>El usuario no proporciona ninguna ubicación, cancelando el diálogo de selección e ruta:</p> <ul style="list-style-type: none"> • El caso de uso finaliza sin la generación de ningún informe.
Post-condiciones	El usuario dispone del reporte solicitado en la ruta indicada.	

Caso de uso 76: Listar llaves

Caso de uso 77: Visualizar llaveros		
Descripción	Completadas las llaves debemos hacer lo mismo con los llaveros. Los casos de uso serán similares, solo que algo más numerosos y complicados ya que nos encontramos ante objetos compuestos.	
Actores principal	Administrador.	
Pre-condiciones	El administrador necesita conocer el conjunto de llaveros existentes. Para ello, loguea en el sistema y se dirige a la tab de zonas.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El administrador pide al sistema que le muestre el conjunto de llaveros existentes. 2. El sistema acude a la base de datos y recupera de ella todos los datos necesarios para mostrárselos al administrador por pantalla y satisfacer su necesidad.
	Alternativos (*)	Ninguno.
Post-condiciones	El administrador visualiza los llaveros existentes, habiendo conseguido realizar el caso de uso con éxito.	

Caso de uso 77: Visualizar llaveros



Caso de uso 78: Visualizar llavero		
Descripción	Observar el conjunto de llaveros existentes puede no ser del todo suficiente, ya que existen datos que se escapan al anterior caso de uso, como pueden ser el conjunto de llaves que los conforman.	
Actores principal	Administrador.	
Pre-condiciones	El usuario requiere al sistema que le muestre todos los datos disponibles sobre un llavero que se encuentra visualizando.	
Flujos	Básico	<ol style="list-style-type: none">1. El usuario selecciona el llavero que desea visualizar.2. El sistema, gracias a la base de datos, muestra por pantalla todos los datos que el usuario estaba buscando.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario observa por pantalla todos los datos que describen el llavero seleccionado.	

Caso de uso 78: Visualizar llavero



Caso de uso 79: Añadir llavero	
Descripción	Si el conjunto de llaves puede variar, el conjunto de llaveros no es menos.
Actores principal	Administrador.
Pre-condiciones	El administrador se encuentra logueado en el sistema y precisa dar de alta un nuevo llavero.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El administrador selecciona las llaves que conformarán el nuevo llavero y pide al sistema crear uno nuevo con éstas. 2. El sistema le proporciona un formulario para que el administrador termine de definir el nuevo llavero. 3. El administrador lo rellena y pide al sistema que realice el alta. 4. El sistema requiere al administrador una autorización previa para realizar la operación que éste ha solicitado. 5. El administrador la otorga. (*) 6. El sistema realiza el registro del nuevo llavero en el sistema. (*)
	<p>Alternativos (*)</p> <p>El administrador decide abortar la inserción del nuevo llavero:</p> <ul style="list-style-type: none"> • El sistema obedece, no realiza ninguna acción y la ejecución del caso de uso finaliza. <p>El sistema se percata de que el usuario ha introducido algún dato con un formato no deseado:</p> <ul style="list-style-type: none"> • Le avisa del error que acaba de cometer y pasa al final del punto 2 sin modificar ningún dato del formulario.
Post-condiciones	El sistema alberga un nuevo llavero con las características que el administrador ha proporcionado.

Caso de uso 79: Añadir llavero



Caso de uso 80: Borrar llavero	
Descripción	Borrar un llavero es otra de las funcionalidades que debemos proporcionar, ya que éstos no perdurarán eternamente.
Actores principal	Administrador.
Pre-condiciones	El usuario accede al sistema con la intención de realizar el borrado de alguno de los llaveros que no quiere que permanezcan en el sistema.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El administrador accede a la sección de llaveros y selecciona aquellos que desea eliminar, iniciando el proceso de borrado con la pulsación del botón creado para tal fin. 2. El sistema despliega un dialogo de aviso en el que advierte al administrador de que va a proceder con el borrado de los llaveros seleccionados, necesitando previamente una autorización para asegurarse que el borrado es el que el administrador realmente quiere realizar. 3. El administrador da su visto bueno. (*) 4. El sistema elimina del sistema todos los llaveros seleccionados.
	<p>Alternativos (*)</p> <p>El administrador cambia de idea y no ve necesario realizar el borrado:</p> <ul style="list-style-type: none"> • El sistema aborta el caso de uso y se llega al final de éste.
Post-condiciones	El usuario consigue su objetivo: eliminar los llaveros seleccionados.

Caso de uso 80: Borrar llavero

Caso de uso 81: Dar de baja llavero	
Descripción	Al igual que ocurría con las llaves, la eliminación de llaveros implica la pérdida de la trazabilidad entre los accesos y éstos, con que si se quiere seguir manteniéndola debe proporcionarse una forma de dar de baja llaveros.
Actores principal	Administrador.
Pre-condiciones	El administrador loguea en el sistema con el objetivo de dar de baja alguno de los llaveros que residen en éste.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario accede a la pestaña donde se muestran los llaveros y selecciona aquellos que desea dar de baja, pidiendo al sistema que realice dicha acción. 2. El sistema obedece siempre y cuando reciba una respuesta afirmativa al mensaje de aviso que enviará al usuario explicándole la operación que se dispone a realizar. (*)
	<p>Alternativos (*)</p> <p>El usuario responde negativamente al aviso:</p> <ul style="list-style-type: none"> • El sistema da por concluido el caso de uso no llevando a cabo ninguna acción.
Post-condiciones	El administrador consigue dar de baja los llaveros que pretendía.

Caso de uso 81: Dar de baja llavero



Caso de uso 82: Modificar llavero	
Descripción	Con los tres siguientes casos de uso haremos capaz al usuario de poder modificar los datos o composición de cualquiera de los llaveros que se proponga.
Actores principal	Administrador.
Pre-condiciones	El usuario inicia este caso de uso persiguiendo modificar alguno de los datos que componen uno de los llaveros que se encuentra visualizando.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El administrador selecciona el llavero a modificar y transmite al sistema su deseo. 2. El sistema despliega un formulario con todos los datos del llavero, de modo que el administrador pueda editar lo que estime oportuno. 3. Tras la edición, el administrador pide al sistema que realice la actualización de campos. 4. El sistema envía al administrador un aviso pidiéndole que confirme que desea modificar el llavero seleccionado. 5. El administrador responde positivamente y la actualización tiene lugar en la base de datos. (*)
	<p>Alternativos (*)</p> <p>El administrador introduce datos con un formato erróneo:</p> <ul style="list-style-type: none"> • El sistema avisa al usuario del error acontecido y retorna al final del punto 2. <p>El administrador aborta la actualización contestando negativamente a la advertencia:</p> <ul style="list-style-type: none"> • El sistema no modifica ningún registro en la base de datos y da por concluido el caso de uso.
Post-condiciones	El administrador consigue que el llavero que constaba con datos erróneos ahora lo haga con válidos.

Caso de uso 82: Modificar llavero



Caso de uso 83: Añadir llave	
Descripción	Informática Navarra S.A. puede estimar oportuno modificar algún llavero añadiéndole más llaves.
Actores principal	Administrador.
Pre-condiciones	El usuario quiere modificar uno de los llaveros que se encuentra visualizando mediante la adición de alguna nueva llave existente.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario selecciona las llaves y llavero en el que insertarlas, instando al sistema a que realice los cambios necesarios para llevar a cabo esta actualización. 2. El sistema cumple con las intenciones del usuario siempre y cuando éste conteste 'Sí' a su dialogo de aviso. (*)
	<p>Alternativos (*)</p> <p>El usuario contesta 'No' ya que finalmente no desea añadir las llaves al llavero:</p> <ul style="list-style-type: none"> • El caso de uso finaliza sin ejecutarse ninguna sentencia sobre la base de datos.
Post-condiciones	El administrador consigue añadir las llaves al llavero seleccionado, pasando a constar éste con un conjunto de llaves distinto al que poseía antes de comenzar este caso de uso.

Caso de uso 83: Añadir llave

Caso de uso 84: Borrar llave	
Descripción	También puede suceder que alguna llave deje de constar en algún llavero, siendo necesario implementar el caso de uso opuesto al anterior.
Actores principal	Administrador.
Pre-condiciones	El administrador se encuentra registrado en el sistema, visualizando un llavero y desea borrar algunas de las llaves que aúna.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario selecciona las llaves que desea que dejen de formar parte del llavero y pide al sistema que las elimine. 2. El sistema pide al usuario que confirme sus pretensiones. 3. El usuario responde afirmativamente. (*) 4. El sistema se encarga de realizar el borrado.
	<p>Alternativos (*)</p> <p>El usuario contesta negativamente:</p> <ul style="list-style-type: none"> • Se salta al final del caso de uso quedando el llavero inalterado.
Post-condiciones	Las llaves seleccionadas por el usuario dejan de formar parte del llavero.

Caso de uso 84: Borrar llave



Caso de uso 85: Listar llaveros		
Descripción	Otro de los requisitos que el cliente nos impone es el que pueda listar todos los llaveros registrados en el sistema.	
Actores principal	Administrador.	
Pre-condiciones	El usuario se encuentra en la sección de llaveros y se dispone a obtener un reporte de éstos.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El administrador pide al sistema que elabore un informe con todos los llaveros de los que dispone. 2. El sistema responde pidiéndole que le ofrezca una ruta en la que depositar dicho informe. 3. El usuario se la proporciona. (*) 4. El sistema genera el reporte solicitado y lo deposita en la ubicación indicada.
	Alternativos (*)	El usuario no se la proporciona: <ul style="list-style-type: none"> • El sistema no elabora ningún tipo de informe y da por finalizada la ejecución del caso de uso.
Post-condiciones	El usuario puede disfrutar del informe solicitado en la ubicación señalada.	

Caso de uso 85: Listar llaveros

Caso de uso 86: Visualizar almacenes de llaves		
Descripción	El usuario debe poder visualizar toda la información referente a los distintos almacenes existentes y qué elementos almacenan. En un principio, se define el poder observar los almacenes que existen y a continuación poder visualizar toda la información de cada almacén.	
Actores principal	Administrador.	
Pre-condiciones	El administrador necesita conocer qué almacenes se encuentran registrados en el sistema.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario loguea y accede a la sección de almacenes. 2. El sistema se encarga de recuperar de la base de datos los principales datos de los almacenes existentes y mostrárselos.
	Alternativos (*)	Ninguno.
Post-condiciones	El administrador visualiza en pantalla todos los almacenes de datos existentes.	

Caso de uso 86: Visualizar almacenes de llaves



Caso de uso 87: Visualizar almacén de llaves		
Descripción	Una vez que el cliente puede obtener todos los almacenes existentes, debería poder seleccionar uno y visualizar toda su composición.	
Actores principal	Administrador.	
Pre-condiciones	El administrador precisa conocer el contenido de alguno de los almacenes que visualiza en pantalla.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El administrador selecciona el almacén del que desea obtener toda su información. 2. El sistema accede a la base de datos, rescata toda la información que dispone de éste y la muestra en pantalla.
	Alternativos (*)	Ninguno.
Post-condiciones	El administrador obtiene la información que buscaba y el caso de uso finaliza con éxito.	

Caso de uso 87: Visualizar almacén de llaves

Caso de uso 88: Añadir almacén de llaves		
Descripción	No es lo normal, pero nunca se puede desechar el hecho de que pueda surgir algún nuevo almacén en el que Informática Navarra S.A. desee almacenar llaves y llaveros.	
Actores principal	Administrador.	
Pre-condiciones	El usuario loguea en el sistema con la intención de realizar el alta de un nuevo almacén de llaves.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario acude a la sección de almacenes e indica al sistema que desea realizar el alta de uno nuevo. 2. El sistema le proporciona un formulario que deberá rellenar. 3. El usuario obedece y tras completarlo le pide al sistema que proceda con el alta. 4. Éste inserta en la base de datos un registro con el nuevo almacén previa autorización por parte del usuario. (*)
	Alternativos (*)	<p>El usuario proporciona algún dato que no cumple con el formato esperado:</p> <ul style="list-style-type: none"> • El sistema avisa al usuario de su error y vuelve al final del punto 2 sin modificar ningún campo. <p>El usuario cambia de idea y aborta dar de alta el nuevo almacén:</p> <ul style="list-style-type: none"> • El caso de uso salta del inicio al final del punto 4, pasando por alto la adición al sistema del nuevo almacén.
Post-condiciones	El usuario consigue que el nuevo almacén sea registrado en la base de datos.	

Caso de uso 88: Añadir almacén de llaves



Caso de uso 89: Borrar almacén de llaves		
Descripción	Reducir el número de almacenes en los que guardar llaves y llaveros debería ser posible en la aplicación informática.	
Actores principal	Administrador.	
Pre-condiciones	El usuario loguea en el sistema con el objetivo de borrar alguno de los almacenes existentes.	
Flujos	Básico	<ol style="list-style-type: none">1. El administrador selecciona los almacenes que desea eliminar y comunica al sistema su intención.2. Éste consulta al administrador para tener la certeza de que está seguro de querer realizar la acción propuesta.3. El administrador contesta afirmativamente, permitiendo que el sistema ejecute la tarea. (*)
	Alternativos (*)	El administrador responde negativamente: <ul style="list-style-type: none">• El sistema no realiza ningún tipo de borrado y da por concluido el flujo básico del caso de uso.
Post-condiciones	El sistema ya no dispone de los almacenes de datos eliminados por el usuario.	

Caso de uso 89: Borrar almacén de llaves



Caso de uso 90: Modificar almacén de llaves	
Descripción	A continuación tratamos todos los casos de uso que pueden ocasionar la modificación del estado de cualquiera de los almacenes existentes.
Actores principal	Administrador.
Pre-condiciones	El usuario se percata de que uno de los almacenes consta con datos erróneos y se dirige a la sección dedicada a éstos persiguiendo actualizarlo de modo que todos los datos de éste sean correctos.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El administrador selecciona el almacén cuyos datos desea modificar y transmite al sistema su querer. 2. Éste le muestra un formulario con los datos del almacén para que pueda editarlos a su antojo. 3. Una vez el administrador dispone de todo listo, pide al sistema que proceda con la actualización. 4. El sistema envía un aviso al administrador indicándole la operación que se dispone a realizar y pidiendo permiso para ello. 5. El administrador da el visto bueno. (*) 6. La actualización tiene lugar. (*)
	<p>Alternativos (*)</p> <p>El administrador no da el visto bueno:</p> <ul style="list-style-type: none"> • El caso de uso finaliza sin completarse. <p>El administrador introduce algún dato con formato erróneo:</p> <ul style="list-style-type: none"> • El sistema le avisa del error que acaba de cometer y retorna al final del punto 2.
Post-condiciones	El usuario hace que el almacén cuyos datos eran erróneos en un principio ahora sean correctos.

Caso de uso 90: Modificar almacén de llaves



Caso de uso 91: Añadir llave	
Descripción	La distribución de llaves en los distintos almacenes variará a lo largo del tiempo. Esto hace que debamos implementar funciones de adición, sustracción y modificación del número de copias de cualquier llave.
Actores principal	Administrador.
Pre-condiciones	El usuario se encuentra en la sección dedicada a los almacenes y quiere añadir a uno de ellos una llave.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario selecciona la llave y almacén que desea relacionar y clica en el botón de 'Añadir llave'. 2. El sistema le proporciona un campo para que indique el número de copias de la llave que residirán en el almacén. 3. El usuario las indica y pide al sistema que proceda. 4. Éste lo hace pero antes debe recibir el visto bueno por parte del usuario. (*)
	<p>Alternativos (*)</p> <p>El usuario indica un número de copias no comprendido entre 1 y 99:</p> <ul style="list-style-type: none"> • El sistema finaliza el caso de uso con un mensaje de error. <p>El usuario acaba viendo innecesario introducir la llave en el almacén, con lo que contesta 'No' al mensaje de advertencia:</p> <ul style="list-style-type: none"> • El sistema concluye sin realizar ninguna inserción y sin mostrar ningún tipo de aviso.
Post-condiciones	El administrador dispone de copias de la llave en el almacén indicado.

Caso de uso 91: Añadir llave



Caso de uso 92: Borrar llave	
Descripción	La extracción de una llave de un almacén debe ser reflejado también en el sistema.
Actores principal	Administrador.
Pre-condiciones	El administrador se encuentra logueado en la aplicación y desea borrar alguna de las llaves que pueblan un almacén concreto.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario selecciona las llaves que desea eliminar del almacén que se encuentra visualizando e indica al sistema que efectúe las operaciones necesarias para ello. 2. El sistema despliega un diálogo en el que indica al usuario las llaves que se dispone a borrar e instándole a que confirme la operación. 3. El usuario la confirma y el borrado surte efecto. (*)
	<p>Alternativos (*)</p> <p>El usuario contesta con un 'No':</p> <ul style="list-style-type: none"> • El sistema no elimina ningún registro de la base de datos.
Post-condiciones	El usuario ha conseguido que las llaves seleccionadas no permanezcan en el almacén.

Caso de uso 92: Borrar llave

Caso de uso 93: Modificar copias llave	
Descripción	Puede no variar la existencia de la llave pero sí el número de copias de las que se dispone. De ahí nace este caso de uso.
Actores principal	Administrador.
Pre-condiciones	El usuario se dispone a modificar el número de copias existentes de una llave en un almacén.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El administrador selecciona la llave dentro del almacén y pide al sistema poder modificar su número de copias. 2. El sistema le muestra en un campo el número de copias registradas y el administrador lo edita con el número de copias deseadas. 3. La actualización tiene lugar siempre y cuando el administrador dé su visto bueno (*).
	<p>Alternativos (*)</p> <p>El administrador introduce un número de copias no permitido:</p> <ul style="list-style-type: none"> • El sistema indica el error acontecido y no actualiza el número de copias. <p>El administrador aborta la actualización:</p> <ul style="list-style-type: none"> • El caso de uso llega a su fin sin haber efectuado cambio alguno en el sistema.
Post-condiciones	El usuario consigue actualizar el número de copias existentes de una llave en un almacén.

Caso de uso 93: Modificar copias llave



Caso de uso 94: Añadir llavero	
Descripción	Los llaveros poseen las mismas necesidades que las llaves, por lo que debemos crear casos de uso análogos para ellos.
Actores principal	Administrador.
Pre-condiciones	El usuario se registra en el sistema y accede a la sección de almacenes ante la necesidad de añadir un llavero a uno de ellos.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El administrador selecciona el llavero y almacén que desea relacionar y clicla el botón 'Añadir llavero'. 2. El sistema le proporciona un campo en el que indicar el número de copias del llavero a introducir en el almacén. 3. El administrador rellena el campo y pide al sistema que realice la inserción. 4. Éste muestra por pantalla un diálogo en el que avisa al administrador de la acción que realizará si éste le da permiso para ello. 5. El administrador lo otorga. (*) 6. El sistema inserta el llavero en el almacén indicado. (*)
	<p>Alternativos (*)</p> <p>El usuario no da su visto bueno a la inserción:</p> <ul style="list-style-type: none"> • El sistema no realiza ninguna acción y el caso de uso termina. <p>A la hora de realizar la inserción, el sistema se percata de que el número de copias indicado es incorrecto:</p> <ul style="list-style-type: none"> • Avisa al administrador del error que ha cometido y permanece inalterado a la espera de que el usuario vuelva a intentar la inserción o pase a ejecutar otro caso de uso.
Post-condiciones	La base de datos se ve alterada mediante la introducción de un llavero en uno de sus almacenes.

Caso de uso 94: Añadir llavero



Caso de uso 95: Borrar llavero	
Descripción	El borrado de llaveros será similar al realizado con las llaves.
Actores principal	Administrador.
Pre-condiciones	El administrador accede a la aplicación con ánimo de borrar alguno de los llaveros que constan en un almacén.
Flujos	Básico <ol style="list-style-type: none">1. El administrador despliega el contenido del almacén que desea modificar y selecciona de él aquellos llaveros que desee desechar. Acto seguido, pulsa el botón 'Borrar llavero'.2. El sistema responde avisándole de que va a eliminar los llaveros que indica, pero que antes necesita asegurarse de que la selección es correcta.3. El administrador le indica que lo es, hecho que desencadena que el caso de uso finalice correctamente. (*)
	Alternativos (*)
Post-condiciones	El almacén ya no dispone de los llaveros borrados por el usuario.

Caso de uso 95: Borrar llavero



Caso de uso 96: Modificar copias llavero	
Descripción	Cuando el número de copias de un llavero varíe el usuario deberá ejecutar este caso de uso.
Actores principal	Administrador.
Pre-condiciones	El usuario precisa modificar el número de copias existentes de un llavero en un almacén. Para ello loguea en el sistema y se dirige a la sección de llaveros.
Flujos	Básico <ol style="list-style-type: none">1. El usuario selecciona el almacén y llavero que desea modificar.2. Tras recibir la orden de modificación, el sistema hace visible un campo con el número de copias disponibles del llavero, número que el usuario modifica y pide al sistema que almacene.3. El sistema realiza la actualización si el usuario da su visto bueno. (*)
	Alternativos (*) <p>El usuario introduce un número de copias no permitido:</p> <ul style="list-style-type: none">• El sistema le avisa del error que ha cometido y permanece inalterado a la espera de acontecimientos. <p>El sistema aborta la modificación ya que no la estima oportuna:</p> <ul style="list-style-type: none">• El sistema no realiza ninguna acción y la ejecución del caso de uso concluye.
Post-condiciones	El usuario ha conseguido modificar el número de copias de un llavero existentes en uno de los almacenes.

Caso de uso 96: Modificar copias llavero

Caso de uso 97: Listar almacenes de llaves		
Descripción	El cliente desea poder obtener siempre que lo necesite un informe con todos los almacenes existentes y los elementos que éstos almacenan.	
Actores principal	Administrador.	
Pre-condiciones	El administrador precisa conocer el estado de los almacenes a través de un reporte.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El administrador pide al sistema que elabore un informe con los almacenes existentes y las llaves y llaveros que guardan. 2. El sistema le pide que indique en qué ubicación desea recibir el reporte. 3. El usuario selecciona el <i>path</i> a través de un cuadro de diálogo. (*) 4. El sistema se encarga de generar el reporte y depositarlo en la ruta indicada.
	Alternativos (*)	<p>El usuario no selecciona ninguna ubicación, cancelando el cuadro de diálogo:</p> <ul style="list-style-type: none"> • El caso de uso finaliza.
Post-condiciones	El administrador puede disfrutar del informe requerido en la ubicación indicada.	

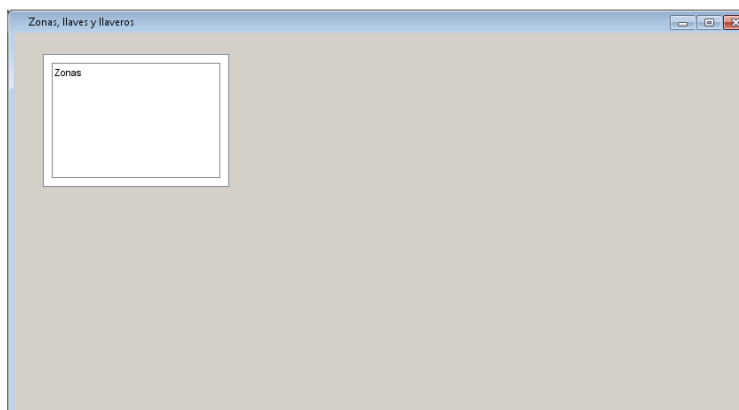
Caso de uso 97: Listar almacenes de llaves

4. Diseño

4.1. Caso de uso 65 y 66: Visualizar zonas del centro y visualizar zona del centro

Comenzaremos solucionando la visualización de las zonas que componen el centro. Quitando los atributos que corresponden a fechas, cada zona está compuesta por un código, una descripción y un nivel. El que sean tan pocos atributos nos lleva a agrupa ambos casos de uso en un mismo diseño.

Para empezar, diseñar la interfaz gráfica que permita el desarrollo de ambos. Pensamos que con una tabla bastará:



Interfaz gráfica propuesta



Ahora corresponde diseñar un proceso que nos ofrezca la visualización de todas las zonas. Podríamos:

1. Crear un objeto de tipo Zona.
2. Pedir al objeto que obtenga todas las zonas registradas en el centro.
3. Llevar los resultados obtenidos (descripción y nivel) a la tabla.

EL método que debe realizar la búsqueda del punto 2 es:

MÉTODO	FUNCIONALIDAD
<i>obtenerZonas</i>	Método que realiza la búsqueda de todas las zonas existentes en la sede de Informática Navarra S.A. y que se encuentran registradas en el sistema.

Método a implementar y a añadir a la API

Para acabar de aclarar todo el proceso, tenemos estos diagramas de actividades y de secuencia:

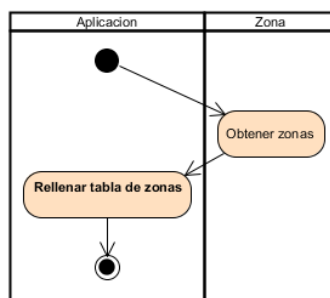


Diagrama de actividades

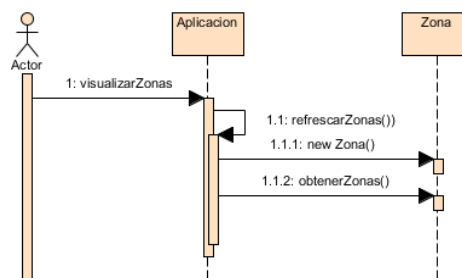


Diagrama de secuencia



4.2. Caso de uso 67: Añadir zona del centro

Los elementos gráficos generados para la visualización son escasos si queremos realizar más operaciones sobre las zonas visualizadas. Con el fin de proporcionar al usuario el resto de funcionalidad que demanda, instalamos una botonera en la parte inferior de la tabla, para que desde allí pueda invocar todas las operaciones que necesite, y un formulario con el que pueda introducir (y modificar) nuevas zonas:

Interfaz gráfica propuesta

El formulario deberá estar compuesto por los campos que determinan una zona, que son: código, descripción y nivel. Los dos primeros son aleatorios, de escritura “libre”, mientras que el tercero deberá ser uno de los niveles existentes, hecho que nos lleva a crear un JComboBox para él.

Con todo ésto dispondremos de todos los elementos necesarios para poder ejecutar la siguiente secuencia de sucesos:

1. Crear un objeto de tipo Zona con los datos recogidos del formulario.
2. Insertarlo en la base de datos utilizando el método que desarrollamos para ello al implementar la API inicial (*insertarZonaBD*).
3. Refrescar la tabla de zonas para que así aparezca en ella la nueva zona insertada.



No es un proceso para nada complejo, pero resulta mucho más sencillo de comprender si realizamos sus diagramas:

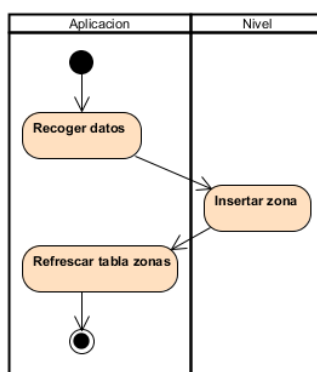


Diagrama de actividades

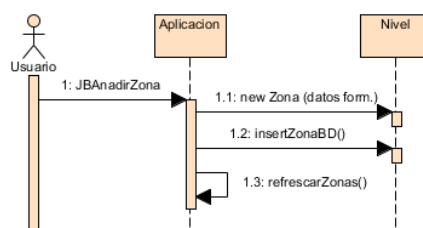


Diagrama de secuencia

4.3. Caso de uso 68: Borrar zona del centro

En el caso opuesto al anterior, nos encontramos con el de borrar zonas del centro. La interfaz gráfica no cambia, salvo por el botón de borrado que añadimos a la botonera.

La estrategia diseñada se basa en la seguida en otros casos de uso anteriores, y será la de recorrer todas las filas seleccionadas en la tabla de zonas e ir eliminándolas una a una:

1. Obtendremos las zonas seleccionadas.
2. Las recorreremos una a una eliminándolas de la base de datos.
3. Finalmente, refrescaremos la tabla de zonas para que no aparezcan las que han sido borradas.



A la hora de obtener cada zona en el segundo paso, necesitamos añadir a la API inicial un constructor que la obtenga a partir de su descripción, que si atendemos al esquema relacional es de tipo *UNIQUE* (de ahí que los resultados sean siempre inequívocos, únicos y podamos implementar este constructor):

MÉTODO	FUNCIONALIDAD
<i>Zona</i>	Constructor que crea un objeto de tipo Zona a partir de su descripción y la consulta a los registros existentes en la base de datos.

Método a implementar y a añadir a la API

¿Sencillo? Pues lo es más aún si atendemos a estos diagramas:

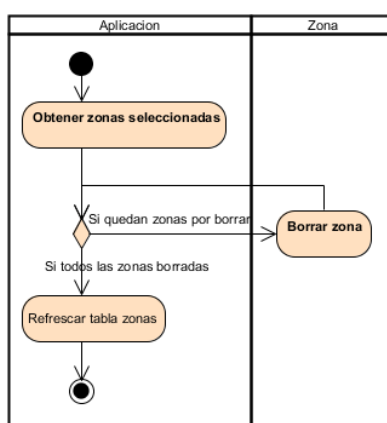


Diagrama de actividades

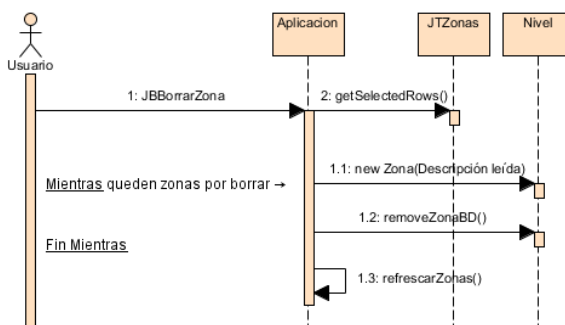


Diagrama de secuencia



4.4. Caso de uso 69: Modificar zona del centro

No por ser poco utilizado debe ser olvidado este caso de uso. Que el usuario necesite modificar una zona no es un hecho anormal, y podemos permitirselo si añadimos un botón de modificación a los ya existentes en la botonera. Su pulsación podría desencadenar algo parecido a:

1. Crear la nueva zona con los datos recogidos del formulario.
2. Recuperar la zona vieja de la base de datos.
3. Actualizar la zona vieja con la nueva.
4. Refrescar la tabla de zonas, para que los cambios sean visibles en ésta.

De forma algo más detallada tendríamos:

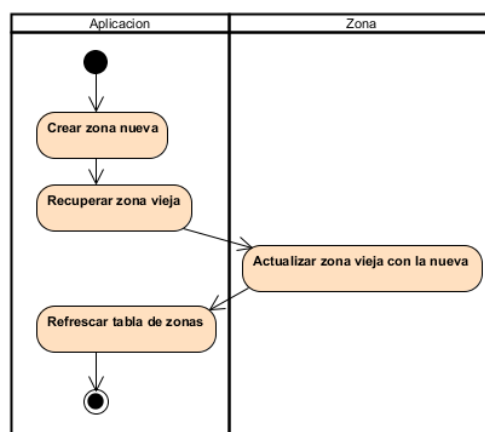


Diagrama de actividades

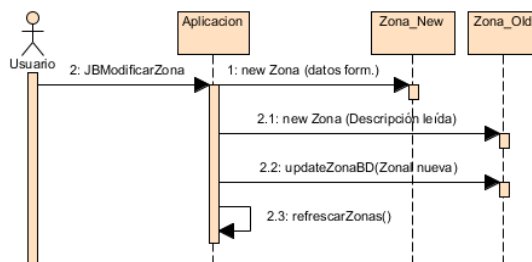


Diagrama de secuencia

Como vemos, no hay nada que añadir a la API, y con este diseño debería bastar para codificar el caso de uso.

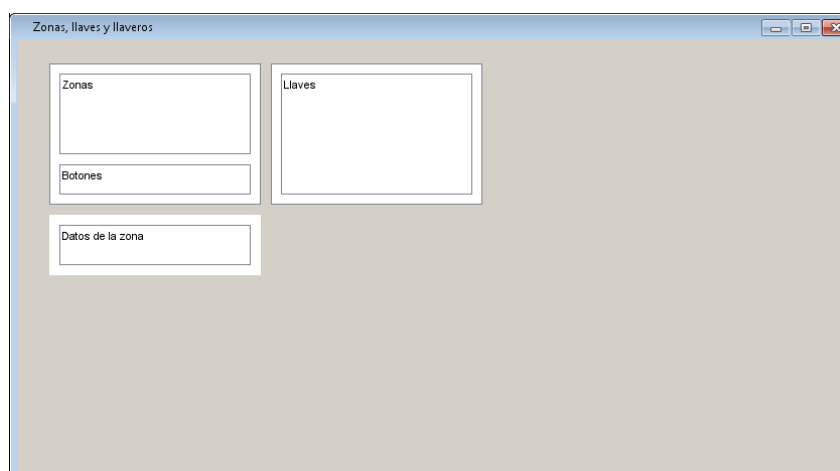


4.5. Caso de uso 70: Visualizar llaves

Tras haber diseñado la visualización y edición de las zonas que pueden componer el centro, nos adentramos en los elementos que se asocian a cada una de ellas: las llaves.

En un principio diseñaremos una vía para que el usuario pueda visualizar las llaves que pertenecen a una zona, mientras que en la parte de ‘almacenes’ lo haremos para que pueda observar la totalidad de todas ellas.

Como siempre, realizamos la visualización de elementos variables a través de una JTable:



Interfaz gráfica propuesta

Clicar sobre una zona debe desencadenar este proceso:

1. Crear un objeto de tipo Zona con la descripción leída en ésta.
2. Realizar la búsqueda de las llaves que le pertenecen.
3. Rellenar la tabla de llaves con los resultados obtenidos en el punto anterior (código y descripción de la llave).

El método de búsqueda empleado en el punto dos aún no ha sido implementado y debe ser asignado a una de las clases existentes (ninguna mejor que Zona):

MÉTODO	FUNCIONALIDAD
<i>obtenerLlaves</i>	Método que obtiene los principales datos de las llaves pertenecientes a la zona que lo ejecuta.

Método a implementar y a añadir a la API



El flujo de actividades que resuelven el caso de uso puede ser definido con mayor detalle a través de estos diagramas de actividades...

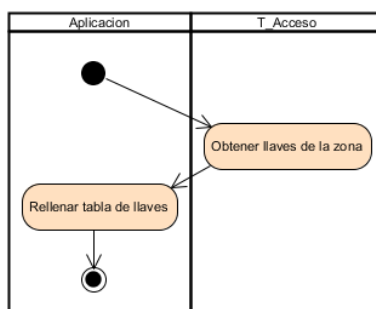


Diagrama de actividades

... y de secuencia:

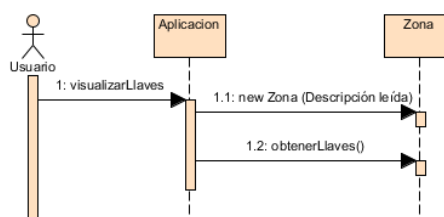
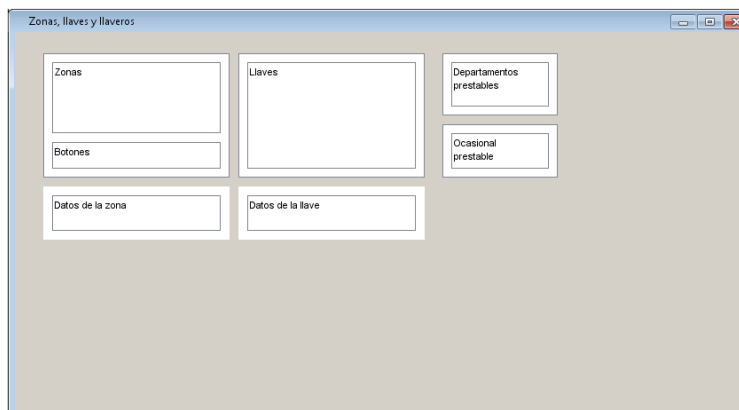


Diagrama de secuencia

4.6. Caso de uso 71: Visualizar llave

A los códigos y descripciones que conseguimos con el caso de uso anterior puede sumarse las notas que puedan existir en alguna de las llaves y todos los departamentos y personal ocasional que puedan acceder a su préstamo. Por ésto y por los casos de uso que suceden a éste, añadimos un panel que actúe como formulario para la muestra y recogida de datos y dos tablas que muestren los departamentos y personal ocasional que puedan acceder al préstamo de cada llave:



Interfaz gráfica propuesta



Pulsando cualquier llave podemos conseguirse la muestra de todos sus datos en el nuevo formulario de la siguiente forma:

1. Crear una Llave con el código leído.
2. Obtener y mostrar todos sus datos en el formulario.
3. Consultar los departamentos que pueden tomarla en préstamo y llevarlos a la primera de las dos nuevas tablas.
4. Recuperar el personal ocasional a quien puede serle prestada y rellenar con él la tabla destinada a tal fin.

Parece (y es) un proceso algo más complejo que cualquiera de los que hemos realizado en esta iteración, pero no debería darnos ningún problema si seguimos estos útiles diagramas:

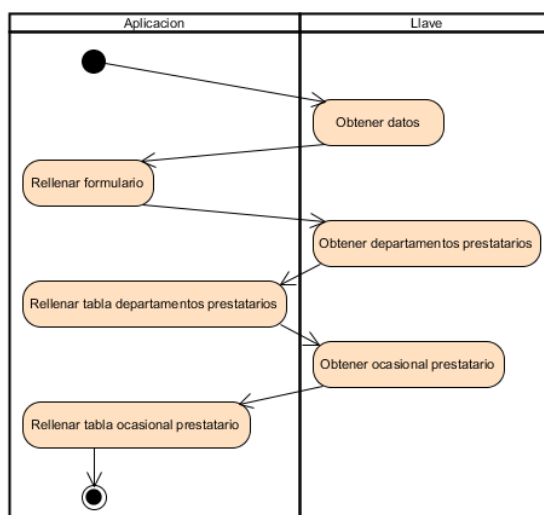


Diagrama de actividades

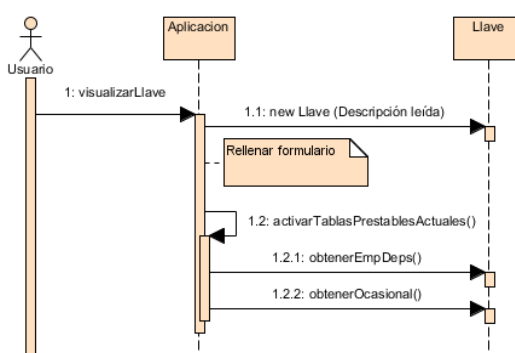


Diagrama de secuencia



No debemos olvidar añadir a la API los métodos *obtenerEmpDeps* y *obtenerOcasional*:

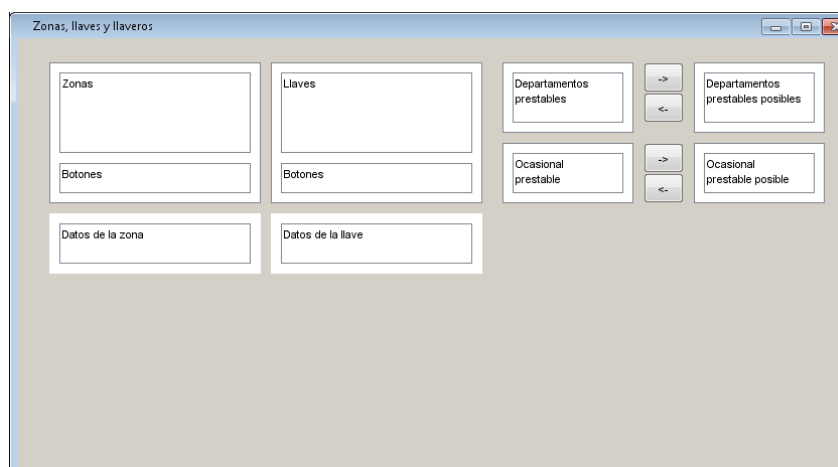
MÉTODO	FUNCIONALIDAD
<i>obtenerEmpDeps</i>	Método que obtiene todos los departamentos (junto con sus empresas) a quienes está permitido prestar la llave que ejecuta el método.
<i>obtenerOcasional</i>	Método similar al anterior, solo que en lugar de buscar los departamentos busca sobre el personal de tipo ocasional.

Métodos a implementar y a añadir a la API

4.7. Caso de uso 72: Añadir llave

Llegamos a un punto en el que el desarrollo gráfico (en lo que respecta a las llaves) será máximo. Echando la vista atrás, a lo que tenemos creado hasta el momento, disponemos de un formulario en el que poder introducir los datos de la llave y dos tablas en las que visualizar los departamentos y personal ocasional que la pueden tomar en préstamo. Tenemos que idear algún mecanismo de insertar elementos a estas tablas, y tras mucho discurrir se nos ocurre que podría ser buena idea idear dos tablas homólogas a éstas pero con contenido “contrario” (todos aquellos departamentos y personal ocasional que en estos momentos no puedan tomar en préstamo la llave seleccionada). Así, mediante cuatro botones de “trasvase” podríamos crear un pequeño juego de inclusión y exclusión de elementos mediante la adición o sustracción de las filas que componen los cuatro *DefaultTableModel*.

Para finalizar, no olvidar el botón que inicie todo el proceso de adición, pues si no todo lo que hagamos no tendrá sentido. Al ser el primer botón que creamos para las llaves, lo que haremos es crear una botonera bajo la tabla de llaves e incluirlo:



Interfaz gráfica propuesta



Tras pulsar el botón de inserción:

1. Obtendremos la zona en la que insertar la llave.
2. Crearemos un objeto de tipo Llave con todos los datos recogidos de la interfaz gráfica (formulario y tablas de prestables).
3. Insertaremos la llave.
4. Insertaremos todos los prestatarios posibles de la llave.
5. Refrescaremos la tabla de llaves para que la recientemente insertada entre a formar parte (se haga visible).

Gran parte de la funcionalidad necesaria para llevar a buen puerto este proceso ya fue desarrollada en la API inicial, pero al no disponer de las claves primarias de empresas y departamentos en las tablas de prestatarios posibles nos vemos obligados a crear métodos que las obtengan a partir de los nombres de empresas y departamentos. Hacemos notar que esto es posible gracias a que ambos atributos son únicos (no pueden existir empresas o departamentos dentro de una empresa con el mismo nombre); de no ser así se debería buscar otra solución. Ambos métodos son:

MÉTODO	FUNCIONALIDAD
<i>obtenerCodEmp</i>	Método que obtiene el código de empresa dado su nombre.
<i>obtenerCodDep</i>	Método que obtiene el código de departamento dado su nombre.

Métodos a implementar y a añadir a la API



Estos diagramas de actividades y de secuencia seguro que ayudan al lector a comprender todo este proceso:

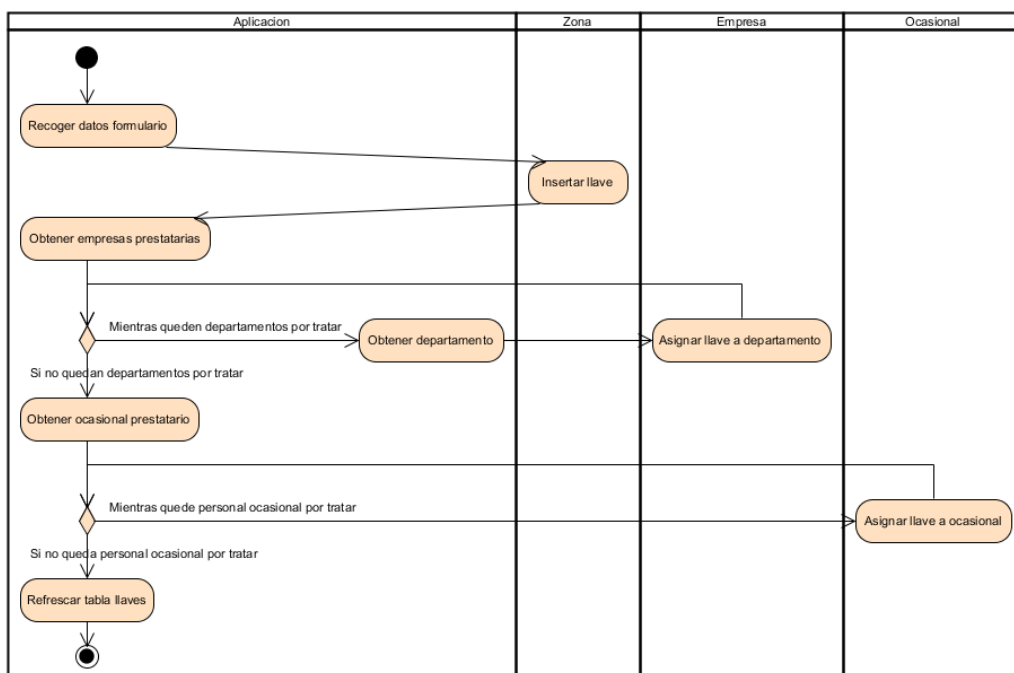


Diagrama de actividades

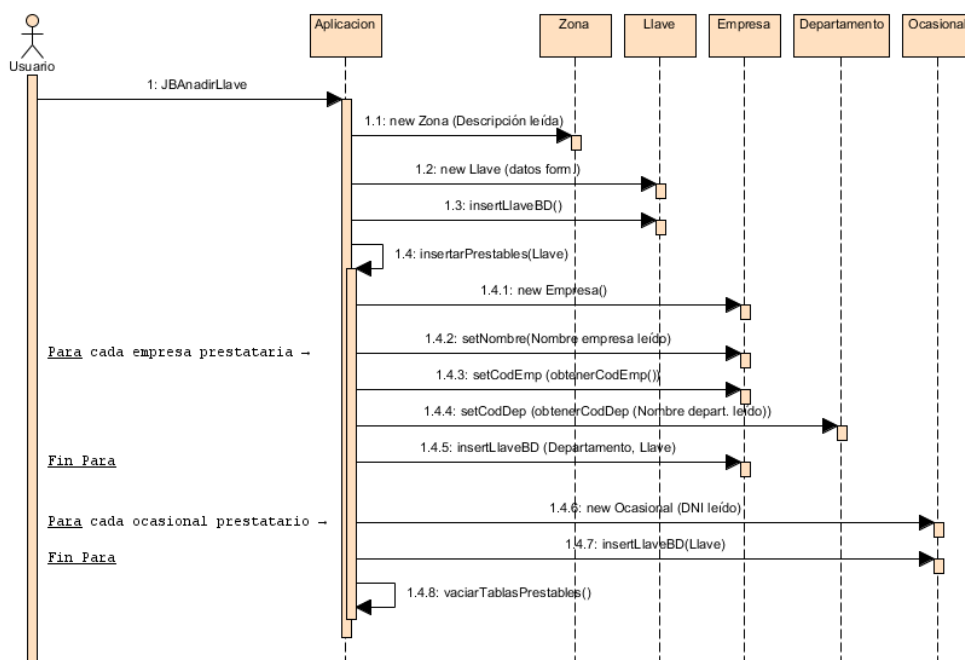


Diagrama de secuencia



4.8. Caso de uso 73: Borrar llave

La eliminación de llaves resultará mucho más sencilla gracias a las claves foráneas que pueblan el esquema relacional y el botón de borrado que añadimos a la botonera.

Borrar una llave es tan sencillo como:

1. Obtener las llaves seleccionadas.
2. Eliminarlas una a una.
3. Refrescar la tabla de llaves y vaciar las de prestables, dejando todo deseleccionado, como si acabásemos de seleccionar la zona.

A la hora de recorrer todas las llaves necesitamos un constructor que nos permita ir recuperándolas con sólo conocer su código:

MÉTODO	FUNCIONALIDAD
<i>Llave</i>	Constructor que crea una Llave tomando como parámetro de entrada su código. En caso de existir varias con el mismo código proporciona la más actual.

Método a implementar y a añadir a la API

Al final, todo el proceso se traduce a:

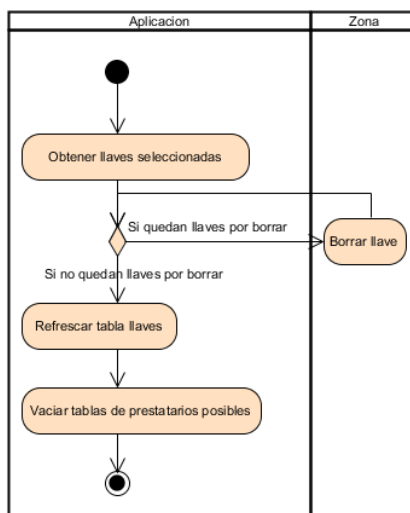


Diagrama de actividades

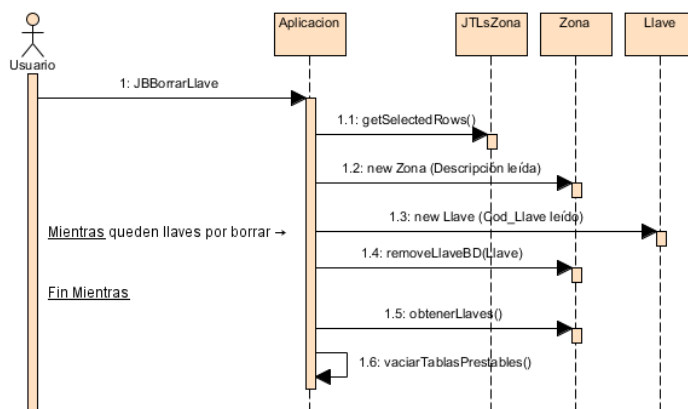


Diagrama de secuencia

4.9. Caso de uso 74: Dar de baja llave

Llegados al caso de uso en que debemos dar de baja una llave no podemos más que repetir lo mismo que en la anterior iteración. Dar de baja un elemento sigue los mismos pasos que eliminarlo, lo único que cambia es el tipo de operación que se hace sobre éste: borrado o baja. Por lo cual, si el proceso es el mismo y la operación a realizar la tenemos ya implementada, sólo sería necesario añadir un botón a la botonera para poder implementar este proceso y diagramas:

1. Obtener las llaves seleccionadas.
2. Dar de baja una a una todas ellas.
3. Refrescar la tabla de llaves y vaciar las de prestables.

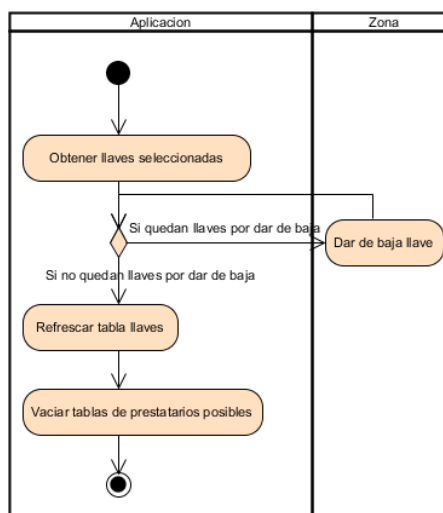


Diagrama de actividades

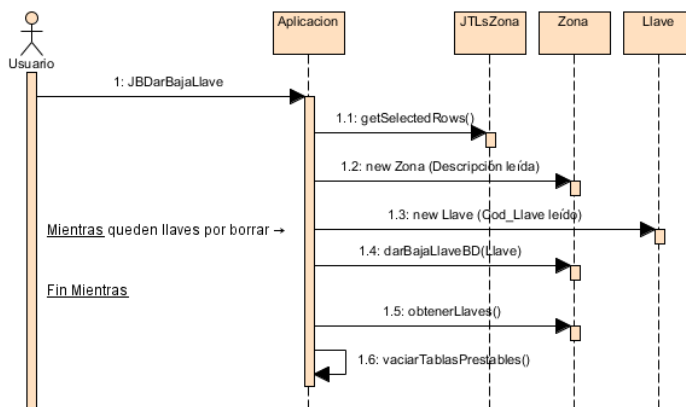


Diagrama de secuencia

4.10. Caso de uso 75: Modificar llave

Todos los componentes gráficos diseñados para dar de alta una llave vendrán fantásticamente en este caso de uso, no siendo necesario crear más que un simple botón que transmita al sistema la intención del usuario de modificar una llave.

Los procesos de modificación diseñados hasta el momento son válidos, solo que en este caso habrá que actualizar los prestatarios posibles:

1. Crear la nueva llave con los datos del formulario.
2. Recuperar la vieja llave de la base de datos.
3. Actualizar la vieja llave con la nueva recientemente creada.
4. Eliminar todos los prestatarios posibles que pudieran existir e insertar los que en estos momentos existan en las tablas de prestatarios posibles.
5. Refrescar la tabla de llaves para que los cambios se hagan visibles en ella.

No es necesario modificar nada en la API para conseguir codificar este flujo de actividades...

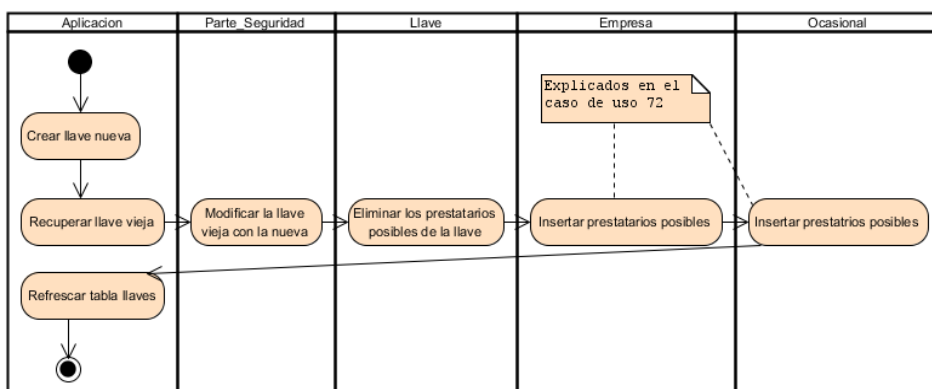


Diagrama de actividades



... y este intercambio de mensajes:

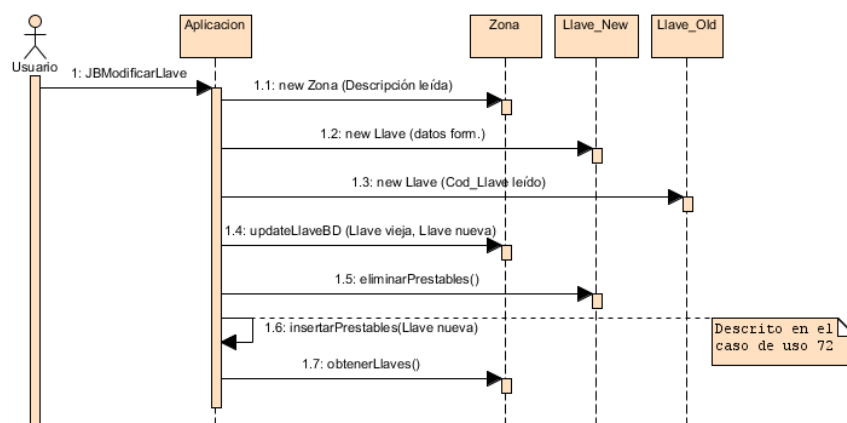
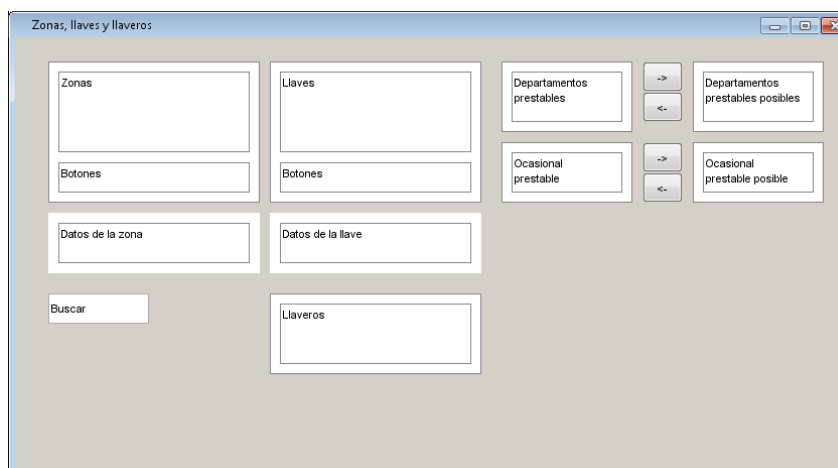


Diagrama de secuencia

4.11. Caso de uso 77: Visualizar llaveros

Aquí comienzan a surgir nuevos componentes gráficos, pues los necesarios para relacionar zonas y llaves ya los tenemos pero los que deben mostrar llaveros no. En un principio optamos por una JTable colocada en la parte inferior de llaves acompañada de un campo que permita realizar búsquedas:



Interfaz gráfica propuesta

La muestra de los llaveros en lo que a su totalidad (datos principales) corresponde no está relacionada con ningún otro elemento, pues no optamos por mostrar las llaves que posee cada llavero (lo haremos en el siguiente caso de uso) y un llavero puede contener llaves de varias zonas (sería ilógico pues relacionarlas con éstas). Esta falta de relación entre entes hace que mostremos



de manera automática todos los llaveros al iniciar la pestaña con el siguiente proceso:

1. Crear un objeto Almacen por defecto.
2. Obtener los llaveros existentes (código y descripción) que pasan el filtro de búsqueda.
3. Refrescar la tabla de llaveros con la información obtenida en el anterior punto.

El método de búsqueda a añadir a la API es:

MÉTODO	FUNCIONALIDAD
<i>obtenerLlaverosTotales</i>	Método que obtiene los principales datos de todos los llaveros existentes en la base de datos que cumplen con el filtro pasado como parámetro de entrada.

Método a implementar y a añadir a la API

El proceso resulta trivial. Aun así, lo acompañamos de estos aclarativos diagramas:

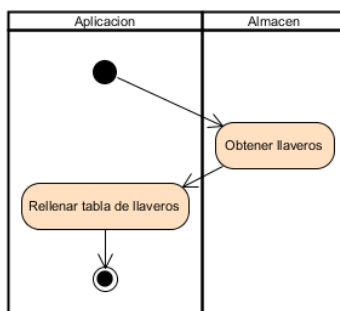


Diagrama de actividades

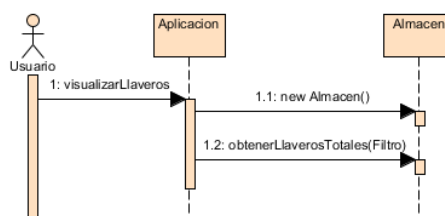


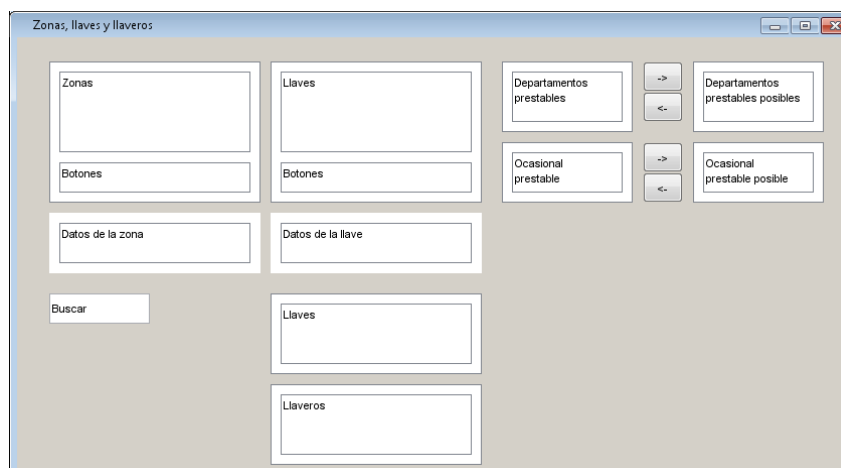
Diagrama de secuencia



4.12. Caso de uso 78: Visualizar llavero

Si con el caso de uso anterior conseguimos obtener el código y descripción de todos los llaveros, sólo nos faltaría conocer las llaves que los conforman. ¿Cómo podemos mostrarlas? Con una JTable, evidentemente. Su ubicación podría ser indistinta, pero optaremos por desplazar la tabla de llaveros hacia abajo y colocar la de llaves en el lugar que deja libre. ¿Por qué? Si atendemos a los futuros casos de uso, deberemos mostrar el contenido de los almacenes de algún modo, y reutilizar estas tablas en esta disposición será una buena idea. De todos modos, dediquémonos al caso de uso que nos atañe y posteriormente ya volveremos a lo que acabamos de comentar.

La interfaz gráfica de la pestaña de zonas quedaría pues de la siguiente forma:



Interfaz gráfica propuesta

Una cosa debe quedar muy clara. Toda inserción, borrado, actualización y actualización que pudiera haberse dado y que se dará en la tabla de llaves asociada a las zonas también deberá realizarse en esta nueva tabla de llaves. Esto hace que debamos revisar las actualizaciones de tablas realizadas en anteriores casos de uso y modificar las pertinentes para que también se realicen en esta tabla. En futuros casos de uso se tendrán en cuenta ambas tablas.

Con todo ésto, si pulsamos cualquier llavero de la tabla, la de llaves mostrará las que pertenecen a éste de la siguiente forma:

1. Recuperar el llavero seleccionado a través de su código.
2. Obtener las llaves que lo componen a través de su método *obtenerLlaves* y el filtro recogido del campo de texto.
3. Refrescar la tabla de llaves con los resultados obtenidos de la búsqueda anterior.



No disponemos del método *obtenerLlaves*, con que habrá que añadirlo a nuestra API:

MÉTODO	FUNCIONALIDAD
<i>obtenerLlaves</i>	Método que obtiene el conjunto de llaves que forman parte del llavero que lo llama y que pasan el filtro que éste recoge como parámetro de entrada.

Método a implementar y a añadir a la API

Si llevamos la estrategia diseñada a un nivel mayor de detalle y concreción, obtenemos estos diagramas:

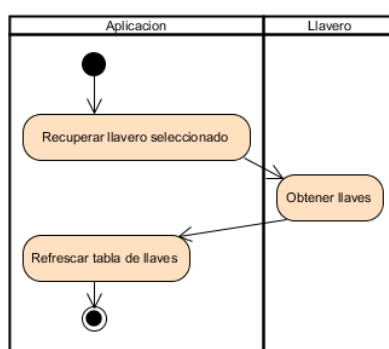


Diagrama de actividades

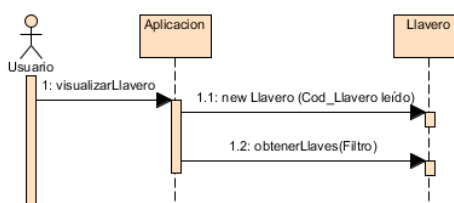


Diagrama de secuencia

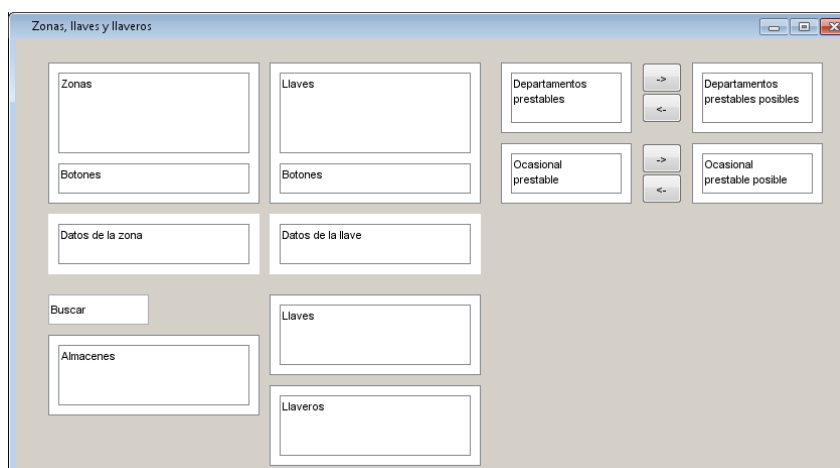
4.13. Caso de uso 79: Añadir llavero

Añadir un llavero consiste en agrupar un conjunto de llaves bajo un código y una descripción. Los dos últimos pueden definirse en dos campos de texto, el problema viene a la hora de definir las llaves que lo conformarán.

Un llavero puede estar formado por dos o más llaves pertenecientes a una o varias zonas. Estudiando el resto de casos de uso nos encontramos que el usuario debe poder visualizar todos los llaveros y su contenido, con que reutilizar la tabla de llaves diseñada para el segundo caso de modo que muestre todas las llaves existentes puede ser una buena idea. Si desarrollamos el comienzo del caso de uso 87 de modo que podamos listar todas las llaves, seremos capaces de desarrollar éste.



Comenzamos creando una tabla que pueda recoger todos los almacenes existentes:



Interfaz gráfica propuesta

Nos proponemos pues que 'Almacenes' contenga el almacén 'Todos' que se encargará de mostrar en el inicio todas las llaves (a desarrollar) y todos los llaveros (desarrollado hace dos casos de uso). Para ello, solo debemos ampliar el diseño creado en el caso de uso 77 de la siguiente forma:

1. Crear un objeto Almacen por defecto.
2. Obtener las llaves existentes (código, zona y descripción) que pasan el filtro de búsqueda a través de éste.
3. Obtener los llaveros existentes (código y descripción) que pasan el filtro de búsqueda.
4. Refreshar la tabla de llaves con la información obtenida en el anterior punto.
5. Refreshar la tabla de llaveros con la información obtenida en el punto 3.

El método homólogo a *obtenerLlaverosTotales* pero tratando las llaves puede ser siguiente:

MÉTODO	FUNCIONALIDAD
<i>obtenerLlavesTotales</i>	Método que obtiene los principales datos de todas las llaves registradas en el sistema y que cumplen con el filtro proporcionado.

Método a implementar y a añadir a la API



Podemos plasmar este proceso en los siguientes diagramas:

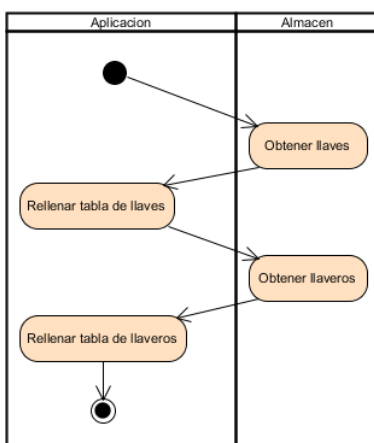


Diagrama de actividades

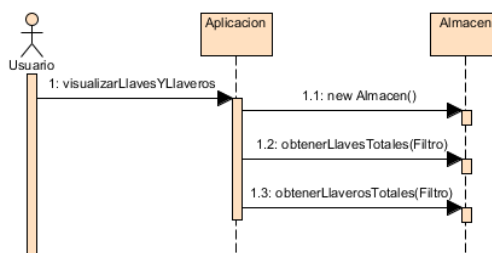


Diagrama de secuencia

Si todo sucede según lo previsto, dispondríamos ya ambas tablas rellenas con todas las llaves y llaveros existentes. En este punto podemos crear un formulario con el que definir los atributos de los llaveros y una botonera bajo el panel de llaveros que nos permita operar sobre ellos, añadiéndole un botón de adición que sólo esté operativo sí y solo sí el panel de llaves muestra el conjunto total de llaves y se encuentran seleccionadas dos o más de ellas:

Interfaz gráfica propuesta



La selección de un conjunto de llaves, edición de los datos del llavero y pulsación del botón de inserción desencadenará la siguiente secuencia de actividades:

1. Crear un objeto de tipo Llavero con los datos del formulario e introducir en él todas las llaves seleccionadas en la tabla.
2. Insertar el llavero en la base de datos mediante el método *insertLlaveroBD*.
3. Refrescar la tabla de llaveros para que muestre el nuevo.
4. Desmarcar cualquier llave seleccionada en la tabla de llaves.

Puede ayudar mucho observar todo este flujo de actividades a través de un diagrama de actividades...

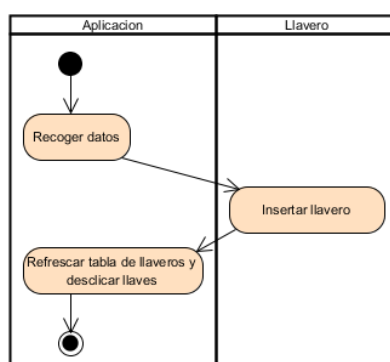


Diagrama de actividades

... y otro de secuencia:

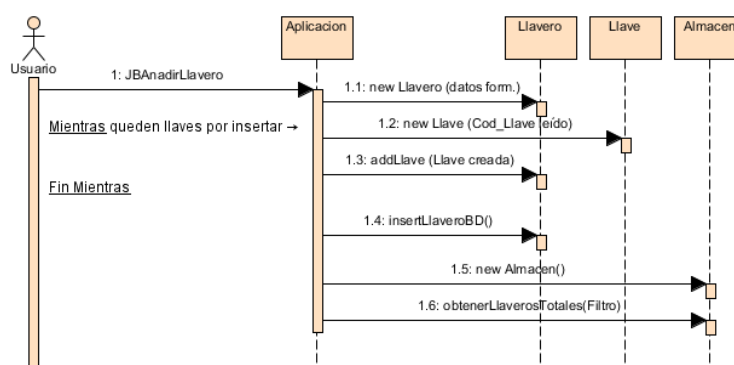


Diagrama de secuencia



4.14. Caso de uso 80: Borrar llavero

Borrar un llavero resulta mucho más sencillo que crearlo, pues al igual que ocurrió en otras ocasiones, sólo con el hecho de crear un constructor que pueda recuperar llaveros a partir de su código tendremos los mecanismos necesarios para tal fin.

Para empezar, como siempre, los componentes gráficos. Tenemos la tabla en la que visualizar los llaveros, con que solo nos resta añadir un botón de borrado a la botonera.

La estrategia: la misma empleada en anteriores borrados pero variando los objetos que la ejecutan:

1. Obtener los llaveros seleccionados.
2. Recorrerlos y eliminarlos uno a uno.
3. Refrescar la tabla de llaveros para que todos los borrados dejen de formar parte de ésta.

Este es el constructor a definir en la API:

MÉTODO	FUNCIONALIDAD
<i>Llavero</i>	Constructor que crea un objeto Llavero a través de la información almacenada en la base de datos y el código proporcionado. En caso de que existan dos o más llaveros con el mismo código, siempre devolverá el llavero más reciente.

Método a implementar y a añadir a la API

Ayuda mucho visualizar todo mediante diagramas de actividades y de secuencia:

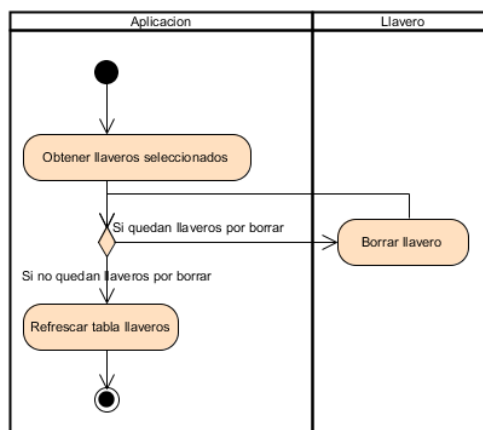
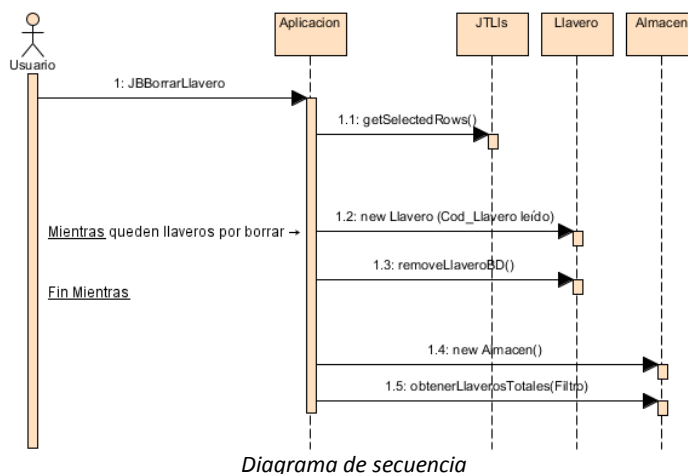


Diagrama de actividades



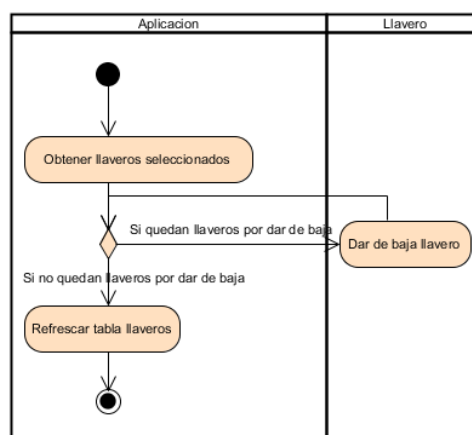
4.15. Caso de uso 81: Dar de baja llavero

Nos encontramos ante otro caso de uso en el que solo es necesario añadir un botón para que éste se desarrolle:

1. Obtener los llaveros seleccionados.
2. Dar de baja uno a uno todos los llaveros obtenidos en el punto anterior.
3. Refrescar la tabla de llaveros para que solo aparezcan aquellos que estén operativos.

Como se puede apreciar, la estrategia es idéntica a la utilizada en el caso de uso anterior salvo por la operación a realizar, que esta ocasión será dar de baja.

Se trata de un proceso bastante sencillo de entender y codificar, ya que la operación de dar de baja está implementada desde que se desarrolló la API inicial y las estructuras de control son idénticas a las del caso de uso 80. De todos modos, realizar los diagramas de actividades y de secuencia siempre ayuda mucho a asentar todo el diseño ideado:



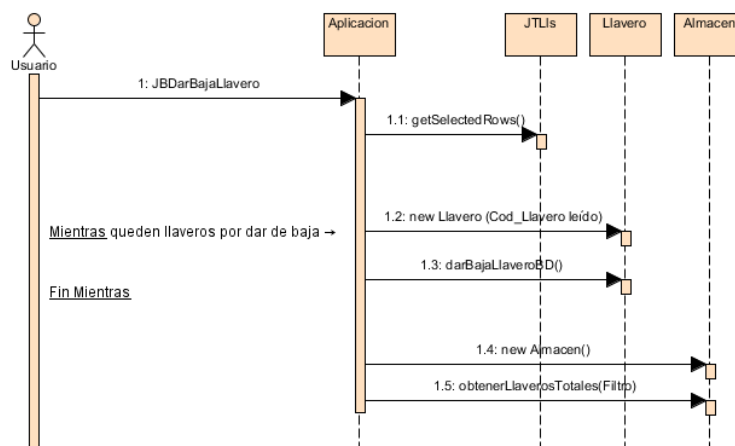


Diagrama de secuencia

4.16. Caso de uso 82: Modificar llavero

La actualización de un llavero bien puede darse por la modificación de sus datos (código o descripción) o por la composición del mismo o por ambas cosas a la vez. En el primero de los casos (el que ahora tratamos) atenderemos única y exclusivamente a la tabla 'Llaveros', modificando alguna de las tuplas que la pueblan de la siguiente forma:

1. Creamos un objeto Llavero con los datos recogidos del formulario.
2. Recogemos el llavero viejo seleccionado a través de su código.
3. Actualizamos el llavero viejo con el nuevo (solo los datos, no sus llaves).
4. Refreshamos la tabla de llaveros para que los cambios efectuados se hagan visibles.

En este caso, otra posible solución podría ser asignar al llavero nuevo las llaves del viejo y actualizar completamente el viejo con el nuevo (datos y llaves), consiguiendo el mismo resultado pero con una menor eficiencia, ya que trataría todas las llaves, acción que resulta innecesaria.



En un principio no estimamos que sea necesario crear ningún método nuevo en la API. Eso sí, es necesario añadir a la botonera un botón de modificación que inicie todo el proceso antes descrito y bajo estas líneas definido:

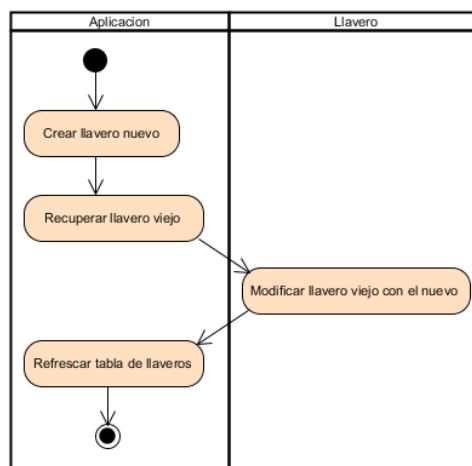


Diagrama de actividades

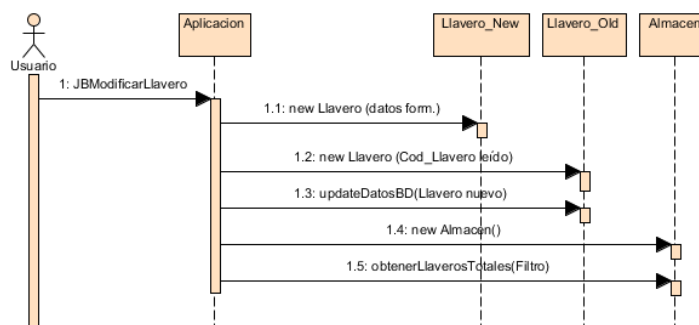


Diagrama de secuencia

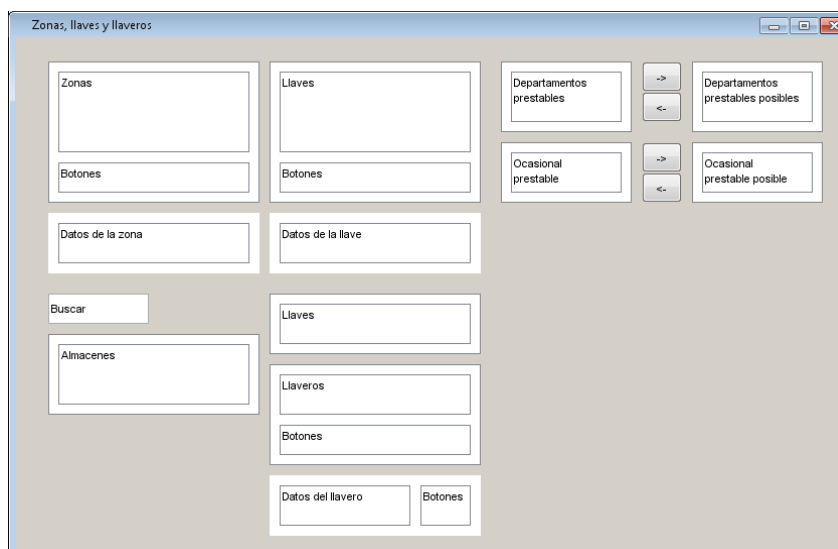
4.17. Caso de uso 83: Añadir llave

Con éste y el siguiente caso de uso tratamos la parte de modificación del llavero que restaba: modificar su contenido, no sus datos.

Al seleccionar un llavero para su visualización decidimos mostrar las llaves que lo componen en la tabla que queda justo encima de la de llaveros. Esto nos lleva a que las llaves deban de ser introducidas desde la tabla de llaves asociada a las zonas, a menos que queramos idear una nueva tabla. La segunda opción es rechazada ya que otra tabla más que conllevaría más desventajas (sobrecarga de componentes gráficos, complejidad de uso,...) que ventajas (poder introducir llaves de varias zonas a la vez). Por lo general los llaveros tienden a estar compuestos con llaves de la misma zona, pero de no ser así el usuario siempre puede realizar la operación en dos ciclos y así evitar tener que trabajar con una tabla más. Todo esto nos lleva a la conclusión de que utilizando la tabla de llaves pertenecientes a zonas solo necesitaremos un simple botón para añadir las llaves seleccionadas en ésta al llavero que se



encuentre seleccionado. Para no confundir los botones de adición de llaveros y de adición de llaves, colocamos este último junto al formulario de datos:



Interfaz gráfica propuesta

La estrategia a seguir es:

1. Recuperar dos instancias del llavero seleccionado (el viejo y el nuevo).
2. Añadir al llavero nuevo todas las llaves seleccionadas en la tabla de llaves correspondiente a las zonas.
3. Actualizar el llavero viejo con el nuevo, de modo que las llaves de éste pasen a formar parte del primero.
4. La tabla de llaveros no cambia pero sí la de llaves, con que procede refrescarla.

No creemos que pueda surgir ningún tipo de duda sobre las operaciones a realizar para plasmar la estrategia en código. De todos modos, aquí van los diagramas de actividades y de secuencia:

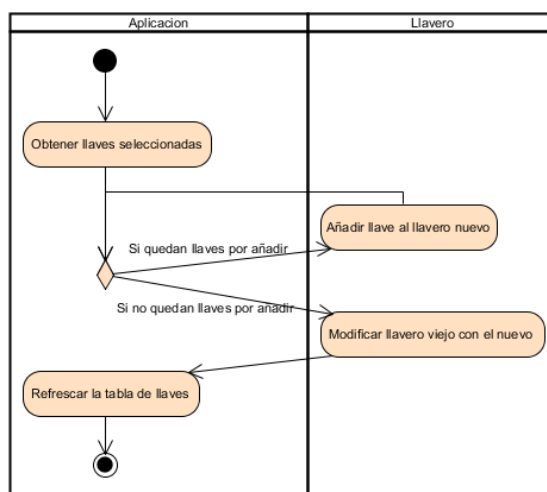
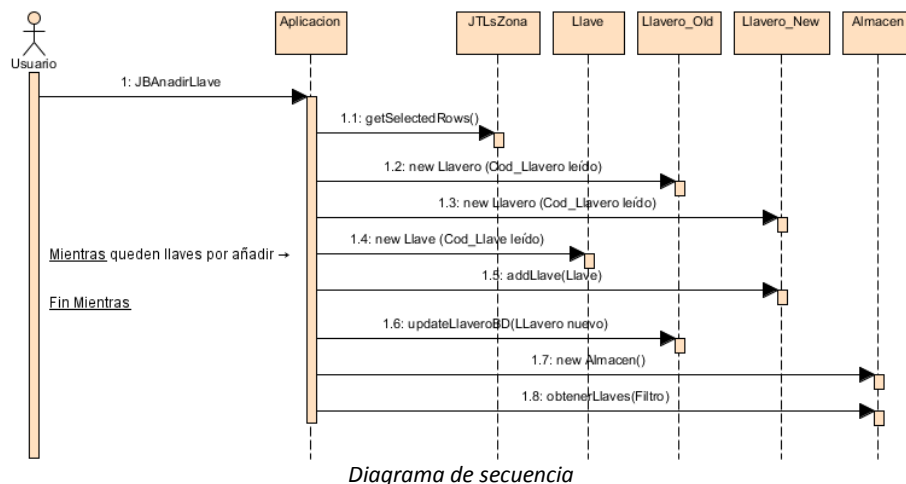


Diagrama de actividades



4.18. Caso de uso 84: Borrar llave

Para borrar llaves seguiremos el mismo proceso que el diseñado para añadirlas, solo que cambiaremos la operación de añadir (*addLlave*) por la de eliminar (*removeLlave*). No creemos preciso explicar nada más, ya que los métodos y estructuras de control están diseñadas (no es necesario implementar nada nuevo en la API); simplemente haremos constar que es necesario añadir un botón de borrado junto al de adición.

Recordamos el flujo de actividades y diagramas a seguir por si el lector lo encuentra de interés:

1. Recuperar dos instancias del llavero seleccionado.
2. Eliminar del llavero nuevo todas aquellas llaves seleccionadas en la tabla de llaves de la sección de almacenes (no la de llaves asociadas a una zona).
3. Actualizar el llavero viejo con el nuevo, de modo que las llaves eliminadas también sean eliminadas en la base de datos.
4. Refrescar las tablas de almacenes.

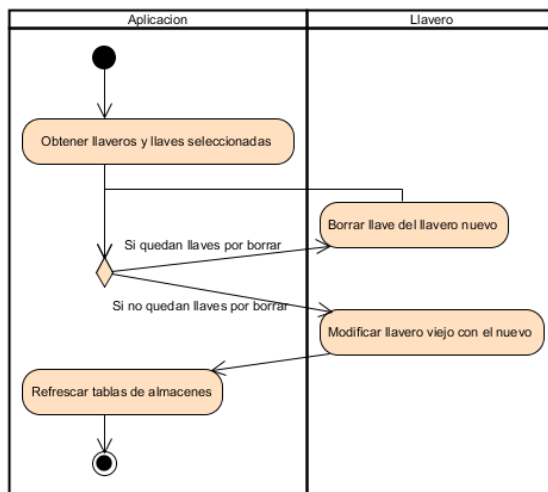


Diagrama de actividades

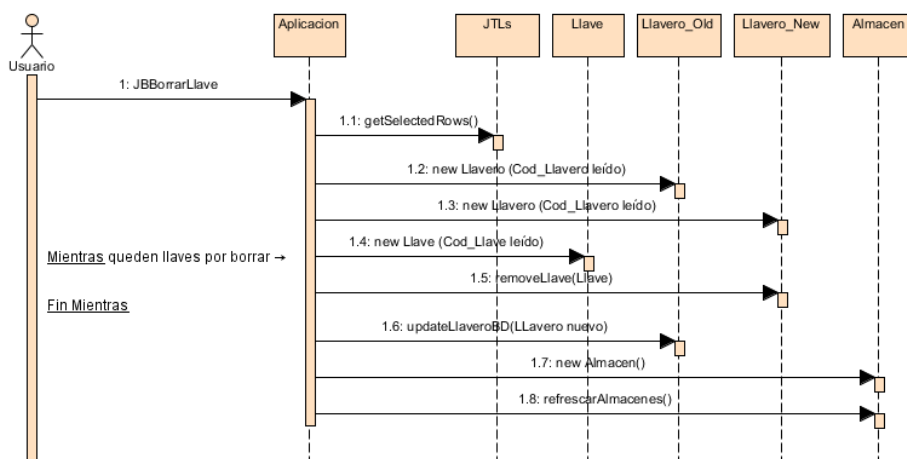


Diagrama de secuencia

4.19. Caso de uso 86: Visualizar almacenes de llaves

Entramos en el último grupo de casos de uso a implementar en esta iteración. Para comenzar nos centramos en la visualización de los almacenes. El caso de uso 79 nos obligó a crear la tabla de almacenes con un único almacén existente ('Todos'), con que la parte gráfica ya la tenemos diseñada.

¿Qué restaría? Hacer que la tabla muestre no solo 'Todos', sino que además sume a este almacén todos los existentes en el sistema. Para ello hará falta seguir esta secuencia de actividades:

1. Crear un objeto de tipo Almacen.
2. Realizar la búsqueda de los almacenes existentes que pasan el filtro y unirlos a 'Todos'.
3. Refrescar la tabla de almacenes con los resultados obtenidos en el punto anterior.

Este proceso viene enmarcado en el proceso de inicialización comenzado en el caso de uso 77 y continuado en el 79. Acabamos de completarlo quedando de la siguiente manera:

1. Crear un objeto Almacen por defecto.
2. Obtener los almacenes existentes (descripción) que pasan el filtro de búsqueda a través de éste.
3. Obtener las llaves existentes (código, zona y descripción) que pasan el filtro de búsqueda.
4. Obtener los llaveros existentes (código y descripción) que pasan el filtro de búsqueda.
5. Refrescar la tabla de almacenes con la información obtenida en el punto 2.
6. Refrescar la tabla de llaves con la información obtenida en el tercer punto.
7. Refrescar la tabla de llaveros con la información obtenida en el punto número 4.



Estos diagramas ilustran el proceso de inicialización de la tab 'Zonas, llaves y llaveros' en lo que corresponde a la parte de almacenes (mitad inferior de la pestaña):

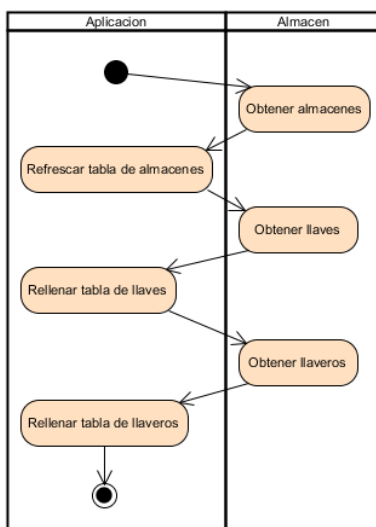


Diagrama de actividades

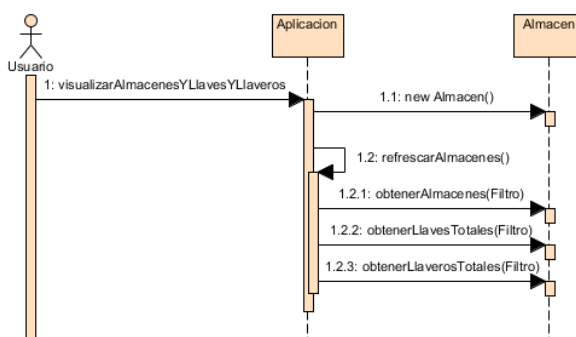


Diagrama de secuencia

Nota: Agrupamos todas las operaciones bajo refrescarAlmacenes.



4.20. Caso de uso 87: Visualizar almacén de llaves

Era imposible poder mostrar de golpe el contenido de todos los almacenes en la interfaz, con que hemos visto mucho más lógico hacer que cada vez que seleccionemos uno de la tabla de almacenes, las de llaves y llaveros que la acompañan se actualicen según lo que guarde el almacén.

El modo de actuar es el que sigue:

- a. Si se trata del almacén 'Todos':
 1. Creamos un Almacen y le asignamos de código 'TODO' y de descripción 'Todos'.
 2. Refreshamos la tabla de llaves con el resultado obtenido de llamar a *obtenerLlavesTotales*.
 3. Refreshamos la tabla de llaveros con el resultado obtenido de llamar a *obtenerLlaverosTotales*.
- b. En caso de tratarse de cualquier otro almacén individual/concreto:
 1. Creamos un Almacen con la descripción del mismo.
 2. Rellenamos la tabla de llaves con los resultados obtenidos al llamar a *obtenerLlaves*.
 3. Rellenamos la tabla de llaveros con los resultados obtenidos al llamar a *obtenerLlaveros*.

Cosas a tener muy en cuenta:

- *obtenerLlavesTotales* y *obtenerLlaverosTotales* son métodos que muestran toda la información de llaves y llaveros comentada hasta el momento, pero deberemos modificar la consulta para que además ofrezcan el número de copias existentes y las que están en préstamo de ambos elementos.
- *obtenerLlaves* y *obtenerLlaveros* mostrarán los atributos básicos (sin recuentos) de llaves y llaveros existentes en un almacén concreto. No debemos confundir el primer método con el de mismo nombre existente en la clase Llaveros. Uno mostrará las llaves existentes en el almacén mientras que el otro mostrará las llaves existentes en un llaveros.
- Los almacenes poseen descripciones únicas y debido a que la tabla de almacenes muestra solo las descripciones de éstos, crearemos un constructor que cree objetos de tipo Almacen a partir de su descripción y los datos residentes en la base de datos.

Debemos pues, modificar los dos primeros métodos (consultas SQL) y añadir los dos restantes y el nuevo constructor a la API.



Se clarifica mucho el caso de uso si se atiende a los diagramas de actividades y de secuencia:

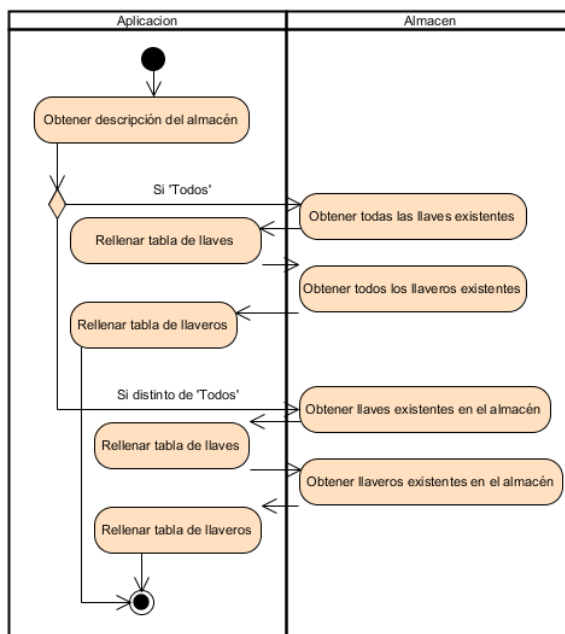


Diagrama de actividades

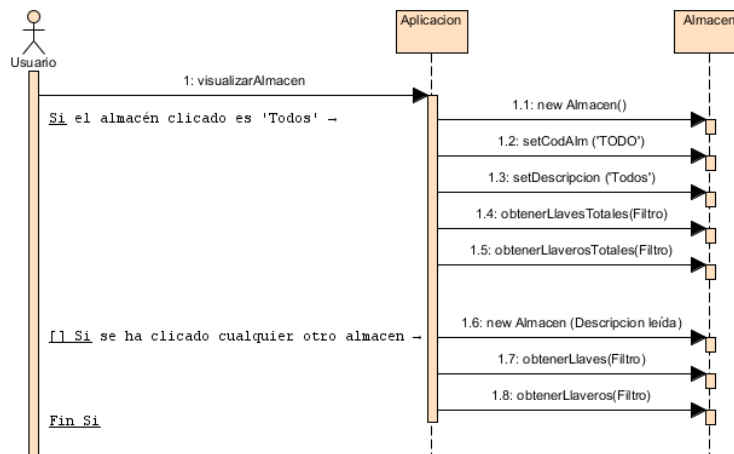


Diagrama de secuencia



4.21. Caso de uso 88: Añadir almacén de llaves

Un almacén de llaves está compuesto por un código, una descripción y todo el material que pueda albergar, pero como bien hicimos a la hora de dividir y describir los casos de uso, trataremos en éste la creación de un almacén vacío y en los siguientes otorgaremos al usuario la posibilidad de modificarlo a su antojo.

Como componentes gráficos nuevos debemos añadir a la interfaz una botonera que nos permita manipular los almacenes existentes, bien sea para añadir nuevos, para eliminarlos, modificarlos,... y un formulario que nos permita obtener y enviar datos al sistema:

Interfaz gráfica propuesta

Insertar un nuevo almacén es un proceso trivial:

1. Crear un objeto de tipo Almacen con los datos recogidos del formulario (código y descripción).
2. Insertar el objeto en la base de datos.
3. Refrescar todas las tablas correspondiente a los almacenes, mostrando los elementos iniciales (almacén 'Todos').



Plasmar este proceso en los correspondientes diagramas de diseño da como resultado:

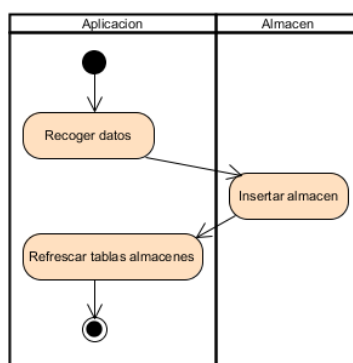


Diagrama de actividades

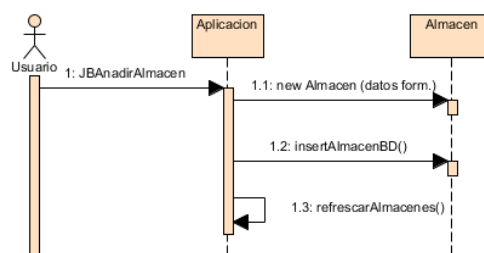


Diagrama de secuencia

4.22. Caso de uso 89: Borrar almacén de llaves

Un único botón añadido a la botonera de la tabla de almacen podría ser el desencadenante de este proceso:

1. Obtener los almacen seleccionados.
2. Recorrerlos y eliminarlos uno a uno.
3. Refrescar todas las tablas correspondientes a los almacen de modo que reflejen el estado real actual de la base de datos.

Como vemos, se trata del mismo proceso de borrado que el utilizado para anteriores entes y el constructor que siempre debíamos construir en este punto (en nuestro caso sería el de Almacén, tomando como entrada su descripción) ya ha sido creado con anterioridad.



Con todos los elementos diseñados y creados, no debería resultar complicado codificar los siguientes diagramas de actividades y de secuencia:

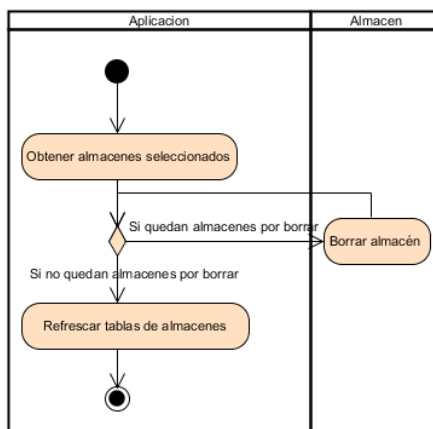


Diagrama de actividades

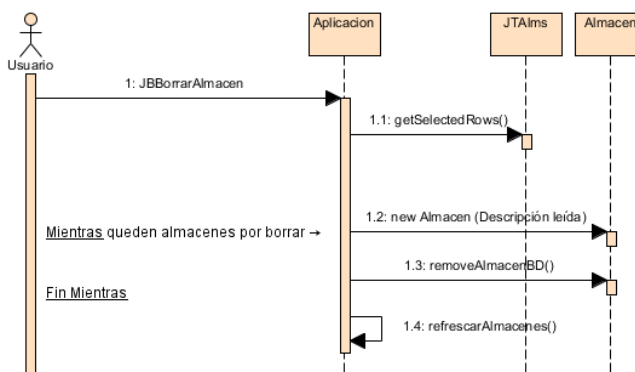


Diagrama de secuencia



4.23. Caso de uso 90: Modificar almacén de llaves

Aquí también tenemos que recordar lo mismo que en el caso de uso 88: diseñaremos a continuación cómo modificar los datos del almacén, no su contenido (realizado en los siguientes casos de uso).

Lo que vamos a hacer es añadir un nuevo botón a los dos existentes en la botonera de almacenes para que el proceso de actualización se inicie:

1. Crear un nuevo Almacen con los datos recogidos del formulario.
2. Recuperar de la base de datos el Almacen a editar.
3. Actualizar este almacén con el nuevo.
4. Refrescar las tablas de almacenes ya que si no los cambios serán efectivos pero no visibles.

Como siempre, creamos un diagrama de actividades...

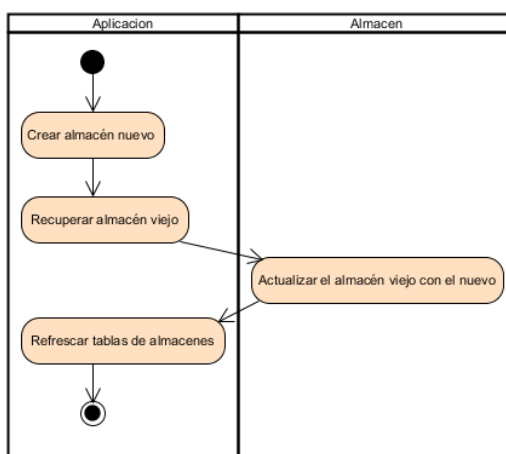


Diagrama de actividades

... y otro de secuencia:

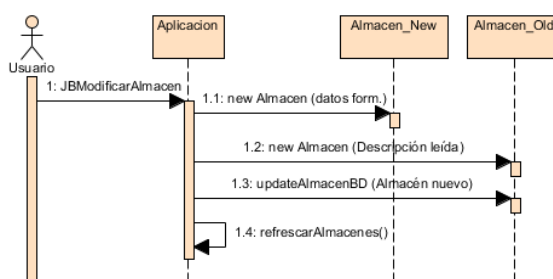


Diagrama de secuencia



4.24. Caso de uso 91: Añadir llave

Comenzamos los casos de uso que permitirán definir el contenido exacto que posee un almacén, comenzando por las llaves que pueden poblarlo y pasando posteriormente a los llaveros.

El haber diseñado la visualización del contenido del almacén sobre las tablas de llaves y llaveros hace que la única vía existente para añadir llaves sea la de seleccionarlas de la tabla de llaves asociada a las zonas (mismo problema que teníamos al insertar llaves en un llavero). El único componente que restaría sería tener una vía para iniciar el caso de uso y otra para indicar cuántas copias de la llave se introducen en el almacén, ambas remediadas con un botón y un campo de texto que quedarán a la derecha de la tabla de llaves:

Interfaz gráfica propuesta

Tras seleccionar el usuario un almacén, una llave e indicar el número de copias a insertar, el sistema deberá:

1. Obtener el almacén, llave y número de copias a insertar.
 - a. Si la llave existe:
 1. Obtener dos instancias GLL_Copias de la llave.
 2. Añadir a la instancia nueva el número de copias.
 3. Actualizar el GLL_Copias viejo con el nuevo.
 - b. Si la llave no existe:
 1. Crear un GLL_Copias nuevo con la llave y el número de copias leídos.
 2. Insertar el nuevo GLL_Copias en el almacén.
2. Refrescar la tabla de llaves del almacén.



Esta estrategia no requiere añadir ningún método a la API que disponemos. El proceso resulta un poco más complicado que los últimos diseñados, pero estudiando sus diagramas la complejidad se reduce drásticamente:

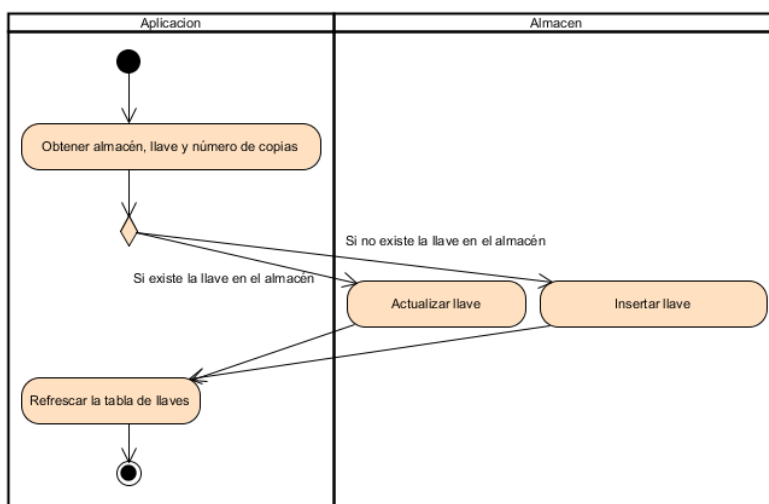


Diagrama de actividades

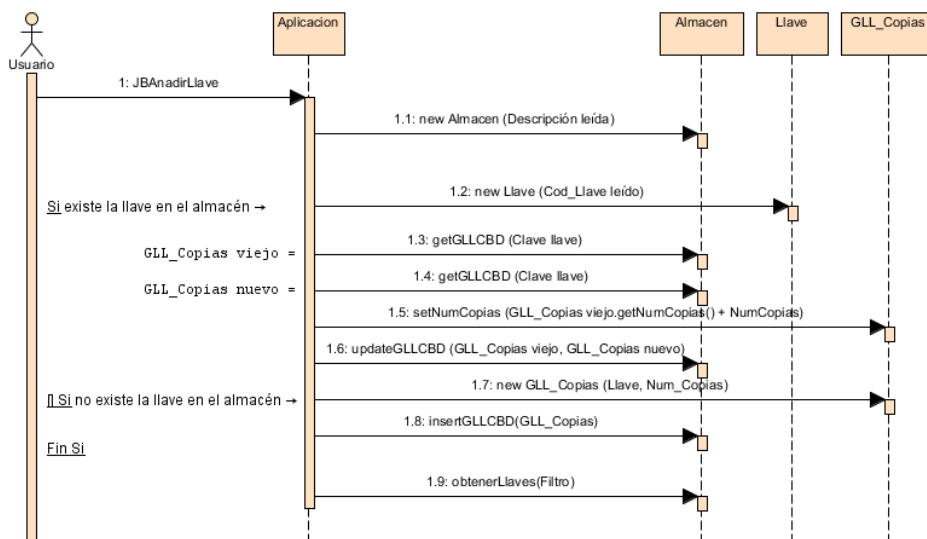


Diagrama de secuencia



4.25. Caso de uso 92: Borrar llave

El borrado de llaves necesita de un nuevo botón en la interfaz gráfica para que el sistema pueda:

1. Obtener el almacén y las llaves seleccionadas (en este caso sí emplearemos la tabla de llaves de la sección de almacenes).
2. Recorrer las llaves eliminándolas una a una del almacén.
3. Refrescar las tablas de almacenes y llaves asociadas a las zonas.

Disponemos de los métodos de obtención de almacén, selección de llaves, recorrido y borrado, con que no es necesario variar el contenido de la API para codificar estos diagramas:

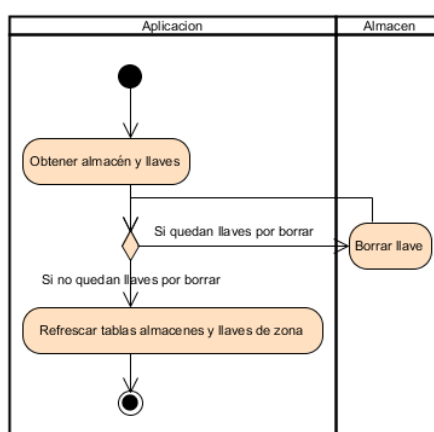


Diagrama de actividades

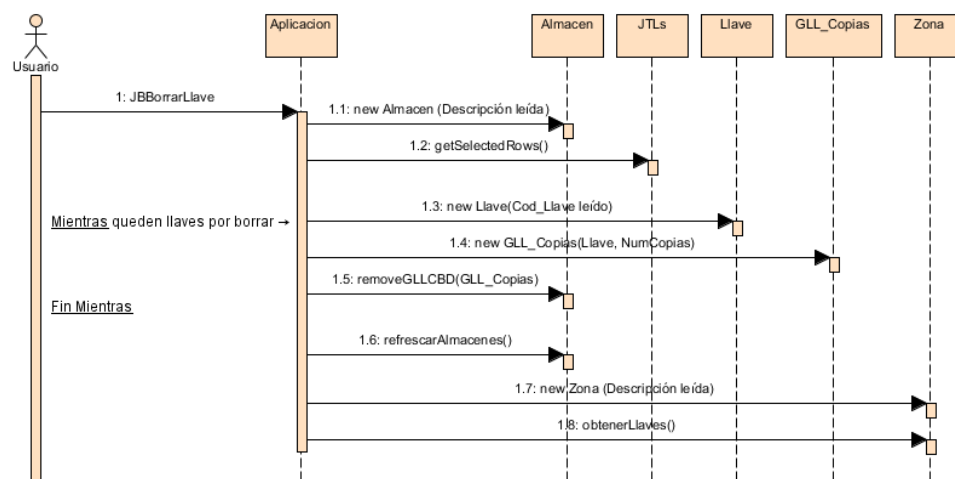


Diagrama de secuencia



4.26. Caso de uso 93: Modificar copias llave

Para conseguir modificar las copias disponibles de una llave en un almacén no necesitamos crear ningún componente gráfico nuevo. Si nos fijamos, con poder seleccionar la llave e indicar el nuevo número de copias bastaría, y eso podemos hacerlo con la tabla de llaves y el campo de texto ideado para añadir llaves. Solo nos falta una cosa: el botón que indique al sistema que debe realizar la actualización.

La modificación puede realizarse de la siguiente forma:

1. Obtener el almacén, la llave y el número de copias a asignarle.
2. Obtener dos instancias GLL_Copias de la llave.
3. Asignar a la instancia nueva el número de copias leído.
4. Actualizar el GLL_Copias viejo con el nuevo.
5. Refrescar la tabla de llaves para que la llave modificada conste con el número de copias asignado.

Si echamos la vista atrás podremos percatarnos de que se trata del mismo proceso que el que utilizamos para añadir una llave existente, utilizando el mismo método pero obteniendo un resultado distinto debido a la implementación del método de actualización creado en la API.

Con todo lo necesario ya desarrollado, solo falta codificar las actividades y pasos de mensajes definidos en estos dos diagramas:

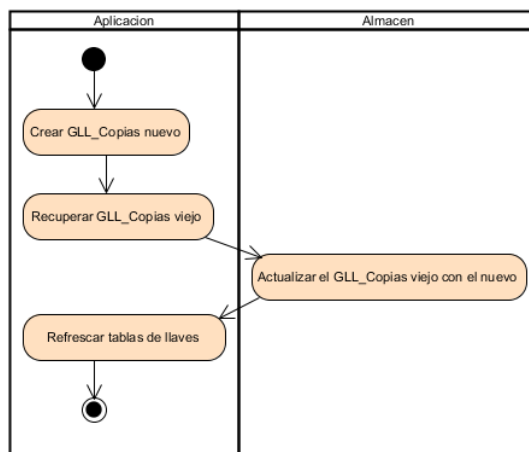


Diagrama de actividades

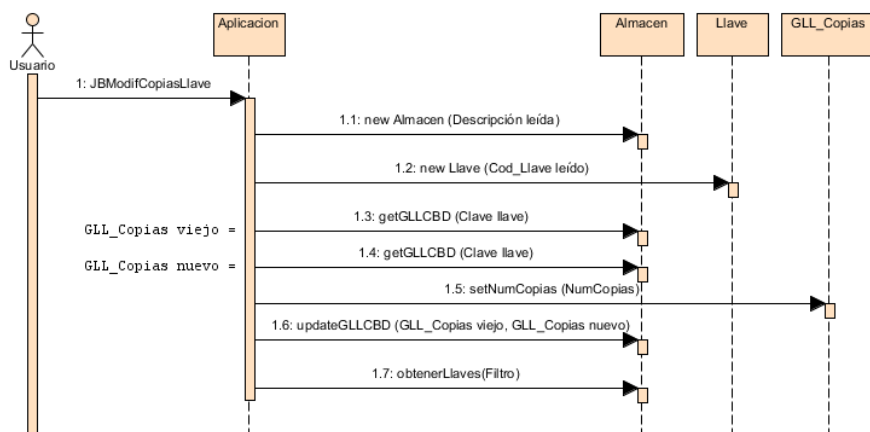
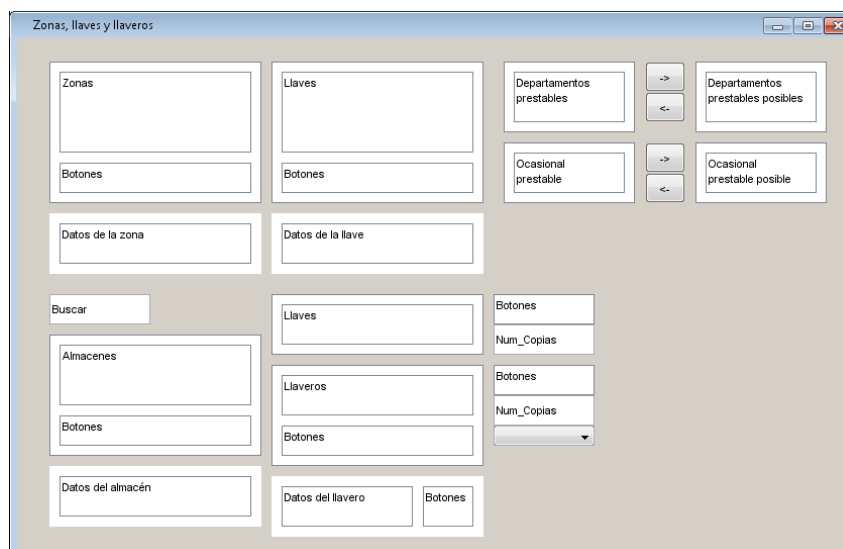


Diagrama de secuencia

4.27. Caso de uso 94: Añadir llavero

Las formas de proceder son muy similares en el caso de los llaveros, pero no iguales. Para añadir una llave podíamos hacer uso de la tabla de llaves asociada a las zonas, pero en este caso no tenemos ninguna otra tabla de llaveros en la que poder apoyarnos. Tampoco disponemos de un campo en el que poder indicar el número de copias a introducir en un almacén ni del botón de adición, con que iniciaremos el diseño de este caso de uso añadiendo a la interfaz gráfica un JComboBox con el que seleccionar el llavero a introducir, un JTextField para el número de copias y una botonera con un botón inicial que inicie el desarrollo de sucesos:



Interfaz gráfica propuesta



La inserción de un llavero en un almacén podría ser implementada de la siguiente forma:

1. Obtener el almacén, llavero y número de copias indicadas por el usuario.
 - a. Si el llavero existe:
 1. Obtener dos instancias GLL_Copias del llavero.
 2. Añadir a la instancia nueva el número de copias.
 3. Actualizar el GLL_Copias viejo con el nuevo.
 - b. Si el llavero no existe:
 1. Crear un GLL_Copias nuevo con el llavero y el número de copias.
 2. Insertar el nuevo GLL_Copias en el almacén.
2. Refrescar la tabla de llaveros del almacén, para que los datos que constan en ella sean los actuales.

Los diagramas son calcados a los del caso de uso 91, solo que en lugar de introducir llaves introduciremos llaveros y en lugar de seleccionar desde una tabla lo haremos desde un JComboBox:

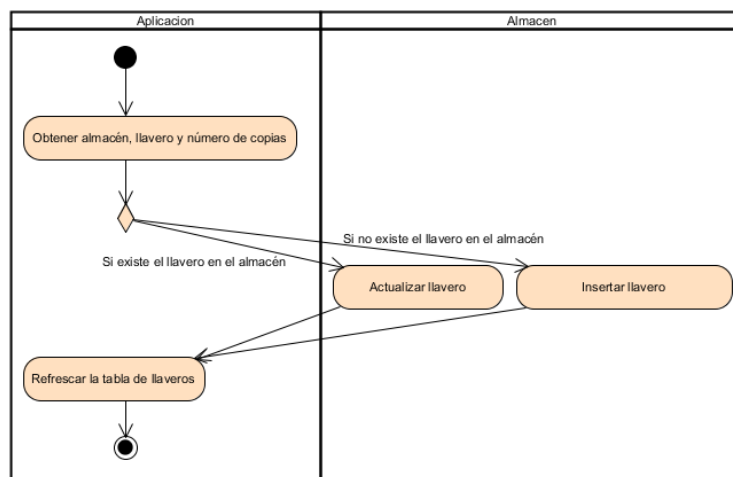


Diagrama de actividades

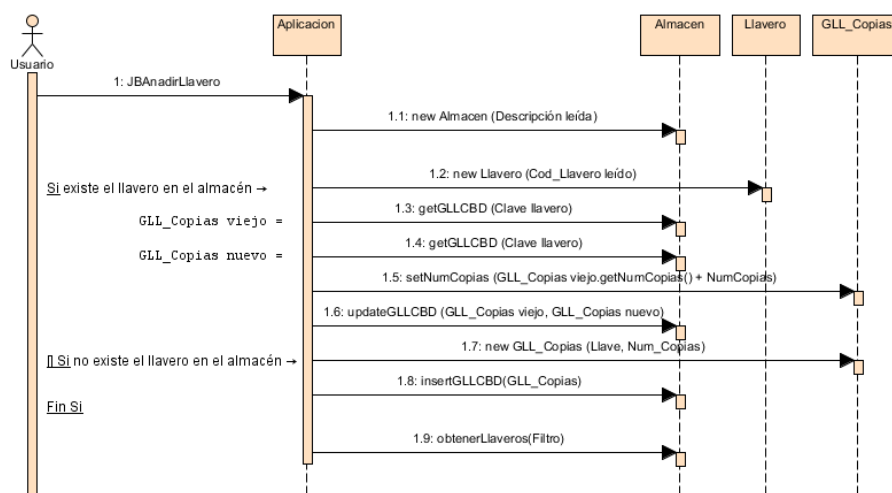


Diagrama de secuencia



4.28. Caso de uso 95: Borrar llavero

Así como diseñamos una estrategia para añadir llaveros a los almacenes, también debemos crearla para borrarlos de ellos. El borrado de llaveros de los almacenes (al igual que como sucede con las llaves, aunque no lo mencionásemos en su debido momento) no implica el borrado de éstos del sistema salvo que no resten más copias en ninguno del resto de almacenes, pero éste es un hecho que no debe preocuparnos ya que es tarea de los disparadores creados en la primera iteración de la fase de Construcción. Para llevar a buen puerto este caso de uso solo necesitaremos dos cosas: la tabla de donde seleccionarlos (ya diseñada) y un botón que inicie el borrado (por añadir a la botonera).

Una vez el usuario seleccione todos los llaveros que desea eliminar del almacén y pulse el botón de borrado, el sistema deberá:

1. Obtener el almacén y los llaveros seleccionados dentro de éste.
2. Recorrerlos borrándolos uno a uno del almacén.
3. Refrescar las tablas de almacenes.

Si refinamos y detallamos un poco más este proceso obtendremos los siguientes diagramas de actividades y de secuencia:

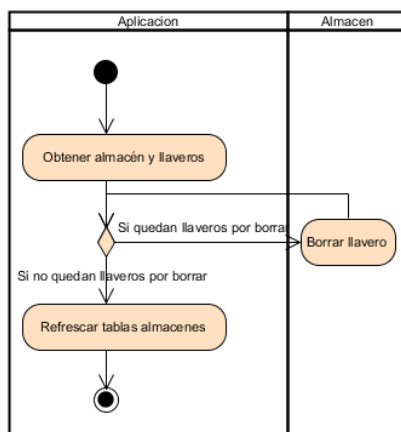


Diagrama de actividades

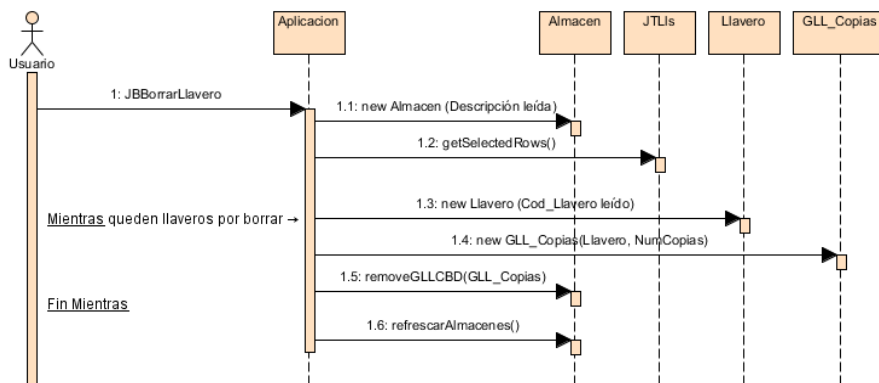


Diagrama de secuencia



4.29. Caso de uso 96: Modificar copias llavero

La actualización del número de copias de un llavero en un almacén sigue la misma lógica que la de las llaves, con que con añadir un botón para tal fin debería bastarnos en un principio para desencadenar este flujo de acciones:

1. Obtener el almacén, el llavero y el nuevo número de copias.
2. Obtener dos instancias GLL_Copias del llavero.
3. Asignar al GLL_Copias que represente la copia válida el número de copias indicado por el usuario.
4. Actualizar el GLL_Copias viejo con el nuevo.
5. Refrescar la tabla de llaveros para que los cambios sean visibles.

Si nos fijamos, la estrategia empleada es casi idéntica a la de las llaves, lo único que cambia es que instanciamos el objeto GLL_Copias con un objeto.

Disponemos de todo lo necesario, con que diseñemos los diagramas necesarios:

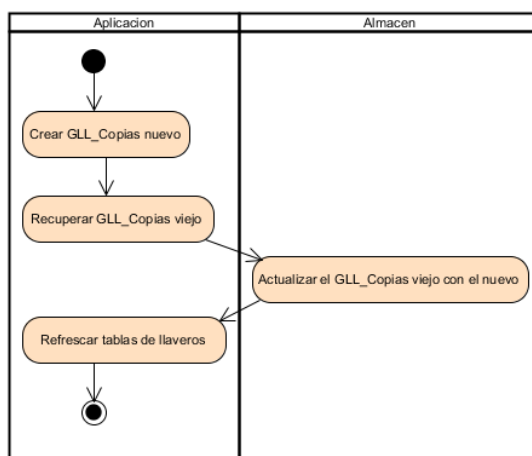


Diagrama de actividades

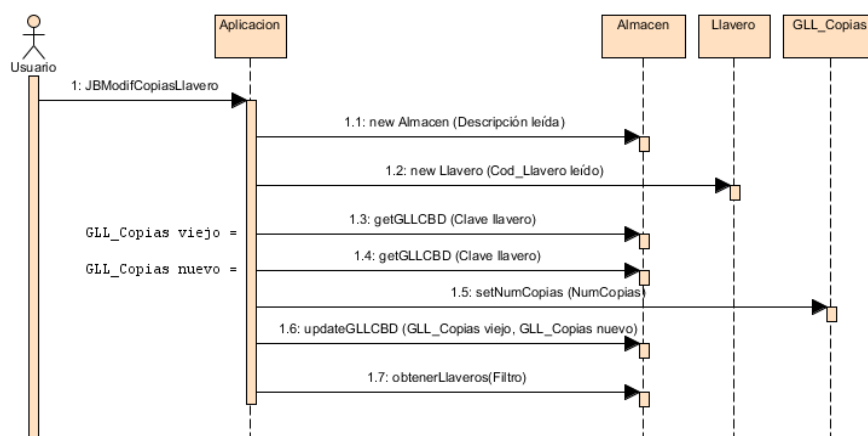


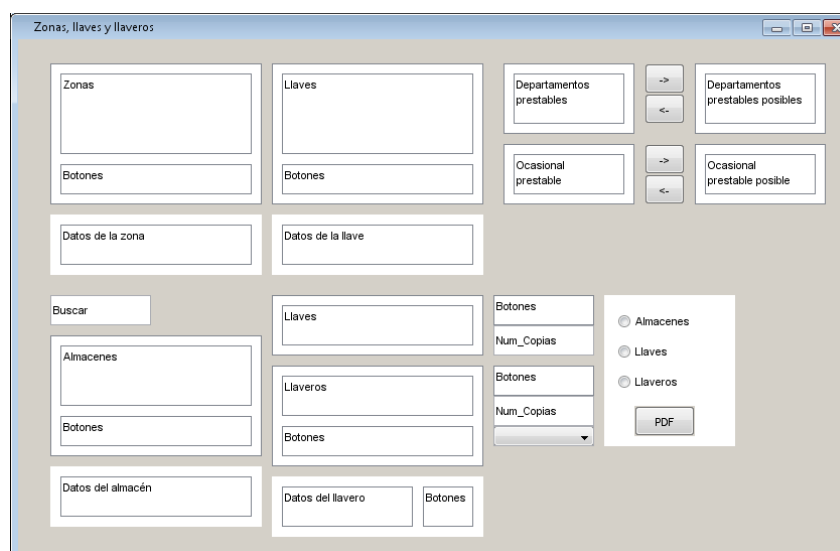
Diagrama de secuencia



4.30. Casos de uso 76, 85 y 97: Listar llaves, llaveros y almacenes

Nos enfrentamos a unos casos de uso distintos al resto y que hemos dejado para el final de la iteración para tratarlos así todos juntos. El cliente nos exigió explícitamente poder generar reportes de las llaves, llaveros y almacenes existentes y para evitar colocar tres botones dispersados por la pestaña optamos por crear un pequeño panel dentro de ésta donde poder definir y generar reportes. Nos basaremos en las tablas creadas para los almacenes, consiguiendo así implementar las necesidades impuestas por el cliente (informes de llaves, llaveros y almacenes) y además ofrecer al usuario la posibilidad de obtener reportes de los elementos existentes en cada uno de los almacenes.

Entre los componentes existentes en Swing, puede ser muy buena idea optar por un Button Group y colocarlo en la parte inferior derecha; libre y fácil de localizar:



Interfaz gráfica propuesta

Tal y como hicimos para generar los anteriores informes, nos ayudaremos de un JFileChooser y la clase *GeneradorInformes* para codificar la siguiente estrategia:

1. Desplegar un JFileChooser para que el usuario elija en qué ruta desea que el sistema deposite el informe.
2. Crear un objeto de tipo *GeneradorInformes*.
3. Llamar al método *generarInformeAvanzado* pasándole los títulos y subtítulos que deberá contener el reporte, así como las tablas (las de la sección de almacenes) y el estado de los botones que definirán el informe a generar.



generarInformeAvanzado fue implementado en la anterior iteración y *JFileChooser* es un componente de la API estándar, con que en un principio no observamos ningún elemento que debamos crear en la nuestra antes para poder codificar estos diagramas:

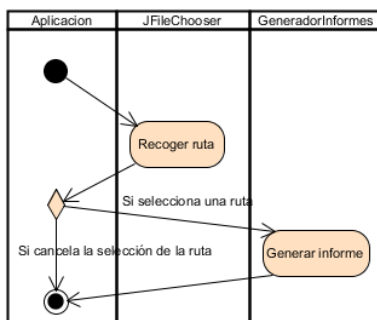


Diagrama de actividades

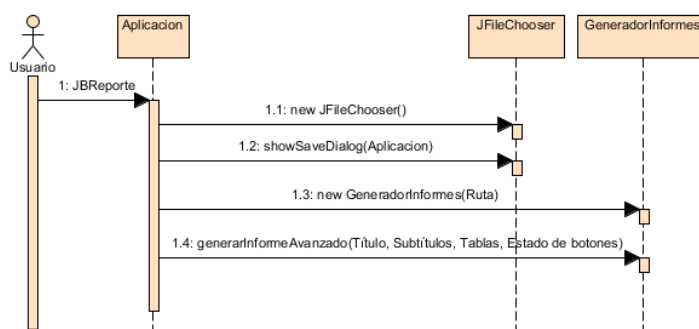


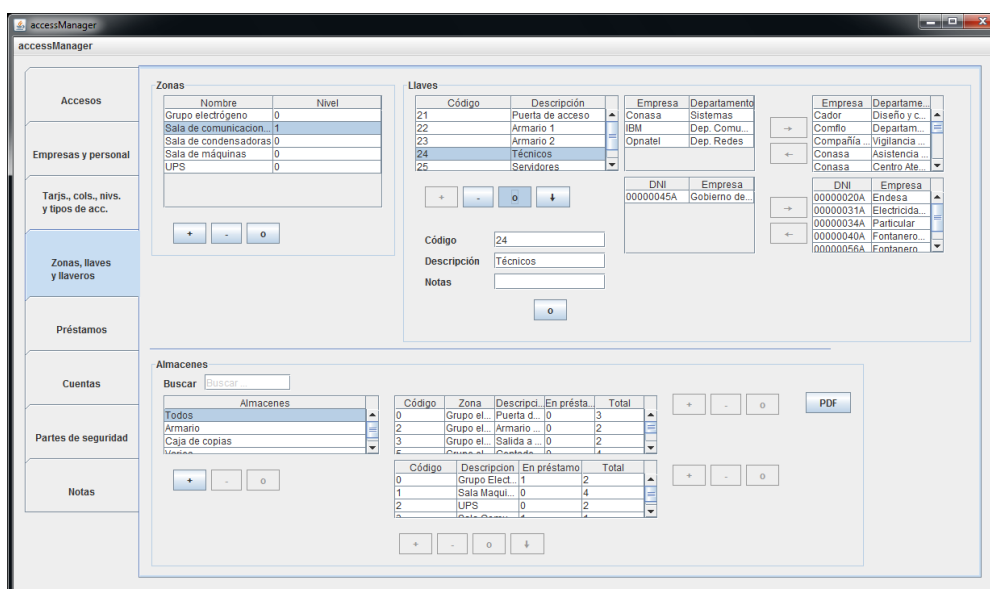
Diagrama de secuencia



5. Codificación e integración

La pestaña que integra todos los datos relativos a zonas, llaves y llaveros resulta de complejidad media con respecto a todas las demás, no por la dificultad de los procedimientos que exige o las consultas que realiza, sino más bien por los tipos de datos que maneja, ya que todos son datos compuestos que exigen muchas relaciones entre ellos y cuya muestra por pantalla de manera simple y clara es un hecho que hay que meditar mucho y analizar con cuidado, además de la dificultad que posee realizar una correcta activación y desactivación del buen número de botones y campos existentes.

Tras dos días dedicados a la tarea de codificación hemos obtenido la siguiente interfaz:



Interfaz de la pestaña 'Zonas, llaves y llaveros'

La integración de la pestaña con el resto es instantánea, ya que empleamos la reservada para la ocasión desde el comienzo del desarrollo. Ha sido desarrollada con los mismos componentes que las restantes (API y base de datos) y las pruebas que a continuación detallamos pondrán de manifiesto que todos ellos siguen interactuando a la perfección.



6. Pruebas

6.1. De unidad

Una vez que hemos completado las etapas de análisis, diseño y codificación llegamos a la última y decisiva, la de pruebas, que determinará si el trabajo realizado en estos cuatro días ha sido correcto o debe ser corregido. A continuación, todos los casos de prueba realizados:

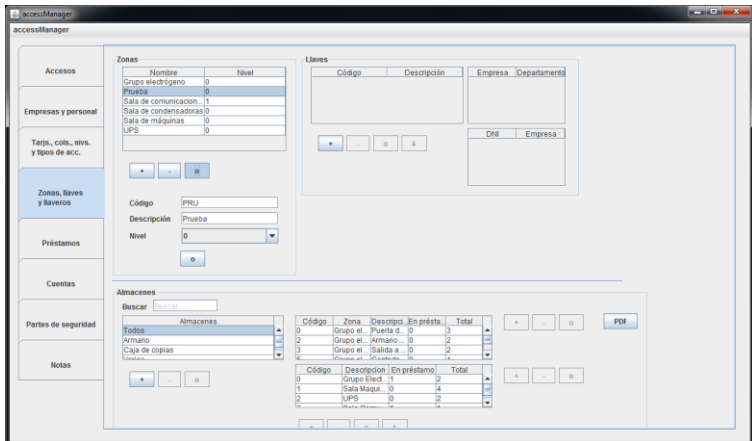
Caso de prueba	6.01 - Casos de uso 65, 70a, 77 y 86: Visualizaciones																																																																																																																																									
Objetivo	Comprobar que las tablas que deben estar cargadas inicialmente lo hacen con los registros que deben.																																																																																																																																									
Actividades	Arrancar la aplicación, situarnos en la pestaña 'Zonas, llaves y llaveros' y comprobar que los datos que en ella se muestran son los existentes en la base de datos.																																																																																																																																									
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. • Utilidad phpMyAdmin. 																																																																																																																																									
Datos de entrada	La población de prueba residente en la base de datos.																																																																																																																																									
Salida esperada	<p>Las tablas de zonas, almacenes, llaves y llaveros deberían mostrar respectivamente la siguiente información:</p> <table border="1"> <thead> <tr> <th>Nombre</th> <th>Nivel</th> </tr> </thead> <tbody> <tr> <td>Sala de condensadoras</td> <td>0</td> </tr> <tr> <td>Grupo electrógeno</td> <td>0</td> </tr> <tr> <td>Sala de comunicaciones</td> <td>1</td> </tr> <tr> <td>Sala de máquinas</td> <td>0</td> </tr> <tr> <td>UPS</td> <td>0</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Almacenes</th> </tr> </thead> <tbody> <tr> <td>Todos</td> </tr> <tr> <td>Armario</td> </tr> <tr> <td>Caja de copias</td> </tr> <tr> <td>Varios</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Código</th> <th>Zona</th> <th>Descripción</th> <th>En préstamo</th> <th>Total</th> </tr> </thead> <tbody> <tr><td>0</td><td>Grupo electrógeno</td><td>Puerta de acceso</td><td>0</td><td>3</td></tr> <tr><td>2</td><td>Grupo electrógeno</td><td>Armario principal</td><td>0</td><td>2</td></tr> <tr><td>3</td><td>Grupo electrógeno</td><td>Salida a patio</td><td>0</td><td>2</td></tr> <tr><td>5</td><td>Grupo electrógeno</td><td>Contadores</td><td>0</td><td>4</td></tr> <tr><td>6</td><td>Sala de máquinas</td><td>Puerta de acceso</td><td>0</td><td>4</td></tr> <tr><td>7</td><td>Sala de máquinas</td><td>Puerta 2</td><td>0</td><td>2</td></tr> <tr><td>8</td><td>Sala de máquinas</td><td>Armario 1</td><td>0</td><td>1</td></tr> <tr><td>9</td><td>Sala de máquinas</td><td>Armario 2</td><td>0</td><td>1</td></tr> <tr><td>10</td><td>Sala de máquinas</td><td>Cuadros de fuerzas</td><td>0</td><td>3</td></tr> <tr><td>11</td><td>UPS</td><td>Puertade acceso</td><td>0</td><td>2</td></tr> <tr><td>12</td><td>UPS</td><td>Baterias</td><td>0</td><td>2</td></tr> <tr><td>13</td><td>UPS</td><td>Armarios</td><td>0</td><td>2</td></tr> <tr><td>14</td><td>UPS</td><td>Cuadros</td><td>0</td><td>2</td></tr> <tr><td>15</td><td>Sala de condensadoras</td><td>Puerta de acceso</td><td>0</td><td>1</td></tr> <tr><td>16</td><td>Sala de condensadoras</td><td>Armario 1</td><td>0</td><td>2</td></tr> <tr><td>17</td><td>Sala de condensadoras</td><td>Armario 2</td><td>1</td><td>5</td></tr> <tr><td>18</td><td>Sala de condensadoras</td><td>Mantenimiento</td><td>1</td><td>4</td></tr> <tr><td>19</td><td>Sala de condensadoras</td><td>Cuadros</td><td>1</td><td>4</td></tr> <tr><td>20</td><td>Sala de condensadoras</td><td>Cajones</td><td>1</td><td>4</td></tr> <tr><td>21</td><td>Sala de comunicaciones</td><td>Puerta de acceso</td><td>0</td><td>4</td></tr> <tr><td>22</td><td>Sala de comunicaciones</td><td>Armario 1</td><td>0</td><td>4</td></tr> <tr><td>23</td><td>Sala de comunicaciones</td><td>Armario 2</td><td>0</td><td>4</td></tr> <tr><td>24</td><td>Sala de comunicaciones</td><td>Técnicos</td><td>0</td><td>4</td></tr> </tbody> </table>	Nombre	Nivel	Sala de condensadoras	0	Grupo electrógeno	0	Sala de comunicaciones	1	Sala de máquinas	0	UPS	0	Almacenes	Todos	Armario	Caja de copias	Varios	Código	Zona	Descripción	En préstamo	Total	0	Grupo electrógeno	Puerta de acceso	0	3	2	Grupo electrógeno	Armario principal	0	2	3	Grupo electrógeno	Salida a patio	0	2	5	Grupo electrógeno	Contadores	0	4	6	Sala de máquinas	Puerta de acceso	0	4	7	Sala de máquinas	Puerta 2	0	2	8	Sala de máquinas	Armario 1	0	1	9	Sala de máquinas	Armario 2	0	1	10	Sala de máquinas	Cuadros de fuerzas	0	3	11	UPS	Puertade acceso	0	2	12	UPS	Baterias	0	2	13	UPS	Armarios	0	2	14	UPS	Cuadros	0	2	15	Sala de condensadoras	Puerta de acceso	0	1	16	Sala de condensadoras	Armario 1	0	2	17	Sala de condensadoras	Armario 2	1	5	18	Sala de condensadoras	Mantenimiento	1	4	19	Sala de condensadoras	Cuadros	1	4	20	Sala de condensadoras	Cajones	1	4	21	Sala de comunicaciones	Puerta de acceso	0	4	22	Sala de comunicaciones	Armario 1	0	4	23	Sala de comunicaciones	Armario 2	0	4	24	Sala de comunicaciones	Técnicos	0	4
Nombre	Nivel																																																																																																																																									
Sala de condensadoras	0																																																																																																																																									
Grupo electrógeno	0																																																																																																																																									
Sala de comunicaciones	1																																																																																																																																									
Sala de máquinas	0																																																																																																																																									
UPS	0																																																																																																																																									
Almacenes																																																																																																																																										
Todos																																																																																																																																										
Armario																																																																																																																																										
Caja de copias																																																																																																																																										
Varios																																																																																																																																										
Código	Zona	Descripción	En préstamo	Total																																																																																																																																						
0	Grupo electrógeno	Puerta de acceso	0	3																																																																																																																																						
2	Grupo electrógeno	Armario principal	0	2																																																																																																																																						
3	Grupo electrógeno	Salida a patio	0	2																																																																																																																																						
5	Grupo electrógeno	Contadores	0	4																																																																																																																																						
6	Sala de máquinas	Puerta de acceso	0	4																																																																																																																																						
7	Sala de máquinas	Puerta 2	0	2																																																																																																																																						
8	Sala de máquinas	Armario 1	0	1																																																																																																																																						
9	Sala de máquinas	Armario 2	0	1																																																																																																																																						
10	Sala de máquinas	Cuadros de fuerzas	0	3																																																																																																																																						
11	UPS	Puertade acceso	0	2																																																																																																																																						
12	UPS	Baterias	0	2																																																																																																																																						
13	UPS	Armarios	0	2																																																																																																																																						
14	UPS	Cuadros	0	2																																																																																																																																						
15	Sala de condensadoras	Puerta de acceso	0	1																																																																																																																																						
16	Sala de condensadoras	Armario 1	0	2																																																																																																																																						
17	Sala de condensadoras	Armario 2	1	5																																																																																																																																						
18	Sala de condensadoras	Mantenimiento	1	4																																																																																																																																						
19	Sala de condensadoras	Cuadros	1	4																																																																																																																																						
20	Sala de condensadoras	Cajones	1	4																																																																																																																																						
21	Sala de comunicaciones	Puerta de acceso	0	4																																																																																																																																						
22	Sala de comunicaciones	Armario 1	0	4																																																																																																																																						
23	Sala de comunicaciones	Armario 2	0	4																																																																																																																																						
24	Sala de comunicaciones	Técnicos	0	4																																																																																																																																						

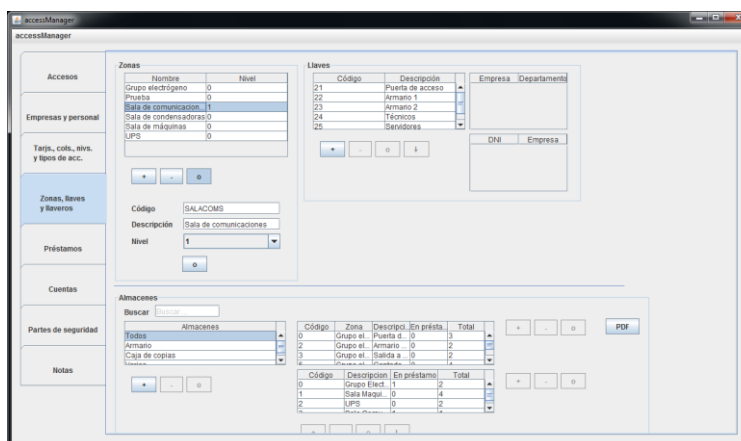


	<table border="1"> <thead> <tr> <th>Código</th> <th>Descripcion</th> <th>En préstamo</th> <th>Total</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Grupo Electrogeno</td> <td>1</td> <td>2</td> </tr> <tr> <td>1</td> <td>Sala Maquinas</td> <td>0</td> <td>4</td> </tr> <tr> <td>2</td> <td>UPS</td> <td>0</td> <td>2</td> </tr> <tr> <td>3</td> <td>Sala Comunicaciones</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	Código	Descripcion	En préstamo	Total	0	Grupo Electrogeno	1	2	1	Sala Maquinas	0	4	2	UPS	0	2	3	Sala Comunicaciones	1	1
Código	Descripcion	En préstamo	Total																		
0	Grupo Electrogeno	1	2																		
1	Sala Maquinas	0	4																		
2	UPS	0	2																		
3	Sala Comunicaciones	1	1																		
Salida observada	<p>La esperada:</p> <p>Nota: El buscador ideado para la sección de almacenes es probado y podemos asegurar que funciona correctamente.</p>																				
Evaluación/ conclusión	<p>Los casos de uso '65, 70a, 77 y 86 - Visualizaciones' han sido realizados satisfactoriamente.</p>																				

Caso de prueba para los casos de uso '65, 70a, 77 y 86 - Visualizaciones'



Caso de prueba	6.02 - Casos de uso 66 y 70b: Visualizar zona del centro y sus llaves														
Objetivo	Verificar que al seleccionar una zona podemos visualizar por pantalla todos los datos y llaves que le pertenecen.														
Actividades	Desplegar accessManager, situarnos en la tab de zonas y clicar en una que disponga de llaves y otra que no, observando que todo se visualiza correctamente.														
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. • Utilidad phpMyAdmin. 														
Datos de entrada	<p>Insertamos una nueva zona sin llaves...</p> <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: @ INSERT INTO Zona VALUES ('PRU',CURDATE(),'Prueba',0,'2010-01-01 00:00:01');</pre> <p>... para así pasar a visualizar ésta y la de 'Sala de comunicación'.</p>														
Salida esperada	<p>En el primer caso se espera observar 'PRU, Prueba, 0' y ninguna llave, mientras que en el segundo 'SALACOMS, Sala de comunicaciones, 1' y las llaves:</p> <table border="1"> <thead> <tr> <th>Código</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>21</td> <td>Puerta de acceso</td> </tr> <tr> <td>22</td> <td>Armario 1</td> </tr> <tr> <td>23</td> <td>Armario 2</td> </tr> <tr> <td>24</td> <td>Técnicos</td> </tr> <tr> <td>25</td> <td>Servidores</td> </tr> <tr> <td>26</td> <td>Mantenimiento</td> </tr> </tbody> </table>	Código	Descripción	21	Puerta de acceso	22	Armario 1	23	Armario 2	24	Técnicos	25	Servidores	26	Mantenimiento
Código	Descripción														
21	Puerta de acceso														
22	Armario 1														
23	Armario 2														
24	Técnicos														
25	Servidores														
26	Mantenimiento														
Salida observada	<p>La esperada:</p> 														

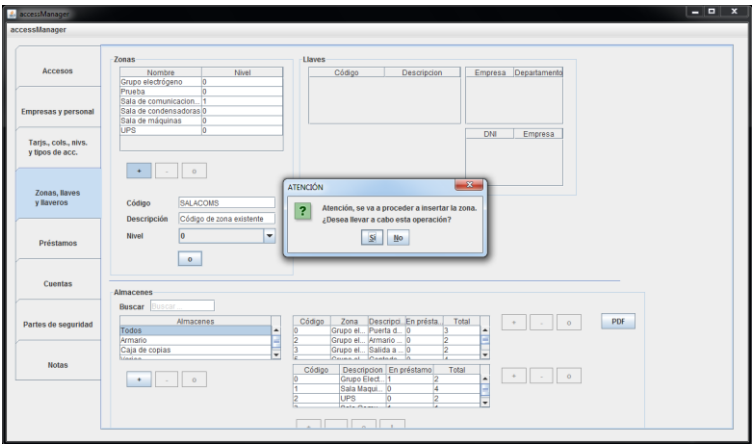
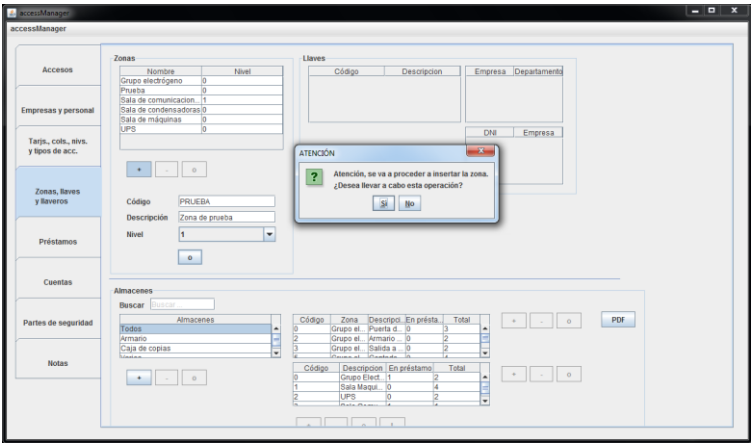


**Evaluación/
conclusión**

Los casos de uso '66 y 70b - Visualizar zona del centro y sus llaves' han sido realizados satisfactoriamente.

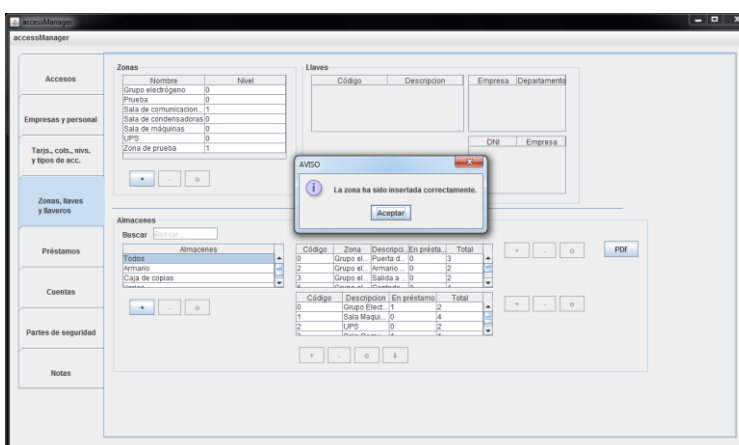
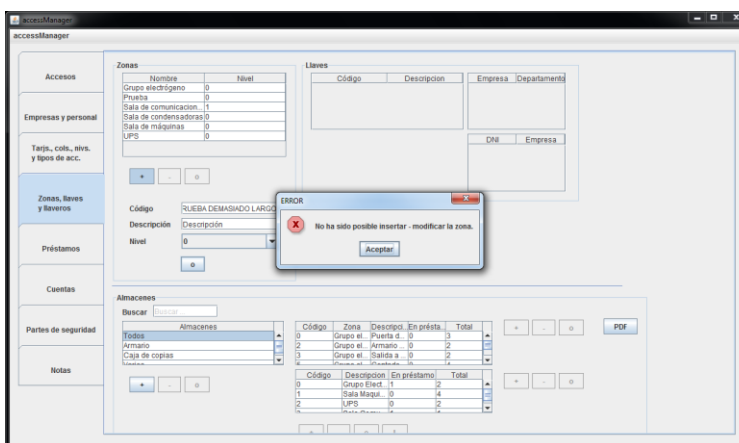
Caso de prueba para los casos de uso '66 y 70b - Visualizar zona del centro y sus llaves'



Caso de prueba	6.03 - Caso de uso 67: Añadir zona del centro
Objetivo	Probar que el usuario puede añadir nuevas zonas al sistema siempre y cuando introduzca datos con un formato válido.
Actividades	Abrir la aplicación, acceder a la pestaña de zonas y probar a insertar una errónea y otra válida, verificando que el sistema responde como deseamos.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos resultante del caso de prueba anterior.
Datos de entrada	<p>Probaremos a insertar estas dos zonas:</p> <ul style="list-style-type: none">• Errónea: 'CODIGO DE PRUEBA DEMASIADO LARGO, Descripción, 0'. (Formato erróneo)  <ul style="list-style-type: none">• Válida: 'PRUEBA, Zona de prueba, 1' 
Salida esperada	El primer intento deberá desembocar en error, sin realizar ninguna inserción en la base de datos, mientras que el segundo no ofrecerá problema alguno.

La esperada. En orden obtenemos:

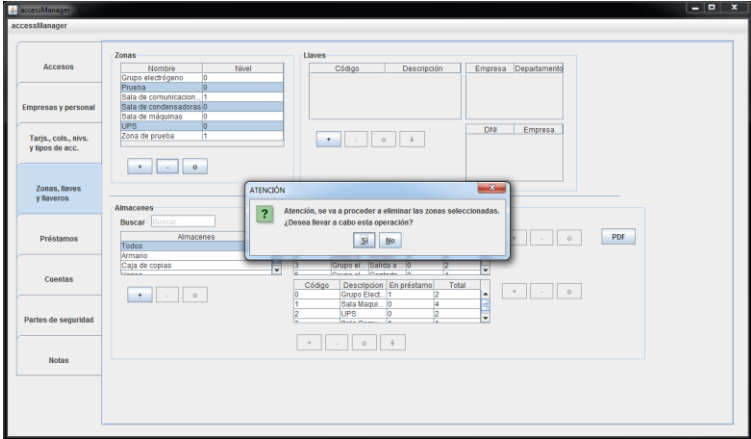
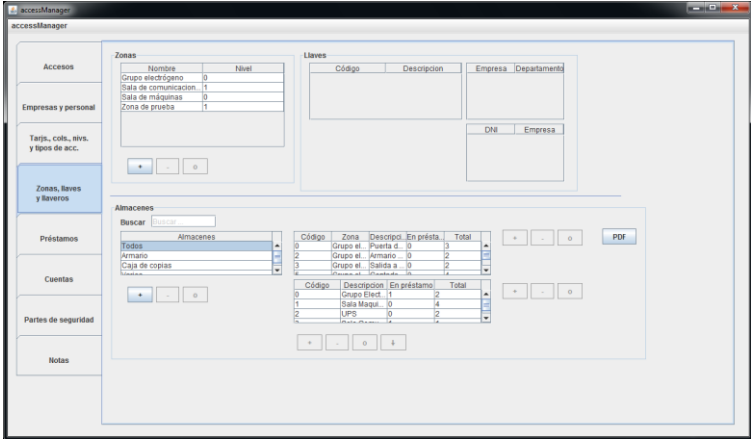
Salida observada



Evaluación/
conclusión

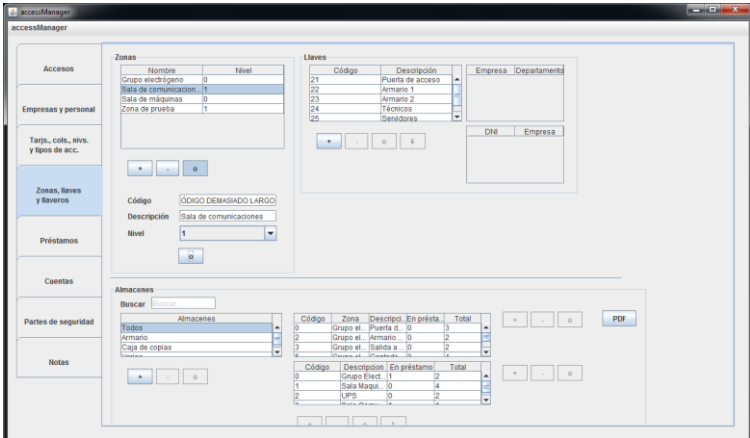
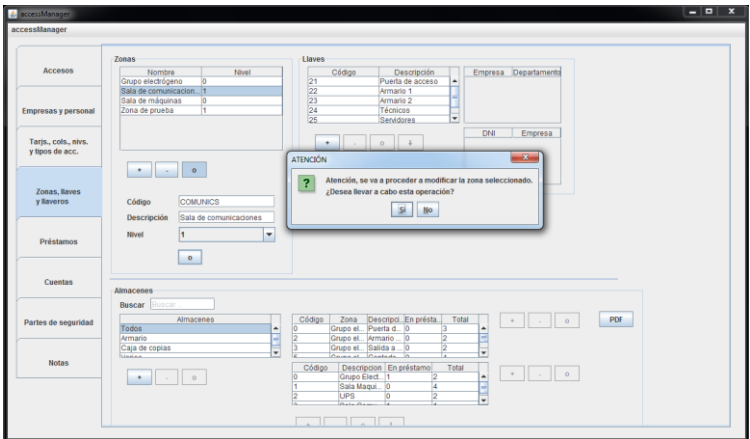
El caso de uso '67 - Añadir zona del centro' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '67 - Añadir zona del centro'

Caso de prueba	6.04 - Caso de uso 68: Borrar zona del centro
Objetivo	Chequear que el borrado de zonas se realiza sin provocar ningún error indeseado.
Actividades	Loguear en el sistema, eliminar varias zonas y comprobar que el borrado se realiza correctamente.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior.
Datos de entrada	<p>Eliminaremos las siguientes zonas:</p> 
Salida esperada	<p>Sólo deberán quedar en la tabla de zonas las zonas con descripción:</p> <ul style="list-style-type: none"> • 'Grupo eléctrico' • 'Sala de comunicaciones' • 'Sala de máquinas' • 'Zona de prueba'
Salida observada	<p>La esperada:</p> 
Evaluación/conclusión	El caso de uso '68 - Borrar zona del centro' ha sido realizado satisfactoriamente.

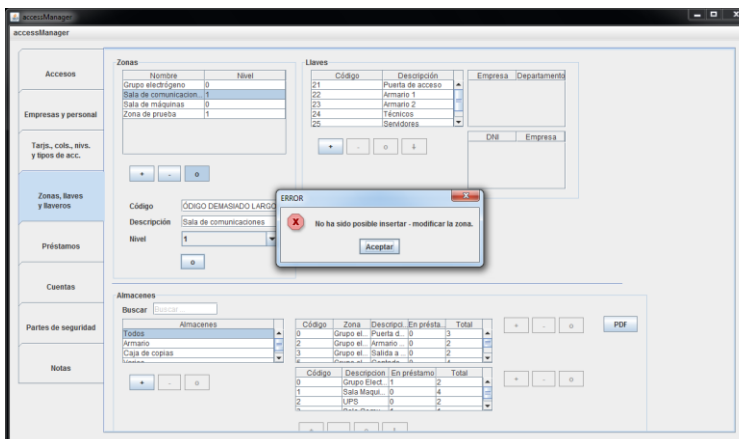
Caso de prueba para el caso de uso '68 - Borrar zona del centro'



Caso de prueba	6.05 - Caso de uso 69: Modificar zona del centro
Objetivo	Comprobar que el sistema permite modificar cualquiera de las zonas del centro.
Actividades	Acceder al sistema, dirigirnos a la pestaña 'Zonas, llaves y llaveros' y modificar una zona con datos erróneos y válidos, comprobando que la salida observada es la esperada.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos resultante del caso de prueba anterior.
Datos de entrada	<p>Modificamos 'Sala de comunicaciones' cambiándole su código por uno con formato no válido...</p>  <p>... y otro correcto:</p> 
Salida esperada	El sistema no permitirá que la primera actualización se lleve a cabo, pero si la segunda.

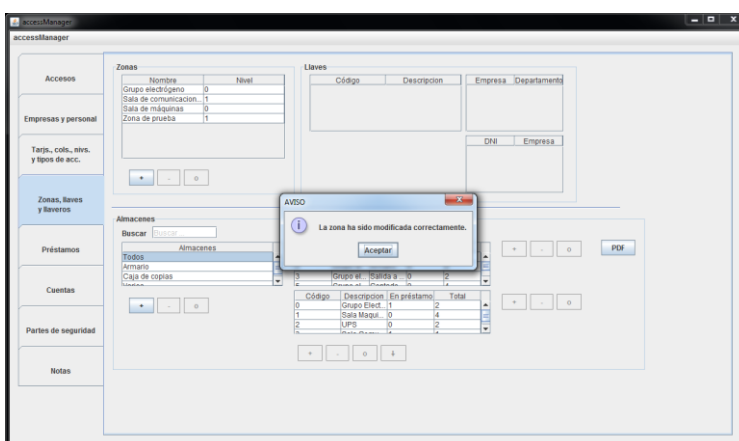
La esperada:

- Para el primer caso:



Salida observada

- En el segundo:

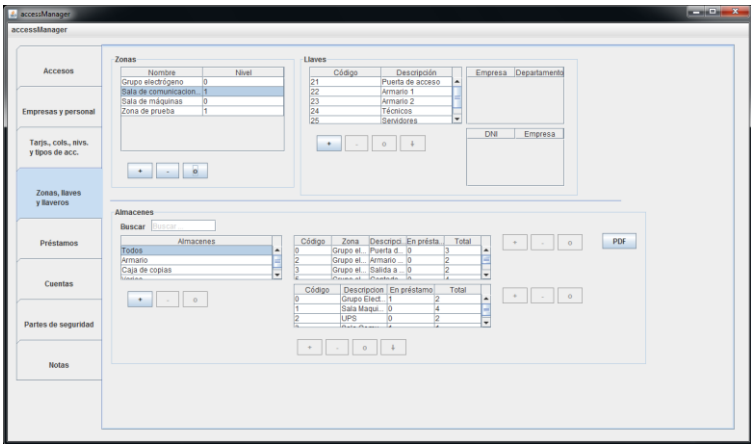


Evaluación/
conclusión

El caso de uso '69 - Modificar zona del centro' ha sido realizado satisfactoriamente.

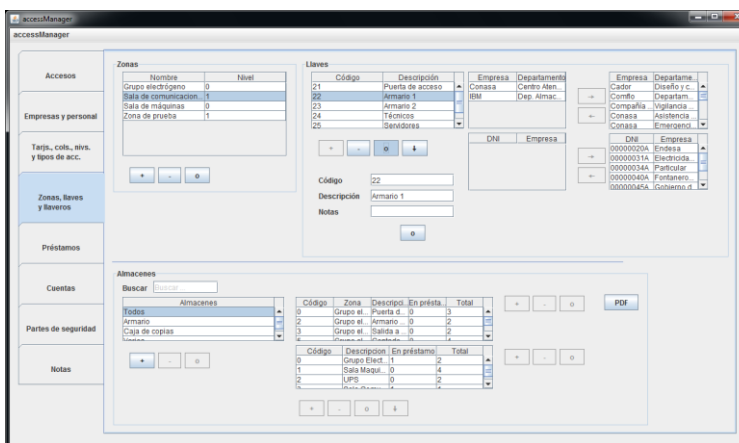
Caso de prueba para el caso de uso '69 - Modificar zona del centro'



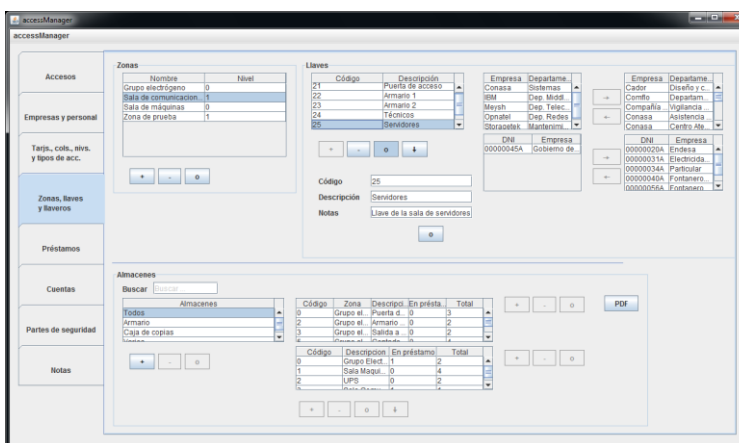
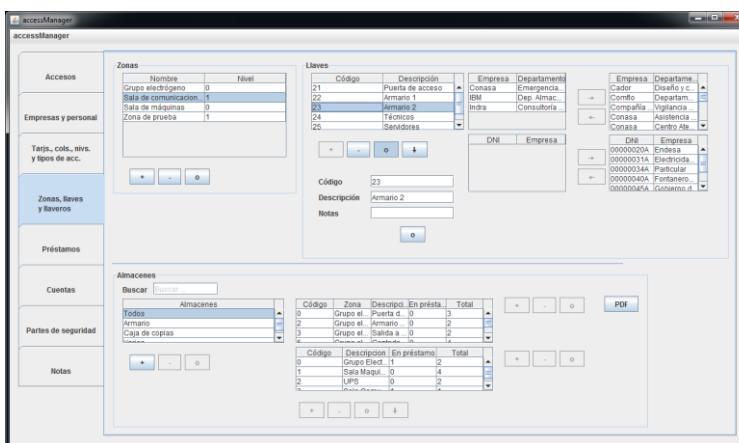
Caso de prueba	6.06 - Caso de uso 71: Visualizar llave																																																										
Objetivo	Verificar que el sistema muestra todos los datos de cualquier llave si el usuario se lo solicita.																																																										
Actividades	Loguear en accessManager, acceder a la pestaña de zonas donde seleccionaremos una así como varias de las llaves que posee, comprobando que siempre es posible la visualización de los datos que las componen.																																																										
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior. • Utilidad phpMyAdmin. 																																																										
Datos de entrada	<p>Jugaremos con ‘Sala de comunicaciones’, ya que dispone de muchas llaves, y más concretamente con las llaves con código 22, 23 y 25:</p> 																																																										
Salida esperada	<p>Las llaves seleccionadas deberán tener estos datos...</p> <table border="1" data-bbox="563 1256 898 1346"> <thead> <tr> <th>Código</th> <th>Descripción</th> <th>Notas</th> <th>Otros datos a tener en cuenta</th> </tr> </thead> <tbody> <tr> <td>22</td> <td>Armario 1</td> <td></td> <td></td> </tr> <tr> <td>23</td> <td>Armario 2</td> <td></td> <td></td> </tr> <tr> <td>25</td> <td>Servidores</td> <td></td> <td>Llave de la sala de servidores</td> </tr> </tbody> </table> <p>... y los siguientes posibles prestatarios respectivamente:</p> <table border="1" data-bbox="563 1453 778 1520"> <thead> <tr> <th>Cod_Llave</th> <th>Campo1</th> <th>Campo2</th> </tr> </thead> <tbody> <tr> <td>22</td> <td>CONA</td> <td>CAU</td> </tr> <tr> <td>22</td> <td>IBM</td> <td>ALDA</td> </tr> </tbody> </table> <table border="1" data-bbox="563 1550 778 1639"> <thead> <tr> <th>Cod_Llave</th> <th>Campo1</th> <th>Campo2</th> </tr> </thead> <tbody> <tr> <td>23</td> <td>CONA</td> <td>EMER</td> </tr> <tr> <td>23</td> <td>IBM</td> <td>ALDA</td> </tr> <tr> <td>23</td> <td>INDR</td> <td>CTYC</td> </tr> </tbody> </table> <table border="1" data-bbox="563 1668 858 1816"> <thead> <tr> <th>Cod_Llave</th> <th>Campo1</th> <th>Campo2</th> </tr> </thead> <tbody> <tr> <td>25</td> <td>CONA</td> <td>SIST</td> </tr> <tr> <td>25</td> <td>IBM</td> <td>MIDD</td> </tr> <tr> <td>25</td> <td>MEYS</td> <td>TELE</td> </tr> <tr> <td>25</td> <td>OPNA</td> <td>DRED</td> </tr> <tr> <td>25</td> <td>STOR</td> <td>MASE</td> </tr> <tr> <td>25</td> <td>0000045A</td> <td>Gobierno de Navarra</td> </tr> </tbody> </table>	Código	Descripción	Notas	Otros datos a tener en cuenta	22	Armario 1			23	Armario 2			25	Servidores		Llave de la sala de servidores	Cod_Llave	Campo1	Campo2	22	CONA	CAU	22	IBM	ALDA	Cod_Llave	Campo1	Campo2	23	CONA	EMER	23	IBM	ALDA	23	INDR	CTYC	Cod_Llave	Campo1	Campo2	25	CONA	SIST	25	IBM	MIDD	25	MEYS	TELE	25	OPNA	DRED	25	STOR	MASE	25	0000045A	Gobierno de Navarra
Código	Descripción	Notas	Otros datos a tener en cuenta																																																								
22	Armario 1																																																										
23	Armario 2																																																										
25	Servidores		Llave de la sala de servidores																																																								
Cod_Llave	Campo1	Campo2																																																									
22	CONA	CAU																																																									
22	IBM	ALDA																																																									
Cod_Llave	Campo1	Campo2																																																									
23	CONA	EMER																																																									
23	IBM	ALDA																																																									
23	INDR	CTYC																																																									
Cod_Llave	Campo1	Campo2																																																									
25	CONA	SIST																																																									
25	IBM	MIDD																																																									
25	MEYS	TELE																																																									
25	OPNA	DRED																																																									
25	STOR	MASE																																																									
25	0000045A	Gobierno de Navarra																																																									



La espera, como puede observarse:



Salida observada

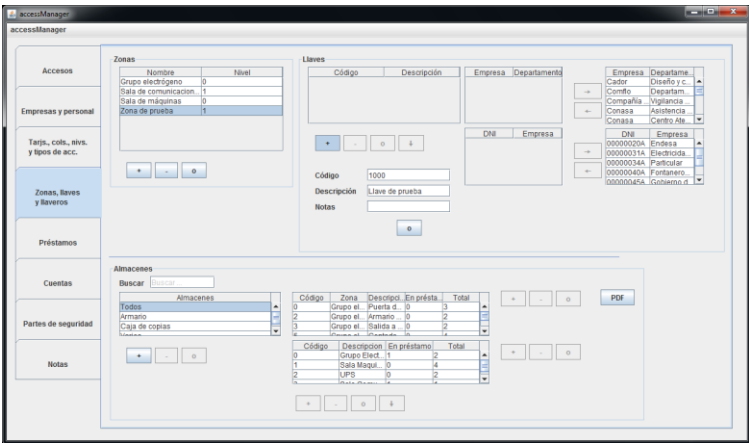
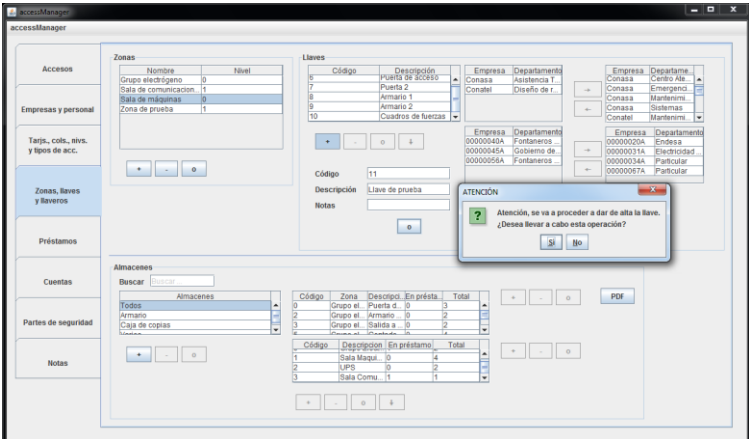


Evaluación/
conclusión

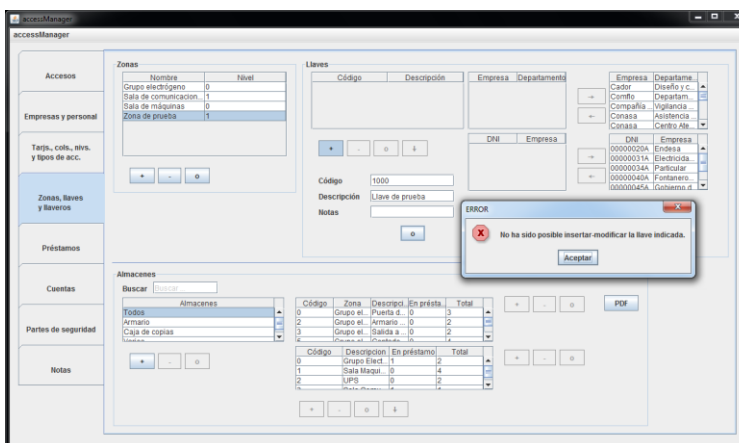
El caso de uso '71 - Visualizar llave' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '71 - Visualizar llave'

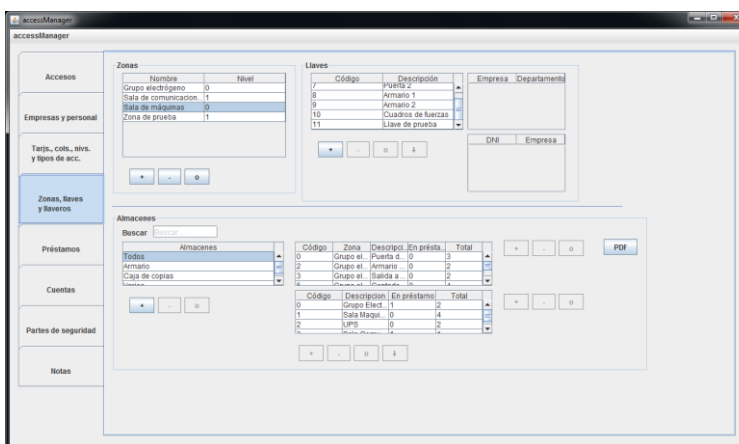


Caso de prueba	6.07 - Caso de uso 72: Añadir llave
Objetivo	Chequear que es posible añadir nuevas llaves a las zonas que deseemos siempre y cuando lo hagamos según las normas establecidas.
Actividades	Arrancar la aplicación, logear y situarnos en la tab de zonas. Seleccionaremos un par de ellas, en las que intentaremos introducir una llave errónea y otra válida. Nuestro objetivo será comprobar que la primera inserción es inviable, pero no así la segunda.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior.
Datos de entrada	<p>La prueba será realizada sobre las zonas 'Zona de prueba' y 'Sala de máquinas', insertando las llaves '1000, Llave de prueba,' y '11, Llave de prueba,' (junto con varios tipos de prestatarios posibles seleccionados al azar) respectivamente:</p>  
Salida esperada	El primer intento hará que el sistema muestre una pantalla de error y no ejecute ninguna inserción en la base de datos, mientras que el segundo insertará la llave en la zona 'Sala de máquinas'.

La esperada:



Salida observada

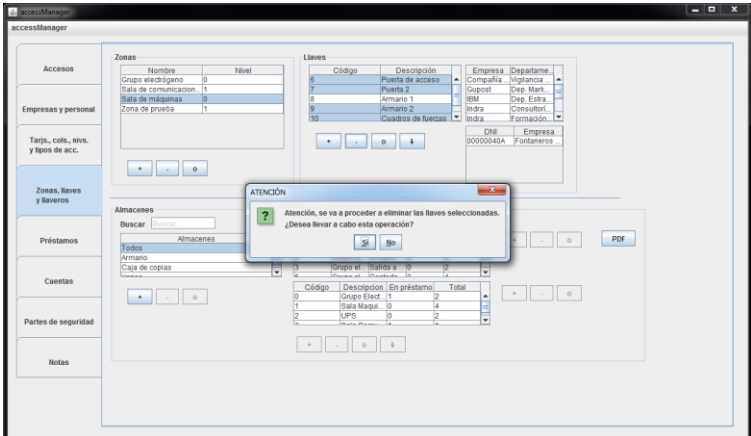
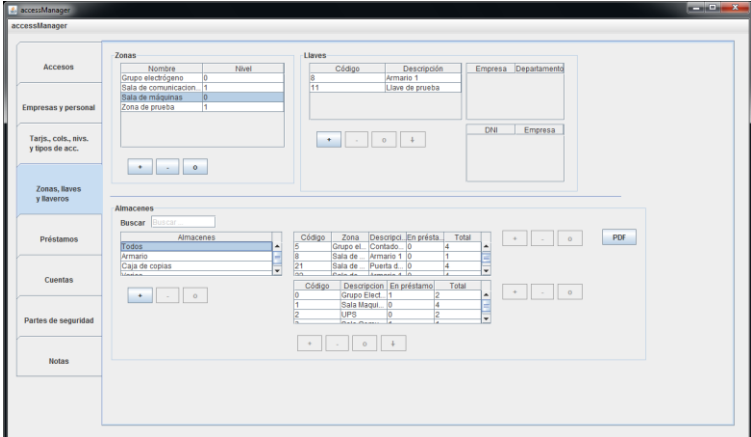


Nota: El generador de código por defecto funciona correctamente.

Evaluación/
conclusión

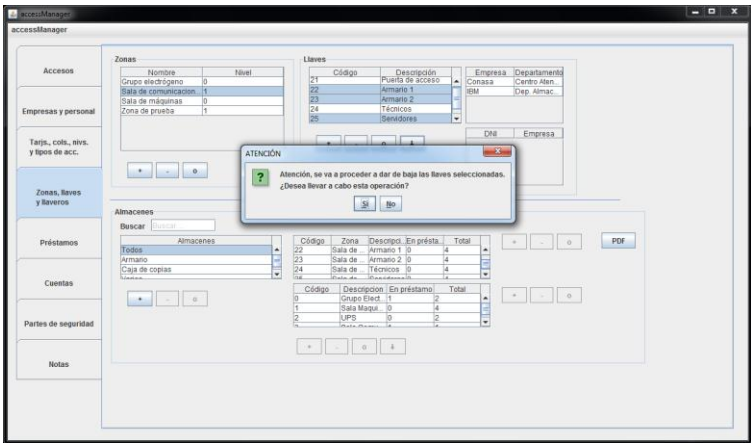
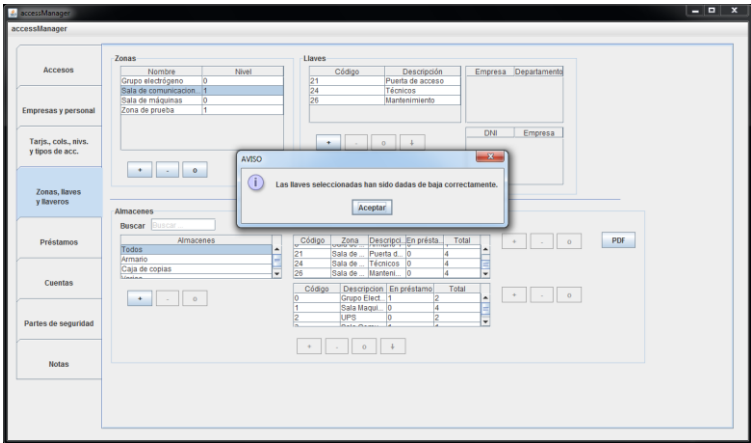
El caso de uso '72 - Añadir llave' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '72 - Añadir llave'

Caso de prueba	6.08 - Caso de uso 73: Borrar llave
Objetivo	Verificar que podemos borrar cualquier llave existente en cualquier zona del centro.
Actividades	Desplegar accessManager, situarnos en la pestaña de zonas, seleccionar una y borrar varias de sus llaves, comprobando que éstas desaparecen y solo permanecen las restantes.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior.
Datos de entrada	<p>Seleccionamos la zona 'Sala de máquinas' y probamos a eliminar las siguiente llaves:</p> 
Salida esperada	Las llaves 6, 7, 9 y 10 deberán desaparecer de la tabla de llaves, permaneciendo sólo las de código 8 y 11, además de no constar en ningún almacén.
Salida observada	<p>La esperada:</p> 
Evaluación/conclusión	El caso de uso '73 - Borrar llave' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '73 - Borrar llave'



Caso de prueba	6.9 - Caso de uso 74: Dar de baja llave
Objetivo	Comprobar que la operación de baja de llaves se realizada conforme a lo diseñado. Tras realizar la baja de llaves éstas deberán permanecer en la base de datos pero dejar de formar parte de la aplicación (manejable mediante la interfaz gráfica).
Actividades	Loguear en el sistema, situarnos en la sección de llaves y dar de baja algunas de ellas, comprobando que dejan de formar parte de la interfaz gráfica pero siguen permaneciendo en el sistema.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos resultante del caso de prueba anterior.• Utilidad phpMyAdmin.
Datos de entrada	<p>Podríamos realizar esta prueba con cualquier zona que disponga de llaves, pero en este caso optaremos por volver a ‘Sala de comunicaciones’ y dar de baja las siguientes llaves:</p> 
Salida esperada	Las llaves 22, 23 y 25 desaparecerán de ambas tablas de llaves, permaneciendo en el sistema pero disponiendo ya de fecha de baja.
Salida observada	La esperada: 



Cod_Llave	F_Alta_L	F_Baja_L	Cod_Zona	F_Alta_Z	Descripcion	Notas
Número de llave	Fecha y hora de inserción de la llave	Fecha y hora de baja de la llave	Zona a la que pertenece	Fecha y hora de inserción de la zona	Descripción de la cerradura que abre	Otros datos a tener en cuenta
22	2010-01-01 00:00:01	2010-06-04 10:22:57	COMUNICS	2010-06-04 09:00:22	Armario 1	
23	2010-01-01 00:00:01	2010-06-04 10:22:57	COMUNICS	2010-06-04 09:00:22	Armario 2	
25	2010-01-01 00:00:01	2010-06-04 10:22:57	COMUNICS	2010-06-04 09:00:22	Servidores	Llave de la sala de servidores

Evaluación/ conclusión **El caso de uso '74 - Dar de baja llave' ha sido realizado satisfactoriamente.**

Caso de prueba para el caso de uso '74 - Dar de baja llave'



Caso de prueba	6.10 - Caso de uso 75: Modificar llave
-----------------------	---

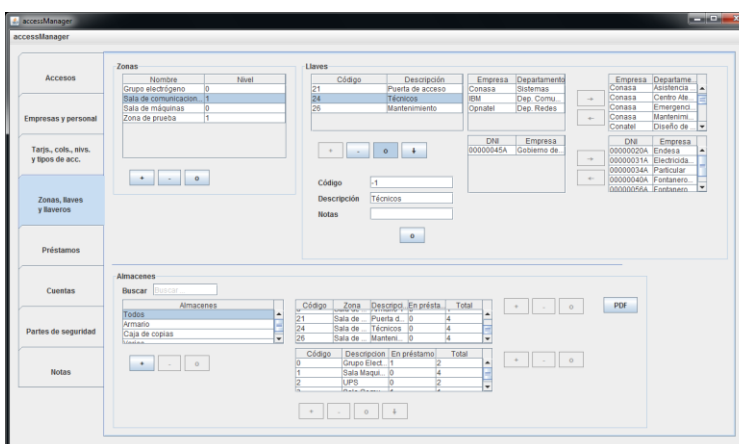
Objetivo	Probar que el proceso de actualización de llaves ha sido diseñado y codificado de forma correcta, obteniendo siempre los resultados esperados.
-----------------	--

Actividades	Acceder al sistema, escoger una zona y llave y modificarla de forma prohibida y permitida, comprobando la reacción del sistema en ambos casos para sacar así conclusiones.
--------------------	--

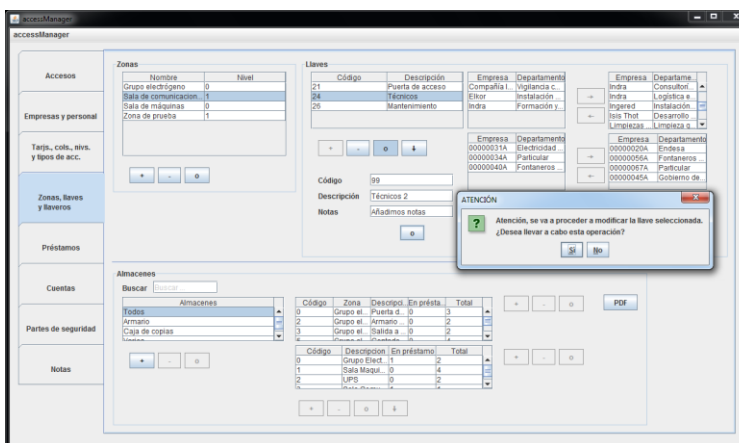
- | | |
|--------------------------------|--|
| Técnicas y herramientas | <ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior. |
|--------------------------------|--|

Realizaremos este caso de uso con la llave número 24 de la zona ‘Sala de comunicaciones’. Para comenzar, intentaremos modificarla de forma que conste con datos erróneos. No cabe posibilidad de provocar errores con los posibles prestatarios de la misma, con que nos basaremos en el código para conseguir lo que queremos:

Datos de entrada



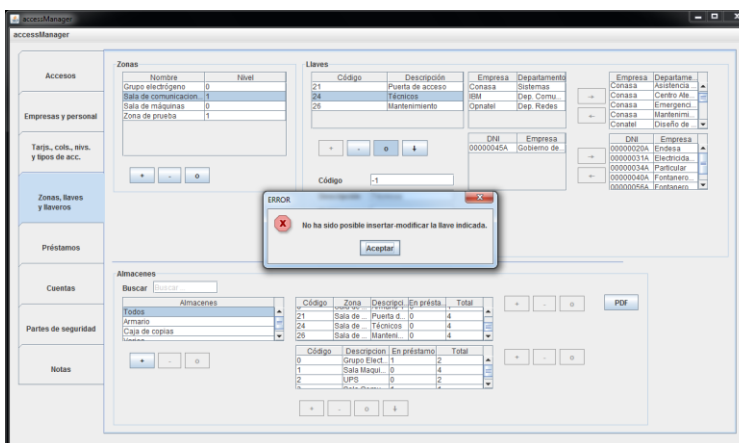
Tras probar a modificarla erróneamente, probaremos a hacerlo válidamente variando todos los campos posibles:



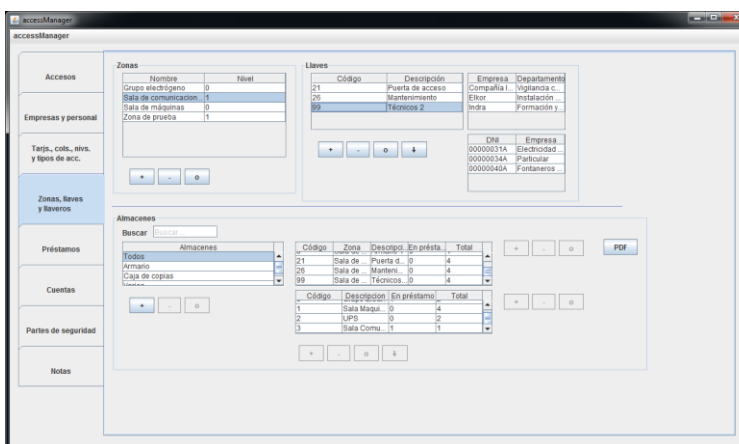
Salida esperada	En el primer caso solo obtendremos del sistema un mensaje de error, mientras que en el segundo el mensaje será de confirmación y la actualización efectiva tanto en la tabla de llaves asociada a zonas como la de almacenes.
------------------------	---



La esperada:



Salida observada



Evaluación/
conclusión

El caso de uso '75 - Modificar llave' ha sido realizado satisfactoriamente.

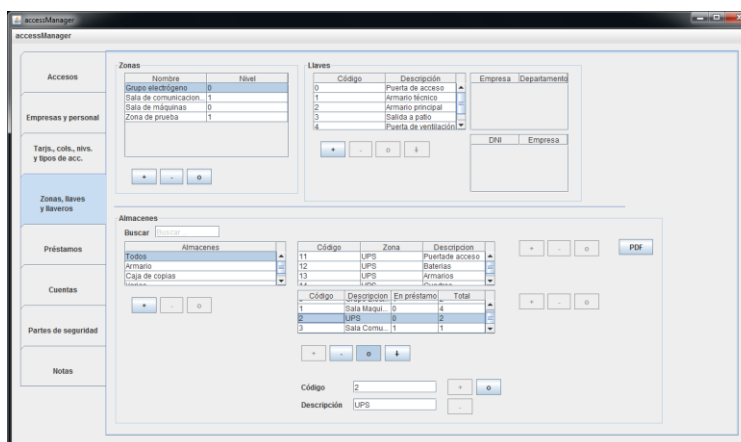
Caso de prueba para el caso de uso '75 - Modificar llave'



Caso de prueba	6.11 - Caso de uso 78: Visualizar llavero																														
Objetivo	Chequear que el sistema muestra los datos y llaves que componen cada llavero que seleccionemos.																														
Actividades	Loguear en el sistema, acceder a la pestaña 'Zonas, llaves y llaveros' y seleccionar varios de los llaveros existentes, comprobando que los datos que se muestran son veraces con la realidad de la base de datos (su instantánea).																														
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos con población de prueba.• Utilidad phpMyAdmin.																														
Datos de entrada	Disponemos de cuatro llaveros creados por defecto de los cuales cualquiera podría servir para esta prueba. En este caso seleccionaremos los de código 1 y 2.																														
Salida esperada	<p>Los llaveros deberán mostrar estos datos...</p> <table border="1"><thead><tr><th>Código</th><th>Descripción</th></tr></thead><tbody><tr><td>1</td><td>Sala Maquinas</td></tr><tr><td>2</td><td>UPS</td></tr></tbody></table> <p>... y estar formados por las siguientes llaves:</p> <table border="1"><thead><tr><th>Código</th><th>Zona</th><th>Descripción</th></tr></thead><tbody><tr><td>6</td><td>Sala de máquinas</td><td>Puerta de acceso</td></tr><tr><td>7</td><td>Sala de máquinas</td><td>Puerta 2</td></tr></tbody></table> <table border="1"><thead><tr><th>Código</th><th>Zona</th><th>Descripción</th></tr></thead><tbody><tr><td>11</td><td>UPS</td><td>Puertade acceso</td></tr><tr><td>12</td><td>UPS</td><td>Baterias</td></tr><tr><td>13</td><td>UPS</td><td>Armanios</td></tr><tr><td>14</td><td>UPS</td><td>Cuadros</td></tr></tbody></table>	Código	Descripción	1	Sala Maquinas	2	UPS	Código	Zona	Descripción	6	Sala de máquinas	Puerta de acceso	7	Sala de máquinas	Puerta 2	Código	Zona	Descripción	11	UPS	Puertade acceso	12	UPS	Baterias	13	UPS	Armanios	14	UPS	Cuadros
Código	Descripción																														
1	Sala Maquinas																														
2	UPS																														
Código	Zona	Descripción																													
6	Sala de máquinas	Puerta de acceso																													
7	Sala de máquinas	Puerta 2																													
Código	Zona	Descripción																													
11	UPS	Puertade acceso																													
12	UPS	Baterias																													
13	UPS	Armanios																													
14	UPS	Cuadros																													
Salida observada	<p>La esperada para el llavero 1...</p>																														



... y para el llavero 2:

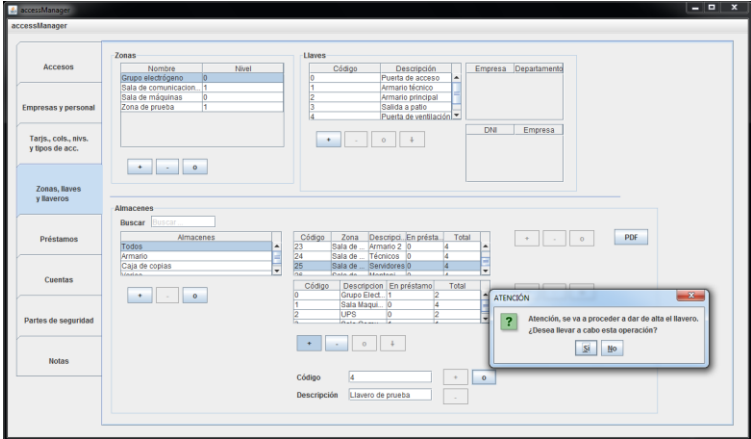
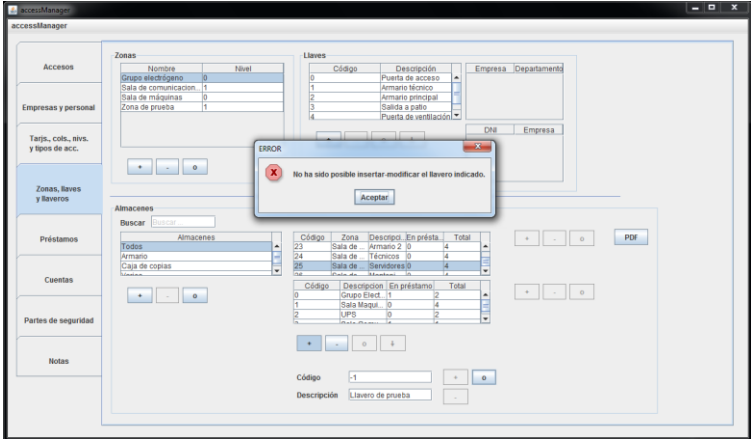


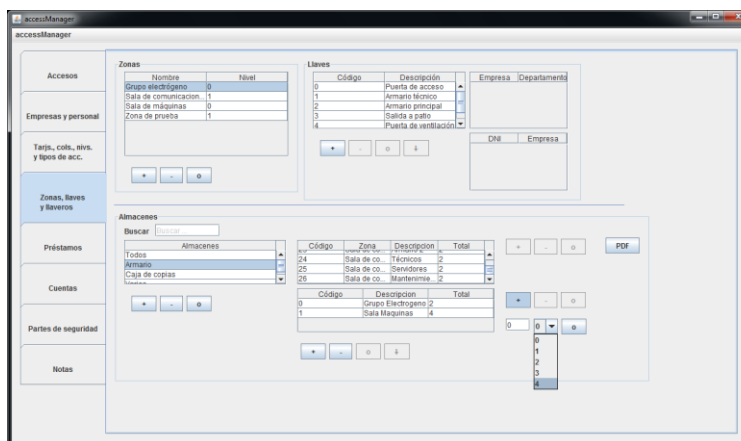
**Evaluación/
conclusión**

El caso de uso '78 - Visualizar llavero' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '78 - Visualizar llavero'



Caso de prueba	6.12 - Caso de uso 79: Añadir llavero
Objetivo	Verificar que accessManager permite al usuario crear llaveros con cualquier número de llaves siempre y cuando los datos que definen el llavero sean válidos.
Actividades	Desplegar la aplicación, situarnos en la pestaña de zonas y probar a crear un llavero con varias llaves, en un principio con datos con un formato erróneo y posteriormente con válidos.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos resultante del caso de prueba anterior.
Datos de entrada	<p>Las llaves que conformarán el llavero son las de código 0, 5, 10, 15, 20 y 25 y los datos del llavero '-1, Llavero de prueba' y '4, Llavero de prueba':</p>  <p>The screenshot shows the 'accessManager' application interface. On the left, there is a navigation menu with options like 'Accesos', 'Empresas y personal', 'Zonas, llaves y llaveros', 'Préstamos', 'Cuentas', 'Partes de seguridad', and 'Notas'. The main area is divided into 'Zonas' and 'Llaves' sections. The 'Llaves' section contains a table with columns: 'Código', 'Descripción', 'Empresa', and 'Departamento'. The table lists several keys with codes 0, 1, 2, 3, and 4. Below the table, there are input fields for 'Código' and 'Descripción'. A dialog box titled 'ATENCIÓN' is overlaid on the interface, asking: 'Atención, se va a proceder a dar de alta el llavero. ¿Desea llevar a cabo esta operación?' with 'Sí' and 'No' buttons.</p>
Salida esperada	El primer llavero deberá generar error al no tener un código comprendido entre 0 y 999, mientras que el segundo entrará a formar parte del sistema.
Salida observada	La esperada:  <p>The screenshot shows the same 'accessManager' application interface. An error dialog box titled 'ERROR' is overlaid, displaying the message: 'No ha sido posible insertar/modificar el llavero indicado.' with an 'Aceptar' button. Below the dialog, the 'Llaves' table is visible, and the 'Código' field in the form below is set to '-1'. The 'Descripción' field contains 'Llavero de prueba'.</p>



El llavero no aparece de forma inmediata en la tabla de llaveros ya que en ésta solo aparecen aquellos que disponen de copias en algún almacén, pero sí que está disponible para ser insertado en el momento que el usuario lo desee.

**Evaluación/
conclusión**

El caso de uso '79 - Añadir llavero' ha sido realizado satisfactoriamente.

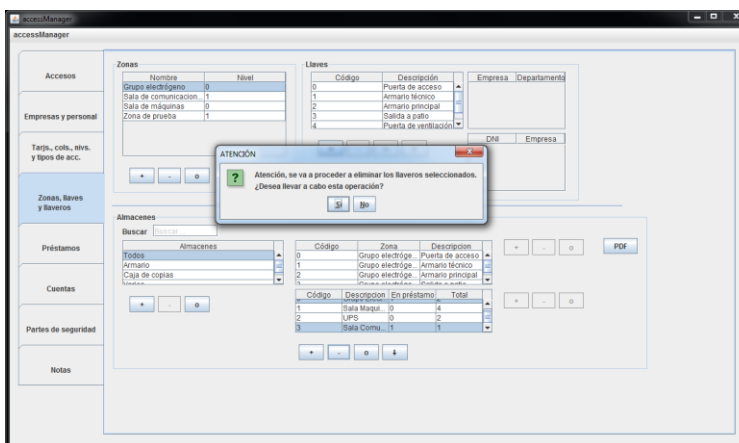
Caso de prueba para el caso de uso '79 - Añadir llavero'



Caso de prueba	6.13 - Caso de uso 80: Borrar llavero																																																																																																																																																																																																																																																																																																
Objetivo	Probar que el borrado de llaveros se realiza sin ningún problema y que éste se ve reflejado en todos los almacenes.																																																																																																																																																																																																																																																																																																
Actividades	Acceder a la sección de llaveros y borrar algunos de ellos, cerciorándonos que dejan de formar parte de la interfaz y de las tablas Llavero, Composicion_Llavero, Almacen_Llavero y Llavero_Prestado en la base de datos.																																																																																																																																																																																																																																																																																																
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior. • Utilidad phpMyAdmin. 																																																																																																																																																																																																																																																																																																
Datos de entrada	<p>Cambiamos de llaveros ya que da igual cuales escoger para realizar la prueba y así tratamos otros distintos a los ya vistos. Para este caso elegimos el 0 y el 3.</p> <p>Esta es la instantánea que disponemos de las tablas que serán afectadas:</p> <table border="1"> <thead> <tr> <th>Cod_Llavero</th> <th>Número de llavero</th> <th>F_Alta_LI</th> <th>Fecha y hora de inserción del llavero</th> <th>F_Baja_LI</th> <th>Fecha y hora de baja del llavero</th> <th>Descripcion</th> <th>Descripción del llavero</th> </tr> </thead> <tbody> <tr><td>0</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>Grupo Electrogeno</td><td></td></tr> <tr><td>1</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>Sala Maquinas</td><td></td></tr> <tr><td>2</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>UPS</td><td></td></tr> <tr><td>3</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>NULL</td><td></td><td>Sala Comunicaciones</td><td></td></tr> <tr><td>4</td><td></td><td>2010-06-04 11:44:44</td><td></td><td>NULL</td><td></td><td>Llavero de prueba</td><td></td></tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Cod_Llavero</th> <th>Número de llavero</th> <th>F_Alta_LI</th> <th>Fecha y hora de inserción del llavero</th> <th>Cod_Llave</th> <th>Número de llave</th> <th>F_Alta_L</th> <th>Fecha y hora de inserción de la llave</th> </tr> </thead> <tbody> <tr><td>0</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>0</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr> <tr><td>0</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>1</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr> <tr><td>0</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>2</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr> <tr><td>0</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>3</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr> <tr><td>1</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>6</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr> <tr><td>1</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>7</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr> <tr><td>2</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>11</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr> <tr><td>2</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>12</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr> <tr><td>2</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>13</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr> <tr><td>2</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>14</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr> <tr><td>3</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>21</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr> <tr><td>3</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>22</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr> <tr><td>3</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>23</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr> <tr><td>3</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>24</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr> <tr><td>3</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>25</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr> <tr><td>4</td><td></td><td>2010-06-04 11:44:44</td><td></td><td>0</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr> <tr><td>4</td><td></td><td>2010-06-04 11:44:44</td><td></td><td>5</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr> <tr><td>4</td><td></td><td>2010-06-04 11:44:44</td><td></td><td>10</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr> <tr><td>4</td><td></td><td>2010-06-04 11:44:44</td><td></td><td>15</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr> <tr><td>4</td><td></td><td>2010-06-04 11:44:44</td><td></td><td>20</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr> <tr><td>4</td><td></td><td>2010-06-04 11:44:44</td><td></td><td>25</td><td></td><td>2010-01-01 00:00:01</td><td></td></tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Cod_Alm</th> <th>Lugar de almacenamiento</th> <th>Cod_Llavero</th> <th>Número de llavero</th> <th>F_Alta_LI</th> <th>Fecha y hora de inserción del llavero</th> <th>Num_Copias</th> <th>Número de copias existentes</th> </tr> </thead> <tbody> <tr><td>A</td><td></td><td>0</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>2</td><td></td></tr> <tr><td>A</td><td></td><td>1</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>4</td><td></td></tr> <tr><td>CC</td><td></td><td>2</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>2</td><td></td></tr> <tr><td>V</td><td></td><td>3</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>1</td><td></td></tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Entrada</th> <th>Fecha y hora de acceso</th> <th>Cod_Llavero</th> <th>Número de llavero prestado</th> <th>F_Alta_LI</th> <th>Fecha y hora de inserción del llavero</th> <th>Devuelto</th> <th>Denota si el llavero ha sido devuelto</th> </tr> </thead> <tbody> <tr><td>2008-12-22</td><td>07:05:00</td><td>3</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>0</td><td></td></tr> <tr><td>2008-12-22</td><td>16:02:00</td><td>0</td><td></td><td>2010-01-01 00:00:01</td><td></td><td>0</td><td></td></tr> </tbody> </table>	Cod_Llavero	Número de llavero	F_Alta_LI	Fecha y hora de inserción del llavero	F_Baja_LI	Fecha y hora de baja del llavero	Descripcion	Descripción del llavero	0		2010-01-01 00:00:01		NULL		Grupo Electrogeno		1		2010-01-01 00:00:01		NULL		Sala Maquinas		2		2010-01-01 00:00:01		NULL		UPS		3		2010-01-01 00:00:01		NULL		Sala Comunicaciones		4		2010-06-04 11:44:44		NULL		Llavero de prueba		Cod_Llavero	Número de llavero	F_Alta_LI	Fecha y hora de inserción del llavero	Cod_Llave	Número de llave	F_Alta_L	Fecha y hora de inserción de la llave	0		2010-01-01 00:00:01		0		2010-01-01 00:00:01		0		2010-01-01 00:00:01		1		2010-01-01 00:00:01		0		2010-01-01 00:00:01		2		2010-01-01 00:00:01		0		2010-01-01 00:00:01		3		2010-01-01 00:00:01		1		2010-01-01 00:00:01		6		2010-01-01 00:00:01		1		2010-01-01 00:00:01		7		2010-01-01 00:00:01		2		2010-01-01 00:00:01		11		2010-01-01 00:00:01		2		2010-01-01 00:00:01		12		2010-01-01 00:00:01		2		2010-01-01 00:00:01		13		2010-01-01 00:00:01		2		2010-01-01 00:00:01		14		2010-01-01 00:00:01		3		2010-01-01 00:00:01		21		2010-01-01 00:00:01		3		2010-01-01 00:00:01		22		2010-01-01 00:00:01		3		2010-01-01 00:00:01		23		2010-01-01 00:00:01		3		2010-01-01 00:00:01		24		2010-01-01 00:00:01		3		2010-01-01 00:00:01		25		2010-01-01 00:00:01		4		2010-06-04 11:44:44		0		2010-01-01 00:00:01		4		2010-06-04 11:44:44		5		2010-01-01 00:00:01		4		2010-06-04 11:44:44		10		2010-01-01 00:00:01		4		2010-06-04 11:44:44		15		2010-01-01 00:00:01		4		2010-06-04 11:44:44		20		2010-01-01 00:00:01		4		2010-06-04 11:44:44		25		2010-01-01 00:00:01		Cod_Alm	Lugar de almacenamiento	Cod_Llavero	Número de llavero	F_Alta_LI	Fecha y hora de inserción del llavero	Num_Copias	Número de copias existentes	A		0		2010-01-01 00:00:01		2		A		1		2010-01-01 00:00:01		4		CC		2		2010-01-01 00:00:01		2		V		3		2010-01-01 00:00:01		1		Entrada	Fecha y hora de acceso	Cod_Llavero	Número de llavero prestado	F_Alta_LI	Fecha y hora de inserción del llavero	Devuelto	Denota si el llavero ha sido devuelto	2008-12-22	07:05:00	3		2010-01-01 00:00:01		0		2008-12-22	16:02:00	0		2010-01-01 00:00:01		0	
Cod_Llavero	Número de llavero	F_Alta_LI	Fecha y hora de inserción del llavero	F_Baja_LI	Fecha y hora de baja del llavero	Descripcion	Descripción del llavero																																																																																																																																																																																																																																																																																										
0		2010-01-01 00:00:01		NULL		Grupo Electrogeno																																																																																																																																																																																																																																																																																											
1		2010-01-01 00:00:01		NULL		Sala Maquinas																																																																																																																																																																																																																																																																																											
2		2010-01-01 00:00:01		NULL		UPS																																																																																																																																																																																																																																																																																											
3		2010-01-01 00:00:01		NULL		Sala Comunicaciones																																																																																																																																																																																																																																																																																											
4		2010-06-04 11:44:44		NULL		Llavero de prueba																																																																																																																																																																																																																																																																																											
Cod_Llavero	Número de llavero	F_Alta_LI	Fecha y hora de inserción del llavero	Cod_Llave	Número de llave	F_Alta_L	Fecha y hora de inserción de la llave																																																																																																																																																																																																																																																																																										
0		2010-01-01 00:00:01		0		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																											
0		2010-01-01 00:00:01		1		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																											
0		2010-01-01 00:00:01		2		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																											
0		2010-01-01 00:00:01		3		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																											
1		2010-01-01 00:00:01		6		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																											
1		2010-01-01 00:00:01		7		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																											
2		2010-01-01 00:00:01		11		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																											
2		2010-01-01 00:00:01		12		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																											
2		2010-01-01 00:00:01		13		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																											
2		2010-01-01 00:00:01		14		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																											
3		2010-01-01 00:00:01		21		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																											
3		2010-01-01 00:00:01		22		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																											
3		2010-01-01 00:00:01		23		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																											
3		2010-01-01 00:00:01		24		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																											
3		2010-01-01 00:00:01		25		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																											
4		2010-06-04 11:44:44		0		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																											
4		2010-06-04 11:44:44		5		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																											
4		2010-06-04 11:44:44		10		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																											
4		2010-06-04 11:44:44		15		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																											
4		2010-06-04 11:44:44		20		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																											
4		2010-06-04 11:44:44		25		2010-01-01 00:00:01																																																																																																																																																																																																																																																																																											
Cod_Alm	Lugar de almacenamiento	Cod_Llavero	Número de llavero	F_Alta_LI	Fecha y hora de inserción del llavero	Num_Copias	Número de copias existentes																																																																																																																																																																																																																																																																																										
A		0		2010-01-01 00:00:01		2																																																																																																																																																																																																																																																																																											
A		1		2010-01-01 00:00:01		4																																																																																																																																																																																																																																																																																											
CC		2		2010-01-01 00:00:01		2																																																																																																																																																																																																																																																																																											
V		3		2010-01-01 00:00:01		1																																																																																																																																																																																																																																																																																											
Entrada	Fecha y hora de acceso	Cod_Llavero	Número de llavero prestado	F_Alta_LI	Fecha y hora de inserción del llavero	Devuelto	Denota si el llavero ha sido devuelto																																																																																																																																																																																																																																																																																										
2008-12-22	07:05:00	3		2010-01-01 00:00:01		0																																																																																																																																																																																																																																																																																											
2008-12-22	16:02:00	0		2010-01-01 00:00:01		0																																																																																																																																																																																																																																																																																											



... y una visión de las ocurrencias que tienen en la interfaz:



Salida esperada

El borrado de los llaveros 0 y 3 debería provocar que todas las tuplas que los nombran en las tablas mencionadas desaparezcan, así como de la interfaz de usuario.

La esperada:

Cod_Llavero	Número de llavero	F_Alta_LI	Fecha y hora de inserción del llavero	F_Baja_LI	Fecha y hora de baja del llavero	Descripcion	Descripción del llavero
1		2010-01-01 00:00:01		NULL			Sala Maquinas
2		2010-01-01 00:00:01		NULL			UPS
4		2010-06-04 11:44:44		NULL			Llavero de prueba

Cod_Llavero	Número de llavero	F_Alta_LI	Fecha y hora de inserción del llavero	Cod_Llave	Número de llave	F_Alta_L	Fecha y hora de inserción de la llave
1		2010-01-01 00:00:01		6		2010-01-01 00:00:01	
1		2010-01-01 00:00:01		7		2010-01-01 00:00:01	
2		2010-01-01 00:00:01		11		2010-01-01 00:00:01	
2		2010-01-01 00:00:01		12		2010-01-01 00:00:01	
2		2010-01-01 00:00:01		13		2010-01-01 00:00:01	
2		2010-01-01 00:00:01		14		2010-01-01 00:00:01	
4		2010-06-04 11:44:44		0		2010-01-01 00:00:01	
4		2010-06-04 11:44:44		5		2010-01-01 00:00:01	
4		2010-06-04 11:44:44		10		2010-01-01 00:00:01	
4		2010-06-04 11:44:44		15		2010-01-01 00:00:01	
4		2010-06-04 11:44:44		20		2010-01-01 00:00:01	
4		2010-06-04 11:44:44		25		2010-01-01 00:00:01	

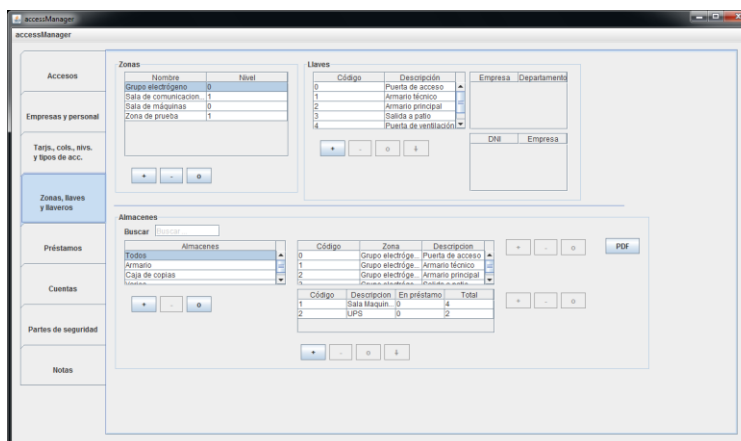
Salida observada

Cod_Alm	Lugar de almacenamiento	Cod_Llavero	Número de llavero	F_Alta_LI	Fecha y hora de inserción del llavero	Num_Copias	Número de copias existentes
A		1		2010-01-01 00:00:01		4	
CC		2		2010-01-01 00:00:01		2	

```

MySQL ha devuelto un valor vacio (i.e., cero columnas). (La consulta tardó 0.0003 seg)
SELECT *
FROM Llavero_Prestado
LIMIT 0, 20

```



**Evaluación/
conclusión**

El caso de uso '80 - Borrar llavero' ha sido realizado satisfactoriamente.

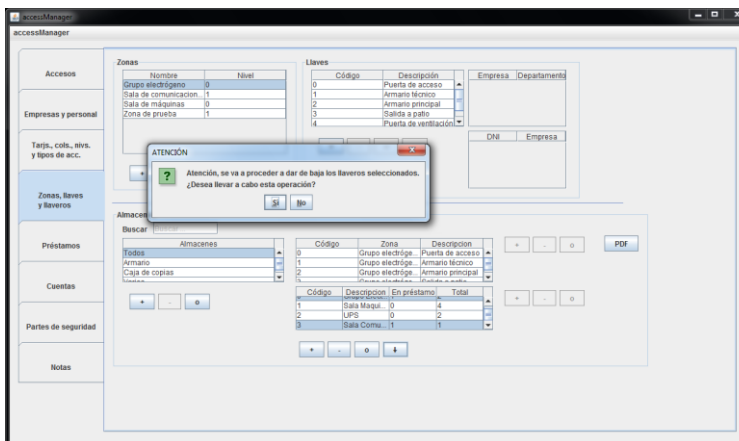
Caso de prueba para el caso de uso '80 - Borrar llavero'



Caso de prueba	6.14 - Caso de uso 81: Dar de baja llavero																																																																																																																																																																																																																																								
Objetivo	Comprobar que el usuario puede realizar la baja de llaveros con el simple hecho de seleccionar los deseados y clicar el botón 'Dar de baja'.																																																																																																																																																																																																																																								
Actividades	Desplegar la aplicación, dirigirnos al almacén 'Todos' y dar de baja los llaveros que deseemos, comprobando que desaparecen de la interfaz pero no del sistema.																																																																																																																																																																																																																																								
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. • Utilidad phpMyAdmin. 																																																																																																																																																																																																																																								
Datos de entrada	<p>El sistema se encuentra en el estado inicial que describe la población de prueba creada. Atendiendo a las tablas que se verán afectadas tendríamos:</p> <table border="1"> <thead> <tr> <th>Cod_Llavero</th> <th>Número de llavero</th> <th>F_Alta_LI</th> <th>Fecha y hora de inserción del llavero</th> <th>F_Baja_LI</th> <th>Fecha y hora de baja del llavero</th> <th>Descripcion</th> <th>Descripción del llavero</th> </tr> </thead> <tbody> <tr><td></td><td>0</td><td>2010-01-01</td><td>00:00:01</td><td>NULL</td><td></td><td>Grupo Electrogeno</td><td></td></tr> <tr><td></td><td>1</td><td>2010-01-01</td><td>00:00:01</td><td>NULL</td><td></td><td>Sala Maquinas</td><td></td></tr> <tr><td></td><td>2</td><td>2010-01-01</td><td>00:00:01</td><td>NULL</td><td></td><td>UPS</td><td></td></tr> <tr><td></td><td>3</td><td>2010-01-01</td><td>00:00:01</td><td>NULL</td><td></td><td>Sala Comunicaciones</td><td></td></tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Cod_Llavero</th> <th>Número de llavero</th> <th>F_Alta_LI</th> <th>Fecha y hora de inserción del llavero</th> <th>Cod_Llave</th> <th>Número de llave</th> <th>F_Alta_L</th> <th>Fecha y hora de inserción de la llave</th> </tr> </thead> <tbody> <tr><td></td><td>0</td><td>2010-01-01</td><td>00:00:01</td><td></td><td>0</td><td>2010-01-01</td><td>00:00:01</td></tr> <tr><td></td><td>0</td><td>2010-01-01</td><td>00:00:01</td><td></td><td>1</td><td>2010-01-01</td><td>00:00:01</td></tr> <tr><td></td><td>0</td><td>2010-01-01</td><td>00:00:01</td><td></td><td>2</td><td>2010-01-01</td><td>00:00:01</td></tr> <tr><td></td><td>0</td><td>2010-01-01</td><td>00:00:01</td><td></td><td>3</td><td>2010-01-01</td><td>00:00:01</td></tr> <tr><td></td><td>1</td><td>2010-01-01</td><td>00:00:01</td><td></td><td>6</td><td>2010-01-01</td><td>00:00:01</td></tr> <tr><td></td><td>1</td><td>2010-01-01</td><td>00:00:01</td><td></td><td>7</td><td>2010-01-01</td><td>00:00:01</td></tr> <tr><td></td><td>2</td><td>2010-01-01</td><td>00:00:01</td><td></td><td>11</td><td>2010-01-01</td><td>00:00:01</td></tr> <tr><td></td><td>2</td><td>2010-01-01</td><td>00:00:01</td><td></td><td>12</td><td>2010-01-01</td><td>00:00:01</td></tr> <tr><td></td><td>2</td><td>2010-01-01</td><td>00:00:01</td><td></td><td>13</td><td>2010-01-01</td><td>00:00:01</td></tr> <tr><td></td><td>2</td><td>2010-01-01</td><td>00:00:01</td><td></td><td>14</td><td>2010-01-01</td><td>00:00:01</td></tr> <tr><td></td><td>3</td><td>2010-01-01</td><td>00:00:01</td><td></td><td>21</td><td>2010-01-01</td><td>00:00:01</td></tr> <tr><td></td><td>3</td><td>2010-01-01</td><td>00:00:01</td><td></td><td>22</td><td>2010-01-01</td><td>00:00:01</td></tr> <tr><td></td><td>3</td><td>2010-01-01</td><td>00:00:01</td><td></td><td>23</td><td>2010-01-01</td><td>00:00:01</td></tr> <tr><td></td><td>3</td><td>2010-01-01</td><td>00:00:01</td><td></td><td>24</td><td>2010-01-01</td><td>00:00:01</td></tr> <tr><td></td><td>3</td><td>2010-01-01</td><td>00:00:01</td><td></td><td>25</td><td>2010-01-01</td><td>00:00:01</td></tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Cod_Alm</th> <th>Lugar de almacenamiento</th> <th>Cod_Llavero</th> <th>Número de llavero</th> <th>F_Alta_LI</th> <th>Fecha y hora de inserción del llavero</th> <th>Num_Copias</th> <th>Número de copias existentes</th> </tr> </thead> <tbody> <tr><td>A</td><td></td><td>0</td><td>2010-01-01</td><td>00:00:01</td><td></td><td>2</td><td></td></tr> <tr><td>A</td><td></td><td>1</td><td>2010-01-01</td><td>00:00:01</td><td></td><td>4</td><td></td></tr> <tr><td>CC</td><td></td><td>2</td><td>2010-01-01</td><td>00:00:01</td><td></td><td>2</td><td></td></tr> <tr><td>V</td><td></td><td>3</td><td>2010-01-01</td><td>00:00:01</td><td></td><td>1</td><td></td></tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Entrada</th> <th>Fecha y hora de acceso</th> <th>Cod_Llavero</th> <th>Número de llavero prestado</th> <th>F_Alta_LI</th> <th>Fecha y hora de inserción del llavero</th> <th>Devoluto</th> <th>Denota si el llavero ha sido devuelto</th> </tr> </thead> <tbody> <tr><td></td><td>2008-12-22 07:05:00</td><td></td><td>3</td><td>2010-01-01</td><td>00:00:01</td><td></td><td>0</td></tr> <tr><td></td><td>2008-12-22 16:02:00</td><td></td><td>0</td><td>2010-01-01</td><td>00:00:01</td><td></td><td>0</td></tr> </tbody> </table>	Cod_Llavero	Número de llavero	F_Alta_LI	Fecha y hora de inserción del llavero	F_Baja_LI	Fecha y hora de baja del llavero	Descripcion	Descripción del llavero		0	2010-01-01	00:00:01	NULL		Grupo Electrogeno			1	2010-01-01	00:00:01	NULL		Sala Maquinas			2	2010-01-01	00:00:01	NULL		UPS			3	2010-01-01	00:00:01	NULL		Sala Comunicaciones		Cod_Llavero	Número de llavero	F_Alta_LI	Fecha y hora de inserción del llavero	Cod_Llave	Número de llave	F_Alta_L	Fecha y hora de inserción de la llave		0	2010-01-01	00:00:01		0	2010-01-01	00:00:01		0	2010-01-01	00:00:01		1	2010-01-01	00:00:01		0	2010-01-01	00:00:01		2	2010-01-01	00:00:01		0	2010-01-01	00:00:01		3	2010-01-01	00:00:01		1	2010-01-01	00:00:01		6	2010-01-01	00:00:01		1	2010-01-01	00:00:01		7	2010-01-01	00:00:01		2	2010-01-01	00:00:01		11	2010-01-01	00:00:01		2	2010-01-01	00:00:01		12	2010-01-01	00:00:01		2	2010-01-01	00:00:01		13	2010-01-01	00:00:01		2	2010-01-01	00:00:01		14	2010-01-01	00:00:01		3	2010-01-01	00:00:01		21	2010-01-01	00:00:01		3	2010-01-01	00:00:01		22	2010-01-01	00:00:01		3	2010-01-01	00:00:01		23	2010-01-01	00:00:01		3	2010-01-01	00:00:01		24	2010-01-01	00:00:01		3	2010-01-01	00:00:01		25	2010-01-01	00:00:01	Cod_Alm	Lugar de almacenamiento	Cod_Llavero	Número de llavero	F_Alta_LI	Fecha y hora de inserción del llavero	Num_Copias	Número de copias existentes	A		0	2010-01-01	00:00:01		2		A		1	2010-01-01	00:00:01		4		CC		2	2010-01-01	00:00:01		2		V		3	2010-01-01	00:00:01		1		Entrada	Fecha y hora de acceso	Cod_Llavero	Número de llavero prestado	F_Alta_LI	Fecha y hora de inserción del llavero	Devoluto	Denota si el llavero ha sido devuelto		2008-12-22 07:05:00		3	2010-01-01	00:00:01		0		2008-12-22 16:02:00		0	2010-01-01	00:00:01		0
Cod_Llavero	Número de llavero	F_Alta_LI	Fecha y hora de inserción del llavero	F_Baja_LI	Fecha y hora de baja del llavero	Descripcion	Descripción del llavero																																																																																																																																																																																																																																		
	0	2010-01-01	00:00:01	NULL		Grupo Electrogeno																																																																																																																																																																																																																																			
	1	2010-01-01	00:00:01	NULL		Sala Maquinas																																																																																																																																																																																																																																			
	2	2010-01-01	00:00:01	NULL		UPS																																																																																																																																																																																																																																			
	3	2010-01-01	00:00:01	NULL		Sala Comunicaciones																																																																																																																																																																																																																																			
Cod_Llavero	Número de llavero	F_Alta_LI	Fecha y hora de inserción del llavero	Cod_Llave	Número de llave	F_Alta_L	Fecha y hora de inserción de la llave																																																																																																																																																																																																																																		
	0	2010-01-01	00:00:01		0	2010-01-01	00:00:01																																																																																																																																																																																																																																		
	0	2010-01-01	00:00:01		1	2010-01-01	00:00:01																																																																																																																																																																																																																																		
	0	2010-01-01	00:00:01		2	2010-01-01	00:00:01																																																																																																																																																																																																																																		
	0	2010-01-01	00:00:01		3	2010-01-01	00:00:01																																																																																																																																																																																																																																		
	1	2010-01-01	00:00:01		6	2010-01-01	00:00:01																																																																																																																																																																																																																																		
	1	2010-01-01	00:00:01		7	2010-01-01	00:00:01																																																																																																																																																																																																																																		
	2	2010-01-01	00:00:01		11	2010-01-01	00:00:01																																																																																																																																																																																																																																		
	2	2010-01-01	00:00:01		12	2010-01-01	00:00:01																																																																																																																																																																																																																																		
	2	2010-01-01	00:00:01		13	2010-01-01	00:00:01																																																																																																																																																																																																																																		
	2	2010-01-01	00:00:01		14	2010-01-01	00:00:01																																																																																																																																																																																																																																		
	3	2010-01-01	00:00:01		21	2010-01-01	00:00:01																																																																																																																																																																																																																																		
	3	2010-01-01	00:00:01		22	2010-01-01	00:00:01																																																																																																																																																																																																																																		
	3	2010-01-01	00:00:01		23	2010-01-01	00:00:01																																																																																																																																																																																																																																		
	3	2010-01-01	00:00:01		24	2010-01-01	00:00:01																																																																																																																																																																																																																																		
	3	2010-01-01	00:00:01		25	2010-01-01	00:00:01																																																																																																																																																																																																																																		
Cod_Alm	Lugar de almacenamiento	Cod_Llavero	Número de llavero	F_Alta_LI	Fecha y hora de inserción del llavero	Num_Copias	Número de copias existentes																																																																																																																																																																																																																																		
A		0	2010-01-01	00:00:01		2																																																																																																																																																																																																																																			
A		1	2010-01-01	00:00:01		4																																																																																																																																																																																																																																			
CC		2	2010-01-01	00:00:01		2																																																																																																																																																																																																																																			
V		3	2010-01-01	00:00:01		1																																																																																																																																																																																																																																			
Entrada	Fecha y hora de acceso	Cod_Llavero	Número de llavero prestado	F_Alta_LI	Fecha y hora de inserción del llavero	Devoluto	Denota si el llavero ha sido devuelto																																																																																																																																																																																																																																		
	2008-12-22 07:05:00		3	2010-01-01	00:00:01		0																																																																																																																																																																																																																																		
	2008-12-22 16:02:00		0	2010-01-01	00:00:01		0																																																																																																																																																																																																																																		



Estimamos oportuno dar de baja los llaveros 0 y 3, para que así los cambios afecten a todas las tablas (en este caso los cambios se materializarían en el no borrado de tuplas):



Salida esperada

Gráficamente, la misma que el caso de prueba anterior; físicamente, las tuplas se conservarán pero todas las que hagan relación a estos llaveros habrán sido actualizadas modificando la fecha de baja de los llaveros.

Salida observada

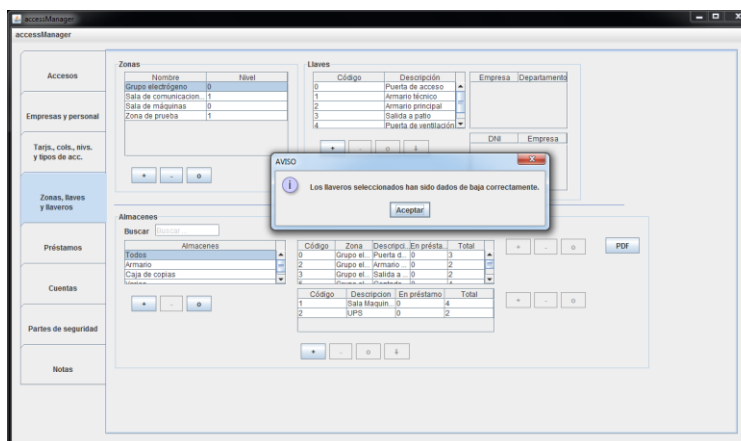
La esperada:

Cod_Llavero	Número de llavero	F_Alta_LI	Fecha y hora de inserción del llavero	F_Baja_LI	Fecha y hora de baja del llavero	Descripcion	Descripción del llavero
0	2010-01-01 00:00:01	2010-01-01 00:00:01		2010-06-04 12:48:00		Grupo Electrogeno	
1	2010-01-01 00:00:01	2010-01-01 00:00:01		NULL		Sala Maquinas	
2	2010-01-01 00:00:01	2010-01-01 00:00:01		NULL		UPS	
3	2010-01-01 00:00:01	2010-01-01 00:00:01		2010-06-04 12:48:00		Sala Comunicaciones	

Cod_Llavero	Número de llavero	F_Alta_LI	Fecha y hora de inserción del llavero	Cod_Llave	Número de llave	F_Alta_L	Fecha y hora de inserción de la llave
0	2010-01-01 00:00:01			0	2010-01-01 00:00:01		
0	2010-01-01 00:00:01			1	2010-01-01 00:00:01		
0	2010-01-01 00:00:01			2	2010-01-01 00:00:01		
0	2010-01-01 00:00:01			3	2010-01-01 00:00:01		
1	2010-01-01 00:00:01			6	2010-01-01 00:00:01		
1	2010-01-01 00:00:01			7	2010-01-01 00:00:01		
2	2010-01-01 00:00:01			11	2010-01-01 00:00:01		
2	2010-01-01 00:00:01			12	2010-01-01 00:00:01		
2	2010-01-01 00:00:01			13	2010-01-01 00:00:01		
2	2010-01-01 00:00:01			14	2010-01-01 00:00:01		
3	2010-01-01 00:00:01			21	2010-01-01 00:00:01		
3	2010-01-01 00:00:01			22	2010-01-01 00:00:01		
3	2010-01-01 00:00:01			23	2010-01-01 00:00:01		
3	2010-01-01 00:00:01			24	2010-01-01 00:00:01		
3	2010-01-01 00:00:01			25	2010-01-01 00:00:01		

Cod_Alm	Lugar de almacenamiento	Cod_Llavero	Número de llavero	F_Alta_LI	Fecha y hora de inserción del llavero	Num_Copias	Número de copias existentes
A		0	2010-01-01 00:00:01			2	
A		1	2010-01-01 00:00:01			4	
CC		2	2010-01-01 00:00:01			2	
V		3	2010-01-01 00:00:01			1	

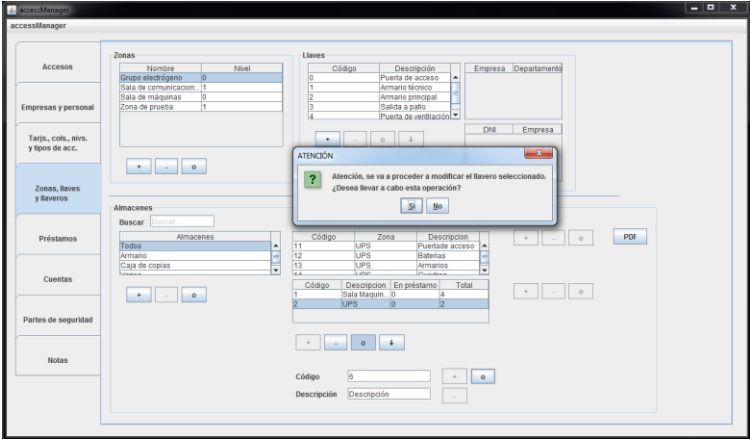
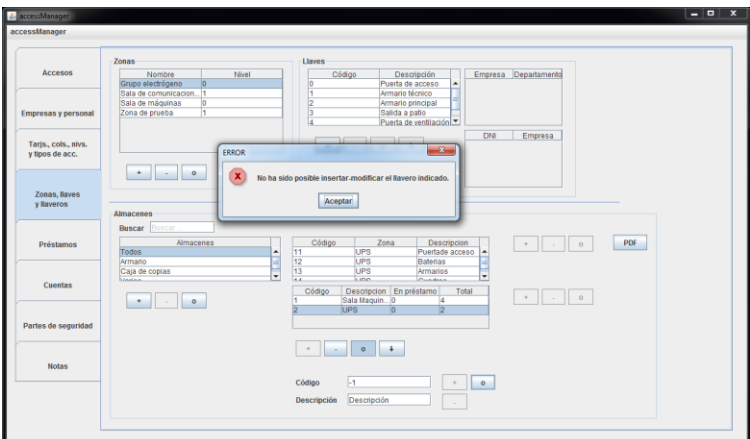
Entrada	Fecha y hora de acceso	Cod_Llavero	Número de llavero prestado	F_Alta_LI	Fecha y hora de inserción del llavero	Devuelto	Denota si el llavero ha sido devuelto
2008-12-22 07:05:00		3	2010-01-01 00:00:01			0	
2008-12-22 16:02:00		0	2010-01-01 00:00:01			0	

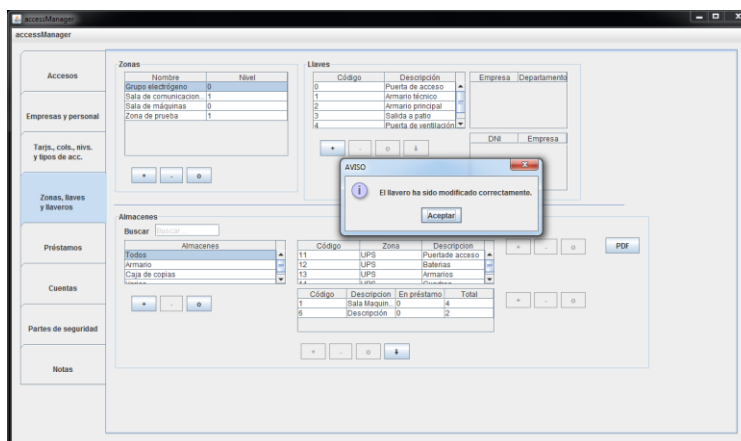


**Evaluación/
conclusión**

El caso de uso '81 - Dar de baja llavero' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '81 - Dar de baja llavero'

Caso de prueba	6.15 - Caso de uso 82: Modificar llavero
Objetivo	Probar que se pueden modificar los datos de cualquier llavero si los reemplazamos por otros válidos.
Actividades	Acceder a la aplicación, situarnos en la tab de zonas y seleccionar cualquiera de los llaveros existentes, modificando sus datos de manera errónea y válida para así comprobar cómo reacciona el sistema.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior.
Datos de entrada	<p>De entre los dos llaveros de los que disponemos, variamos el segundo de manera errónea ('-1, Descripción') y válida ('6, Descripción'):</p> 
Salida esperada	El primer intento debería ser en vano, mientras que el segundo debería producir la actualización del llavero en el sistema.
Salida observada	<p>La esperada:</p> 

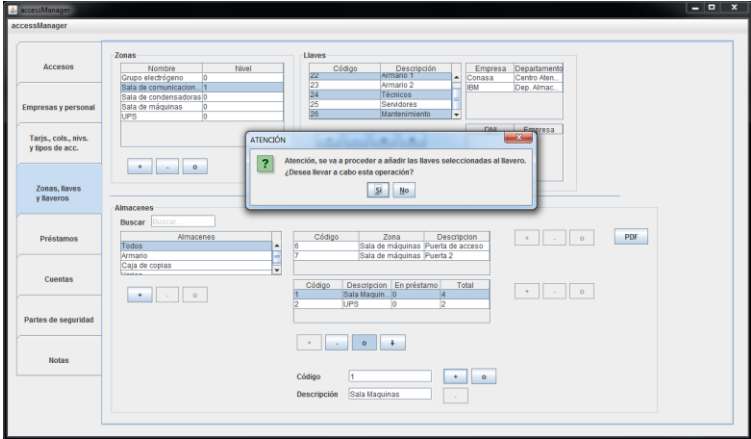
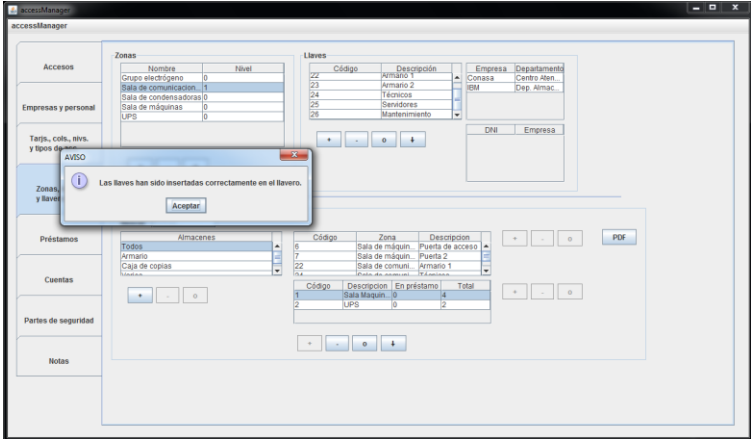


**Evaluación/
conclusión**

El caso de uso '82 - Modificar llavero' ha sido realizado satisfactoriamente.

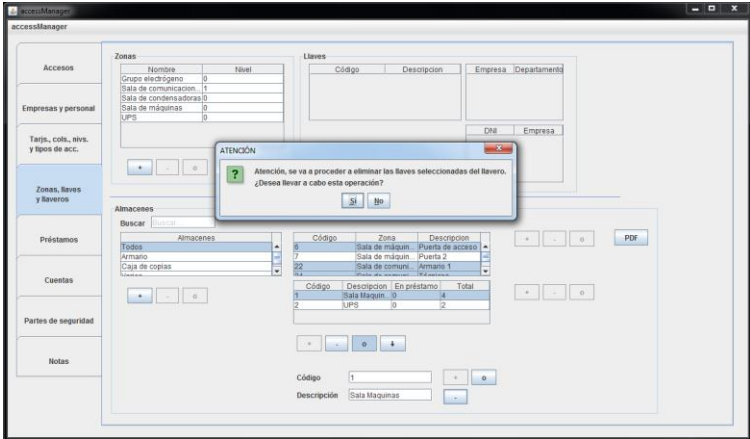
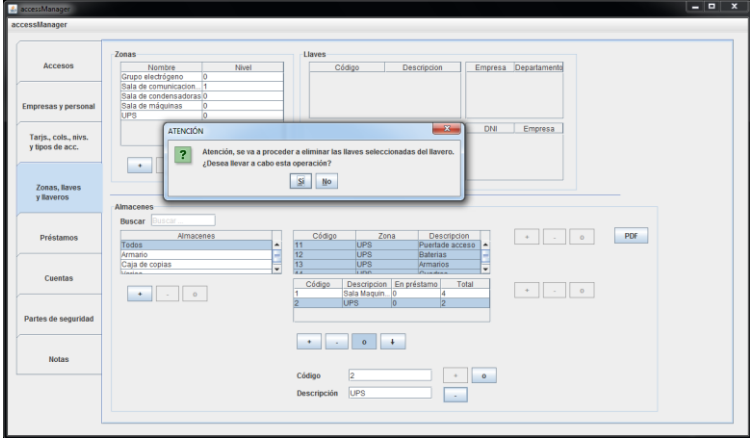
Caso de prueba para el caso de uso '82 - Modificar llavero'



Caso de prueba	6.16 - Caso de uso 83: Añadir llave
Objetivo	Verificar que el usuario podrá añadir las llaves que desee a la composición del llavero, no existiendo la posibilidad de producir ningún error.
Actividades	Loguear en el sistema, dirigirnos a la pestaña 'Zonas, llaves y llaveros', seleccionar un llavero cualquiera y añadirle varias llaves, comprobando que no se genera ninguna situación inesperada.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior.
Datos de entrada	<p>De los dos llaveros de los que disponemos seleccionaremos en este caso el primero y le añadiremos las llaves 22, 24 y 26:</p> 
Salida esperada	Si el llavero consta actualmente con las llaves 6 y 7, tras producirse la inserción deberían unirse a ellas las llaves seleccionadas en la tabla de llaves asociadas a zonas.
Salida observada	<p>La esperada:</p> 
Evaluación/conclusión	El caso de uso '83 - Modificar llavero' ha sido realizado satisfactoriamente.

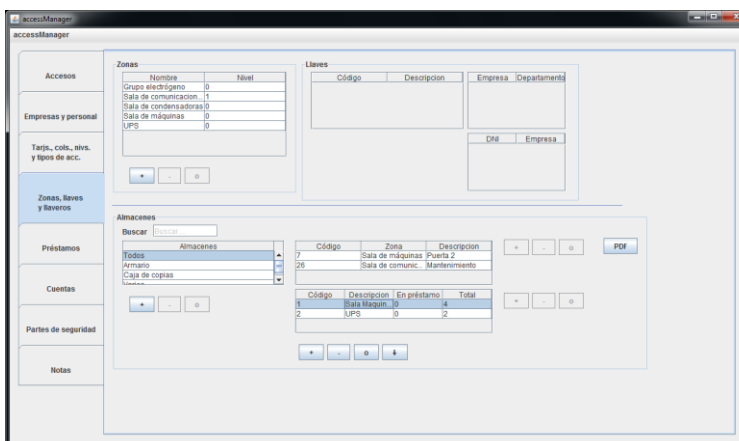
Caso de prueba para el caso de uso '83 - Modificar llavero'



Caso de prueba	6.17 - Caso de uso 84: Borrar llave
Objetivo	Chequear que podemos eliminar llaves de cualquier llavero siempre que lo creamos necesario.
Actividades	Loguear en el sistema y dirigirnos a la tab de zonas, donde seleccionaremos un llavero y eliminaremos algunas de sus llaves, comprobando que el proceso se desarrolla según lo establecido.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos resultante del caso de prueba anterior.
Datos de entrada	<p>Disponemos de los llaveros 1 y 2 para poder realizar este caso de prueba. Haremos dos pruebas:</p> <ul style="list-style-type: none">• Con el llavero 1: Borraremos parte de sus llaves:  <ul style="list-style-type: none">• Con el llavero 2: Borraremos todas sus llaves: 
Salida esperada	<p>En el primer llavero solo deberán permanecer las llaves 7 y 26, que son las únicas no seleccionadas en el borrado.</p> <p>El llavero número 2 debería ser eliminado del sistema (las llaves por este procedimiento y el llavero gracias a uno de los disparadores creados) por no disponer de ninguna llave.</p>

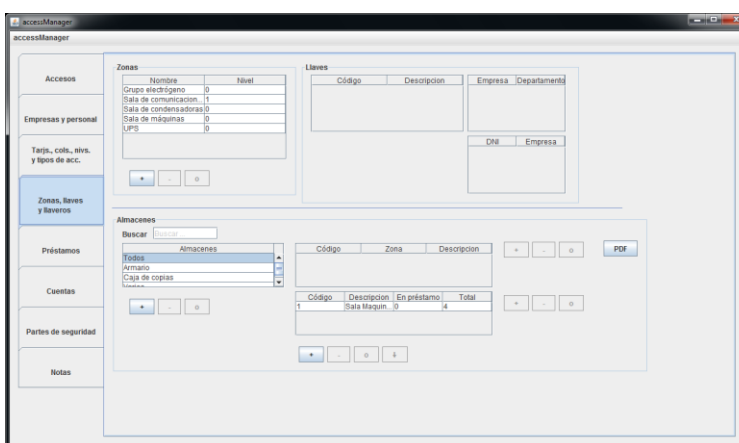


La espera. La primera prueba deja como resultado el llavero 1 con las llaves 6 y 27...



Salida observada

... mientras que la segunda provoca el borrado del llavero:



Evaluación/
conclusión

El caso de uso '84 - Borrar llave' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '84 - Borrar llave'

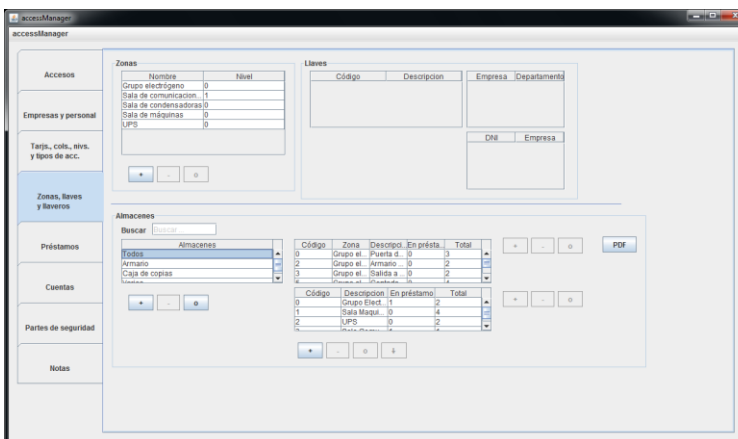


Caso de prueba	6.18 - Caso de uso 87: Visualizar almacén de llaves																																																																																																																																																																																																																												
Objetivo	Comprobar que al clicar en cualquiera de los almacenes las llaves y llaveros que contiene son visualizados en las tablas destinadas para tal uso (las dos que quedan a su derecha).																																																																																																																																																																																																																												
Actividades	Acceder al sistema, clicar en la pestaña de zonas y comprobar que cada vez que clicamos un almacén se visualizan los datos correctos (aquellos que constan en la base de datos).																																																																																																																																																																																																																												
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. • Utilidad phpMyAdmin. 																																																																																																																																																																																																																												
Datos de entrada	Optaremos por comprobar el almacén general ('Todos') y uno en particular (por ejemplo 'Armario').																																																																																																																																																																																																																												
	<p>En el primer caso, la salida debería ser esta...</p> <table border="1"> <thead> <tr> <th>Código</th> <th>Zona</th> <th>Descripcion</th> <th>En préstamo</th> <th>Total</th> </tr> </thead> <tbody> <tr><td>0</td><td>Grupo electrógeno</td><td>Puerta de acceso</td><td>0</td><td>3</td></tr> <tr><td>2</td><td>Grupo electrógeno</td><td>Armario principal</td><td>0</td><td>2</td></tr> <tr><td>3</td><td>Grupo electrógeno</td><td>Salida a patio</td><td>0</td><td>2</td></tr> <tr><td>5</td><td>Grupo electrógeno</td><td>Contadores</td><td>0</td><td>4</td></tr> <tr><td>6</td><td>Sala de máquinas</td><td>Puerta de acceso</td><td>0</td><td>4</td></tr> <tr><td>7</td><td>Sala de máquinas</td><td>Puerta 2</td><td>0</td><td>2</td></tr> <tr><td>8</td><td>Sala de máquinas</td><td>Armario 1</td><td>0</td><td>1</td></tr> <tr><td>9</td><td>Sala de máquinas</td><td>Armario 2</td><td>0</td><td>1</td></tr> <tr><td>10</td><td>Sala de máquinas</td><td>Cuadros de fuerzas</td><td>0</td><td>3</td></tr> <tr><td>11</td><td>UPS</td><td>Puertade acceso</td><td>0</td><td>2</td></tr> <tr><td>12</td><td>UPS</td><td>Baterias</td><td>0</td><td>2</td></tr> <tr><td>13</td><td>UPS</td><td>Armarios</td><td>0</td><td>2</td></tr> <tr><td>14</td><td>UPS</td><td>Cuadros</td><td>0</td><td>2</td></tr> <tr><td>15</td><td>Sala de condensadoras</td><td>Puerta de acceso</td><td>0</td><td>1</td></tr> <tr><td>16</td><td>Sala de condensadoras</td><td>Armario 1</td><td>0</td><td>2</td></tr> <tr><td>17</td><td>Sala de condensadoras</td><td>Armario 2</td><td>1</td><td>5</td></tr> <tr><td>18</td><td>Sala de condensadoras</td><td>Mantenimiento</td><td>1</td><td>4</td></tr> <tr><td>19</td><td>Sala de condensadoras</td><td>Cuadros</td><td>1</td><td>4</td></tr> <tr><td>20</td><td>Sala de condensadoras</td><td>Cajones</td><td>1</td><td>4</td></tr> <tr><td>21</td><td>Sala de comunicaciones</td><td>Puerta de acceso</td><td>0</td><td>4</td></tr> <tr><td>22</td><td>Sala de comunicaciones</td><td>Armario 1</td><td>0</td><td>4</td></tr> <tr><td>23</td><td>Sala de comunicaciones</td><td>Armario 2</td><td>0</td><td>4</td></tr> <tr><td>24</td><td>Sala de comunicaciones</td><td>Técnicos</td><td>0</td><td>4</td></tr> </tbody> </table> <p>...</p> <table border="1"> <thead> <tr> <th>Código</th> <th>Descripcion</th> <th>En préstamo</th> <th>Total</th> </tr> </thead> <tbody> <tr><td>0</td><td>Grupo Electrogeno</td><td>1</td><td>2</td></tr> <tr><td>1</td><td>Sala Maquinas</td><td>0</td><td>4</td></tr> <tr><td>2</td><td>UPS</td><td>0</td><td>2</td></tr> <tr><td>3</td><td>Sala Comunicaciones</td><td>1</td><td>1</td></tr> </tbody> </table> <p>... mientras que en el segundo esta otra:</p> <table border="1"> <thead> <tr> <th>Código</th> <th>Zona</th> <th>Descripcion</th> <th>Total</th> </tr> </thead> <tbody> <tr><td>0</td><td>Grupo electrógeno</td><td>Puerta de acceso</td><td>1</td></tr> <tr><td>5</td><td>Grupo electrógeno</td><td>Contadores</td><td>2</td></tr> <tr><td>6</td><td>Sala de máquinas</td><td>Puerta de acceso</td><td>2</td></tr> <tr><td>7</td><td>Sala de máquinas</td><td>Puerta 2</td><td>1</td></tr> <tr><td>8</td><td>Sala de máquinas</td><td>Armario 1</td><td>1</td></tr> <tr><td>9</td><td>Sala de máquinas</td><td>Armario 2</td><td>1</td></tr> <tr><td>10</td><td>Sala de máquinas</td><td>Cuadros de fuerzas</td><td>2</td></tr> <tr><td>15</td><td>Sala de condensadoras</td><td>Puerta de acceso</td><td>1</td></tr> <tr><td>16</td><td>Sala de condensadoras</td><td>Armario 1</td><td>1</td></tr> <tr><td>17</td><td>Sala de condensadoras</td><td>Armario 2</td><td>3</td></tr> <tr><td>18</td><td>Sala de condensadoras</td><td>Mantenimiento</td><td>2</td></tr> <tr><td>19</td><td>Sala de condensadoras</td><td>Cuadros</td><td>2</td></tr> <tr><td>20</td><td>Sala de condensadoras</td><td>Cajones</td><td>2</td></tr> <tr><td>21</td><td>Sala de comunicaciones</td><td>Puerta de acceso</td><td>2</td></tr> <tr><td>22</td><td>Sala de comunicaciones</td><td>Armario 1</td><td>2</td></tr> <tr><td>23</td><td>Sala de comunicaciones</td><td>Armario 2</td><td>2</td></tr> <tr><td>24</td><td>Sala de comunicaciones</td><td>Técnicos</td><td>2</td></tr> <tr><td>25</td><td>Sala de comunicaciones</td><td>Servidores</td><td>2</td></tr> <tr><td>26</td><td>Sala de comunicaciones</td><td>Mantenimiento</td><td>2</td></tr> </tbody> </table>	Código	Zona	Descripcion	En préstamo	Total	0	Grupo electrógeno	Puerta de acceso	0	3	2	Grupo electrógeno	Armario principal	0	2	3	Grupo electrógeno	Salida a patio	0	2	5	Grupo electrógeno	Contadores	0	4	6	Sala de máquinas	Puerta de acceso	0	4	7	Sala de máquinas	Puerta 2	0	2	8	Sala de máquinas	Armario 1	0	1	9	Sala de máquinas	Armario 2	0	1	10	Sala de máquinas	Cuadros de fuerzas	0	3	11	UPS	Puertade acceso	0	2	12	UPS	Baterias	0	2	13	UPS	Armarios	0	2	14	UPS	Cuadros	0	2	15	Sala de condensadoras	Puerta de acceso	0	1	16	Sala de condensadoras	Armario 1	0	2	17	Sala de condensadoras	Armario 2	1	5	18	Sala de condensadoras	Mantenimiento	1	4	19	Sala de condensadoras	Cuadros	1	4	20	Sala de condensadoras	Cajones	1	4	21	Sala de comunicaciones	Puerta de acceso	0	4	22	Sala de comunicaciones	Armario 1	0	4	23	Sala de comunicaciones	Armario 2	0	4	24	Sala de comunicaciones	Técnicos	0	4	Código	Descripcion	En préstamo	Total	0	Grupo Electrogeno	1	2	1	Sala Maquinas	0	4	2	UPS	0	2	3	Sala Comunicaciones	1	1	Código	Zona	Descripcion	Total	0	Grupo electrógeno	Puerta de acceso	1	5	Grupo electrógeno	Contadores	2	6	Sala de máquinas	Puerta de acceso	2	7	Sala de máquinas	Puerta 2	1	8	Sala de máquinas	Armario 1	1	9	Sala de máquinas	Armario 2	1	10	Sala de máquinas	Cuadros de fuerzas	2	15	Sala de condensadoras	Puerta de acceso	1	16	Sala de condensadoras	Armario 1	1	17	Sala de condensadoras	Armario 2	3	18	Sala de condensadoras	Mantenimiento	2	19	Sala de condensadoras	Cuadros	2	20	Sala de condensadoras	Cajones	2	21	Sala de comunicaciones	Puerta de acceso	2	22	Sala de comunicaciones	Armario 1	2	23	Sala de comunicaciones	Armario 2	2	24	Sala de comunicaciones	Técnicos	2	25	Sala de comunicaciones	Servidores	2	26	Sala de comunicaciones	Mantenimiento	2
Código	Zona	Descripcion	En préstamo	Total																																																																																																																																																																																																																									
0	Grupo electrógeno	Puerta de acceso	0	3																																																																																																																																																																																																																									
2	Grupo electrógeno	Armario principal	0	2																																																																																																																																																																																																																									
3	Grupo electrógeno	Salida a patio	0	2																																																																																																																																																																																																																									
5	Grupo electrógeno	Contadores	0	4																																																																																																																																																																																																																									
6	Sala de máquinas	Puerta de acceso	0	4																																																																																																																																																																																																																									
7	Sala de máquinas	Puerta 2	0	2																																																																																																																																																																																																																									
8	Sala de máquinas	Armario 1	0	1																																																																																																																																																																																																																									
9	Sala de máquinas	Armario 2	0	1																																																																																																																																																																																																																									
10	Sala de máquinas	Cuadros de fuerzas	0	3																																																																																																																																																																																																																									
11	UPS	Puertade acceso	0	2																																																																																																																																																																																																																									
12	UPS	Baterias	0	2																																																																																																																																																																																																																									
13	UPS	Armarios	0	2																																																																																																																																																																																																																									
14	UPS	Cuadros	0	2																																																																																																																																																																																																																									
15	Sala de condensadoras	Puerta de acceso	0	1																																																																																																																																																																																																																									
16	Sala de condensadoras	Armario 1	0	2																																																																																																																																																																																																																									
17	Sala de condensadoras	Armario 2	1	5																																																																																																																																																																																																																									
18	Sala de condensadoras	Mantenimiento	1	4																																																																																																																																																																																																																									
19	Sala de condensadoras	Cuadros	1	4																																																																																																																																																																																																																									
20	Sala de condensadoras	Cajones	1	4																																																																																																																																																																																																																									
21	Sala de comunicaciones	Puerta de acceso	0	4																																																																																																																																																																																																																									
22	Sala de comunicaciones	Armario 1	0	4																																																																																																																																																																																																																									
23	Sala de comunicaciones	Armario 2	0	4																																																																																																																																																																																																																									
24	Sala de comunicaciones	Técnicos	0	4																																																																																																																																																																																																																									
Código	Descripcion	En préstamo	Total																																																																																																																																																																																																																										
0	Grupo Electrogeno	1	2																																																																																																																																																																																																																										
1	Sala Maquinas	0	4																																																																																																																																																																																																																										
2	UPS	0	2																																																																																																																																																																																																																										
3	Sala Comunicaciones	1	1																																																																																																																																																																																																																										
Código	Zona	Descripcion	Total																																																																																																																																																																																																																										
0	Grupo electrógeno	Puerta de acceso	1																																																																																																																																																																																																																										
5	Grupo electrógeno	Contadores	2																																																																																																																																																																																																																										
6	Sala de máquinas	Puerta de acceso	2																																																																																																																																																																																																																										
7	Sala de máquinas	Puerta 2	1																																																																																																																																																																																																																										
8	Sala de máquinas	Armario 1	1																																																																																																																																																																																																																										
9	Sala de máquinas	Armario 2	1																																																																																																																																																																																																																										
10	Sala de máquinas	Cuadros de fuerzas	2																																																																																																																																																																																																																										
15	Sala de condensadoras	Puerta de acceso	1																																																																																																																																																																																																																										
16	Sala de condensadoras	Armario 1	1																																																																																																																																																																																																																										
17	Sala de condensadoras	Armario 2	3																																																																																																																																																																																																																										
18	Sala de condensadoras	Mantenimiento	2																																																																																																																																																																																																																										
19	Sala de condensadoras	Cuadros	2																																																																																																																																																																																																																										
20	Sala de condensadoras	Cajones	2																																																																																																																																																																																																																										
21	Sala de comunicaciones	Puerta de acceso	2																																																																																																																																																																																																																										
22	Sala de comunicaciones	Armario 1	2																																																																																																																																																																																																																										
23	Sala de comunicaciones	Armario 2	2																																																																																																																																																																																																																										
24	Sala de comunicaciones	Técnicos	2																																																																																																																																																																																																																										
25	Sala de comunicaciones	Servidores	2																																																																																																																																																																																																																										
26	Sala de comunicaciones	Mantenimiento	2																																																																																																																																																																																																																										

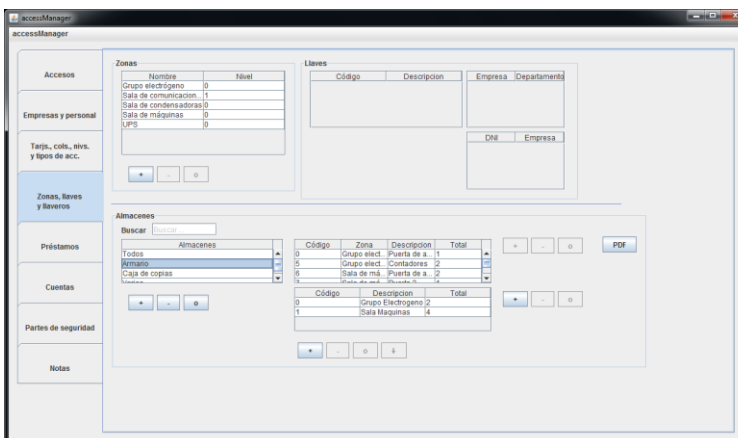


Código	Descripción	Total
0	Grupo Electrogeno	2
1	Sala Maquinas	4

La espera:



Salida observada



Evaluación/
conclusión

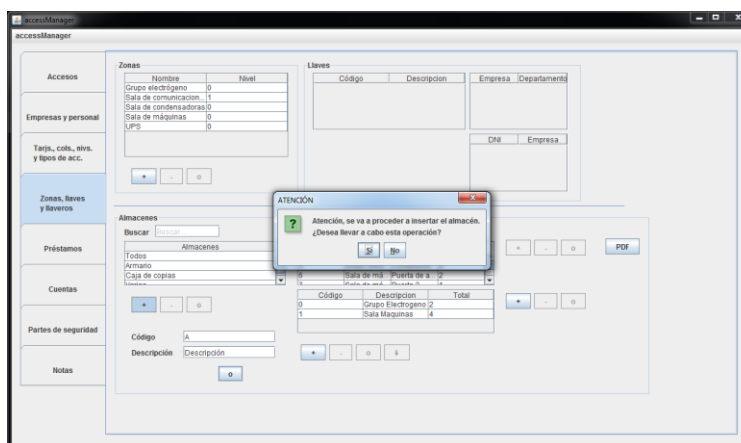
El caso de uso '87 - Visualizar almacén de llaves' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '87 - Visualizar almacén de llaves'

Caso de prueba	6.19 - Caso de uso 88: Añadir almacén de llaves
Objetivo	Probar si es posible dar de alta nuevos almacenes de datos a través de accessManager.
Actividades	Desplegar la aplicación y acudir a la pestaña de zonas. En ella intentaremos añadir almacenes erróneos y válidos, comprobando que solo se introduzcan los segundos.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior.

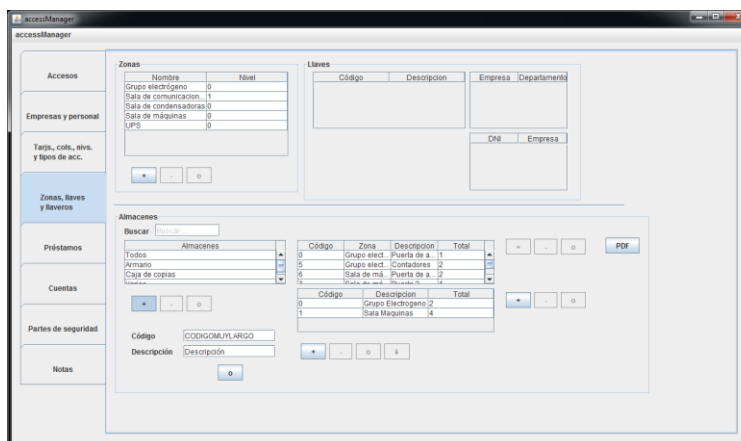
Discernimos las siguientes posibilidades:

- Erróneas:
 'A, Descripción'. (Almacén existente)

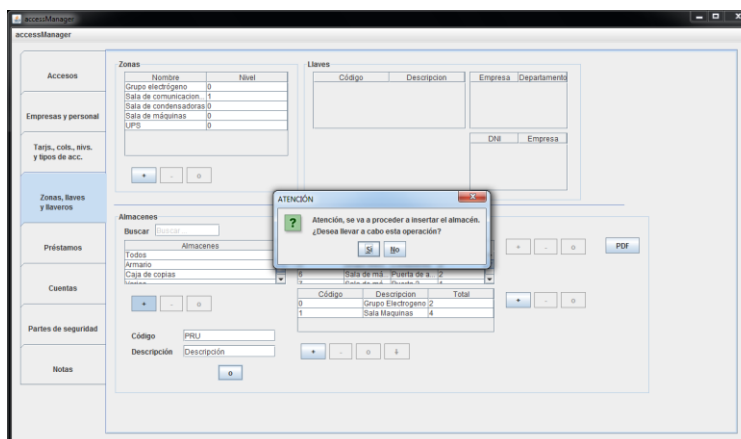


'CODIGOMUYLARGO, Descripción'. (Formato erróneo)

Datos de entrada



- Válida:
'PRU, Descripción'

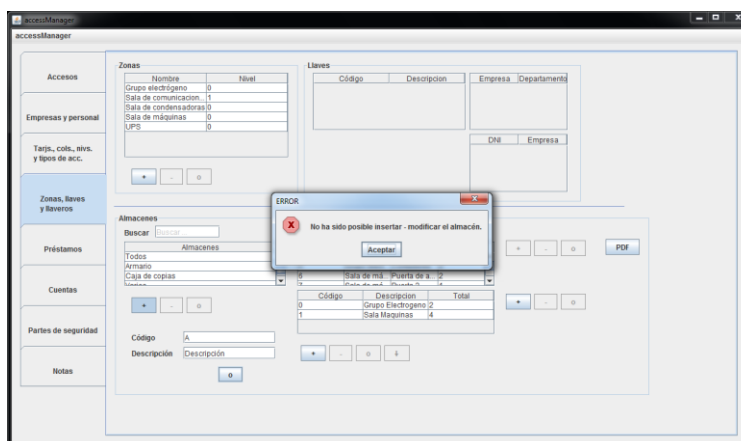


Salida esperada

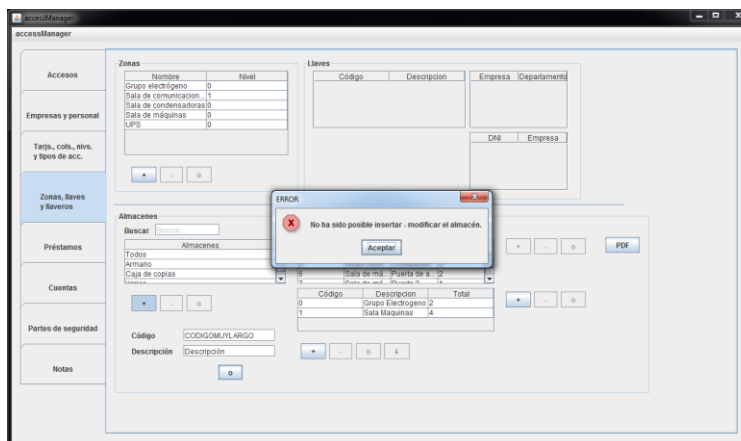
Los dos primeros intentos deberían desembocar en errores y el segundo en una inserción en la base de datos.

La esperada:

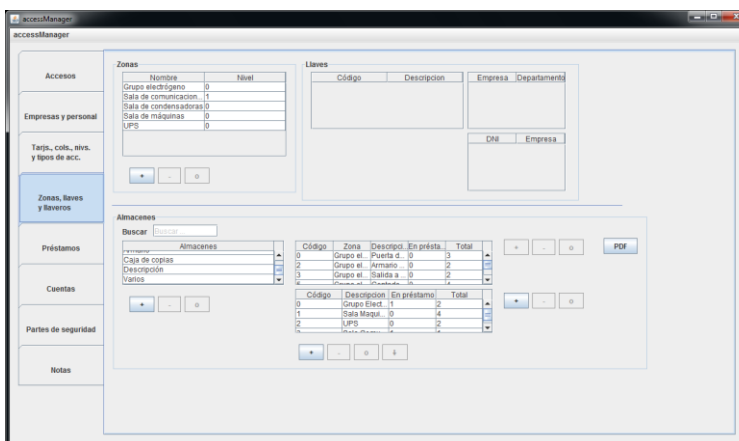
- En los casos erróneos:



Salida observada



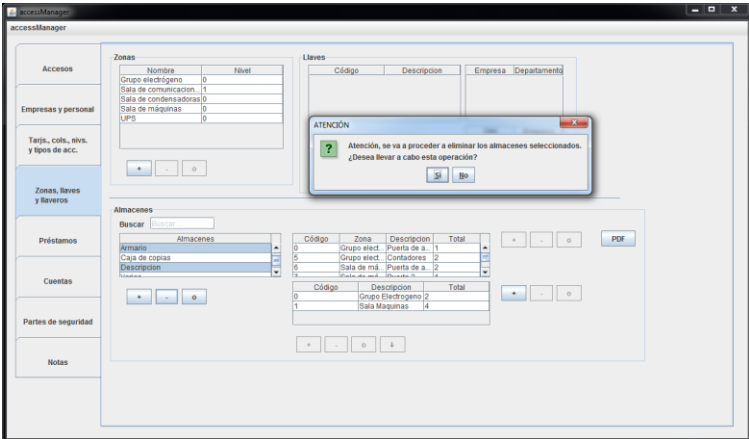
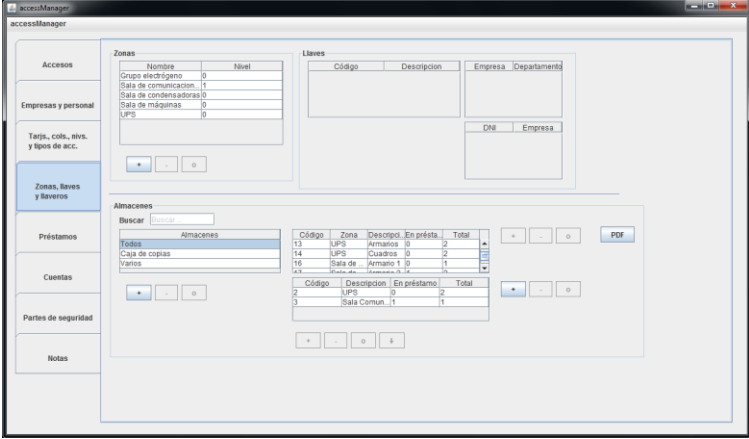
- En el válido:



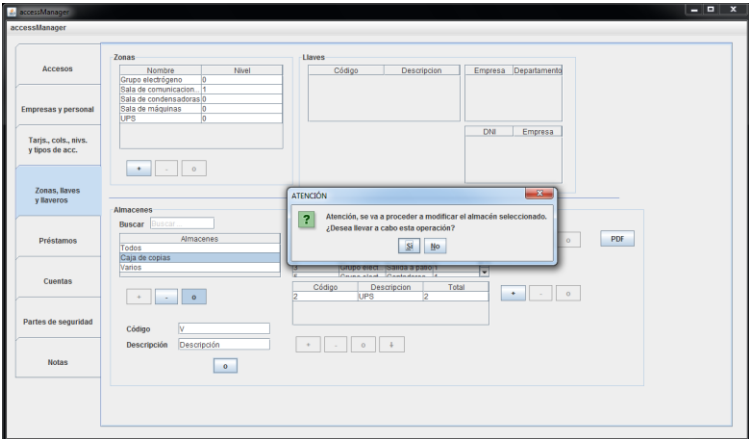
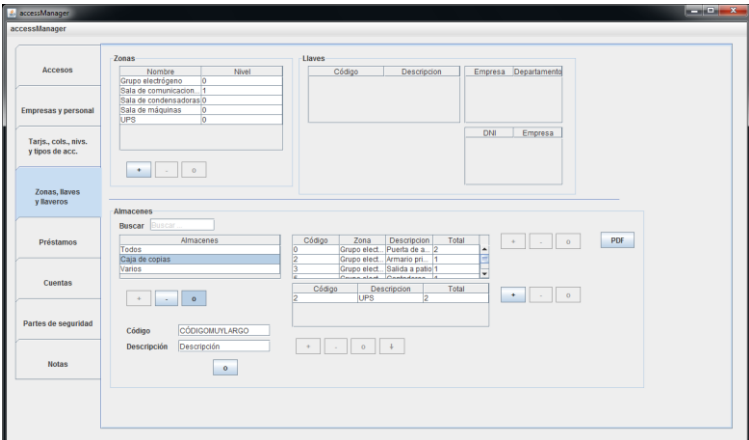
**Evaluación/
conclusión**

El caso de uso '88 - Añadir almacén de llaves' ha sido realizado satisfactoriamente.

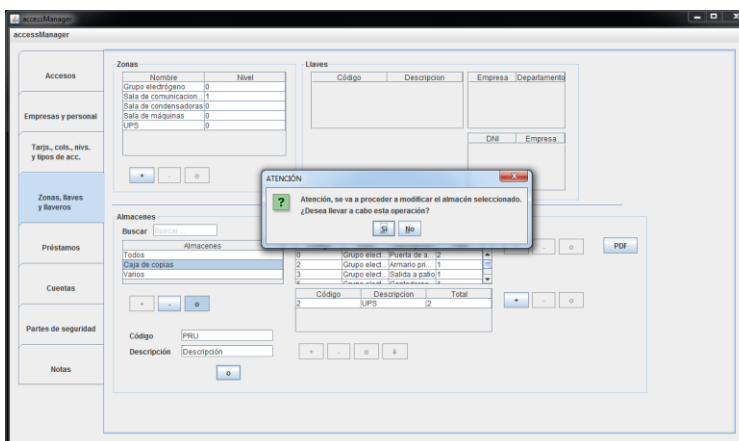
Caso de prueba para el caso de uso '88 - Añadir almacén de llaves'

Caso de prueba	6.20 - Caso de uso 89: Borrar almacén de llaves
Objetivo	Comprobar que no existen problemas a la hora de borrar almacenes.
Actividades	Loguear en accessManager y situarnos en la tab 'Zonas, llaves y llaveros', donde eliminaremos varios almacenes y apreciaremos si todo se desarrolla según lo previsto.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior.
Datos de entrada	<p>Borraremos armarios que dispongan de las únicas copias existentes de algunas llaves o llaveros, comprobando así que tanto almacenes como los elementos que ningún otro almacén contienen son borrados. En este caso, optaremos por eliminar el almacén que creamos con anterioridad ('Descripción') y el de nombre 'Armario':</p> 
Salida esperada	El primer almacén no causa consecuencias más que las de su propio borrado, pero al borrar 'Armario' deberían dejar de existir la llave 15 y los llaveros 0 y 1.
Salida observada	<p>La esperada:</p> 
Evaluación/conclusión	El caso de uso '89 - Borrar almacén de llaves' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '89 - Borrar almacén de llaves'

Caso de prueba	6.21 - Caso de uso 90: Modificar almacén de llaves
Objetivo	Verificar que podemos modificar los datos que determinan cada uno de los almacenes existentes en el sistema siempre y cuando los nuevos que proporcionemos estén dentro del rango de valores permitidos.
Actividades	Desplegar la aplicación y dirigirse a la pestaña de zonas, donde se seleccionarán los almacenes 'Todos' y 'Caja de copias' en los que se intentarán diversas modificaciones, comprobando que el sistema siempre responde como debe.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior.
Datos de entrada	El primer almacén será inaccesible, pero el segundo probaremos a modificarlo con: <ul style="list-style-type: none"> • 'V, Descripción'. (Almacén existente)
	 <ul style="list-style-type: none"> • 'CODIGOMUYLARGO, Descripción'. (Formato de datos erróneo) 

- 'PRU, Descripción'. (Registro válido)



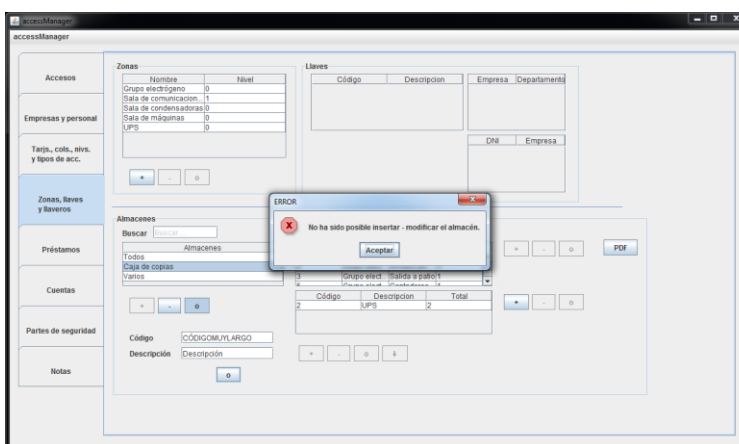
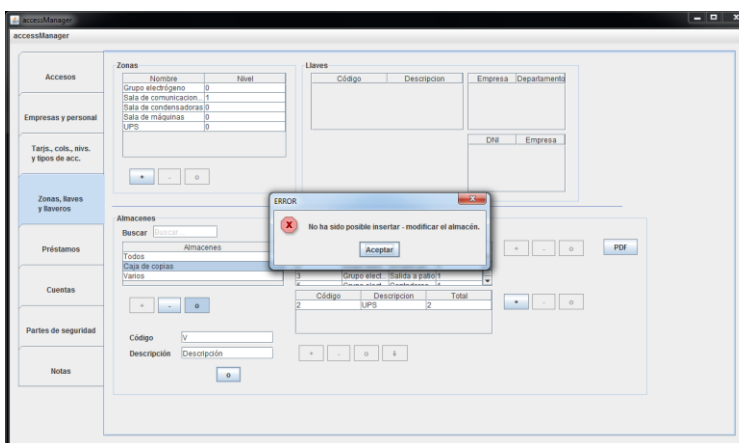
Si nos damos cuenta, son los mismos casos de prueba que para el caso de uso 88, pero varía que no realizamos una inserción, sino una actualización.

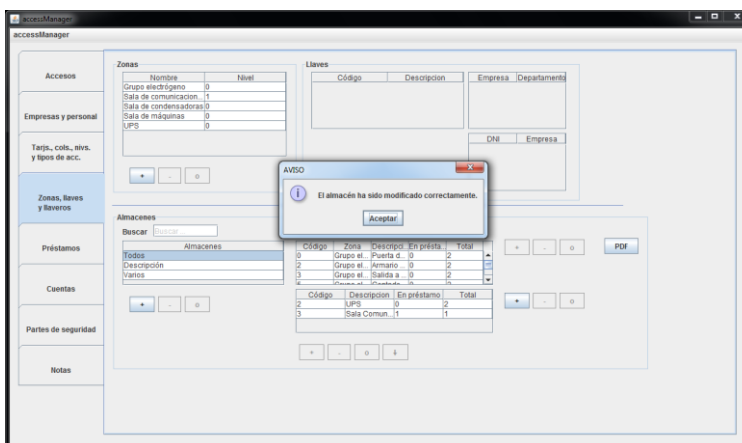
Salida esperada

Los dos primeros intentos deberían ser en vano, mientras que el segundo debería actualizar los datos del almacén haciendo que permanezcan todas las llaves y llaveros inalterados.

La esperada. Por orden tenemos:

Salida observada





El almacén 'Todos' es inaccesible, tal y como la imagen anterior indica (botón 'o' desactivado).

**Evaluación/
conclusión**

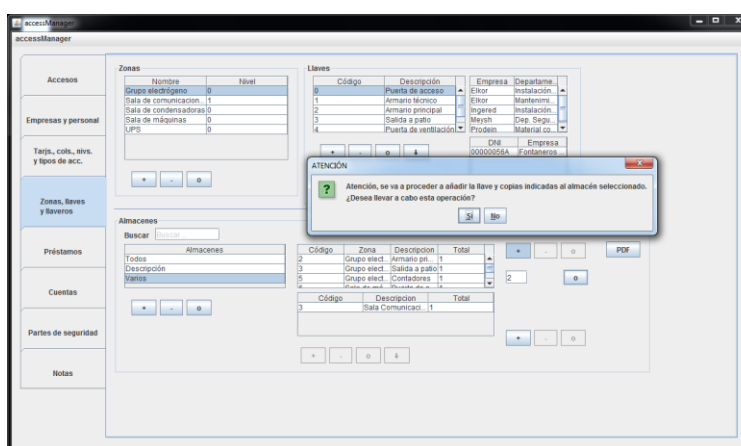
El caso de uso '90 - Modificar almacén de llaves' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '90 - Modificar almacén de llaves'

Caso de prueba	6.22 - Caso de uso 91: Añadir llave
Objetivo	Chequear que podemos añadir a cualquiera de los almacenes (salvo a 'Todos') llaves existentes en cualquiera de las zonas sin que ello desemboque en ningún error si es que actuamos correctamente.
Actividades	Acceder a la aplicación, seleccionar un almacén y probar a realizar varias inserciones de llaves, comprobando que siempre recibimos la salida esperada.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior.

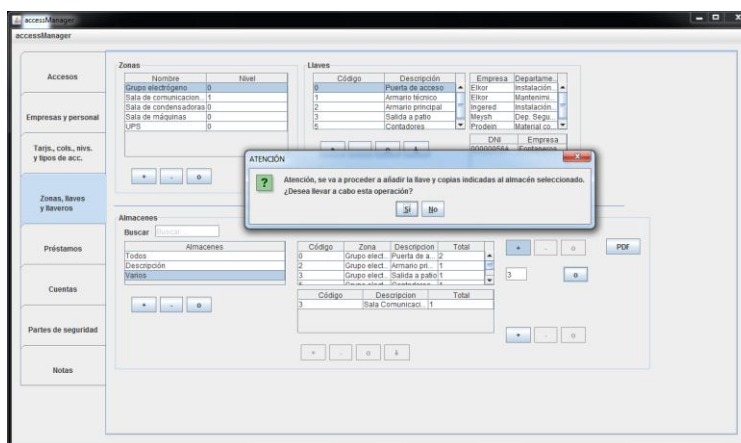
Seleccionaremos el almacén con nombre 'Varios' y probaremos a realizar la siguiente secuencia de inserciones:

- Almacén 'Varios', llave 0, 2 copias:

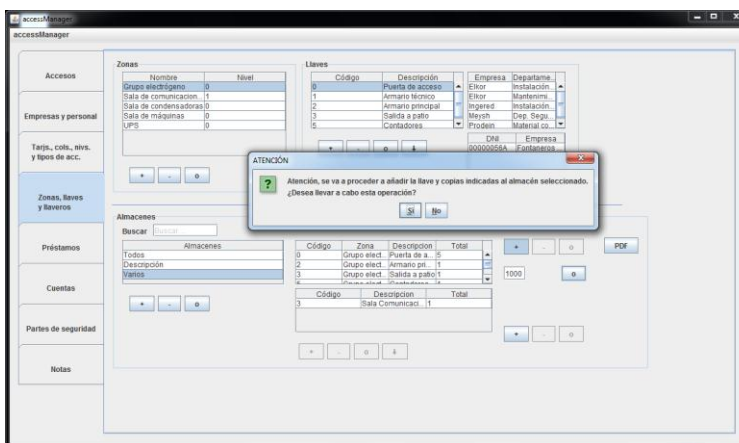


- Almacén 'Varios', llave 0, 3 copias:

Datos de entrada



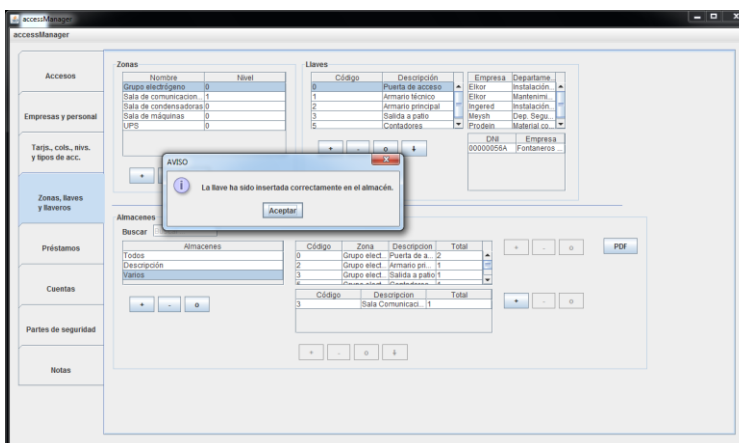
- Almacén 'Varios', llave 0, 100 copias:



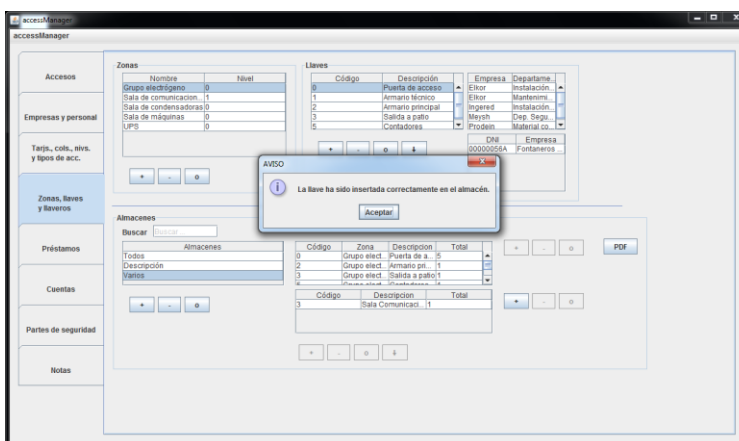
Salida esperada

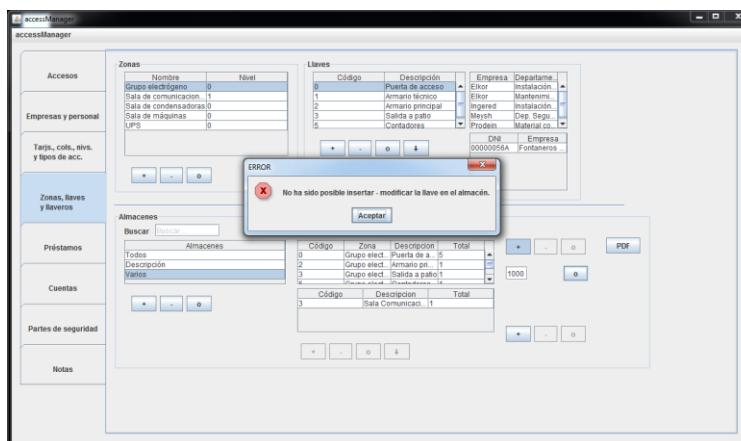
La primera deberá introducir la llave en el almacén, la segunda deberá incrementar el número de copias a 5 y la tercera producir un mensaje de error y no actualizar nada.

La esperada. Esta es la secuencia de resultados obtenidos:



Salida observada

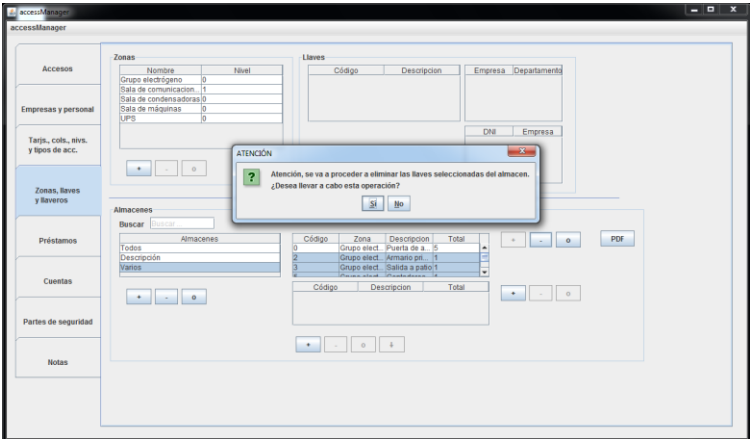
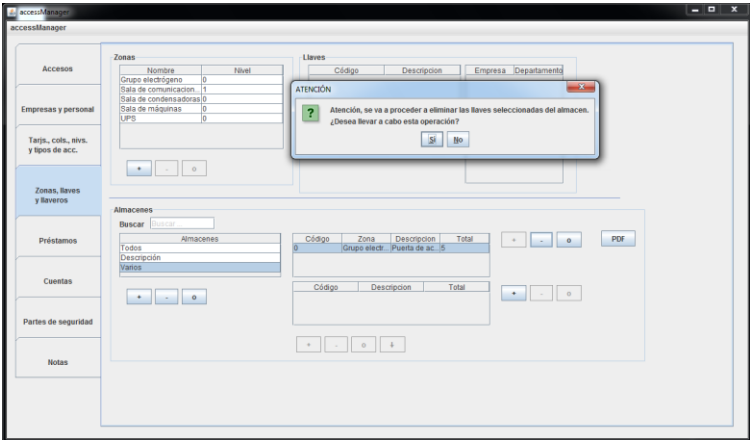




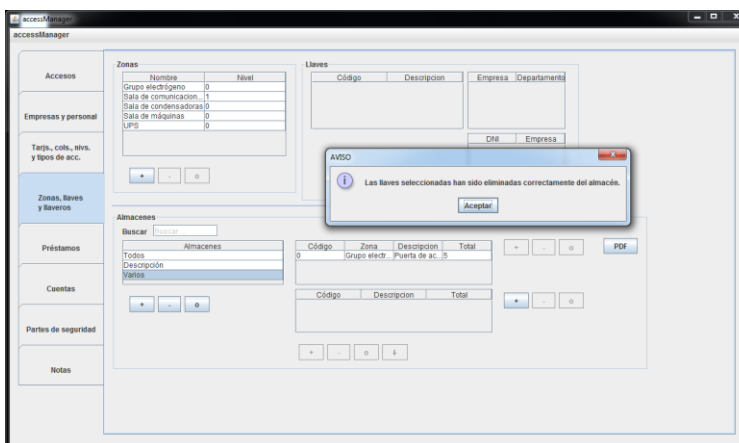
Evaluación/
conclusión

El caso de uso '91 - Añadir llave' ha sido realizado satisfactoriamente.

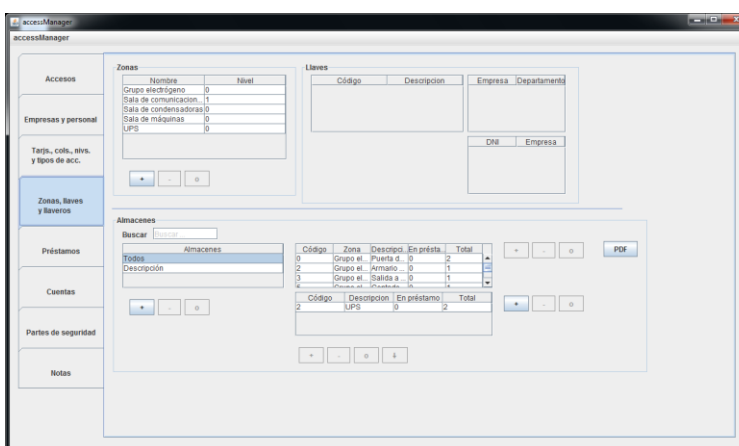
Caso de prueba para el caso de uso '91 - Añadir llave'

Caso de prueba	6.23 - Caso de uso 92: Borrar llave
Objetivo	Comprobar que si el usuario lo estima oportuno siempre podrá borrar cualquier llave que forme parte de cualquier almacén seleccionado.
Actividades	Desplegar la aplicación y clicar sobre la pestaña de zonas, donde seleccionaremos un almacén y eliminaremos algunas de sus llaves, comprobando que todos los acontecimientos se desarrollan según lo establecido.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior. • Utilidad phpMyAdmin.
Datos de entrada	<p>El almacén 'Varios' nos servirá. En un principio eliminamos desde phpMyAdmin el llavero existente y desde accessManager todas las llaves salvo la primera:</p>  <p>Tras quedarnos solo con la 0, la eliminamos:</p> 
Salida esperada	Tras el primer borrado sólo deberá existir en el almacén 'Varios' la llave número 0. A continuación, tras eliminar ésta, deberá desaparecer el almacén 'Varios', comprobando así que el disparador que creamos en la iteración C1 funciona correctamente.

La esperada:



Salida observada



Evaluación/
conclusión

El caso de uso '92 - Borrar llave' ha sido realizado satisfactoriamente.

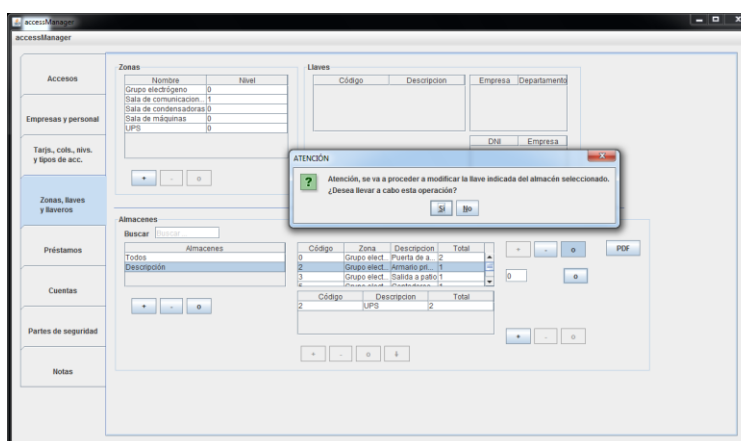
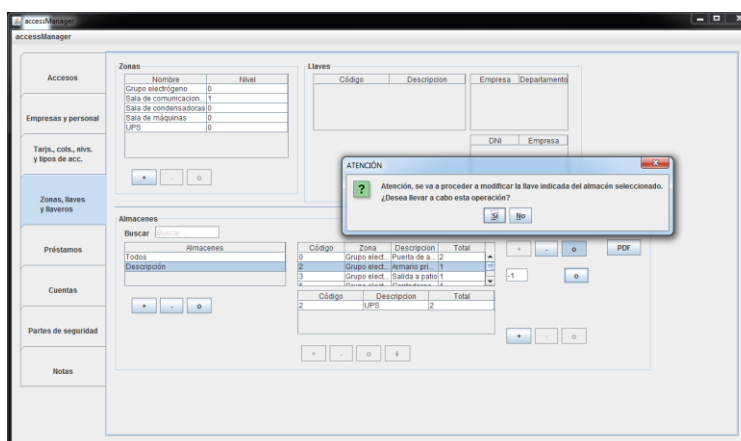
Caso de prueba para el caso de uso '92 - Borrar llave'

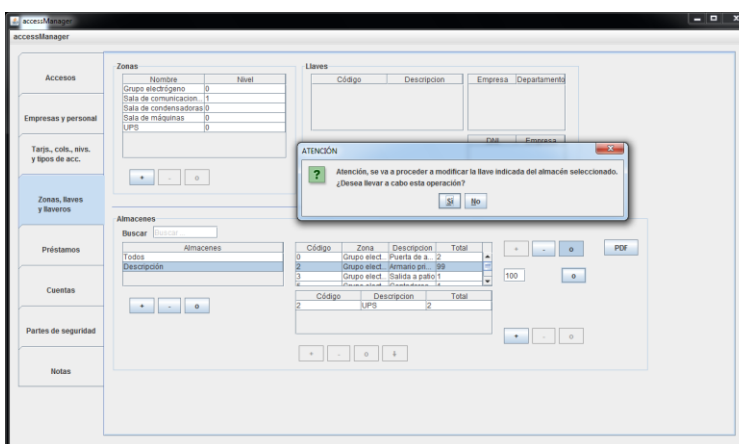
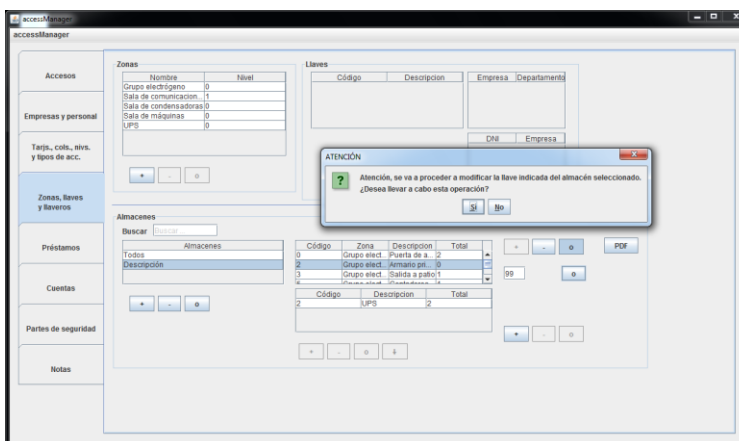


Caso de prueba	6.24 - Caso de uso 93: Modificar copias llave
Objetivo	Verificar que el usuario puede variar el número de copias de cualquier llave almacenada en cualquier almacén (excepto 'Todos') siempre y cuando el número de copias esté comprendido entre 0 y 99.
Actividades	Loguear en accessManager, acceder a la tab de zonas, seleccionar un almacén y una llave y probar a modificar su número de copias de diversas formas (permitidas y no permitidas) observando cómo se comporta el sistema.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos resultante del caso de prueba anterior.

No tenemos muchas opciones en cuanto a lo que almacenes se refiere (solo uno), aunque si las tuviésemos daría igual, todos ellos son igual de válidos para realizar esta prueba (excepto 'Todos', nuevamente). Seleccionamos pues el almacén 'Descripción' y cualquiera de sus llaves, en este ejemplo la número 2, asignándole los números de copias -1, 0, 99 y 100:

Datos de entrada



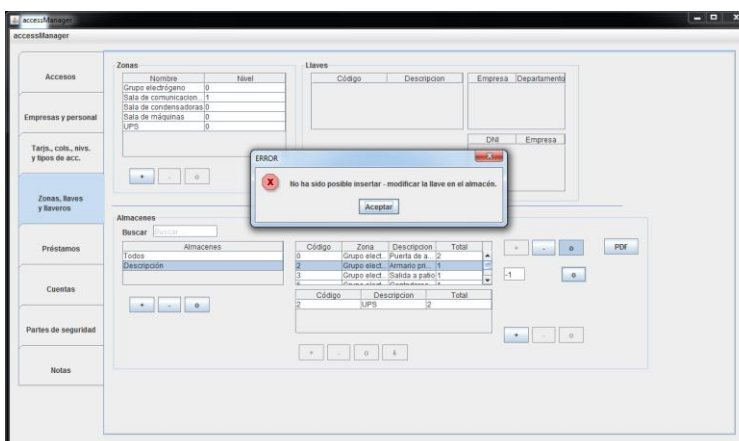


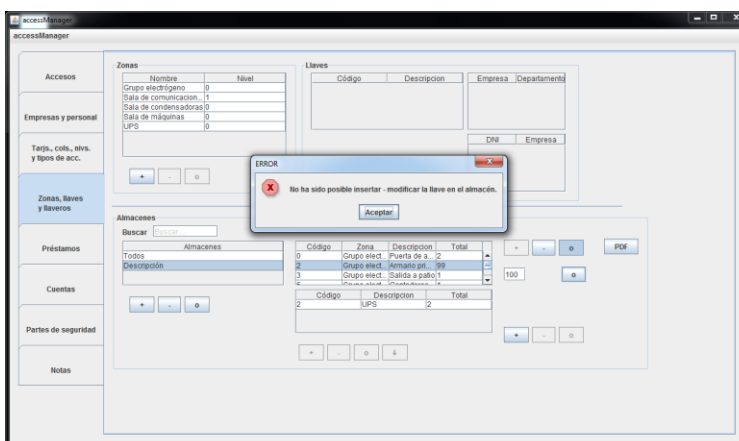
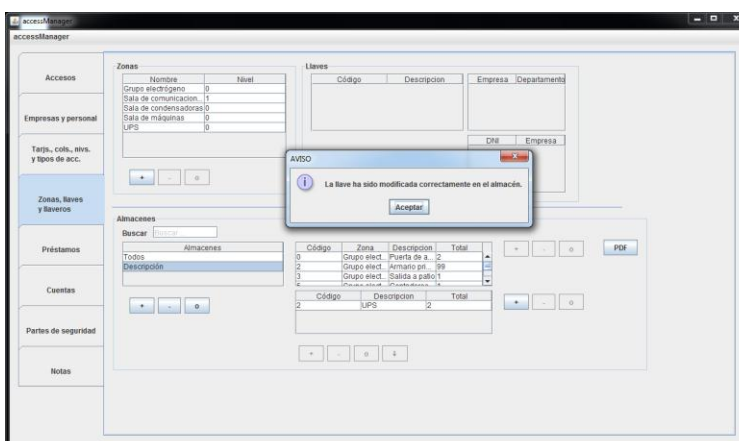
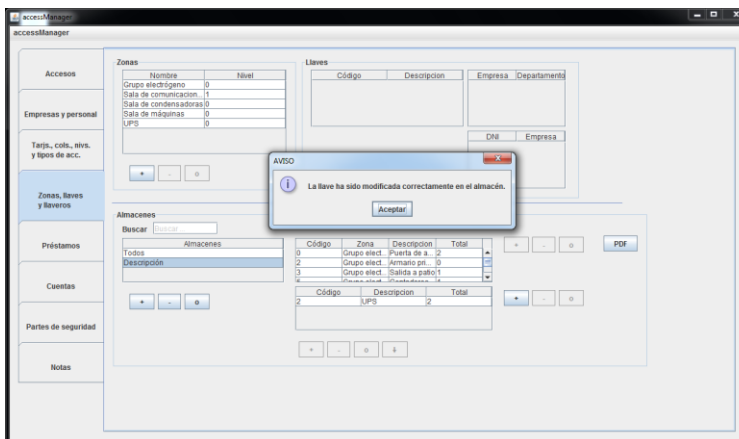
Salida esperada

El primer y último intento producirán mensajes de error, al no encontrarse el número de copias comprendido entre 0 y 99, pero el segundo y el tercero sí serán válidos.

Salida observada

La esperada. Estos son los resultados obtenidos en cada una de las pruebas:





**Evaluación/
conclusión**

El caso de uso '93 - Modificar copias llave' ha sido realizado satisfactoriamente.

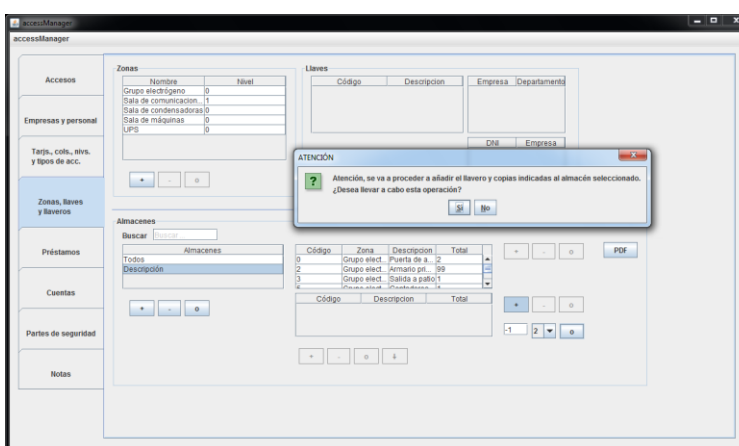
Caso de prueba para el caso de uso '93 - Modificar copias llave'



Caso de prueba	6.25 - Caso de uso 94: Añadir llavero
Objetivo	Probar que al igual que es posible añadir llaves a los diversos almacenes, también pueden ser posible añadidos llaveros.
Actividades	Acceder al sistema, situarnos en la pestaña 'Zona, llaves y llaveros' y probar a introducir varios llaveros, comprobando que las inserciones son llevadas a cabo sin provocar ninguna situación inesperada.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos resultante del caso de prueba anterior.

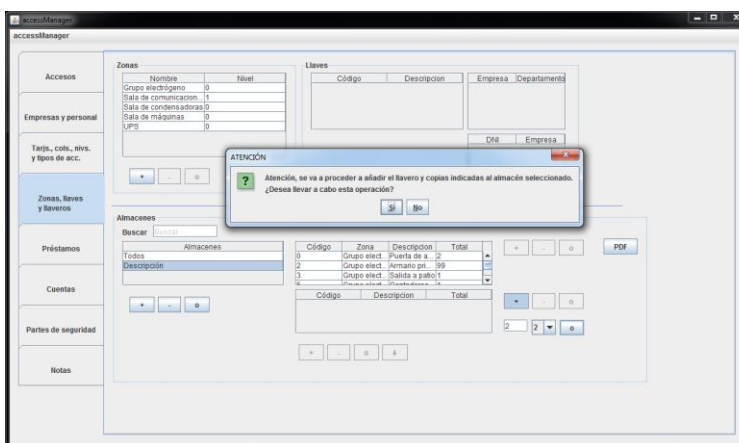
Disponemos del almacén 'Descripción' y los llaveros 0, 1 y 2, los cuales cualquiera podría servir para este caso de prueba. Cogemos el último, y probaremos a introducirlo de estas dos formas:

- Almacén 'Descripción', llavero 2, -1 copias:



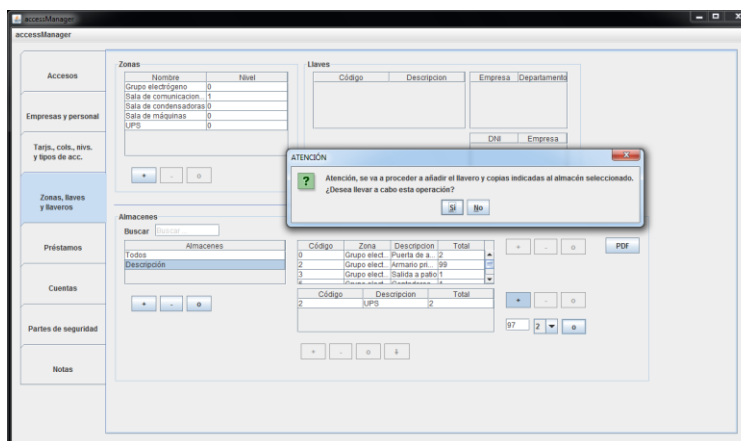
- Almacén 'Descripción', llavero 2, 2 copias:

Datos de entrada





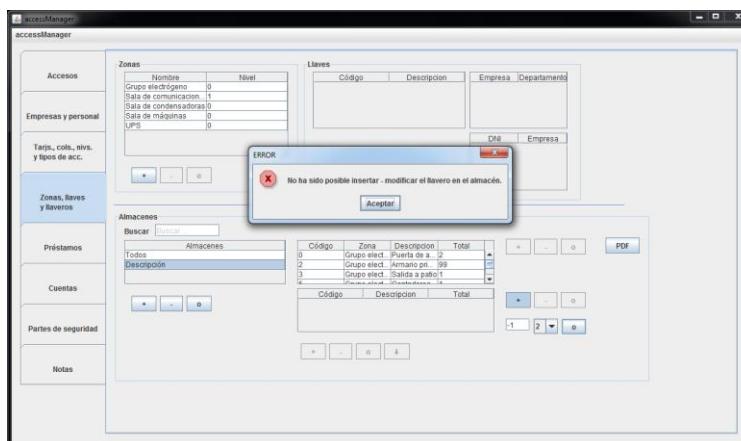
- Almacén 'Descripción', llavero 2, 97 copias:



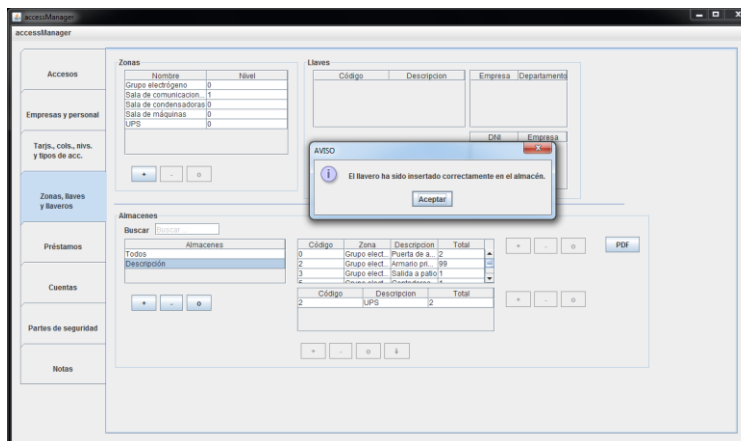
Salida esperada

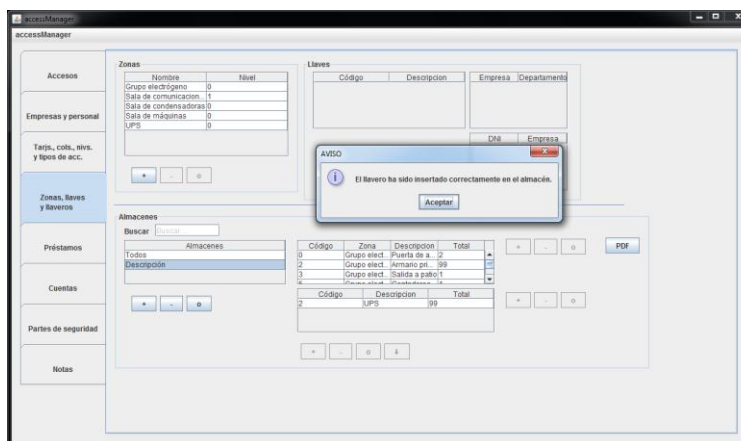
La primera no debe poder realizarse, ya el número de copias no está comprendido entre 0 y 99. No pasa lo mismo en el segundo y tercer caso, que no deberán presentar problemas y dotar al sistema con 2 y 99 copias del llavero respectivamente.

La esperada. En orden obtendríamos:



Salida observada



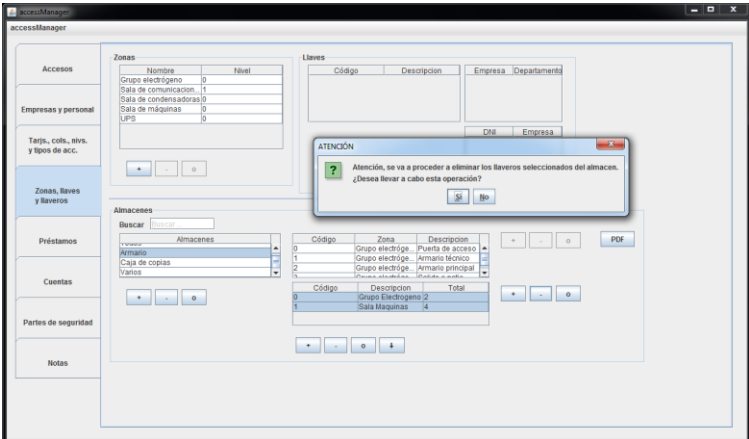
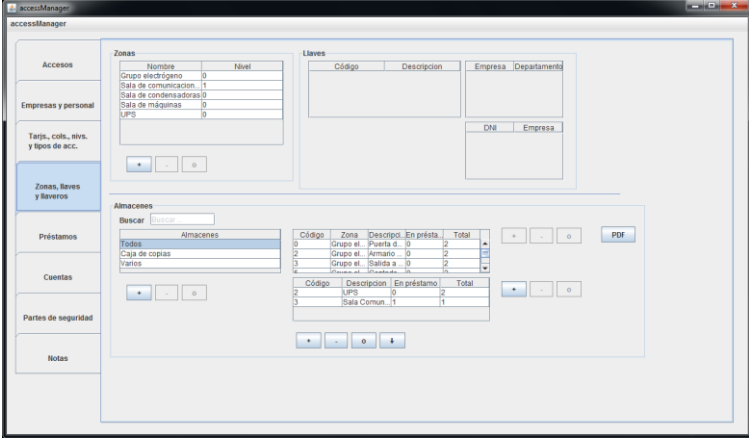


**Evaluación/
conclusión**

El caso de uso '94 - Añadir llavero' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '94 - Añadir llavero'



Caso de prueba	6.26 - Caso de uso 95: Borrar llavero
Objetivo	Comprobar que el usuario puede realizar el borrado de cualquier llavero existente en cualquiera de los almacenes.
Actividades	Desplegar la aplicación, seleccionar el almacén que se estime oportuno y eliminar de él varios llaveros, comprobando que ésto desencadena todas las acciones previstas.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. • Utilidad phpMyAdmin.
Datos de entrada	<p>Volvemos a introducir la población de prueba, ya que nos quedamos con muy pocos registros. Seleccionamos el almacén 'Armario', del cual eliminamos todas sus llaves desde la interfaz y procedemos a eliminar los dos llaveros que almacena:</p> 
Salida esperada	Ambos llaveros deberán desaparecer del almacén y base de datos, además del almacén 'Armario' ya que se quedará sin ningún elemento en su interior.
Salida observada	<p>La esperada:</p> 



Cod_Llavero	Número de llavero	F_Alta_LI	Fecha y hora de inserción del llavero	F_Baja_LI	Fecha y hora de baja del llavero	Descripcion	Descripción del llavero
	2	2010-01-01 00:00:01		NULL		UPS	
	3	2010-01-01 00:00:01		NULL		Sala Comunicaciones	

**Evaluación/
conclusión**

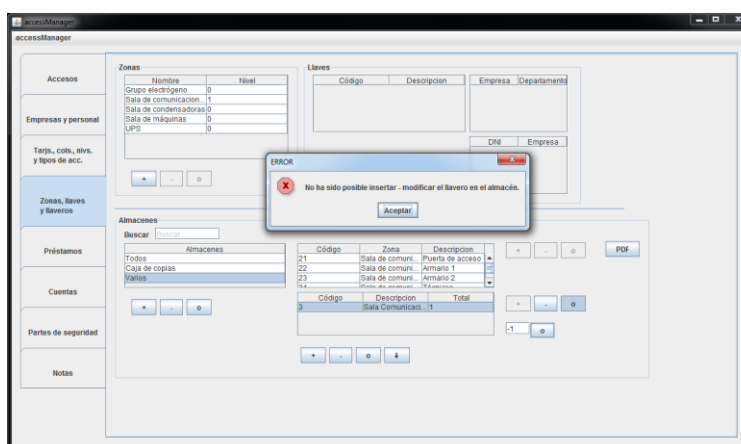
El caso de uso '95 - Borrar llavero' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '95 - Borrar llavero'

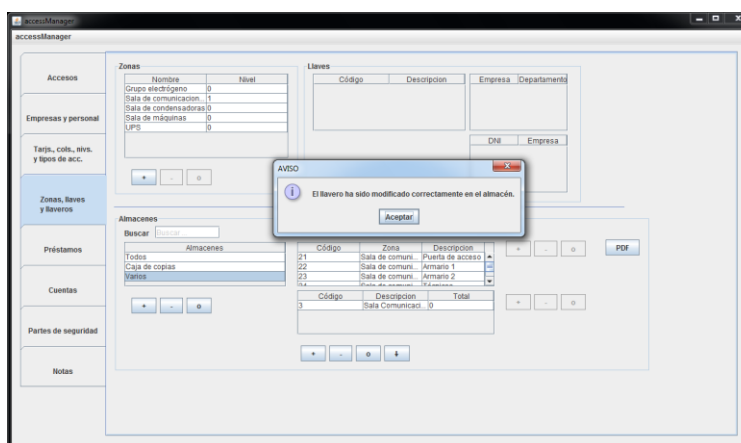


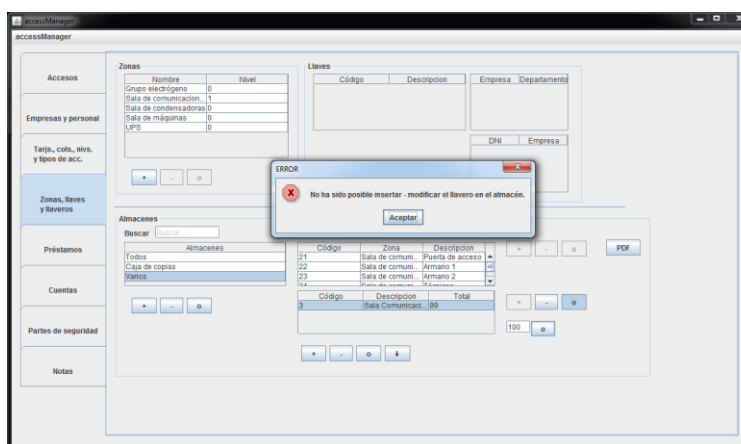
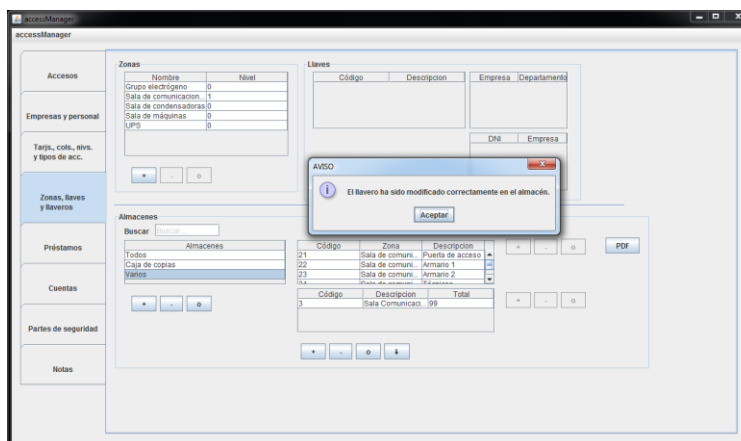
Caso de prueba	6.27 - Caso de uso 96: Modificar copias llavero
Objetivo	Probar que la modificación de copias de cualquier llavero se rige de la misma forma que la de llaves, siendo siempre correcta si el número de copias asignado se encuentra entre 0 y 99.
Actividades	Acceder a la tab 'Zonas, llaves y llaveros', seleccionar cualquiera de los almacenes existentes y modificar el número de copias de cualquiera de sus llaveros, de forma que sometamos a prueba los posibles puntos de fallo del proceso de actualización codificado en la anterior etapa.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos resultante del caso de prueba anterior.
Datos de entrada	Para la ocasión optaremos por el almacén 'Varios' y el llavero número 3, al que asignaremos las mismas copias que las utilizadas en el caso de prueba 6.24: -1, 0, 99 y 100.
Salida esperada	Al igual que nos sucedía con las llaves, un llavero no podrá estar registrado con -1 y 100 copias, pero sí con 0 y 99.

La esperada:



Salida observada





Evaluación/
conclusión

El caso de uso '96 - Modificar copias llavero' ha sido realizado satisfactoriamente.

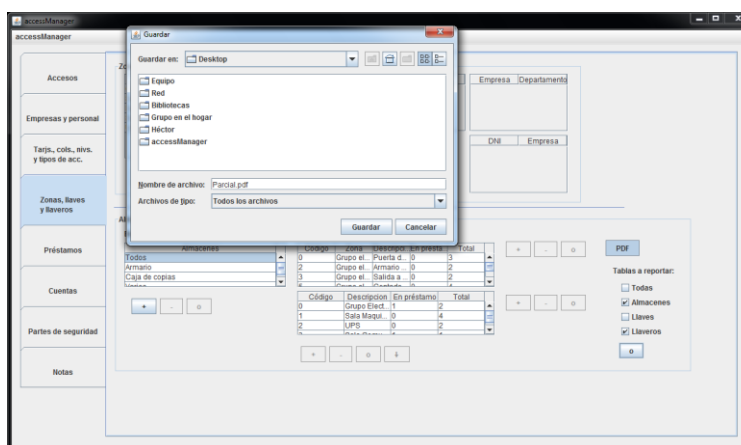
Caso de prueba para el caso de uso '96 - Modificar copias llavero'



Caso de prueba	6.28 - Caso de uso 76, 85 y 97: Listar llaves, llaveros y almacenes
Objetivo	Dejamos para el final estos tres casos de uso ya que están diseñados y se utilizarán conjuntamente. Debemos verificar que el usuario podrá obtener informes de todas las tablas pertenecientes a la sección de almacenes.
Actividades	Loguear en accessManager, definir las tablas de almacenes con los datos que estimemos oportunos y probar a generar reportes de éstas.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos con población de prueba.• Aplicación Adobe Reader.

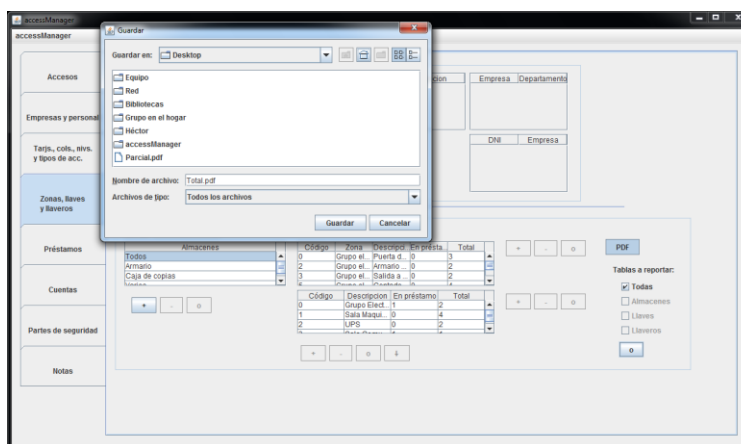
Vamos a probar con los resúmenes ofrecidos por el almacén 'Todos' y las combinaciones:

- Almacenes y llaveros:



- Todos:

Datos de entrada





Esperamos obtener dos archivos en formato PDF con el siguiente contenido:

- Parcial.pdf:

Almacenes	Código	Descripción	En préstamo	Total
Todos	0	Grupo Electrogeno	1	2
Armario	1	Sala Maquinas	0	4
Caja de copias	2	UPS	0	2
Varios	3	Sala Comunicaciones	1	1

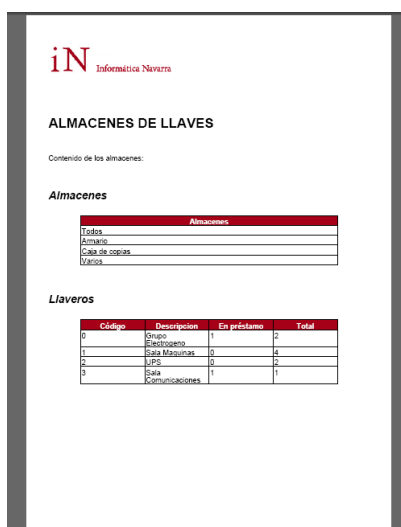
- Total.pdf: Con el contenido de las dos anteriores tablas unidas a ésta:

Código	Zona	Descripción	En préstamo	Total
0	Grupo electrógeno	Puerta de acceso	0	3
2	Grupo electrógeno	Armario principal	0	2
3	Grupo electrógeno	Salida a patio	0	2
5	Grupo electrógeno	Contadores	0	4
6	Sala de máquinas	Puerta de acceso	0	4
7	Sala de máquinas	Puerta 2	0	2
8	Sala de máquinas	Armario 1	0	1
9	Sala de máquinas	Armario 2	0	1
10	Sala de máquinas	Cuadros de fuerzas	0	3
11	UPS	Puertade acceso	0	2
12	UPS	Baterias	0	2
13	UPS	Armarios	0	2
14	UPS	Cuadros	0	2
15	Sala de condensadoras	Puerta de acceso	0	1
16	Sala de condensadoras	Armario 1	0	2
17	Sala de condensadoras	Armario 2	1	5
18	Sala de condensadoras	Mantenimiento	1	4
19	Sala de condensadoras	Cuadros	1	4
20	Sala de condensadoras	Cajones	1	4
21	Sala de comunicaciones	Puerta de acceso	0	4
22	Sala de comunicaciones	Armario 1	0	4
23	Sala de comunicaciones	Armario 2	0	4
24	Sala de comunicaciones	Técnicos	0	4

...


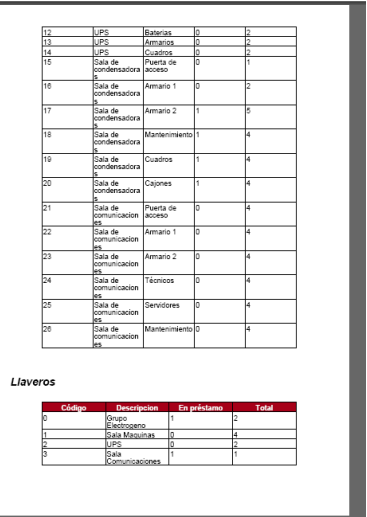
Salida esperada

La espera. Estos son los resultados:



Salida observada



	<div style="display: flex; justify-content: space-between;"> <div style="width: 48%;">  </div> <div style="width: 48%;">  </div> </div>
Evaluación/ conclusión	Los casos de uso '76, 85 y 97 - Listar llaves, llaveros y almacenes' han sido realizados satisfactoriamente.

Caso de prueba para los casos de uso '76, 85 y 97 - Listar llaves, llaveros y almacenes'

6.2. De integración

Toda prueba de integración debe tener como fin validar la eficacia funcional del sistema y subsistemas que lo componen. La arquitectura de nuestro sistema sigue estando formada por una base de datos y una interfaz gráfica y API que han visto incrementado su funcionalidad. A pesar de que el desarrollo del sistema continúa y éste cada vez es más completo, el número de componentes que lo forman permanece inmutable, con que a tenor de los resultados obtenidos en la sección anterior no vemos necesario realizar más pruebas.

6.3. De sistema

Al concluir la redacción de este artefacto llegaremos al final de una iteración en la que habremos incorporado todos los casos de uso que nos propusimos en la fase de planificación de la misma. Las pruebas realizadas han puesto de manifiesto que el funcionamiento es correcto y robusto, y creemos que éste es un documento adecuado y suficiente para poder estudiar y realizar las futuras actividades de mantenimiento de la pestaña 'Zonas, llaves y llaveros'. Llegados a este punto y con los resultados obtenidos, damos por concluida la fase de pruebas de forma satisfactoria.



7. Evaluación de la iteración

El tiempo asignado a realizar la iteración termina y con él todas las actividades planificadas. Todas han sido realizadas según la planificación establecida y la fase de pruebas nos muestra que todo el trabajo realizado en ellas ha sido correcto. Teniendo en cuenta la complejidad que entrañaba agrupar en las reducidas dimensiones de la pestaña tantos elementos y tantos casos de uso, creemos que hemos conseguido todos los objetivos que nos propusimos con un nivel aceptable de calidad. Falta esperar a ver que opiniones arroja el cliente en la reunión de esta tarde.

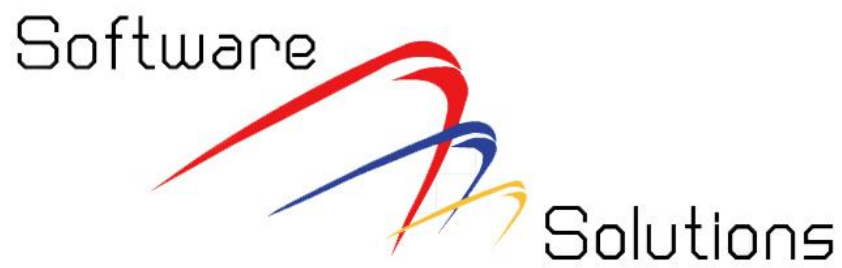
8. Despliegue

La cuarta reunión de despliegue llevada a cabo en la fase de construcción va encaminada a la muestra, prueba y evaluación conjunta con el cliente de la pestaña 'Zonas, llaves y llaveros' desarrollada en esta iteración. Tras mostrar y explicarle el uso de toda la pestaña parece satisfecho y nos hace saber que la funcionalidad que proporciona es más que suficiente para sus pretensiones. Las pruebas realizadas a continuación acaban por asegurarle que dispone de todo lo que desea y nos insta a continuar con el desarrollo del proyecto.

Pamplona, a 17 de marzo de 2010.

Héctor Napal Basterrica
Responsable del proyecto

Pedro Martínez Pérez
*Jefe de área de producción de
Informática Navarra S.A.*



accessManager
Desarrollo de un sistema de control de accesos a un centro

Iteración C7: Préstamos

Fecha de entrega: 19 de marzo de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Estructura del documento

2. Estado del proyecto

3. Análisis

- 3.1. Planificación de la iteración
- 3.2. Análisis de los casos de uso y escenarios

4. Diseño

- 4.1. Caso de uso 98: Visualizar préstamos
- 4.2. Caso de uso 99: Visualizar datos del prestatario
- 4.3. Caso de uso 100: Devolver préstamo
- 4.4. Caso de uso 101: Enviar recordatorio vía E-Mail
- 4.5. Caso de uso 102: Listar préstamos pendientes

5. Codificación e integración

6. Pruebas

- 6.1. De unidad
- 6.2. De integración
- 6.3. De sistema

7. Evaluación de la iteración

8. Despliegue



1. Introducción

1.1. Propósito

Este breve artefacto nace con la intención de recoger todo el trabajo que desarrollaremos a lo largo de los siguientes dos días, implementando la pestaña de 'Préstamos' que deriva de la de 'Accesos'.

1.2. Audiencia

Hasta la transmisión del producto software este será un documento de carácter interno, que será mostrado al cliente durante el despliegue de esta iteración y transferido en la Transición.

1.3. Estructura del documento

Tras estas líneas comenzamos estableciendo en qué punto del proceso de desarrollo nos encontramos. Una vez que sabemos qué pasos debemos seguir, qué casos de uso debemos desarrollar, realizamos la planificación de la iteración y los analizamos a través de sus descripciones de trazo fino en la siguiente sección, la 3. Si pasamos al siguiente apartado podremos observar cómo les hemos dado solución.

La quinta y sexta sección recogen todos los detalles de codificación e integración que estimamos oportunos, así como los casos de prueba diseñados y ejecutados para certificar que el software elaborado en esta iteración cumple con los requisitos de calidad establecidos.

En el siguiente apartado exponemos una evaluación de todo el trabajo realizado en la iteración y terminamos este artefacto aludiendo a todos los aspectos importantes extraídos de las actividades de despliegue.

2. Estado del proyecto

El proceso de desarrollo del software comienza a estar bastante avanzado y empezamos a disponer de un prototipo muy desarrollado. Una vez implementadas las cuatro primeras pestañas, que son las más importantes y complejas, seguimos nuestro camino desarrollando otras cuatro de menor complejidad. En esta iteración nos centramos en crear toda la funcionalidad necesaria para llevar una correcta gestión de los préstamos que quedan pendientes de devolución como consecuencia de los accesos que tienen lugar en el centro.



3. Análisis

3.1. Planificación de la iteración

Este es el *planning* que seguiremos hoy y mañana:

PLAN DE ITERACIÓN C7: PRÉSTAMOS			
ASPECTO	DESCRIPCIÓN		
Objetivos	Desarrollar toda la funcionalidad correspondiente a los préstamos realizados en los accesos y que están pendientes de devolución, analizando, diseñando, codificando y probando todos los casos de uso que la definen.		
Artefactos de entrada	<ul style="list-style-type: none">• <i>Especificación inicial de los casos de uso</i>• <i>Planificación temporal y costes detallados del proyecto</i>• Base de datos, API y aplicación desarrolladas en las iteraciones anteriores.		
Artefactos modificados o creados	<ul style="list-style-type: none">• Diagrama de clases de la API inicial.• Clases Java de la API inicial y de la aplicación.• El presente documento.		
Calendarización detallada	Análisis + Pruebas	18/03/10	<ul style="list-style-type: none">• Planificación de la iteración y gestión de riesgos.• Análisis de los casos de uso y escenarios.• Planificación de las pruebas.• Inicio de la documentación.
	Diseño + Pruebas	18/03/10	<ul style="list-style-type: none">• Diseño de opciones arquitectónicas.• Diseño de pruebas.
	Codificación + Pruebas	18/03/10	<ul style="list-style-type: none">• Codificación e integración de la solución.• Ejecución y evaluación de pruebas.• Evaluación del prototipo.• Preparación de la entrega ejecutable.
		19/03/10	
Despliegue	19/03/10	<ul style="list-style-type: none">• Muestra, prueba y evaluación conjunta con el cliente.• Concreción de los cambios necesarios.• Finalización de toda la documentación.	

Planificación de la iteración



Nombre de tarea	Duración	Comienzo	Fin	Gantt chart (0 to 0)		
Estudio de oportunidad (inception)	7 días	jue 07/01/10	vie 15/01/10	[Bar chart showing task duration]		
Elaboración	16 días	lun 18/01/10	lun 08/02/10	[Bar chart showing task duration]		
Construcción	41 días	mar 09/02/10	mar 06/04/10	[Bar chart showing task duration]		
Iteración C1: La base de datos	4 días	mar 09/02/10	vie 12/02/10	[Bar chart showing task duration]		
Iteración C2: API para la base de datos	5 días	lun 15/02/10	vie 19/02/10	[Bar chart showing task duration]		
Iteración C3: Accesos	5 días	lun 22/02/10	vie 26/02/10	[Bar chart showing task duration]		
Iteración C4: Empresas y personal	5 días	lun 01/03/10	vie 05/03/10	[Bar chart showing task duration]		
Iteración C5: Tarjetas, tipos de acceso y cargos	3 días	lun 08/03/10	mié 10/03/10	[Bar chart showing task duration]		
Iteración C6: Zonas, llaves y llaveros	5 días	jue 11/03/10	mié 17/03/10	[Bar chart showing task duration]		
Iteración C7: Préstamos	2 días	jue 18/03/10	vie 19/03/10	[Bar chart showing task duration]		
Análisis	0,25 días	jue 18/03/10	jue 18/03/10	[Gantt chart showing task flow]		
Diseño	0,25 días	jue 18/03/10	jue 18/03/10	[Gantt chart showing task flow]		
Implementación	1 día	jue 18/03/10	vie 19/03/10	[Gantt chart showing task flow]		
Pruebas	2 días	jue 18/03/10	vie 19/03/10	[Gantt chart showing task flow]		
Despliegue	0,5 días	vie 19/03/10	vie 19/03/10	[Gantt chart showing task flow]		
Iteración C8: Cuentas	2 días	lun 22/03/10	mar 23/03/10	[Bar chart showing task duration]		
Iteración C9: Partes de seguridad	3 días	mié 24/03/10	vie 26/03/10	[Bar chart showing task duration]		
Iteración C10: Notas	3 días	lun 29/03/10	mié 31/03/10	[Bar chart showing task duration]		
Iteración C11: Base de datos	2 días	jue 01/04/10	vie 02/04/10	[Bar chart showing task duration]		
Iteración C12: Diseño, personalización y configuración	2 días	lun 05/04/10	mar 06/04/10	[Bar chart showing task duration]		
Capacidad operacional	0 días	mar 06/04/10	mar 06/04/10	[Bar chart showing task duration]		
Transición	8 días	mié 07/04/10	vie 16/04/10	[Bar chart showing task duration]		

Recursos	Temporales		2 días
	Humanos		1 persona
	Software	Sistema operativo	Microsoft Windows 7
		Ofimática	Microsoft Word 2010 Adobe Acrobat Professional 8
		Diseño	Visual Paradigm 4.2
Codificación		NetBeans 6.8	
Hardware		1 PC	

Casos de uso y escenarios implementados	98 - Visualizar préstamos 99 - Visualizar datos del prestatario 100 - Devolver préstamo 101 - Enviar recordatorio vía E-Mail 102 - Listar préstamos pendientes
--	--

Planificación de la iteración



3.2. Análisis de los casos de uso y escenarios

Iniciemos la fase de análisis. En primer lugar, tras haber seleccionado qué casos de uso implementaremos, vamos a profundizar en ellos, en qué flujo de acciones deberán conllevar, sus precondiciones y los resultados que proporcionarán:

Caso de uso 98: Visualizar préstamos	
Descripción	Todos los accesos que se lleven a cabo en el centro conllevarán el préstamo de tarjetas y/o llaves y llaveros. Es mucho el material que cambia de manos, con lo que parece muy razonable crear un apartado dentro de la aplicación destinado a llevar su control.
Actores principal	Administrador.
Pre-condiciones	El usuario desea conocer qué préstamos están pendientes de devolución.
Flujos	Básico <ol style="list-style-type: none"> 1. El administrador accede a la pestaña de préstamos. 2. El sistema se encarga de realiza la búsqueda y mostrarle qué material está pendiente de devolución.
	Alternativos (*)
Post-condiciones	El usuario visualiza todos los préstamos que en estos momentos están pendientes de ser devueltos.

Caso de uso 98: Visualizar préstamos

Caso de uso 99: Visualizar datos del prestatario	
Descripción	Una vez obtenidos los datos de todos los préstamos que están pendientes de devolución, puede ser necesario conocer los principales datos del prestatario de cara a poder pedirle que devuelva todo el material lo antes posible.
Actores principal	Administrador.
Pre-condiciones	El administrador se encuentra logueado en el sistema y por las razones que sean necesita conocer los datos de la persona que tenga en préstamo algún tipo de material (tarjeta y/o llaves y/o llaveros).
Flujos	Básico <ol style="list-style-type: none"> 1. El usuario pulsa el préstamo de cuyo prestatario necesita conocer sus datos personales. 2. El sistema accede a la base de datos y le proporciona al usuario los datos que busca, permitiendo el envío de un recordatorio vía E-Mail al prestatario en caso de que se disponga de alguna dirección de correo electrónico suya.
	Alternativos (*)
Post-condiciones	El administrador consigue lo que buscaba: visualizar la información del prestatario y disponer de la posibilidad de enviarle un correo electrónico si es que se dispone de alguna dirección.

Caso de uso 99: Visualizar datos del prestatario



Caso de uso 100: Devolver préstamo					
Descripción	La devolución de préstamos deberá irse realizando conforme los trabajadores entreguen de vuelta el material que en su día tomaron prestado. Como el vigilante solo podrá editar los accesos que él vaya dando y se encuentren sin salida y el administrador no tendrá privilegios de modificación en la pestaña de accesos, es necesario que se pueda registrar la devolución de préstamos desde esta pestaña.				
Actores principal	Administrador.				
Pre-condiciones	El usuario recibe de un trabajador algún préstamo pendiente y se dispone a registrar la devolución.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> <ol style="list-style-type: none"> 1. El usuario accede a la pestaña de préstamos y selecciona el que acaba de ser entregado, indicando dicho evento al sistema. 2. Éste le pide aprobación para marcar como devuelto todo aquel material que esté seleccionado en las tablas de préstamos. 3. El usuario le indica que todo es correcto y el sistema puede proceder. (*) </td> </tr> <tr> <td>Alternativos (*)</td> <td> <p>El usuario le expresa al sistema que no está satisfecho con las devoluciones que se van a realizar:</p> <ul style="list-style-type: none"> • El sistema no realiza ninguna acción y da por concluido el caso de uso. </td> </tr> </table>	Básico	<ol style="list-style-type: none"> 1. El usuario accede a la pestaña de préstamos y selecciona el que acaba de ser entregado, indicando dicho evento al sistema. 2. Éste le pide aprobación para marcar como devuelto todo aquel material que esté seleccionado en las tablas de préstamos. 3. El usuario le indica que todo es correcto y el sistema puede proceder. (*) 	Alternativos (*)	<p>El usuario le expresa al sistema que no está satisfecho con las devoluciones que se van a realizar:</p> <ul style="list-style-type: none"> • El sistema no realiza ninguna acción y da por concluido el caso de uso.
	Básico	<ol style="list-style-type: none"> 1. El usuario accede a la pestaña de préstamos y selecciona el que acaba de ser entregado, indicando dicho evento al sistema. 2. Éste le pide aprobación para marcar como devuelto todo aquel material que esté seleccionado en las tablas de préstamos. 3. El usuario le indica que todo es correcto y el sistema puede proceder. (*) 			
Alternativos (*)	<p>El usuario le expresa al sistema que no está satisfecho con las devoluciones que se van a realizar:</p> <ul style="list-style-type: none"> • El sistema no realiza ninguna acción y da por concluido el caso de uso. 				
Post-condiciones	La devolución de material efectuada consta en el sistema.				

Caso de uso 100: Devolver préstamo

Caso de uso 101: Enviar recordatorio vía E-Mail					
Descripción	En el caso de uso 99 comentamos que deberá existir una manera de poder enviar un correo electrónico al prestatario en caso de que se disponga de alguna dirección electrónica suya almacenada.				
Actores principal	Administrador.				
Pre-condiciones	Se ha completado el caso de uso 99, el sistema dispone de alguna dirección de correo electrónico almacenada del prestatario y el usuario estima oportuno enviarle un mensaje recordándole los préstamos que tiene pendientes.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> <ol style="list-style-type: none"> 1. El administrador clic en el botón de 'Enviar recordatorio'. 2. El sistema, tras recibir la aprobación por parte del administrador, elabora un E-Mail con los datos del prestatario y los préstamos que tiene pendientes y lo envía a todas las direcciones de correo electrónico almacenadas de éste. </td> </tr> <tr> <td>Alternativos (*)</td> <td>Ninguno.</td> </tr> </table>	Básico	<ol style="list-style-type: none"> 1. El administrador clic en el botón de 'Enviar recordatorio'. 2. El sistema, tras recibir la aprobación por parte del administrador, elabora un E-Mail con los datos del prestatario y los préstamos que tiene pendientes y lo envía a todas las direcciones de correo electrónico almacenadas de éste. 	Alternativos (*)	Ninguno.
	Básico	<ol style="list-style-type: none"> 1. El administrador clic en el botón de 'Enviar recordatorio'. 2. El sistema, tras recibir la aprobación por parte del administrador, elabora un E-Mail con los datos del prestatario y los préstamos que tiene pendientes y lo envía a todas las direcciones de correo electrónico almacenadas de éste. 			
Alternativos (*)	Ninguno.				
Post-condiciones	El usuario ha enviado un correo electrónico al prestatario que deseaba indicándole qué préstamos tiene pendientes y pidiéndole que los devuelva lo antes posible.				

Caso de uso 101: Enviar recordatorio vía E-Mail

Caso de uso 102: Listar préstamos pendientes	
Descripción	El cliente nos pide que le ofrezcamos una forma de poder listar parte o la totalidad de los préstamos que están pendientes de devolución, de modo que pueda consultarlos sin necesidad de accessManager.
Actores principal	Administrador.
Pre-condiciones	El administrador quiere obtener un listado con parte de los préstamos que están pendientes de devolución.
Flujos	Básico <ol style="list-style-type: none"> 1. El usuario accede al sistema y se sitúa en la pestaña 'Préstamos'. Selecciona de qué préstamos desea que esté formado el listado y pide al sistema que lo genere. 2. Éste abre un diálogo en el que el usuario especificará dónde almacenar el listado. (*) 3. Tras esto, el sistema genera el listado con los datos requeridos y lo deposita en la ruta introducida por el usuario.
	Alternativos (*) <p>El usuario cancela el diálogo de selección de ruta:</p> <ul style="list-style-type: none"> • El caso de uso finaliza.
Post-condiciones	El administrador dispone del listado solicitado.

Caso de uso 102: Listar préstamos pendientes.

4. Diseño

4.1. Caso de uso 98: Visualizar préstamos

Comenzamos el diseño de una nueva pestaña. Tenemos tres tipos de elementos que pueden ser prestados (tarjetas, llaves y llaveros), y el disponer de una pestaña destinada única y exclusivamente al tratamiento de préstamos nos permite poder dedicar una tabla para cada elemento:



Interfaz gráfica propuesta



Estos componentes gráficos son suficientes para poder mostrar todos los préstamos. ¿Cómo los obtendremos? De la siguiente forma:

1. Crearemos un objeto de tipo Tarjeta, otro de tipo Llave y otro de tipo Llavero que realizarán las búsquedas de tarjetas, llaves y llaveros prestados respectivamente.
2. Llevaremos los resultados obtenidos a las tablas.

Todo son operaciones de búsqueda realizadas por un método que implementaremos en cada una de las tres clases:

MÉTODO	FUNCIONALIDAD
<i>obtenerPrestamos</i>	Método que realiza la búsqueda de todos aquellos elementos pertenecientes a la clase que lo invoca y que se encuentran en préstamo.

Método a implementar y a añadir a la API

Además se deberá añadir el método *obtenerColoresTarjsPrest*, que es una derivación de *obtenerColoresTarjs* pero devolviendo solo los colores de las tarjetas en préstamo.

Este proceso pasa de sencillo a trivial si lo visualizamos a través de sus correspondientes diagramas:

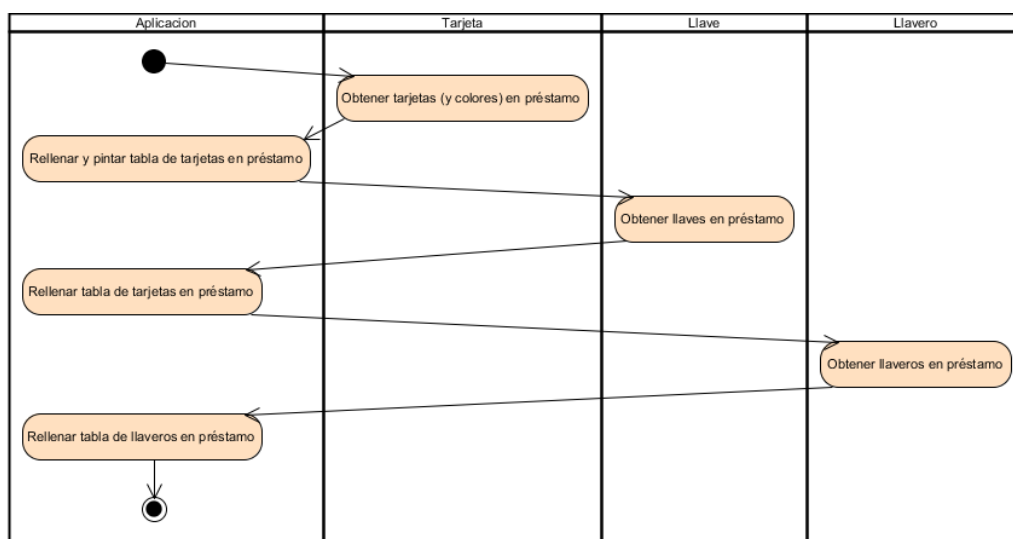
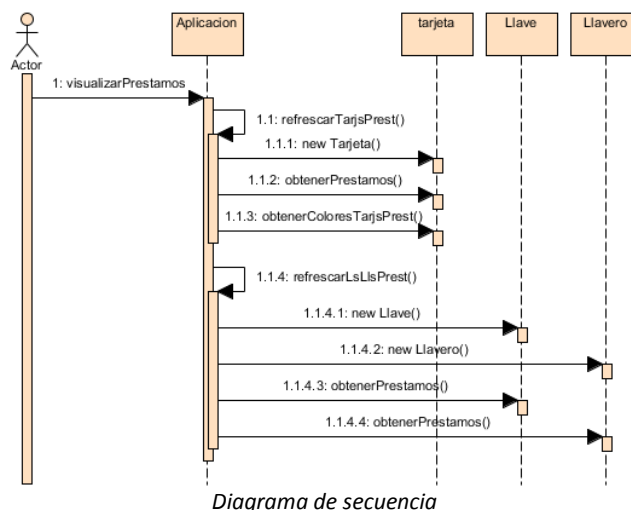


Diagrama de actividades



4.2. Caso de uso 99: Visualizar datos del prestatario

Para poder mostrar los datos de la persona que posee un préstamo es necesario crear un nuevo componente gráfico destinado a tal efecto. Al igual que nos pasó con anteriores formularios, optamos por crear un JPanel con JTextFields que rellenar y que colocaremos en la parte inferior de las tablas de préstamos:



Interfaz gráfica propuesta



Hace ya un tiempo creamos un constructor que nos permitía recuperar una persona a partir de su DNI. En esta ocasión también nos servirá de gran ayuda para llevar a buen puerto este proceso:

- a. Si el elemento seleccionado es una tarjeta:
 - Obtendremos el DNI del prestatario a partir de la Tarjeta recuperada de la base de datos.
 - b. Si el elemento seleccionado es una llave o un llavero:
 - Obtendremos el DNI del destinatario a partir del Acceso recuperado de la base de datos.
1. Recuperamos la persona a través del constructor *Personal(DNI)*.
 2. Rellenamos el formulario con:
 - a. Datos personales: Extraídos del objeto *Personal*.
 - b. Tipo de persona: Gracias a *obtenerTipo*.
 - c. Teléfonos: Mediante el método *obtenerTfns*.

De esta secuencia de acciones se derivan los siguientes métodos nuevos:

MÉTODO	FUNCIONALIDAD
<i>Tarjeta</i>	Constructor que crea un objeto de tipo Tarjeta con el código de tarjeta tomado como parámetro de entrada. Nota: Siempre devuelve la de mayor fecha de alta.
<i>obtenerTipo</i>	Método que devuelve el tipo de persona (periódica, residente,...) del objeto que lo llama.
<i>obtenerEmpresa</i>	Método que indica la empresa para la que trabaja la persona que lo llama. Los resultados pueden ser: <ul style="list-style-type: none"> • Si es un trabajador periódico → Empresa y departamento para el que trabaja. • Si se trata de un trabajador ocasional → Empresa para la que trabaja. • En el caso de ser un trabajador residente → El resultado será siempre “Informática Navarra S.A.”. Nota: Los contactos no son tratados ya que nunca actuarán como prestatarios.
<i>obtenerTfns</i>	Método que proporciona un ResultSet con todos los teléfonos que el sistema posee de la persona que lo llama.

Métodos a implementar y a añadir a la API



Aclaremos todo:

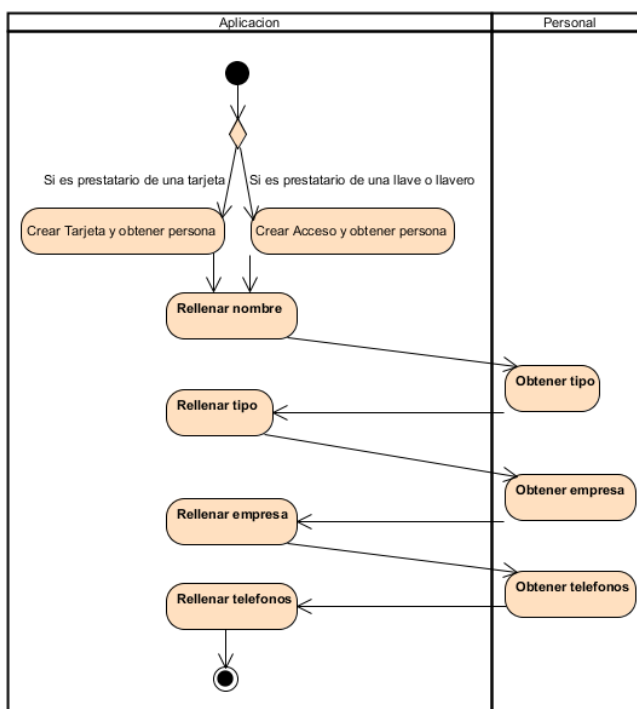


Diagrama de actividades

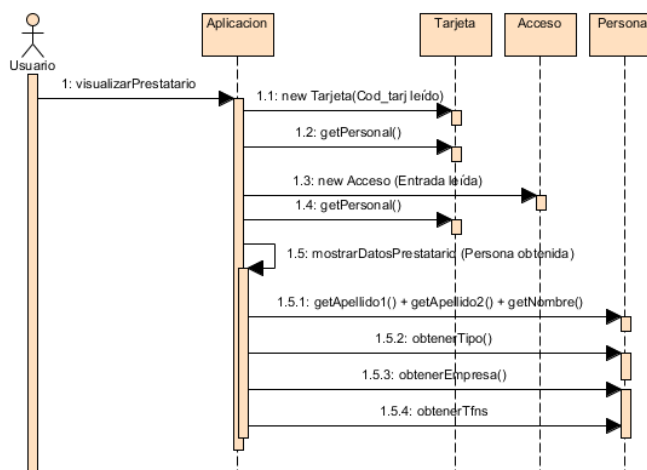
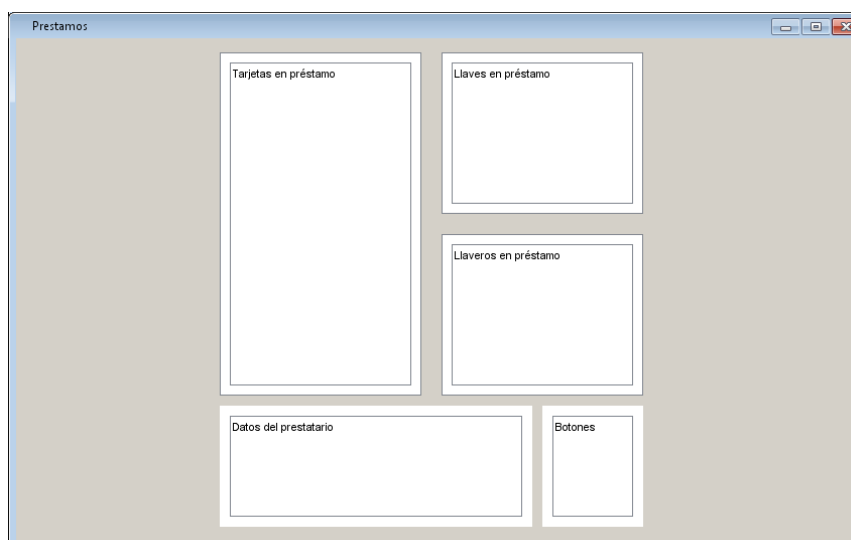


Diagrama de secuencia



4.3. Caso de uso 100: Devolver préstamo

Un simple botón nos permitirá devolver todos los préstamos que se encuentren seleccionados en pantalla:



Interfaz gráfica propuesta

Para llevar a cabo este caso de uso:

1. Obtendremos las tarjetas, llaves y llaveros seleccionados.
2. Recorreremos cada grupo devolviendo uno a uno cada elemento.
3. Refrescaremos las tablas del tab 'Préstamos'.

Tenemos todos los métodos creados con anterioridad, con que no es necesario que añadamos nada a la API.



El proceso se reduce a la siguiente secuencia de acciones:

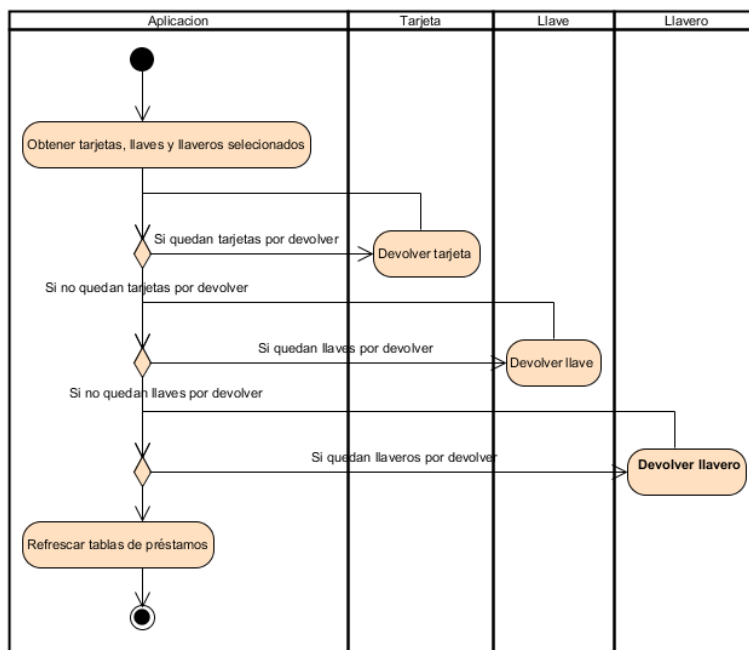


Diagrama de actividades

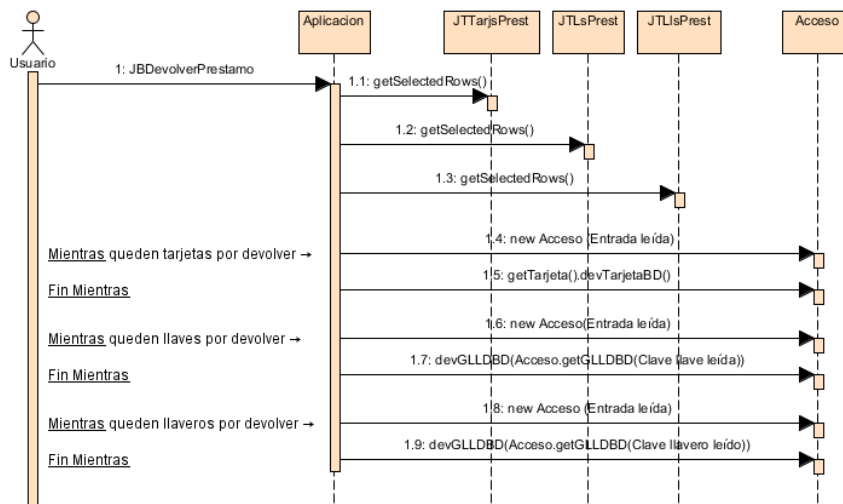


Diagrama de secuencia



4.4. Caso de uso 101: Enviar recordatorio vía E-Mail

Otro caso de uso en el que sólo necesitaremos añadir un nuevo botón. Éste permanecerá activo en aquellos trabajadores de los que se disponga de alguna dirección de correo electrónico almacenada, e inactivo en el caso contrario. Su pulsación desencadenará la siguiente secuencia de actividades:

1. Crear un objeto de tipo Cartero.
2. Hacer que el Cartero envíe un E-Mail a la última persona seleccionada en cualquiera de las tablas con todos los préstamos pendientes de devolución que ésta disponga a día de hoy.

Aquí tenemos una clase de la que no habíamos hecho mención hasta el momento: la clase Cartero. Ésta será una clase Java como cualquier otra que realizará las funciones de composición y envío de E-Mails gracias al paquete *JavaMail*.

A continuación exponemos los diagramas de actividades y de secuencia que emplearemos para la codificación, pero si el lector desea ahondar en cómo se implementa la clase Cartero le aconsejamos que recurra al código fuente proporcionado, al estudio de la API final aportado al final de la fase de Construcción y al capítulo 4 de la presente memoria, en el que se hace una breve mención a *JavaMail*.

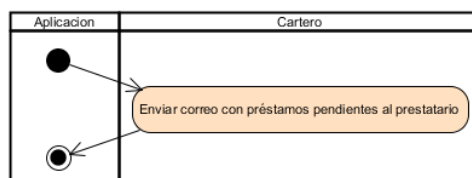


Diagrama de actividades

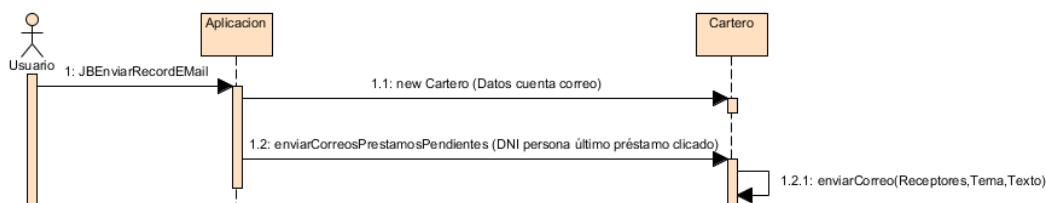
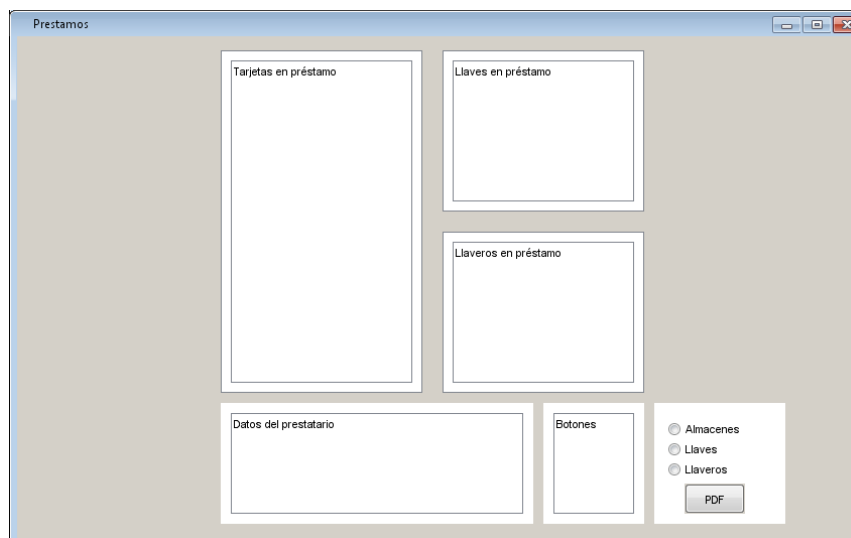


Diagrama de secuencia



4.5. Caso de uso 102: Listar préstamos pendientes

Terminamos esta pestaña proporcionando al usuario la funcionalidad de poder crear listados con los préstamos que están pendientes de devolución. Éste nos ha pedido expresamente que se puedan generar listados de parte o la totalidad de los préstamos que se encuentran pendientes, con que en lugar de añadir un simple botón que genere un listado por defecto, crearemos un Button Group con el que definir qué tablas queremos que aparezcan en el listado. Luego solo tendremos que leer las tablas con las que realizar el listado y pedir a un objeto de tipo `GeneradorInformes` que cree el que deseamos.



Interfaz gráfica propuesta

Lo que haremos será:

1. Desplegar un `JFileChooser` para que el usuario indique en que ubicación desea recibir el informe.
2. Crear un objeto de tipo `GeneradorInformes`.
3. Pasarle a este objeto el título, texto y subtítulos que deberá contener el reporte, junto con las tres tablas y el estado de los botones que componen el Button Group.
4. Pedir a `GeneradorInformes` que cree el informe de acuerdo a toda la información suministrada en el punto anterior.

Disponemos de todos los componentes necesarios para codificar la estrategia diseñada ya que el método que utilizaremos para crear los reportes (`generarInformeAvanzado`) fue creado en la iteración 'C4 - Empresas y personal).



Todo el proceso queda resumido en este diagrama...

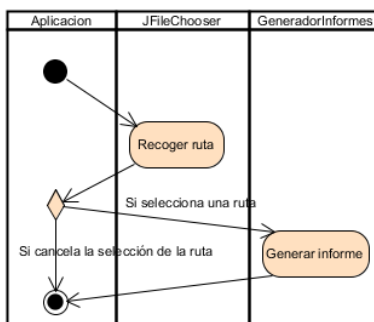


Diagrama de actividades

... y en este otro:

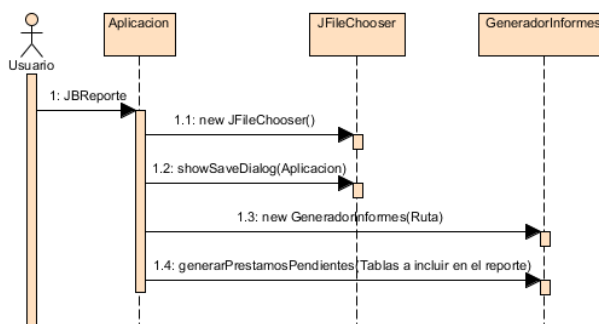


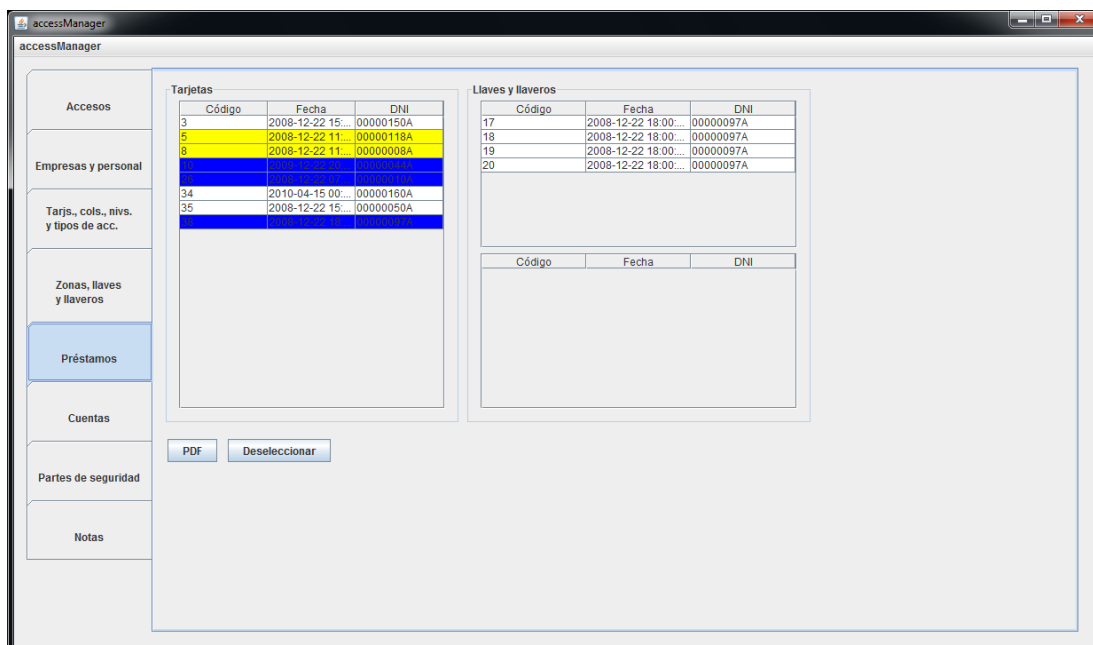
Diagrama de secuencia



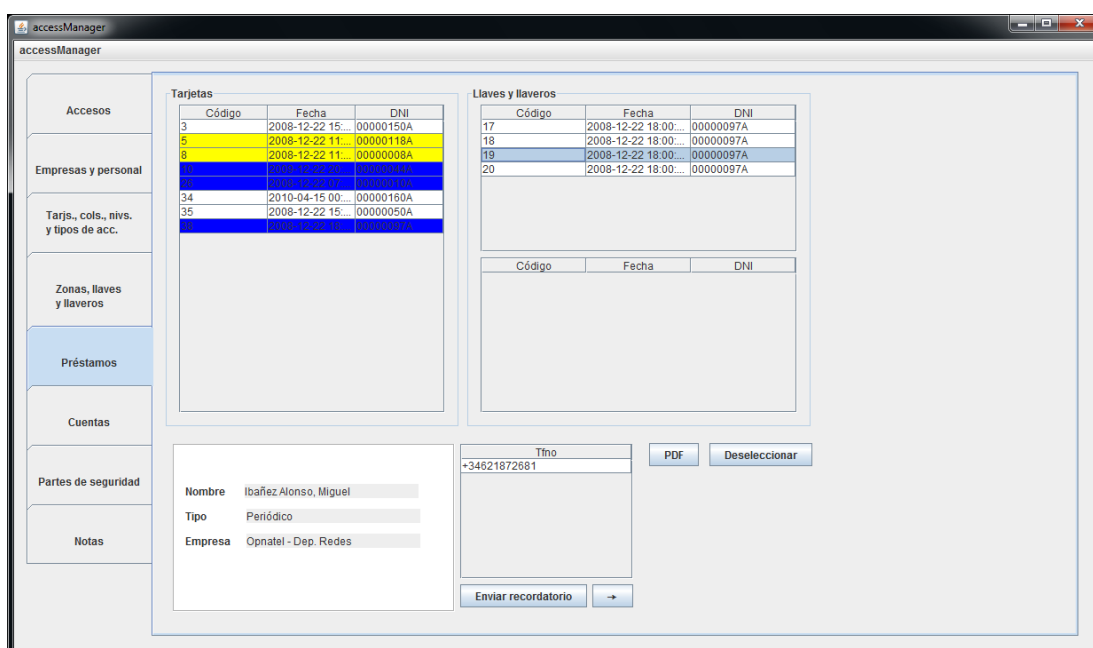
5. Codificación e integración

Es muy probable que nos encontremos ante una de las iteraciones más sencillas de codificar, y si antes no entramos a detallar cómo realizamos la codificación, en esta ocasión carece de más sentido aún. La única clase que puede entrañar algo más de complejidad (por desconocimiento, no por dificultad) es la de Cartero, pero como ya dijimos haremos un rápido estudio de la misma más adelante.

Integramos la pestaña de préstamo con las del resto resultando este diseño inicial:



Interfaz de la pestaña 'Préstamos'



Interfaz expandida de la pestaña 'Préstamos'



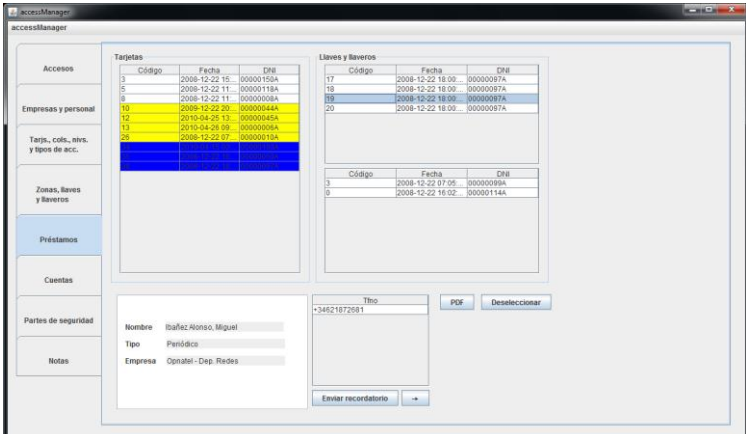
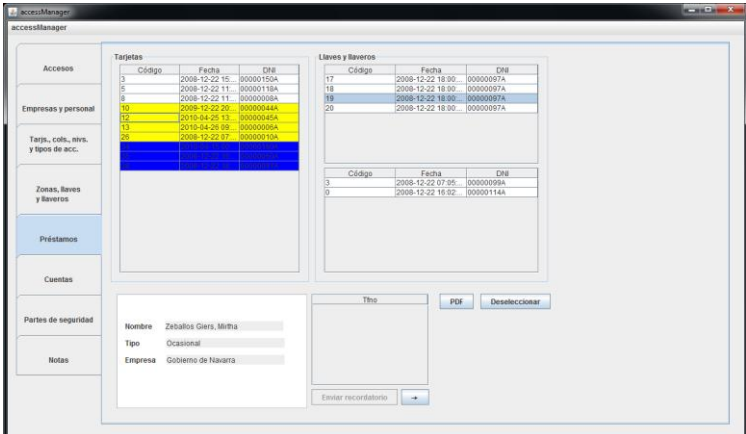
6. Pruebas

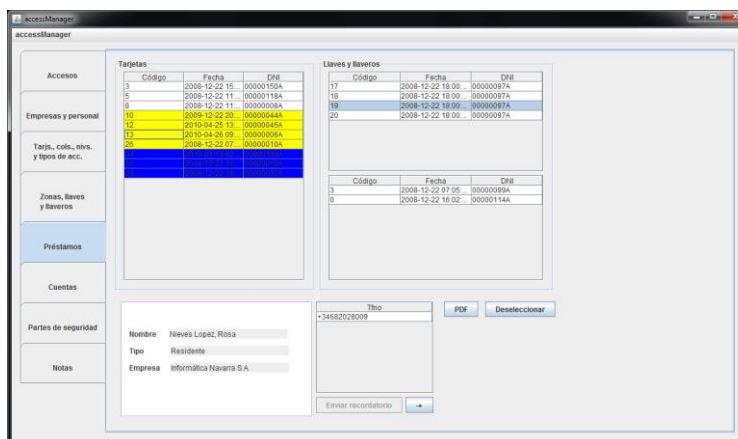
6.1. De unidad

En las siguientes páginas probamos la especificación de los elementos software desarrollados en la iteración:

Caso de prueba	7.01 - Caso de uso 98: Visualizar préstamos																																																																																				
Objetivo	Comprobar que si el administrador accede al tab de préstamos pendientes, podrá consultarlos de un vistazo y sin esfuerzo alguno.																																																																																				
Actividades	Arrancar accessManager y situarnos en la pestaña 'Préstamos', comprobando que lo que allí visualizamos se corresponde con los datos almacenados en la base de datos.																																																																																				
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. • Utilidad phpMyAdmin. 																																																																																				
Datos de entrada	La población de prueba residente en la base de datos.																																																																																				
Salida esperada	<p>Deberíamos observar en las tablas de tarjetas y llaves en préstamo las siguientes tuplas respectivamente:</p> <table border="1"> <thead> <tr> <th>Cod_Tarj</th> <th>F_Alta_T</th> <th>F_Baja_T</th> <th>Cod_Color</th> <th>F_Alta_C</th> <th>DNI</th> </tr> <tr> <th>Número de tarjeta</th> <th>Fecha y hora de inserción de la tarjeta</th> <th>Fecha y hora de baja de la tarjeta</th> <th>Color de la tarjeta</th> <th>Fecha y hora de inserción del color</th> <th>DNI de la persona que la tiene en su posesión</th> </tr> </thead> <tbody> <tr><td>3</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-1</td><td>2010-01-01 00:00:01</td><td>00000150A</td></tr> <tr><td>5</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-1</td><td>2010-01-01 00:00:01</td><td>00000118A</td></tr> <tr><td>8</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-1</td><td>2010-01-01 00:00:01</td><td>00000008A</td></tr> <tr><td>10</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-256</td><td>2010-01-01 00:00:01</td><td>00000044A</td></tr> <tr><td>26</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-256</td><td>2010-01-01 00:00:01</td><td>00000010A</td></tr> <tr><td>34</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-16776961</td><td>2010-01-01 00:00:01</td><td>00000160A</td></tr> <tr><td>35</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-16776961</td><td>2010-01-01 00:00:01</td><td>00000050A</td></tr> <tr><td>38</td><td>2010-01-01 00:00:01</td><td>NULL</td><td>-16776961</td><td>2010-01-01 00:00:01</td><td>00000097A</td></tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Entrada</th> <th>Cod_Llave</th> <th>F_Alta_L</th> <th>Devoluta</th> </tr> <tr> <th>Fecha y hora de acceso</th> <th>Número de llave prestada</th> <th>Fecha y hora de inserción de la llave.</th> <th>Denota si la llave ha sido devuelta</th> </tr> </thead> <tbody> <tr><td>2008-12-22 18:00:00</td><td>17</td><td>2010-01-01 00:00:01</td><td>0</td></tr> <tr><td>2008-12-22 18:00:00</td><td>18</td><td>2010-01-01 00:00:01</td><td>0</td></tr> <tr><td>2008-12-22 18:00:00</td><td>19</td><td>2010-01-01 00:00:01</td><td>0</td></tr> <tr><td>2008-12-22 18:00:00</td><td>20</td><td>2010-01-01 00:00:01</td><td>0</td></tr> </tbody> </table> <p>La de llaveros deberá estar vacía.</p>	Cod_Tarj	F_Alta_T	F_Baja_T	Cod_Color	F_Alta_C	DNI	Número de tarjeta	Fecha y hora de inserción de la tarjeta	Fecha y hora de baja de la tarjeta	Color de la tarjeta	Fecha y hora de inserción del color	DNI de la persona que la tiene en su posesión	3	2010-01-01 00:00:01	NULL	-1	2010-01-01 00:00:01	00000150A	5	2010-01-01 00:00:01	NULL	-1	2010-01-01 00:00:01	00000118A	8	2010-01-01 00:00:01	NULL	-1	2010-01-01 00:00:01	00000008A	10	2010-01-01 00:00:01	NULL	-256	2010-01-01 00:00:01	00000044A	26	2010-01-01 00:00:01	NULL	-256	2010-01-01 00:00:01	00000010A	34	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	00000160A	35	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	00000050A	38	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	00000097A	Entrada	Cod_Llave	F_Alta_L	Devoluta	Fecha y hora de acceso	Número de llave prestada	Fecha y hora de inserción de la llave.	Denota si la llave ha sido devuelta	2008-12-22 18:00:00	17	2010-01-01 00:00:01	0	2008-12-22 18:00:00	18	2010-01-01 00:00:01	0	2008-12-22 18:00:00	19	2010-01-01 00:00:01	0	2008-12-22 18:00:00	20	2010-01-01 00:00:01	0
Cod_Tarj	F_Alta_T	F_Baja_T	Cod_Color	F_Alta_C	DNI																																																																																
Número de tarjeta	Fecha y hora de inserción de la tarjeta	Fecha y hora de baja de la tarjeta	Color de la tarjeta	Fecha y hora de inserción del color	DNI de la persona que la tiene en su posesión																																																																																
3	2010-01-01 00:00:01	NULL	-1	2010-01-01 00:00:01	00000150A																																																																																
5	2010-01-01 00:00:01	NULL	-1	2010-01-01 00:00:01	00000118A																																																																																
8	2010-01-01 00:00:01	NULL	-1	2010-01-01 00:00:01	00000008A																																																																																
10	2010-01-01 00:00:01	NULL	-256	2010-01-01 00:00:01	00000044A																																																																																
26	2010-01-01 00:00:01	NULL	-256	2010-01-01 00:00:01	00000010A																																																																																
34	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	00000160A																																																																																
35	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	00000050A																																																																																
38	2010-01-01 00:00:01	NULL	-16776961	2010-01-01 00:00:01	00000097A																																																																																
Entrada	Cod_Llave	F_Alta_L	Devoluta																																																																																		
Fecha y hora de acceso	Número de llave prestada	Fecha y hora de inserción de la llave.	Denota si la llave ha sido devuelta																																																																																		
2008-12-22 18:00:00	17	2010-01-01 00:00:01	0																																																																																		
2008-12-22 18:00:00	18	2010-01-01 00:00:01	0																																																																																		
2008-12-22 18:00:00	19	2010-01-01 00:00:01	0																																																																																		
2008-12-22 18:00:00	20	2010-01-01 00:00:01	0																																																																																		
Salida observada	<p>La esperada:</p>																																																																																				
Evaluación/conclusión	<p>El caso de uso '98 - Visualizar préstamos' ha sido realizado satisfactoriamente.</p> <p><i>Caso de prueba para el caso de uso '98 – Visualizar préstamos'</i></p>																																																																																				



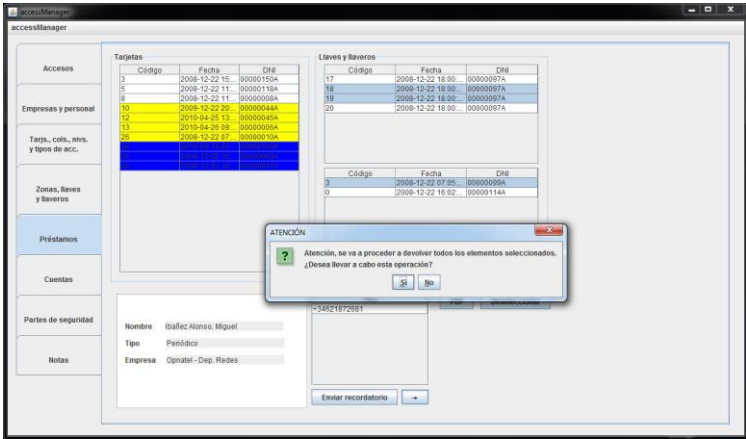
Caso de prueba	7.02 - Caso de uso 99: Visualizar datos del prestatario
Objetivo	Chequear que sea cual sea el tipo de persona que posee el préstamo es posible visualizar correctamente sus datos.
Actividades	Situarnos en el tab de préstamos e intentar visualizar los datos de tres tipos de prestatarios diferentes (periódico, ocasional y residente).
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. • Utilidad phpMyAdmin.
Datos de entrada	<p>Añadimos estos accesos a la población de prueba:</p> <pre>Ejecutar la(s) consulta(s) SQL en la base de datos bd: <input type="text"/> INSERT INTO Acceso VALUES ('00000045A', '2010-04-25 13:02:55', '2010-04-25 14:22:15', 'Trabajo', '012', '2010-01-01 00:00:01', 'Host', '00000003A'); ('00000006A', '2010-04-26 09:00:03', '2010-04-26 09:21:21', 'Trabajo', '013', '2010-01-01 00:00:01', 'Host', '00000003A');</pre> <p>Seleccionamos estos tres tipos de personas:</p> <ul style="list-style-type: none"> • Periódica: 00000097A • Ocasional: 00000045A • Residente: 00000006A
Salida esperada	En todos los casos deberán mostrarse los datos personales del prestatario, quedando activo el botón de envío de recordatorio solo en el primer caso.
Salida observada	<p>La esperada. Por orden:</p>  



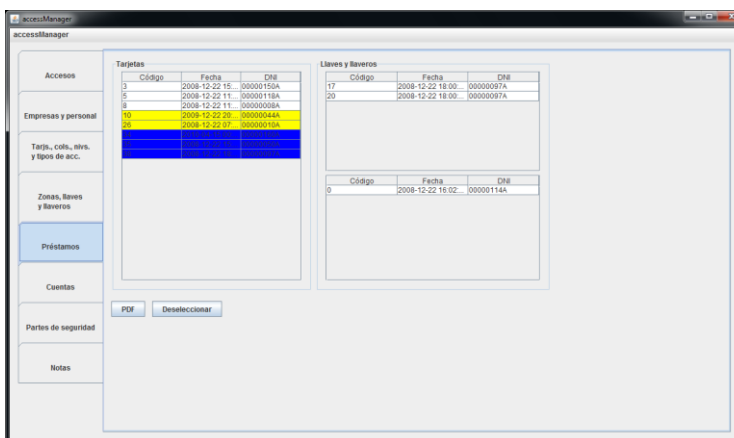
Evaluación/
conclusión

El caso de uso '99 - Visualizar datos del prestatario' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '99 - Visualizar datos del prestatario'

Caso de prueba	7.03 - Caso de uso 100: Devolver préstamo																																																																																																																																																											
Objetivo	Probar que el administrador es capaz de devolver todos los préstamos que desee desde esta pestaña.																																																																																																																																																											
Actividades	Loguear en el sistema, ir a la tab de préstamos, seleccionar varios de ellos y devolverlos, comprobando que las tablas se actualizan en 'Préstamos' y 'Accesos'.																																																																																																																																																											
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. 																																																																																																																																																											
Datos de entrada	<p>Probamos a borrar elementos de todo tipo pertenecientes a varias personas:</p> <ul style="list-style-type: none"> • Tarjetas 12 y 13 – 00000045A (ocasional) y 00000006A (residente). • Llaves 18 y 19 – 00000097A (periódica). • Llavero 3 – 000099A (periódica).  <table border="1" data-bbox="563 1216 1169 1440"> <thead> <tr> <th>DNI</th> <th>Entrada</th> <th>Salida</th> <th>Revisión</th> <th>Motivo</th> </tr> </thead> <tbody> <tr><td>0000044A</td><td>2008-12-22 15:55:00.0</td><td>2008-12-22 15:55:00.0</td><td></td><td></td></tr> <tr><td>0000048A</td><td>2008-12-22 15:55:12.0</td><td>2008-12-22 20:05:00.0</td><td></td><td>Trabajo</td></tr> <tr><td>0000048A</td><td>2008-12-22 15:55:23.0</td><td>2008-12-22 20:05:00.0</td><td></td><td>Trabajo</td></tr> <tr><td>0000048A</td><td>2008-12-22 15:55:52.0</td><td>2008-12-22 20:05:00.0</td><td></td><td>Trabajo</td></tr> <tr><td>0000048A</td><td>2008-12-22 15:55:52.0</td><td>2008-12-22 20:05:00.0</td><td></td><td>Trabajo</td></tr> <tr><td>0000014A</td><td>2008-12-22 16:02:00.0</td><td>2008-12-22 19:10:00.0</td><td></td><td>Revisión</td></tr> <tr><td>0000097A</td><td>2008-12-22 16:02:00.0</td><td></td><td></td><td>Host</td></tr> <tr><td>0000098A</td><td>2008-12-22 16:05:00.0</td><td>2008-12-22 20:15:00.0</td><td></td><td>Host</td></tr> <tr><td>0000045A</td><td>2008-12-22 20:00:00.0</td><td>2008-12-22 20:20:00.0</td><td></td><td>Agendar el servicio</td></tr> <tr><td>0000045A</td><td>2008-12-24 00:20:00.0</td><td>2008-12-25 00:40:00.0</td><td></td><td>Trabajo</td></tr> <tr><td>0000006A</td><td>2009-12-22 13:12:00.0</td><td></td><td></td><td>Trabajo</td></tr> <tr><td>0000045A</td><td>2009-12-22 20:00:00.0</td><td></td><td></td><td>Trabajo</td></tr> <tr><td>0000045A</td><td>2010-04-15 00:20:00.0</td><td></td><td></td><td>Trabajo</td></tr> <tr><td>0000045A</td><td>2010-04-25 13:02:00.0</td><td>2010-04-25 14:22:00.0</td><td></td><td>Trabajo</td></tr> <tr><td>0000006A</td><td>2010-04-26 09:00:00.0</td><td>2010-04-26 09:21:21.0</td><td></td><td>Trabajo</td></tr> </tbody> </table> <table border="1" data-bbox="563 1462 1169 1664"> <thead> <tr> <th>DNI</th> <th>Entrada</th> <th>Salida</th> <th>Host</th> <th>Motivo</th> </tr> </thead> <tbody> <tr><td>0000099A</td><td>2008-12-22 07:00:00.0</td><td>2008-12-22 08:30:00.0</td><td></td><td></td></tr> <tr><td>0000013A</td><td>2008-12-22 07:30:00.0</td><td>2008-12-22 14:45:00.0</td><td></td><td>Trabajo</td></tr> <tr><td>0000009A</td><td>2008-12-22 07:45:00.0</td><td>2008-12-22 08:05:00.0</td><td></td><td>Host</td></tr> <tr><td>0000013A</td><td>2008-12-22 08:31:00.0</td><td>2008-12-22 11:15:00.0</td><td></td><td>Trabajo</td></tr> <tr><td>0000037A</td><td>2008-12-22 08:32:00.0</td><td>2008-12-22 11:15:00.0</td><td></td><td>Trabajo</td></tr> <tr><td>0000012A</td><td>2008-12-22 08:10:00.0</td><td>2008-12-22 11:48:00.0</td><td></td><td>Mantenimiento</td></tr> <tr><td>0000013A</td><td>2008-12-22 08:10:00.0</td><td>2008-12-22 11:48:00.0</td><td></td><td>Mantenimiento</td></tr> <tr><td>0000013A</td><td>2008-12-22 09:20:00.0</td><td>2008-12-22 12:50:00.0</td><td></td><td>Trabajo</td></tr> <tr><td>0000011A</td><td>2008-12-22 11:40:25.0</td><td>2008-12-22 14:25:00.0</td><td></td><td>Sistemas</td></tr> <tr><td>0000011A</td><td>2008-12-22 11:40:40.0</td><td>2008-12-22 14:25:00.0</td><td></td><td>Sistemas</td></tr> <tr><td>0000012A</td><td>2008-12-22 11:40:57.0</td><td>2008-12-22 14:25:00.0</td><td></td><td>Sistemas</td></tr> <tr><td>0000010A</td><td>2008-12-22 13:00:00.0</td><td>2008-12-22 14:10:00.0</td><td></td><td>Trabajo</td></tr> <tr><td>0000006A</td><td>2008-12-22 13:00:00.0</td><td>2008-12-22 14:02:00.0</td><td></td><td>Trabajo</td></tr> <tr><td>0000010A</td><td>2008-12-22 13:02:21.0</td><td>2008-12-22 14:10:00.0</td><td></td><td>Trabajo</td></tr> </tbody> </table>	DNI	Entrada	Salida	Revisión	Motivo	0000044A	2008-12-22 15:55:00.0	2008-12-22 15:55:00.0			0000048A	2008-12-22 15:55:12.0	2008-12-22 20:05:00.0		Trabajo	0000048A	2008-12-22 15:55:23.0	2008-12-22 20:05:00.0		Trabajo	0000048A	2008-12-22 15:55:52.0	2008-12-22 20:05:00.0		Trabajo	0000048A	2008-12-22 15:55:52.0	2008-12-22 20:05:00.0		Trabajo	0000014A	2008-12-22 16:02:00.0	2008-12-22 19:10:00.0		Revisión	0000097A	2008-12-22 16:02:00.0			Host	0000098A	2008-12-22 16:05:00.0	2008-12-22 20:15:00.0		Host	0000045A	2008-12-22 20:00:00.0	2008-12-22 20:20:00.0		Agendar el servicio	0000045A	2008-12-24 00:20:00.0	2008-12-25 00:40:00.0		Trabajo	0000006A	2009-12-22 13:12:00.0			Trabajo	0000045A	2009-12-22 20:00:00.0			Trabajo	0000045A	2010-04-15 00:20:00.0			Trabajo	0000045A	2010-04-25 13:02:00.0	2010-04-25 14:22:00.0		Trabajo	0000006A	2010-04-26 09:00:00.0	2010-04-26 09:21:21.0		Trabajo	DNI	Entrada	Salida	Host	Motivo	0000099A	2008-12-22 07:00:00.0	2008-12-22 08:30:00.0			0000013A	2008-12-22 07:30:00.0	2008-12-22 14:45:00.0		Trabajo	0000009A	2008-12-22 07:45:00.0	2008-12-22 08:05:00.0		Host	0000013A	2008-12-22 08:31:00.0	2008-12-22 11:15:00.0		Trabajo	0000037A	2008-12-22 08:32:00.0	2008-12-22 11:15:00.0		Trabajo	0000012A	2008-12-22 08:10:00.0	2008-12-22 11:48:00.0		Mantenimiento	0000013A	2008-12-22 08:10:00.0	2008-12-22 11:48:00.0		Mantenimiento	0000013A	2008-12-22 09:20:00.0	2008-12-22 12:50:00.0		Trabajo	0000011A	2008-12-22 11:40:25.0	2008-12-22 14:25:00.0		Sistemas	0000011A	2008-12-22 11:40:40.0	2008-12-22 14:25:00.0		Sistemas	0000012A	2008-12-22 11:40:57.0	2008-12-22 14:25:00.0		Sistemas	0000010A	2008-12-22 13:00:00.0	2008-12-22 14:10:00.0		Trabajo	0000006A	2008-12-22 13:00:00.0	2008-12-22 14:02:00.0		Trabajo	0000010A	2008-12-22 13:02:21.0	2008-12-22 14:10:00.0		Trabajo
DNI	Entrada	Salida	Revisión	Motivo																																																																																																																																																								
0000044A	2008-12-22 15:55:00.0	2008-12-22 15:55:00.0																																																																																																																																																										
0000048A	2008-12-22 15:55:12.0	2008-12-22 20:05:00.0		Trabajo																																																																																																																																																								
0000048A	2008-12-22 15:55:23.0	2008-12-22 20:05:00.0		Trabajo																																																																																																																																																								
0000048A	2008-12-22 15:55:52.0	2008-12-22 20:05:00.0		Trabajo																																																																																																																																																								
0000048A	2008-12-22 15:55:52.0	2008-12-22 20:05:00.0		Trabajo																																																																																																																																																								
0000014A	2008-12-22 16:02:00.0	2008-12-22 19:10:00.0		Revisión																																																																																																																																																								
0000097A	2008-12-22 16:02:00.0			Host																																																																																																																																																								
0000098A	2008-12-22 16:05:00.0	2008-12-22 20:15:00.0		Host																																																																																																																																																								
0000045A	2008-12-22 20:00:00.0	2008-12-22 20:20:00.0		Agendar el servicio																																																																																																																																																								
0000045A	2008-12-24 00:20:00.0	2008-12-25 00:40:00.0		Trabajo																																																																																																																																																								
0000006A	2009-12-22 13:12:00.0			Trabajo																																																																																																																																																								
0000045A	2009-12-22 20:00:00.0			Trabajo																																																																																																																																																								
0000045A	2010-04-15 00:20:00.0			Trabajo																																																																																																																																																								
0000045A	2010-04-25 13:02:00.0	2010-04-25 14:22:00.0		Trabajo																																																																																																																																																								
0000006A	2010-04-26 09:00:00.0	2010-04-26 09:21:21.0		Trabajo																																																																																																																																																								
DNI	Entrada	Salida	Host	Motivo																																																																																																																																																								
0000099A	2008-12-22 07:00:00.0	2008-12-22 08:30:00.0																																																																																																																																																										
0000013A	2008-12-22 07:30:00.0	2008-12-22 14:45:00.0		Trabajo																																																																																																																																																								
0000009A	2008-12-22 07:45:00.0	2008-12-22 08:05:00.0		Host																																																																																																																																																								
0000013A	2008-12-22 08:31:00.0	2008-12-22 11:15:00.0		Trabajo																																																																																																																																																								
0000037A	2008-12-22 08:32:00.0	2008-12-22 11:15:00.0		Trabajo																																																																																																																																																								
0000012A	2008-12-22 08:10:00.0	2008-12-22 11:48:00.0		Mantenimiento																																																																																																																																																								
0000013A	2008-12-22 08:10:00.0	2008-12-22 11:48:00.0		Mantenimiento																																																																																																																																																								
0000013A	2008-12-22 09:20:00.0	2008-12-22 12:50:00.0		Trabajo																																																																																																																																																								
0000011A	2008-12-22 11:40:25.0	2008-12-22 14:25:00.0		Sistemas																																																																																																																																																								
0000011A	2008-12-22 11:40:40.0	2008-12-22 14:25:00.0		Sistemas																																																																																																																																																								
0000012A	2008-12-22 11:40:57.0	2008-12-22 14:25:00.0		Sistemas																																																																																																																																																								
0000010A	2008-12-22 13:00:00.0	2008-12-22 14:10:00.0		Trabajo																																																																																																																																																								
0000006A	2008-12-22 13:00:00.0	2008-12-22 14:02:00.0		Trabajo																																																																																																																																																								
0000010A	2008-12-22 13:02:21.0	2008-12-22 14:10:00.0		Trabajo																																																																																																																																																								
Salida esperada	Los registros seleccionados en 'Préstamos' deberán desaparecer y sus homólogos en 'Accesos' deberán quedar como devueltos.																																																																																																																																																											

La espera. Así queda la pestaña de 'Préstamos'...



Salida observada

... y así los accesos probados:

DNI	Entrada	Salida	Motivo
00000118A	2008-12-22 11:40:45.0	2008-12-22 14:25:00.0	Sistemas
00000129A	2008-12-22 11:40:57.0	2008-12-22 14:25:00.0	Sistemas
00000104A	2008-12-22 13:00:02.0	2008-12-22 14:10:00.0	Trabajo
00000094A	2008-12-22 13:00:05.0	2008-12-22 14:02:00.0	Trabajo
00000105A	2008-12-22 13:02:21.0	2008-12-22 14:10:00.0	Trabajo
00000032A	2008-12-22 13:45:00.0	2008-12-22 14:10:00.0	Materia informatica
00000094A	2008-12-22 13:50:01.0	2008-12-22 14:05:00.0	Trabajo
00000067A	2008-12-22 13:50:20.0	2008-12-22 14:55:00.0	Trabajo
00000044A	2008-12-22 15:50:29.0	2008-12-22 16:50:00.0	Revision
00000148A	2008-12-22 16:56:12.0	2008-12-22 20:05:00.0	Trabajo
00000148A	2008-12-22 16:56:23.0	2008-12-22 20:05:00.0	Trabajo
00000150A	2008-12-22 16:56:53.0	2008-12-22 20:05:00.0	Trabajo
00000114A	2008-12-22 16:02:00.0	2008-12-22 19:10:00.0	Revision
00000097A	2008-12-22 18:00:00.0		Host

Datos		Llaves y Seros			Tarjeta	
DNI	Primer apellido	Segundo apellido	Nombre	Tipo	Código	Devoluta
00000097A	Ibañaz	Alonso	Miguel	Llave	17	No
				Llave	18	Si
				Llave	19	No
				Llave	20	Si

DNI	Entrada	Salida	Motivo
00000099A	2008-12-22 07:05:00.0	2008-12-22 08:39:00.0	Host
0000010A	2008-12-22 07:30:00.0	2008-12-22 14:45:00.0	Trabajo
00000050A	2008-12-22 07:45:00.0	2008-12-22 09:05:00.0	Host
00000035A	2008-12-22 08:31:00.0	2008-12-22 11:15:00.0	Trabajo
00000037A	2008-12-22 08:32:00.0	2008-12-22 11:15:00.0	Trabajo
00000129A	2008-12-22 09:10:02.0	2008-12-22 11:48:00.0	Mantenimiento
00000129A	2008-12-22 09:10:52.0	2008-12-22 11:48:00.0	Mantenimiento
00000135A	2008-12-22 10:20:00.0	2008-12-22 12:50:00.0	Trabajo
00000118A	2008-12-22 11:40:20.0	2008-12-22 14:20:00.0	Sistemas
00000129A	2008-12-22 11:40:45.0	2008-12-22 14:25:00.0	Sistemas
00000104A	2008-12-22 11:40:57.0	2008-12-22 14:10:00.0	Trabajo
00000094A	2008-12-22 13:00:05.0	2008-12-22 14:02:00.0	Trabajo
00000105A	2008-12-22 13:02:21.0	2008-12-22 14:10:00.0	Trabajo

Datos		Llaves y Seros			Tarjeta	
DNI	Primer apellido	Segundo apellido	Nombre	Tipo	Código	Devoluta
00000099A	Ibañaz	Alvarez		Llave	13	Si

Apuntes

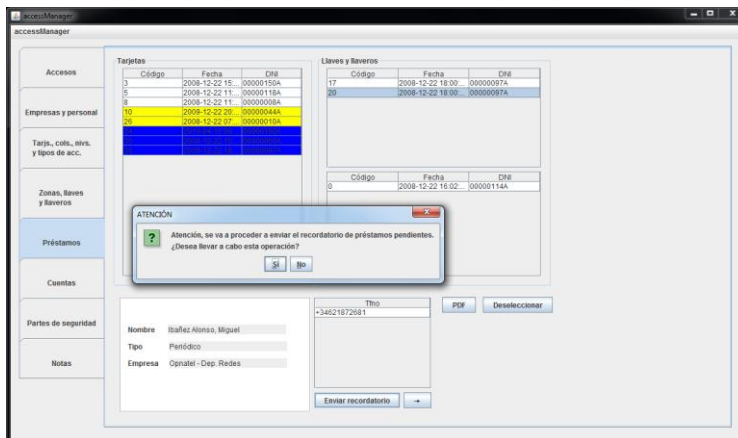
Ahora que disponemos de esta pestaña desarrollada, haremos la comprobación contraria, que la devolución de un préstamo desde la pestaña de accesos se ve reflejada en ésta.

Evaluación/
conclusión

El caso de uso '100 - Devolver préstamo' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '100 - Devolver préstamo'

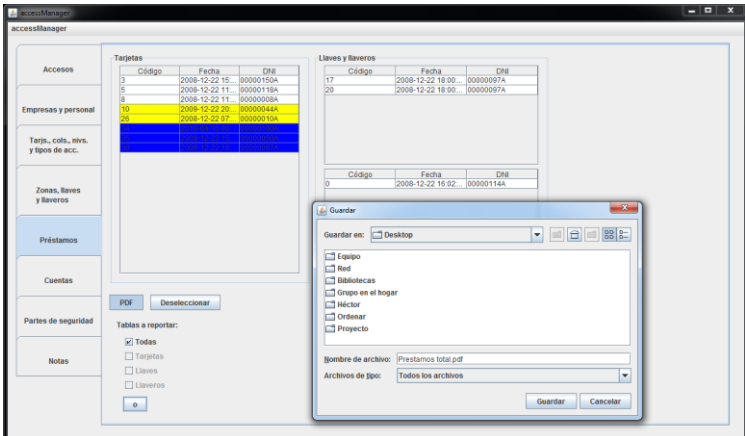
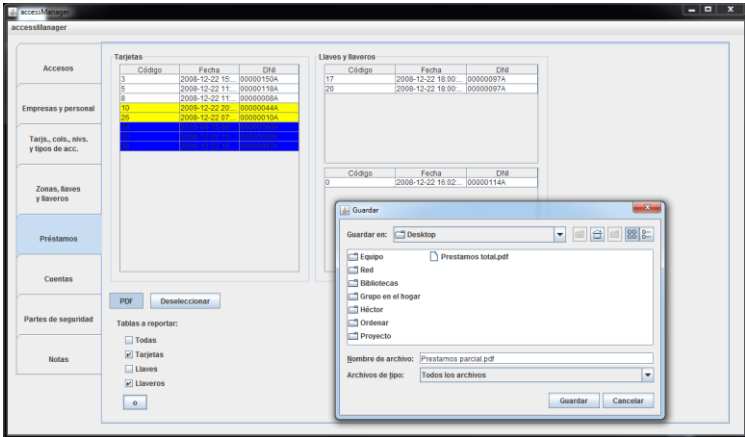


Caso de prueba	7.04 - Caso de uso 101: Enviar recordatorio vía E-Mail
Objetivo	Verificar que si el usuario pulsa el botón de 'Enviar recordatorio', éste se genera y envía al prestatario de forma correcta.
Actividades	Seleccionar un prestatario del que se tenga alguna dirección de correo electrónico y comprobar que el E-Mail es generado y enviado correctamente.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior. • Explorador web.
Datos de entrada	<p>Enviaremos un recordatorio a la persona con DNI 00000097A, ya que dispone de tarjeta y llaves en préstamo y el sistema tiene almacenadas dos direcciones de correo suyas:</p>  <p>The screenshot shows the 'accessManager' application interface. On the left, there is a navigation menu with categories like 'Accesos', 'Empresas y personal', 'Tarjetas, cédul., mtrs. y tipos de acc.', 'Zonas, llaves y llaveros', 'Préstamos', 'Cuentas', and 'Partes de seguridad'. The main area displays two tables: 'Tarjetas' and 'Llaves y llaveros'. The 'Tarjetas' table has columns for 'Código', 'Fecha', and 'DNI', with rows 3, 5, 8, 10, and 20. The 'Llaves y llaveros' table has columns for 'Código', 'Fecha', and 'DNI', with rows 17 and 20. A confirmation dialog box titled 'ATENCIÓN' is overlaid on the screen, asking '¿Desea llevar a cabo esta operación?' with 'Sí' and 'No' buttons. Below the dialog, there is a form with fields for 'Nombre' (Ibañez Alonso, Miguel), 'Tipo' (Periférico), 'Empresa' (Ognatel - Dep. Redes), and 'Título' (+34621872981). There are buttons for 'PDF', 'Deseleccionar', and 'Enviar recordatorio'.</p>
Salida esperada	El prestatario deberá recibir en ambas cuentas un correo electrónico haciéndole saber que dispone de una tarjeta y dos llaves por devolver.
Salida observada	<p>La esperada:</p> <pre> Prestamos pendientes de devolución con 'Informática Navarra S.A.' De: informaticanavarrasa@gmail.com Enviado: martes, 25 de mayo de 2010 12:11:41 Para: hecornapal@gmail.com; mickey1986@hotmail.com Estimado/a Miguel Ibañez, nos ponemos en contacto con usted para hacerle llegar la lista de préstamos pendientes de devolución que posee con nosotros. Le rogamos nos haga llegar el siguiente material lo antes posible. Reciba un cordial saludo, Informática Navarra S.A. ----- TIPO - CÓDIGO - FECHA DE PRÉSTAMO ----- Llave - 17 - 2008-12-22 18:00:00 Llave - 20 - 2008-12-22 18:00:00 Tarjeta - 38 - 2008-12-22 18:00:00 ----- </pre> <p>Prestamos pendientes de devolución con 'Informática Navarra S.A.' Recibidos X</p> <p>informaticanavarrasa@gmail.com para usuario, mickey1986 mostrar detalles 12:11 (1 minuto antes) Responder</p> <p>Estimado/a Miguel Ibañez, nos ponemos en contacto con usted para hacerle llegar la lista de préstamos pendientes de devolución que posee con nosotros. Le rogamos nos haga llegar el siguiente material lo antes posible.</p> <p>Reciba un cordial saludo, Informática Navarra S.A.</p> <p>TIPO - CÓDIGO - FECHA DE PRÉSTAMO</p> <p>Llave - 17 - 2008-12-22 18:00:00 Llave - 20 - 2008-12-22 18:00:00 Tarjeta - 38 - 2008-12-22 18:00:00</p>
Evaluación/ conclusión	El caso de uso '101 - Enviar recordatorio vía E-Mail' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '101 - Enviar recordatorio vía E-Mail'

accessManager · Iteración C7: Préstamos · Página 24 de 27



Caso de prueba	7.05 - Caso de uso 102: Listar préstamos pendientes
Objetivo	Chequear que accessManager permite realizar un listado en PDF con los préstamos que estimemos oportunos.
Actividades	Arrancar la aplicación, situarnos en la pestaña de préstamos y generar un informe total y otro parcial de los préstamos existentes, comprobando que son correctos.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos resultante del caso de prueba anterior.• Aplicación Adobe Reader.
Datos de entrada	<p>Generamos dos tipos de listados: uno total y otro de tarjetas y llaveros:</p>  
Salida esperada	Obtendremos dos listados: uno con todos los préstamos pendientes y otro con sólo las llaves y llaveros.



<p>Salida observada</p>	<p>La esperada:</p> <div style="display: flex; justify-content: space-around;"> <div data-bbox="563 255 940 777"> </div> <div data-bbox="971 255 1348 777"> </div> </div> <p>Nota: Si no se seleccionan elementos a reportar, la aplicación informa del error.</p>
<p>Evaluación/ conclusión</p>	<p>El caso de uso '102 - Listar préstamos pendientes' ha sido realizado satisfactoriamente.</p>

Caso de prueba para el caso de uso '102 - Listar préstamos pendientes'

6.2. De integración

El sistema, en lo que a componentes software se refiere, sigue permaneciendo inalterado, solo que con la aplicación Java desarrollada en mayor grado. Las pruebas realizadas han sido llevadas a cabo sin error alguno, con que no apreciamos que pueda existir algún tipo de problema de integración. Si nos centramos en la integración entre pestañas (más bien deberíamos hablar de la coherencia entre ellas), el caso de prueba 7.03 ha demostrado que las pestañas 'Accesos' y 'Préstamos' se relacionan de forma correcta.

6.3. De sistema

Todos los objetivos impuestos al inicio de la iteración y que tenían como base la implementación de los casos de uso desarrollados han sido cumplidos. Las pruebas han demostrado que la pestaña de préstamos funciona de forma robusta y vemos este documento suficiente de cara a una posterior comprensión de la funcionalidad o mantenimiento de la misma.



7. Evaluación de la iteración

Casi tenemos completada la séptima iteración de la fase de Construcción. Evaluando la pestaña desarrollada podemos decir que hemos conseguido un diseño muy simple pero eficaz en proporcionar la funcionalidad que debe. Permite realizar la devolución de préstamos y el envío de recordatorios con un gesto tan sencillo como es el clicar un botón, y las pruebas llevadas a cabo han demostrado que todo se desarrolla a la perfección, según lo diseñado, con que tenemos argumentos para estar muy satisfechos con el trabajo realizado. Veamos que piensa el cliente...

8. Despliegue

Con este documento y el prototipo hasta el momento desarrollado comenzamos la reunión prevista con el cliente. En ella le explicamos el planteamiento y desarrollo que hemos realizado en esta pestaña, mostrándole y realizando de nuevo todas las pruebas de unidad que ejecutamos en la fase de pruebas, dando nuevamente resultados positivos. El cliente estima que la pestaña se adapta perfectamente a sus necesidades, ofreciéndole toda la funcionalidad necesaria de una forma sencilla, clara y directa, no viendo inconveniente alguno en que pasemos a iniciar la siguiente iteración del proceso de desarrollo.

Pamplona, a 10 de marzo de 2010.

Héctor Napal Basterrica
Responsable del proyecto

Pedro Martínez Pérez
*Jefe de área de producción de
Informática Navarra S.A.*



accessManager
Desarrollo de un sistema de control de accesos a un centro

Iteración C8: Cuentas

Fecha de entrega: 23 de marzo de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Definiciones, acrónimos y abreviaciones
- 1.4. Estructura del documento

2. Estado del proyecto

3. Análisis

- 3.1. Planificación de la iteración
- 3.2. Análisis de los casos de uso y escenarios

4. Diseño

- 4.1. Casos de uso 103 y 104: Visualizar cuentas y visualizar cuenta
- 4.2. Caso de uso 105: Añadir cuenta
- 4.3. Caso de uso 106: Borrar cuenta
- 4.4. Caso de uso 107: Modificar cuenta
- 4.5. Caso de uso 108: Listar cuentas
- 4.6. Casos de uso 109 y 110: Visualizar tipos de usuario y visualizar tipo de usuario
- 4.7. Caso de uso 111: Añadir tipo de usuario
- 4.8. Caso de uso 112: Borrar tipo de usuario
- 4.9. Caso de uso 113: Modificar tipo de usuario
- 4.10. Caso de uso 114: Visualizar accesos al sistema
- 4.11. Caso de uso 115: Listar accesos al sistema
- 4.12. Caso de uso 116: Modificar datos de cuenta

5. Codificación e integración

6. Pruebas

- 6.1. De unidad
- 6.2. De integración
- 6.3. De sistema

7. Evaluación de la iteración

8. Despliegue



1. Introducción

1.1. Propósito

El propósito de este documento es el de recoger todo el trabajo que llevaremos a cabo entre hoy y mañana para conseguir los objetivos de esta iteración.

1.2. Audiencia

El carácter de este documento no varía con respecto a sus siete predecesores. Recoge información significativa de las fases de análisis, diseño, codificación y pruebas, pero el cliente no necesitará de ésta hasta que le sea proporcionado el sistema final.

1.3. Definiciones, acrónimos y abreviaciones

- | | | | |
|---|-----|-----------------------------------|--------------------------------------|
| • | BD | | Base de datos |
| • | API | Application Programming Interface | Interfaz de program. de aplicaciones |
| • | GUI | Graphical User Interface | Interfaz gráfica de usuario |

1.4. Estructura del documento

La sección que comienza tras estas líneas recoge una breve mención a en qué punto del proceso de desarrollo nos encontramos y qué debemos hacer a lo largo de esta iteración.

En el siguiente apartado la planificamos y analizamos los casos de uso que desarrollaremos.

Tras disponer de un buen análisis, en el capítulo 4 diseñamos y explicamos cómo elaboraremos el software que dé respuesta a las funcionalidades requeridas en los caso de uso.

A continuación exponemos los detalles de implementación que creemos necesario que conozca el lector, desplegamos todo el conjunto de pruebas realizadas para asegurar la calidad del software y dedicamos los dos últimos capítulos a evaluar el trabajo desarrollado y los resultados obtenidos en la etapa de despliegue.

2. Estado del proyecto

Una vez completada la iteración de préstamos, iniciamos una que tendrá como objetivo no el manejar datos que tengan relación con la vida diaria del centro, con el control de accesos, si no la de tratar todos aquellos datos concernientes a la aplicación en si misma: las cuentas existentes, los tipos de usuario con los que se relacionan y la frecuencia con la que son utilizadas para acceder al sistema.



3. Análisis

3.1. Planificación de la iteración

Esta es la disposición de actividades para hoy y mañana:

PLAN DE ITERACIÓN C8: CUENTAS			
ASPECTO	DESCRIPCIÓN		
Objetivos	Implementar y documentar toda la pestaña 'Cuentas', que recogerá toda la funcionalidad que tenga relación con el uso de accessManager.		
Artefactos de entrada	<ul style="list-style-type: none"> • <i>Especificación inicial de los casos de uso</i> • <i>Planificación temporal y costes detallados del proyecto</i> • Base de datos, API y aplicación desarrolladas en las iteraciones anteriores. 		
Artefactos modificados o creados	<ul style="list-style-type: none"> • Diagrama de clases de la API inicial. • Clases Java de la API inicial y de la aplicación. • El presente documento. 		
Calendarización detallada	Análisis + Pruebas	22/03/10	<ul style="list-style-type: none"> • Planificación de la iteración y gestión de riesgos. • Análisis de los casos de uso y escenarios. • Planificación de las pruebas. • Inicio de la documentación.
	Diseño + Pruebas	22/03/10	<ul style="list-style-type: none"> • Diseño de opciones arquitectónicas. • Diseño de pruebas.
	Codificación + Pruebas	22/03/10	<ul style="list-style-type: none"> • Codificación e integración de la solución. • Ejecución y evaluación de pruebas. • Evaluación del prototipo. • Preparación de la entrega ejecutable.
		23/03/10	
Despliegue	23/03/10	<ul style="list-style-type: none"> • Muestra, prueba y evaluación conjunta con el cliente. • Concreción de los cambios necesarios. • Finalización de toda la documentación. 	

Planificación de la iteración



Nombre de tarea	Duración	Comienzo	Fin	0	lun 22 mar	0	0
Estudio de oportunidad (inception)	7 días	jue 07/01/10	vie 15/01/10				
Elaboración	16 días	lun 18/01/10	lun 08/02/10				
Construcción	41 días	mar 09/02/10	mar 06/04/10				
Iteración C1: La base de datos	4 días	mar 09/02/10	vie 12/02/10				
Iteración C2: API para la base de datos	5 días	lun 15/02/10	vie 19/02/10				
Iteración C3: Accesos	5 días	lun 22/02/10	vie 26/02/10				
Iteración C4: Empresas y personal	5 días	lun 01/03/10	vie 05/03/10				
Iteración C5: Tarjetas, tipos de acceso y cargos	3 días	lun 08/03/10	mié 10/03/10				
Iteración C6: Zonas, llaves y llaveros	5 días	jue 11/03/10	mié 17/03/10				
Iteración C7: Préstamos	2 días	jue 18/03/10	vie 19/03/10				
Iteración C8: Cuentas	2 días	lun 22/03/10	mar 23/03/10				
Análisis	0,25 días	lun 22/03/10	lun 22/03/10				
Diseño	0,25 días	lun 22/03/10	lun 22/03/10				
Implementación	1 día	lun 22/03/10	mar 23/03/10				
Pruebas	2 días	lun 22/03/10	mar 23/03/10				
Despliegue	0,5 días	mar 23/03/10	mar 23/03/10				
Iteración C9: Partes de seguridad	3 días	mié 24/03/10	vie 26/03/10				
Iteración C10: Notas	3 días	lun 29/03/10	mié 31/03/10				
Iteración C11: Base de datos	2 días	jue 01/04/10	vie 02/04/10				
Iteración C12: Diseño, personalización y configuración	2 días	lun 05/04/10	mar 06/04/10				
Capacidad operacional	0 días	mar 06/04/10	mar 06/04/10				
Transición	8 días	mié 07/04/10	vie 16/04/10				

Recursos	Temporales		2 días
	Humanos		1 persona
	Software	Sistema operativo	Microsoft Windows 7
		Ofimática	Microsoft Word 2010 Adobe Acrobat Professional 8
		Diseño	Visual Paradigm 4.2
		Codificación	NetBeans 6.8
Hardware		1 PC	

Casos de uso y escenarios implementados	103 - Visualizar cuentas 104 - Visualizar cuenta 105 - Añadir cuenta 106 - Borrar cuenta 107 - Modificar cuenta 108 - Listar cuentas 109 - Visualizar tipos de usuario 110 - Visualizar tipo de usuario 111 - Añadir tipo de usuario 112 - Borrar tipo de usuario 113 - Modificar tipo de usuario 114 - Visualizar accesos al sistema 115 - Listar accesos al sistema 116 - Modificar datos de cuenta (primer caso de uso 'Importante')
--	--

Planificación de la iteración



3.2. Análisis de los casos de uso y escenarios

Las descripciones de trazo grueso que desarrollamos en la especificación inicial son refinadas de la siguiente forma:

Caso de uso 103: Visualizar cuentas		
Descripción	El administrador debe tener una vía para saber qué cuentas existen en el sistema y qué usuarios las poseen.	
Actores principal	Administrador.	
Pre-condiciones	El administrador se encuentra logueado en el sistema y desea consultar qué cuentas están registradas.	
Flujos	Básico	1. El usuario accede a la pestaña 'Cuentas'. 2. El sistema rescata toda la información necesaria y la muestra al usuario.
	Alternativos (*)	Ninguno.
Post-condiciones	El administrador visualiza los principales datos de las cuentas existentes en el sistema.	

Caso de uso 103: Visualizar cuentas

Caso de uso 104: Visualizar cuenta		
Descripción	Además de poder visualizar el conjunto de cuentas, puede que el administrador desee conocer todos los datos que componen una de ellas.	
Actores principal	Administrador.	
Pre-condiciones	El usuario se encuentra visualizando el conjunto de cuentas registradas en el sistema y desea conocer los datos de una de ellas.	
Flujos	Básico	1. El administrador selecciona una cuenta y pide al sistema que le muestre toda la información que posea de ésta. 2. El sistema recupera todos los datos y se los proporciona al usuario.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario dispone de todos los datos que buscaba.	

Caso de uso 104: Visualizar cuenta



Caso de uso 105: Añadir cuenta	
Descripción	La lista de usuarios que harán uso de accessManager a lo largo de toda la vida del centro será muy variable. Puede que surjan nuevos administradores, autorizadores, vigilantes,... que dejen o entren a trabajar nuevas empresas y departamentos... está claro que se hace imprescindible tener una forma de poder crear cuentas nuevas.
Actores principal	Administrador.
Pre-condiciones	Surge la necesidad de crear un nuevo usuario y el administrador se dispone a ello.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El administrador indica al sistema que desea crear una nueva cuenta. 2. El sistema le proporciona un formulario que el usuario deberá rellenar. 3. Tras editar todos los campos, el usuario le pide al sistema que realice el alta de la cuenta. 4. El sistema le contesta pidiéndole una autorización previa. 5. El usuario la otorga y así el sistema puede proceder con el alta de la nueva cuenta. (*)
	<p>Alternativos (*)</p> <p>El usuario introduce algún campo con un formato indeseado:</p> <ul style="list-style-type: none"> • El sistema le avisa del error que acaba de cometer y vuelve al punto 2 sin modificar ningún campo. <p>El usuario no da su visto bueno al registro de la cuenta:</p> <ul style="list-style-type: none"> • El sistema no realiza ninguna inserción en la base de datos y da por concluido el caso de uso.
Post-condiciones	El sistema dispone de una nueva cuenta.

Caso de uso 105: Añadir cuenta



Caso de uso 106: Borrar cuenta	
Descripción	Como dijimos en el caso de uso anterior, la lista de usuarios varía, y así como se proporciona la posibilidad de añadir cuentas, también debe existir la de eliminarlas.
Actores principal	Administrador.
Pre-condiciones	El usuario se encuentra en la pestaña de cuentas y desea eliminar alguna de ellas.
Flujos	Básico <ol style="list-style-type: none">1. El usuario selecciona las cuentas que no desea que persistan en el sistema y le pide a éste que las borre.2. El sistema le envía un mensaje al usuario avisándole de la acción que se dispone a realizar y pidiéndole que confirme si está seguro de eliminar las cuentas que ha seleccionado.3. El usuario contesta afirmativamente. (*)4. El sistema borra las cuentas seleccionadas.
	Alternativos (*) <p>El usuario contesta negativamente:</p> <ul style="list-style-type: none">• El sistema no realiza ningún borrado y el caso de uso finaliza.
Post-condiciones	El usuario consigue que las cuentas que seleccionó dejen de formar parte del sistema.

Caso de uso 106: Borrar cuenta



Caso de uso 107: Modificar cuenta	
Descripción	¿Qué pasa si se desea otorgar un tipo de usuario diferente a una cuenta para que ésta vea reducidos o aumentados sus privilegios? Parece claro que el que haya una forma de modificar el tipo de usuario (o cualquiera de sus otros datos) resulta más atractivo que tener que borrar y añadir una nueva cuenta.
Actores principal	Administrador.
Pre-condiciones	El usuario no está conforme con las características de una cuenta.
Flujos	Básico <ol style="list-style-type: none"> 1. El usuario selecciona la cuenta que desea modificar y traslada al sistema su intención. 2. Éste despliega un formulario con todos los datos de la cuenta seleccionada. 3. El usuario edita todo lo que estime oportuno y le pide al sistema que actualice el registro. 4. El sistema le pregunta si está seguro de querer llevar a cabo los cambios solicitados. 5. El usuario le indica que sí y el sistema procede. (*)
	Alternativos (*) <p>El usuario introduce algún campo erróneo:</p> <ul style="list-style-type: none"> • El sistema le informa del error que ha cometido y el caso de uso termina. <p>El usuario opta por abortar la modificación:</p> <ul style="list-style-type: none"> • Ésta no tiene lugar.
Post-condiciones	El usuario consigue que la cuenta que deseaba modificar ahora conste con los datos correctos.

Caso de uso 107: Modificar cuenta

Caso de uso 108: Listar cuentas	
Descripción	Uno de los requisitos que nos exige el cliente es que pueda tener la posibilidad de listar las cuentas existentes en el sistema.
Actores principal	Administrador.
Pre-condiciones	El administrador accede al sistema, se sitúa en el tab 'Cuentas' y se dispone a obtener un reporte con todas las que existen en estos momentos.
Flujos	Básico <ol style="list-style-type: none"> 1. El administrador pide al sistema poder disponer de un reporte en PDF con todas las cuentas de accessManager. 2. El sistema le pide al usuario la ubicación en la que almacenar el listado. 3. El usuario se la proporciona y el sistema se encarga de elaborarlo. (*)
	Alternativos (*) <p>El usuario opta por no ofrecer una ruta:</p> <ul style="list-style-type: none"> • El sistema no realiza ningún reporte y el caso de uso termina.
Post-condiciones	El administrador dispone del reporte solicitado en la ruta indicada.

Caso de uso 108: Listar cuentas



Caso de uso 109: Visualizar tipos de usuario		
Descripción	Éste y los cuatro casos de uso que le siguen no tienen tanta importancia pero resulta de utilidad desarrollarlos.	
Actores principal	Administrador.	
Pre-condiciones	El usuario desea visualizar de qué tipos de usuarios se componen las cuentas del sistema.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El administrador loguea en el sistema y se sitúa en la pestaña de cuentas. 2. El sistema accede a la base de datos para obtener y mostrar al usuario los datos que requiere.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario dispone de los datos de todos los tipos de usuario registrados en el sistema.	

Caso de uso 109: Visualizar tipos de usuario

Caso de uso 110: Visualizar tipo de usuario		
Descripción	Además de proporcionar una vía para poder visualizar todos los tipos de usuarios existentes, también hay que proporcionar otra de cara a que el administrador pueda consultar los datos de uno específico.	
Actores principal	Administrador.	
Pre-condiciones	El administrador se encuentra visualizando los tipos de usuario y quiere obtener toda la información de uno de ellos.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona el tipo de usuario a consultar. 2. El sistema recupera los datos de éste y se los proporciona al usuario en un formulario.
	Alternativos (*)	Ninguno.
Post-condiciones	El administrador visualiza todos los datos que conforman el tipo de usuario que seleccionó.	

Caso de uso 110: Visualizar tipo de usuario



Caso de uso 111: Añadir tipo de usuario	
Descripción	El caso de uso 111 nos pide que sea posible añadir un tipo de usuario a los ya existentes.
Actores principal	Administrador.
Pre-condiciones	El usuario necesita dar de alta un nuevo tipo de usuario. Abre accessManager, se loguea y se dirige a la tab de cuentas.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario indica al sistema su intención de añadir un nuevo tipo de usuario. 2. El sistema le proporciona un formulario que deberá rellenar y aprobar para que el alta tenga lugar. (*)
	<p>Alternativos (*)</p> <p>El usuario rellena el formulario con algún dato con formato no válido:</p> <ul style="list-style-type: none"> • El sistema informa al usuario de que se ha producido un error y que el alta no ha podido realizarse. • Se vuelve al final del punto 1. <p>El usuario finalmente no aprueba el alta del nuevo tipo de usuario:</p> <ul style="list-style-type: none"> • El sistema no realiza ninguna inserción y el caso de uso finaliza.
Post-condiciones	El sistema pasa a almacenar el nuevo tipo de usuario.

Caso de uso 111: Añadir tipo de usuario

Caso de uso 112: Borrar tipo de usuario	
Descripción	Pueden ser varios los motivos que puedan llevar al administrador a eliminar un tipo de usuario.
Actores principal	Administrador.
Pre-condiciones	El administrador desea eliminar algún tipo de usuario.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El administrador selecciona los tipos de usuario que no desea que sigan formando parte del sistema, y le indica a éste que los elimine. 2. El sistema lanza una advertencia al administrador avisándole de la acción que está a punto de llevarse a cabo. 3. El administrador da el visto bueno y la acción tiene lugar. (*)
	<p>Alternativos (*)</p> <p>El administrador no da el visto bueno:</p> <ul style="list-style-type: none"> • El sistema no borra ningún registro de la base de datos y da por concluido el caso de uso.
Post-condiciones	El sistema ya no dispone de los tipos de usuario eliminados.

Caso de uso 112: Borrar tipo de usuario



Caso de uso 113: Modificar tipo de usuario	
Descripción	Este caso de uso permitirá al usuario modificar cualquiera de los datos de los que consta un tipo de usuario.
Actores principal	Administrador.
Pre-condiciones	Un tipo de usuario consta con datos erróneos, el administrador se da cuenta y se dispone a corregirlo.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario selecciona el tipo de usuario que desea editar. 2. El sistema le proporciona todos los datos del tipo de usuario a través de un formulario. 3. El usuario modifica los que estima oportunos y pide al sistema que lleve a cabo la actualización. (*) 4. El sistema le pide una autorización para poder actuar. 5. El usuario se la da. (*)
	<p>Alternativos (*)</p> <p>El usuario modifica los datos con otros que tienen un formato no deseado:</p> <ul style="list-style-type: none"> • El sistema le indica el error y que no puede realizar la actualización. <p>El usuario responde negativamente ante el aviso por parte del sistema:</p> <ul style="list-style-type: none"> • Éste no realiza ninguna acción y da por concluido el caso de uso.
Post-condiciones	El administrador consigue que el tipo de usuario que constaba con datos erróneos ahora lo haga con válidos.

Caso de uso 113: Modificar tipo de usuario

Caso de uso 114: Visualizar accesos al sistema	
Descripción	El administrador del sistema debe tener la posibilidad de consultar qué usuarios han accedido a él.
Actores principal	Administrador.
Pre-condiciones	El usuario loguea en el sistema con la intención de conocer qué usuarios lo han utilizado.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El administrador selecciona la pestaña de cuentas y define la búsqueda que desea realizar. 2. El sistema se encarga de mostrarle qué usuarios han accedido al sistema según los parámetros de búsqueda proporcionados y los datos residentes en la base de datos.
	<p>Alternativos (*)</p> <p>Ninguno.</p>
Post-condiciones	El usuario visualiza los accesos al sistema que cumplen con los parámetros establecidos en la búsqueda.

Caso de uso 114: Visualizar accesos al sistema



Caso de uso 115: Listar accesos al sistema	
Descripción	El administrador puede consultar qué accesos se han producido en el sistema bajo unas condiciones dadas, pero también nos pide que pueda almacenar los resultados de esta búsqueda a través de un reporte en PDF.
Actores principal	Administrador.
Pre-condiciones	El usuario se encuentra en el sistema, y más concretamente en la tab 'Cuentas' con la intención de obtener un reporte con los datos que visualiza en la tabla de accesos al sistema.
Flujos	Básico <ol style="list-style-type: none">1. El administrador pide al sistema que genere un reporte con los datos que pueblan la tabla de accesos al sistema.2. Éste le requiere una ruta en la que depositar el listado PDF.3. El usuario le ofrece la deseada y le indica que genere el reporte. (*)
	Alternativos (*) <p>El usuario no le indica ninguna ruta al decidir que finalmente no desea obtener el reporte:</p> <ul style="list-style-type: none">• El sistema da por finalizado el caso de uso sin realizar ninguna operación.
Post-condiciones	El usuario obtiene el reporte que deseaba en la ubicación que ha proporcionado a accessManager.

Caso de uso 115: Listar accesos al sistema



Caso de uso 116: Modificar datos de cuenta					
Descripción	Las cuentas se generan con un nombre de usuario asignado por el administrador y una contraseña por defecto ("cambiar"). Esta más que claro que será necesario que los usuarios cambien al menos una vez los datos de sus cuentas, y más veces si lo estiman oportuno.				
Actores principal	Administrador, Autorizador, Vigilante y RP.				
Pre-condiciones	El usuario recibe una cuenta para usar en el sistema o ve necesario cambiar los datos que conforman alguna de las que tenga. Para ello loguea en el sistema con la cuenta que desea modificar.				
Flujos	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="background-color: #e0e0e0; width: 20%; text-align: center; vertical-align: middle;">Básico</td> <td> <ol style="list-style-type: none"> 1. El usuario indica al sistema que desea actualizar los datos de su cuenta. 2. Éste le muestra un dialogo con sus datos actuales. 3. El usuario decide que dato desea cambiar (nombre de usuario o contraseña). 4. Si el usuario elige cambiar: <ol style="list-style-type: none"> a. El nombre de usuario: <ul style="list-style-type: none"> • El sistema le proporciona un formulario en el que deberá escribir su contraseña actual y el nuevo nombre de usuario. b. La contraseña: <ul style="list-style-type: none"> • El sistema le proporciona un formulario en el que deberá escribir su contraseña actual y su nueva contraseña por dos ocasiones. 5. Si los datos actuales son válidos, el sistema avisa al usuario que va a proceder a actualizar los datos indicados, pidiéndole una aprobación previa. (*) 6. El usuario da su aprobación. (*) 7. El sistema recoge los datos y realiza la modificación. (*) </td> </tr> <tr> <td style="background-color: #e0e0e0; text-align: center; vertical-align: middle;">Alternativos (*)</td> <td> <p>El usuario termina por no dar su aprobación:</p> <ul style="list-style-type: none"> • El sistema da por concluido en este punto el caso de uso. <p>La contraseña introducida no es válida o el usuario introduce un nombre de usuario existente o dos contraseñas distintas o un nombre de usuario o contraseña con un formato indeseado:</p> <ul style="list-style-type: none"> • El sistema avisa del error y vuelve al punto 2, sin modificar ningún dato. </td> </tr> </table>	Básico	<ol style="list-style-type: none"> 1. El usuario indica al sistema que desea actualizar los datos de su cuenta. 2. Éste le muestra un dialogo con sus datos actuales. 3. El usuario decide que dato desea cambiar (nombre de usuario o contraseña). 4. Si el usuario elige cambiar: <ol style="list-style-type: none"> a. El nombre de usuario: <ul style="list-style-type: none"> • El sistema le proporciona un formulario en el que deberá escribir su contraseña actual y el nuevo nombre de usuario. b. La contraseña: <ul style="list-style-type: none"> • El sistema le proporciona un formulario en el que deberá escribir su contraseña actual y su nueva contraseña por dos ocasiones. 5. Si los datos actuales son válidos, el sistema avisa al usuario que va a proceder a actualizar los datos indicados, pidiéndole una aprobación previa. (*) 6. El usuario da su aprobación. (*) 7. El sistema recoge los datos y realiza la modificación. (*) 	Alternativos (*)	<p>El usuario termina por no dar su aprobación:</p> <ul style="list-style-type: none"> • El sistema da por concluido en este punto el caso de uso. <p>La contraseña introducida no es válida o el usuario introduce un nombre de usuario existente o dos contraseñas distintas o un nombre de usuario o contraseña con un formato indeseado:</p> <ul style="list-style-type: none"> • El sistema avisa del error y vuelve al punto 2, sin modificar ningún dato.
Básico	<ol style="list-style-type: none"> 1. El usuario indica al sistema que desea actualizar los datos de su cuenta. 2. Éste le muestra un dialogo con sus datos actuales. 3. El usuario decide que dato desea cambiar (nombre de usuario o contraseña). 4. Si el usuario elige cambiar: <ol style="list-style-type: none"> a. El nombre de usuario: <ul style="list-style-type: none"> • El sistema le proporciona un formulario en el que deberá escribir su contraseña actual y el nuevo nombre de usuario. b. La contraseña: <ul style="list-style-type: none"> • El sistema le proporciona un formulario en el que deberá escribir su contraseña actual y su nueva contraseña por dos ocasiones. 5. Si los datos actuales son válidos, el sistema avisa al usuario que va a proceder a actualizar los datos indicados, pidiéndole una aprobación previa. (*) 6. El usuario da su aprobación. (*) 7. El sistema recoge los datos y realiza la modificación. (*) 				
Alternativos (*)	<p>El usuario termina por no dar su aprobación:</p> <ul style="list-style-type: none"> • El sistema da por concluido en este punto el caso de uso. <p>La contraseña introducida no es válida o el usuario introduce un nombre de usuario existente o dos contraseñas distintas o un nombre de usuario o contraseña con un formato indeseado:</p> <ul style="list-style-type: none"> • El sistema avisa del error y vuelve al punto 2, sin modificar ningún dato. 				
Post-condiciones	El usuario dispone de su cuenta actualizada con los datos que desea.				

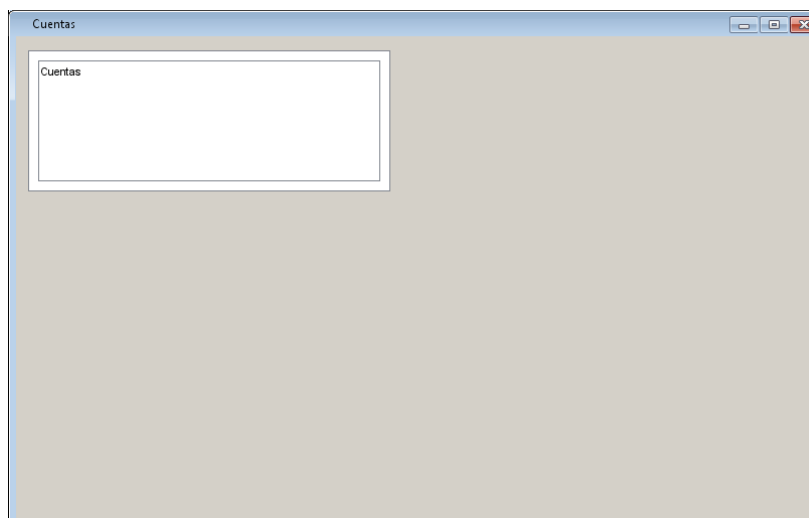
Caso de uso 116: Modificar datos de cuenta



4. Diseño

4.1. Casos de uso 103 y 104: Visualizar cuentas y visualizar cuenta

Iniciamos la iteración y con ella el diseño de una nueva pestaña dentro de accessManager. Nos centramos en primer lugar en visualizar todas las cuentas y los datos que las conforman. Si atendemos al modelo de datos nos daremos cuenta que todas están formadas por cuatro atributos, con que puede ser buena idea agrupar los dos primeros casos de uso en una tabla que muestre todos los datos de todas las cuentas existentes:



Interfaz gráfica propuesta

Para rellenarla debemos acudir a la tabla Cuenta de la base de datos. Éste es el proceso que seguiremos:

1. Crearemos un objeto de tipo Cuenta, a través del cual realizar la obtención de todos los datos.
2. Rellenaremos la tabla con los resultados obtenidos en la anterior consulta.

Visto lo visto, tenemos que crear un método que obtenga todos los datos de todas las cuentas:

MÉTODO	FUNCIONALIDAD
<i>obtenerCuentas</i>	Método que realiza la búsqueda de todas las cuentas existentes en el sistema, devolviendo la totalidad de la tabla Cuenta.

Método a implementar y a añadir a la API



Resulta un proceso trivial, pero nunca está de más visualizarlo a través de un diagrama de actividades y otro de secuencia:

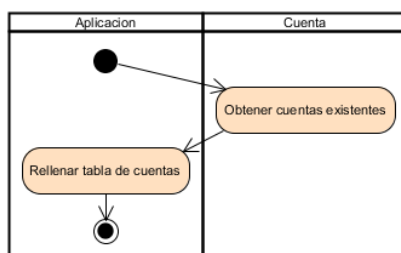


Diagrama de actividades

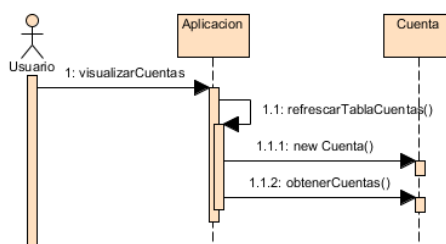
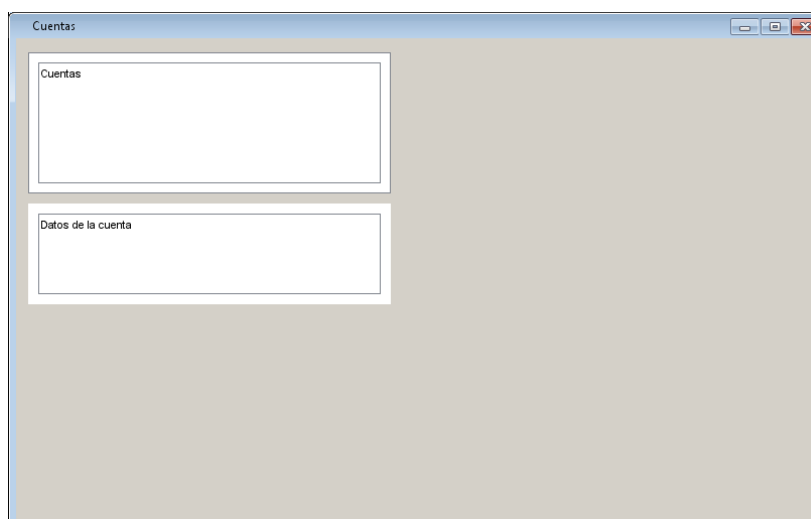


Diagrama de secuencia

4.2. Caso de uso 105: Añadir cuenta

Ahora sí que necesitaremos crear el formulario que nos permita proporcionar al sistema los datos necesarios para dar de alta cuentas. Es natural seguir la misma lógica que hemos venido utilizando hasta ahora y crear un JPanel debajo de la tabla de muestra de cuentas donde introducir los datos necesarios:



Interfaz gráfica propuesta



Crear una nueva cuenta será tan fácil como:

1. Crear un objeto de tipo Cuenta con los datos recogidos del formulario.
2. Insertar la cuenta en la base de datos.
3. Refrescar la tabla de cuentas para que aparezca en ella el nuevo registro.

Tenemos a nuestra disposición todo lo que necesitamos (el constructor Cuenta, el método de inserción y el de refresco), con que no necesitamos añadir nada a la API para conseguir completar este proceso:

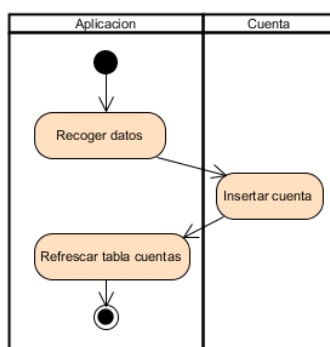


Diagrama de actividades

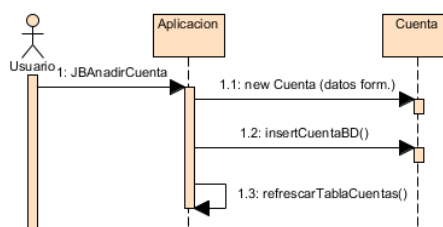


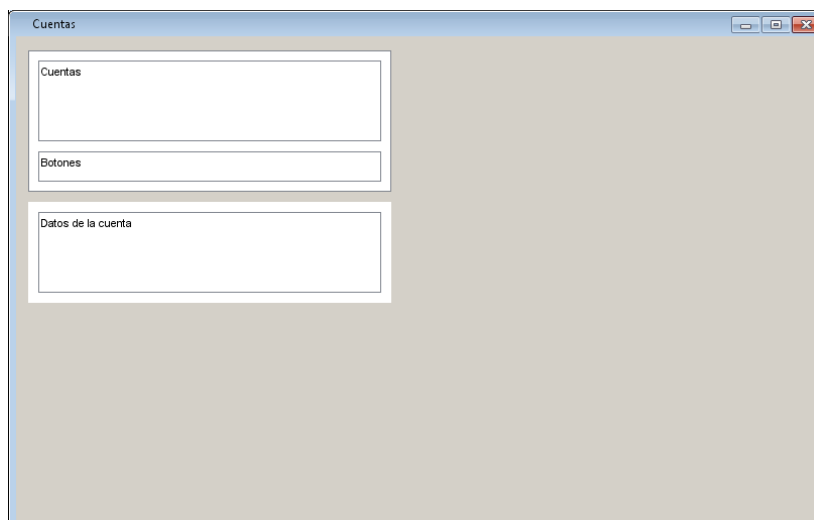
Diagrama de secuencia

Nota: Para facilitar la selección de la persona a la que asignaremos la cuenta, añadimos en el mismo panel de datos un JTextField que nos permita realizar una búsqueda sobre todo el personal registrado en el sistema que podría llegar a disponer de una cuenta.



4.3. Caso de uso 106: Borrar cuenta

Para borrar cuentas podemos basarnos en los casos de uso de borrado ya realizados, en los que solo nos hacía falta un botón...



Interfaz gráfica propuesta

... y seguir esta secuencia de actividades:

1. Obtener las cuentas seleccionadas.
2. Recorrer las cuentas seleccionadas eliminándolas una a una.
3. Refrescar las tablas de cuentas y tipos de usuario (una vez dispongamos de ella).

Para mejorar la eficiencia del borrado, en lugar de crear continuamente las cuentas a través de un constructor que las recupere, lo que haremos será crear un objeto Cuenta en la que actualizaremos continuamente su nombre de usuario (clave primaria de la tabla) y llamaremos a su método de borrado.

Todo este proceso queda reflejado en estos dos diagramas:

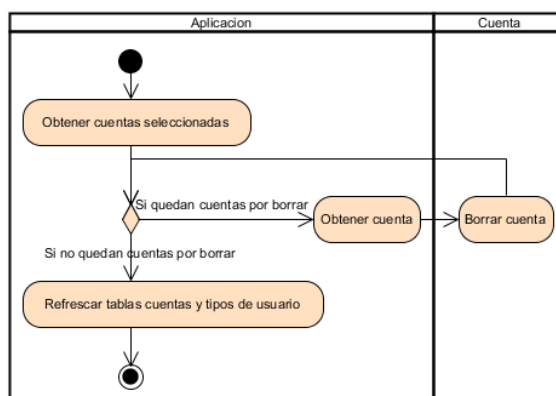


Diagrama de actividades

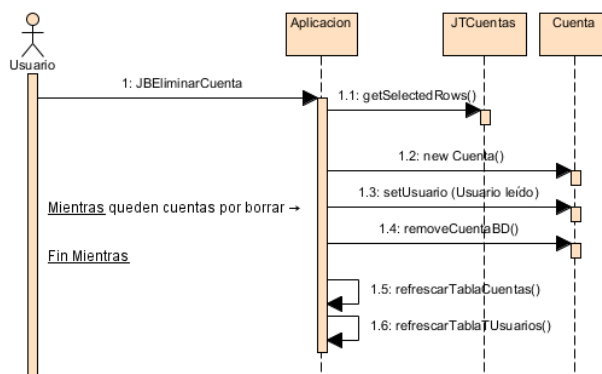


Diagrama de secuencia

4.4. Caso de uso 107: Modificar cuenta

Si nos basamos en la interfaz gráfica que disponemos y añadimos otro botón más al JPanel de botones que tenemos bajo la tabla de cuentas, podremos modificar una cuenta de la siguiente forma:

1. Crearemos una cuenta nueva con la información que recojamos del formulario.
2. Recuperaremos la cuenta a actualizar de la base de datos.
3. Actualizaremos la cuenta vieja con la nueva.
4. Refrescaremos la tabla de cuentas para que refleje la cuenta ya modificada.

Aquí tampoco surge ningún método nuevo que añadir la API. Si nos fijamos, los únicos métodos que necesitaremos son los de crear una cuenta a partir de su nombre de usuario (creado desde el caso 1), el de actualizar una cuenta (ya en la API inicial) y el de refrescar la tabla de cuentas (disponemos de él desde el inicio de la iteración).

Con todo lo necesario deberíamos ser capaces de codificar estos diagramas:

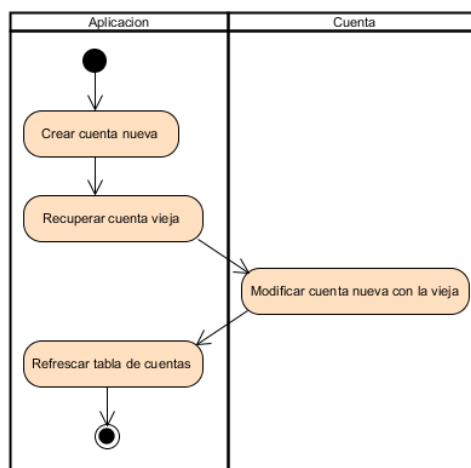


Diagrama de actividades

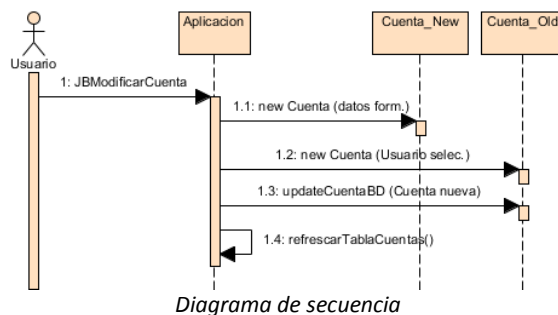


Diagrama de secuencia

4.5. Caso de uso 108: Listar cuentas

Este caso de uso resulta análogo al de listar accesos: basta con añadir un botón que inicie todo el proceso y ayudarnos de un JFileChooser y la clase GeneradorInformes. Recordemos la forma de proceder:

1. Desplegar un JFileChooser para recoger el destino del informe.
2. Crear un objeto de tipo GeneradorInformes.
3. En lugar de llamar a *generarInforme("ACCESOS", "Accesos realizados:", JTAccesos)*, llamar a *generarInforme("CUENTAS", "Cuentas existentes en el sistema:", JTCuentas)*.

Por lo tanto, nada que añadir a la API; basta con seguir el diagrama de actividades...

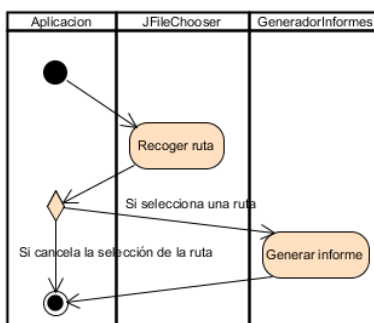


Diagrama de actividades

... y el de secuencia:

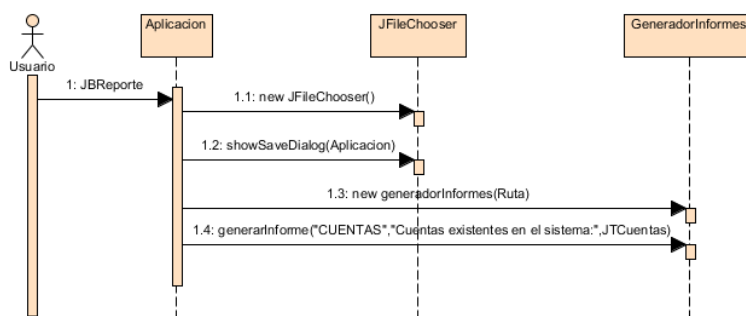
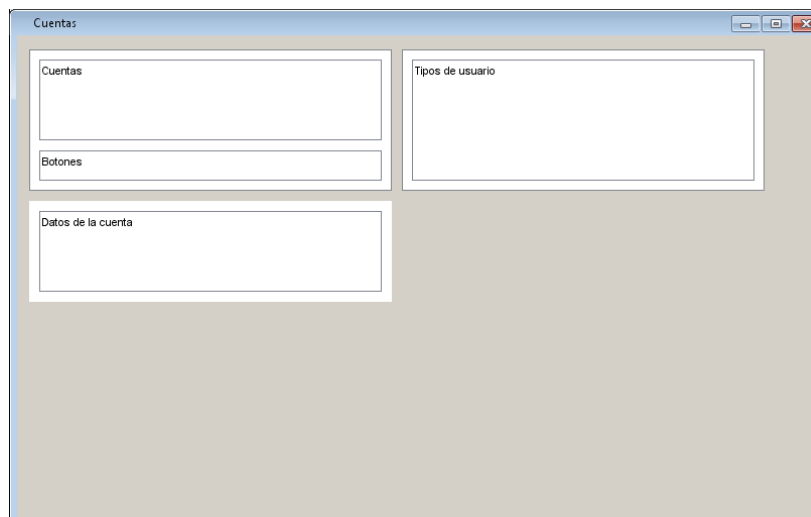


Diagrama de secuencia



4.6. Casos de uso 109 y 110: Visualizar tipos de usuario y visualizar tipo de usuario

Nos encontramos en una situación muy similar a la vivida con los casos de uso 103 y 104. El tipo de usuario está compuesto por dos atributos, 'Tipo' y 'Descripcion', con que podemos utilizar con total tranquilidad una única tabla para mostrar todos los datos de todos los tipos de usuario. La colocaremos a la derecha de la de cuentas, como mostramos a continuación:



Interfaz gráfica propuesta

Rellenarla es tan fácil como:

1. Crear un objeto T_Usuario.
2. Usar este objeto para realizar la búsqueda de todos los tipos de usuario.
3. Introducir en la tabla de tipos de usuario el resultado obtenido en la búsqueda anterior.

Necesitaremos añadir pues este método a la API:

MÉTODO	FUNCIONALIDAD
<i>obtenerTUsuarios</i>	Método que obtiene todos los datos de todos los tipos de usuarios existentes en el sistema.

Método a implementar y a añadir a la API



A pesar de que los diagramas de actividades y de secuencia son idénticos a los de las cuentas, nunca viene mal recordarlos:

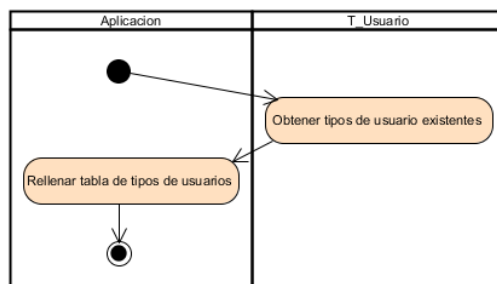


Diagrama de actividades

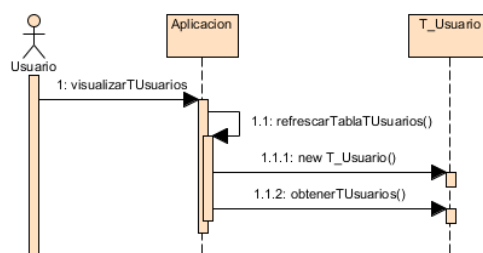
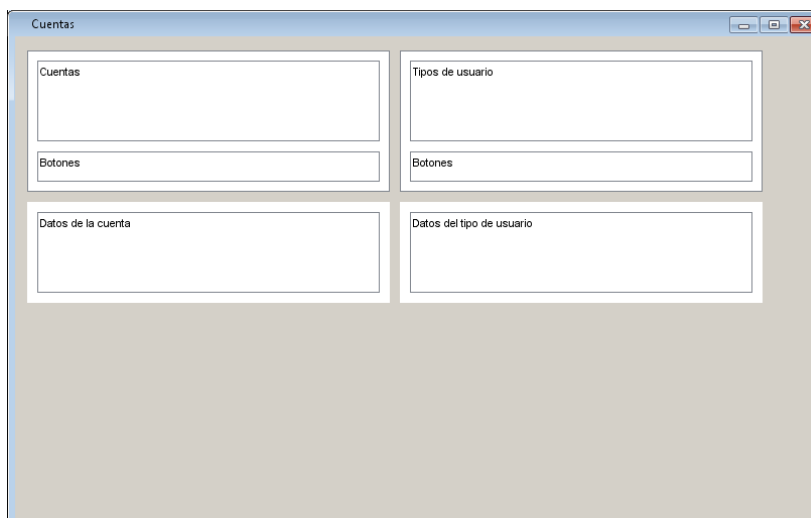


Diagrama de secuencia

4.7. Caso de uso 111: Añadir tipo de usuario

Este caso de uso ya nos obliga a tener que incorporar dos elementos gráficos al panel de cuentas: un panel que albergará los botones para operar con los tipos de usuario (de momento solo poseerá el botón de 'Añadir tipo de usuario') y un panel que actúe como formulario para poder mandar y recibir información de la aplicación:



Interfaz gráfica propuesta



¿Cómo se crea un nuevo tipo de usuario? He aquí la solución:

1. Crear un objeto T_Usuario con todos los datos que constan en el formulario de entrada.
2. Insertar el objeto en la base de datos.
3. Refrescar la tabla de tipos de usuario para que aparezca en ella el nuevo tipo de usuario.

Fácil, ¿verdad? Aún lo es más si visualizamos todo el flujo de actividades y mensajes:

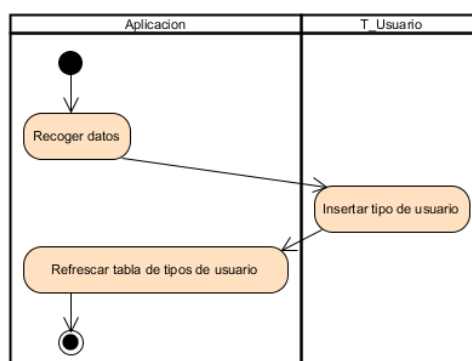


Diagrama de actividades

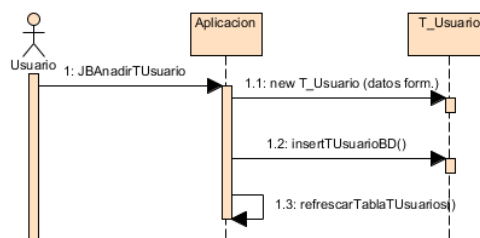


Diagrama de secuencia

4.8. Caso de uso 112: Borrar tipo de usuario

El borrado de tipos de usuario seguirá el mismo proceso que el de cuentas. Para ello, necesitamos añadir un botón de borrado al panel, para que así el usuario pueda seleccionar los tipos de usuario a borrar y con clicarlo se desate esta secuencia de acciones:

1. Obtener tipos de usuario seleccionados.
2. Recorrerlos y eliminarlos uno a uno.
3. Refrescar las tablas del tab 'Cuentas' para que todos los cambios (borrados) se hagan visibles.



Estos son los diagramas ideados para la ocasión:

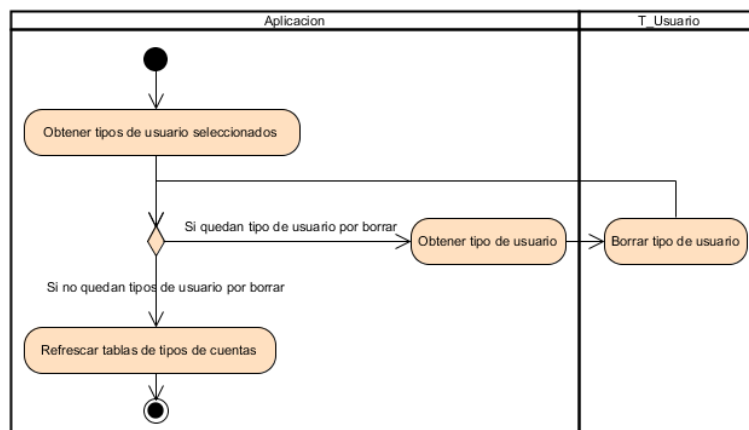


Diagrama de actividades

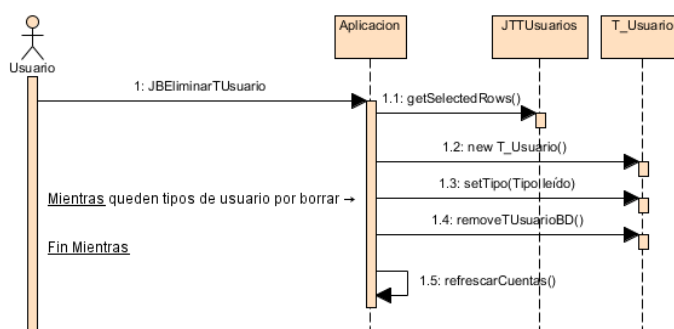


Diagrama de secuencia

4.9. Caso de uso 113: Modificar tipo de usuario

Damos por terminado el grupo de casos de uso destinado a los tipos de usuario diseñando el proceso de modificación:

1. Crear un tipo de usuario con los datos recogidos del formulario.
2. Recuperar el tipo de usuario a actualizar de la base de datos.
3. Actualizar el tipo de usuario viejo con el nuevo.
4. Refrescar las tablas de la pestaña de cuentas para que la modificación del tipo de usuario se refleje en todas ellas.

Ni que decir tiene que este proceso será lanzado al pulsar el botón de actualización que debemos añadir al panel de botones.



Echemos un vistazo a los diagramas que nos ayudarán en la fase de codificación:

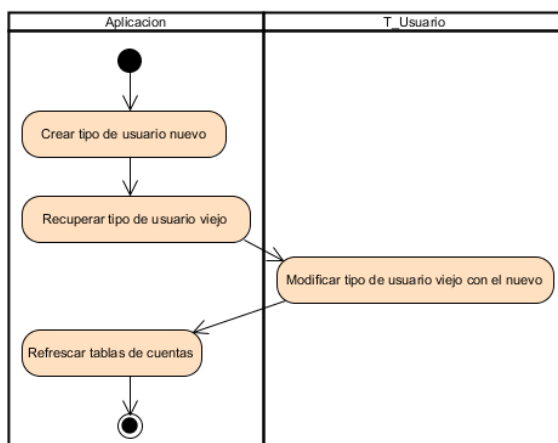


Diagrama de actividades

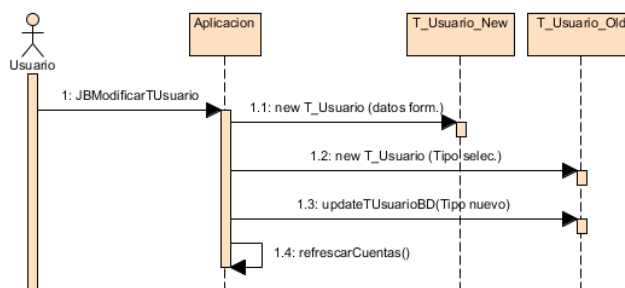


Diagrama de secuencia

4.10. Caso de uso 114: Visualizar accesos al sistema

Completamos la pestaña con una sección dedicada a los accesos que los diversos usuarios de la aplicación realizan a ésta.

En primer lugar cabe tratar la visualización de los mismos, que podrá realizarse a través de una JTable, tal y como venimos mostrando el resto de los componentes del sistema. Es una tabla muy cambiante, que registrará nuevas inserciones con mucha frecuencia, lo que nos obliga a pensar en la idea de crear un buscador que permita filtrar las tuplas que pueblan la tabla 'Acceso_Sistema'.



Una vez analizado qué debe mostrar la tabla podemos realizar un sencillo esbozo de cómo quedará la interfaz de la pestaña 'Cuentas':

La interfaz gráfica propuesta para la pestaña 'Cuentas' se muestra en un recuadro con el título 'Cuentas'. El diseño está dividido en varias secciones:

- Una sección superior izquierda con el título 'Cuentas' y un campo de texto debajo.
- Una sección superior derecha con el título 'Tipos de usuario' y un campo de texto debajo.
- Una sección inferior izquierda con el título 'Datos de la cuenta' y un campo de texto debajo.
- Una sección inferior derecha con el título 'Datos del tipo de usuario' y un campo de texto debajo.
- Una sección inferior central con el título 'Parámetros para restringir la búsqueda' y un campo de texto debajo.
- Una sección inferior amplia con el título 'Accesos al sistema' y un campo de texto debajo.
- En cada una de las secciones superiores, hay un botón etiquetado como 'Botones'.

Interfaz gráfica propuesta

Para rellenarla podemos seguir este sencillo proceso:

1. Crear un objeto de tipo Acceso_Sistema.
2. Llamar a *obtenerAccesosSistema* pasándole todos los parámetros que entran en juego a la hora de realizar (delimitar) la búsqueda.
3. Rellenar la tabla de accesos al sistema con los resultados obtenidos en el punto anterior.

De nuevo nos toca crear y añadir a la API un método que realice una operación de tipo SELECT sobre una tabla:

MÉTODO	FUNCIONALIDAD
<i>obtenerAccesosSistema</i>	Método que obtiene todos los accesos realizados al sistema que cumplen con los parámetros pasados como entrada.

Método a implementar y a añadir a la API



La mecánica a seguir es la misma que para cualquiera de los casos de uso de búsqueda que hemos venido implementando a lo largo de todo el desarrollo software:

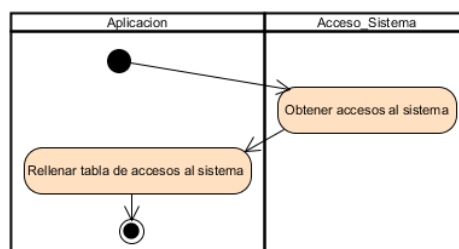


Diagrama de actividades

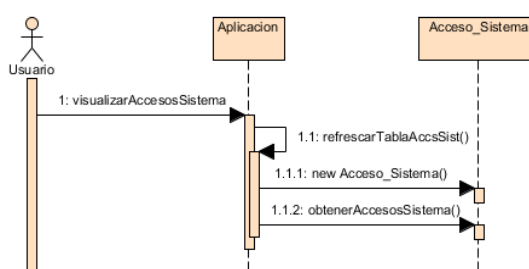


Diagrama de secuencia

4.11. Caso de uso 115: Listar accesos al sistema

Listar los accesos que se han realizado al sistema es calcado a listar las cuentas existentes, o los accesos realizados al centro,... con que nos limitaremos a especificar la estrategia y a acompañarla de los correspondientes diagramas.

1. Desplegar un JFileChooser para que el usuario indique en que ubicación desea recibir el reporte.
2. Crear un objeto de tipo GeneradorInformes.
3. Llamar al método `generarInforme("ACCESOS AL SISTEMA", "Historial de accesos al sistema:", JTHistorialAccesosSist)`.

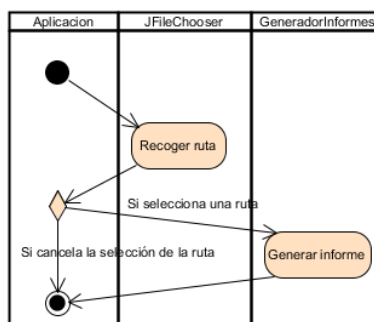


Diagrama de actividades

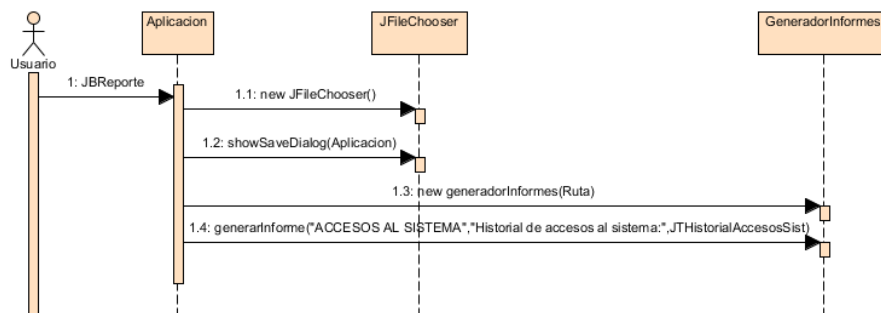
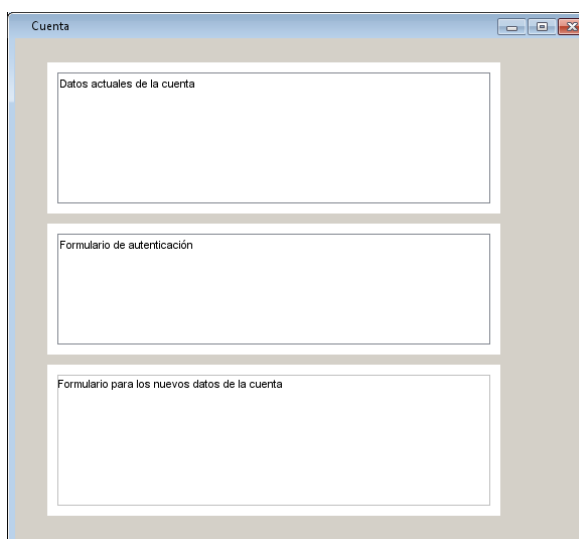


Diagrama de secuencia

4.12. Caso de uso 116: Modificar datos de cuenta

Comenzamos la implementación de los casos de uso de carácter 'Importante' con una funcionalidad no relacionada directamente con la gestión de accesos y que debe poder ejecutar cualquier tipo de usuario: modificar los datos de la cuenta. Para ofrecerla, resulta muy correcto llevarla al menú de accessManager y ejecutarla en un JDialog, ya que se trata de un proceso que es ejecutado por todos los tipos de usuario y con poca frecuencia como para dedicarle sitio en el frame principal:



Interfaz gráfica propuesta

El estado inicial del diálogo es el de muestra del primer panel, desplegándose los otros dos una vez el usuario solicite variar alguno de los datos de su cuenta.



Este debe ser el proceso a seguir:

1. Recuperar la cuenta vieja de la base de datos.
2. Recoger la contraseña y ver si casa con la de la cuenta vieja.
 - a. Si casa:
 - Creamos la cuenta nueva idéntica a la vieja y...
 - a. ... si lo que se modifica es el nombre de usuario
 1. Se modifica el nombre de usuario de la cuenta nueva.
 - b. ... si lo que se modifica es la contraseña
 1. Comprobamos que ambas contraseñas nuevas coinciden.
 2. Se modifica la contraseña de la cuenta nueva.
 - Actualizamos la cuenta nueva con la vieja.
 - Refrescamos el panel de muestra de datos de la cuenta.

Veamos los diagramas de actividades y de secuencia para aclarar todo este flujo de actividades:

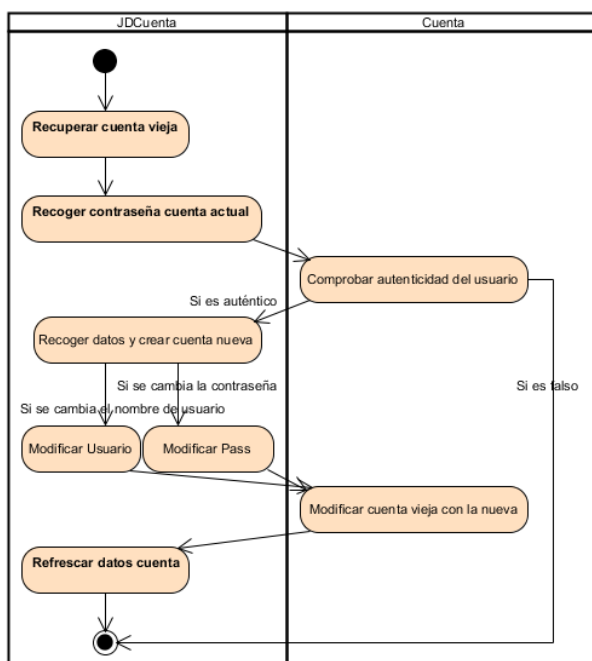


Diagrama de actividades

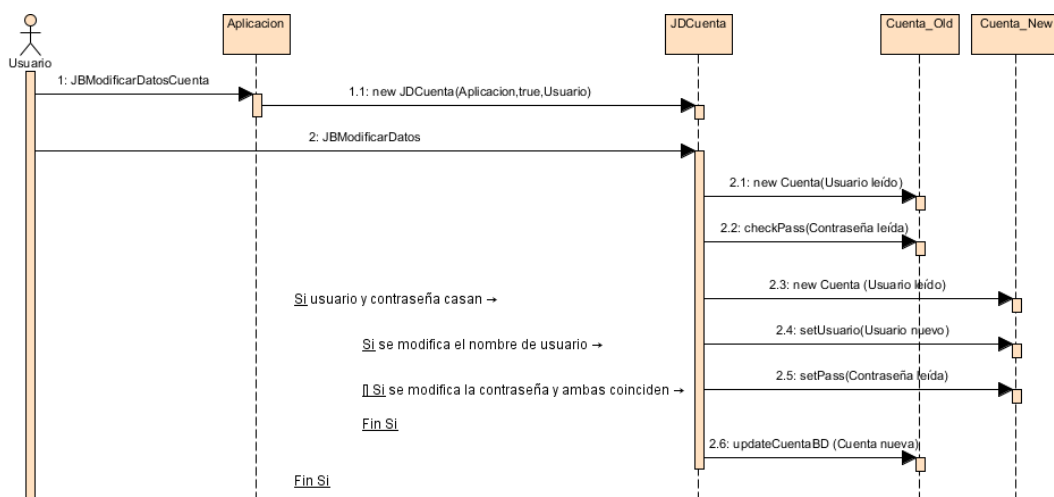


Diagrama de secuencia

5. Codificación e integración

La verdad es que poca cosa cabe decir con respecto a la codificación de esta iteración, ya que los casos de uso no distan apenas de los ya realizados en iteraciones anteriores y todos ellos siguen siendo escritos con ayuda de la API en la clase que representa la interfaz principal de accessManager (*Aplicación.java*).

Esta es la vista que poseen en este momento la interfaces de 'Cuentas' y 'Cuenta':

The screenshot shows the 'accessManager' application window. The 'Cuentas' tab is active, displaying a table of accounts and a 'Historial de accesos al sistema' section.

DNI	Tipo	Usuario	Contraseña
00000000A	AUTORIZADOR	a	*****
00000000A	ADMINISTRADOR	hnapalb	*****
00000001A	VIGILANTE	rsalinas	*****
00000002A	VIGILANTE	ghaydeed	*****
00000003A	VIGILANTE	jcasados	*****
000000018A	AUTORIZADOR	mgamiz	*****
00000028A	AUTORIZADOR	lcarmona	*****
00000043A	AUTORIZADOR	tcavido	*****
00000163A	RP	jgarciap	*****
00000168A	RP	ledominquez	*****

Tipo	Descripcion
ADMINISTRADOR	Administrador del sistema
AUTORIZADOR	Autorizador de departamentos
RP	Responsable de personal de ...
VIGILANTE	Vigilante del centro

Historial de accesos al sistema

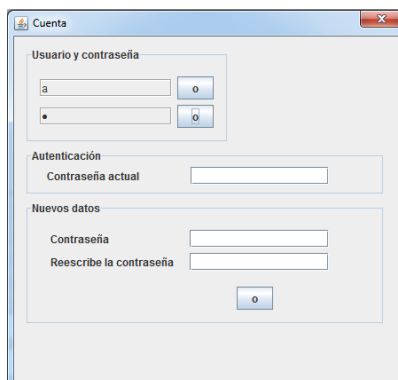
Buscar: []

lunes 24 de mayo de 2010 00:00 [x] Todo el día

lunes 24 de mayo de 2010 00:00 [PDF]

Usuario	Tipo	DNI	Fecha
a	AUTORIZADOR	00000000A	2010-05-24 08:47:04.0
a	AUTORIZADOR	00000000A	2010-05-24 08:47:31.0
a	AUTORIZADOR	00000000A	2010-05-24 09:03:30.0
a	AUTORIZADOR	00000000A	2010-05-24 09:17:05.0
a	AUTORIZADOR	00000000A	2010-05-24 09:57:41.0
a	AUTORIZADOR	00000000A	2010-05-24 10:02:26.0
a	AUTORIZADOR	00000000A	2010-05-24 10:02:58.0
a	AUTORIZADOR	00000000A	2010-05-24 10:15:59.0

Interfaz de la pestaña 'Cuentas'



Interfaz del diálogo 'Cuenta'

Sin más preámbulos pasemos a realizar las pruebas que aseguren que todo el trabajo desarrollado en la iteración ha desembocado en un software robusto, eficaz y de calidad.

6. Pruebas

6.1. De unidad

Estas son las pruebas diseñadas a lo largo de la iteración y que demuestran que los componentes software desarrollados responden tal y como se espera de ellos:

Caso de prueba	8.01 - Casos de uso 103, 104, 109, 110 y 114: Visualizaciones																																																																																																				
Objetivo	Probar que al acceder a la pestaña 'Cuentas' las tablas que visualiza el usuario reflejan la realidad.																																																																																																				
Actividades	Loguear en el sistema e ir a la pestaña de cuentas, comprobando que lo que ésta muestra es lo que en verdad reside en la base de datos.																																																																																																				
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. • Utilidad phpMyAdmin. 																																																																																																				
Datos de entrada	La población de prueba residente en la base de datos.																																																																																																				
Salida esperada	<p>Las tablas que componen la tab 'Cuentas' deberían mostrar las siguientes tuplas:</p> <table border="1"> <thead> <tr> <th>DNI</th> <th>DNI de la persona a la que pertenece la cuenta</th> <th>Tipo</th> <th>Tipo de usuario</th> <th>Usuario</th> <th>Nombre de usuario</th> <th>Pass</th> <th>Contraseña</th> </tr> </thead> <tbody> <tr> <td>00000000A</td> <td></td> <td>AUTORIZADOR</td> <td></td> <td>a</td> <td></td> <td>a</td> <td></td> </tr> <tr> <td>00000000A</td> <td></td> <td>ADMINISTRADOR</td> <td></td> <td>hnapalb</td> <td></td> <td>cambiar</td> <td></td> </tr> <tr> <td>00000001A</td> <td></td> <td>VIGILANTE</td> <td></td> <td>rsalinash</td> <td></td> <td>cambiar</td> <td></td> </tr> <tr> <td>00000002A</td> <td></td> <td>VIGILANTE</td> <td></td> <td>ghaydeed</td> <td></td> <td>cambiar</td> <td></td> </tr> <tr> <td>00000003A</td> <td></td> <td>VIGILANTE</td> <td></td> <td>jcasados</td> <td></td> <td>cambiar</td> <td></td> </tr> <tr> <td>00000018A</td> <td></td> <td>AUTORIZADOR</td> <td></td> <td>mgamiz</td> <td></td> <td>cambiar</td> <td></td> </tr> <tr> <td>00000028A</td> <td></td> <td>AUTORIZADOR</td> <td></td> <td>lcarmona</td> <td></td> <td>cambiar</td> <td></td> </tr> <tr> <td>00000043A</td> <td></td> <td>AUTORIZADOR</td> <td></td> <td>tcavido</td> <td></td> <td>cambiar</td> <td></td> </tr> <tr> <td colspan="8" style="text-align: center;">...</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Tipo</th> <th>Tipo de usuario</th> <th>Descripción</th> <th>Descripción del tipo de usuario</th> </tr> </thead> <tbody> <tr> <td>ADMINISTRADOR</td> <td></td> <td>Administrador del sistema</td> <td></td> </tr> <tr> <td>AUTORIZADOR</td> <td></td> <td>Autorizador de departamentos</td> <td></td> </tr> <tr> <td>RP</td> <td></td> <td>Responsable de personal de departamento(s)</td> <td></td> </tr> <tr> <td>VIGILANTE</td> <td></td> <td>Vigilante del centro</td> <td></td> </tr> </tbody> </table>	DNI	DNI de la persona a la que pertenece la cuenta	Tipo	Tipo de usuario	Usuario	Nombre de usuario	Pass	Contraseña	00000000A		AUTORIZADOR		a		a		00000000A		ADMINISTRADOR		hnapalb		cambiar		00000001A		VIGILANTE		rsalinash		cambiar		00000002A		VIGILANTE		ghaydeed		cambiar		00000003A		VIGILANTE		jcasados		cambiar		00000018A		AUTORIZADOR		mgamiz		cambiar		00000028A		AUTORIZADOR		lcarmona		cambiar		00000043A		AUTORIZADOR		tcavido		cambiar		...								Tipo	Tipo de usuario	Descripción	Descripción del tipo de usuario	ADMINISTRADOR		Administrador del sistema		AUTORIZADOR		Autorizador de departamentos		RP		Responsable de personal de departamento(s)		VIGILANTE		Vigilante del centro	
DNI	DNI de la persona a la que pertenece la cuenta	Tipo	Tipo de usuario	Usuario	Nombre de usuario	Pass	Contraseña																																																																																														
00000000A		AUTORIZADOR		a		a																																																																																															
00000000A		ADMINISTRADOR		hnapalb		cambiar																																																																																															
00000001A		VIGILANTE		rsalinash		cambiar																																																																																															
00000002A		VIGILANTE		ghaydeed		cambiar																																																																																															
00000003A		VIGILANTE		jcasados		cambiar																																																																																															
00000018A		AUTORIZADOR		mgamiz		cambiar																																																																																															
00000028A		AUTORIZADOR		lcarmona		cambiar																																																																																															
00000043A		AUTORIZADOR		tcavido		cambiar																																																																																															
...																																																																																																					
Tipo	Tipo de usuario	Descripción	Descripción del tipo de usuario																																																																																																		
ADMINISTRADOR		Administrador del sistema																																																																																																			
AUTORIZADOR		Autorizador de departamentos																																																																																																			
RP		Responsable de personal de departamento(s)																																																																																																			
VIGILANTE		Vigilante del centro																																																																																																			



Usuario	Nombre de usuario	Fecha	Fecha y hora de entrada en el sistema
a		2010-05-24	08:47:04
a		2010-05-24	08:47:31
a		2010-05-24	09:03:30
a		2010-05-24	09:17:05
a		2010-05-24	09:57:41
a		2010-05-24	10:02:26
a		2010-05-24	10:02:58
a		2010-05-24	10:15:59
a		2010-05-24	10:34:47
a		2010-05-24	11:19:56
a		2010-05-24	11:20:57
a		2010-05-24	14:48:39
a		2010-05-24	16:19:06
a		2010-05-24	19:04:17
a		2010-05-24	19:04:40
a		2010-05-24	20:52:20

La espera:

Salida observada

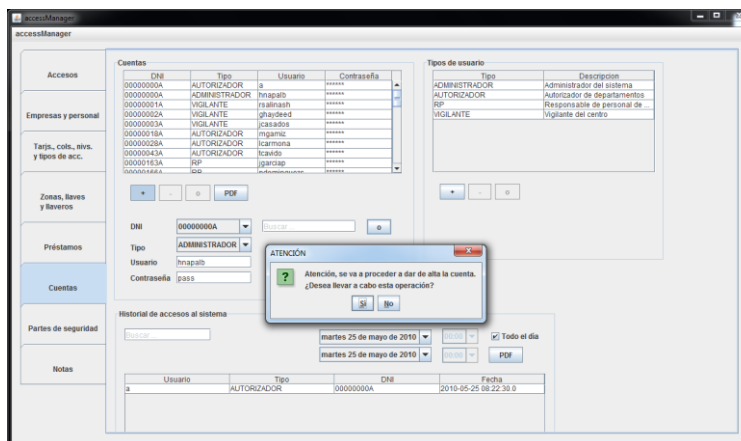
The screenshot shows the 'accessManager' application interface. It features a sidebar with navigation options like 'Accesos', 'Empresas y personal', 'Tarjetas, coils, naves y tipos de acc.', 'Zonas, bares y Biveros', 'Préstamos', 'Cuentas', 'Partes de seguridad', and 'Notas'. The main area is divided into several sections: 'Cuentas' (Accounts) with a table of user details, 'Tipos de usuario' (User types) with a table of roles, and 'Historial de accesos al sistema' (System access history) with a table of login events. The interface includes search filters, date pickers, and a 'PDF' export button.

Evaluación/ conclusión **Los casos de uso '103, 104, 109, 110 y 114 - Visualizaciones' han sido realizados satisfactoriamente.**

Caso de prueba para los casos de uso '103, 104, 109, 110 y 114 - Visualizaciones'

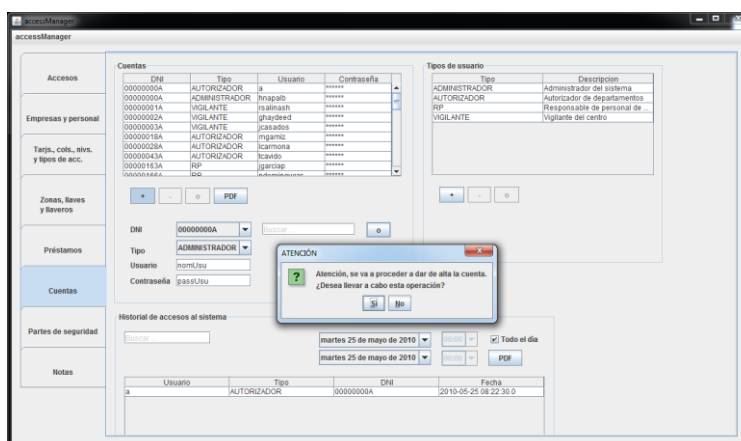
Caso de prueba	8.02 - Caso de uso 105: Añadir cuenta
Objetivo	Comprobar que el administrador podrá añadir al sistema las cuentas que desee siempre y cuando introduzca datos válidos.
Actividades	Acceder a la sección de cuentas e intentar la inserción de varias cuentas, observando qué resultados se producen.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba.

- Las cuentas a insertar son:
- 'hnapalb,pass,ADMINISTRADOR,00000000A'. (Usuario existente)



Datos de entrada

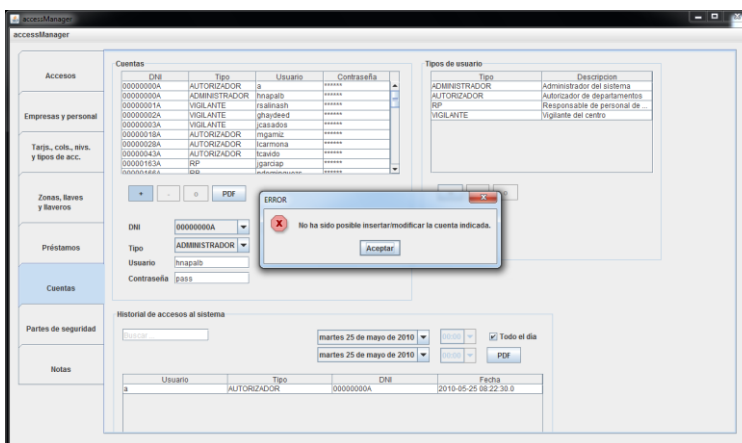
- 'usuariodemadiadolargoparaentraqui, pass,ADMINISTRADOR,00000000A'. (Formato erróneo)
- 'nomUsu,passUsu,ADMINISTRADOR,00000000A'. (Usuario válido)



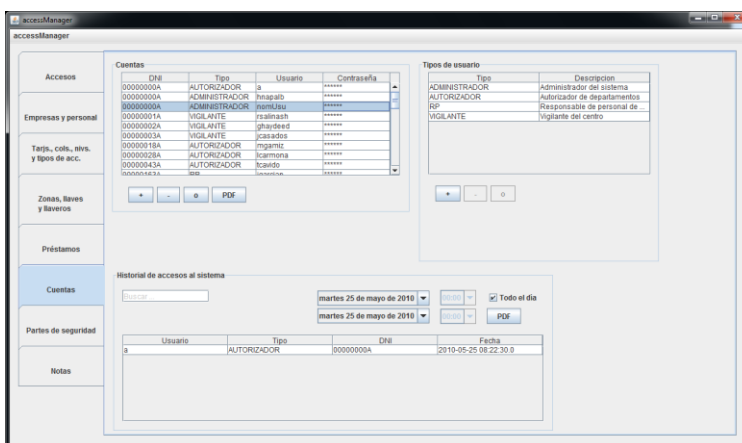
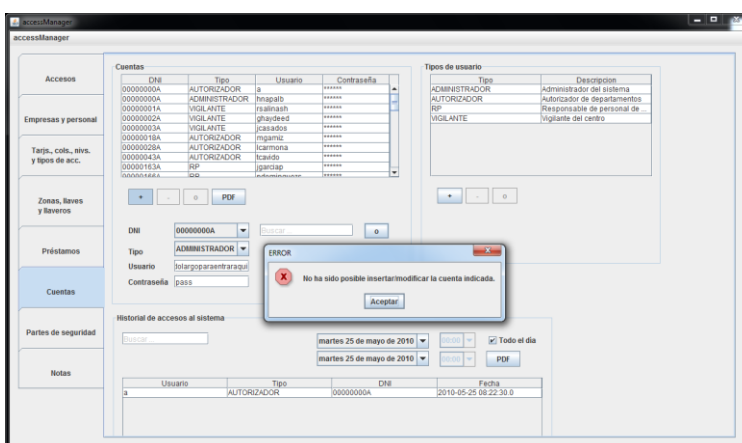
Además, nos cercioraremos de que el personal ocasional nunca podría disponer de una cuenta.

Salida esperada	Los dos primeros intentos deben dar como resultado un aviso de error y ninguna inserción, mientras que el tercero sí que dará de alta una nueva cuenta en el sistema.
------------------------	---

La espera:



Salida observada



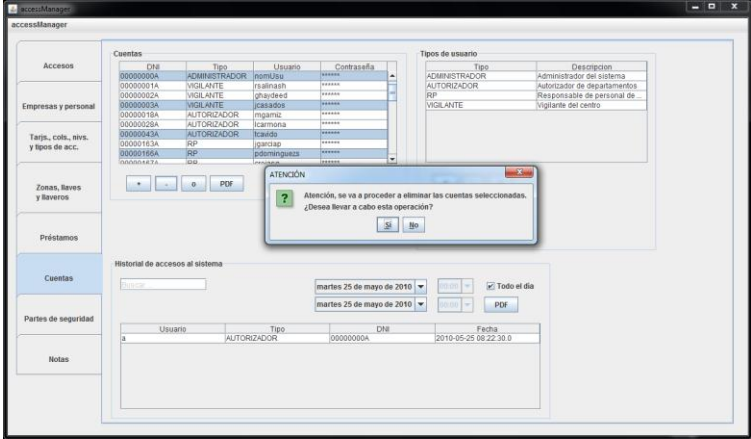
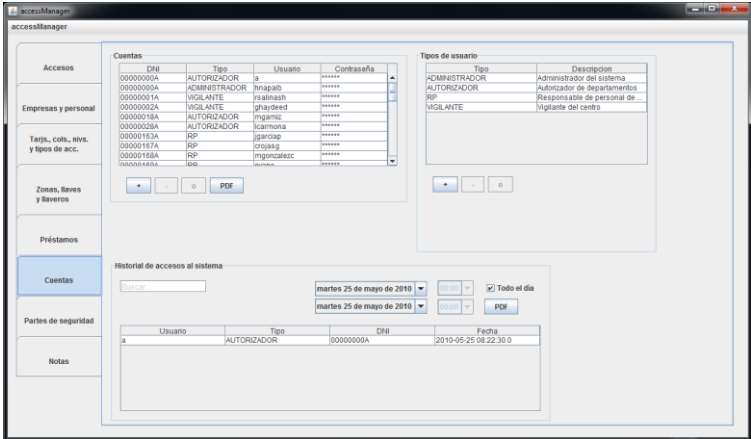
Añadir que el personal ocasional no aparece en la lista disponible para realizar las altas de cuentas.

Evaluación/
conclusión

El caso de uso '105 - Añadir cuenta' ha sido realizado satisfactoriamente.

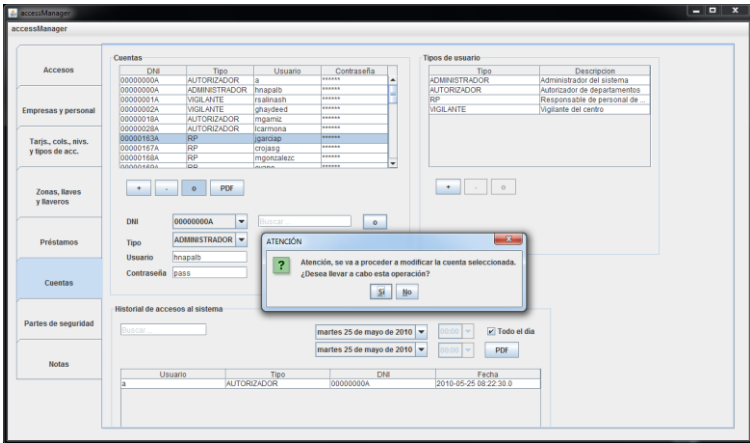
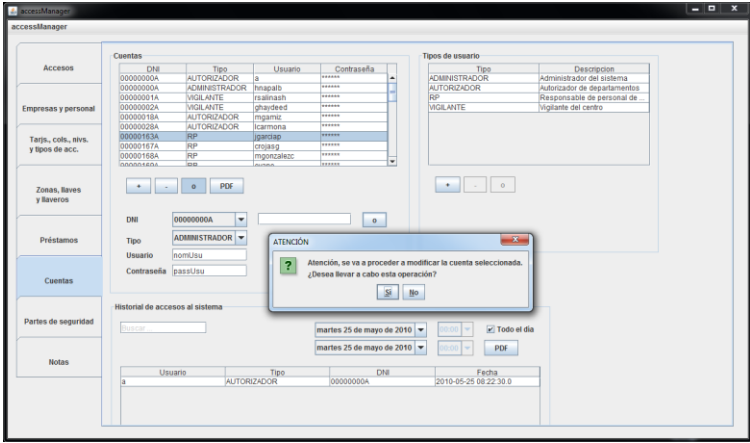
Caso de prueba para el caso de uso '105 - Añadir cuenta'



Caso de prueba	8.03 - Caso de uso 106: Borrar cuenta
Objetivo	Verificar que el usuario puede realizar el borrado de cuentas según lo establecido, sin que ello provoque algún tipo de error en el sistema.
Actividades	Acceder a accessManager, situarnos en la tab de cuentas, seleccionar varias de ellas y hacer que desaparezcan del sistema.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos resultante del caso de prueba anterior.
Datos de entrada	<p>Probamos a borrar una cuenta de cada tipo asegurándonos que estén salteadas en la tabla:</p> 
Salida esperada	Las cuentas seleccionadas deberán dejar de formar parte del sistema, no apareciendo en la tabla de cuentas.
Salida observada	La esperada: 
Evaluación/ conclusión	El caso de uso '106 - Borrar cuenta' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '106 - Borrar cuenta'

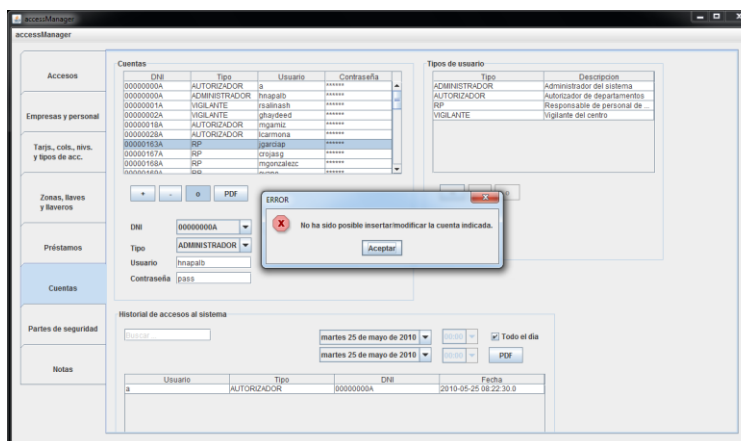


Caso de prueba	8.04 - Caso de uso 107: Modificar cuenta
Objetivo	Chequear que el administrador puede modificar los datos de cualquier cuenta existente el sistema.
Actividades	Seleccionar una cuenta, editar sus campos y comprobar que la operación no se realiza si estos son erróneos y que se realiza si son válidos.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos resultante del caso de prueba anterior. Modificaremos la cuenta de tipo RP que posee la persona con DNI 00000163A, asignándole datos erróneos... <ul style="list-style-type: none">• 'hnapalb,pass,ADMINISTRADOR,00000000A'. (Usuario existente) 
Datos de entrada	<ul style="list-style-type: none">• 'usuariodemadiadolargoparaentraqui, pass,ADMINISTRADOR,00000000A'. (Formato erróneo) ... y válidos: <ul style="list-style-type: none">• 'nomUsu,passUsu,ADMINISTRADOR,00000000A'. (Usuario válido) 
Salida esperada	Al igual que ocurría con el caso de adición, los dos primeros intentos serán fallidos y el último actualizará la tupla seleccionada con la indicada.

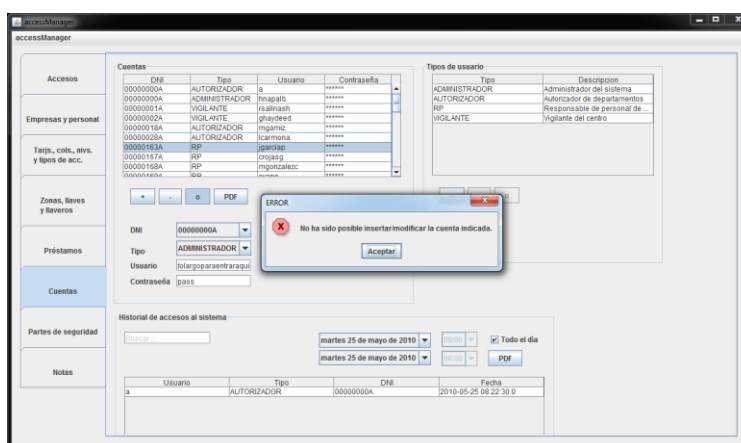


La espera:

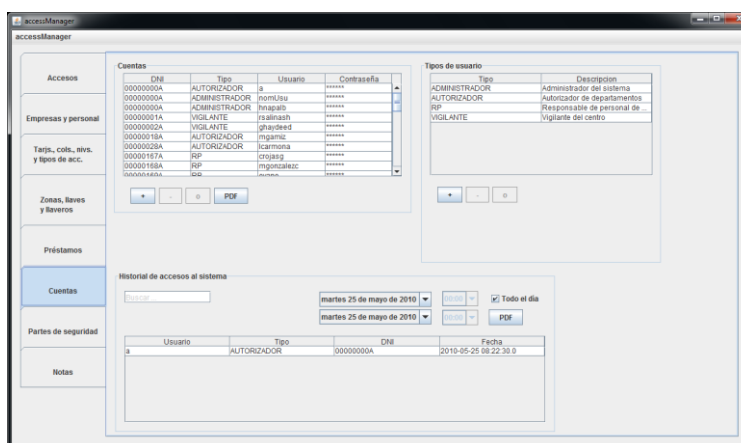
- Primer y segundo intento:



Salida observada



- Tercer intento:

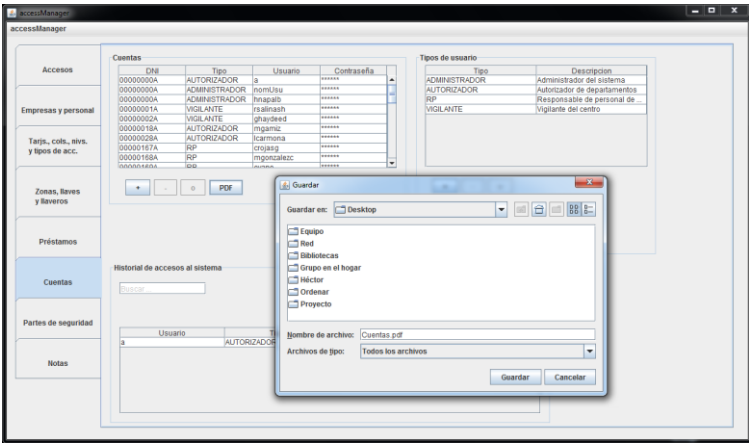


Evaluación/
conclusión

El caso de uso '107 - Modificar cuenta' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '107 - Modificar cuenta'



Caso de prueba	8.05 - Caso de uso 108: Listar cuentas																																																																																																																																																																								
Objetivo	Verificar que el usuario puede obtener en todo momento un reporte en formato PDF con las cuentas existentes en el sistema.																																																																																																																																																																								
Actividades	Arrancar la aplicación, situarnos en la tab de cuentas y pedir al sistema que genere el reporte de éstas. Acto seguido, abrir el informe generado y comprobar que es correcto.																																																																																																																																																																								
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior. • Utilidad phpMyAdmin. • Aplicación Adobe Reader. 																																																																																																																																																																								
Datos de entrada	<p>Generamos el listado de cuentas:</p>  <p>The screenshot shows the 'Cuentas' (Users) tab in the accessManager application. It displays a table with columns for DNI, Tipo, Usuario, and Contraseña. A 'Guardar' dialog box is open, showing the file name 'Cuentas.pdf' and the location 'Desktop'.</p>																																																																																																																																																																								
Salida esperada	<p>El reporte con todas las cuentas existentes en la tabla Cuenta:</p> <table border="1" data-bbox="564 1223 1110 1664"> <thead> <tr> <th>Usuario</th> <th>Nombre de usuario</th> <th>Pass</th> <th>Contraseña</th> <th>Tipo</th> <th>Tipo de usuario</th> <th>DNI</th> <th>DNI de la persona a la que pertenece la cuenta</th> </tr> </thead> <tbody> <tr><td>a</td><td></td><td>a</td><td></td><td>AUTORIZADOR</td><td></td><td>00000000A</td><td></td></tr> <tr><td>aesteroo</td><td></td><td>cambiar</td><td>RP</td><td></td><td></td><td>00000179A</td><td></td></tr> <tr><td>bmoratlil</td><td></td><td>cambiar</td><td>RP</td><td></td><td></td><td>00000184A</td><td></td></tr> <tr><td>cabellana</td><td></td><td>cambiar</td><td>RP</td><td></td><td></td><td>00000194A</td><td></td></tr> <tr><td>cbarberop</td><td></td><td>cambiar</td><td>RP</td><td></td><td></td><td>00000180A</td><td></td></tr> <tr><td>cchavezc</td><td></td><td>cambiar</td><td>RP</td><td></td><td></td><td>00000183A</td><td></td></tr> <tr><td>clarad</td><td></td><td>cambiar</td><td>RP</td><td></td><td></td><td>00000176A</td><td></td></tr> <tr><td>crojasg</td><td></td><td>cambiar</td><td>RP</td><td></td><td></td><td>00000167A</td><td></td></tr> <tr><td>cruiza</td><td></td><td>cambiar</td><td>RP</td><td></td><td></td><td>00000185A</td><td></td></tr> <tr><td>csevillae</td><td></td><td>cambiar</td><td>RP</td><td></td><td></td><td>00000182A</td><td></td></tr> <tr><td>cvaldivial</td><td></td><td>cambiar</td><td>RP</td><td></td><td></td><td>00000174A</td><td></td></tr> <tr><td>czufigae</td><td></td><td>cambiar</td><td>RP</td><td></td><td></td><td>00000175A</td><td></td></tr> <tr><td>ebaezag</td><td></td><td>cambiar</td><td>RP</td><td></td><td></td><td>00000195A</td><td></td></tr> <tr><td>ejimenezt</td><td></td><td>cambiar</td><td>RP</td><td></td><td></td><td>00001000A</td><td></td></tr> <tr><td>ereulag</td><td></td><td>cambiar</td><td>RP</td><td></td><td></td><td>00000186A</td><td></td></tr> <tr><td>evane</td><td></td><td>cambiar</td><td>RP</td><td></td><td></td><td>00000169A</td><td></td></tr> <tr><td>gdelicadod</td><td></td><td>cambiar</td><td>RP</td><td></td><td></td><td>00000198A</td><td></td></tr> <tr><td>gesterob</td><td></td><td>cambiar</td><td>RP</td><td></td><td></td><td>00000181A</td><td></td></tr> <tr><td>ghaydeed</td><td></td><td>cambiar</td><td>VIGILANTE</td><td></td><td></td><td>00000002A</td><td></td></tr> <tr><td>hnapalb</td><td></td><td>cambiar</td><td>ADMINISTRADOR</td><td></td><td></td><td>00000000A</td><td></td></tr> </tbody> </table> <p>...</p>	Usuario	Nombre de usuario	Pass	Contraseña	Tipo	Tipo de usuario	DNI	DNI de la persona a la que pertenece la cuenta	a		a		AUTORIZADOR		00000000A		aesteroo		cambiar	RP			00000179A		bmoratlil		cambiar	RP			00000184A		cabellana		cambiar	RP			00000194A		cbarberop		cambiar	RP			00000180A		cchavezc		cambiar	RP			00000183A		clarad		cambiar	RP			00000176A		crojasg		cambiar	RP			00000167A		cruiza		cambiar	RP			00000185A		csevillae		cambiar	RP			00000182A		cvaldivial		cambiar	RP			00000174A		czufigae		cambiar	RP			00000175A		ebaezag		cambiar	RP			00000195A		ejimenezt		cambiar	RP			00001000A		ereulag		cambiar	RP			00000186A		evane		cambiar	RP			00000169A		gdelicadod		cambiar	RP			00000198A		gesterob		cambiar	RP			00000181A		ghaydeed		cambiar	VIGILANTE			00000002A		hnapalb		cambiar	ADMINISTRADOR			00000000A	
Usuario	Nombre de usuario	Pass	Contraseña	Tipo	Tipo de usuario	DNI	DNI de la persona a la que pertenece la cuenta																																																																																																																																																																		
a		a		AUTORIZADOR		00000000A																																																																																																																																																																			
aesteroo		cambiar	RP			00000179A																																																																																																																																																																			
bmoratlil		cambiar	RP			00000184A																																																																																																																																																																			
cabellana		cambiar	RP			00000194A																																																																																																																																																																			
cbarberop		cambiar	RP			00000180A																																																																																																																																																																			
cchavezc		cambiar	RP			00000183A																																																																																																																																																																			
clarad		cambiar	RP			00000176A																																																																																																																																																																			
crojasg		cambiar	RP			00000167A																																																																																																																																																																			
cruiza		cambiar	RP			00000185A																																																																																																																																																																			
csevillae		cambiar	RP			00000182A																																																																																																																																																																			
cvaldivial		cambiar	RP			00000174A																																																																																																																																																																			
czufigae		cambiar	RP			00000175A																																																																																																																																																																			
ebaezag		cambiar	RP			00000195A																																																																																																																																																																			
ejimenezt		cambiar	RP			00001000A																																																																																																																																																																			
ereulag		cambiar	RP			00000186A																																																																																																																																																																			
evane		cambiar	RP			00000169A																																																																																																																																																																			
gdelicadod		cambiar	RP			00000198A																																																																																																																																																																			
gesterob		cambiar	RP			00000181A																																																																																																																																																																			
ghaydeed		cambiar	VIGILANTE			00000002A																																																																																																																																																																			
hnapalb		cambiar	ADMINISTRADOR			00000000A																																																																																																																																																																			



La esperada:

Salida observada

iN Informática Navarra

CUENTAS

Cuentas existentes en el sistema:

ID	Tipo	Departamento	Contraseña
00000000	AUTORIZADOR	z	*****
00000000A	ADMINISTRADO	nomUsu	*****
00000000A	ADMINISTRADO	R	*****
00000000A	ADMINISTRADO	R	*****
00000001A	VISITANTE	tsainnah	*****
00000002A	VISITANTE	ghayced	*****
00000008A	AUTORIZADOR	msamir	*****
00000008A	AUTORIZADOR	tsamona	*****
00000107A	RP	tsbaga	*****
00000108A	RP	msocelaco	*****
00000109A	RP	evane	*****
00000109A	RP	tsnara	*****
00000111A	RP	paonaleto	*****
00000112A	RP	mised	*****
00000113A	RP	tsosopral	*****
00000114A	RP	tsaldival	*****
00000115A	RP	tsalfiga	*****
00000116A	RP	tsarad	*****
00000117A	RP	msocem	*****
00000118A	RP	tsmopon	*****
00000118A	RP	tsesico	*****
00000118A	RP	tsarben	*****
00000118A	RP	tsesicob	*****
00000118A	RP	tsesitae	*****
00000118A	RP	tsobance	*****
00000118A	RP	tsocallal	*****
00000118A	RP	tsoga	*****
00000118A	RP	tsnaba	*****
00000118A	RP	tsnandize	*****
00000118A	RP	tsnencav	*****
00000118A	RP	tsnaga	*****

Evaluación/
conclusión

El caso de uso '108 - Listar cuentas' ha sido realizado satisfactoriamente.

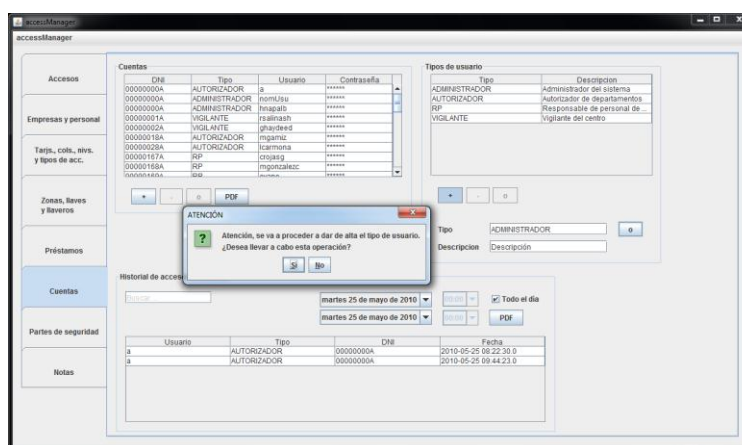
Caso de prueba para el caso de uso '108 - Listar cuentas'



Caso de prueba	8.06 - Caso de uso 111: Añadir tipo de usuario
Objetivo	Comprobar que el administrador puede añadir tipos de usuario al sistema sin que el sistema llegue a una situación inestable, de error.
Actividades	Acceder al sistema, situarnos en la sección de tipos de usuario y probar a introducir tipos erróneos y válidos, demostrando que en el primer caso no se producen inserciones pero sí en el segundo.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos resultante del caso de prueba anterior.

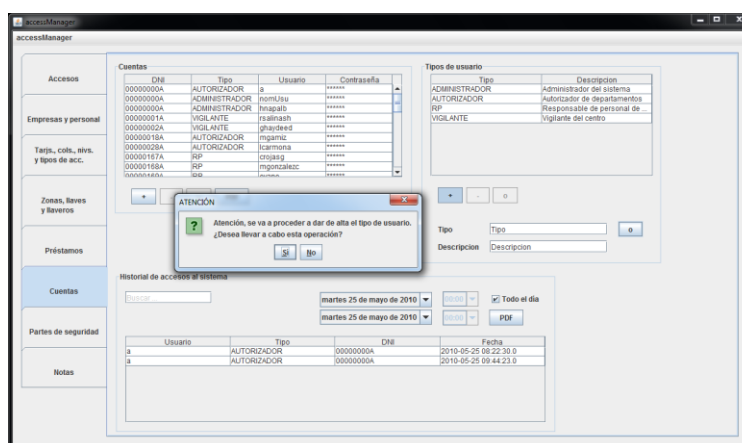
Probaremos a insertar:

- 'ADMINISTRADOR, Descripción'. (Tipo de usuario existente)



Datos de entrada

- 'tipodeusuariodemadiadolargoparaentrraqui, Descripción'. (Formato erróneo)
- 'Tipo, Descripción'. (Tipo de usuario válido)

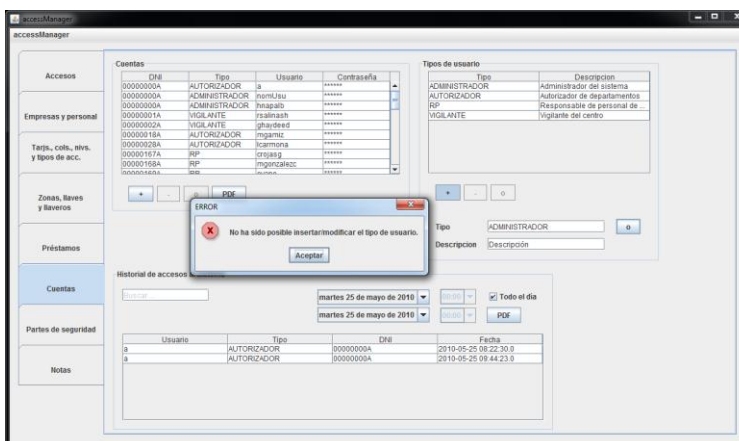


Salida esperada

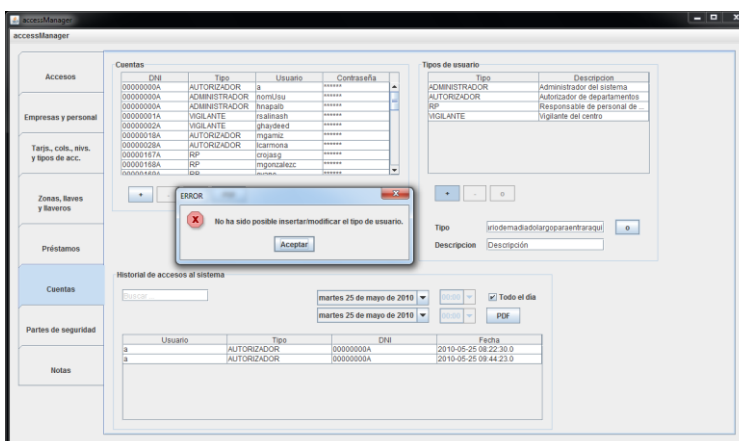
Esperamos que los dos primeros intentos sean en balde, pero no así el tercero, que dará como resultado un nuevo tipo de usuario en el sistema.

La esperada:

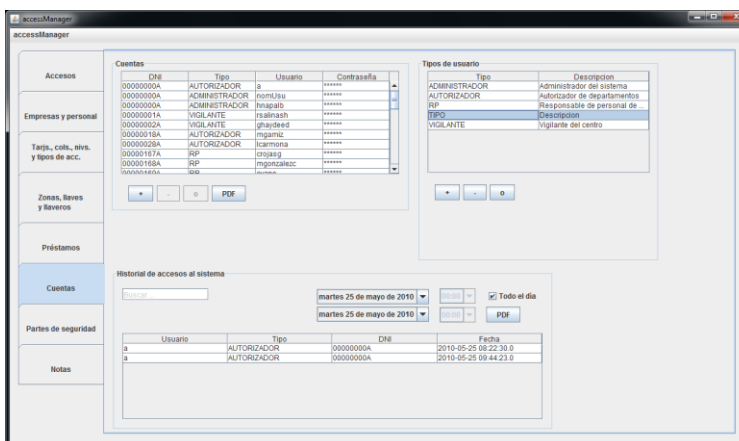
- Inserciones erróneas:



Salida observada



- Inserción válida:

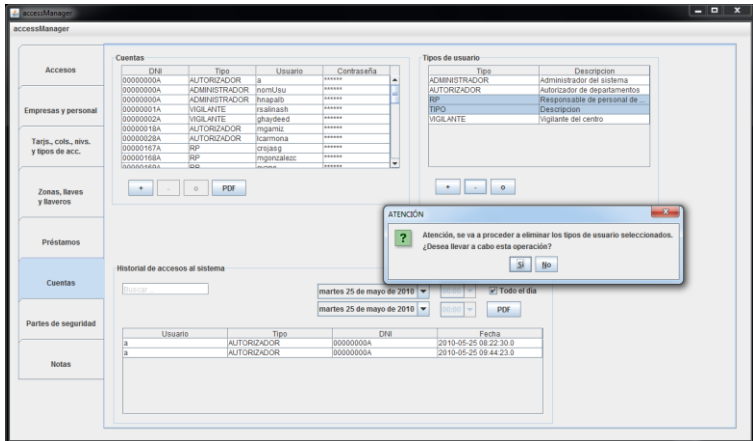
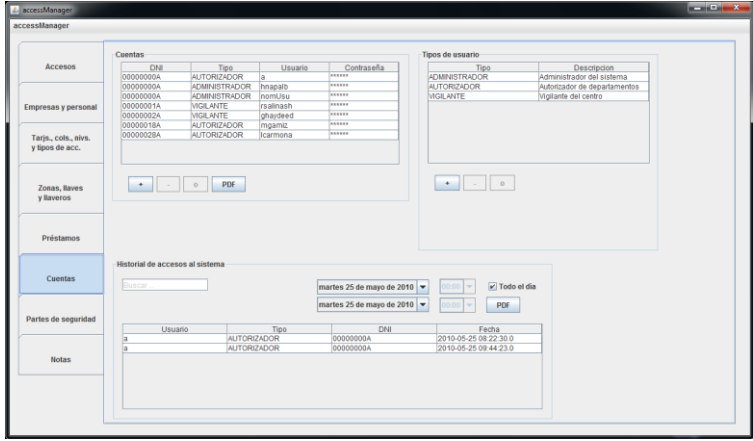


Evaluación/
conclusión

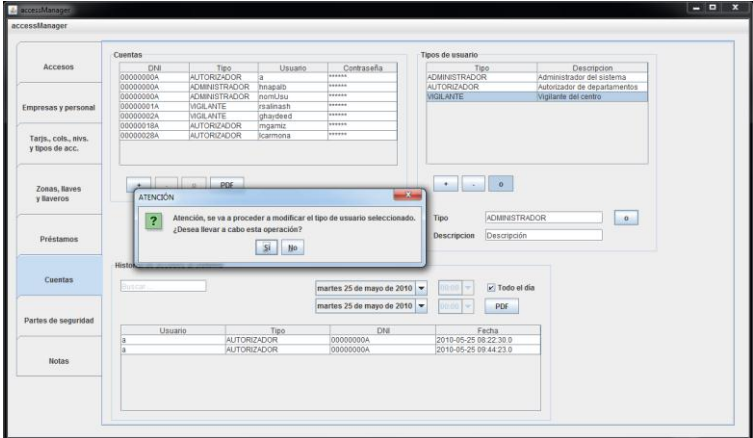
El caso de uso '111 - Añadir tipo de usuario' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '111 - Añadir tipo de usuario'



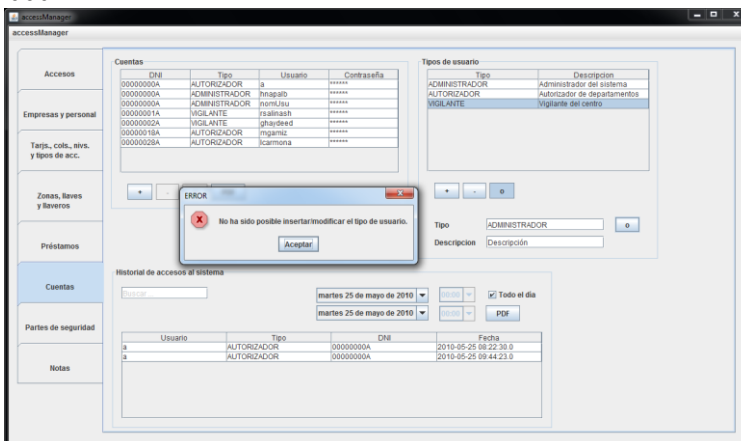
Caso de prueba	8.07 - Caso de uso 112: Borrar tipo de usuario
Objetivo	Probar que el borrado de tipos de usuario se realiza de acuerdo a los pasos establecidos en el momento del análisis y diseño.
Actividades	Acceder a accessManager y situarnos en la tab 'Cuentas', donde seleccionaremos dos tipos de usuario a eliminar y verificaremos que son eliminados junto con todas las cuentas que estén relacionadas con éstos.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos resultante del caso de prueba anterior.
Datos de entrada	<p>Vamos a borrar el tipo de usuario creado en el caso anterior ('Tipo') y el 'RP':</p>  <p>The screenshot shows the 'accessManager' application interface. On the left, there is a navigation menu with options like 'Accesos', 'Empresas y personal', 'Tarifa, cols. niv. y tipos de acc.', 'Zonas, bases y servicios', 'Préstamos', 'Cuentas', 'Partes de seguridad', and 'Notas'. The main area is divided into two panes. The left pane is titled 'Cuentas' and contains a table with columns: 'DNI', 'Tipo', 'Usuario', and 'Contraseña'. The right pane is titled 'Tipos de usuario' and contains a table with columns: 'Tipo' and 'Descripción'. A dialog box titled 'ATENCIÓN' is overlaid on the interface, with the text: 'Atención, se va a proceder a eliminar los tipos de usuario seleccionados. ¿Desea llevar a cabo esta operación?' and buttons for 'Sí' and 'No'.</p>
Salida esperada	Los tipos de usuario seleccionados para el borrado deberán abandonar la interfaz, así como todas las cuentas de personas que desempeñen el rol de responsable de personal.
Salida observada	La esperada:  <p>The screenshot shows the 'accessManager' application interface after the deletion process. The 'Tipos de usuario' table is now empty, indicating that the selected user types have been successfully removed. The 'Cuentas' table remains visible on the left.</p>
Evaluación/conclusión	El caso de uso '112 - Borrar tipo de usuario' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '112 - Borrar tipo de usuario'

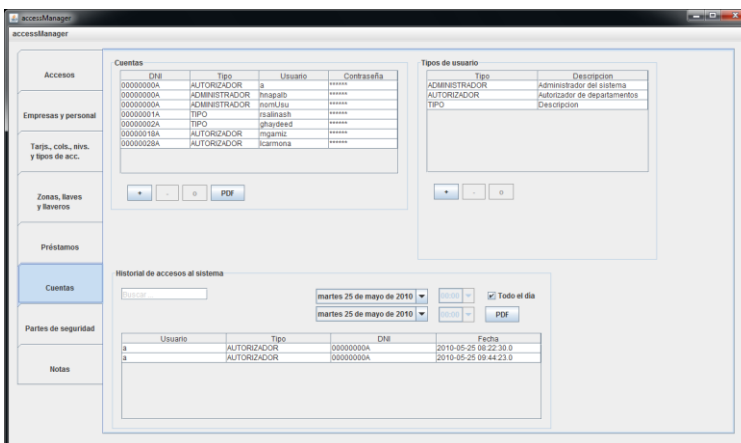
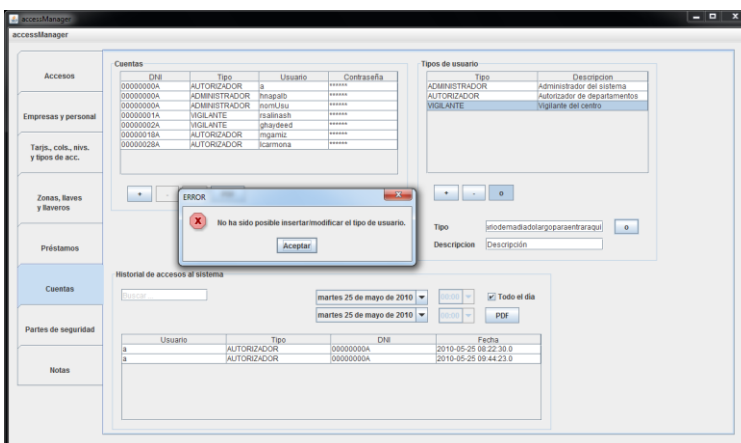
Caso de prueba	8.08 - Caso de uso 113: Modificar tipo de usuario
Objetivo	Asegurar que si el administrador necesita modificar alguno de los datos que componen un tipo de usuario, va a poder llevar a cabo esta operación sin ningún problema.
Actividades	Loguear en el sistema, seleccionar un tipo de usuario y modificarlo errónea y válidamente, comprobando que hemos implementado correctamente el caso de uso.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior.
Datos de entrada	<p>Estas pruebas pueden ser realizadas con cualquiera de los tipos de usuarios existentes. En este caso de prueba, optaremos por modificar el tipo de usuario 'Vigilante'.</p> <p>Las actualizaciones a intentar serán las mismas que las del caso de prueba 111, solo que en lugar de insertar un nuevo tipo deberemos comprobar que se modifica el seleccionado, así como las cuentas relacionadas con éste. Recordamos que las entradas a probar son:</p> <ul style="list-style-type: none"> • 'ADMINISTRADOR, Descripción'. (Tipo de usuario existente)  <ul style="list-style-type: none"> • 'tipodeusuariodemadiadolargoparaentraqui, Descripción'. (Formato erróneo) • 'Tipo, Descripción'. (Tipo de usuario válido)
Salida esperada	<p>La descrita anteriormente: las dos primeras pruebas darán como únicos resultados mensajes de error, mientras que la tercera provocará la existencia de un nuevo tipo de usuario en la base de datos y la actualización de todas las cuentas que la referencian.</p>



La espera:



Salida observada



Evaluación/
conclusión

El caso de uso '113 - Modificar tipo de usuario' ha sido realizado satisfactoriamente.

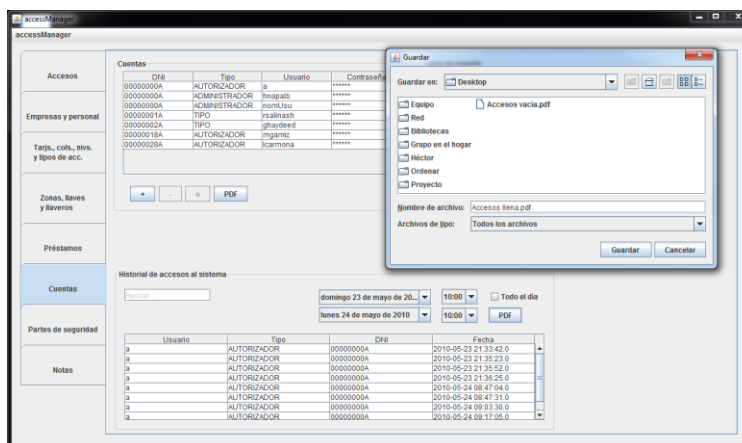
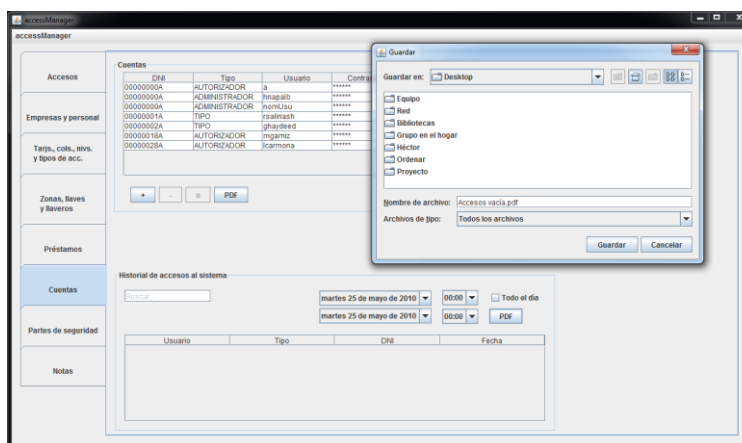
Caso de prueba para el caso de uso '113 - Modificar tipo de usuario'



Caso de prueba	8.09 - Caso de uso 115: Listar accesos al sistema
Objetivo	Verificar que el usuario puede obtener siempre que lo desee un reporte en PDF con los accesos visualizados en la tabla de accesos al sistema.
Actividades	Arrancar la aplicación, situarnos en la tab 'Cuentas', definir la búsqueda y pedir al sistema que genere los listados de accesos al sistema que estimemos oportunos.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos resultante del caso de prueba anterior.• Aplicación Adobe Reader.

Generamos dos reportes de accesos al sistema, uno vacío y otro acotando la búsqueda:

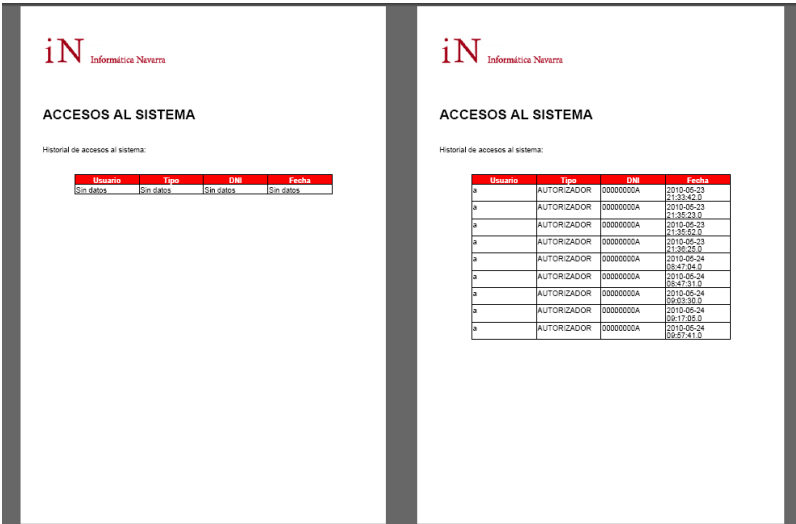
Datos de entrada



Salida esperada

Dos reportes en formato PDF con los accesos visualizados en el momento en que se ejecuta esta funcionalidad.



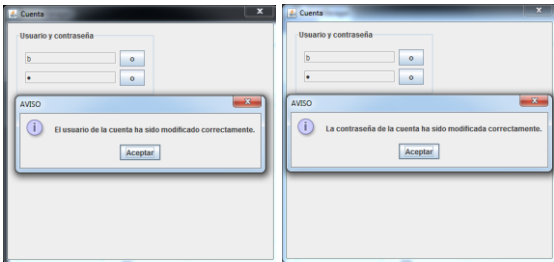
	La esperada:
Salida observada	
Evaluación/ conclusión	El caso de uso '115 - Listar accesos al sistema' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '115 - Listar accesos al sistema'



Caso de prueba	8.10 - Caso de uso 116: Modificar datos de cuenta
Objetivo	Comprobar que el diseño y funcionalidad del diálogo que permite a cualquier tipo de usuario modificar los datos de su cuenta ha sido diseñado e implementado con total corrección.
Actividades	Acceder a la aplicación, abrir el diálogo de datos de la cuenta y probar que todos los caminos posibles en la actualización se ejecutan según lo previsto.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos con población de prueba.
Datos de entrada	<p>Planeamos la siguiente estrategia de prueba:</p> <ol style="list-style-type: none">1. Datos de la cuenta actual no válidos<ol style="list-style-type: none">1.1. Campos con formato no deseado1.2. No casan los datos proporcionados con los de la base de datos2. Datos de la cuenta actual válidos<ol style="list-style-type: none">2.1. Cambiar nombre de usuario<ol style="list-style-type: none">2.1.1. Introducir un nombre de usuario erróneo<ol style="list-style-type: none">2.1.1.1. El nombre de usuario ya existe2.1.1.2. Campo con formato no deseado2.1.2. Introducir un nombre de usuario válido (en el ejemplo cambiamos de 'a' a 'b').2.2. Cambiar la contraseña<ol style="list-style-type: none">2.2.1. Las dos contraseñas introducidas no son idénticas<ol style="list-style-type: none">2.2.1.1. Campos con formato no deseado2.2.1.2. Ambas contraseñas no casan2.2.2. Las dos contraseñas introducidas son idénticas<ol style="list-style-type: none">2.2.2.1. Campos con formato no deseado2.2.2.2. Contraseñas válidas (en el ejemplo, cambiamos de 'a' a 'b').
Salida esperada	<p>Las respuestas esperadas a cada uno de los caminos anteriores son:</p> <ol style="list-style-type: none">1. Error2. Datos de la cuenta actual válidos<ol style="list-style-type: none">2.1. Cambiar nombre de usuario<ol style="list-style-type: none">2.1.1. Error2.1.2. El nombre de usuario es actualizado correctamente2.2. Cambiar la contraseña<ol style="list-style-type: none">2.2.1.1. Error2.2.2. Las dos contraseñas introducidas son idénticas<ol style="list-style-type: none">2.2.2.1. Error2.2.2.2. La contraseña es actualizada correctamente



Salida observada	<p>La esperada. Los caminos que preveíamos que conllevarían a error lo hacen y cuando todo es correcto se produce la actualización de los datos de la cuenta:</p> 
Evaluación/ conclusión	El caso de uso '116 - Modificar datos de cuenta' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '116 - Modificar datos de cuenta'

6.2. De integración

Tal y como viene sucediendo a lo largo de las diversas iteraciones de construcción, no requerimos de pruebas de integración que verifiquen el diseño o la arquitectura del sistema ya que esta no cambia. Las capas BD, API y GUI siguen manteniéndose tal y como las diseñamos y siguen relacionándose correctamente tras las pruebas de unidad ejecutadas.

6.3. De sistema

Todos los casos de uso establecidos en la planificación de la iteración han sido solventados de forma correcta con los recursos asignados.

7. Evaluación de la iteración

Completamos otros 14 casos de uso y las sensaciones siguen siendo buenas. Hemos conseguido:

- Desarrollar toda la parte del software que nos propusimos en el tiempo asignado.
- Realizar 10 casos de prueba, todos ellos probados con resultados positivos.
- Documentar toda la iteración explicando todos los detalles que serán de importancia en el futuro mantenimiento del sistema.

Son objetivos que marcan el grado de satisfacción que podemos tener tras el trabajo y creemos que todos han sido cumplidos con creces.



8. Despliegue

La conclusión de la fase de pruebas con obtención de buenos resultados nos lleva a dar por finalizadas las etapas de creación y podemos pasar a la de evaluación. Esta etapa, la de despliegue, no difiere de las anteriores ya que sigue estando basada en la exposición y prueba conjunta con el cliente del prototipo conseguido tras finalizar la iteración. Todo sucede según lo previsto: el cliente entiende y está satisfecho con todo el trabajo realizado; cree que la pestaña diseñada es muy sencilla de manejar y proporciona toda la funcionalidad (casos de uso) que éste esperaba de ella, por lo que nos da el visto bueno para pasar a la siguiente iteración del proceso de desarrollo.

Pamplona, a 23 de marzo de 2010.

Héctor Napal Basterrica
Responsable del proyecto

Pedro Martínez Pérez
*Jefe de área de producción de
Informática Navarra S.A.*



accessManager
Desarrollo de un sistema de control de accesos a un centro

Iteración C9: Partes de seguridad

Fecha de entrega: 26 de marzo de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Estructura del documento

2. Estado del proyecto

3. Análisis

- 3.1. Planificación de la iteración
- 3.2. Análisis de los casos de uso y escenarios

4. Diseño

- 4.1. Casos de uso 117, 118 y 119: Funciones de inicialización
- 4.2. Caso de uso 120: Visualizar partes de seguridad
- 4.3. Caso de uso 121: Visualizar parte de seguridad
- 4.4. Caso de uso 122: Añadir parte de seguridad
- 4.5. Caso de uso 123: Borrar parte de seguridad
- 4.6. Caso de uso 124: Modificar parte de seguridad
- 4.7. Caso de uso 125: Listar parte de seguridad

5. Codificación e integración

6. Pruebas

- 6.1. De unidad
- 6.2. De integración
- 6.3. De sistema

7. Evaluación de la iteración

8. Despliegue



1. Introducción

1.1. Propósito

Redactamos este artefacto con la finalidad de que sirva de documento de consulta para el cliente y los técnicos que realicen el futuro mantenimiento del sistema.

1.2. Audiencia

Documento de carácter interno hasta la transmisión del producto software al cliente. Se entregará junto con el resto de artefactos, formando entre todos una pormenorizada documentación de cómo ha sido desarrollado el sistema y cómo está constituido.

1.3. Estructura del documento

Comenzamos el artefacto comentando en qué punto del desarrollo nos encontramos y “dejando caer” a qué dedicaremos esta iteración. Una vez situados, en la sección 3 planificamos y analizamos todos los casos de uso que implementaremos en esta iteración. A continuación documentamos extensamente cómo hemos diseñado todos los casos de uso propuestos, explicando qué estrategias escogeremos y acompañándolas de diagramas de actividades y secuencia. El quinto capítulo está dedicado a la codificación e integración de todo el diseño realizado. Tras éste apartado pasamos a exponer todas las pruebas que hemos ejecutado junto con los resultados obtenidos, para terminar realizando una evaluación de la iteración en la sección 7 y cita de los datos más relevantes de la actividad de despliegue en la 8.

2. Estado del proyecto

Completada la implementación de toda la funcionalidad imprescindible de la que debía constar el sistema y cumpliendo con la planificación propuesta (no se observa ningún tipo de retraso), continuamos con el desarrollo de los siguientes casos de uso.

Hasta el momento hemos codificado una aplicación que podía proporcionar toda la funcionalidad para la que fue ideada (la gestión de accesos a un centro) pero necesitaba de un trabajo “extra” que automatizaremos al comienzo de esta iteración. Una vez dispongamos de toda la serie de procesos de inicialización, nos centraremos en el desarrollo de la pestaña ‘Partes de seguridad’, que permitirá agilizar los procesos diarios que existen entre el administrador y los vigilantes de seguridad del centro.



3. Análisis

3.1. Planificación de la iteración

Antes que nada hay que planificar todo el trabajo que vamos a realizar en los tres días que durará la iteración:

PLAN DE ITERACIÓN C9: PARTES DE SEGURIDAD			
ASPECTO	DESCRIPCIÓN		
Objetivos	Crear un conjunto de procesos software que se inicien con la aplicación y permitan desarrollar buena parte de las tareas de mantenimiento necesarias. Tras concluir esta etapa, nos centraremos en desarrollar todos los casos de uso que conforman toda la visualización y edición de partes de seguridad.		
Artefactos de entrada	<ul style="list-style-type: none"> • <i>Especificación inicial de los casos de uso</i> • <i>Planificación temporal y costes detallados del proyecto</i> • Base de datos, API y aplicación desarrolladas en las iteraciones anteriores. 		
Artefactos modificados o creados	<ul style="list-style-type: none"> • Diagrama de clases de la API inicial. • Clases Java de la API inicial y de la aplicación. • El presente documento. 		
Calendarización detallada	Análisis + Pruebas	24/03/10	<ul style="list-style-type: none"> • Planificación de la iteración y gestión de riesgos. • Análisis de los casos de uso y escenarios. • Planificación de las pruebas. • Inicio de la documentación.
	Diseño + Pruebas	24/03/10	<ul style="list-style-type: none"> • Diseño de opciones arquitectónicas. • Diseño de pruebas.
	Codificación + Pruebas	25/03/10	<ul style="list-style-type: none"> • Codificación e integración de la solución. • Ejecución y evaluación de pruebas. • Evaluación del prototipo. • Preparación de la entrega ejecutable.
		26/03/10	
Despliegue	26/03/10	<ul style="list-style-type: none"> • Muestra, prueba y evaluación conjunta con el cliente. • Concreción de los cambios necesarios. • Finalización de toda la documentación. 	

Planificación de la iteración



Nombre de tarea	Duración	Comienzo	Fin	0	mié 24 mar	0	vie 26 mar	0
Estudio de oportunidad (inception)	7 días	jue 07/01/10	vie 15/01/10					
Elaboración	16 días	lun 18/01/10	lun 08/02/10					
Construcción	41 días	mar 09/02/10	mar 06/04/10					
Iteración C1: La base de datos	4 días	mar 09/02/10	vie 12/02/10					
Iteración C2: API para la base de datos	5 días	lun 15/02/10	vie 19/02/10					
Iteración C3: Accesos	5 días	lun 22/02/10	vie 26/02/10					
Iteración C4: Empresas y personal	5 días	lun 01/03/10	vie 05/03/10					
Iteración C5: Tarjetas, tipos de acceso y cargos	3 días	lun 08/03/10	mié 10/03/10					
Iteración C6: Zonas, llaves y llaveros	5 días	jue 11/03/10	mié 17/03/10					
Iteración C7: Préstamos	2 días	jue 18/03/10	vie 19/03/10					
Iteración C8: Cuentas	2 días	lun 22/03/10	mar 23/03/10					
Iteración C9: Partes de seguridad	3 días	mié 24/03/10	vie 26/03/10					
Análisis	0,5 días	mié 24/03/10	mié 24/03/10					
Diseño	0,5 días	mié 24/03/10	mié 24/03/10					
Implementación	1,5 días	jue 25/03/10	vie 26/03/10					
Pruebas	3 días	mié 24/03/10	vie 26/03/10					
Despliegue	0,5 días	vie 26/03/10	vie 26/03/10					
Iteración C10: Notas	3 días	lun 29/03/10	mié 31/03/10					
Iteración C11: Base de datos	2 días	jue 01/04/10	vie 02/04/10					
Iteración C12: Diseño, personalización y configuración	2 días	lun 05/04/10	mar 06/04/10					
Capacidad operacional	0 días	mar 06/04/10	mar 06/04/10					
Transición	8 días	mié 07/04/10	vie 16/04/10					

Recursos	Temporales		3 días
	Humanos		1 persona
	Software	Sistema operativo	Microsoft Windows 7
		Ofimática	Microsoft Word 2010 Adobe Acrobat Professional 8
		Diseño	Visual Paradigm 4.2
Codificación		NetBeans 6.8	
Hardware		1 PC	

Casos de uso y escenarios implementados	117 - Recordar personal a punto de caducar 118 - Recordar préstamos pendientes 119 - Borrar datos obsoletos 120 - Visualizar partes de seguridad 121 - Visualizar parte de seguridad 122 - Añadir parte de seguridad 123 - Borrar parte de seguridad 124 - Modificar parte de seguridad 125 - Listar parte de seguridad
--	---

Planificación de la iteración

3.2. Análisis de los casos de uso y escenarios

Las tres primeras descripciones de trazo fino están asignadas al sistema y serán las encargadas de facilitar al administrador de éste el mantenimiento. Las seis restantes están dirigidas a profundizar en todos aquellos casos de uso relacionados con los partes que deben redactar los vigilantes de seguridad en cada servicio.



Caso de uso 117: Recordar personal a punto de caducar		
Descripción	En el momento de diseñar la base de datos que da soporte a todo el sistema, comentamos que las empresas que trabajan en la sede de Informática Navarra S.A. deben actualizar de forma periódica a su personal. Hasta el desarrollo de este proyecto, el jefe de área de producción era el encargado de avisar diariamente a los departamentos cuyo personal estaba a punto de caducar, pero una vez que accessManager esté en funcionamiento podremos automatizar esta tarea.	
Actores principal	Sistema.	
Pre-condiciones	El usuario que accede al sistema es el primer vigilante en hacerlo a lo largo del día.	
Flujos	Básico	El sistema obtiene qué departamentos tienen su personal a <i>d1</i> , <i>d2</i> y <i>d3</i> días de caducar y envía un E-Mail a todos los responsables de personal de éstos para avisarles de la situación.
	Alternativos (*)	Ninguno.
Post-condiciones	El sistema insta a los responsables de personal de los departamentos implicados a que actualicen su personal si no quieren que éste sea eliminado, liberando al administrador de esta responsabilidad.	

Caso de uso 117: Recordar personal a punto de caducar

Caso de uso 118: Recordar préstamos pendientes		
Descripción	La tarea anteriormente citada no es la única que realiza el administrador. Recordemos que existirán préstamos que no sean devueltos a la salida del centro. Informática Navarra S.A. no puede dar por perdido todo este material, lo que hace que el jefe de área de producción, además de recordar a los departamentos que le envíen la lista de su personal, también es el encargado de ir recordando a todas aquellas personas que tengan préstamos pendientes que los devuelvan con la mayor brevedad posible. Le ayudaremos también en esta tarea.	
Actores principal	Sistema.	
Pre-condiciones	El vigilante del primer turno del día arranca la aplicación.	
Flujos	Básico	El sistema obtiene el conjunto de trabajadores que poseen préstamos pendientes desde hace <i>d1</i> , <i>d2</i> y <i>d3</i> días y a todos aquellos de los que disponga de alguna dirección de correo electrónico les envía un mensaje informándoles del material que tienen pendiente de devolución.
	Alternativos (*)	Ninguno.
Post-condiciones	Los prestatarios que cumplen con las características impuestas son avisados vía E-Mail, ayudando así al jefe de área de producción en la recuperación del material "extraviado".	

Caso de uso 118: Recordar préstamos pendientes



Caso de uso 119: Borrar datos obsoletos					
Descripción	accessManager es al fin y al cabo un almacén de datos que alguien debe mantener si quiere conservarlo en condiciones óptimas; no podemos dejar que crezca a su libre albedrío. Es verdad que siempre deberá existir un administrador del sistema que lo conozca y pueda realizar sobre éste un mantenimiento de garantías, pero nunca está de más que ayudemos con aquellas tareas que se puedan automatizar.				
Actores principal	Administrador.				
Pre-condiciones	El sistema registra el acceso al sistema del primer vigilante del día.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> accessManager obtiene y borra: <ul style="list-style-type: none"> • Todos aquellos trabajadores ocasionales cuyo permiso de acceso ha caducado. • Todos aquellos departamentos cuya fecha máxima de renovación de personal haya expirado. • Todas aquellas personas que no constan en ninguna tabla especializada. • Todos los accesos, llaveros, llaves, niveles, colores y tarjetas (en desuso, dados de baja) cuya existencia en la base de datos excede de <i>d</i> días. </td> </tr> <tr> <td>Alternativos (*)</td> <td>Ninguno.</td> </tr> </table>	Básico	accessManager obtiene y borra: <ul style="list-style-type: none"> • Todos aquellos trabajadores ocasionales cuyo permiso de acceso ha caducado. • Todos aquellos departamentos cuya fecha máxima de renovación de personal haya expirado. • Todas aquellas personas que no constan en ninguna tabla especializada. • Todos los accesos, llaveros, llaves, niveles, colores y tarjetas (en desuso, dados de baja) cuya existencia en la base de datos excede de <i>d</i> días. 	Alternativos (*)	Ninguno.
	Básico	accessManager obtiene y borra: <ul style="list-style-type: none"> • Todos aquellos trabajadores ocasionales cuyo permiso de acceso ha caducado. • Todos aquellos departamentos cuya fecha máxima de renovación de personal haya expirado. • Todas aquellas personas que no constan en ninguna tabla especializada. • Todos los accesos, llaveros, llaves, niveles, colores y tarjetas (en desuso, dados de baja) cuya existencia en la base de datos excede de <i>d</i> días. 			
Alternativos (*)	Ninguno.				
Post-condiciones	La población residente en la base de datos está actualizada y es la idónea para el correcto funcionamiento del sistema.				

Caso de uso 119: Borrar datos obsoletos



Caso de uso 120: Visualizar partes de seguridad		
Descripción	El centro está custodiado constantemente por vigilantes jurados que deben realizar un parte de incidencias por servicio. Esta tarea se realizaba antes de forma manual y en el momento que bien el administrador o bien el vigilante deseaban consultar cualquier dato acudían al “taco” de partes. Ahora, al informatizar esta tarea y ser tan grande el volumen de partes de seguridad que pueden llegar a almacenarse, se debe dotar a la aplicación de un método que permita visualizar todos o parte de ellos, para después seleccionar el que se desea visionar (tratado en el siguiente caso de uso).	
Actores principal	Administrador y Vigilante.	
Pre-condiciones	El usuario accede al sistema y se dirige a la pestaña ‘Partes de seguridad’ con la intención de visionar los existentes.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El sistema muestra en un principio todos los partes de seguridad. 2. El vigilante restringe la muestra de partes mediante los parámetros de búsqueda de los que dispone si lo cree preciso. (*) 3. El sistema actualiza los partes visionados mostrando solo los que cumplan con los criterios establecidos.
	Alternativos (*)	<p>El usuario no filtra la búsqueda ya que los resultados que observa por pantalla cumplen con sus expectativas:</p> <ul style="list-style-type: none"> • El sistema da por finalizado el caso de uso sin realizar ninguna acción más.
Post-condiciones	El usuario visualiza en pantalla el conjunto de partes de seguridad que deseaba.	

Caso de uso 120: Visualizar partes de seguridad



Caso de uso 121: Visualizar parte de seguridad		
Descripción	Una vez que se visualiza los partes de seguridad que se desean, es necesario completar el proceso de visionado dando la posibilidad al usuario de acceder a la información que éstos almacenan.	
Actores principal	Administrador y Vigilante.	
Pre-condiciones	El usuario se encuentra logueado en el sistema, más concretamente en la tab de partes de seguridad, y desea visualizar los datos que componen un parte que observa en pantalla.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona el parte de seguridad del que desea obtener los datos que lo componen. 2. El sistema se encarga de acudir a la base de datos, rescatar los datos del parte y mostrárselos al usuario en pantalla.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario ha conseguido lo que buscaba: visualizar todos los datos del parte de seguridad seleccionado.	

Caso de uso 121: Visualizar parte de seguridad

Caso de uso 122: Añadir parte de seguridad		
Descripción	Un nuevo vigilante de seguridad comienza su labor diaria en el centro y a lo largo de su turno de trabajo se dispone a crear el parte de seguridad que debe redactar.	
Actores principal	Vigilante.	
Pre-condiciones	El vigilante se encuentra logueado en el sistema y visita la sección destinada a los partes de seguridad.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El vigilante transmite al sistema su intención de crear un nuevo parte de seguridad. 2. El sistema le proporciona un formulario para tal fin. 3. El vigilante lo rellena y pide al sistema que lo almacene. 4. El sistema obedecerá, pero antes deberá recibir una confirmación previa. (*)
	Alternativos (*)	<p>El usuario proporciona datos que no tienen el formato esperado por el sistema:</p> <ul style="list-style-type: none"> • El sistema le avisa del error y vuelve al punto 2, sin haber modificado ningún dato. <p>El usuario termina por no dar su visto bueno a la inserción del nuevo parte:</p> <ul style="list-style-type: none"> • El sistema da por concluido el caso de uso sin realizar ninguna inserción.
Post-condiciones	El sistema dispone de un nuevo parte de seguridad en su tabla Parte_Seguridad.	

Caso de uso 122: Añadir parte de seguridad



Caso de uso 123: Borrar parte de seguridad	
Descripción	Ésta será una funcionalidad muy poco utilizada y dirigida especialmente al administrador, aunque el vigilante también podrá hacer uso de ella. No es normal que cualquiera borre un parte de seguridad, más aún el vigilante, que dispone de la opción de edición, pero el cliente nos ha pedido que proveamos la posibilidad de borrar los partes de seguridad que se desee almacenar.
Actores principal	Administrador y Vigilante.
Pre-condiciones	El usuario quiere eliminar alguno de los partes de seguridad que visualiza en pantalla.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario selecciona los partes de seguridad a eliminar y pide al sistema que llevé a cabo la operación de borrado. 2. El sistema envía un mensaje al usuario avisándole de qué partes va a eliminar y si está seguro de continuar con la acción. 3. El usuario responde afirmativamente ante esta pregunta. (*) 4. El sistema borra los partes de seguridad indicados.
	<p>Alternativos (*)</p> <p>El usuario responde negativamente:</p> <ul style="list-style-type: none"> • El sistema no borra ningún elemento de la base de datos y el caso de uso llega a su fin.
Post-condiciones	accessManager ya no dispone de los partes de seguridad que el usuario ha seleccionado y eliminado.

Caso de uso 123: Borrar parte de seguridad



Caso de uso 124: Modificar parte de seguridad		
Descripción	Esta sí que será una funcionalidad muy utilizada, ya que lo normal es que un vigilante de seguridad abra un parte de seguridad al iniciar su servicio y lo vaya completando a medida que éste transcurre.	
Actores principal	Vigilante.	
Pre-condiciones	Surge alguna incidencia que el vigilante debe anotar o éste percata de que un parte posee algún dato erróneo, lo que lleva al vigilante a entrar en el sistema, buscar el parte y seleccionarlo.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El vigilante indica al sistema que desea editar el parte que se encuentra visualizando. 2. El sistema hace los campos del formulario editables. (*) 3. El usuario edita todos lo que estima oportuno y pide al sistema que los registre. 4. El sistema obedece tras una previa aprobación por parte del usuario. (*)
	Alternativos (*)	<p>El sistema no permite la edición del formulario ya que el usuario no dispone de privilegios suficientes.</p> <p>El usuario introduce datos con el formato indebido:</p> <ul style="list-style-type: none"> • El sistema avisa al usuario del error y no realiza ninguna actualización. <p>El usuario aborta la edición del parte:</p> <ul style="list-style-type: none"> • El caso de uso finaliza y ningún registro es modificado.
Post-condiciones	El parte de seguridad, que en un principio no era correcto, ahora sí lo es tras ser editado por el vigilante.	

Caso de uso 124: Modificar parte de seguridad



Caso de uso 125: Listar parte de seguridad					
Descripción	Para que administrador y vigilantes de seguridad puedan disponer de copias en papel de los partes de seguridad que redactan, el cliente nos pide que proporcionemos una vía para poder exportarlos en formato PDF.				
Actores principal	Administrador y Vigilante.				
Pre-condiciones	El administrador o vigilante se encuentra visualizando un parte de seguridad y cree conveniente reportarlo.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> <ol style="list-style-type: none"> 1. El usuario pide al sistema que exporte el parte que se visualiza en pantalla. 2. El sistema le pide al usuario que indique la ruta en donde desea almacenarlo. 3. El vigilante la indica. (*) 4. El sistema se encarga de crear el reporte con el parte de seguridad. </td> </tr> <tr> <td>Alternativos (*)</td> <td> El usuario no proporciona ninguna ruta: <ul style="list-style-type: none"> • El sistema no elabora ningún reporte y el caso de uso finaliza. </td> </tr> </table>	Básico	<ol style="list-style-type: none"> 1. El usuario pide al sistema que exporte el parte que se visualiza en pantalla. 2. El sistema le pide al usuario que indique la ruta en donde desea almacenarlo. 3. El vigilante la indica. (*) 4. El sistema se encarga de crear el reporte con el parte de seguridad. 	Alternativos (*)	El usuario no proporciona ninguna ruta: <ul style="list-style-type: none"> • El sistema no elabora ningún reporte y el caso de uso finaliza.
	Básico	<ol style="list-style-type: none"> 1. El usuario pide al sistema que exporte el parte que se visualiza en pantalla. 2. El sistema le pide al usuario que indique la ruta en donde desea almacenarlo. 3. El vigilante la indica. (*) 4. El sistema se encarga de crear el reporte con el parte de seguridad. 			
Alternativos (*)	El usuario no proporciona ninguna ruta: <ul style="list-style-type: none"> • El sistema no elabora ningún reporte y el caso de uso finaliza. 				
Post-condiciones	El usuario puede disfrutar del parte de seguridad en formato PDF.				

Caso de uso 125: Listar parte de seguridad

4. Diseño

4.1. Casos de uso 117, 118 y 119: Funciones de inicialización

Trataremos estos tres casos de uso en un mismo bloque ya que todos son funciones que deberán ejecutarse una vez cada día y en el mismo intervalo de tiempo.

Tenemos las tareas de:

- Recordar personal a punto de caducar.
- Recordar préstamos pendientes.
- Borrar datos obsoletos.

En un principio debemos idear cómo y cuándo deberán ser ejecutadas. Al ser tareas periódicas de mantenimiento deberían tratarse como tareas programadas por parte del servidor, pero creemos que es una buena idea tratarlas aquí, en la misma aplicación.

¿Cómo podemos hacer que sólo sean ejecutadas una vez cada día? Pensemos con qué frecuencia utilizan la aplicación cada tipo de usuario. Parece claro que pensar en los responsables de personal es irrisorio, pues conectarán con la base de datos las veces necesarias para renovar a su personal, que serán pocas a lo largo del año. Por otro lado tenemos el administrador, que sí que es cierto que tendrá un uso bastante continuado del sistema, pero no en sábados y



domingos, festivos, vacaciones,... En cambio, en los vigilantes tendremos a unos usuarios que siempre acudirán al sistema, de forma diaria. Trabajemos con ellos pues.

Tenemos que “al acceder el sistema el primer vigilante del día” realizar los casos de uso que vamos a tratar. De aquí se desprende que necesitaremos una función que resuelva esa incógnita: el primer vigilante del día.

MÉTODO	FUNCIONALIDAD
<i>primerVigilante</i>	Método que denota si el acceso realizado al sistema es el del primer vigilante del día.

Método a implementar y a añadir a la API

Si *primerVigilante* retorna *true*, actuaremos. ¿Cómo? Realizando las tres actividades anteriormente citadas. Nos queda pues establecer los modos de proceder, qué secuencia de acciones pueden solucionar estos problemas.

Las dos primeras actividades, que consisten en avisar a personas, pueden ser solventadas mediante el uso de JavaMail y la agenda de direcciones de correo electrónico de la que disponemos. La idea es crear dos métodos en la clase *Cartero* que envíen correos de aviso según los días que queden para la caducidad del personal o los transcurridos desde el primer préstamo que tenga una persona por devolver:

```
// Enviamos los E-Mails de actualizacion del personal  
Cartero.enviarCorreosRenovacionPersonal(30);  
Cartero.enviarCorreosRenovacionPersonal(15);  
Cartero.enviarCorreosRenovacionPersonal(7);
```

```
// Enviamos los E-Mails de peticion de devolucion de prestamos  
Cartero.enviarCorreosPrestamosPendientes(30);  
Cartero.enviarCorreosPrestamosPendientes(15);  
Cartero.enviarCorreosPrestamosPendientes(7);
```

Ambos métodos no tienen más dificultad que la de:

- Obtener todos los departamentos (junto con sus responsables de personal) cuya periodicidad sumada al día de hoy sea su fecha de caducidad y enviarles un correo avisándoles de la situación.
- Obtener todos aquellos prestatarios y préstamos pendientes de devolución cuyo primer préstamo más *d* días dé como resultado el día de hoy.



Éstos procesos se concretan en:

MÉTODO	FUNCIONALIDAD
<i>enviarCorreosRenovacionPersonal</i>	Método que obtiene los datos de todos aquellos departamentos cuyo personal caduca en d días y les envía un correo electrónico de aviso.
<i>enviarCorreosPrestamosPendientes</i>	Método que obtiene los datos de todas aquellas personas que poseen préstamos pendientes de devolución desde hace d días y les envía un mensaje de correo electrónico para recordárselo.
<i>obtenerDatosRenovacionPersonal</i>	Método empleado por <i>enviarCorreosRenovacionPersonal</i> para obtener todos los datos de los departamentos a avisar. Los datos retornados por este método serán los que use el primero para elaborar y enviar los E-Mails necesarios.
<i>obtenerDatosPrestamosPendientes</i>	Método análogo al anterior pero usado por <i>enviarCorreosPrestamosPendientes</i> para obtener los datos y préstamos de todos los prestatarios.

Métodos a implementar y a añadir a la API

Con la tercera actividad, la de borrar datos obsoletos, debemos dividir el trabajo en varios objetos, ya que tenemos que:

- Eliminar el personal ocasional que ya no puede acceder al centro → Clase Ocasional.
- Eliminar departamentos cuyo personal haya caducado → Clase Departamento.
- Eliminar todas aquellas personas que residen en Personal y que no tengan su especialización en alguna tabla derivada → Clase Personal.
- Eliminar todos los accesos, llaveros, llaves, niveles, colores y tarjetas (datos de baja) que lleven residiendo en el sistema más de d días → Clases Acceso, Llaverero, Llave, Nivel, Color y Tarjeta respectivamente.



Los métodos que realizan todas estas acciones y que debemos implementar son:

MÉTODO	FUNCIONALIDAD
<i>eliminarOcasional</i>	Método que elimina todo el personal ocasional que ya no dispone de permiso para acceder al centro.
<i>eliminarDepartamentos</i>	Método que elimina todos aquellos departamentos cuyo personal ha caducado.
<i>eliminarPersonal</i>	Método que elimina todas aquellas personas que existen en Personal pero no en ninguna de las tablas derivadas de ésta.
<i>eliminarAccesos</i>	Método que elimina los accesos desde los que han transcurrido <i>d</i> días.
<i>eliminarLlaveros</i>	Método que borra los llaveros dados de baja desde hace más de <i>d</i> días.
<i>eliminarLlaves</i>	Idént. con las llaves.
<i>eliminarNiveles</i>	Idént. con las niveles.
<i>eliminarColores</i>	Idént. con los colores.
<i>eliminarTarjetas</i>	Idént. con las tarjetas.

Métodos a implementar y a añadir a la API

Son métodos sencillos concretados con una simple *query* de borrado. Nos queda recordar que todas estas actividades tienen lugar en el mismo instante que las que realizamos para mantener la tabla de Accesos al día, con que éste sería el momento de agrupar los casos de uso 12, 117, 118 y 119.

Para aclarar todo esto y debido a la complejidad e importancia que posee la inicialización en caso de tratarse del primer acceso de un vigilante al sistema en el día, a continuación mostramos los diagramas que definen todo este proceso:

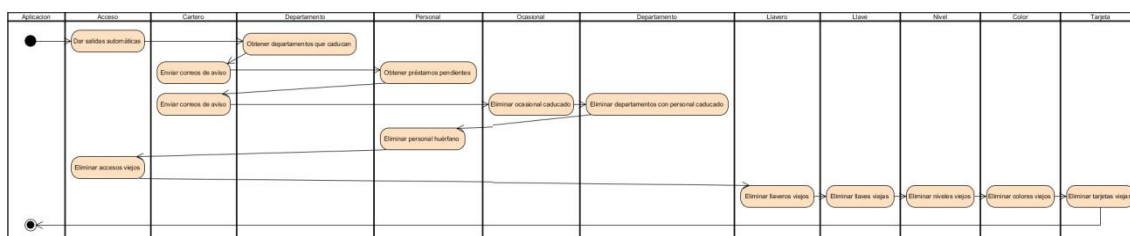


Diagrama de actividades de la inicialización

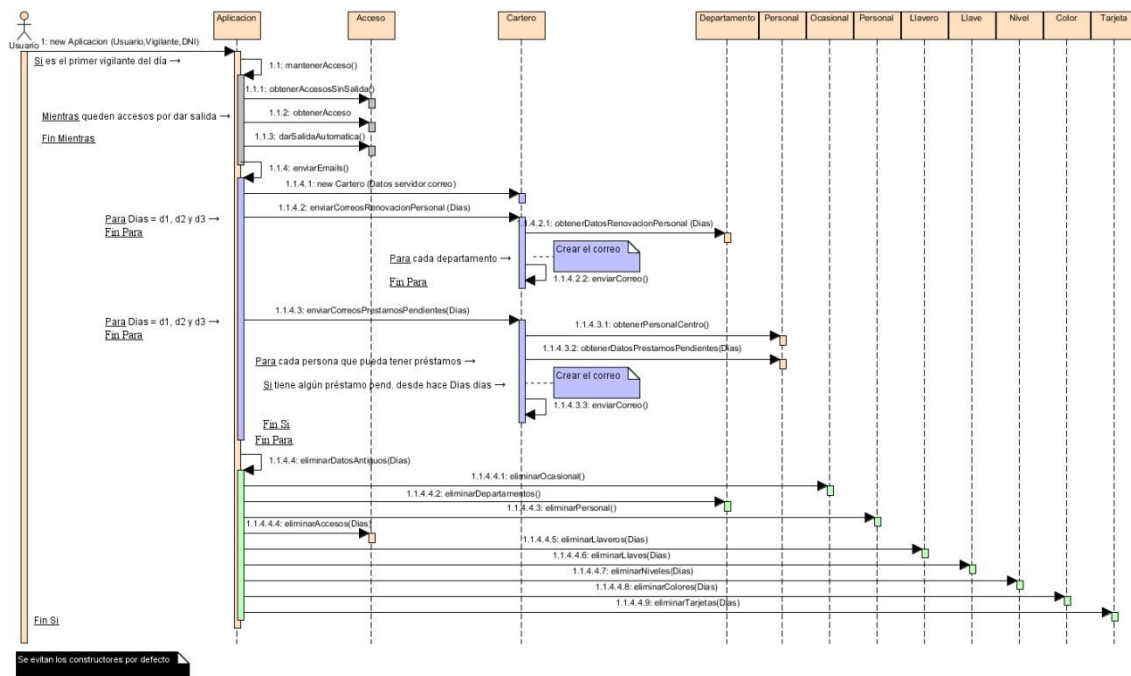


Diagrama de secuencia de la inicialización

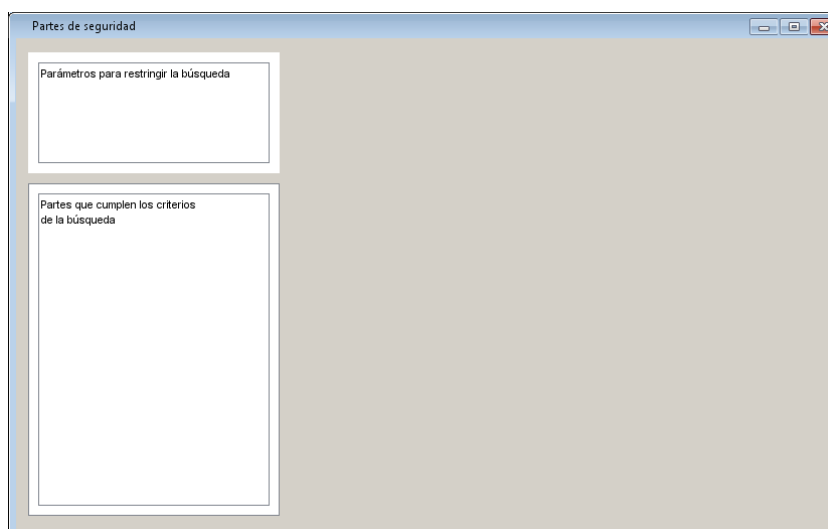
4.2. Caso de uso 120: Visualizar partes de seguridad

Pasamos ya ha desarrollar los siguientes cinco casos de usos propuestos en esta iteración, que irán a parar a la pestaña 'Partes de seguridad' y los cuales requieren un diseño en los mismos términos que veníamos haciendo hasta antes de comenzar esta iteración.

Atendiendo a las descripciones de trazo fino realizadas en la fase de análisis, queda claro que no podemos tratar de forma conjunta los casos de uso de visualización. Disponemos de toda la pestaña para buscar, visualizar y editar partes de seguridad. Al realizarse éstos en papel en estos momentos, llevaremos el símil a la aplicación, creando en la parte izquierda la búsqueda y visualización de todos los partes almacenados mientras que dejaremos la parte derecha para la visualización de cada parte.



De momento nos centramos en la búsqueda y visualización total, diseñando un panel con los parámetros que permitan definir la búsqueda y una tabla que colocaremos debajo de éste para mostrar los resultados que proporcione:



Interfaz gráfica propuesta

En un principio nos bastará con estos componentes gráficos. Los parámetros a definir serán los de código de vigilante y fecha-horas, mientras que los resultados pueden quedar definidos perfectamente por la fecha y hora en la que se redacta cada parte.

En un principio haremos que se muestren todos los partes sin tener en cuenta ningún parámetro de búsqueda, pero si el usuario define alguno deberá ejecutarse de nuevo el caso de uso, que queda definido de la siguiente forma:

1. Crear un objeto de tipo Parte_Seguridad.
2. Recoger los parámetros de búsqueda.
3. Realizar la búsqueda de todos aquellos partes que cumplan con los criterios establecidos.
4. Refrescar la tabla de partes con los resultados obtenidos.

La API no dispone de ningún método que realice la búsqueda de partes, con que debemos añadirsele:

MÉTODO	FUNCIONALIDAD
<i>obtenerPartes</i>	Método que obtiene la fecha y hora de redacción de todos aquellos partes que cumplan con las condiciones impuestas a través de los parámetros que recibe como entrada.

Método a implementar y a añadir a la API



Estos son los diagramas que definen el proceso antes mencionado:

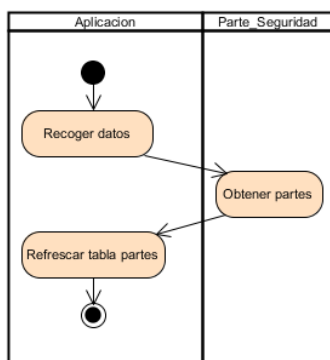


Diagrama de actividades

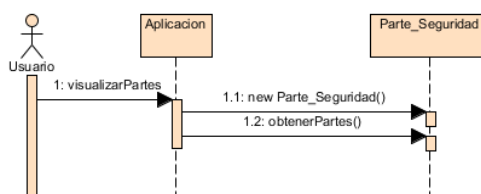
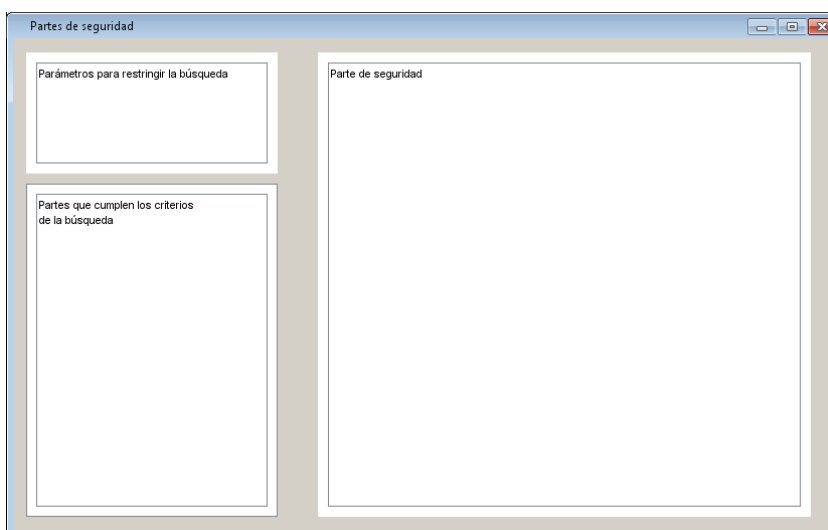


Diagrama de secuencia

4.3. Caso de uso 121: Visualizar parte de seguridad

Una vez que el usuario ya dispone de los partes de seguridad que desee listados en la JTable de la izquierda, debemos proporcionar una forma de poder visualizar todos los datos que componen cada uno de ellos. Optamos por seguir el mismo proceso que utilizamos con los accesos, diseñar un parte en lo que resta de pestaña y hacer que en él aparezcan los datos de los partes que vayamos seleccionando:



Interfaz gráfica propuesta



La lista de partes está formada por las claves primarias de éstos, con que podemos:

1. Crear un objeto de tipo Parte_Seguridad con la fecha leída de la tabla.
2. Mostrar en el parte todos los datos de los que esté compuesto el objeto antes creado.

No disponemos aún del constructor que cree un objeto Parte_Seguridad a partir de su fecha, con que es el momento perfecto para añadirlo a la API:

MÉTODO	FUNCIONALIDAD
<i>Parte_Seguridad</i>	Método que crea un objeto de tipo Parte_Seguridad obteniendo los valores de todos sus atributos de la base de datos a partir de la fecha pasada como parámetro de entrada.

Método a implementar y a añadir a la API

Completamos el diseño con este diagrama de actividades...

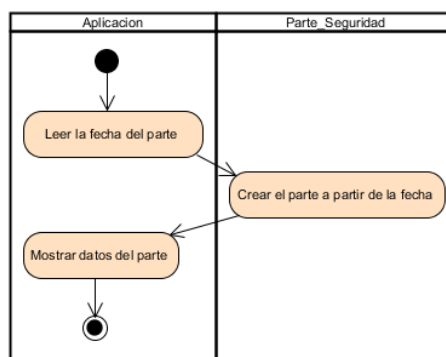


Diagrama de actividades

... y este de secuencia, que describen con un mayor grado de detalle la estrategia definida en la anterior página:

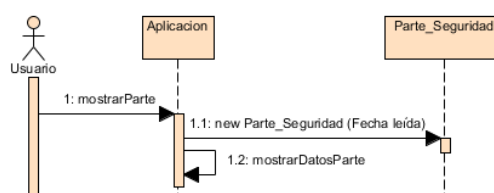
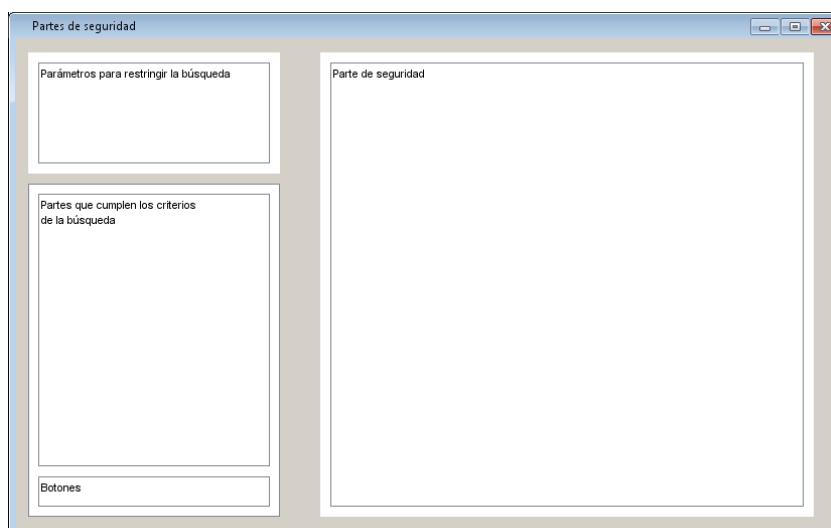


Diagrama de secuencia

4.4. Caso de uso 122: Añadir parte de seguridad

Entramos a diseñar todas las operaciones que se podrán realizar sobre un parte, lo que conlleva tener que añadir botones para poder efectuarlas. Añadimos una botonera bajo la tabla de partes:



Interfaz gráfica propuesta

Para comenzar agregamos el botón 'Añadir parte', que limpiará el parte de seguridad que se visualiza haciendo los campos editables, y activará el botón 'Grabar', que insertará o modificará el parte que se encuentre en edición. La idea es:

1. Crear un Parte_Seguridad con los datos recogidos del formulario.
2. Insertar el objeto en la base de datos mediante el método *insertParteSeguridadBD*.
3. Actualizar la tabla de partes para que muestre el parte si es preciso (si los parámetros de búsqueda están establecidos de una forma que lo permitan).

No resulta necesario modificar la API ya que disponemos de todo lo necesario para implementar esta secuencia de actividades:

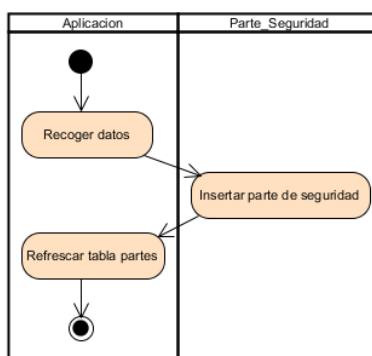
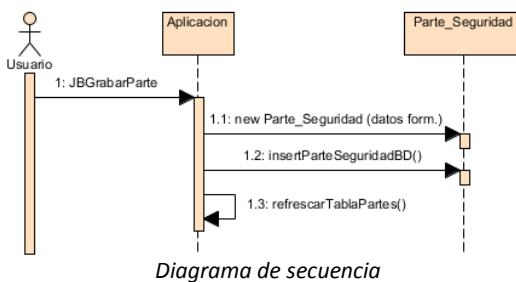


Diagrama de actividades



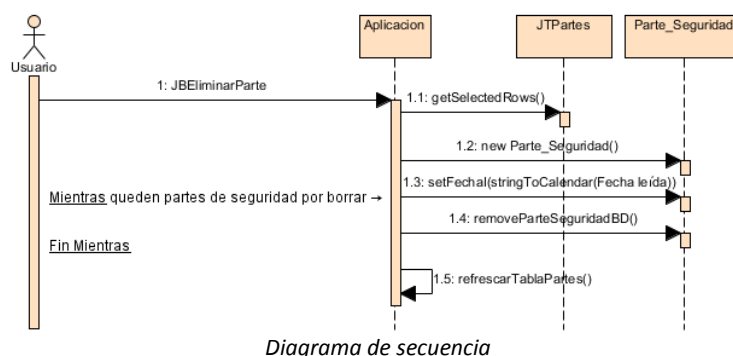
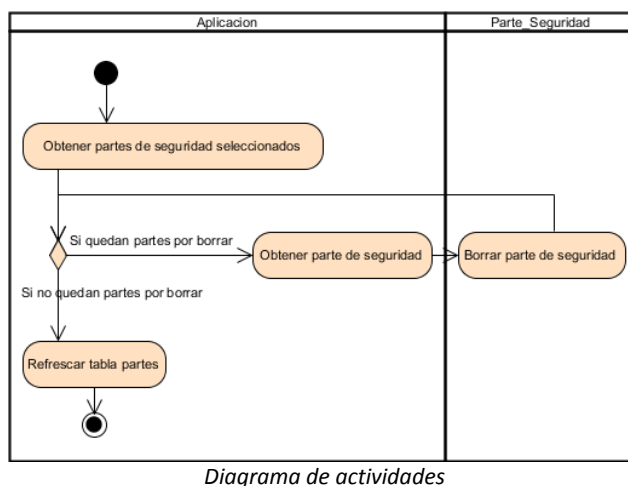
4.5. Caso de uso 123: Borrar parte de seguridad

Si añadimos un botón de borrado al panel de botones podremos...

- 1. Obtener los partes de seguridad seleccionados.
- 2. Recorrerlos y eliminarlos uno a uno.
- 3. Refrescar la tabla de partes para que se visualicen todos los que correspondan.

... y con ello tendremos el caso de uso 123 resuelto.

Estos son los diagramas ideados para la ocasión:





4.6. Caso de uso 124: Modificar parte de seguridad

Para las inserciones presionábamos un botón que borraba el formulario y otro que lo grababa, pues en el caso que nos ocupa no presionaremos ningún botón de borrado, si no que editaremos el parte que estemos visualizando en caso de ser posible (si somos el propietario del mismo y no hemos redactado más de un parte después de éste) y emplearemos el mismo botón para grabar los cambios.

La estrategia definida es:

1. Crear un parte de seguridad con los datos recogidos del formulario.
2. Recuperar el parte de seguridad a actualizar de la base de datos.
3. Actualizar el parte viejo con el nuevo.
4. Refrescar la tabla de partes ya que puede ser que se haya modificado la fecha de redacción.

Como ya disponemos del constructor que hace falta y el método de actualización implementados, con crear los diagramas de actividades y secuencia damos por concluido el diseño de este caso de uso:

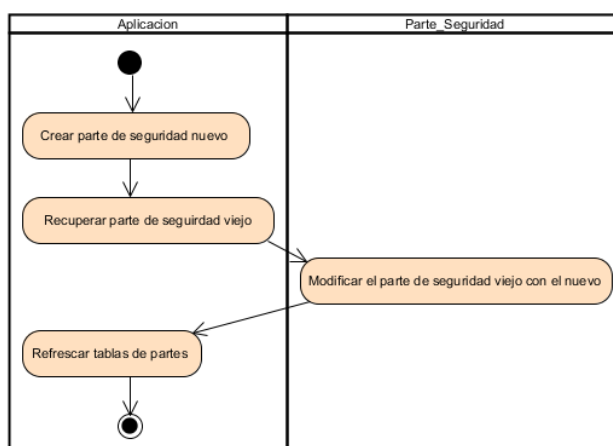


Diagrama de actividades

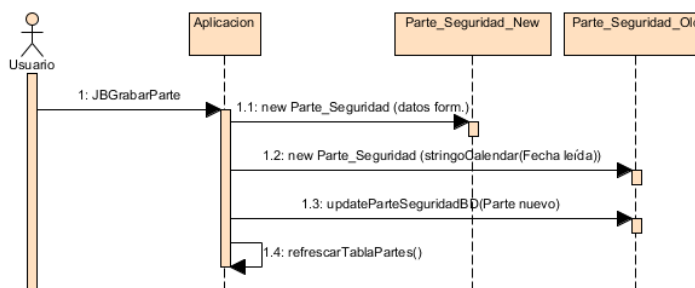


Diagrama de secuencia



4.7. Caso de uso 125: Listar parte de seguridad

Este caso es calcado al de cualquier otro que tenga por objeto crear un reporte, lo único que cambiará es la forma de generarlo, ya que en este caso no nos parece muy acertado mostrar los datos mediante una tabla y sí el llevar el parte de seguridad visualizado en pantalla al papel. Para ello:

1. Desplegaremos un JFileChooser para que el usuario indique en qué ubicación desea recibir el reporte.
2. Crearemos un objeto de tipo GeneradorInformes.
3. *Llamaremos al método generarParte(JPanelParte).*

generarParte no consta en la API, con que procede añadirlo. Éste, en lugar de generar un reporte mediante una tabla, lo hará printeando en el archivo PDF el JPanel recibido como entrada, de modo que se conseguirá el mismo efecto que el visualizado por pantalla.

MÉTODO	FUNCIONALIDAD
<i>generarParte</i>	Método que crea un reporte en formato PDF con el parte de seguridad que el usuario se encuentra visualizando por pantalla.

Método a implementar y a añadir a la API

Los diagramas de actividades y secuencia son casi calcados con respecto a los empleados en otros casos de uso de generación de reportes; lo único que cambiará es la forma de crear el informe:

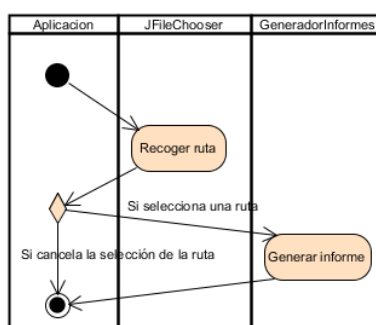


Diagrama de actividades

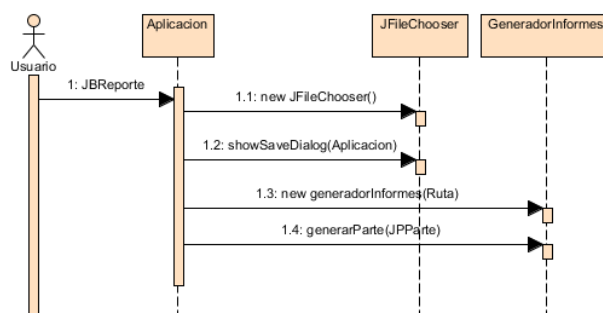


Diagrama de secuencia



5. Codificación e integración

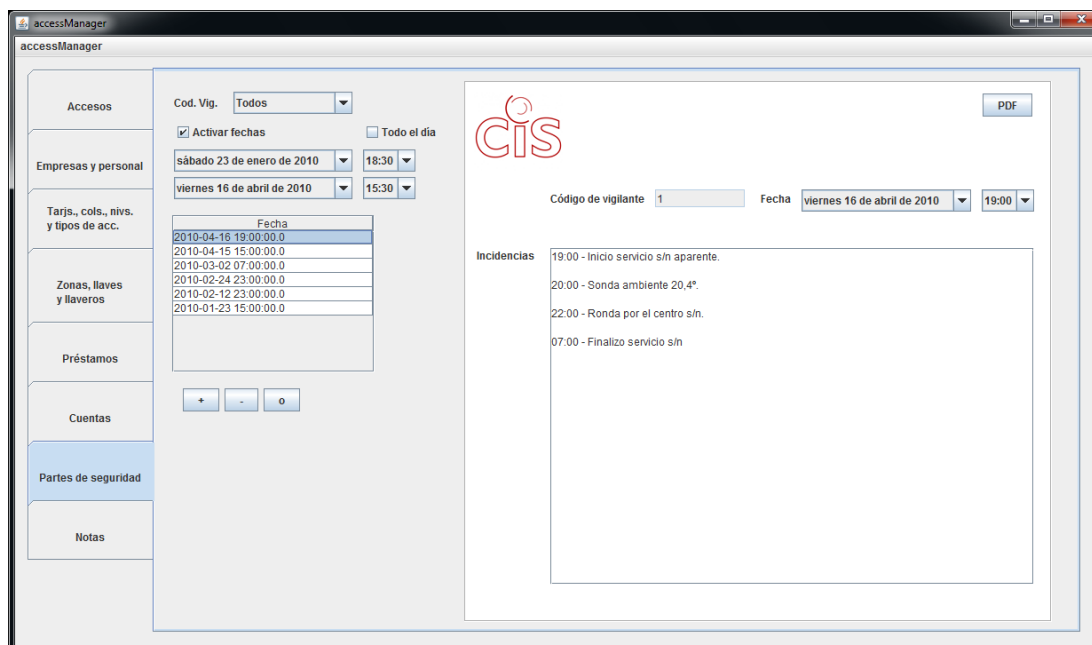
La estructura del sistema sigue sin variar, con que no merece la pena que comentemos nada. En cuanto a la codificación, no entraña más complejidad que la saber cómo y cuándo activar o desactivar los campos del parte de seguridad para que sean o no editables. Para tal fin diseñamos este método:

MÉTODO	FUNCIONALIDAD
<i>obtenerPartesEditables</i>	Método que devuelve un ArrayList con las fechas de todos aquellos partes de seguridad que puedan ser editados por el usuario.

Método a implementar y a añadir a la API

Con este método podremos saber siempre que se clique sobre un parte para visualizarlo si éste es editable, y actuar en consecuencia (activando-desactivando los campos del formulario).

La interfaz diseñada a lo largo de la fase de diseño queda materializada en un principio de la siguiente forma:



Interfaz de la pestaña 'Partes de seguridad'



6. Pruebas

6.1. De unidad

No son muchas y en un principio son sencillas de diseñar y ejecutar, salvo la primera en la que sí que tendremos que andar con más cautela.

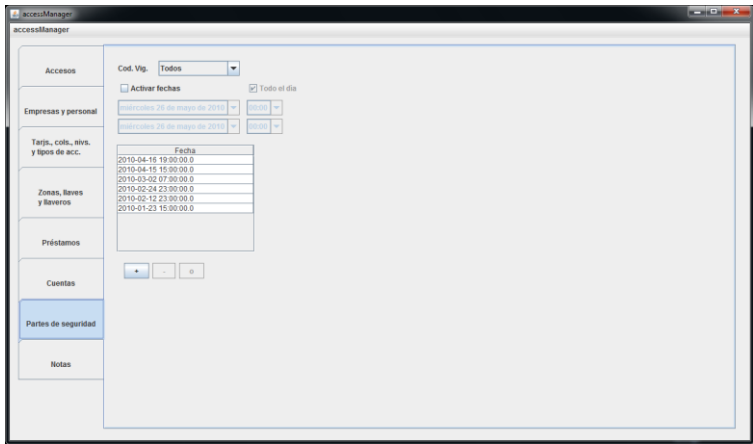
Caso de prueba	9.01 - Casos de uso 117, 118 y 119: Inicialización																																																														
Objetivo	Comprobar que en el proceso de inicialización se desarrollan todas las actividades que diseñamos, llevando a cabo las funciones para las que fueron creadas.																																																														
Actividades	Vaciar la tabla Acceso_Sistema, loguear en el sistema con una cuenta de vigilante y comprobar que todas las tareas han dado el resultado esperado.																																																														
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. • Utilidad phpMyAdmin. • Explorador web. 																																																														
Datos de entrada	<p>Modificaremos y añadiremos a la población existente la necesaria para probar este caso de uso:</p> <ul style="list-style-type: none"> • Nos hacemos responsables de personal de estos departamentos cuya fecha de última actualización ha sido modificada para que hoy debamos recibir E-Mails: <table border="1"> <thead> <tr> <th>Cod_Emp</th> <th>Cod_Dep</th> <th>Nombre</th> <th>Cod_Acc</th> <th>Periodicidad</th> <th>F_Ult_Actu</th> <th>DNI</th> </tr> <tr> <th>Código de empresa</th> <th>Código de departamento</th> <th>Nombre del departamento</th> <th>Código de acceso que posee</th> <th>Periodicidad con la que renueva la lista de personal que trabaja en el centro</th> <th>Fecha y hora de la última actualización de personal realizada</th> <th>DNI del autorizador responsable del departamento</th> </tr> </thead> <tbody> <tr> <td>CADO</td> <td>DICO</td> <td>Diseño y construcción</td> <td>THT</td> <td>60</td> <td>2010-02-25 00:00:00</td> <td>00000028A</td> </tr> <tr> <td>CIS</td> <td>VA</td> <td>Vigilancia con arma</td> <td>VIG</td> <td>365</td> <td>2009-05-11 00:00:00</td> <td>00000018A</td> </tr> <tr> <td>COMF</td> <td>DEPV</td> <td>Departamento de veritas</td> <td>THCA</td> <td>60</td> <td>2010-03-20 00:00:00</td> <td>00000043A</td> </tr> <tr> <td>CONA</td> <td>ATEC</td> <td>Asistencia Técnica</td> <td>THTA</td> <td>60</td> <td>2010-01-09 00:00:00</td> <td>00000043A</td> </tr> </tbody> </table> <ul style="list-style-type: none"> • Hacemos que se envíen correos de recordatorios de personal para la personas desde cuya fecha de préstamo hayan transcurrido 520 días, haciendo que el trabajador con DNI 00000097A, que posee la dirección de correo de prueba, deba recibir hoy avisos. • Ocasional que deberá ser eliminado: <table border="1"> <thead> <tr> <th>Empresa</th> <th>Cod_Acc</th> <th>F_Caducidad</th> <th>Notas</th> </tr> <tr> <th>Empresa para la que trabaja el trabajador ocasional</th> <th>Código de acceso proporcionado</th> <th>Fecha de caducidad del permiso para acceder al centro</th> <th>Ciños datos</th> </tr> </thead> <tbody> <tr> <td>Particular</td> <td>THTA</td> <td>2009-06-28 07:57:42</td> <td>NULL</td> </tr> <tr> <td>Particular</td> <td>MHTA</td> <td>2010-02-15 21:51:47</td> <td>NULL</td> </tr> <tr> <td>Fontaneros S.A</td> <td>THT</td> <td>2010-05-16 23:00:21</td> <td>Decirle que acuda en primer lugar a hablar con Raq...</td> </tr> </tbody> </table> <ul style="list-style-type: none"> • Empresas y personas que deberán ser eliminadas: <pre> Ejecutar la(s) consulta(s) SQL en la base de datos bd: -- Empresa sin departamentos INSERT INTO Empresa VALUES ("E", "C", "N", "D"); -- Persona sin especializacion INSERT INTO Personal VALUES ("0", "A1", "A2", "N", "0"); </pre>	Cod_Emp	Cod_Dep	Nombre	Cod_Acc	Periodicidad	F_Ult_Actu	DNI	Código de empresa	Código de departamento	Nombre del departamento	Código de acceso que posee	Periodicidad con la que renueva la lista de personal que trabaja en el centro	Fecha y hora de la última actualización de personal realizada	DNI del autorizador responsable del departamento	CADO	DICO	Diseño y construcción	THT	60	2010-02-25 00:00:00	00000028A	CIS	VA	Vigilancia con arma	VIG	365	2009-05-11 00:00:00	00000018A	COMF	DEPV	Departamento de veritas	THCA	60	2010-03-20 00:00:00	00000043A	CONA	ATEC	Asistencia Técnica	THTA	60	2010-01-09 00:00:00	00000043A	Empresa	Cod_Acc	F_Caducidad	Notas	Empresa para la que trabaja el trabajador ocasional	Código de acceso proporcionado	Fecha de caducidad del permiso para acceder al centro	Ciños datos	Particular	THTA	2009-06-28 07:57:42	NULL	Particular	MHTA	2010-02-15 21:51:47	NULL	Fontaneros S.A	THT	2010-05-16 23:00:21	Decirle que acuda en primer lugar a hablar con Raq...
Cod_Emp	Cod_Dep	Nombre	Cod_Acc	Periodicidad	F_Ult_Actu	DNI																																																									
Código de empresa	Código de departamento	Nombre del departamento	Código de acceso que posee	Periodicidad con la que renueva la lista de personal que trabaja en el centro	Fecha y hora de la última actualización de personal realizada	DNI del autorizador responsable del departamento																																																									
CADO	DICO	Diseño y construcción	THT	60	2010-02-25 00:00:00	00000028A																																																									
CIS	VA	Vigilancia con arma	VIG	365	2009-05-11 00:00:00	00000018A																																																									
COMF	DEPV	Departamento de veritas	THCA	60	2010-03-20 00:00:00	00000043A																																																									
CONA	ATEC	Asistencia Técnica	THTA	60	2010-01-09 00:00:00	00000043A																																																									
Empresa	Cod_Acc	F_Caducidad	Notas																																																												
Empresa para la que trabaja el trabajador ocasional	Código de acceso proporcionado	Fecha de caducidad del permiso para acceder al centro	Ciños datos																																																												
Particular	THTA	2009-06-28 07:57:42	NULL																																																												
Particular	MHTA	2010-02-15 21:51:47	NULL																																																												
Fontaneros S.A	THT	2010-05-16 23:00:21	Decirle que acuda en primer lugar a hablar con Raq...																																																												



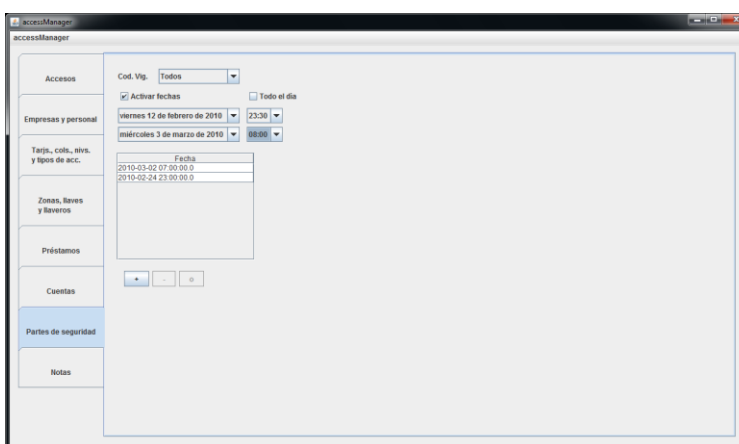
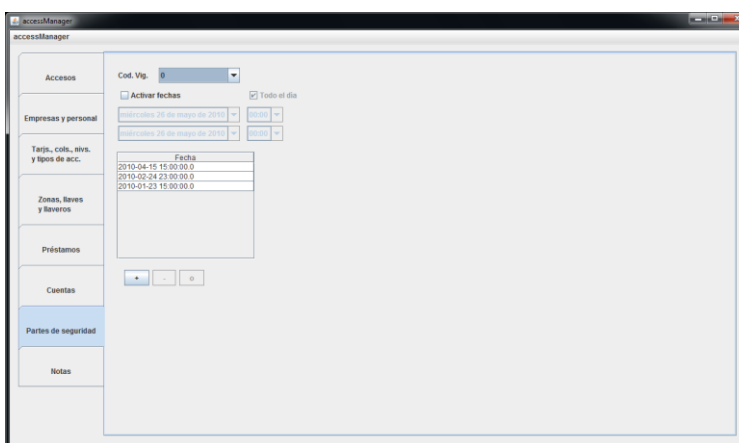
	<ul style="list-style-type: none"> Accesos que deberán ser eliminados: <table border="1" data-bbox="564 264 1300 380"> <thead> <tr> <th>DNI</th> <th>Entrada</th> <th>Salida</th> <th>Motivo</th> <th>Cod_Tarj</th> <th>F_Alta_T</th> <th>Otros_Datos</th> <th>DNI_Ins</th> </tr> <tr> <th>DNI de la persona que accede</th> <th>Fecha y hora de entrada</th> <th>Fecha y hora de salida</th> <th>Motivo de acceso</th> <th>Número de tarjeta empleada</th> <th>Fecha y hora de inserción de la tarjeta</th> <th>Otros datos de interés</th> <th>DNI del usuario que realiza la inserción del acceso (por lo general el vigilante de seguridad)</th> </tr> </thead> <tbody> <tr> <td>0000006A</td> <td>2008-12-22 13:12:05</td> <td>NULL</td> <td>Trabajo</td> <td>39</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>0000001A</td> </tr> <tr> <td>00000045A</td> <td>2008-12-22 20:00:00</td> <td>NULL</td> <td>Trabajo</td> <td>10</td> <td>2010-01-01 00:00:01</td> <td>NULL</td> <td>0000002A</td> </tr> </tbody> </table> Llaveros, llaves, niveles, colores y tarjetas que deberán ser eliminadas: <pre data-bbox="564 488 1244 600"> Ejecutar la(s) consulta(s) SQL en la base de datos bd: @ -- Llavero viejo INSERT INTO Llavero VALUES (005,'2007-01-01 00:00:01','2008-01-01 00:00:01','D'); -- Llave vieja INSERT INTO Llave VALUES (999,'2007-01-01 00:00:01','2008-01-01 00:00:01','GRUPELEC','2010-01-01 00:00:01','D',null); Ejecutar la(s) consulta(s) SQL en la base de datos bd: @ -- Nivel viejo INSERT INTO Nivel VALUES (2,'2007-01-01 00:00:01','2008-01-01 00:00:01','D'); -- Color viejo INSERT INTO Color VALUES (-2,'2007-01-01 00:00:01','2008-01-01 00:00:01',2,'2007-01-01 00:00:01'); Ejecutar la(s) consulta(s) SQL en la base de datos bd: @ -- Tarjeta vieja INSERT INTO Tarjeta VALUES ('999','2007-01-01 00:00:01','2008-01-01 00:00:01',-2,'2007-01-01 00:00:01',null); </pre> 	DNI	Entrada	Salida	Motivo	Cod_Tarj	F_Alta_T	Otros_Datos	DNI_Ins	DNI de la persona que accede	Fecha y hora de entrada	Fecha y hora de salida	Motivo de acceso	Número de tarjeta empleada	Fecha y hora de inserción de la tarjeta	Otros datos de interés	DNI del usuario que realiza la inserción del acceso (por lo general el vigilante de seguridad)	0000006A	2008-12-22 13:12:05	NULL	Trabajo	39	2010-01-01 00:00:01	NULL	0000001A	00000045A	2008-12-22 20:00:00	NULL	Trabajo	10	2010-01-01 00:00:01	NULL	0000002A
DNI	Entrada	Salida	Motivo	Cod_Tarj	F_Alta_T	Otros_Datos	DNI_Ins																										
DNI de la persona que accede	Fecha y hora de entrada	Fecha y hora de salida	Motivo de acceso	Número de tarjeta empleada	Fecha y hora de inserción de la tarjeta	Otros datos de interés	DNI del usuario que realiza la inserción del acceso (por lo general el vigilante de seguridad)																										
0000006A	2008-12-22 13:12:05	NULL	Trabajo	39	2010-01-01 00:00:01	NULL	0000001A																										
00000045A	2008-12-22 20:00:00	NULL	Trabajo	10	2010-01-01 00:00:01	NULL	0000002A																										
Salida esperada	Todas las tuplas diseñadas para que cumplan las condiciones de borrado deberán ser eliminadas, así como enviados todos los E-Mails de aviso que deban.																																
Salida observada	<p>La esperada. Los registros han sido eliminados...</p> <pre data-bbox="564 1064 861 1288"> AVISO: 4 fila(s) eliminada(s) AVISO: 41 fila(s) eliminada(s) AVISO: 0 fila(s) eliminada(s) AVISO: 0 fila(s) eliminada(s) AVISO: 25 fila(s) eliminada(s) AVISO: 1 fila(s) eliminada(s) AVISO: 1 fila(s) eliminada(s) AVISO: 1 fila(s) eliminada(s) AVISO: 1 fila(s) eliminada(s) AVISO: 1 fila(s) eliminada(s) AVISO: 0 fila(s) eliminada(s) </pre> <p>... y los correos electrónicos enviados:</p> <table border="1" data-bbox="564 1388 1292 1467"> <tbody> <tr> <td>informaticanavarra@g</td> <td>Prestamos pendientes de devolución con 'Informática Navarra S.A.'</td> <td>14:47</td> </tr> <tr> <td>informaticanavarra@g</td> <td>Actualización de personal para el centro de 'Informática Navarra S.A.'</td> <td>14:47</td> </tr> <tr> <td>informaticanavarra@g</td> <td>Actualización de personal para el centro de 'Informática Navarra S.A.'</td> <td>14:47</td> </tr> <tr> <td>informaticanavarra@g</td> <td>Actualización de personal para el centro de 'Informática Navarra S.A.'</td> <td>14:47</td> </tr> </tbody> </table>	informaticanavarra@g	Prestamos pendientes de devolución con 'Informática Navarra S.A.'	14:47	informaticanavarra@g	Actualización de personal para el centro de 'Informática Navarra S.A.'	14:47	informaticanavarra@g	Actualización de personal para el centro de 'Informática Navarra S.A.'	14:47	informaticanavarra@g	Actualización de personal para el centro de 'Informática Navarra S.A.'	14:47																				
informaticanavarra@g	Prestamos pendientes de devolución con 'Informática Navarra S.A.'	14:47																															
informaticanavarra@g	Actualización de personal para el centro de 'Informática Navarra S.A.'	14:47																															
informaticanavarra@g	Actualización de personal para el centro de 'Informática Navarra S.A.'	14:47																															
informaticanavarra@g	Actualización de personal para el centro de 'Informática Navarra S.A.'	14:47																															
Evaluación/ conclusión	Los casos de uso '117, 118 y 119 - Inicialización' han sido realizados satisfactoriamente.																																

Caso de prueba para los casos de uso '117, 118 y 119 - Inicializaciones'



Caso de prueba	9.02 - Caso de uso 120: Visualizar partes de seguridad														
Objetivo	Probar que tanto el administrador como el vigilante pueden observar los partes de seguridad almacenados en el sistema que cumplen con ciertas características (parámetros de búsqueda).														
Actividades	Loguear en el sistema, acceder a la tab 'Partes de seguridad' y realizar búsquedas sobre éstos, comprobando que siempre reflejan la realidad de la base de datos.														
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos con población de prueba.• Utilidad phpMyAdmin.														
Datos de entrada	Realizaremos búsquedas sobre los partes variando el redactor y las fechas, comprobando que todo funciona correctamente.														
Salida esperada	<p>Que la tabla siempre contenga los partes que existen en la base de datos. En un principio deberán observarse todos ...</p> <table border="1"><thead><tr><th>Fecha</th><th>Fecha y hora del servicio</th></tr></thead><tbody><tr><td>2010-01-23</td><td>15:00:00</td></tr><tr><td>2010-02-12</td><td>23:00:00</td></tr><tr><td>2010-02-24</td><td>23:00:00</td></tr><tr><td>2010-03-02</td><td>07:00:00</td></tr><tr><td>2010-04-15</td><td>15:00:00</td></tr><tr><td>2010-04-16</td><td>19:00:00</td></tr></tbody></table> <p>... y luego realizaremos variaciones con todos los parámetros.</p>	Fecha	Fecha y hora del servicio	2010-01-23	15:00:00	2010-02-12	23:00:00	2010-02-24	23:00:00	2010-03-02	07:00:00	2010-04-15	15:00:00	2010-04-16	19:00:00
Fecha	Fecha y hora del servicio														
2010-01-23	15:00:00														
2010-02-12	23:00:00														
2010-02-24	23:00:00														
2010-03-02	07:00:00														
2010-04-15	15:00:00														
2010-04-16	19:00:00														
Salida observada	<p>La esperada:</p> <ul style="list-style-type: none">• En un principio: 														

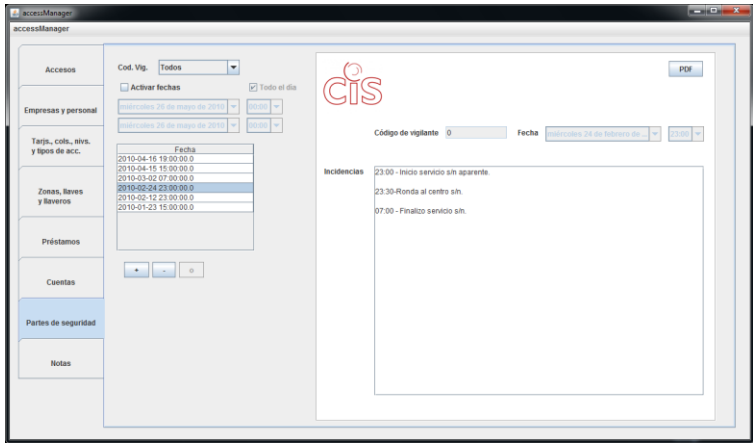
- Algunas variaciones:



**Evaluación/
conclusión**

El caso de uso '120 - Visualizar partes de seguridad' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '120 - Visualizar partes de seguridad'

Caso de prueba	9.03 - Caso de uso 121: Visualizar parte de seguridad
Objetivo	Verificar que es posible realizar el visionado de los partes existentes en la tabla. Además, certificar que los partes se hacen editables sólo cuando deben.
Actividades	Loguear en el sistema tanto como administrador como vigilante (son los dos roles que tendrán acceso a esta sección) y realizar el visionado de todos los partes oportunos, comprobando que el sistema se comporta como debe.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba.
Datos de entrada	<p>Logueando como administrador:</p> <ul style="list-style-type: none"> • Clicar sobre un parte de cada vigilante y comprobar que se puede visualizar el contenido, pero no editarlo. Partes: '2010-01-23 15:00:00' (vig. 0), '2010-02-12 23:00:00' (vig. 1) y '2010-03-02 07:00:00' (vig. 2). <p>Accediendo como vigilante (vigilante con código 0, para este caso de prueba):</p> <ul style="list-style-type: none"> • Clicar sobre un parte de cada vigilante y comprobar que se puede visualizar el contenido, pero no editarlo. Partes: '2010-02-12 23:00:00' (vig. 1) y '2010-03-02 07:00:00' (vig. 2). • Clicar sobre los partes propios, debiendo ser editables solo los dos últimos.
Salida esperada	La descrita en 'Datos de entrada': el administrador solo podrá visionar, mientras que el vigilante podrá visionar y editar los dos últimos partes redactados ('2010-02-24 23:00:00' y '2010-04-15 15:00:00').
Salida observada	<p>La esperada:</p> <ul style="list-style-type: none"> • De parte del administrador se pueden visualizar todos pero no editarlos:  <p>The screenshot shows the 'accessManager' web application. On the left, there is a navigation menu with options like 'Accesos', 'Empresas y personal', 'Tarifa, coll., niv. y tipos de acc.', 'Zonas, Seves y Severos', 'Préstamos', 'Cuentas', 'Partes de seguridad', and 'Notas'. The 'Partes de seguridad' option is selected. The main content area displays a table of security parts with columns for 'Fecha' and 'Código de vigilante'. The table lists several parts with dates and codes. Below the table, there are controls for 'Activar fechas' and 'Todo el día'. On the right side, there is a section for 'Incidencias' with a list of incidents and their details, including times and descriptions.</p>



- De parte del vigilante...

... los pertenecientes a otros vigilantes o antiguos no se pueden editar ni eliminar:

accessManager

Accesos: Col. Vp. Todos

Empresas y personal: Activar fechas

Tarj., col., niv. y tipos de acc.:

Fecha
2010-04-16 19:00:00
2010-04-15 15:00:00
2010-03-20 07:00:00
2010-02-24 23:00:00
2010-02-12 23:00:00
2010-01-23 15:00:00

Partes de seguridad

Código de vigilante: 2 Fecha: martes 2 de marzo de 2010

Incidencias:

- 07:00 - Inicio servicio sin aparente
- 13:00 - 00000117A comunica que han sido conectados armarios de LIPS a la central de alarmas.
- 15:00 - Finalizo servicio sin

accessManager

Accesos: Col. Vp. Todos

Empresas y personal: Activar fechas

Tarj., col., niv. y tipos de acc.:

Fecha
2010-04-16 19:00:00
2010-04-15 15:00:00
2010-03-20 07:00:00
2010-02-24 23:00:00
2010-02-12 23:00:00
2010-01-23 15:00:00

Partes de seguridad

Código de vigilante: 0 Fecha: sábado 23 de febrero de 2010

Incidencias:

- Abro y cierro el servicio sin incidencias.

... pero los dos últimos sí:

accessManager

Accesos: Col. Vp. Todos

Empresas y personal: Activar fechas

Tarj., col., niv. y tipos de acc.:

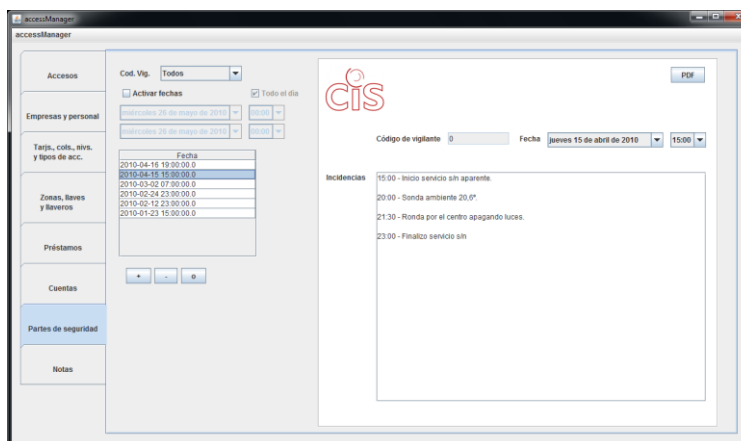
Fecha
2010-04-16 19:00:00
2010-04-15 15:00:00
2010-03-20 07:00:00
2010-02-24 23:00:00
2010-02-12 23:00:00
2010-01-23 15:00:00

Partes de seguridad

Código de vigilante: 0 Fecha: miércoles 24 de febrero de 2010 23:00

Incidencias:

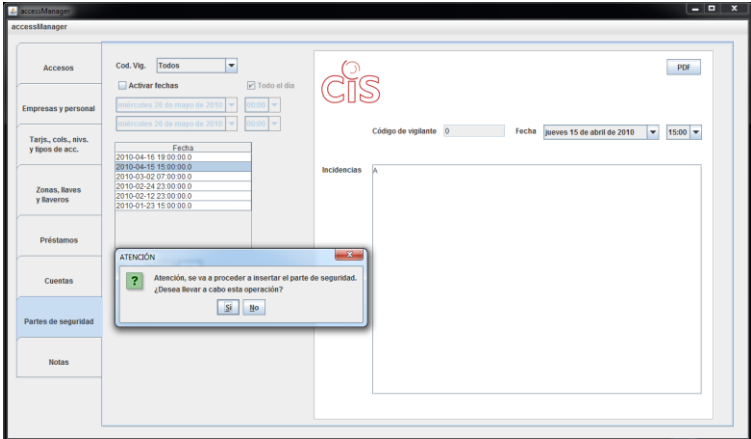
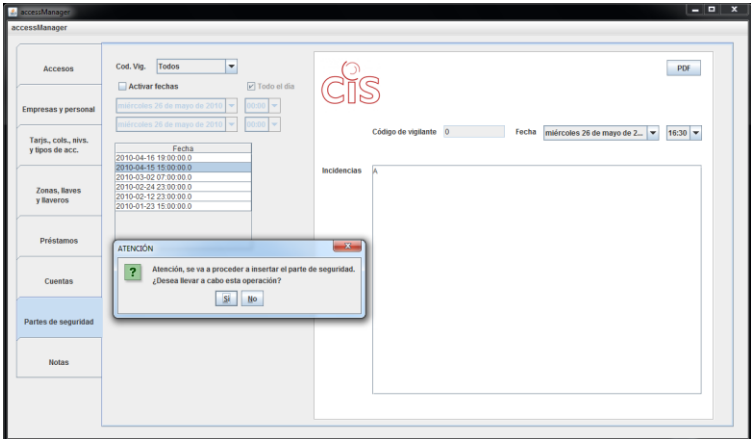
- 23:00 - Inicio servicio sin aparente
- 23:30 - Ronda al centro sin.
- 07:00 - Finalizo servicio sin.



**Evaluación/
conclusión**

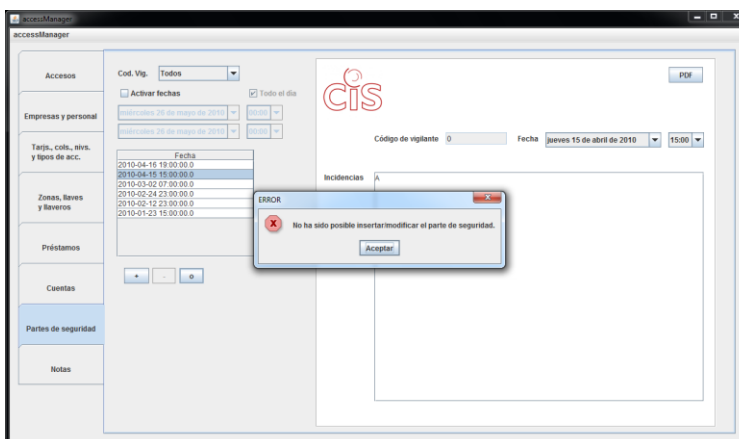
El caso de uso '121 - Visualizar parte de seguridad' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '121 - Visualizar parte de seguridad'

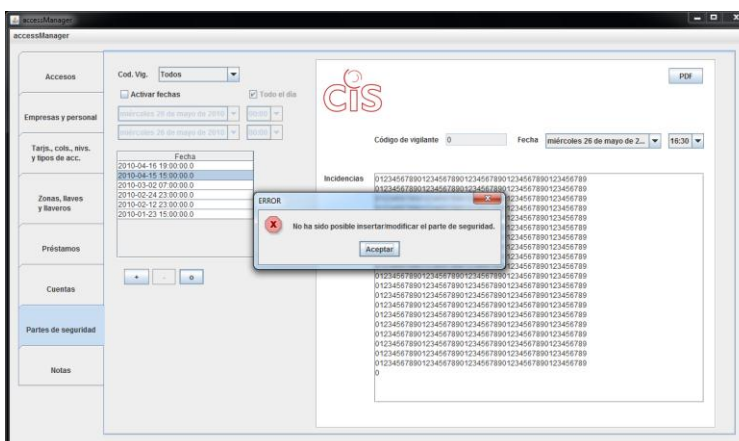
Caso de prueba	9.04 - Caso de uso 122: Añadir parte de seguridad
Objetivo	Chequear que el vigilante de seguridad puede introducir cuando desee un nuevo parte en el sistema, siendo rechazado únicamente si se introducen dos con la misma fecha o si la longitud de 'Incidencias' es mayor de 1000.
Actividades	Acceder a accessManager como vigilante y probar a insertar un parte válido y otros dos erróneos.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba.
<p>Datos de entrada</p>	<p>Probaremos a insertar los partes:</p> <ul style="list-style-type: none"> • '2010-04-15 15:00:00, A'. (Existente)  <ul style="list-style-type: none"> • '2010-05-26 15:00:00,0123456...'. (Formato no aceptable) • '2010-05-26 16:30:00,A'. (Parte válido) 
Salida esperada	Los dos primeros intentos deberían dar lugar a mensajes de error por parte del sistema, sin que llegue a producirse ninguna inserción, mientras que el tercer intento sí que debería pasar a formar parte de la base de datos y hacer que el parte con fecha '2010-02-24 23:00:00' fuera no-editable.

La espera:

- Si insertamos un parte existente:

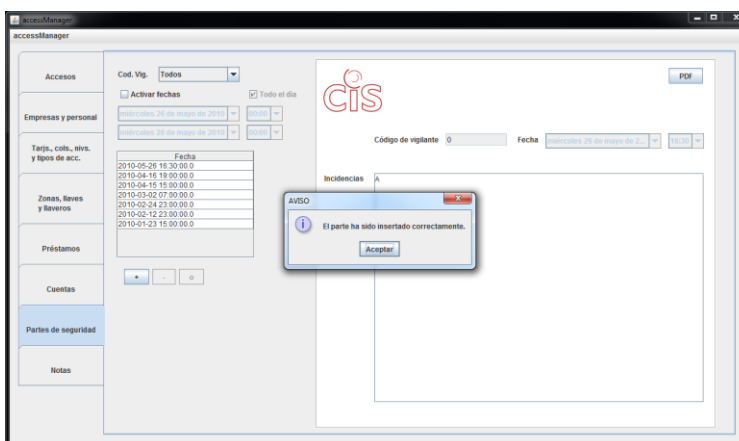


- Si lo que hacemos es insertar un parte con datos no válidos:



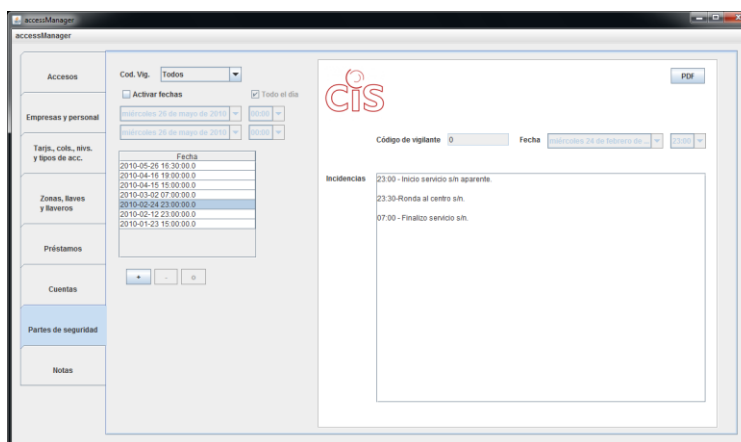
Salida observada

- La inserción de un parte válido provoca la inserción de éste en la base de datos...





... y que el que era hasta entonces el penúltimo parte pase a ser no-editable:



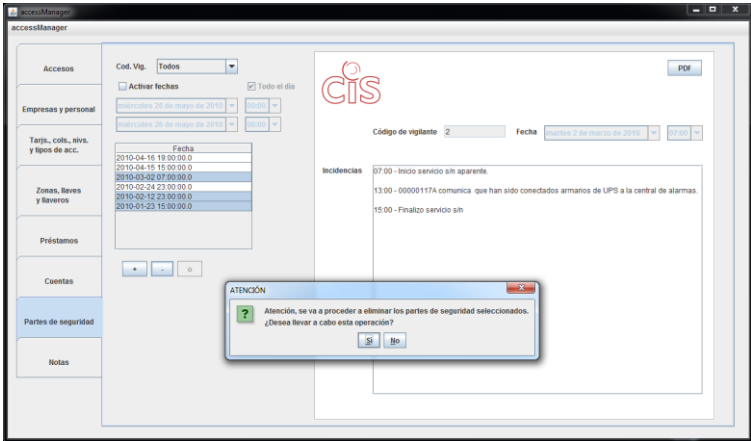
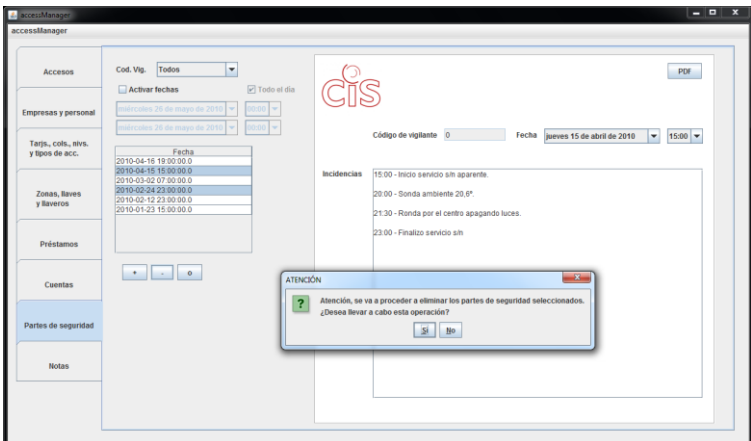
Nota: El establecimiento de la fecha por defecto a la hora de crear un nuevo parte es correcto.

**Evaluación/
conclusión**

El caso de uso '122 - Añadir parte de seguridad' ha sido realizado satisfactoriamente.

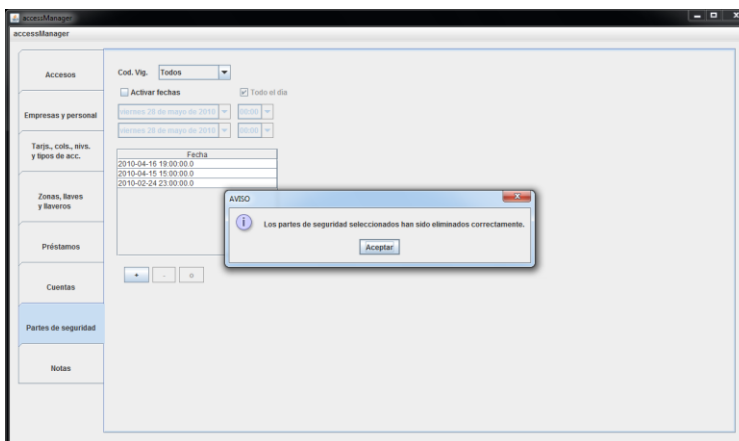
Caso de prueba para el caso de uso '122 - Añadir parte de seguridad'



Caso de prueba	9.05 - Caso de uso 123: Borrar parte de seguridad
Objetivo	Asegurar que el administrador podrá borrar cualquier conjunto de partes que se proponga, mientras que el vigilante solo podrá hacerlo en las mismas condiciones que las puestas para la edición.
Actividades	Abrir la aplicación desde una cuenta de administrador y otra de vigilante (la del que tiene el código de vigilante 0 en este caso) y probar que se pueden eliminar solo los partes permitidos.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos con población de prueba.
Datos de entrada	<p>Accediendo como administrador eliminaremos un parte de cada vigilante:</p>  <p>Al hacer la prueba como vigilante probaremos a intentar borrar partes ajenos, un conjunto de partes propios que contengan editables y no editables y otro conjunto con solo editables:</p> 
Salida esperada	Como administrador deberíamos poder borrar cualquier conjunto de partes, mientras que como vigilante solo debería funcionar el último caso.

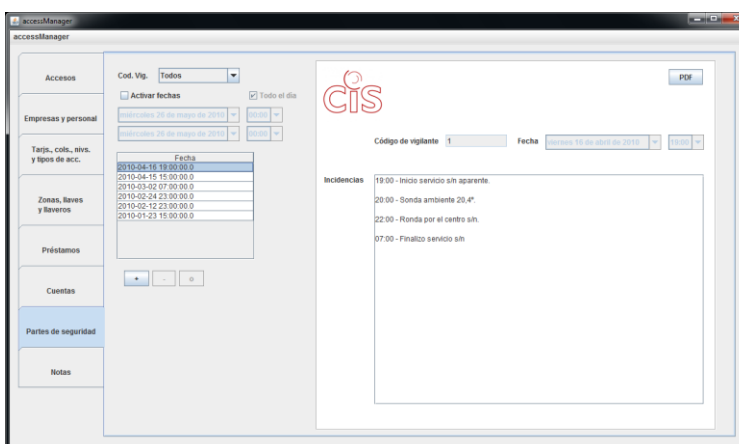
La esperada:

- Administrador:

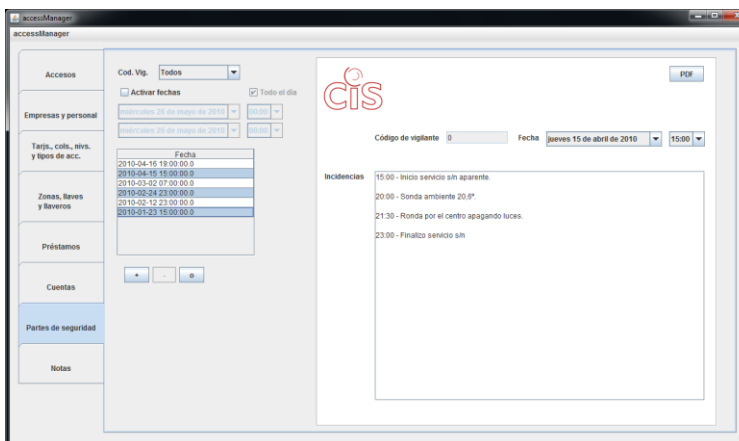


- Vigilante:
En las partes ajenos no disponemos activo el botón de borrado:

Salida observada

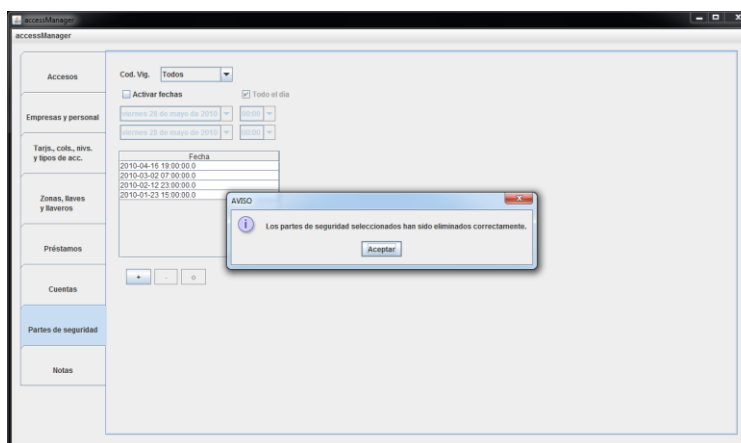


- En los partes propios:
Si no todos son editables tampoco disponemos activado el botón de borrado...





... pero si lo son sí que podemos realizar el borrado:

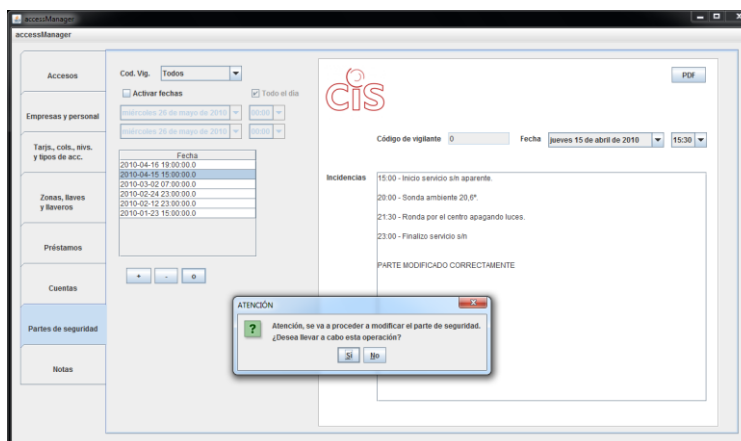


**Evaluación/
conclusión**

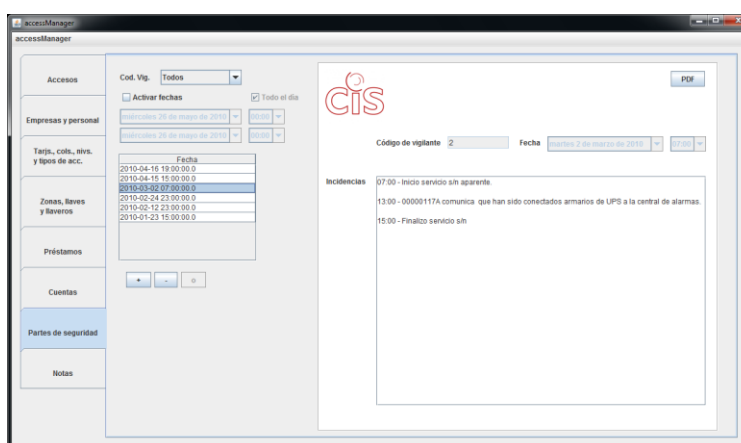
El caso de uso '123 - Borrar parte de seguridad' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '123 - Borrar parte de seguridad'

Caso de prueba	9.06 - Caso de uso 124: Modificar parte de seguridad
Objetivo	Verificar que un administrador nunca podrá modificar ningún parte, en cambio, los vigilantes sí que podrán siempre que lo intenten en partes editables (los dos últimos partes redactados).
Actividades	Entrar en accessManager como administrador y comprobar que no se puede modificar ninguno de los partes existentes. Después, acceder como vigilante y comprobar que las modificaciones siempre se llevan a cabo (siempre que se pueda editar el parte, claro está).
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba.
Datos de entrada	<p>En primer lugar logueamos en el sistema como administrador, vamos a la pestaña de partes e intentamos modificar alguno.</p> <p>Tras realizar todas las pruebas con el usuario 'Administrador', cambiamos de rol y accedemos como 'Vigilante' (de nuevo con la cuenta del vigilante cuyo código es el 0) y probamos a editar un parte anterior al penúltimo redactado y el último o penúltimo.</p>
Salida esperada	<p>Los intentos del administrador deberán ser en balde, ya que no es que no pueda grabar la modificación de los partes, es que no tiene la posibilidad ni de comenzar a editarlo al ser los campos no-editables y además estar el botón 'Grabar parte' desactivado.</p> <p>Al realizar las pruebas con el vigilante veremos que obtenemos la misma respuesta que el administrador con los partes no editables, pero con los que sí lo sean podremos modificar los campos que deseemos y actualizarlos siempre que no impliquen un parte repetido o con longitud de 'Incidencias' mayor de 1000, tal y como ocurría con la inserción:</p> <div data-bbox="563 1173 1315 1615" data-label="Image"> </div>

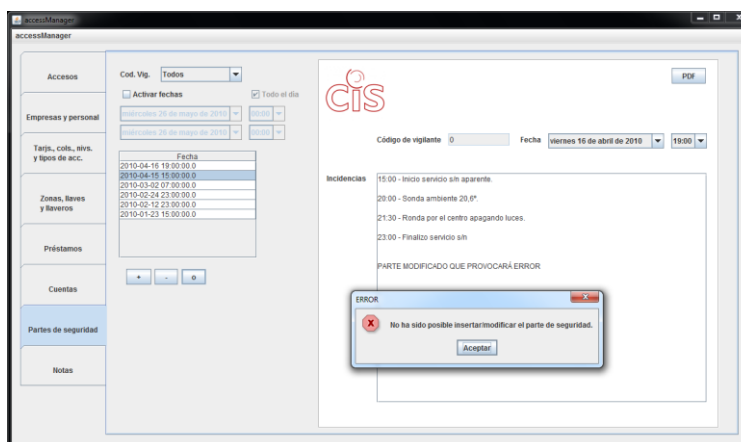


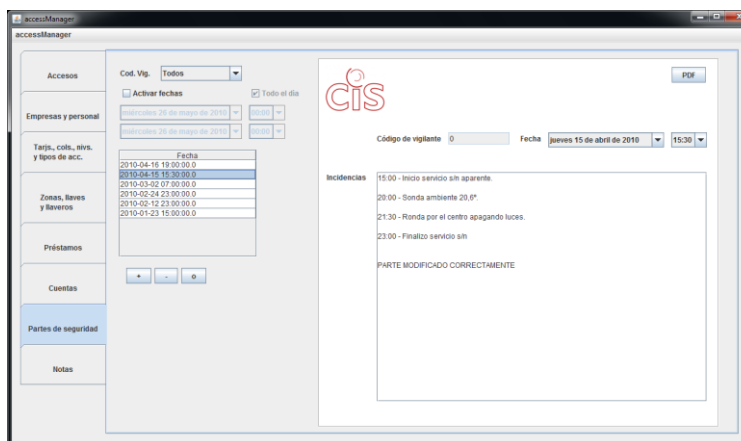
La espera: el administrador no puede editar ningún parte...



... mientras que si el vigilante lo prueba con alguno de sus partes editables, los cambios sí tienen lugar si se intenta una actualización válida:

Salida observada



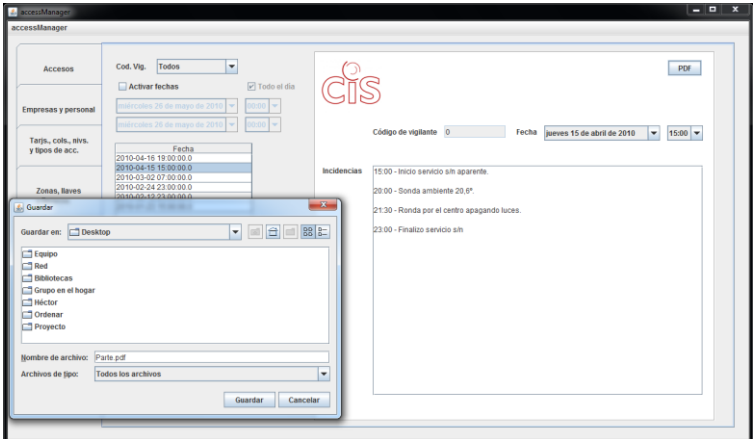


**Evaluación/
conclusión**

El caso de uso '124 - Modificar parte de seguridad' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '124 - Modificar parte de seguridad'



Caso de prueba	9.07 - Caso de uso 125: Listar parte de seguridad
Objetivo	Comprobar que tanto administrador como vigilante pueden obtener un reporte en formato PDF del parte de seguridad que se encuentren visualizando.
Actividades	Desplegar accessManager, ir a la sección de partes, seleccionar uno y pedir al sistema que elabore un reporte con éste, percatándonos de que el resultado final es el esperado.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos con población de prueba.• Aplicación Adobe Reader.
Datos de entrada	<p>Nos dirigimos a la tab 'Partes de seguridad', escogemos el que tiene fecha '2010-04-15 15:00:00' y pedimos al sistema que genere un informe con éste:</p>  <p>The screenshot shows the 'accessManager' interface. On the left, there are filters for 'Accesos', 'Empresas y personal', 'Tarifa, coils, elev.', and 'Zonas, Baves'. The main area displays a table of incidents with columns for 'Fecha' and 'Incidencias'. A 'Guardar' dialog box is open in the foreground, showing the file name 'Parte.pdf' and the location 'Desktop'.</p>
Salida esperada	Tras finalizar el caso de uso deberá existir en la ruta que hayamos indicado un archivo PDF que contenga el parte de seguridad que estábamos visionando a la hora de realizar la petición de dicho reporte.

<p>Salida observada</p>	<p>La esperada. Acudimos a la ubicación que indicamos y encontramos un archivo PDF con el siguiente contenido:</p> <div data-bbox="561 295 970 819" data-label="Image"> </div>
<p>Evaluación/ conclusión</p>	<p>El caso de uso '125 - Listar parte de seguridad' ha sido realizado satisfactoriamente.</p>

Caso de prueba para el caso de uso '125 - Listar parte de seguridad'

6.2. De integración

Penúltima pestaña que tratamos y el sistema sigue sin crecer en lo que a componentes se refiere, pero sí que lo hace la dimensión de los existentes. Carece de total interés realizar cualquier tipo de prueba de integración; hasta el inicio de la iteración todos los componentes han interactuado correctamente y tras terminar ésta siguen comportándose igual.

6.3. De sistema

La especificación de requisitos del software queda probada desde el mismo momento en que se desarrollan y prueban con éxito todos los casos de uso propuestos para esta iteración. El funcionamiento de accessManager sigue siendo correcto, robusto y la documentación la adecuada.



7. Evaluación de la iteración

Terminamos una iteración que si bien es verdad que en un principio no se encontraba entre las imprescindibles, el desarrollo de la misma va a facilitar muchísimo el trabajo de los vigilantes y que el administrador pueda estar informado constantemente de todo lo que suceda en el centro.

Hemos conseguido en tres días un sistema de edición de partes sencillo pero a la vez muy eficaz, llevando todos los casos de uso a buen puerto y de una manera satisfactoria a tenor de lo que las pruebas muestran.

Creemos que todo está listo para que el cliente dé “carpetazo” a esta iteración, con que podemos pasar a la reunión de despliegue con éste y ver qué impresiones tiene.

8. Despliegue

Nos citamos con el cliente en la sede de Informática Navarra S.A. para llevar a cabo la actividad de despliegue. Acudimos a ésta provistos de este documento y el prototipo actual instalado en un ordenador portátil; tras darnos el cliente la bienvenida comenzamos con la explicación. Exponemos el diseño que hemos realizado en la pestaña ‘Partes de seguridad’, llevando a cabo la prueba de cada caso de uso desde los roles de administrador y vigilante de seguridad. Al finalizar las pruebas el cliente nos hace saber que se encuentra muy gratamente sorprendido, ya que le ha gustado mucho cómo hemos plasmado la realidad en la aplicación, cómo hemos llevado la manipulación de los partes de seguridad en la aplicación de forma casi calcada a como se hace hoy en día manualmente. Es muy consciente de la facilidad de operación que éste va a suponer tanto para él como para el conjunto de vigilantes, se encuentra muy satisfecho y ante la pregunta de si ve necesario incorporar o cambiar algún aspecto nos contesta que en un principio no tiene ninguna pega, que ve todo bien y que podemos continuar con el desarrollo en cuanto lo estimemos oportuno.

Pamplona, a 26 de marzo de 2010.

Héctor Napal Basterrica
Responsable del proyecto

Pedro Martínez Pérez
*Jefe de área de producción de
Informática Navarra S.A.*



accessManager
Desarrollo de un sistema de control de accesos a un centro

Iteración C10: Notas

Fecha de entrega: 31 de marzo de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Estructura del documento

2. Estado del proyecto

3. Análisis

- 3.1. Planificación de la iteración
- 3.2. Análisis de los casos de uso y escenarios

4. Diseño

- 4.1. Caso de uso 126: Redactar nota
- 4.2. Casos de uso 127 y 130: Visualizar notas recibidas y enviadas
- 4.3. Casos de uso 128 y 131: Visualizar nota recibida/enviada
- 4.4. Casos de uso 129 y 132: Borrar nota recibida/enviada

5. Codificación e integración

6. Pruebas

- 6.1. De unidad
- 6.2. De integración
- 6.3. De sistema

7. Evaluación de la iteración

8. Despliegue



1. Introducción

1.1. Propósito

Recoger todo el proceso de desarrollo realizado a lo largo de la décima iteración de construcción. De esta forma documentamos todo el análisis, diseño, codificación y pruebas que han sido necesarias para implementar los casos de uso del 126 al 132, que describen toda la funcionalidad que deberá tener la aplicación para permitir la comunicación inter-usuario a través de ella.

1.2. Audiencia

Al igual que el resto de artefactos, éste documento puede considerarse de carácter interno hasta la segunda iteración de la fase de Transición, donde deberá ser entregado junto al resto como parte de la documentación del sistema. A partir de entonces pasará a ser un documento en manos del cliente y que tendrá un gran valor a la hora de realizar las futuras actividades de mantenimiento.

1.3. Estructura del documento

En la segunda sección de este documento hacemos un breve inciso en el estado del proyecto que nos encontramos y qué objetivos persigue esta iteración. A continuación dedicamos una sección al análisis de ésta, planificándola detalladamente y realizando las descripciones de trazo fino de los casos de uso que diseñaremos en el siguiente apartado, el 4. En el quinto capítulo mencionamos todos los detalles que estimamos convenientes en la codificación e integración del sistema, para después pasar a explicar todo el desarrollo de las actividades de prueba realizadas en el sexto. Concluimos este artefacto realizando una evaluación de todo el trabajo realizado y las conclusiones extraídas de la actividad de despliegue.

2. Estado del proyecto

Nos embarcamos en el desarrollo de la última pestaña que creamos al inicio de la etapa de Construcción: la de notas. Las anteriores iteraciones han sido completadas de manera satisfactoria y lo que la planificación del proyecto manda en este momento es desarrollar los casos de uso concernientes a esta pestaña.



3. Análisis

3.1. Planificación de la iteración

Una vez seleccionados los casos de uso procede realizar una planificación algo más detallada de la que tenemos desde la fase de Elaboración; debemos refinarla, desmenuzarla.

PLAN DE ITERACIÓN C10: NOTAS			
ASPECTO	DESCRIPCIÓN		
Objetivos	Desarrollar toda la funcionalidad que debe proporcionar la pestaña 'Notas', lo que se traduce en implementar todos los casos de uso que acordamos con el cliente y que describen todo el sistema de intercomunicación entre usuarios.		
Artefactos de entrada	<ul style="list-style-type: none"> • Especificación inicial de los casos de uso • Planificación temporal y costes detallados del proyecto • Base de datos, API y aplicación desarrolladas en las iteraciones anteriores. 		
Artefactos modificados o creados	<ul style="list-style-type: none"> • Diagrama de clases de la API inicial. • Clases Java de la API inicial y de la aplicación. • El presente documento. 		
Calendarización detallada	Análisis + Pruebas	29/03/10	<ul style="list-style-type: none"> • Planificación de la iteración y gestión de riesgos. • Análisis de los casos de uso y escenarios. • Planificación de las pruebas. • Inicio de la documentación.
	Diseño + Pruebas	29/03/10	<ul style="list-style-type: none"> • Diseño de opciones arquitectónicas. • Diseño de pruebas.
	Codificación + Pruebas	30/03/10	<ul style="list-style-type: none"> • Codificación e integración de la solución. • Ejecución y evaluación de pruebas. • Evaluación del prototipo. • Preparación de la entrega ejecutable.
		31/03/10	
Despliegue	31/03/10	<ul style="list-style-type: none"> • Muestra, prueba y evaluación conjunta con el cliente. • Concreción de los cambios necesarios. • Finalización de toda la documentación. 	

Planificación de la iteración



Nombre de tarea	Duración	Comienzo	Fin	dom 26 mar	0	0	mar 30 mar	0	0
+ Estudio de oportunidad (inception)	7 días	jue 07/01/10	vie 15/01/10						
+ Elaboración	16 días	lun 18/01/10	lun 08/02/10						
- Construcción	41 días	mar 09/02/10	mar 06/04/10						
+ Iteración C1: La base de datos	4 días	mar 09/02/10	vie 12/02/10						
+ Iteración C2: API para la base de datos	5 días	lun 15/02/10	vie 19/02/10						
+ Iteración C3: Accesos	5 días	lun 22/02/10	vie 26/02/10						
+ Iteración C4: Empresas y personal	5 días	lun 01/03/10	vie 05/03/10						
+ Iteración C5: Tarjetas, tipos de acceso y cargos	3 días	lun 08/03/10	mié 10/03/10						
+ Iteración C6: Zonas, llaves y llaveros	5 días	jue 11/03/10	mié 17/03/10						
+ Iteración C7: Préstamos	2 días	jue 18/03/10	vie 19/03/10						
+ Iteración C8: Cuentas	2 días	lun 22/03/10	mar 23/03/10						
+ Iteración C9: Partes de seguridad	3 días	mié 24/03/10	vie 26/03/10						
- Iteración C10: Notas	3 días	lun 29/03/10	mié 31/03/10						
Análisis	0,5 días	lun 29/03/10	lun 29/03/10						
Diseño	0,5 días	lun 29/03/10	lun 29/03/10						
Implementación	1,5 días	mar 30/03/10	mié 31/03/10						
Pruebas	3 días	lun 29/03/10	mié 31/03/10						
Despliegue	0,5 días	mié 31/03/10	mié 31/03/10						
+ Iteración C11: Base de datos	2 días	jue 01/04/10	vie 02/04/10						
+ Iteración C12: Diseño, personalización y configuración	2 días	lun 05/04/10	mar 06/04/10						
Capacidad operacional	0 días	mar 06/04/10	mar 06/04/10						
+ Transición	8 días	mié 07/04/10	vie 16/04/10						

Recursos	Temporales		3 días
	Humanos		1 persona
	Software	Sistema operativo	Microsoft Windows 7
		Ofimática	Microsoft Word 2010 Adobe Acrobat Professional 8
		Diseño	Visual Paradigm 4.2
		Codificación	NetBeans 6.8
Hardware		1 PC	

Casos de uso y escenarios implementados	126 - Redactar nota
	127 - Visualiza notas recibidas
	128 - Visualizar nota recibida
	129 - Borrar nota recibida
	130 - Visualizar notas enviadas
	131 - Visualizar nota enviada
	132 - Eliminar nota enviada

Planificación de la iteración

3.2. Análisis de los casos de uso y escenarios

La actividad de análisis nos lleva a transformar las descripciones de trazo grueso en otras de trazo fino, especificando así todo el proceso que deberá desarrollarse en la ejecución de cada caso de uso. En las siguientes páginas plasmamos todas y cada una de las descripciones que posteriormente diseñaremos.



Caso de uso 126: Redactar nota	
Descripción	Ahora, en la iteración destinada a las notas, vamos proporcionar un servicio sencillo de mensajería. En primer lugar describimos cómo un trabajador podría enviar un mensaje a otro.
Actores principal	Administrador, Autorizador y Vigilante.
Pre-condiciones	Un trabajador quiere enviar un mensaje a otro utilizando el servicio de mensajería de accessManager. Para ello loguea en el sistema y se sitúa en la tab de notas.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario indica al sistema que desea redactar una nota. 2. El sistema le proporciona un formulario con el que redactarla e indicar a qué destinatarios irá dirigida. 3. El usuario la escribe e indica sus receptores. 4. El sistema recoge toda la información y pide al usuario que apruebe el envío. 5. El usuario da su visto bueno y el sistema de encarga de enviar la nota. (*)
	<p>Alternativos (*)</p> <p>El usuario escribe una nota con alguno de sus datos con un formato erróneo:</p> <ul style="list-style-type: none"> • El sistema indica el error ocurrido y no envía ninguna nota, pasándose al punto 2 con el formulario intacto. <p>El usuario ve que se ha equivocado en alguno de los campos, o por las razones que sea decide abortar el envío de la nota contestando negativamente ante el aviso de envío:</p> <ul style="list-style-type: none"> • El sistema cierra el diálogo de aviso y no ejecuta ninguna acción más.
Post-condiciones	El usuario ha enviado la nota a los destinatarios que deseaba.

Caso de uso 126: Redactar nota



Caso de uso 127: Visualizar notas recibidas	
Descripción	Una vez que tenemos la posibilidad de enviar notas, pasamos al administración de éstas, tanto las de carácter entrante (recibidas) como las de carácter saliente (enviadas). En cuanto a las primeras cabe decir que no podemos proporcionar simultáneamente una visión de todas ellas con todos los datos que las componen, ya que la cantidad de notas que un usuario puede disponer es ilimitada. Nos encontramos ante un caso muy similar al de los partes de seguridad. Por tanto, atenderemos al visionado de todas las notas recibidas y el contenido de cada una de ellas en éste y el siguiente caso de uso respectivamente.
Actores principal	Administrador, Autorizador y Vigilante.
Pre-condiciones	El usuario accede a la pestaña de notas con la intención de comprobar sus notas entrantes.
Flujos	Básico 1. El usuario accede a la sección de notas entrantes. 2. El sistema se encarga de obtener todas las notas entrantes de éste y mostrárselas.
	Alternativos (*) Ninguno.
Post-condiciones	El usuario conoce qué notas componen su “bandeja de entrada”.

Caso de uso 127: Visualizar notas recibidas

Caso de uso 128: Visualizar nota recibida	
Descripción	Tras disponer de la lista de notas que tenemos entrantes, nos queda proporcionar la posibilidad de leerlas.
Actores principal	Administrador, Autorizador y Vigilante.
Pre-condiciones	El usuario se encuentra visionando las notas que posee entrantes y desea realizar la lectura de alguna de ellas.
Flujos	Básico 1. El usuario selecciona la nota que desea leer. 2. El sistema acude a la base de datos para recuperar los datos de ésta y mostrárselos al usuario a través de un formulario.
	Alternativos (*) Ninguno.
Post-condiciones	El usuario visualiza por pantalla toda la composición de la nota.

Caso de uso 128: Visualizar nota recibida



Caso de uso 129: Borrar nota recibida					
Descripción	Las notas recibidas no tienen por qué ser almacenadas de manera indefinida; el usuario cuando no necesite más de una de ellas debe poder eliminarla.				
Actores principal	Administrador, Autorizador y Vigilante.				
Pre-condiciones	El usuario se encuentra logueado en el sistema, más concretamente visualizando sus notas entrantes en la pestaña 'Notas', y desea dejar de conservar algunas de ellas.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> <ol style="list-style-type: none"> 1. El usuario selecciona las notas que desea eliminar de su "bandeja de entrada", trasladando al sistema su intención de no seguir conservándolas. 2. El sistema le envía un mensaje indicándole que va a proceder con el borrado de éstas, pero que previamente necesita su autorización. 3. El usuario la da y así el sistema puede finalizar el caso de uso con el objetivo cumplido. (*) </td> </tr> <tr> <td>Alternativos (*)</td> <td> <p>El usuario decide no eliminar las notas respondiendo 'No':</p> <ul style="list-style-type: none"> • El sistema no elimina ningún registro de la base de datos y da por concluido el caso de uso. </td> </tr> </table>	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona las notas que desea eliminar de su "bandeja de entrada", trasladando al sistema su intención de no seguir conservándolas. 2. El sistema le envía un mensaje indicándole que va a proceder con el borrado de éstas, pero que previamente necesita su autorización. 3. El usuario la da y así el sistema puede finalizar el caso de uso con el objetivo cumplido. (*) 	Alternativos (*)	<p>El usuario decide no eliminar las notas respondiendo 'No':</p> <ul style="list-style-type: none"> • El sistema no elimina ningún registro de la base de datos y da por concluido el caso de uso.
	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona las notas que desea eliminar de su "bandeja de entrada", trasladando al sistema su intención de no seguir conservándolas. 2. El sistema le envía un mensaje indicándole que va a proceder con el borrado de éstas, pero que previamente necesita su autorización. 3. El usuario la da y así el sistema puede finalizar el caso de uso con el objetivo cumplido. (*) 			
Alternativos (*)	<p>El usuario decide no eliminar las notas respondiendo 'No':</p> <ul style="list-style-type: none"> • El sistema no elimina ningún registro de la base de datos y da por concluido el caso de uso. 				
Post-condiciones	Las notas borradas dejan de constar en la lista de notas entrantes del usuario.				

Caso de uso 129: Borrar nota recibida

Caso de uso 130: Visualizar notas enviadas					
Descripción	Todo lo que proponemos para la recepción de notas también lo haremos para el envío, además con una funcionalidad nueva añadida que comentaremos en el siguiente caso de uso. En éste tratamos el visionado de todas las notas que hemos enviado y seguimos conservando.				
Actores principal	Administrador, Autorizador y Vigilante.				
Pre-condiciones	El usuario despliega accessManager y se dirige a la pestaña 'Notas' con la idea de observar las que ha enviado y sigue almacenando.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> <ol style="list-style-type: none"> 1. El usuario accede a la sección de notas enviadas. 2. El sistema recupera de la base de datos la lista de notas de las que dispone el usuario como enviadas y las muestra por pantalla. </td> </tr> <tr> <td>Alternativos (*)</td> <td>Ninguno.</td> </tr> </table>	Básico	<ol style="list-style-type: none"> 1. El usuario accede a la sección de notas enviadas. 2. El sistema recupera de la base de datos la lista de notas de las que dispone el usuario como enviadas y las muestra por pantalla. 	Alternativos (*)	Ninguno.
	Básico	<ol style="list-style-type: none"> 1. El usuario accede a la sección de notas enviadas. 2. El sistema recupera de la base de datos la lista de notas de las que dispone el usuario como enviadas y las muestra por pantalla. 			
Alternativos (*)	Ninguno.				
Post-condiciones	El usuario consigue lo que se proponía: visualizar qué notas dispone en el "buzón de salida".				

Caso de uso 130: Visualizar notas enviadas



Caso de uso 131: Visualizar nota enviada		
Descripción	El usuario, así como puede visualizar qué notas ha recibido, también podrá visualizar las que ha enviado. El visionado de la información con la que fue redactada quizá no tenga tanto sentido como pueda ser el de las recibidas, pero el usuario nos pide que pueda visualizar qué personas la han leído, y esto sí que puede ser de gran utilidad.	
Actores principal	Administrador, Autorizador y Vigilante.	
Pre-condiciones	El usuario se encuentra visualizando las notas que componen su "bandeja de salida" y desea consultar una nota de las que la componen, bien sea por recordar qué escribió en ella o por comprobar que destinatarios la han leído.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario clics sobre la nota que desea visualizar. 2. El sistema se encarga de desplegar en pantalla todos los datos que posee de ésta.
	Alternativos (*)	Ninguno.
Post-condiciones	El usuario visualiza en pantalla todos los datos que buscaba.	

Caso de uso 131: Visualizar nota enviada

Caso de uso 132: Borrar nota enviada		
Descripción	A medida que transcurre el tiempo el usuario reusará seguir conservando notas que redactó, ya sea porque todos sus destinatarios la hayan leído, porque no le interese estar informado sobre su situación,... Entonces, proporcionaremos una forma de poder borrar notas enviadas al igual que la proporcionamos con las recibidas.	
Actores principal	Administrador, Autorizador y Vigilante.	
Pre-condiciones	El usuario loguea en el sistema, clics la pestaña de notas y se dirige a la sección de notas enviadas.	
Flujos	Básico	<ol style="list-style-type: none"> 1. El usuario selecciona el conjunto de notas que no desea conservar y clics sobre el botón de borrado. 2. El sistema pregunta al usuario si está seguro de querer continuar con este borrado. 3. El usuario contesta afirmativamente. (*) 4. El sistema actualiza la base de datos de modo que el usuario no visualice más las notas seleccionadas pero sus receptores puedan seguir conservándolas.
	Alternativos (*)	<p>El usuario contesta negativamente:</p> <ul style="list-style-type: none"> • El sistema no modifica ningún dato y el caso de uso finaliza.
Post-condiciones	El usuario descarta las notas que había seleccionado, no volviendo a disponer de ellas.	

Caso de uso 132: Borrar nota enviada



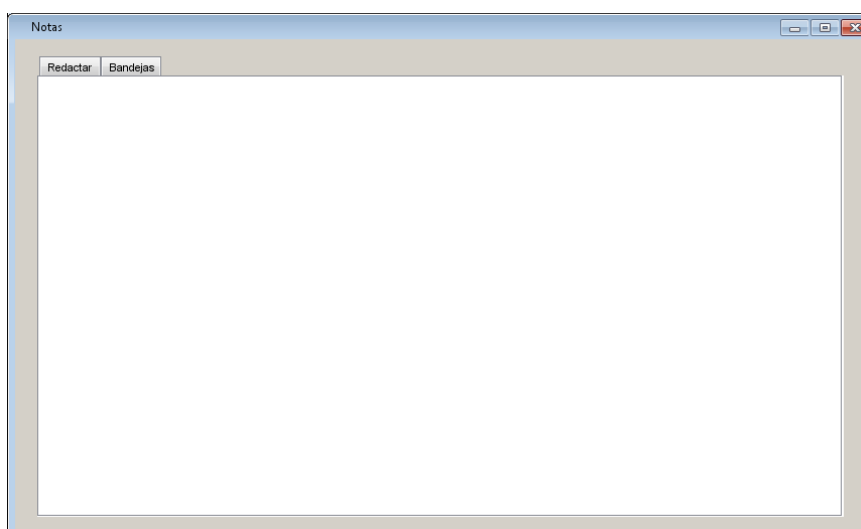
4. Diseño

4.1. Caso de uso 126: Redactar nota

Comenzamos con toda la parte de diseño, analizando cómo distribuiremos los casos de uso en la pestaña, qué componentes nos permitirán desarrollarlos, llevarlos a cabo. Tenemos dos actividades muy diferenciadas:

- Redactar notas
- Visionar notas
 - Entrantes
 - Salientes

Como no podemos hacer ambas cosas a la vez, resulta muy interesante crear un JTabbedPane con el que podamos aprovechar al máximo las proporciones (espacio) de la pestaña:

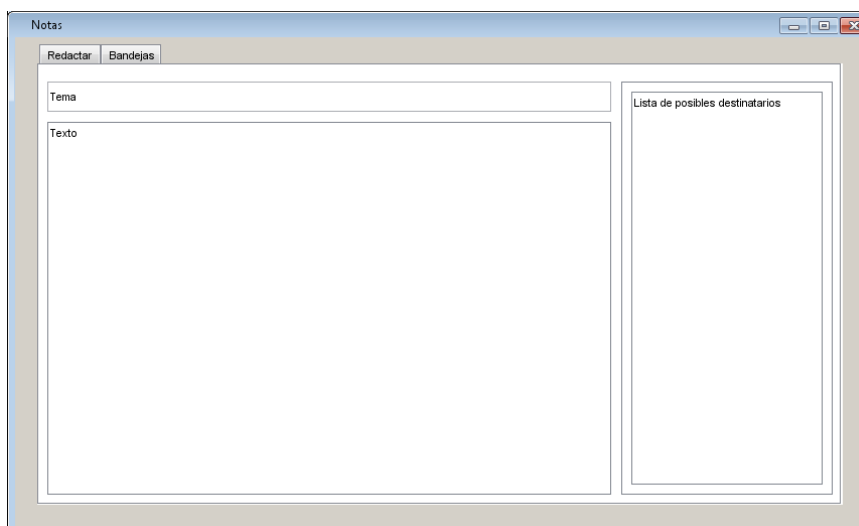


Interfaz gráfica propuesta

A 'Redactar' solo vamos a asignar la actividad de redactar notas. Para ello necesitaremos crear campos para la estructura que debe tener cada nota, que son tema, texto y destinatarios. Los dos primeros son campos de texto fácilmente representables con un JTextField y un JTextArea, pero el tercero puede contener de uno a varios elementos (receptores). La mejor forma de solucionar éste inconveniente es crear una tabla con todos los usuarios que puedan recibir notas y que sea el redactor quien seleccione dentro de ésta.



Los tres componentes gráficos podrían quedar distribuidos de la siguiente manera:



Interfaz gráfica propuesta

La lista de destinatarios, al igual que otros componentes gráficos con contenido variable, será rellenada al iniciarse la aplicación mediante el método *obtenerDestinatarios*, que debemos crear y añadir a la API:

MÉTODO	FUNCIONALIDAD
<i>obtenerDestinatarios</i>	Método que devuelve primer apellido, segundo apellido y nombre de todos aquellos usuarios que tienen acceso al servicio de mensajería (administradores, autorizadores y vigilantes).

Método a implementar y a añadir a la API

Con todo en disposición, el usuario puede redactar y enviar notas. El sistema proporcionará esta funcionalidad de la siguiente forma:

1. Crearemos un objeto de tipo Nota con los datos obtenidos del formulario (dos campos de texto y un HashMap).
2. Realizaremos el envío de ésta mediante el método *insertNotaBD*.
3. Borraremos el formulario de la nota redactada.
4. Refrescaremos ambas "bandejas" para que la nota recientemente enviada conste en las que sea oportuno.

La primera y segunda acción podemos realizarla sin problemas; la tercera será activada tras el caso de uso 130.



Estos diagramas pueden ayudarnos en la fase de codificación:

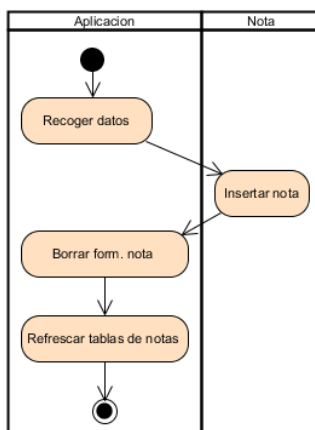


Diagrama de actividades

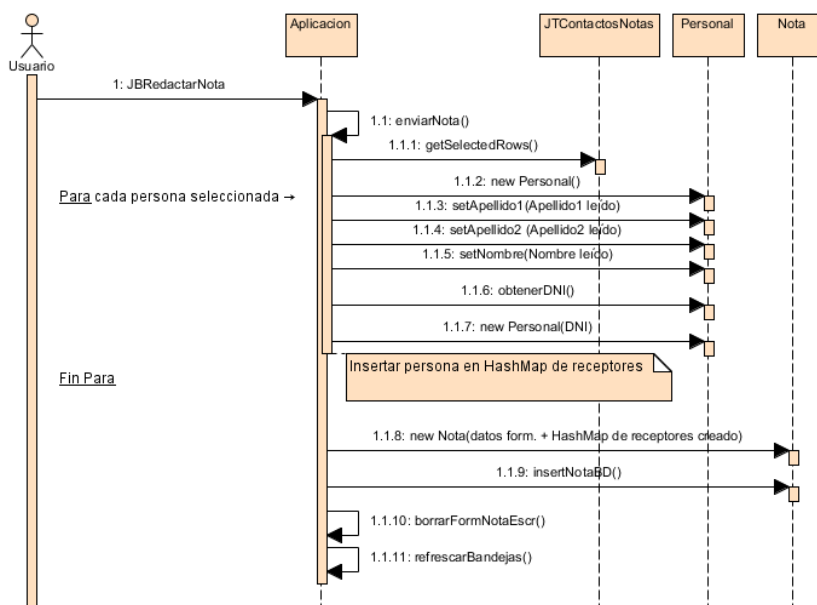


Diagrama de secuencia

El haber optado por la muestra de los contactos a través de sus nombres y apellidos hace que debamos implementar y añadir a la API este método para poder obtener los DNIs de cada destinatario:

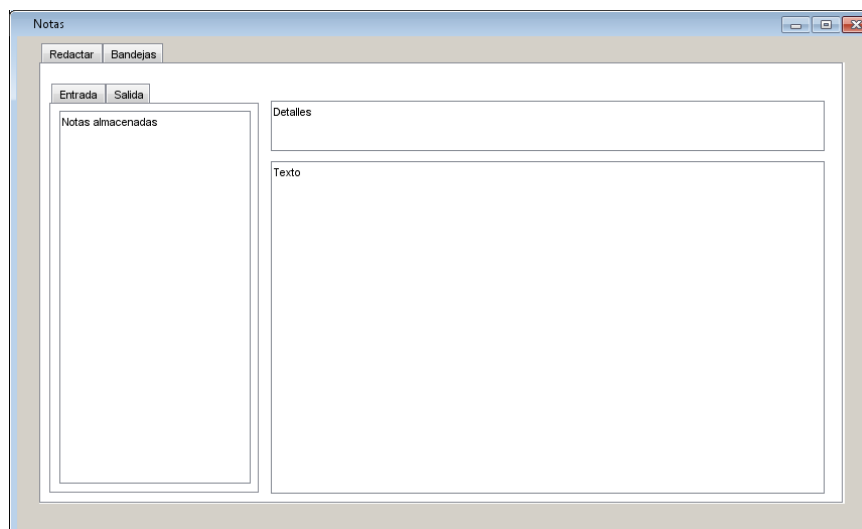
MÉTODO	FUNCIONALIDAD
<i>obtenerDNI</i>	Método que obtiene el DNI de una persona a partir de sus atributos de apellidos y Nombre.

Método a implementar y a añadir a la API



4.2. Casos de uso 127 y 130: Visualizar notas recibidas y enviadas

El segundo grupo de actividades (visualización de notas) nos lleva a la segunda pestaña del JTabbedPane creado en la pestaña 'Notas'. Dentro de la visualización de notas tenemos las recibidas y las enviadas, compartiendo ambos grupos las mismas acciones (visualizar todas las notas y los datos de cada una). Ante esta coincidencia de funcionalidades podemos crear otro JTabbedPane con dos pestañas ('Entrada' y 'Salida') destinado única y exclusivamente a mostrar las notas y un formulario común a ambos casos de uso que permita observar los datos de los que están compuestas:



Interfaz gráfica propuesta

La muestra de notas tanto recibidas como enviadas siguen el mismo proceso, por eso las estudiamos y diseñamos conjuntamente...

1. Crear un objeto de tipo Nota.
2. Obtenemos el conjunto de notas que ha recibido y enviado el usuario y que sigue conservando (Descartada = *false*).
3. Rellenamos la tabla de notas de entrada/salida con los resultados obtenidos.

... aunque en nuestra aplicación serán llamadas a la vez (*refrescarBandejas*) o por separado (*refrescarbandejaEntrada* y *refrescarBandejaSalida*).



Esta secuencia de operaciones carece de complejidad alguna, tal y como muestran sus diagramas de actividades y de secuencia:

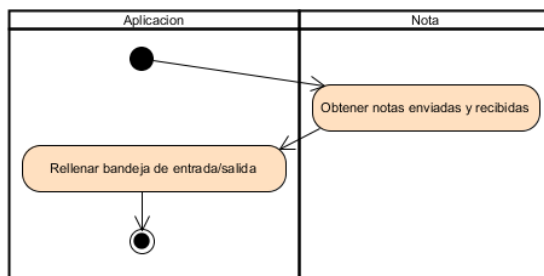


Diagrama de actividades

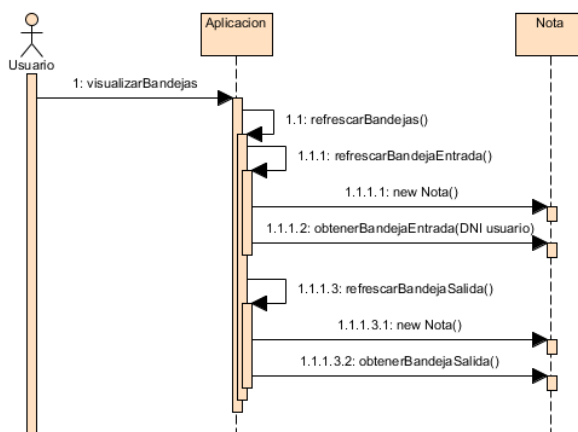


Diagrama de secuencia



4.3. Casos de uso 128 y 131: Visualizar nota recibida/enviada

En esta ocasión no se puede visualizar una nota recibida y otra enviada a la vez, pero su estudio es simultáneo ya que solo varía la tabla de la que recoger la fecha y redactor (clave primaria para obtener la nota) y la forma de mostrar los destinatarios de la misma (sin control de lectura en el primer caso y con control en el segundo).

El proceso sería:

1. Recoger datos de la nota:
 - Fecha
 - DNI
 - a. Si se trata de una nota recibida: Crear un objeto de tipo Personal con el que obtendremos el DNI del redactor.
 - b. Si se trata de una nota enviada: Obtener el DNI a través del atributo global *DNI_Usu*.
2. Obtener el objeto Nota a partir de los datos recogidos.
3. Mostrar la nota:
 - Crear una String con los receptores en función del tipo de nota que ha si clicada para visualizarla (recibida o enviada) y rellenar el campo asignado a los destinatarios.
 - Rellenar el resto de campos.
4. Si se trata de una nota recibida, marcarla como leída si es que no lo estaba.

Necesitaremos añadir a la API estos métodos para llevar a buen fin el anterior flujo de actividades:

MÉTODO	FUNCIONALIDAD
<i>obtenerDNI</i>	Método que obtiene el DNI de una persona a partir de la String "Apellido1 Apellido2, Nombre".
<i>Nota</i>	Método que crea un objeto de tipo Nota obteniendo los valores de todos sus atributos de la base de datos a partir de la fecha y DNI pasados como parámetros de entrada.

Métodos a implementar y a añadir a la API

El primero de los métodos es una variante del que ya teníamos desarrollado, solo que se encarga de extraer los valores de la String y después llamar al método original.



Resulta difícil perderse en esta explicación si además la dotamos de estos aclaratorios diagramas:

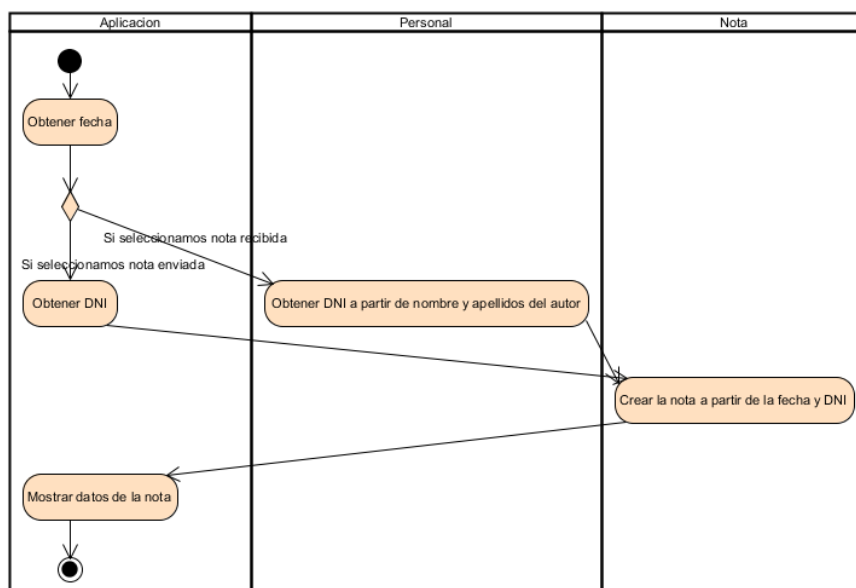


Diagrama de actividades

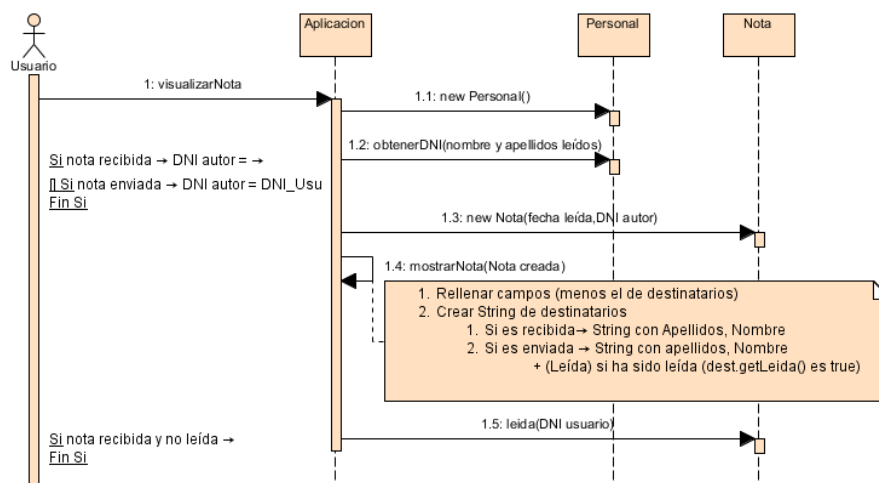


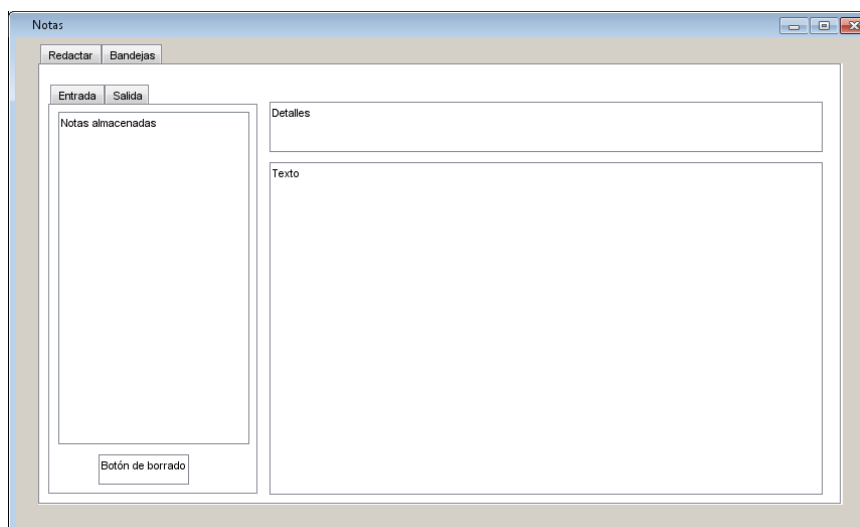
Diagrama de secuencia

4.4. Casos de uso 129 y 132: Borrar nota recibida/enviada

Tal y como fue ideado el esquema relacional que da forma a la base de datos, no podemos eliminar ningún registro perteneciente a ninguna nota, ni por parte del redactor ni por parte del destinatario ya que perderíamos bien la nota en su totalidad o bien las referencias de a qué personas ha sido enviada. Por lo tanto, lo único que nos quedó en su momento fue crear tres atributos ('Descartada', 'Leída' y 'Descartada') que nos permitieran realizar toda la gestión y dejar en manos de los disparadores el borrado de las tuplas.



Por todo lo comentado, borrar una nota se traduce a marcarla como descartada. Esta acción es llevada a cabo por ambos tipos de notas (recibidas y enviadas), con que decidimos crear un botón de borrado común bajo el panel pestañado...



Interfaz gráfica propuesta

... que desencadene esta secuencia de acciones:

1. Obtener las notas seleccionadas para el borrado.
2. Recorrer cada nota seleccionada obteniéndola como se explicó en el caso de uso anterior y marcar como descartado el Destinatario o la Nota en función de si realizamos la acción de borrado sobre notas recibidas o enviadas.
3. Refrescar la bandeja de entrada o la de salida dependiendo sobre cuál hayamos descartado notas.

Este caso de uso hace que el atributo 'Descartada' de las tablas Nota o Destinatario deban ser actualizadas de *false* a *true*. En la implementación de la API inicial ya desarrollamos métodos de actualización de notas y destinatarios pero en esta ocasión, con la intención de simplificar el código, se suman estos métodos a los ya existentes en la API:

MÉTODO	FUNCIONALIDAD
<i>descartada</i>	Método que marca una Nota como descartada.
<i>descartada</i>	Método que marca como descartado un Destinatario de la Nota sobre la que se realiza la llama al método.

Métodos a implementar y a añadir a la API



La estrategia diseñada se concreta en:

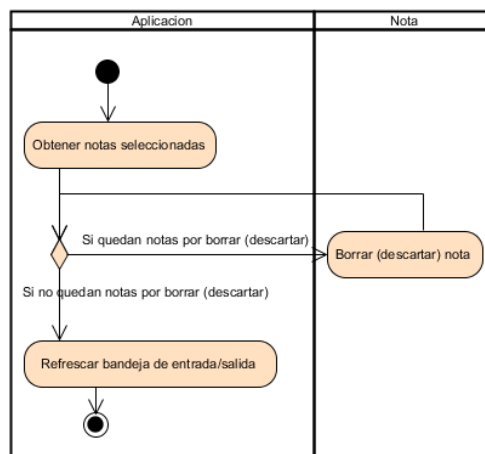


Diagrama de actividades

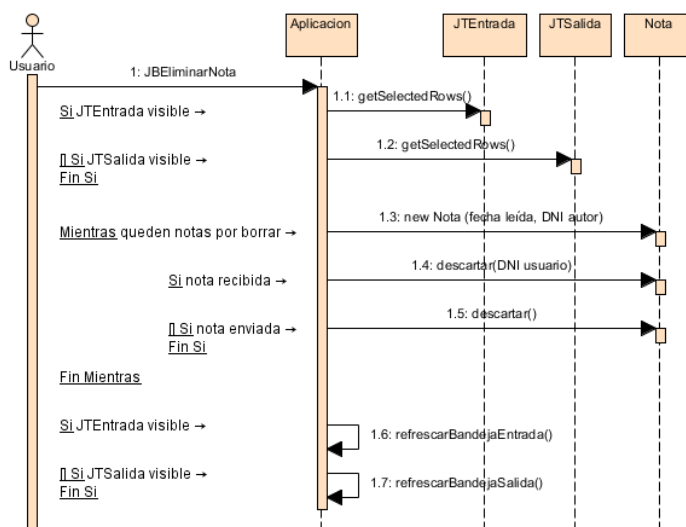


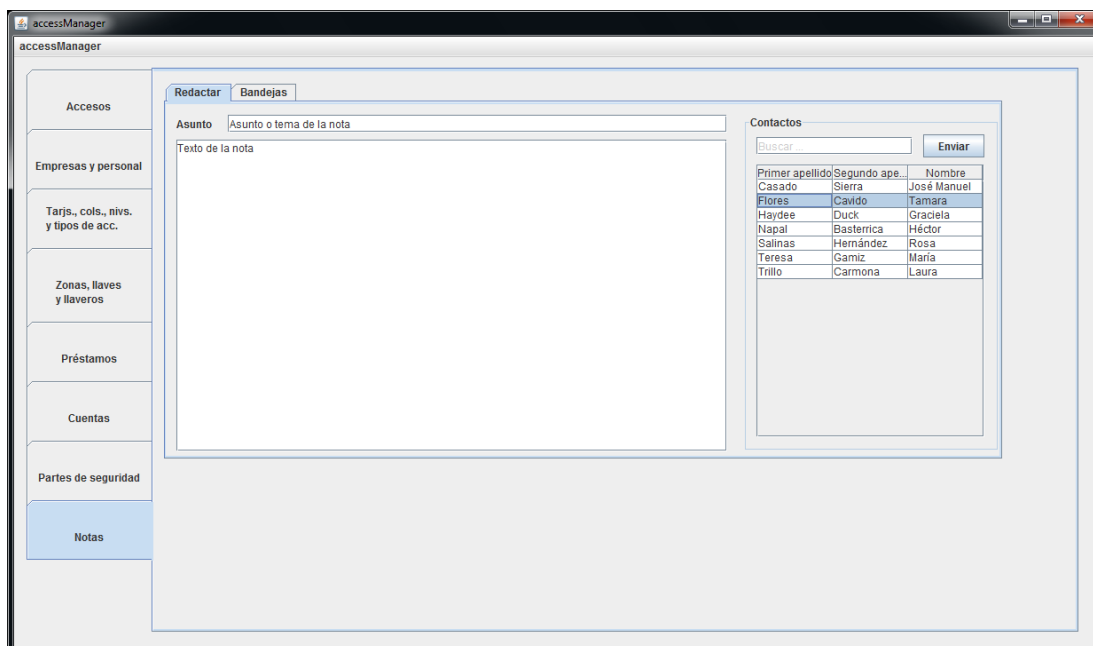
Diagrama de secuencia

5. Codificación e integración

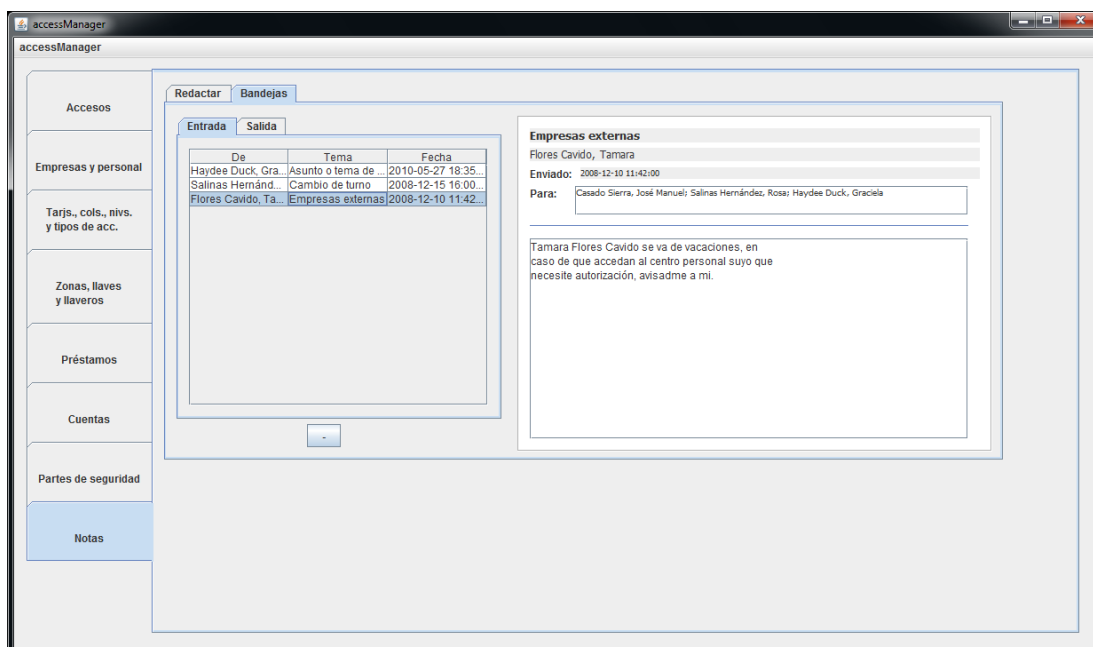
La codificación de 'Notas' tiene lugar una vez ha concluido la etapa de diseño y no aparenta conllevar un grado de complejidad a tener en cuenta si se siguen con cuidado las estrategias y diagramas que hemos expuesto en la sección anterior.



Esta es la apariencia que tendrá la pestaña en un principio:



Interfaz de la pestaña 'Notas': Redactar



Interfaz de la pestaña 'Notas': Bandejas

La integración del sistema sigue siendo la misma; no existen elementos nuevos si no que más desarrollados y refinados.

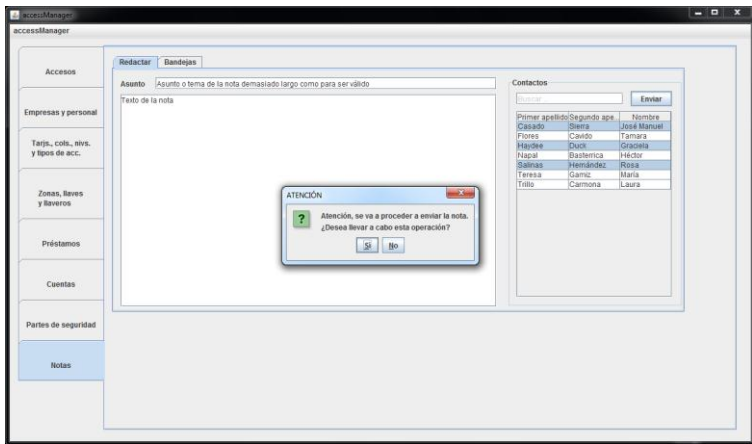


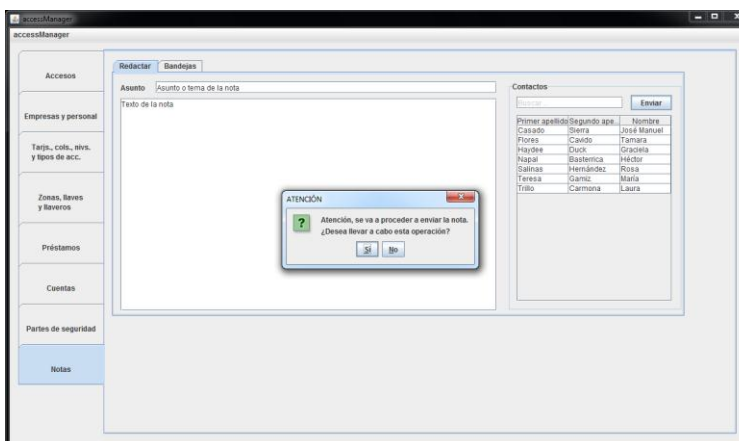
6. Pruebas

6.1. De unidad

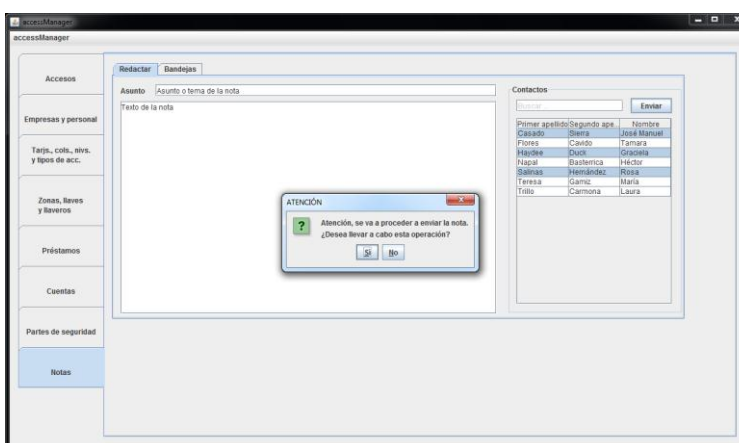
Llega la hora de probar la eficacia funcional de cada caso de uso desarrollado. Todos están muy relacionados, y podría idearse un caso de prueba que los probase todos en cadena, pero preferimos ir uno a uno para facilitar el trabajo.

Nota: Los usuarios que tomamos como ejemplos en los casos de prueba de notas recibidas es el de DNI 00000002A y el de notas enviadas el de DNI 00000043A.

Caso de prueba	10.01 - Casos de uso 126: Redactar nota
Objetivo	Comprobar que el usuario puede redactar notas y enviarlas al resto de usuarios que disponen de mensajería siempre y cuando las notas sean redactadas correctamente (formato de datos válido).
Actividades	Loguear en el sistema, situarnos en la pestaña de notas, redactar una nota correcta e incorrectamente y verificar que todo sucede de acuerdo a como se tenía previsto.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos con población de prueba.• Utilidad phpMyAdmin.
Datos de entrada	<p>En un principio probaremos a redactar una nota con algún dato con formato erróneo y otra sin destinatarios...</p>  <p>The screenshot shows the 'accessManager' application interface. On the left is a navigation menu with categories like 'Accesos', 'Empresas y personal', 'Tarjetas, colas, vivs. y tipos de acc.', 'Zonas, Reves y Reveros', 'Préstamos', 'Cuentas', 'Partes de seguridad', and 'Notas'. The main area is titled 'Redactar' and contains a form with an 'Asunto' field (with a warning message: 'Asunto o tema de la nota demasiado largo como para ser válido') and a 'Texto de la nota' field. On the right, there is a 'Contactos' table with columns for 'Primer apellido', 'Segundo ape.', and 'Nombre'. A confirmation dialog box is open in the center, asking 'Atención, se va a proceder a enviar la nota. ¿Desea llevar a cabo esta operación?' with 'Sí' and 'No' buttons.</p>



... y a continuación mandaremos una válida a tres personas:



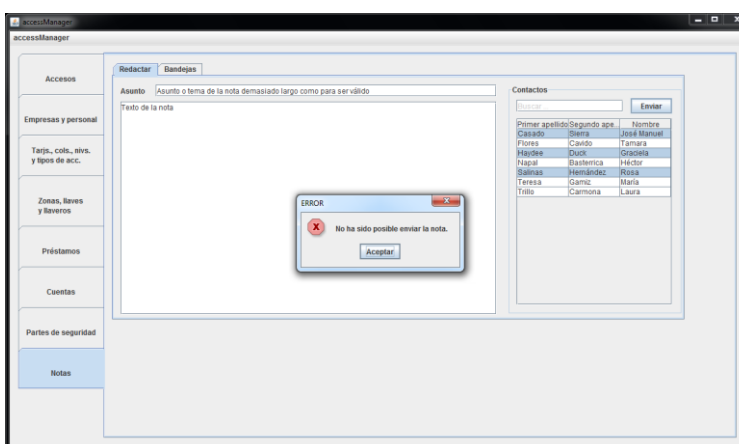
Salida esperada

Las dos primeras pruebas darán como únicos resultados mensajes de error en pantalla, pero la última realizará el envío de la nota a todos sus destinatarios.

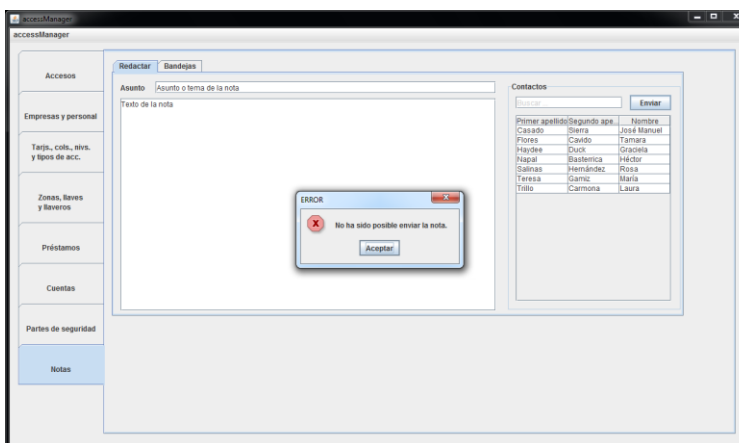
Salida observada

La esperada:

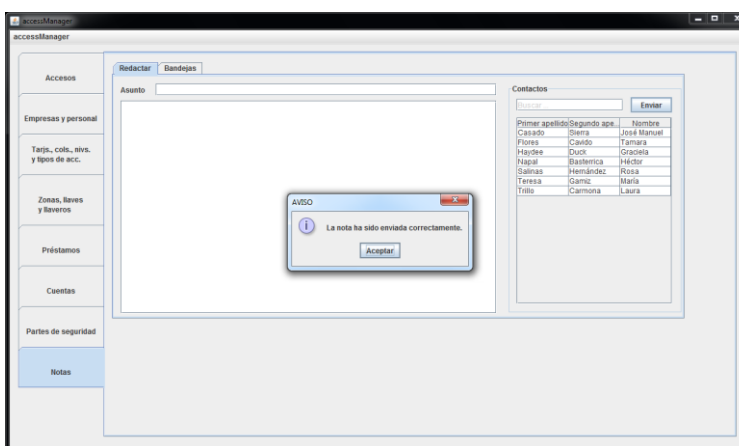
- En el caso de crear notas no válidas (datos con formato no admitido):



- Si no se seleccionan destinatarios:



- El envío de notas no da problemas si se redactan bien:



Fecha	DNI_A	Tema	Texto	Descartada
2010-05-27 18:35:02	00000002A	Asunto o tema de la nota	Texto de la nota	0

Fecha	DNI_A	DNI_D	Leída	Descartada
2010-05-27 18:35:02	00000002A	00000003A	0	0
2010-05-27 18:35:02	00000002A	00000002A	0	0
2010-05-27 18:35:02	00000002A	00000001A	0	0

**Evaluación/
conclusión**

El caso de uso '126 - Redactar nota' ha sido realizado satisfactoriamente.

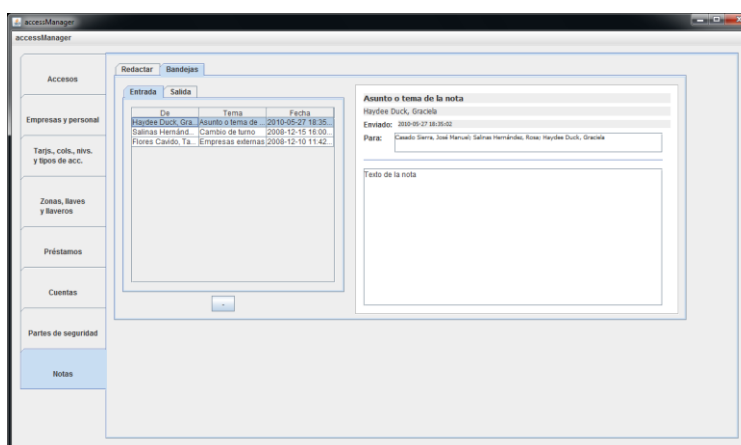
Caso de prueba para el caso de uso '126 - Redactar nota'

Caso de prueba	10.02 - Casos de uso 127: Visualizar notas recibidas																																																																																																																								
Objetivo	Verificar que el usuario puede visualizar todas las notas que le hayan sido enviadas y no eliminadas.																																																																																																																								
Actividades	Acceder a la aplicación, clicar la tab 'Notas' e ir a la "bandeja de entrada", comprobando que contenga las notas que debe tener.																																																																																																																								
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior. • Utilidad phpMyAdmin. 																																																																																																																								
Datos de entrada	<p>La población de la tabla Destinatarios, en la que modificamos las notas recibidas por el usuario de prueba para que exista una descartada y las demás no:</p> <table border="1"> <thead> <tr> <th>Fecha</th> <th>Fecha y hora de redacción</th> <th>DNI_A</th> <th>DNI del autor</th> <th>DNI_D</th> <th>DNI del destinatario</th> <th>Leida</th> <th>Denota si ha sido marcada como leída</th> <th>Descartada</th> <th>Denota si ha sido marcada como descartada</th> </tr> </thead> <tbody> <tr> <td></td> <td>2008-12-04 11:30:00</td> <td>00000018A</td> <td></td> <td>00000002A</td> <td></td> <td>0</td> <td></td> <td>0</td> <td></td> </tr> <tr> <td></td> <td>2008-12-10 11:42:00</td> <td>00000043A</td> <td></td> <td>00000002A</td> <td></td> <td>0</td> <td></td> <td>0</td> <td></td> </tr> <tr> <td></td> <td>2008-12-15 16:00:00</td> <td>00000001A</td> <td></td> <td>00000002A</td> <td></td> <td>0</td> <td></td> <td>0</td> <td></td> </tr> <tr> <td></td> <td>2010-05-27 18:35:02</td> <td>00000002A</td> <td></td> <td>00000002A</td> <td></td> <td>0</td> <td></td> <td>0</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Fecha</th> <th>Fecha y hora de redacción</th> <th>DNI_A</th> <th>DNI del autor</th> <th>DNI_D</th> <th>DNI del destinatario</th> <th>Leida</th> <th>Denota si ha sido marcada como leída</th> <th>Descartada</th> <th>Denota si ha sido marcada como descartada</th> </tr> </thead> <tbody> <tr> <td></td> <td>2008-12-04 11:30:00</td> <td>00000018A</td> <td></td> <td>00000002A</td> <td></td> <td>0</td> <td></td> <td>0</td> <td>1</td> </tr> <tr> <td></td> <td>2008-12-10 11:42:00</td> <td>00000043A</td> <td></td> <td>00000002A</td> <td></td> <td>0</td> <td></td> <td>0</td> <td>0</td> </tr> <tr> <td></td> <td>2008-12-15 16:00:00</td> <td>00000001A</td> <td></td> <td>00000002A</td> <td></td> <td>0</td> <td></td> <td>0</td> <td>0</td> </tr> <tr> <td></td> <td>2010-05-27 18:35:02</td> <td>00000002A</td> <td></td> <td>00000002A</td> <td></td> <td>0</td> <td></td> <td>0</td> <td>0</td> </tr> </tbody> </table> <p>Además, de las tres no descartadas hacemos que una haya sido descartada por el redactor:</p> <table border="1"> <thead> <tr> <th>Fecha</th> <th>Fecha y hora de redacción</th> <th>DNI_A</th> <th>DNI del autor</th> <th>Tema</th> <th>Tema/título de la nota</th> <th>Texto</th> <th>Texto/contenido de la nota</th> <th>Descartada</th> <th>Denota si ha sido marcada como descartada</th> </tr> </thead> <tbody> <tr> <td></td> <td>2008-12-15 16:00:00</td> <td>00000001A</td> <td></td> <td>Cambio de turno</td> <td></td> <td>Hola, Estaría interesado en cambiar el turno de pa...</td> <td></td> <td>1</td> <td></td> </tr> </tbody> </table>	Fecha	Fecha y hora de redacción	DNI_A	DNI del autor	DNI_D	DNI del destinatario	Leida	Denota si ha sido marcada como leída	Descartada	Denota si ha sido marcada como descartada		2008-12-04 11:30:00	00000018A		00000002A		0		0			2008-12-10 11:42:00	00000043A		00000002A		0		0			2008-12-15 16:00:00	00000001A		00000002A		0		0			2010-05-27 18:35:02	00000002A		00000002A		0		0		Fecha	Fecha y hora de redacción	DNI_A	DNI del autor	DNI_D	DNI del destinatario	Leida	Denota si ha sido marcada como leída	Descartada	Denota si ha sido marcada como descartada		2008-12-04 11:30:00	00000018A		00000002A		0		0	1		2008-12-10 11:42:00	00000043A		00000002A		0		0	0		2008-12-15 16:00:00	00000001A		00000002A		0		0	0		2010-05-27 18:35:02	00000002A		00000002A		0		0	0	Fecha	Fecha y hora de redacción	DNI_A	DNI del autor	Tema	Tema/título de la nota	Texto	Texto/contenido de la nota	Descartada	Denota si ha sido marcada como descartada		2008-12-15 16:00:00	00000001A		Cambio de turno		Hola, Estaría interesado en cambiar el turno de pa...		1	
Fecha	Fecha y hora de redacción	DNI_A	DNI del autor	DNI_D	DNI del destinatario	Leida	Denota si ha sido marcada como leída	Descartada	Denota si ha sido marcada como descartada																																																																																																																
	2008-12-04 11:30:00	00000018A		00000002A		0		0																																																																																																																	
	2008-12-10 11:42:00	00000043A		00000002A		0		0																																																																																																																	
	2008-12-15 16:00:00	00000001A		00000002A		0		0																																																																																																																	
	2010-05-27 18:35:02	00000002A		00000002A		0		0																																																																																																																	
Fecha	Fecha y hora de redacción	DNI_A	DNI del autor	DNI_D	DNI del destinatario	Leida	Denota si ha sido marcada como leída	Descartada	Denota si ha sido marcada como descartada																																																																																																																
	2008-12-04 11:30:00	00000018A		00000002A		0		0	1																																																																																																																
	2008-12-10 11:42:00	00000043A		00000002A		0		0	0																																																																																																																
	2008-12-15 16:00:00	00000001A		00000002A		0		0	0																																																																																																																
	2010-05-27 18:35:02	00000002A		00000002A		0		0	0																																																																																																																
Fecha	Fecha y hora de redacción	DNI_A	DNI del autor	Tema	Tema/título de la nota	Texto	Texto/contenido de la nota	Descartada	Denota si ha sido marcada como descartada																																																																																																																
	2008-12-15 16:00:00	00000001A		Cambio de turno		Hola, Estaría interesado en cambiar el turno de pa...		1																																																																																																																	
Salida esperada	La "bandeja de entrada" debería contener las tres notas no descartadas por el usuario.																																																																																																																								
Salida observada	<p>La esperada:</p>																																																																																																																								
Evaluación/conclusión	El caso de uso '127 - Visualizar notas recibidas' ha sido realizado satisfactoriamente.																																																																																																																								

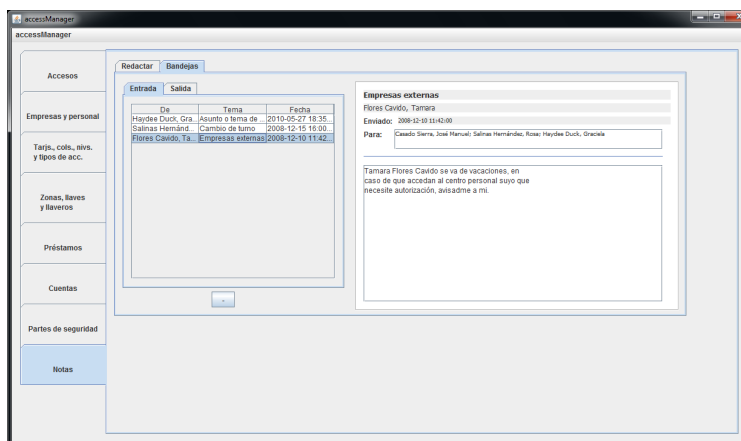
Caso de prueba para el caso de uso '127 - Visualizar notas recibidas'

Caso de prueba	10.03 - Casos de uso 128: Visualizar nota recibida
Objetivo	Probar que al clicar sobre cualquier nota recibida, ésta se despliega en el formulario mostrando al usuario todos sus datos.
Actividades	Iniciar accessManager, ir a la bandeja de entrada y seleccionar varias notas comprobando que se abren correctamente.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior. • Utilidad phpMyAdmin.
Datos de entrada	Seleccionar la primera y última nota de la “bandeja de entrada”:
Salida esperada	Las dos notas se deberán desplegar y marcarse como leídas en la base de datos.

La esperada: las notas se despliegan...



Salida observada



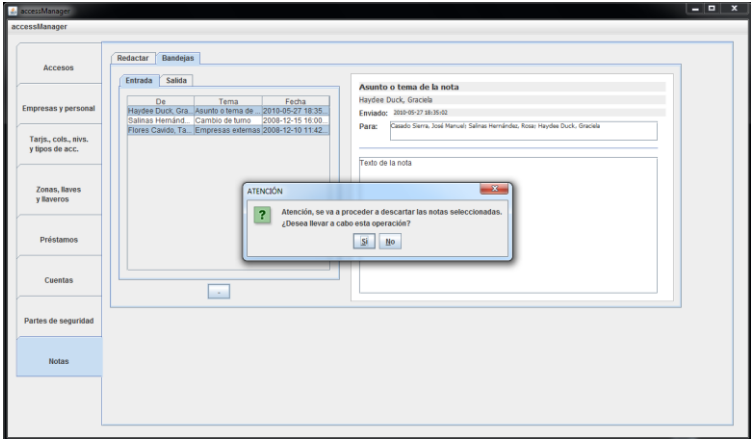
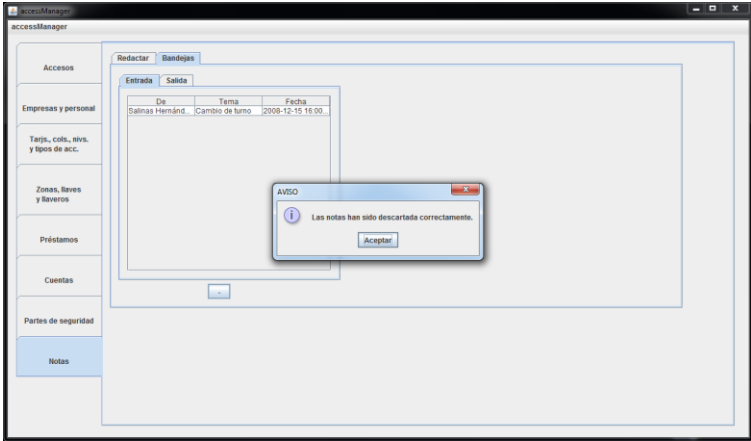
... y marcan como leídas:

Fecha	Fecha y hora de redacción	DNI_A	DNI del autor	DNI_D	DNI del destinatario	Leída	Denota si ha sido marcada como leída	Descartada	Denota si ha sido marcada como descartada
2008-12-04	11:30:00	00000018A		00000002A			0		1
2008-12-10	11:42:00	00000043A		00000002A			1		0
2008-12-15	16:00:00	00000001A		00000002A			0		0
2010-05-27	18:35:02	00000002A		00000002A			1		0

Evaluación/ conclusión	El caso de uso '128 - Visualizar nota recibida' ha sido realizado satisfactoriamente.
-------------------------------	--

Caso de prueba para el caso de uso '128 - Visualizar nota recibida'

accessManager · Iteración C10: Notas · Página 24 de 32

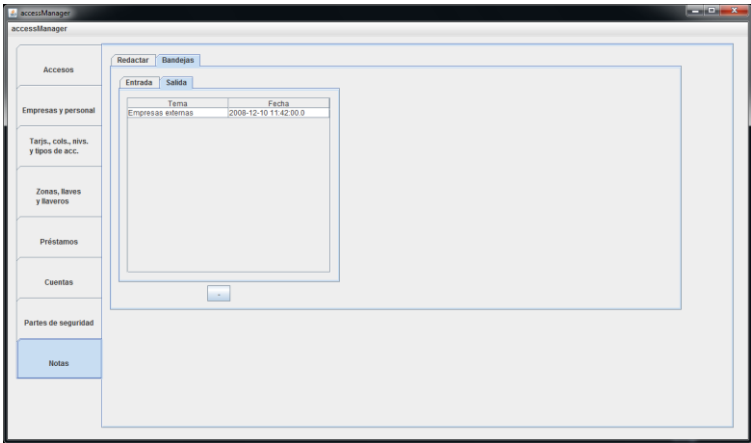
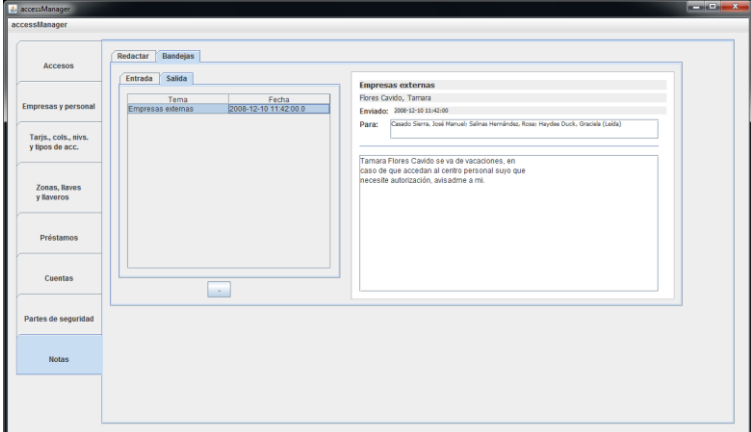
Caso de prueba	10.04 - Casos de uso 129: Borrar nota recibida																																																		
Objetivo	Chequear que el borrado de notas recibidas hace que no vuelvan a mostrarse en la “bandeja de entrada” pero sigan residiendo en el sistema marcadas como descartadas.																																																		
Actividades	Loguear en el sistema, situarnos en la pestaña de notas (más concretamente en la “bandeja de entrada”) y eliminar varias de ellas comprobando que el caso de uso se desarrolla como preveíamos.																																																		
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior. • Utilidad phpMyAdmin. 																																																		
Datos de entrada	<p>Eliminamos la primera y tercera nota recibidas:</p> 																																																		
Salida esperada	Solo deberá quedar visible la nota con fecha '2008-12-15 16:00:00' y las tuplas de la tabla Destinatarios con DNI 00000002 deberán estar marcadas con: 01, 11, 00, 11.																																																		
Salida observada	<p>La esperada:</p>  <table border="1" data-bbox="563 1803 1300 1921"> <thead> <tr> <th>Fecha</th> <th>Fecha y hora de redacción</th> <th>DNI_A</th> <th>DNI del autor</th> <th>DNI_D</th> <th>DNI del destinatario</th> <th>Leída</th> <th>Denota si ha sido marcada como leída</th> <th>Descartada</th> <th>Denota si ha sido marcada como descartada</th> </tr> </thead> <tbody> <tr> <td>2008-12-04</td> <td>11:30:00</td> <td>00000018A</td> <td>00000018A</td> <td>00000002A</td> <td>00000002A</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>2008-12-10</td> <td>11:42:00</td> <td>00000043A</td> <td>00000043A</td> <td>00000002A</td> <td>00000002A</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>2008-12-15</td> <td>16:00:00</td> <td>00000001A</td> <td>00000001A</td> <td>00000002A</td> <td>00000002A</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>2010-05-27</td> <td>18:35:02</td> <td>00000002A</td> <td>00000002A</td> <td>00000002A</td> <td>00000002A</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	Fecha	Fecha y hora de redacción	DNI_A	DNI del autor	DNI_D	DNI del destinatario	Leída	Denota si ha sido marcada como leída	Descartada	Denota si ha sido marcada como descartada	2008-12-04	11:30:00	00000018A	00000018A	00000002A	00000002A	0	0	1	1	2008-12-10	11:42:00	00000043A	00000043A	00000002A	00000002A	1	1	0	0	2008-12-15	16:00:00	00000001A	00000001A	00000002A	00000002A	0	0	1	1	2010-05-27	18:35:02	00000002A	00000002A	00000002A	00000002A	1	1	0	0
Fecha	Fecha y hora de redacción	DNI_A	DNI del autor	DNI_D	DNI del destinatario	Leída	Denota si ha sido marcada como leída	Descartada	Denota si ha sido marcada como descartada																																										
2008-12-04	11:30:00	00000018A	00000018A	00000002A	00000002A	0	0	1	1																																										
2008-12-10	11:42:00	00000043A	00000043A	00000002A	00000002A	1	1	0	0																																										
2008-12-15	16:00:00	00000001A	00000001A	00000002A	00000002A	0	0	1	1																																										
2010-05-27	18:35:02	00000002A	00000002A	00000002A	00000002A	1	1	0	0																																										

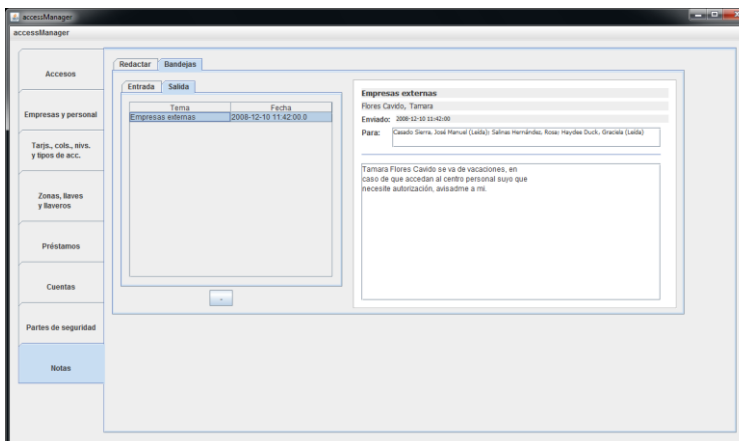


Evaluación/ conclusión	El caso de uso '129 - Borrar nota recibida' ha sido realizado satisfactoriamente. <i>Caso de prueba para el caso de uso '129 - Borrar nota recibida'</i>
-----------------------------------	--

Caso de prueba	10.05 - Casos de uso 130: Visualizar notas enviadas																														
Objetivo	Demostrar que se puede visualizar todas las notas enviadas y no descartadas.																														
Actividades	Loguear en el sistema, ir a la tab 'Bandejas' en la pestaña 'Notas' y observar que la de "bandeja de salida" recoge todas las notas que debiera.																														
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior. • Utilidad phpMyAdmin. 																														
Datos de entrada	<p>La población existente en la tabla Notas perteneciente al usuario de prueba, a la que hemos actualizado la segunda tupla el atributo Descartada a <i>true</i>:</p> <table border="1"> <thead> <tr> <th>Fecha</th> <th>Fecha y hora de redacción</th> <th>DNI_A</th> <th>DNI del autor</th> <th>Tema</th> <th>Tema/título de la nota.</th> <th>Texto</th> <th>Texto/contenido de la nota</th> <th>Descartada</th> <th>Denota si ha sido marcada como descartada</th> </tr> </thead> <tbody> <tr> <td>2008-12-10</td> <td>11:42:00</td> <td>00000043A</td> <td></td> <td>Empresas externas</td> <td></td> <td>Tamara Flores Cavido se va de vacaciones, en caso...</td> <td></td> <td></td> <td>0</td> </tr> <tr> <td>2008-12-11</td> <td>18:38:00</td> <td>00000043A</td> <td></td> <td>Alta de empresas</td> <td></td> <td>Antes de dar de alta a una empresa, consúltalo con...</td> <td></td> <td></td> <td>1</td> </tr> </tbody> </table>	Fecha	Fecha y hora de redacción	DNI_A	DNI del autor	Tema	Tema/título de la nota.	Texto	Texto/contenido de la nota	Descartada	Denota si ha sido marcada como descartada	2008-12-10	11:42:00	00000043A		Empresas externas		Tamara Flores Cavido se va de vacaciones, en caso...			0	2008-12-11	18:38:00	00000043A		Alta de empresas		Antes de dar de alta a una empresa, consúltalo con...			1
Fecha	Fecha y hora de redacción	DNI_A	DNI del autor	Tema	Tema/título de la nota.	Texto	Texto/contenido de la nota	Descartada	Denota si ha sido marcada como descartada																						
2008-12-10	11:42:00	00000043A		Empresas externas		Tamara Flores Cavido se va de vacaciones, en caso...			0																						
2008-12-11	18:38:00	00000043A		Alta de empresas		Antes de dar de alta a una empresa, consúltalo con...			1																						
Salida esperada	Visualizar como nota saliente la que tiene fecha '2008-12-10 11:42:00', y no la segunda, que se encuentra descartada por el autor.																														
Salida observada	<p>La esperada:</p>																														
Evaluación/ conclusión	El caso de uso '130 - Visualizar notas enviadas' ha sido realizado satisfactoriamente.																														

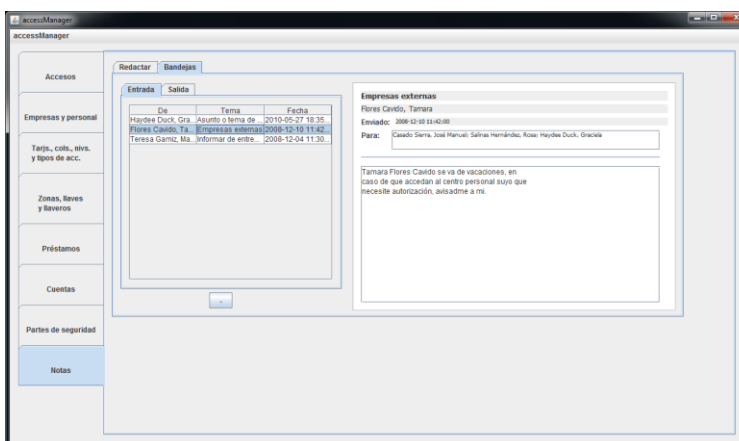
Caso de prueba para el caso de uso '130 - Visualizar notas enviadas'

Caso de prueba	10.06 - Casos de uso 131: Visualizar nota enviada																																								
Objetivo	Verificar que el usuario puede abrir cualquiera de las notas que posee en su “bandeja de salida” y observar todos los datos que la componen, así como la lectura de la misma.																																								
Actividades	Situarnos en la “bandeja de salida” del usuario de prueba y comprobar que se puede visualizar los datos de cualquier nota según lo establecido.																																								
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior. • Utilidad phpMyAdmin. 																																								
Datos de entrada	<p>Visualizaremos la única nota enviada que el usuario posee por dos ocasiones: una inicial y otra tras haber hecho que la lea uno de sus receptores:</p>  <table border="1"> <thead> <tr> <th>Fecha</th> <th>Fecha y hora de redacción</th> <th>DNI_A</th> <th>DNI del autor</th> <th>DNI_D</th> <th>DNI del destinatario</th> <th>Leída</th> <th>Denota si ha sido marcada como leída</th> <th>Descartada</th> <th>Denota si ha sido marcada como descartada</th> </tr> </thead> <tbody> <tr> <td>2008-12-10 11:42:00</td> <td>2008-12-10 11:42:00</td> <td>00000043A</td> <td>00000001A</td> <td></td> <td></td> <td>0</td> <td></td> <td>0</td> <td></td> </tr> <tr> <td>2008-12-10 11:42:00</td> <td>2008-12-10 11:42:00</td> <td>00000043A</td> <td>00000002A</td> <td></td> <td></td> <td>1</td> <td></td> <td>1</td> <td></td> </tr> <tr> <td>2008-12-10 11:42:00</td> <td>2008-12-10 11:42:00</td> <td>00000043A</td> <td>00000003A</td> <td></td> <td></td> <td>0</td> <td></td> <td>0</td> <td></td> </tr> </tbody> </table>	Fecha	Fecha y hora de redacción	DNI_A	DNI del autor	DNI_D	DNI del destinatario	Leída	Denota si ha sido marcada como leída	Descartada	Denota si ha sido marcada como descartada	2008-12-10 11:42:00	2008-12-10 11:42:00	00000043A	00000001A			0		0		2008-12-10 11:42:00	2008-12-10 11:42:00	00000043A	00000002A			1		1		2008-12-10 11:42:00	2008-12-10 11:42:00	00000043A	00000003A			0		0	
Fecha	Fecha y hora de redacción	DNI_A	DNI del autor	DNI_D	DNI del destinatario	Leída	Denota si ha sido marcada como leída	Descartada	Denota si ha sido marcada como descartada																																
2008-12-10 11:42:00	2008-12-10 11:42:00	00000043A	00000001A			0		0																																	
2008-12-10 11:42:00	2008-12-10 11:42:00	00000043A	00000002A			1		1																																	
2008-12-10 11:42:00	2008-12-10 11:42:00	00000043A	00000003A			0		0																																	
Salida esperada	En la primera ocasión la nota deberá desplegarse y mostrar que el usuario con DNI 00000002A la ha leído. En la segunda deberá haberla leído José Manuel Casado Sierra (DNI 00000003A).																																								
Salida observada	<p>La esperada. Tenemos:</p> 																																								



Fecha	Fecha y hora de redacción	DNI_A	DNI del autor	DNI_D	DNI del destinatario	Leída	Denota si ha sido marcada como leída	Descartada	Denota si ha sido marcada como descartada
2008-12-10 11:42:00	2008-12-10 11:42:00	00000043A	00000001A			0		0	
2008-12-10 11:42:00	2008-12-10 11:42:00	00000043A	00000002A			1		1	
2008-12-10 11:42:00	2008-12-10 11:42:00	00000043A	00000003A			1		0	

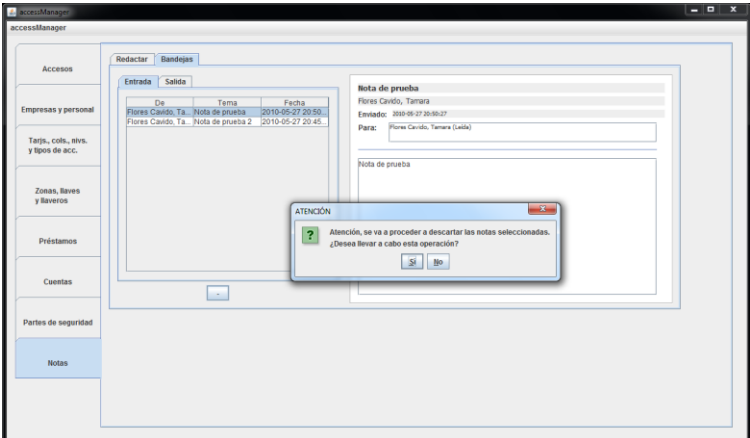
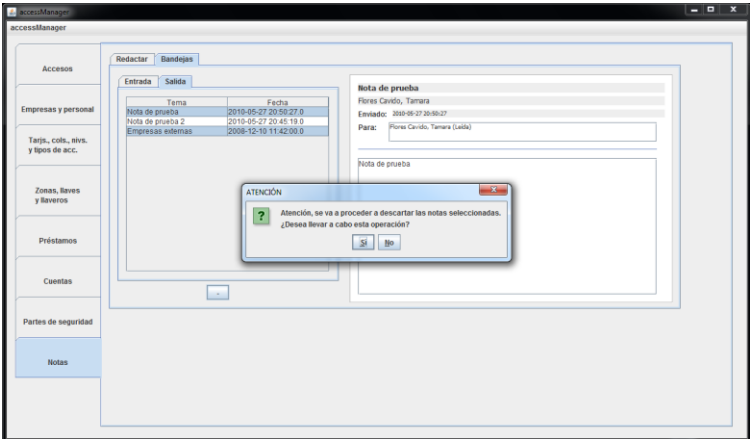
Observamos como los destinatarios no visualizan quiénes han leído la nota, pero el autor sí:

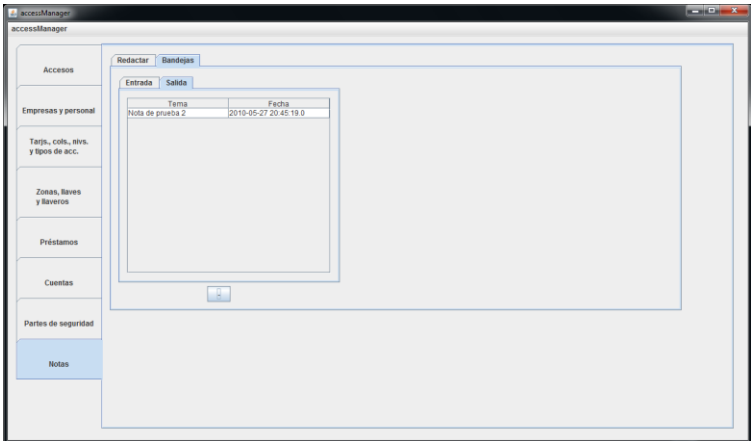


**Evaluación/
conclusión**

El caso de uso '131 - Visualizar nota enviada' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '131 - Visualizar nota enviada'

Caso de prueba	10.07 - Casos de uso 132: Borrar nota enviada
Objetivo	Comprobar que el usuario puede eliminar cualquier conjunto de notas enviadas, permaneciendo éstas a disposición de todos sus destinatarios o siendo eliminadas de la base de datos si todos los usuarios relacionados con ellas las han dado por descartadas.
Actividades	Entrar en la aplicación, situarnos en la pestaña 'Notas' y acceder a la "bandeja de salida", donde daremos de baja varias de ellas y comprobaremos que el sistema se comporta como debe.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. • Utilidad phpMyAdmin.
Datos de entrada	<p>Antes que nada añadiremos dos notas más a la única existente en la bandeja de salida: una destinada a varios usuarios y otra a nosotros. Acto seguido iremos a la bandeja de entrada y descartaremos la nota que nos acabamos de dirigir:</p>  <p>Ahora viene la prueba en sí, eliminando la primera y tercera nota:</p> 
Salida esperada	La eliminación de la primera nota debería provocar la desaparición de ésta de la interfaz y de la base de datos, mientras que la tercera deberá desaparecer de la interfaz pero no de la base de datos.

<p style="text-align: center;">Salida observada</p>	<p>La esperada:</p>  <table border="1" data-bbox="563 734 1295 831"> <thead> <tr> <th>Fecha</th> <th>Fecha y hora de redacción</th> <th>DNI_A</th> <th>DNI del autor</th> <th>DNI_D</th> <th>DNI del destinatario</th> <th>Leída</th> <th>Denota si ha sido marcada como leída</th> <th>Descartada</th> <th>Denota si ha sido marcada como descartada</th> </tr> </thead> <tbody> <tr> <td>2010-05-27 20:45:19</td> <td>2010-05-27 20:45:19</td> <td>00000043A</td> <td>00000043A</td> <td>00000001A</td> <td>00000001A</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>2010-05-27 20:45:19</td> <td>2010-05-27 20:45:19</td> <td>00000043A</td> <td>00000043A</td> <td>00000028A</td> <td>00000028A</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>2010-05-27 20:45:19</td> <td>2010-05-27 20:45:19</td> <td>00000043A</td> <td>00000043A</td> <td>00000043A</td> <td>00000043A</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	Fecha	Fecha y hora de redacción	DNI_A	DNI del autor	DNI_D	DNI del destinatario	Leída	Denota si ha sido marcada como leída	Descartada	Denota si ha sido marcada como descartada	2010-05-27 20:45:19	2010-05-27 20:45:19	00000043A	00000043A	00000001A	00000001A	0	0	0	0	2010-05-27 20:45:19	2010-05-27 20:45:19	00000043A	00000043A	00000028A	00000028A	0	0	0	0	2010-05-27 20:45:19	2010-05-27 20:45:19	00000043A	00000043A	00000043A	00000043A	0	0	0	0
	Fecha	Fecha y hora de redacción	DNI_A	DNI del autor	DNI_D	DNI del destinatario	Leída	Denota si ha sido marcada como leída	Descartada	Denota si ha sido marcada como descartada																															
2010-05-27 20:45:19	2010-05-27 20:45:19	00000043A	00000043A	00000001A	00000001A	0	0	0	0																																
2010-05-27 20:45:19	2010-05-27 20:45:19	00000043A	00000043A	00000028A	00000028A	0	0	0	0																																
2010-05-27 20:45:19	2010-05-27 20:45:19	00000043A	00000043A	00000043A	00000043A	0	0	0	0																																
<p>Evaluación/ conclusión</p>	<p>El caso de uso '132 - Borrar nota enviada' ha sido realizado satisfactoriamente.</p>																																								

Caso de prueba para el caso de uso '132 - Borrar nota enviada'

6.2. De integración

Ninguna prueba de integración realizada ya que son innecesarias. Tal y como nos viene pasando desde hace ya muchas iteraciones, el número de componentes no crece, si no que aumenta en tamaño, en complejidad. La aplicación sigue proporcionando la funcionalidad para la que ha sido diseñada con que damos por asentado que base de datos, API e interfaz gráfica siguen comunicándose correctamente.

6.3. De sistema

Finalizada la iteración procede comprobar que:

- Se han incorporado todos los casos de uso:
 - Algo evidente. En la planificación de la iteración nos propusimos realizar los casos de uso del 126 al 132, y este documento pone de manifiesto que así lo hemos hecho.
- El funcionamiento es correcto y robusto:
 - Aquí debemos remitirnos a los casos de prueba ejecutados y mostrados al comienzo de esta sección, donde todos y cada uno de ellos han sido superados con éxito.
- La documentación es la adecuada:
 - Adecuada y suficiente para comprender qué se ha querido implementar, qué servicios se han querido proporcionar y cómo hemos dado solución a éstos.



7. Evaluación de la iteración

El objetivo de esta iteración era el de crear un sencillo sistema de mensajería que permitiera a los usuarios de accessManager comunicarse, y en estos tres días hemos conseguido implementarlo con unos resultados bastante satisfactorios. El símil con cualquier servicio de correo electrónico lo hace asimilable rápidamente y las pruebas realizadas nos aseguran el funcionamiento deseado. Veamos qué piensa el cliente.

8. Despliegue

El día 31 de marzo de 2010 tiene lugar en la sede de Informática Navarra S.A. una reunión con el fin de recibir la aprobación del desarrollo de la iteración 'C10 - Notas' por parte del cliente.

Nos presentamos en el centro, al igual que en anteriores ocasiones, con el documento de iteración y el prototipo resultante del desarrollo realizado hasta el momento. La reunión comienza con una exposición de la interfaz de la pestaña 'Notas' y qué funcionalidad hemos querido que recoja. Mostramos al cliente cómo debe utilizar los componentes que la conforman para obtener los resultados deseados, haciéndole ver que todos los casos de uso se desarrollan de forma correcta, sin desembocar en ningún tipo de error.

Al igual que ocurrió con la interior iteración, el cliente también está muy satisfecho con el trabajo que hemos realizado y ve muy acertada la idea de llevar el símil de estructura y funcionamiento de una cuenta de correo electrónico a la redacción y visualización de notas. No aprecia ningún elemento que deba ser rediseñado o cambiado de momento, con que lo único que hace es animarnos a seguir trabajando de la misma forma en lo poco que resta del ciclo de desarrollo.

Pamplona, a 31 de marzo de 2010.

Héctor Napal Basterrica
Responsable del proyecto

Pedro Martínez Pérez
*Jefe de área de producción de
Informática Navarra S.A.*



accessManager
Desarrollo de un sistema de control de accesos a un centro

Iteración C11: Base de datos

Fecha de entrega: 2 de abril de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Definiciones, acrónimos y abreviaciones
- 1.4. Estructura del documento

2. Estado del proyecto

3. Análisis

- 3.1. Planificación de la iteración
- 3.2. Análisis de los casos de uso y escenarios

4. Diseño

- 4.1. Caso de uso 133: Importar base de datos
- 4.2. Caso de uso 134: Exportar base de datos

5. Codificación e integración

6. Pruebas

- 6.1. De unidad
- 6.2. De integración
- 6.3. De sistema

7. Evaluación de la iteración

8. Despliegue



1. Introducción

1.1. Propósito

Documentar las actividades llevadas a cabo en estos dos días, de modo que un futuro lector pueda entender a la perfección el trabajo que en ellos se ha realizado.

1.2. Audiencia

Por el momento este artefacto permanecerá en manos del personal de Software Solutions. Una vez que se lleve a cabo la transmisión del producto al cliente, este pasará a las del administrador y podrá ser consultado siempre que éste lo estime oportuno.

1.3. Definiciones, acrónimos y abreviaciones

- SGBD Sistema gestor de bases de datos

1.4. Estructura del documento

Comenzamos la iteración dedicada a la base de datos comentando en qué punto del camino nos encontramos, qué hemos realizado y qué debemos hacer. En el siguiente capítulo traducimos los pasos que deben seguir en una selección de casos de uso a implementar, analizándolos minuciosamente habiendo creado previamente una planificación que detalle todas las actividades a realizar.

Abrimos el capítulo 4 diseñando el caso de uso 133, dedicado a crear un procedimiento por el cual poder importar ficheros con código SQL a accessManager, y lo cerramos con el 134, donde diseñaremos la acción inversa: exportar código SQL de la base de datos de modo que el administrador del sistema tenga una forma sencilla de poder hacer *backups* de la misma. Dedicamos el quinto apartado a hablar de los detalles que habrá que tener en cuenta en la codificación e integración de todo el diseño, y en el sexto, como siempre destinado a la actividad de pruebas, exponemos todas las pruebas que hemos realizado para demostrar que nuestro trabajo ha dado como fruto una implementación robusta y de calidad.

Finalizamos este artefacto realizando una evaluación de la iteración y exponiendo la opinión que el cliente nos ha hecho llegar en la actividad de despliegue.

2. Estado del proyecto

A estas alturas del proceso de desarrollo ya disponemos de toda la funcionalidad que requería el cliente plasmada en la aplicación. Nos queda únicamente dos iteraciones en la fase de Construcción y en ésta abordaremos los dos últimos casos de uso, que permitirán la importación y exportación de la base de datos.

accessManager · Iteración C11: Base de datos · Página 3 de 14



3. Análisis

3.1. Planificación de la iteración

En la siguiente tabla se puede observar el plan de iteración ideado para la ocasión:

PLAN DE ITERACIÓN C11: BASE DE DATOS			
ASPECTO	DESCRIPCIÓN		
Objetivos	Implementar los dos últimos casos de uso descritos en la <i>Especificación inicial de los casos de uso</i> , que permitirán al administrador importar y exportar contenido de la base de datos sin tener que mediar con el SGBD.		
Artefactos de entrada	<ul style="list-style-type: none"> • <i>Especificación inicial de los casos de uso</i> • <i>Planificación temporal y costes detallados del proyecto</i> • Base de datos, API y aplicación desarrolladas en las iteraciones anteriores. 		
Artefactos modificados o creados	<ul style="list-style-type: none"> • Clase Java que constituye el JFrame principal de la aplicación. • El presente documento. 		
Calendarización detallada	Análisis + Pruebas	01/04/10	<ul style="list-style-type: none"> • Planificación de la iteración y gestión de riesgos. • Análisis de los casos de uso y escenarios. • Planificación de las pruebas. • Inicio de la documentación.
	Diseño + Pruebas	01/04/10	<ul style="list-style-type: none"> • Diseño de opciones arquitectónicas. • Diseño de pruebas.
	Codificación + Pruebas	01/04/10	<ul style="list-style-type: none"> • Codificación e integración de la solución. • Ejecución y evaluación de pruebas. • Evaluación del prototipo. • Preparación de la entrega ejecutable.
		02/04/10	
Despliegue	02/04/10	<ul style="list-style-type: none"> • Muestra, prueba y evaluación conjunta con el cliente. • Concreción de los cambios necesarios. • Finalización de toda la documentación. 	

Planificación de la iteración



Nombre de tarea	Duración	Comienzo	Fin	jue 01 abr		
				0	0	0
+ Estudio de oportunidad (inception)	7 días	jue 07/01/10	vie 15/01/10			
+ Elaboración	16 días	lun 18/01/10	lun 08/02/10			
- Construcción	41 días	mar 09/02/10	mar 06/04/10			
+ Iteración C1: La base de datos	4 días	mar 09/02/10	vie 12/02/10			
+ Iteración C2: API para la base de datos	5 días	lun 15/02/10	vie 19/02/10			
+ Iteración C3: Accesos	5 días	lun 22/02/10	vie 26/02/10			
+ Iteración C4: Empresas y personal	5 días	lun 01/03/10	vie 05/03/10			
+ Iteración C5: Tarjetas, tipos de acceso y cargos	3 días	lun 08/03/10	mié 10/03/10			
+ Iteración C6: Zonas, llaves y llaveros	5 días	jue 11/03/10	mié 17/03/10			
+ Iteración C7: Préstamos	2 días	jue 18/03/10	vie 19/03/10			
+ Iteración C8: Cuentas	2 días	lun 22/03/10	mar 23/03/10			
+ Iteración C9: Partes de seguridad	3 días	mié 24/03/10	vie 26/03/10			
+ Iteración C10: Notas	3 días	lun 29/03/10	mié 31/03/10			
- Iteración C11: Base de datos	2 días	jue 01/04/10	vie 02/04/10			
Análisis	0,25 días	jue 01/04/10	jue 01/04/10			
Diseño	0,25 días	jue 01/04/10	jue 01/04/10			
Implementación	1 día	jue 01/04/10	vie 02/04/10			
Pruebas	2 días	jue 01/04/10	vie 02/04/10			
Despliegue	0,5 días	vie 02/04/10	vie 02/04/10			
+ Iteración C12: Diseño, personalización y configuración	2 días	lun 05/04/10	mar 06/04/10			
Capacidad operacional	0 días	mar 06/04/10	mar 06/04/10			
+ Transición	8 días	mié 07/04/10	vie 16/04/10			

Recursos	Temporales		2 días
	Humanos		1 persona
	Software	Sistema operativo	Microsoft Windows 7
		Ofimática	Microsoft Word 2010 Adobe Acrobat Professional 8
		Diseño	Visual Paradigm 4.2
Codificación		NetBeans 6.8	
Hardware		1 PC	

Casos de uso y escenarios implementados	133 - Importar base de datos 134 - Exportar base de datos
--	--

Planificación de la iteración

3.2. Análisis de los casos de uso y escenarios

Dos son los casos de uso que tenemos que implementar, ambos relacionados con la administración de la base de datos y los cuales permitirán importar y exportar código SQL de ésta.



Caso de uso 133: Importar base de datos	
Descripción	Cabe la posibilidad de que la base de datos deba ser restaurada o surja un conjunto de registros que deban ser insertados. Parece muy mala idea ir insertando todas y cada una de las tuplas una a una mediante la interfaz gráfica, con que dotaremos de la funcionalidad de importación que puede proporcionar el SGBD pero sin tener que pasar por éste.
Actores principal	Administrador.
Pre-condiciones	El administrador de la base de datos desea importar un fichero con código SQL con sentencias a ejecutar sobre la base de datos.
Flujos	<p>Básico</p> <ol style="list-style-type: none"> 1. El usuario indica al sistema que desea importar un fichero SQL. 2. El sistema abre un diálogo en el que el usuario deberá buscar y seleccionar el fichero a importar. (*) 3. El sistema realiza la importación de todo el código existente en el fichero indicado. (*)
	<p>Alternativos (*)</p> <p>El usuario cancela la selección del fichero:</p> <ul style="list-style-type: none"> • El sistema no realiza ninguna acción y el caso de uso concluye. <p>El fichero SQL contiene algún tipo de error:</p> <ul style="list-style-type: none"> • El sistema avisa del error que ha sucedido y aborta la importación de código en el punto en que éste se produjo.
Post-condiciones	El administrador importa a la base de datos todo el código SQL que residía en el fichero seleccionado.

Caso de uso 133: Importar base de datos



Caso de uso 134: Exportar base de datos					
Descripción	La población residente en la base de datos irá aumentando a medida que los días se sucedan. El administrador no puede arriesgarse a perder toda esta valiosa información con que deberá realizar copias de la misma periódicamente. Con este caso de uso pretendemos proporcionarle una vía fácil y cómo mediante la cual exportar toda la población a un fichero sin tener que mediar con el SGBD.				
Actores principal	Administrador.				
Pre-condiciones	El administrador desea exportar todo el contenido de la base de datos a un fichero.				
Flujos	<table border="1"> <tr> <td>Básico</td> <td> <ol style="list-style-type: none"> 1. El administrador transmite al sistema su intención. 2. El sistema abre un diálogo en el que el administrador deberá indicar la ruta y el nombre del fichero en el que depositar toda la información. 3. El administrador indicar la ruta y el sistema realiza la exportación solicitada. (*) </td> </tr> <tr> <td>Alternativos (*)</td> <td> El administrador cancela el cuadro de selección de fichero: <ul style="list-style-type: none"> • El sistema no realiza ninguna exportación. </td> </tr> </table>	Básico	<ol style="list-style-type: none"> 1. El administrador transmite al sistema su intención. 2. El sistema abre un diálogo en el que el administrador deberá indicar la ruta y el nombre del fichero en el que depositar toda la información. 3. El administrador indicar la ruta y el sistema realiza la exportación solicitada. (*) 	Alternativos (*)	El administrador cancela el cuadro de selección de fichero: <ul style="list-style-type: none"> • El sistema no realiza ninguna exportación.
	Básico	<ol style="list-style-type: none"> 1. El administrador transmite al sistema su intención. 2. El sistema abre un diálogo en el que el administrador deberá indicar la ruta y el nombre del fichero en el que depositar toda la información. 3. El administrador indicar la ruta y el sistema realiza la exportación solicitada. (*) 			
Alternativos (*)	El administrador cancela el cuadro de selección de fichero: <ul style="list-style-type: none"> • El sistema no realiza ninguna exportación. 				
Post-condiciones	El administrador consigue lo que perseguía: crear un fichero con toda la población existente en la base de datos.				

Caso de uso 127: Visualizar notas recibidas

4. Diseño

4.1. Caso de uso 133: Importar base de datos

Los dos casos de uso a diseñar se basan casi exclusivamente en crear procedimientos de lectura y escritura en ficheros, que lean sentencias y las ejecuten o que las creen a raíz de los datos contenidos en la base de datos y las escriban.

En este primer caso debemos importar un fichero que estará compuesto por sentencias SQL. Éstas sentencias terminan con el carácter ';', con que lo que tenemos que hacer es ir leyendo y ejecutándolas:

- | |
|---|
| <ol style="list-style-type: none"> 1. Recoger el fichero a importar. 2. Abrir el fichero. 3. <u>Mientras</u> no FDF
 Crear <i>query</i> leyendo caracteres hasta encontrar el ';'.
 Ejecutar <i>query</i>.
 <u>Fin Mientras</u> 4. Cerrar el fichero. 5. Refrescar toda la información de la aplicación. |
|---|

accessManager · Iteración C11: Base de datos · Página 7 de 14



La lectura y escritura de ficheros puede ser realizada con el manejo de la API Standard de Java, con que no será necesario añadir ningún método a la creada por nosotros.

La lectura y ejecución de sentencias SQL queda descrita a la perfección en estos dos diagramas:

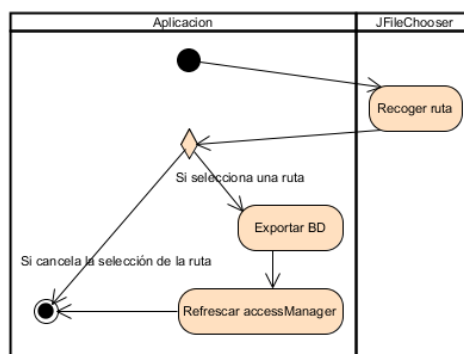


Diagrama de actividades

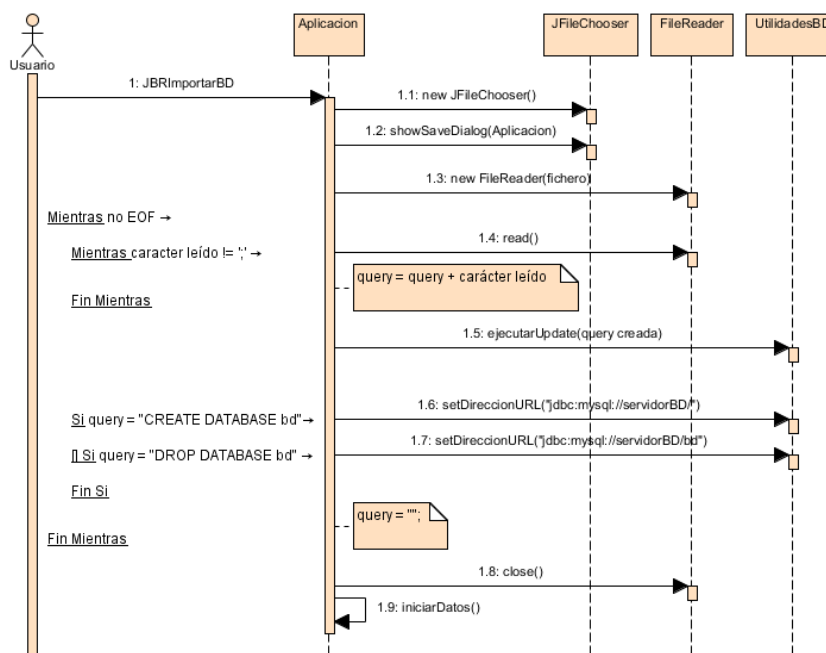


Diagrama de secuencia

El recorrido que realizamos es muy simple. La única dificultad que encontraremos es la de evitar caracteres especiales, comentarios,... dentro del fichero, pero si el lector acude al código proporcionado con esta memoria verá que con un par de comprobaciones nuestro problema desaparecerá.

Otro tema a tener en cuenta es sobre qué dirección ejecutamos la sentencia, ya que si tenemos la base de datos creada deberá hacerse sobre "jdbc:mysql://servidorBD/bd" y en caso contrario sobre "jdbc:mysql://servidorBD/".



4.2. Caso de uso 134: Exportar base de datos

Para exportar la población de la base de datos deberemos obtener todas y cada una de las filas que pueblan las tablas de la base de datos y escribirlas en un fichero:

1. Obtener el conjunto de tablas existentes en un ArrayList.
2. Abrir el fichero.
3. Para cada tabla existente en el ArrayList
Obtener todas las tuplas que residen en ella.
Crear una sentencia de tipo INSERT por cada una y la escribirla en el fichero.
Fin Mientras
4. Cerrar el fichero.

En este caso tendríamos los siguientes diagramas de actividades y de secuencia:

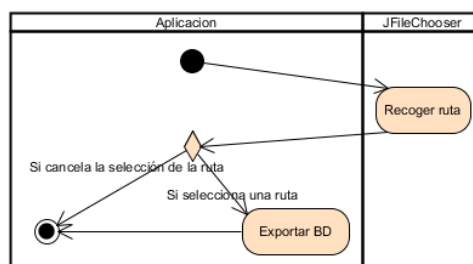


Diagrama de actividades

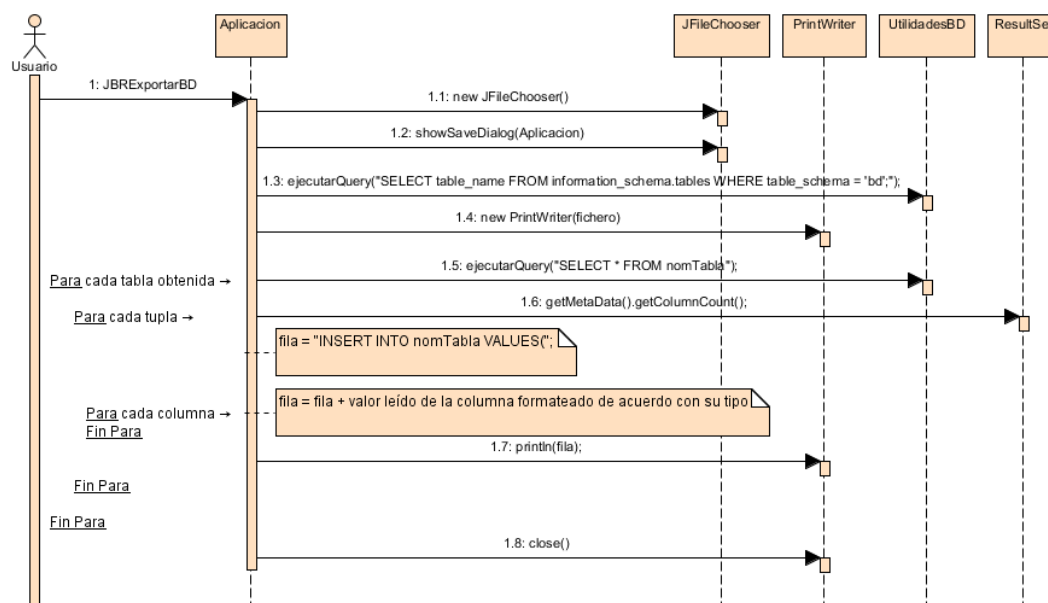


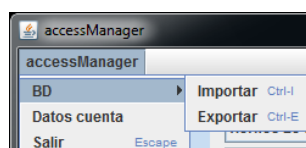
Diagrama de secuencia



5. Codificación e integración

Es la iteración más sencilla de codificar con diferencia, pues solo requiere programar un par de consultas y luego saber movernos por los resultados obtenidos (ResultSets) y ficheros (leer y escribir).

No hemos comentado nada al respecto, pero ambos casos de uso serán utilizados poco frecuentemente en comparación con el resto y al no requerir de interfaz gráfica (salvo los JFileChoosers) haremos que se inicien a través del menú 'accessManager' situado en la parte superior izquierda del frame principal:



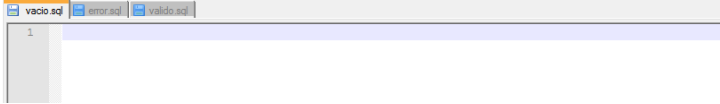
Interfaz del menú 'accessManager'

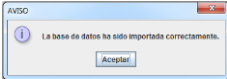
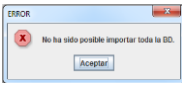
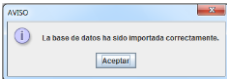
Aplicación.java es el encargado de ejecutar en exclusiva los casos de uso descritos y en su integración con el sistema solo habrá que tener en cuenta la base de datos y una de las clases de la API (*UtilidadesBD.java*).

6. Pruebas

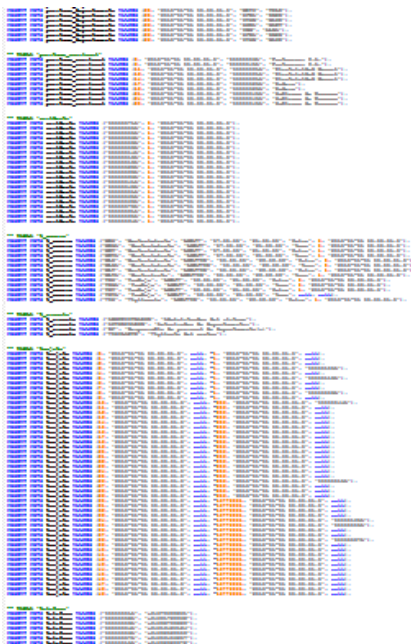
6.1. De unidad

La etapa de pruebas va a ser muy rápida y sencilla de llevar a cabo, pues solo disponemos de dos casos de uso y que además presentan pocas variantes (flujos de sucesos) posibles:

Caso de prueba	11.01 - Casos de uso 133: Importar base de datos
Objetivo	Comprobar que podemos importar un fichero SQL a través del menú 'accessManager'.
Actividades	Loguear en el sistema, desplegar el menú 'accessManager' y probar a importar ficheros válidos y erróneos.
Técnicas y herramientas	<ul style="list-style-type: none">• Aplicación accessManager.• Base de datos con población de prueba.• Utilidad phpMyAdmin.• Aplicación Notepad++.
Datos de entrada	Diseñaremos tres ficheros SQL: <ul style="list-style-type: none">• Fichero vacío 

	<ul style="list-style-type: none"> Fichero con código SQL erróneo. <pre> 1 INSERT INTO Nivel VALUES (8,'2010-05-28 00:00:00',null,'Nivel de prueba'); 2 INSERT INTO Nivel VALUES (8,'2010-05-28 00:00:00',null,'Nivel de prueba repetido'); 3 INSERT INTO Nivel VALUES (9,'2010-05-28 00:00:00',null,'Nivel de prueba que no debiera insertarse');</pre> Fichero con código SQL válido. <pre> 1 INSERT INTO Nivel VALUES (6,'2010-05-28 00:00:00',null,'Nivel de prueba'); 2 INSERT INTO Nivel VALUES (7,'2010-05-28 00:00:00',null,'Nivel de prueba');</pre> 																																																																																
Salida esperada	<p>El primer fichero no producirá ningún efecto, el segundo dará error al introducir la segunda tupla (insertará la primera) y el tercero será importado correctamente.</p>																																																																																
Salida observada	<p>La esperada:</p> <ul style="list-style-type: none"> En el primer caso no desemboca en error, pero la base de datos no es modificada:  La segunda prueba desemboca en error tras ser introducido el código SQL hasta que éste se produce:  <table border="1" data-bbox="564 1133 1273 1216"> <thead> <tr> <th>Cod_Niv</th> <th>Nivel de acceso</th> <th>F_Alta_N</th> <th>Fecha y hora de inserción del nivel</th> <th>F_Baja_N</th> <th>Fecha y hora de baja del nivel</th> <th>Descripcion</th> <th>Descripción del nivel de acceso</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Acceso restringido: Zonas comunes</td> <td></td> </tr> <tr> <td>1</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Acceso total: Zonas comunes + Host</td> <td></td> </tr> <tr> <td>8</td> <td></td> <td>2010-05-28 00:00:00</td> <td></td> <td>NULL</td> <td></td> <td>Nivel de prueba</td> <td></td> </tr> </tbody> </table> <ul style="list-style-type: none"> Al importar un fichero SQL totalmente correcto el sistema reacciona como esperabamos:  <table border="1" data-bbox="564 1469 1283 1597"> <thead> <tr> <th>Cod_Niv</th> <th>Nivel de acceso</th> <th>F_Alta_N</th> <th>Fecha y hora de inserción del nivel</th> <th>F_Baja_N</th> <th>Fecha y hora de baja del nivel</th> <th>Descripcion</th> <th>Descripción del nivel de acceso</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Acceso restringido: Zonas comunes</td> <td></td> </tr> <tr> <td>1</td> <td></td> <td>2010-01-01 00:00:01</td> <td></td> <td>NULL</td> <td></td> <td>Acceso total: Zonas comunes + Host</td> <td></td> </tr> <tr> <td>6</td> <td></td> <td>2010-05-28 00:00:00</td> <td></td> <td>NULL</td> <td></td> <td>Nivel de prueba</td> <td></td> </tr> <tr> <td>7</td> <td></td> <td>2010-05-28 00:00:00</td> <td></td> <td>NULL</td> <td></td> <td>Nivel de prueba</td> <td></td> </tr> <tr> <td>8</td> <td></td> <td>2010-05-28 00:00:00</td> <td></td> <td>NULL</td> <td></td> <td>Nivel de prueba</td> <td></td> </tr> </tbody> </table>	Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y hora de inserción del nivel	F_Baja_N	Fecha y hora de baja del nivel	Descripcion	Descripción del nivel de acceso	0		2010-01-01 00:00:01		NULL		Acceso restringido: Zonas comunes		1		2010-01-01 00:00:01		NULL		Acceso total: Zonas comunes + Host		8		2010-05-28 00:00:00		NULL		Nivel de prueba		Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y hora de inserción del nivel	F_Baja_N	Fecha y hora de baja del nivel	Descripcion	Descripción del nivel de acceso	0		2010-01-01 00:00:01		NULL		Acceso restringido: Zonas comunes		1		2010-01-01 00:00:01		NULL		Acceso total: Zonas comunes + Host		6		2010-05-28 00:00:00		NULL		Nivel de prueba		7		2010-05-28 00:00:00		NULL		Nivel de prueba		8		2010-05-28 00:00:00		NULL		Nivel de prueba	
Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y hora de inserción del nivel	F_Baja_N	Fecha y hora de baja del nivel	Descripcion	Descripción del nivel de acceso																																																																										
0		2010-01-01 00:00:01		NULL		Acceso restringido: Zonas comunes																																																																											
1		2010-01-01 00:00:01		NULL		Acceso total: Zonas comunes + Host																																																																											
8		2010-05-28 00:00:00		NULL		Nivel de prueba																																																																											
Cod_Niv	Nivel de acceso	F_Alta_N	Fecha y hora de inserción del nivel	F_Baja_N	Fecha y hora de baja del nivel	Descripcion	Descripción del nivel de acceso																																																																										
0		2010-01-01 00:00:01		NULL		Acceso restringido: Zonas comunes																																																																											
1		2010-01-01 00:00:01		NULL		Acceso total: Zonas comunes + Host																																																																											
6		2010-05-28 00:00:00		NULL		Nivel de prueba																																																																											
7		2010-05-28 00:00:00		NULL		Nivel de prueba																																																																											
8		2010-05-28 00:00:00		NULL		Nivel de prueba																																																																											
Evaluación/ conclusión	<p>El caso de uso '133 - Importar base de datos' ha sido realizado satisfactoriamente.</p> <p><i>Caso de prueba para el caso de uso '133 - Importar base de datos'</i></p>																																																																																



Caso de prueba	11.02 - Casos de uso 134: Exportar base de datos																																																																																											
Objetivo	Verificar que accessManager es capaz de exportar toda la población que puebla su base de datos si el usuario lo desea.																																																																																											
Actividades	Acceder al sistema, desplegar el menú 'accessManager' y exportar el contenido existente en la base de datos.																																																																																											
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos resultante del caso de prueba anterior. • Utilidad phpMyAdmin. • Aplicación Notepad++. 																																																																																											
Datos de entrada	La base de datos.																																																																																											
Salida esperada	Un fichero con toda la población existente en la base de datos. Se visualizará si éste contiene a los registros existentes y además se verificará una tabla que pueda incitar a errores (que tenga varios tipos de datos), como es el caso de Departamento.																																																																																											
Salida observada	<p>La esperada:</p>  <table border="1"> <thead> <tr> <th>Cod_Emp</th> <th>Cod_Dep</th> <th>Nombre</th> <th>Cod_Acc</th> <th>Periodicidad</th> <th>F_Ult_Actu</th> <th>DNI</th> </tr> <tr> <th>Código de empresa</th> <th>Código de departamento</th> <th>Nombre del departamento</th> <th>Código de acceso que posee</th> <th>Periodicidad con la que renueva la lista de personal que trabaja en el centro</th> <th>Fecha y hora de la última actualización de personal realizada</th> <th>DNI del autorizador responsable del departamento</th> </tr> </thead> <tbody> <tr> <td>CADO</td> <td>DICO</td> <td>Diseño y construcción</td> <td>THT</td> <td>60</td> <td>2010-01-07 00:00:00</td> <td>00000028A</td> </tr> <tr> <td>CIS</td> <td>VA</td> <td>Vigilancia con arma</td> <td>VIG</td> <td>365</td> <td>2010-01-31 00:00:00</td> <td>00000018A</td> </tr> <tr> <td>COMF</td> <td>DEPV</td> <td>Departamento de vertas</td> <td>THCA</td> <td>60</td> <td>2010-01-03 00:00:00</td> <td>00000043A</td> </tr> <tr> <td>CONA</td> <td>ATEC</td> <td>Asistencia Técnica</td> <td>THTA</td> <td>60</td> <td>2010-01-09 00:00:00</td> <td>00000043A</td> </tr> <tr> <td>CONA</td> <td>CAU</td> <td>Centro Atención Usuario 24x7</td> <td>MLT</td> <td>30</td> <td>2010-01-02 00:00:00</td> <td>00000018A</td> </tr> <tr> <td>CONA</td> <td>EMER</td> <td>Emergencias 24x7</td> <td>MLT</td> <td>30</td> <td>2010-01-02 00:00:00</td> <td>00000028A</td> </tr> <tr> <td>CONA</td> <td>MANT</td> <td>Mantenimiento H/S 24x7</td> <td>MLTA</td> <td>60</td> <td>2010-01-01 00:00:00</td> <td>00000018A</td> </tr> <tr> <td>CONA</td> <td>SIST</td> <td>Sistemas</td> <td>MHTA</td> <td>60</td> <td>2010-01-03 00:00:00</td> <td>00000028A</td> </tr> <tr> <td>CONT</td> <td>DIRE</td> <td>Diseño de redes</td> <td>THT</td> <td>365</td> <td>2010-01-09 00:00:00</td> <td>00000028A</td> </tr> <tr> <td>CONT</td> <td>MARE</td> <td>Mantenimiento de redes</td> <td>MHT</td> <td>30</td> <td>2010-01-01 00:00:00</td> <td>00000028A</td> </tr> <tr> <td>ELEC</td> <td>CONS</td> <td>Consultoría</td> <td>THTA</td> <td>30</td> <td>2010-01-08 00:00:00</td> <td>00000028A</td> </tr> </tbody> </table> <p>...</p>	Cod_Emp	Cod_Dep	Nombre	Cod_Acc	Periodicidad	F_Ult_Actu	DNI	Código de empresa	Código de departamento	Nombre del departamento	Código de acceso que posee	Periodicidad con la que renueva la lista de personal que trabaja en el centro	Fecha y hora de la última actualización de personal realizada	DNI del autorizador responsable del departamento	CADO	DICO	Diseño y construcción	THT	60	2010-01-07 00:00:00	00000028A	CIS	VA	Vigilancia con arma	VIG	365	2010-01-31 00:00:00	00000018A	COMF	DEPV	Departamento de vertas	THCA	60	2010-01-03 00:00:00	00000043A	CONA	ATEC	Asistencia Técnica	THTA	60	2010-01-09 00:00:00	00000043A	CONA	CAU	Centro Atención Usuario 24x7	MLT	30	2010-01-02 00:00:00	00000018A	CONA	EMER	Emergencias 24x7	MLT	30	2010-01-02 00:00:00	00000028A	CONA	MANT	Mantenimiento H/S 24x7	MLTA	60	2010-01-01 00:00:00	00000018A	CONA	SIST	Sistemas	MHTA	60	2010-01-03 00:00:00	00000028A	CONT	DIRE	Diseño de redes	THT	365	2010-01-09 00:00:00	00000028A	CONT	MARE	Mantenimiento de redes	MHT	30	2010-01-01 00:00:00	00000028A	ELEC	CONS	Consultoría	THTA	30	2010-01-08 00:00:00	00000028A
Cod_Emp	Cod_Dep	Nombre	Cod_Acc	Periodicidad	F_Ult_Actu	DNI																																																																																						
Código de empresa	Código de departamento	Nombre del departamento	Código de acceso que posee	Periodicidad con la que renueva la lista de personal que trabaja en el centro	Fecha y hora de la última actualización de personal realizada	DNI del autorizador responsable del departamento																																																																																						
CADO	DICO	Diseño y construcción	THT	60	2010-01-07 00:00:00	00000028A																																																																																						
CIS	VA	Vigilancia con arma	VIG	365	2010-01-31 00:00:00	00000018A																																																																																						
COMF	DEPV	Departamento de vertas	THCA	60	2010-01-03 00:00:00	00000043A																																																																																						
CONA	ATEC	Asistencia Técnica	THTA	60	2010-01-09 00:00:00	00000043A																																																																																						
CONA	CAU	Centro Atención Usuario 24x7	MLT	30	2010-01-02 00:00:00	00000018A																																																																																						
CONA	EMER	Emergencias 24x7	MLT	30	2010-01-02 00:00:00	00000028A																																																																																						
CONA	MANT	Mantenimiento H/S 24x7	MLTA	60	2010-01-01 00:00:00	00000018A																																																																																						
CONA	SIST	Sistemas	MHTA	60	2010-01-03 00:00:00	00000028A																																																																																						
CONT	DIRE	Diseño de redes	THT	365	2010-01-09 00:00:00	00000028A																																																																																						
CONT	MARE	Mantenimiento de redes	MHT	30	2010-01-01 00:00:00	00000028A																																																																																						
ELEC	CONS	Consultoría	THTA	30	2010-01-08 00:00:00	00000028A																																																																																						



	<pre> -- TABLA 'departamento' INSERT INTO departamento VALUES ('CIBO', 'CIBO', 'Diablo y construcción', 'TET', 60, '2010-01-07 00:00:00.0', '00000028'); INSERT INTO departamento VALUES ('CIS', 'VA', 'Vigilancia con arma', 'VIG', 300, '2010-01-31 00:00:00.0', '00000018'); INSERT INTO departamento VALUES ('COMF', 'DEPU', 'Departamento de ventas', 'TRCA', 60, '2010-01-03 00:00:00.0', '00000438'); INSERT INTO departamento VALUES ('CORA', 'ATEC', 'Asistencia Técnica', 'TEA', 60, '2010-01-09 00:00:00.0', '00000464'); INSERT INTO departamento VALUES ('CORA', 'COP', 'Centro Atención Operativa 24x7', 'MCT', 30, '2010-01-02 00:00:00.0', '00000158'); INSERT INTO departamento VALUES ('CORA', 'EMER', 'Emergencias 24x7', 'MCT', 30, '2010-01-02 00:00:00.0', '00000288'); INSERT INTO departamento VALUES ('CORA', 'MANT', 'Mantenimiento 24x7', 'MLTA', 60, '2010-01-01 00:00:00.0', '00000158'); INSERT INTO departamento VALUES ('CORA', 'REIT', 'Reparación', 'MCT', 60, '2010-01-02 00:00:00.0', '00000288'); INSERT INTO departamento VALUES ('COMT', 'CISE', 'Diablo de redes', 'TET', 300, '2010-01-09 00:00:00.0', '00000028'); INSERT INTO departamento VALUES ('COMT', 'MANT', 'Mantenimiento de redes', 'MCT', 30, '2010-01-01 00:00:00.0', '00000288'); INSERT INTO departamento VALUES ('ELEC', 'COM', 'Comunicación', 'TEA', 30, '2010-01-02 00:00:00.0', '00000288'); INSERT INTO departamento VALUES ('ELEC', 'INSTA', 'Instalación y mantenimiento de equipos', 'MCT', 30, '2010-01-25 00:00:00.0', '00000188'); INSERT INTO departamento VALUES ('ELRO', 'EMER', 'Servicio de emergencia 24h', 'MLTA', 30, '2010-01-01 00:00:00.0', '00000288'); INSERT INTO departamento VALUES ('ELRO', 'TET', 'Instalación de equipos', 'MCT', 30, '2010-01-22 00:00:00.0', '00000188'); INSERT INTO departamento VALUES ('ELRO', 'MANT', 'Mantenimiento', 'MCT', 120, '2010-01-01 00:00:00.0', '00000438'); INSERT INTO departamento VALUES ('GOTO', 'INSTA', 'Dep. Habilitación', 'TRCA', 30, '2010-01-30 00:00:00.0', '00000158'); INSERT INTO departamento VALUES ('IEM', 'ALIS', 'Dep. Almacenamiento y datos', 'MCT', 60, '2010-01-01 00:00:00.0', '00000188'); INSERT INTO departamento VALUES ('IEM', 'COM', 'Dep. Comunicaciones integradas', 'MCT', 30, '2010-01-28 00:00:00.0', '00000288'); INSERT INTO departamento VALUES ('IEM', 'ESAR', 'Dep. Estrategia y arquitectura', 'TRC', 60, '2010-01-02 00:00:00.0', '00000438'); INSERT INTO departamento VALUES ('IEM', 'MIDI', 'Dep. Multimedia', 'MCT', 60, '2010-01-28 00:00:00.0', '00000438'); INSERT INTO departamento VALUES ('IEM', 'SEPR', 'Dep. Seguridad y privacidad', 'MCT', 60, '2010-01-30 00:00:00.0', '00000188'); INSERT INTO departamento VALUES ('IEM', 'CISO', 'Consultoría legal y organizativa', 'TRCA', 30, '2010-01-21 00:00:00.0', '00000438'); INSERT INTO departamento VALUES ('IEM', 'CTIC', 'Consultoría técnica y comunicaciones', 'TET', 30, '2010-01-30 00:00:00.0', '00000288'); INSERT INTO departamento VALUES ('IEM', 'FYRE', 'Formación y pruebas', 'TRCA', 30, '2010-01-21 00:00:00.0', '00000438'); INSERT INTO departamento VALUES ('IEM', 'LOIS', 'Legislación e infraestructura', 'TRC', 120, '2010-01-22 00:00:00.0', '00000188'); INSERT INTO departamento VALUES ('IEM', 'INSTA', 'Instalación y mantenimiento', 'MCT', 30, '2010-01-02 00:00:00.0', '00000288'); INSERT INTO departamento VALUES ('IEM', 'CREO', 'Desarrollo de software', 'MCT', 30, '2010-01-01 00:00:00.0', '00000188'); INSERT INTO departamento VALUES ('LILE', 'LISE', 'Simplicia general', 'MCT', 30, '2010-02-02 00:00:00.0', '00000188'); INSERT INTO departamento VALUES ('METS', 'SESI', 'Dep. Seguridad contra incendios', 'MCT', 60, '2010-01-01 00:00:00.0', '00000188'); INSERT INTO departamento VALUES ('METS', 'TRES', 'Dep. Telecomunicaciones', 'MCT', 30, '2010-01-01 00:00:00.0', '00000288'); INSERT INTO departamento VALUES ('ORGA', 'ORGD', 'Dep. Redes', 'MCT', 60, '2010-01-28 00:00:00.0', '00000438'); INSERT INTO departamento VALUES ('ORGA', 'MENS', 'Herramienta', 'TRCA', 60, '2010-01-03 00:00:00.0', '00000438'); INSERT INTO departamento VALUES ('ORGA', 'MENS', 'Herramienta contra incendios', 'MCT', 60, '2010-01-03 00:00:00.0', '00000188'); </pre>
Evaluación/ conclusión	El caso de uso '134 - Exportar base de datos' ha sido realizado satisfactoriamente.

Caso de prueba para el caso de uso '134 - Exportar base de datos'

6.2. De integración

Como ya comentamos en la sección 5, ésta es la iteración que menor grado de relación existe entre sus componentes software. Hasta el momento todos han interactuado perfectamente y en esta iteración, que no solo toma parte la API mediante una clase, se han desarrollado ambas pruebas de unidad sin observar errores, con lo que no vemos necesario realizar ninguna más.

6.3. De sistema

Los casos de uso 133 y 134 han sido implementados y probados con éxito. Además, los acompañamos de este artefacto, que ayudará a que cualquier persona ajena al proyecto pueda entender perfectamente qué problemas tratan y cómo les dan solución.

7. Evaluación de la iteración

Ahora sí que podemos decir que tenemos la fase de Construcción casi terminada. Ésta iteración no implicaba a priori ninguna complicación seria y así ha sido. Tras su desarrollo podemos decir que la hemos completado con éxito, cumpliendo todos los objetivos propuestos y otorgando a accessManager de las funcionalidades de importar y exportar código SQL, que en un futuro serán de mucha utilidad al administrador del sistema. Veremos oportuno iniciar la iteración C12 si el cliente da su visto bueno.




8. Despliegue

Tal y como se estableció en la calendarización del proyecto, el mismo día que finaliza la fase de pruebas tiene lugar la reunión de despliegue correspondiente a esta iteración. La misma es de duración muy corta, pero suficiente para exponer al cliente todo el trabajo realizado y que éste nos dé su visto bueno. Nos citamos para dentro de cuatro días, donde le mostraremos el sistema ya definido casi al completo a falta de mínimos detalles.

Pamplona, a 2 de abril de 2010.



Héctor Napal Basterrica
Responsable del proyecto



Pedro Martínez Pérez
*Jefe de área de producción de
Informática Navarra S.A.*



accessManager

Desarrollo de un sistema de control de accesos a un centro

Iteración C12: Diseño, personalización y configuración

Fecha de entrega: 6 de abril de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Estructura del documento

2. Estado del proyecto

3. Análisis

- 3.1. Planificación de la iteración
- 3.2. Análisis de los casos de uso y escenarios

4. Diseño

- 4.1. Visualización de componentes gráficos
- 4.2. Refresco de datos
- 4.3. Rediseño de la aplicación

5. Codificación e integración

- 5.1. Identificación
- 5.2. Accesos
- 5.3. Empresas y personal
- 5.4. Tarjetas, tipos de acceso y cargos
- 5.5. Zonas, llaves y llaveros
- 5.6. Préstamos
- 5.7. Cuentas
- 5.8. Partes de seguridad
- 5.9. Notas
- 5.10. Menú 'accessManager'

6. Pruebas

- 6.1. De unidad
- 6.2. De integridad
- 6.3. De sistema

7. Evaluación de la iteración

8. Despliegue



1. Introducción

1.1. Propósito

El propósito con el que nace este artefacto es el de documentar todas las actividades de rediseño, personalización y configuración que llevaremos a cabo en accessManager para su puesta a punto.

1.2. Audiencia

Este escrito va dirigido al cliente para el que se desarrolla este proyecto, de modo que pueda consultarlo para el futuro estudio o mantenimiento del sistema.

De carácter interno, salvo en la actividad de despliegue, será proporcionado al cliente junto con el resto de documentación en la segunda iteración de transición, en el momento de la entrega.

1.3. Estructura del documento

La última iteración de la fase de Construcción comienza como todas las anteriores: realizando un breve inciso sobre en qué punto del proceso de desarrollo nos encontramos para así pasar en la siguiente sección a realizar el análisis de todo el trabajo que desarrollaremos en la fase de diseño. Ésta ha sido documentada en el cuarto apartado de este artefacto, creando tres secciones dentro de éste en las que hablamos de cómo conseguiremos una correcta visualización de los componentes gráficos, un aceptable refresco de datos y una interfaz más “amigable” que la que disponemos.

En el capítulo 5 hablamos de los detalles que tendremos que tener en cuenta en la fase de codificación y hacemos una comparativa de los cambios visuales ha sufrido la aplicación en esta iteración.

El siguiente capítulo expone las pruebas que certifican que las fases de análisis, diseño y codificación han sido realizadas correctamente.

Para finalizar redactamos una pequeña evaluación de los resultados que hemos obtenido con esta iteración y los hechos reseñables que hemos extraído de la actividad de despliegue con el cliente.

2. Estado del proyecto

Llegamos a la última iteración de la fase de Construcción y accessManager ya empieza a tener muy buen aspecto. Tenemos todos los casos de uso implementados pero en estos momentos pueden ejecutarlos cualquier tipo de usuario, cuando esto en realidad no debería ser posible. En éstos dos días resolveremos este problema, además de hacer un buen “lavado de imagen” a la aplicación.



3. Análisis

3.1. Planificación de la iteración

A continuación mostramos el plan de iteración desarrollado para guiar nuestros pasos en estos dos días:

PLAN DE ITERACIÓN C12: DISEÑO, PERSONALIZACIÓN Y CONFIGURACIÓN			
ASPECTO	DESCRIPCIÓN		
Objetivos	Configurar accessManager de modo que su contenido se ajuste al que cada usuario deba poder acceder según el rol que desempeña. Además, diseñaremos una estrategia para que los datos de la aplicación se mantengan actualizados y la rediseñaremos de modo que podamos hacerla más usable y atractiva.		
Artefactos de entrada	<ul style="list-style-type: none"> • <i>Especificación inicial de los casos de uso</i> • <i>Planificación temporal y costes detallados del proyecto</i> • Base de datos, API y aplicación desarrolladas en las iteraciones anteriores. 		
Artefactos modificados o creados	<ul style="list-style-type: none"> • Clase Java que constituye el JFrame principal de la aplicación. • El presente documento. 		
Calendarización detallada	Análisis + Pruebas	05/04/10	<ul style="list-style-type: none"> • Planificación de la iteración y gestión de riesgos. • Planificación de las pruebas. • Inicio de la documentación.
	Diseño + Pruebas	05/04/10	<ul style="list-style-type: none"> • Diseño de opciones arquitectónicas. • Diseño de pruebas.
	Codificación + Pruebas	05/04/10	<ul style="list-style-type: none"> • Codificación e integración de la solución. • Ejecución y evaluación de pruebas. • Evaluación del prototipo. • Preparación de la entrega ejecutable.
		06/04/10	
Despliegue	06/04/10	<ul style="list-style-type: none"> • Muestra, prueba y evaluación conjunta con el cliente. • Concreción de los cambios necesarios. • Finalización de toda la documentación. 	

Planificación de la iteración



Nombre de tarea	Duración	Comienzo	Fin	0	lun 05 abr	0	mié 07	0
Estudio de oportunidad (inception)	7 días	jue 07/01/10	vie 15/01/10					
Elaboración	16 días	lun 18/01/10	lun 08/02/10					
Construcción	41 días	mar 09/02/10	mar 06/04/10					
Iteración C1: La base de datos	4 días	mar 09/02/10	vie 12/02/10					
Iteración C2: API para la base de datos	5 días	lun 15/02/10	vie 19/02/10					
Iteración C3: Accesos	5 días	lun 22/02/10	vie 26/02/10					
Iteración C4: Empresas y personal	5 días	lun 01/03/10	vie 05/03/10					
Iteración C5: Tarjetas, tipos de acceso y cargos	3 días	lun 08/03/10	mié 10/03/10					
Iteración C6: Zonas, llaves y llaveros	5 días	jue 11/03/10	mié 17/03/10					
Iteración C7: Préstamos	2 días	jue 18/03/10	vie 19/03/10					
Iteración C8: Cuentas	2 días	lun 22/03/10	mar 23/03/10					
Iteración C9: Partes de seguridad	3 días	mié 24/03/10	vie 26/03/10					
Iteración C10: Notas	3 días	lun 29/03/10	mié 31/03/10					
Iteración C11: Base de datos	2 días	jue 01/04/10	vie 02/04/10					
Iteración C12: Diseño, personalización y configuración	2 días	lun 05/04/10	mar 06/04/10					
Análisis	0,25 días	lun 05/04/10	lun 05/04/10					
Diseño	0,25 días	lun 05/04/10	lun 05/04/10					
Implementación	1 día	lun 05/04/10	mar 06/04/10					
Pruebas	2 días	lun 05/04/10	mar 06/04/10					
Despliegue	0,5 días	mar 06/04/10	mar 06/04/10					
Capacidad operacional	0 días	mar 06/04/10	mar 06/04/10					
Transición	8 días	mié 07/04/10	vie 16/04/10					

Recursos	Temporales	2 días		
	Humanos	1 persona		
	Software	Sistema operativo	Microsoft Windows 7	
		Ofimática	Microsoft Word 2010 Adobe Acrobat Professional 8	
		Diseño	Visual Paradigm 4.2	
Codificación		NetBeans 6.8		
Hardware	1 PC			

Casos de uso y escenarios implementados	Ninguno.
---	----------

Planificación de la iteración

3.2. Análisis de los casos de uso y escenarios

Ninguno.

4. Diseño

4.1. Visualización de componentes gráficos

Durante la fase de Construcción hemos implementado la aplicación de modo “general”, pudiendo ser utilizada casi toda la funcionalidad que proporciona el frame principal por todos los usuarios. Esto tiene sentido si de lo que tratamos es de poder desarrollar todo con la mayor comodidad posible, pero una vez que ya disponemos de todo codificado y probado, hay que diseñar una estrategia que nos permita mostrar u ocultar los componentes gráficos en función del tipo de usuario que utilice la aplicación.



Atendiendo a la especificación de los casos de uso y comprobando caso por caso qué actores tienen acceso a él, diseñamos la siguiente tabla que relaciona pestañas/menús y el acceso que posee los actores sobre ellos:

	Accesos	Emprs. y pers.	Tarjs., cols.,...	Zonas, llaves y llaveros	Préstos.	Cuentas	Partes de seg.	Notas	BD	Cuenta	
Admi.	Visualizar	Total					Vis. y elim.	Total			
Auto.	Ninguna	Total	Ninguna					Total	Ninguna	Total	
Vigi.	Total	Ninguna					Total	Ninguna		Total	
RP	Ninguna	Total ¹	Ninguna							Total	

Relación de pestañas y funcionalidad accedida según el tipo de usuarios

¹ Salvo la edición de los paneles de empresas y departamentos.

Par conseguir cumplir la tabla superior, deberemos:

ADMINISTRADOR	
Accesos	Hacer invisibles los dos paneles de botones: <ul style="list-style-type: none"> • JPAccesosB1 • JPAccesosB2
Partes de seguridad	Hacer invisible el botón: <ul style="list-style-type: none"> • JBanadirParte <p>No podrá modificar ningún parte debido a la estrategia de edición de campos desarrollada en su momento (iteración 'C9 - Partes de seguridad').</p>
AUTORIZADOR	
Accesos Tarjs., cols., nivs.,... Zonas, llaves y llaveros Préstamos Cuentas Partes de seguridad BD	Hacer invisibles las pestañas: <ul style="list-style-type: none"> • JPAccesos • JPTACs • JPZsLsLls • JPPrestamos • JPCuentas • JPPartes <p>Hacer invisible el menú:</p> <ul style="list-style-type: none"> • JMIBD <p>Las consultas a la base de datos han sido realizadas de modo que solo puedan visualizar aquellas empresas y departamentos de los que sean responsables.</p>

Relación de pestañas y componentes gráficos a ocultar según el tipo de usuario



VIGILANTE	
Empresas y personal Tarjs., cols., nivs.,... Zonas, llaves y llaveros Préstamos Cuentas BD	Hacer invisibles las pestañas: <ul style="list-style-type: none">• JPEmpresasPersonal• JPTACs• JPZsLsLls• JPPrestamos• JPCuentas Hacer invisible el menú: <ul style="list-style-type: none">• JMIBD
RP	
Accesos Tarjs., cols., nivs.,... Zonas, llaves y llaveros Préstamos Cuentas Partes de seguridad Notas BD	La misma que Autorizador pero sin acceso a 'Notas' y 'Empresas y personal restringido': <ul style="list-style-type: none">• JPAccesos• JPTACs• JPZsLsLls• JPPrestamos• JPCuentas• JPPartes• JPNotas Hacer invisible el menú: <ul style="list-style-type: none">• JMIBD Hacer invisibles los paneles de 'Empresas y personal': <ul style="list-style-type: none">• JPEmpresasB• JPDepartamentoB Las empresas y departamentos visualizados serán única y exclusivamente aquellos de los que ejerzan como responsables de personal.

Relación de pestañas y componentes gráficos a ocultar según el tipo de usuario



Lo único que debemos hacer es codificar un método que se ejecutará al final del constructor del frame principal (*Aplicacion*) y que se encargará de ocultar todos los componentes gráficos innecesarios (mencionados en la tabla anterior) según el tipo de usuario que acceda a la aplicación:

- Si el usuario es ADMINISTRADOR →
 - Ocultamos JPAccesosB1 y JPAccesosB2
 - Ocultamos JBAadirParte
- [] si el usuario es VIGILANTE →
 - Borramos las pestañas 1, 2, 3, 4 y 6
 - Ocultamos el menú JMIBD
- [] si el usuario es AUTORIZADOR o RP →
 - Si el usuario es RP
 - Borramos la última pestaña y los paneles de edición de empresas y departamentos.
 - Borramos las pestañas 0, 2, 3, 4, 5 y 6
 - Ocultamos el menú JMIBD

Esta sería la secuencia de inicialización de *Aplicación.java*:

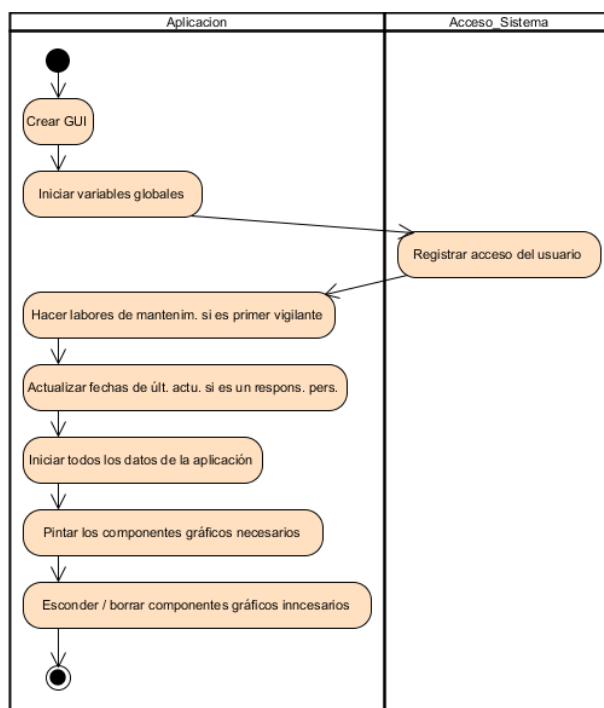


Diagrama de actividades

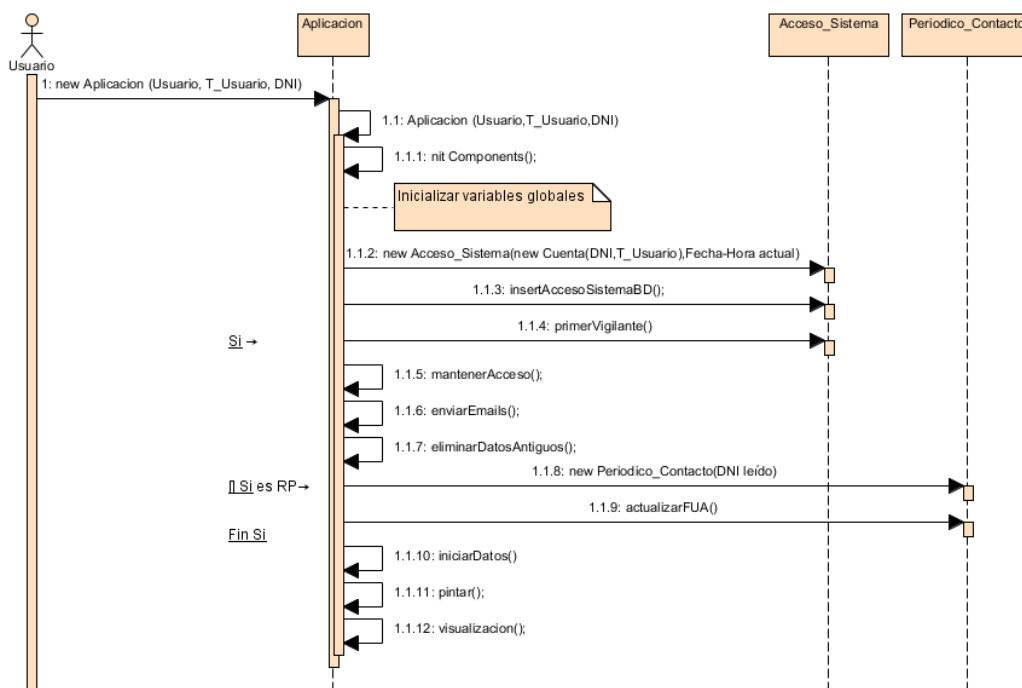


Diagrama de secuencia

4.2. Refresco de datos

El que accessManager sea una aplicación multiusuario y concurrente hace que los datos que alberga puedan variar de una forma bastante continuada. Para posibilitar que el usuario pueda disponer continuamente de la información que se encuentra manejando actualizada diseñamos una simple pero eficaz estrategia de actualización de datos. Al haber creado métodos específicos para la inicialización de cada pestaña en el constructor de Aplicación, podemos llamarlos cada vez que el contenido de una pestaña sea mostrado al usuario (evento *componentShown*).

4.3. Rediseño de la aplicación

Rediseñamos la interfaz gráfica de modo que todos los componentes sean visibles correctamente, estén alineados entre sí,... cambiamos letras por iconos,... obteniendo los resultados que mostramos en la siguiente sección.



5. Codificación e integración

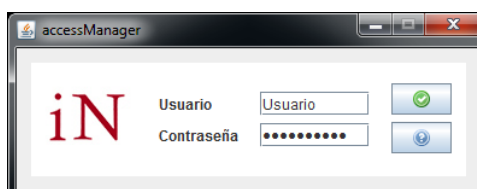
La codificación de la parte de configuración resulta bastante sencilla, ya que el único método a implementar es el descrito en la sección anterior y únicamente consta de una serie de ramas condicionales y operaciones de ocultación/borrado muy simples.

La competente al rediseño se realiza a través del diseñador gráfico y no merece la pena que hagamos comentarios al respecto. Lo que sí que ofrecemos a lo largo de las siguientes páginas es una breve comparación de los cambios estéticos realizado.

5.1. Identificación



Interfaz de logueo en el sistema - Antes



Interfaz de logueo en el sistema - Después



5.2. Accesos

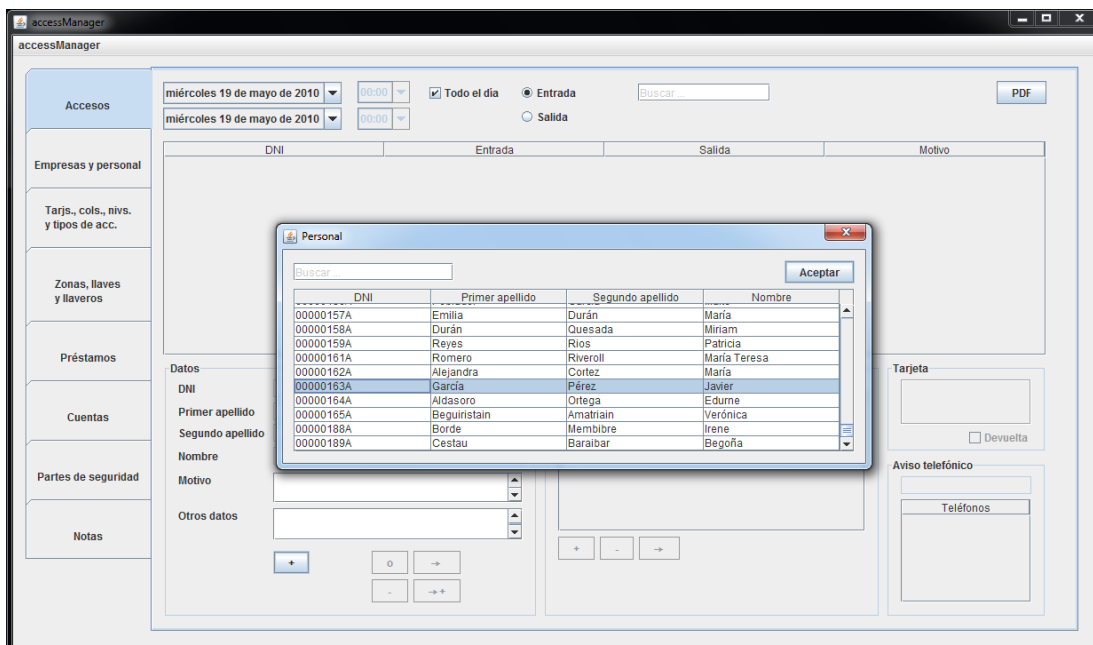
The screenshot shows the 'Accesos' tab in the accessManager application. The interface includes a sidebar with categories like 'Empresas y personal', 'Tarjetas, cols., nivs. y tipos de acc.', 'Zonas, llaves y llaveros', 'Préstamos', 'Cuentas', 'Partes de seguridad', and 'Notas'. The main area has a header with date and time selectors, a search box, and radio buttons for 'Entrada' and 'Salida'. Below this is a table with columns for 'DNI', 'Entrada', 'Salida', and 'Motivo'. The bottom section contains input fields for 'DNI', 'Primer apellido', 'Segundo apellido', 'Nombre', 'Motivo', and 'Otros datos', along with a 'Devuelta' checkbox and a 'Teléfonos' field.

Interfaz de la pestaña 'Accesos' - Antes

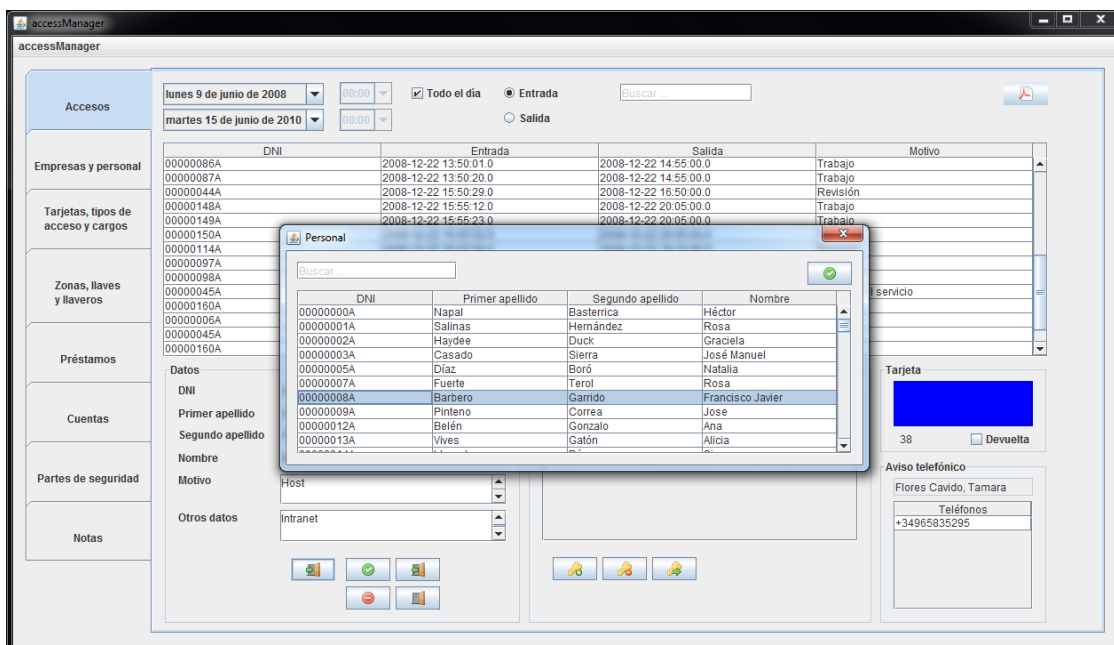
The screenshot shows the 'Accesos' tab after data entry. The table is populated with records. The input fields are filled with data for a specific user: DNI 0000097A, Alonzo Miguel, Host, and Intranet. The 'Llaves y llaveros' table shows three entries for 'Llave' with codes 17, 18, and 19, all marked 'No' for 'Devuelta'. The 'Tarjeta' field shows ID 38, and the 'Aviso telefónico' field shows 'Flores Cavido, Tamara' and phone number '+34965835295'.

DNI	Entrada	Salida	Motivo
0000098A	2008-12-22 13:50:01.0	2008-12-22 14:55:00.0	Trabajo
0000097A	2008-12-22 13:50:20.0	2008-12-22 14:55:00.0	Trabajo
0000044A	2008-12-22 15:50:29.0	2008-12-22 16:50:00.0	Revisión
00000148A	2008-12-22 15:55:12.0	2008-12-22 20:05:00.0	Trabajo
00000148A	2008-12-22 15:55:23.0	2008-12-22 20:05:00.0	Trabajo
00000150A	2008-12-22 15:55:52.0	2008-12-22 20:05:00.0	Trabajo
00000114A	2008-12-22 16:02:00.0	2008-12-22 19:10:00.0	Revisión
0000097A	2008-12-22 18:00:00.0		Host
0000098A	2008-12-22 18:05:00.0	2008-12-22 20:15:00.0	Host
0000045A	2008-12-22 20:00:00.0	2008-12-22 20:20:00.0	Aprender el servicio
00000160A	2008-12-24 00:20:01.0	2008-12-25 02:40:00.0	Trabajo
0000006A	2009-12-22 13:12:05.0		Trabajo
0000045A	2009-12-22 20:00:00.0		Trabajo
00000160A	2010-04-15 00:20:55.0		Trabajo

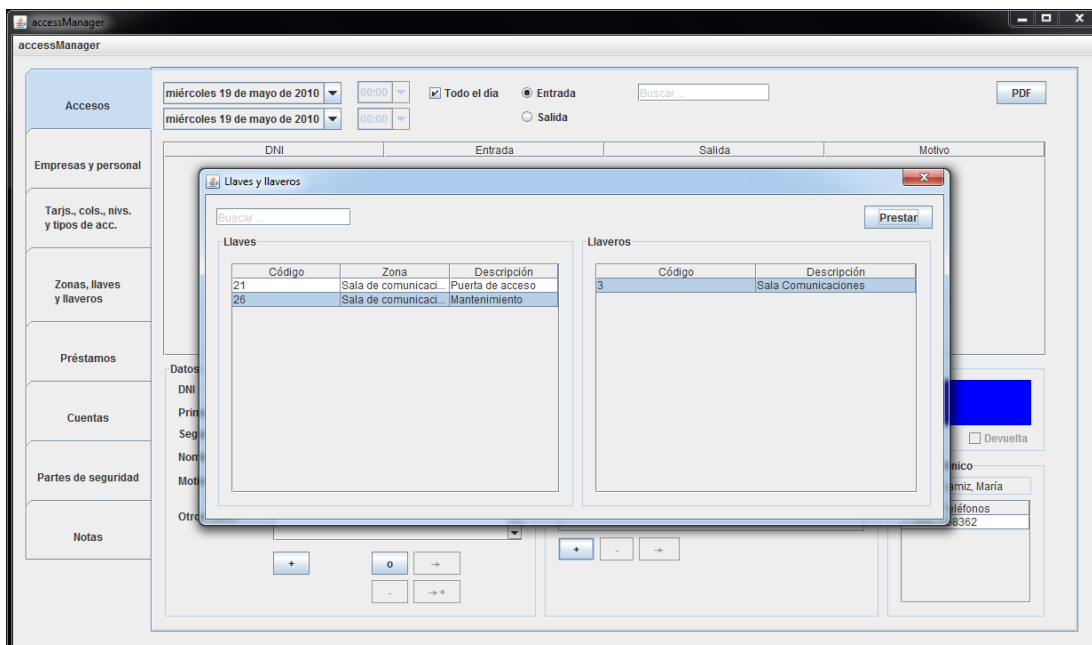
Interfaz de la pestaña 'Accesos' - Después



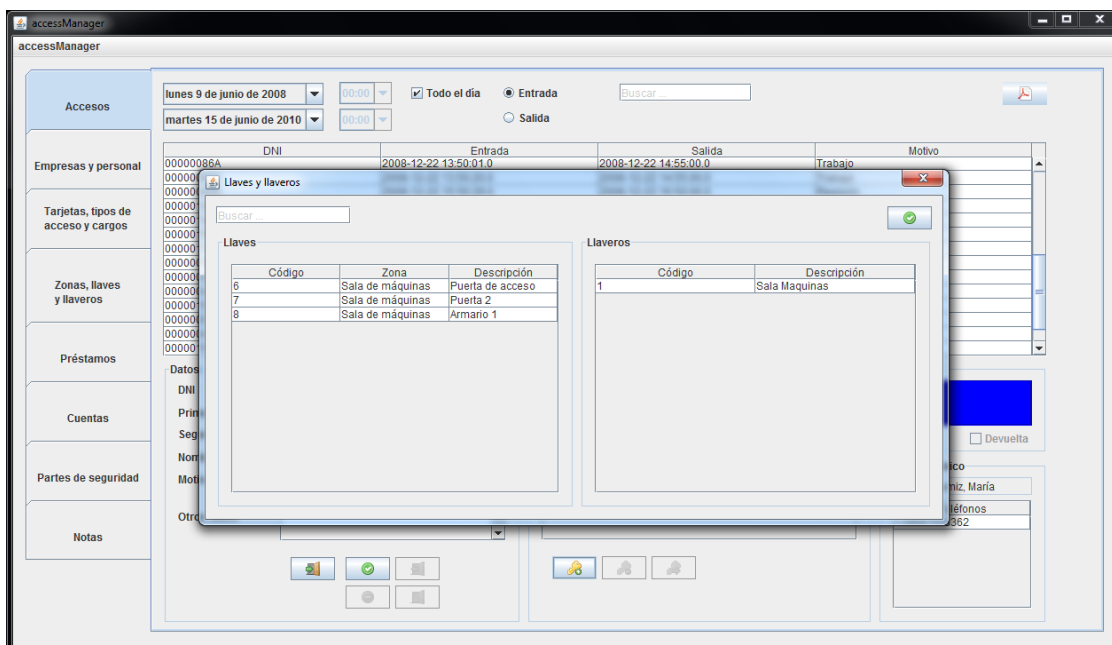
Interfaz de selección de persona - Antes



Interfaz de selección de persona - Después



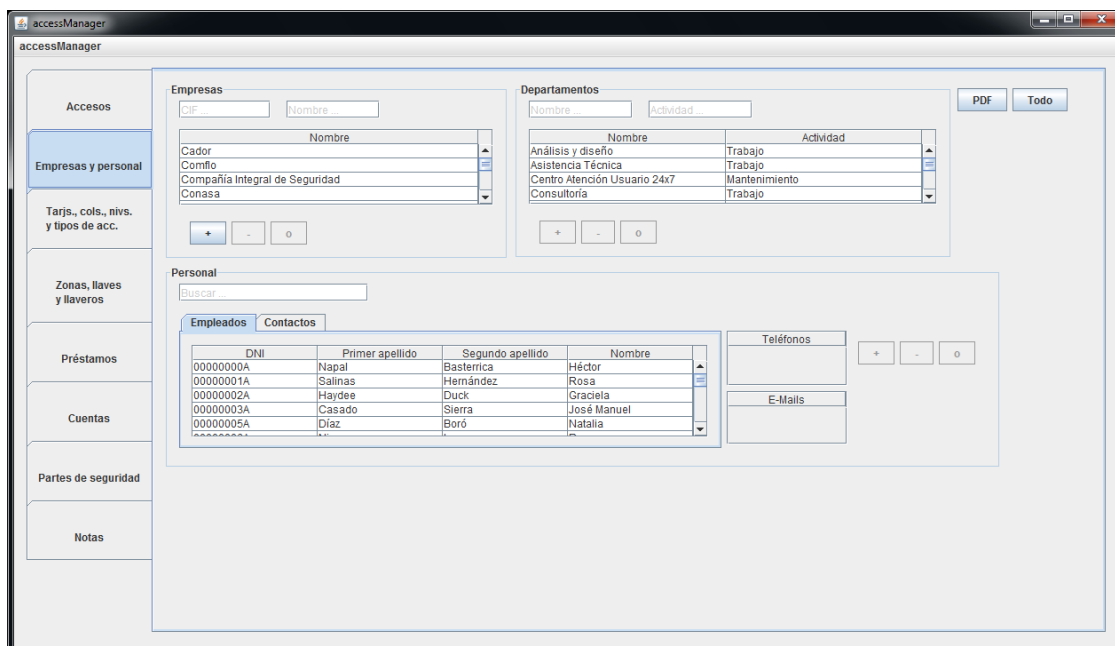
Interfaz de selección de préstamos - Antes



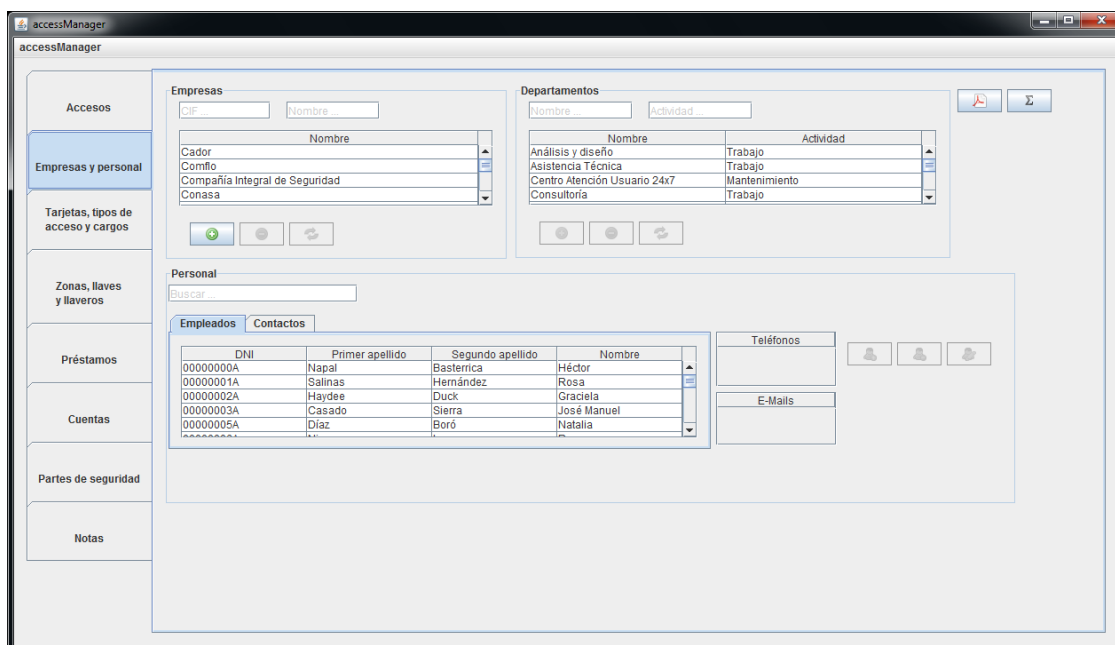
Interfaz de selección de préstamos - Después



5.3. Empresas y personal



Interfaz de la pestaña 'Empresas y personal' - Antes



Interfaz de la pestaña 'Empresas y personal' - Después



Empresas y personal

Empresas	Departamentos
Color	Asistencia Técnica
Comfy	Centro de Atención al Usuario 24x7
Compañía Integral de Seguridad	Emergencias 24x7
Conecta	Mantenimiento 24x7

Personal

Empleado	Contacto
00000074	Fuente
00000124	Rodríguez
00000124	Almeida
00000124	Benito
00000124	Palencia

Empresas y personal

Empresas	Departamentos
Color	Asistencia Técnica
Comfy	Centro de Atención al Usuario 24x7
Compañía Integral de Seguridad	Emergencias 24x7
Conecta	Mantenimiento 24x7

Personal

Empleado	Contacto
00000074	Fuente
00000124	Rodríguez
00000124	Almeida
00000124	Benito
00000124	Palencia

Empresas y personal

Empresas	Departamentos
Color	Asistencia Técnica
Comfy	Centro de Atención al Usuario 24x7
Compañía Integral de Seguridad	Emergencias 24x7
Conecta	Mantenimiento 24x7

Personal

Empleado	Contacto
00000074	Fuente
00000124	Rodríguez
00000124	Almeida
00000124	Benito
00000124	Palencia

Empresas y personal

Empresas	Departamentos
Color	Asistencia Técnica
Comfy	Centro de Atención al Usuario 24x7
Compañía Integral de Seguridad	Emergencias 24x7
Conecta	Mantenimiento 24x7

Personal

Empleado	Contacto
00000074	Fuente
00000124	Rodríguez
00000124	Almeida
00000124	Benito
00000124	Palencia

Empresas y personal

Empresas	Departamentos
Color	Asistencia Técnica
Comfy	Centro de Atención al Usuario 24x7
Compañía Integral de Seguridad	Emergencias 24x7
Conecta	Mantenimiento 24x7

Personal

Empleado	Contacto
00000074	Fuente
00000124	Rodríguez
00000124	Almeida
00000124	Benito
00000124	Palencia

Empresas y personal

Empresas	Departamentos
Color	Asistencia Técnica
Comfy	Centro de Atención al Usuario 24x7
Compañía Integral de Seguridad	Emergencias 24x7
Conecta	Mantenimiento 24x7

Personal

Empleado	Contacto
00000074	Fuente
00000124	Rodríguez
00000124	Almeida
00000124	Benito
00000124	Palencia

Empresas y personal

Empresas	Departamentos
Color	Asistencia Técnica
Comfy	Centro de Atención al Usuario 24x7
Compañía Integral de Seguridad	Emergencias 24x7
Conecta	Mantenimiento 24x7

Personal

Empleado	Contacto
00000074	Fuente
00000124	Rodríguez
00000124	Almeida
00000124	Benito
00000124	Palencia

Empresas y personal

Empresas	Departamentos
Color	Asistencia Técnica
Comfy	Centro de Atención al Usuario 24x7
Compañía Integral de Seguridad	Emergencias 24x7
Conecta	Mantenimiento 24x7

Personal

Empleado	Contacto
00000074	Fuente
00000124	Rodríguez
00000124	Almeida
00000124	Benito
00000124	Palencia

Interfaces parciales de la pestaña 'Empresas y personal' - Antes



accessManager

Accesos

Empresas y personal

Empresas y personal

Tarjetas, tipo de acceso y cargos

Zonas, áreas y Baños

Personal

Partes de seguridad

Notas

DNI	Primer apellido	Segundo apellido	Nombre	Actividad
00000000	Padre	Padre	Padre	
00000001	Padre	Padre	Padre	
00000002	Padre	Padre	Padre	
00000003	Padre	Padre	Padre	
00000004	Padre	Padre	Padre	
00000005	Padre	Padre	Padre	
00000006	Padre	Padre	Padre	
00000007	Padre	Padre	Padre	
00000008	Padre	Padre	Padre	
00000009	Padre	Padre	Padre	
00000010	Padre	Padre	Padre	

accessManager

Accesos

Empresas y personal

Empresas y personal

Tarjetas, tipo de acceso y cargos

Zonas, áreas y Baños

Personal

Partes de seguridad

Notas

DNI	Primer apellido	Segundo apellido	Nombre	Actividad
00000000	Padre	Padre	Padre	
00000001	Padre	Padre	Padre	
00000002	Padre	Padre	Padre	
00000003	Padre	Padre	Padre	
00000004	Padre	Padre	Padre	
00000005	Padre	Padre	Padre	
00000006	Padre	Padre	Padre	
00000007	Padre	Padre	Padre	
00000008	Padre	Padre	Padre	
00000009	Padre	Padre	Padre	
00000010	Padre	Padre	Padre	

accessManager

Accesos

Empresas y personal

Empresas y personal

Tarjetas, tipo de acceso y cargos

Zonas, áreas y Baños

Personal

Partes de seguridad

Notas

DNI	Primer apellido	Segundo apellido	Nombre	Actividad
00000000	Padre	Padre	Padre	
00000001	Padre	Padre	Padre	
00000002	Padre	Padre	Padre	
00000003	Padre	Padre	Padre	
00000004	Padre	Padre	Padre	
00000005	Padre	Padre	Padre	
00000006	Padre	Padre	Padre	
00000007	Padre	Padre	Padre	
00000008	Padre	Padre	Padre	
00000009	Padre	Padre	Padre	
00000010	Padre	Padre	Padre	

accessManager

Accesos

Empresas y personal

Empresas y personal

Tarjetas, tipo de acceso y cargos

Zonas, áreas y Baños

Personal

Partes de seguridad

Notas

DNI	Primer apellido	Segundo apellido	Nombre	Actividad
00000000	Padre	Padre	Padre	
00000001	Padre	Padre	Padre	
00000002	Padre	Padre	Padre	
00000003	Padre	Padre	Padre	
00000004	Padre	Padre	Padre	
00000005	Padre	Padre	Padre	
00000006	Padre	Padre	Padre	
00000007	Padre	Padre	Padre	
00000008	Padre	Padre	Padre	
00000009	Padre	Padre	Padre	
00000010	Padre	Padre	Padre	

accessManager

Accesos

Empresas y personal

Empresas y personal

Tarjetas, tipo de acceso y cargos

Zonas, áreas y Baños

Personal

Partes de seguridad

Notas

DNI	Primer apellido	Segundo apellido	Nombre	Actividad
00000000	Padre	Padre	Padre	
00000001	Padre	Padre	Padre	
00000002	Padre	Padre	Padre	
00000003	Padre	Padre	Padre	
00000004	Padre	Padre	Padre	
00000005	Padre	Padre	Padre	
00000006	Padre	Padre	Padre	
00000007	Padre	Padre	Padre	
00000008	Padre	Padre	Padre	
00000009	Padre	Padre	Padre	
00000010	Padre	Padre	Padre	

accessManager

Accesos

Empresas y personal

Empresas y personal

Tarjetas, tipo de acceso y cargos

Zonas, áreas y Baños

Personal

Partes de seguridad

Notas

DNI	Primer apellido	Segundo apellido	Nombre	Actividad
00000000	Padre	Padre	Padre	
00000001	Padre	Padre	Padre	
00000002	Padre	Padre	Padre	
00000003	Padre	Padre	Padre	
00000004	Padre	Padre	Padre	
00000005	Padre	Padre	Padre	
00000006	Padre	Padre	Padre	
00000007	Padre	Padre	Padre	
00000008	Padre	Padre	Padre	
00000009	Padre	Padre	Padre	
00000010	Padre	Padre	Padre	

accessManager

Accesos

Empresas y personal

Empresas y personal

Tarjetas, tipo de acceso y cargos

Zonas, áreas y Baños

Personal

Partes de seguridad

Notas

DNI	Primer apellido	Segundo apellido	Nombre	Actividad
00000000	Padre	Padre	Padre	
00000001	Padre	Padre	Padre	
00000002	Padre	Padre	Padre	
00000003	Padre	Padre	Padre	
00000004	Padre	Padre	Padre	
00000005	Padre	Padre	Padre	
00000006	Padre	Padre	Padre	
00000007	Padre	Padre	Padre	
00000008	Padre	Padre	Padre	
00000009	Padre	Padre	Padre	
00000010	Padre	Padre	Padre	

accessManager

Accesos

Empresas y personal

Empresas y personal

Tarjetas, tipo de acceso y cargos

Zonas, áreas y Baños

Personal

Partes de seguridad

Notas

DNI	Primer apellido	Segundo apellido	Nombre	Actividad
00000000	Padre	Padre	Padre	
00000001	Padre	Padre	Padre	
00000002	Padre	Padre	Padre	
00000003	Padre	Padre	Padre	
00000004	Padre	Padre	Padre	
00000005	Padre	Padre	Padre	
00000006	Padre	Padre	Padre	
00000007	Padre	Padre	Padre	
00000008	Padre	Padre	Padre	
00000009	Padre	Padre	Padre	
00000010	Padre	Padre	Padre	

Interfaces parciales de la pestaña 'Empresas y personal' - Después



5.4. Tarjetas, tipos de acceso y cargos

The screenshot shows the 'accessManager' application window with the 'Tarjetas, tipos de acceso y cargos' tab selected. The interface is divided into several sections:

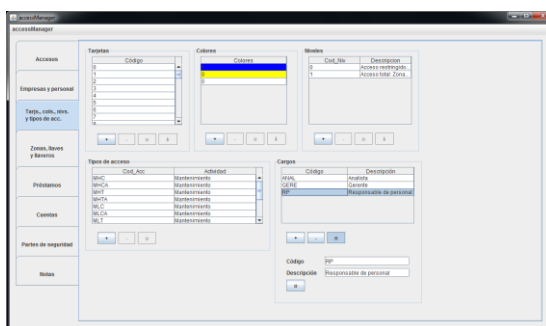
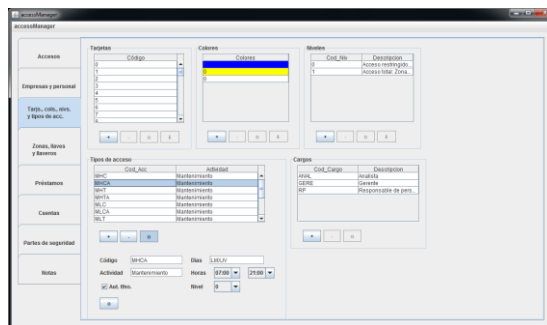
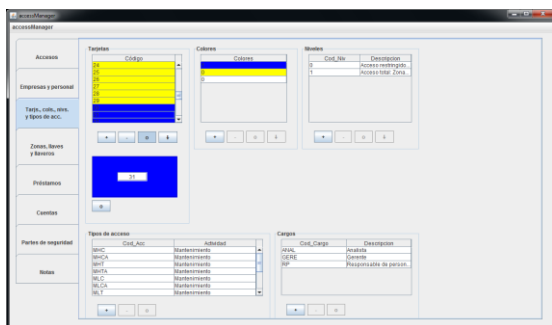
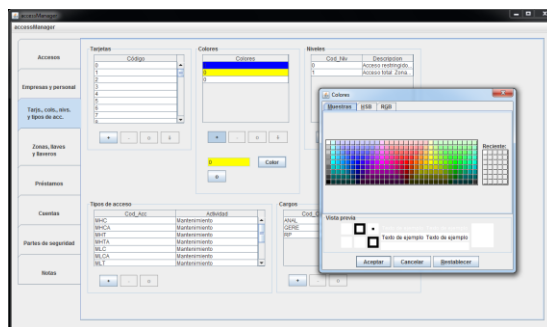
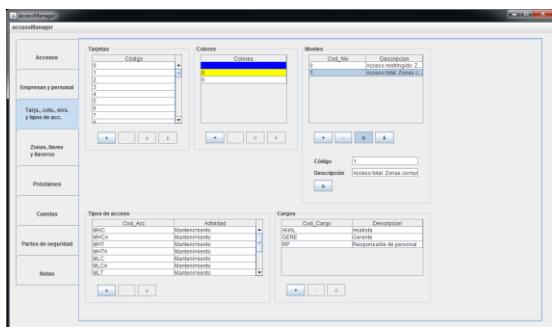
- Tarjetas:** A list of card codes from 1 to 9.
- Colores:** A list of colors, with '0' highlighted in yellow.
- Niveles:** A table with columns 'Cod_Niv' and 'Descripción'. It contains two rows: '0' with 'Acceso restringido...' and '1' with 'Acceso total. Zona...'. Below the table are control buttons: '+', '-', '0', and a downward arrow.
- Tipos de acceso:** A table with columns 'Cod_Acc' and 'Actividad'. It lists several maintenance codes (MHC, MHCA, MHT, MHTA, MLC, MLCA, MLT) all with 'Mantenimiento' as the activity. Below the table are control buttons: '+', '-', and '0'.
- Cargos:** A table with columns 'Cod_Cargo' and 'Descripción'. It lists 'ANAL' (Analista), 'GERE' (Gerente), and 'RP' (Responsable de person...). Below the table are control buttons: '+', '-', and '0'.

Interfaz de la pestaña 'Tarjetas, tipos de acceso y cargos' - Antes

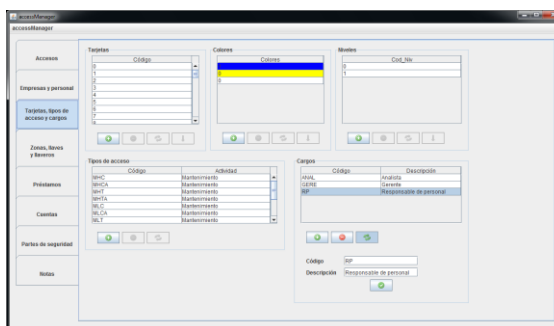
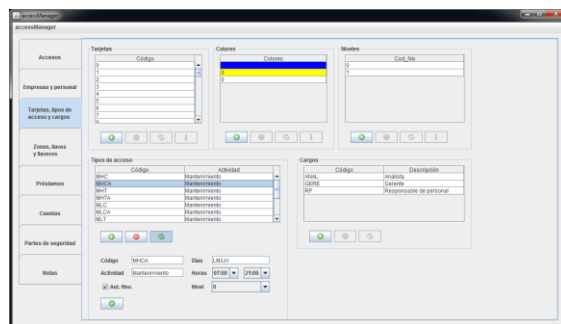
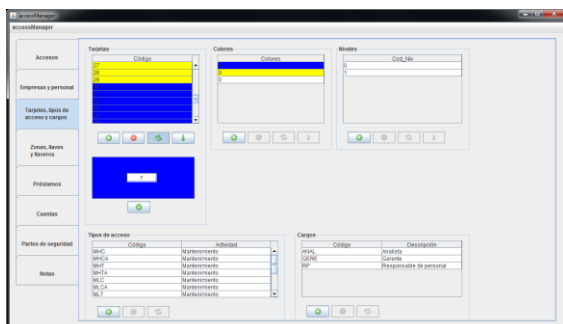
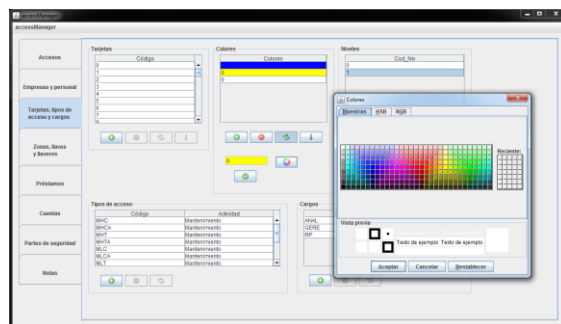
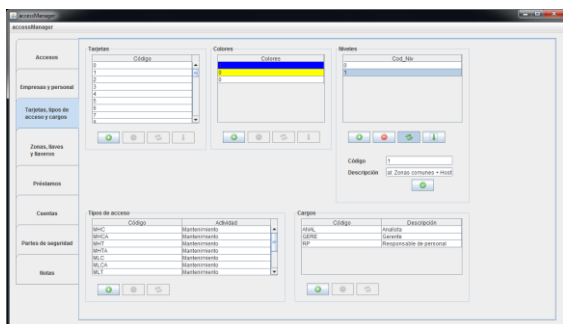
The screenshot shows the 'accessManager' application window with the 'Tarjetas, tipos de acceso y cargos' tab selected, showing changes from the previous state:

- Tarjetas:** The list now includes '0' at the top, followed by '1' through '8'.
- Colores:** The list now includes '0' at the top, followed by '0'.
- Niveles:** The table now includes '0' with 'Cod_Niv' and '1' with 'Cod_Niv'. The 'Descripción' column is empty. Below the table are control buttons: '+', a circle, a refresh icon, and a downward arrow.
- Tipos de acceso:** The table now includes '0' at the top, followed by 'MHC' through 'MLT'. Below the table are control buttons: '+', a circle, and a refresh icon.
- Cargos:** The table now includes 'ANAL', 'GERE', and 'RP'. Below the table are control buttons: '+', a circle, and a refresh icon.

Interfaz de la pestaña 'Tarjetas, tipos de acceso y cargos' - Después



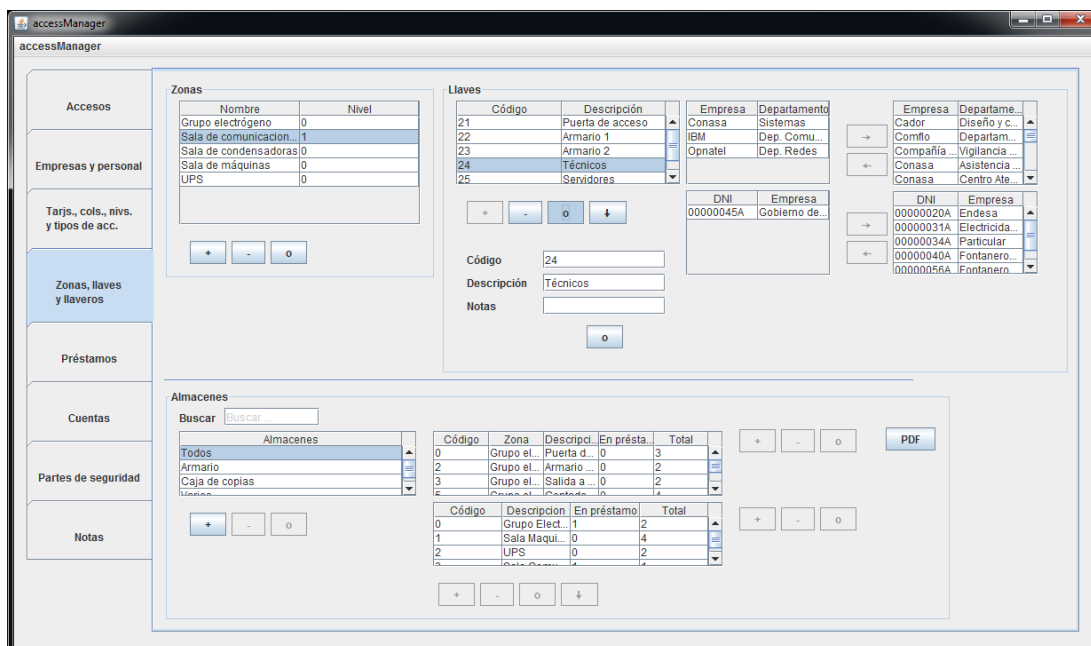
Interfaces parciales de la pestaña 'Tarjetas, tipos de acceso y cargos' - Antes



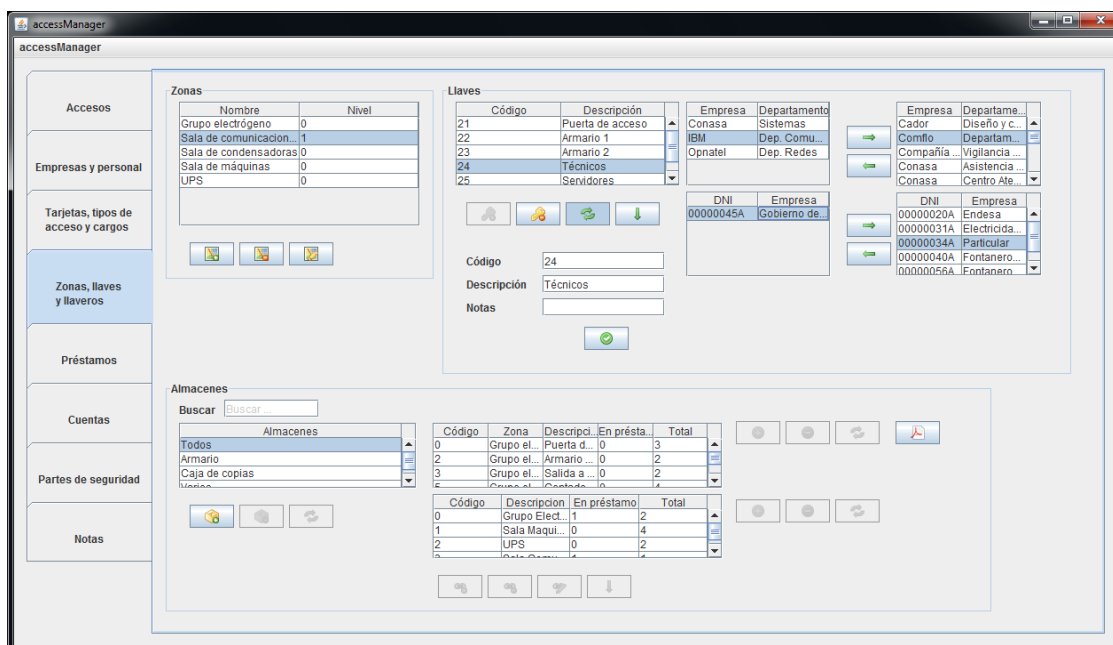
Interfaces parciales de la pestaña 'Tarjetas, tipos de acceso y cargos' - Después



5.5. Zonas, llaves y llaveros



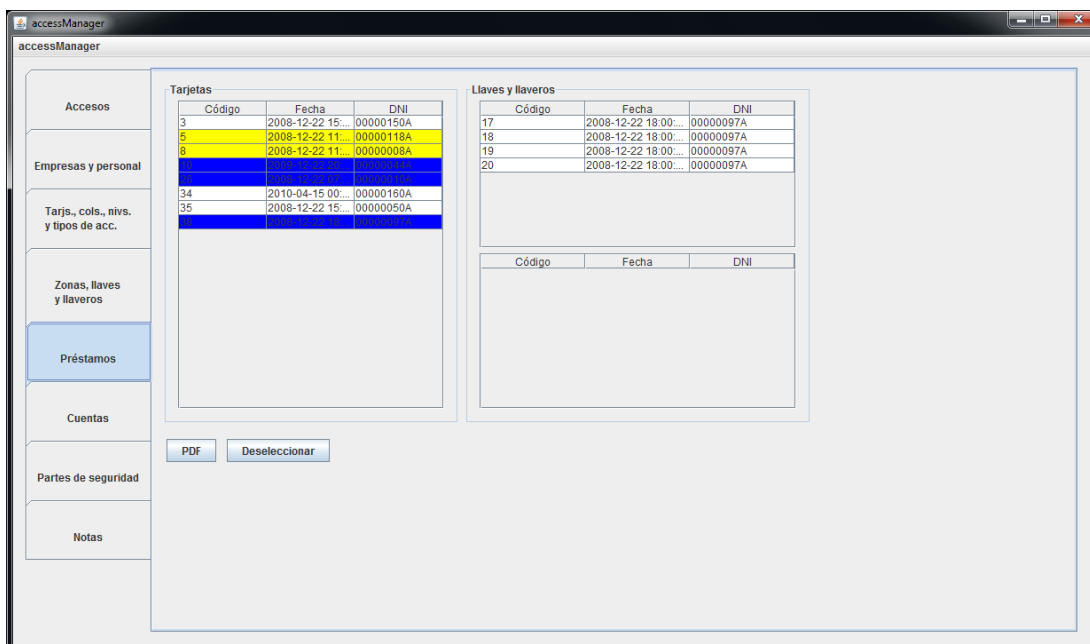
Interfaz de la pestaña 'Zonas, llaves y llaveros' - Antes



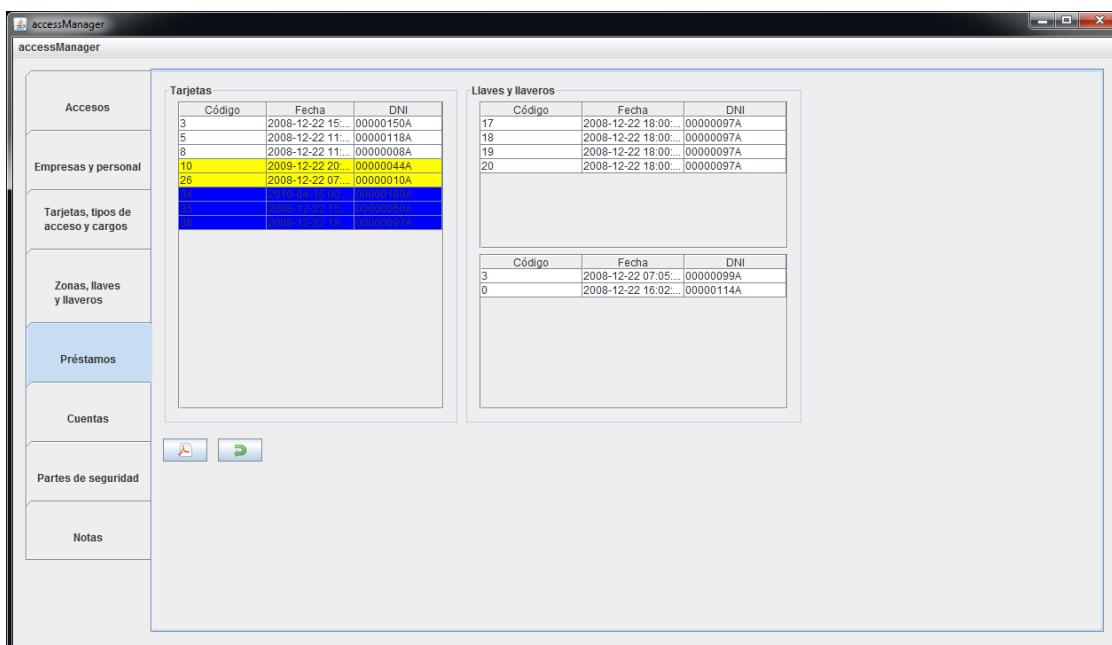
Interfaz de la pestaña 'Zonas, llaves y llaveros' - Después



5.6. Préstamos



Interfaz de la pestaña 'Préstamos' - Antes



Interfaz de la pestaña 'Préstamos' - Después



accessManager

accessManager

Accesos

Empresas y personal

Tarjetas, cols., nivs. y tipos de acc.

Zonas, llaves y llaveros

Préstamos

Cuentas

Partes de seguridad

Notas

Tarjetas

Código	Fecha	DNI
3	2008-12-22 15:...	00000150A
5	2008-12-22 11:...	00000118A
8	2008-12-22 11:...	00000008A
10	2009-12-22 20:...	00000044A
26	2008-12-22 07:...	00000010A
34	2010-04-15 00:...	00000160A
35	2008-12-22 15:...	00000050A
37	2008-12-22 15:...	00000070A

Llaves y llaveros

Código	Fecha	DNI
17	2008-12-22 18:00:...	00000097A
18	2008-12-22 18:00:...	00000097A
19	2008-12-22 18:00:...	00000097A
20	2008-12-22 18:00:...	00000097A

Tfno

+34621872681

PDF Deseleccionar

Enviar recordatorio

Interfaz expandida de la pestaña 'Préstamos' - Antes

accessManager

accessManager

Accesos

Empresas y personal

Tarjetas, tipos de acceso y cargos

Zonas, llaves y llaveros

Préstamos

Cuentas

Partes de seguridad

Notas

Tarjetas

Código	Fecha	DNI
3	2008-12-22 15:...	00000150A
5	2008-12-22 11:...	00000118A
8	2008-12-22 11:...	00000008A
10	2009-12-22 20:...	00000044A
26	2008-12-22 07:...	00000010A
34	2010-04-15 00:...	00000160A
35	2008-12-22 15:...	00000050A
37	2008-12-22 15:...	00000070A

Llaves y llaveros

Código	Fecha	DNI
17	2008-12-22 18:00:...	00000097A
18	2008-12-22 18:00:...	00000097A
19	2008-12-22 18:00:...	00000097A
20	2008-12-22 18:00:...	00000097A

Teléfonos

+34621872681

Tablas a reportar:

Todas

Tarjetas

Llaves

Llaveros

Enviar recordatorio

Interfaz expandida de la pestaña 'Préstamos' - Después



5.7. Cuentas

The screenshot shows the 'accessManager' application window with the 'Cuentas' tab selected. The interface is divided into several sections:

- Accesos:** A sidebar menu on the left.
- Empresas y personal:** A section containing a table of user accounts.
- Tarjetas, cols., nivs. y tipos de acc.:** A section containing a table of user types.
- Zonas, llaves y llaveros:** A section with navigation buttons.
- Préstamos:** A section with navigation buttons.
- Cuentas:** The active section, containing a search bar, date filters, and a table of system access history.
- Partes de seguridad:** A section with navigation buttons.
- Notas:** A section with navigation buttons.

The 'Cuentas' table (before changes) is as follows:

DNI	Tipo	Usuario	Contraseña
00000000A	AUTORIZADOR	a	*****
00000000A	ADMINISTRADOR	hnapalb	*****
00000001A	VIGILANTE	rsalinash	*****
00000002A	VIGILANTE	ghaydeed	*****
00000003A	VIGILANTE	jcasados	*****
00000018A	AUTORIZADOR	mpgamiz	*****
00000028A	AUTORIZADOR	lcarmona	*****
00000043A	AUTORIZADOR	tcavido	*****
00000163A	RP	jgarciap	*****
00000168A	RP	edominicauze	*****

The 'Tipos de usuario' table (before changes) is as follows:

Tipo	Descripcion
ADMINISTRADOR	Administrador del sistema
AUTORIZADOR	Autorizador de departamentos
RP	Responsable de personal de ...
VIGILANTE	Vigilante del centro

The 'Historial de accesos al sistema' table (before changes) is as follows:

Usuario	Tipo	DNI	Fecha
a	AUTORIZADOR	00000000A	2010-05-24 08:47:04.0
a	AUTORIZADOR	00000000A	2010-05-24 08:47:31.0
a	AUTORIZADOR	00000000A	2010-05-24 09:03:30.0
a	AUTORIZADOR	00000000A	2010-05-24 09:17:05.0
a	AUTORIZADOR	00000000A	2010-05-24 09:57:41.0
a	AUTORIZADOR	00000000A	2010-05-24 10:02:26.0
a	AUTORIZADOR	00000000A	2010-05-24 10:02:58.0
a	AUTORIZADOR	00000000A	2010-05-24 10:15:59.0

Interfaz de la pestaña 'Cuentas' - Antes

The screenshot shows the 'accessManager' application window with the 'Cuentas' tab selected, after changes have been made. The interface is similar to the previous screenshot but with updated data:

- Empresas y personal:** The table now includes a user 'hnapalb' with type 'ADMINISTRADOR' and DNI '00000000A'.
- Tipos de usuario:** The table now includes a user type 'ADMINISTRADOR' with description 'Administrador del sistema'.
- Historial de accesos al sistema:** The table now shows access history for 'hnapalb' as 'ADMINISTRADOR' with DNI '00000000A' on '2010-06-15'.

The 'Cuentas' table (after changes) is as follows:

DNI	Tipo	Usuario	Contraseña
00000000A	AUTORIZADOR	a	*****
00000000A	ADMINISTRADOR	hnapalb	*****
00000001A	VIGILANTE	rsalinash	*****
00000002A	VIGILANTE	ghaydeed	*****
00000003A	VIGILANTE	jcasados	*****
00000018A	AUTORIZADOR	mpgamiz	*****
00000028A	AUTORIZADOR	lcarmona	*****
00000043A	AUTORIZADOR	tcavido	*****
00000163A	RP	jgarciap	*****
00000168A	RP	edominicauze	*****

The 'Tipos de usuario' table (after changes) is as follows:

Tipo	Descripcion
ADMINISTRADOR	Administrador del sistema
AUTORIZADOR	Autorizador de departamentos
RP	Responsable de personal de de...
VIGILANTE	Vigilante del centro

The 'Historial de accesos al sistema' table (after changes) is as follows:

Usuario	Tipo	DNI	Fecha
hnapalb	ADMINISTRADOR	00000000A	2010-06-15 10:12:12.0
hnapalb	ADMINISTRADOR	00000000A	2010-06-15 11:42:45.0
hnapalb	ADMINISTRADOR	00000000A	2010-06-15 11:43:37.0
hnapalb	ADMINISTRADOR	00000000A	2010-06-15 11:44:13.0
hnapalb	ADMINISTRADOR	00000000A	2010-06-15 11:45:21.0
hnapalb	ADMINISTRADOR	00000000A	2010-06-15 11:49:13.0
hnapalb	ADMINISTRADOR	00000000A	2010-06-15 11:50:26.0
hnapalb	ADMINISTRADOR	00000000A	2010-06-15 11:51:32.0

Interfaz de la pestaña 'Cuentas' - Después



Cuenta

Usuario y contraseña

a o

• o

Autenticación

Contraseña actual

Nuevos datos

Contraseña

Reescribe la contraseña

o

Interfaz del diálogo 'Cuenta' - Antes

Cuenta

Usuario y contraseña

hnapalb o

•••••• o

Autenticación

Contraseña actual

Nuevos datos

Contraseña

Reescribe la contraseña

o

Interfaz del diálogo 'Cuenta' - Después



5.8. Partes de seguridad

The screenshot shows the 'Partes de seguridad' tab in the accessManager application. The left sidebar contains a navigation menu with items: Accesos, Empresas y personal, Tarjs., cols., nivs. y tipos de acc., Zonas, llaves y llaveros, Préstamos, Cuentas, Partes de seguridad (highlighted), and Notas. The main content area is divided into two sections. The top section contains a 'Cod. Vig.' dropdown set to 'Todos', a checkbox for 'Activar fechas' (checked), and a 'Todo el día' checkbox (unchecked). Below these are two date-time pickers: 'sábado 23 de enero de 2010' at '18:30' and 'viernes 16 de abril de 2010' at '15:30'. A table lists dates and times, with '2010-04-16 19:00:00.0' selected. The bottom section contains '+', '-', and '0' buttons. The right section features the 'CIS' logo, a 'PDF' button, a 'Código de vigilante' field with '1', and a 'Fecha' field with 'viernes 16 de abril de 2010' and '19:00'. The 'Incidencias' list contains: '19:00 - Inicio servicio s/n aparente.', '20:00 - Sonda ambiente 20,4°.', '22:00 - Ronda por el centro s/n.', and '07:00 - Finalizo servicio s/n'.

Interfaz de la pestaña 'Partes de seguridad' - Antes

The screenshot shows the 'Partes de seguridad' tab in the accessManager application after changes. The left sidebar is identical to the previous screenshot. The main content area has been updated. The top section now has 'Activar fechas' unchecked and 'Todo el día' checked. The date-time pickers are now 'martes 15 de junio de 2010' at '00:00' and 'martes 15 de junio de 2010' at '00:00'. The table lists dates and times, with '2010-04-16 19:00:00.0' selected. The bottom section contains 'i', 'i', and a circle button. The right section features the 'CIS' logo, a print icon, a 'Código de vigilante' field with '1', and a 'Fecha' field with 'viernes 16 de abril de 2010' and '19:00'. The 'Incidencias' list is identical to the previous screenshot.

Interfaz de la pestaña 'Partes de seguridad' - Después



5.9. Notas

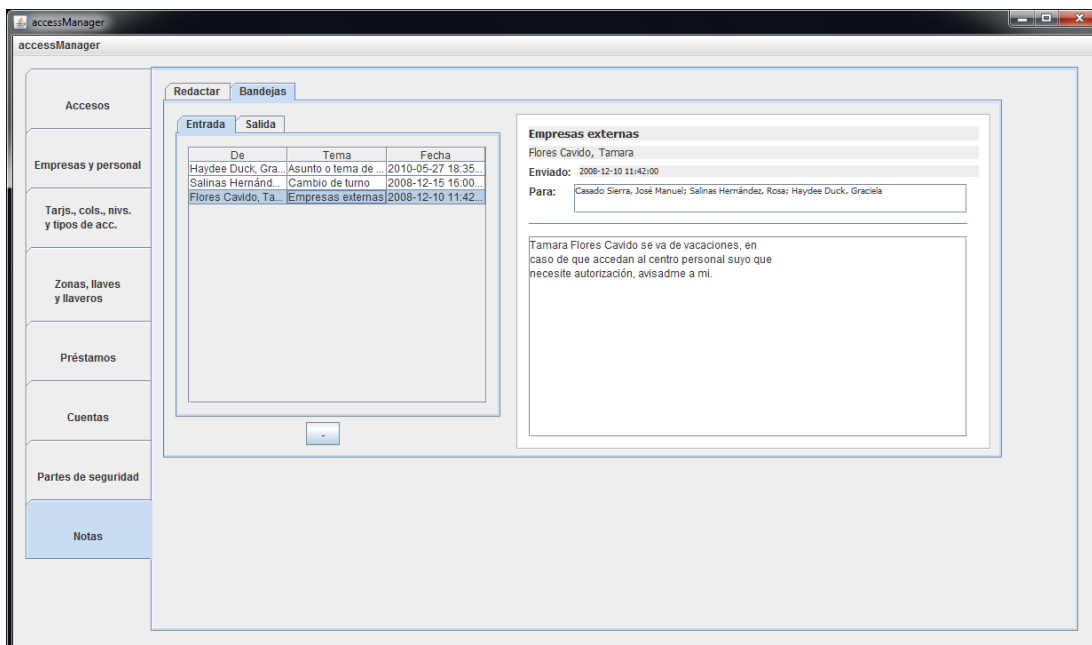
The screenshot shows the 'Notas' (Notes) tab in the 'accessManager' application. The interface is divided into a left sidebar with navigation options and a main content area. The 'Notas' option is selected in the sidebar. The main content area has two tabs: 'Redactar' (Compose) and 'Bandejas' (Inboxes). The 'Redactar' tab is active, showing a form with an 'Asunto' (Subject) field containing 'Asunto o tema de la nota' and a large 'Texto de la nota' (Note text) area. To the right, there is a 'Contactos' (Contacts) section with a search box and an 'Enviar' (Send) button. Below the search box is a table of contacts.

Primer apellido	Segundo ape...	Nombre
Casado	Sierra	José Manuel
Flores	Cavido	Tamara
Haydee	Duck	Graciela
Napal	Basterrica	Héctor
Salinas	Hernández	Rosa
Teresa	Gamiz	María
Trillo	Carmona	Laura

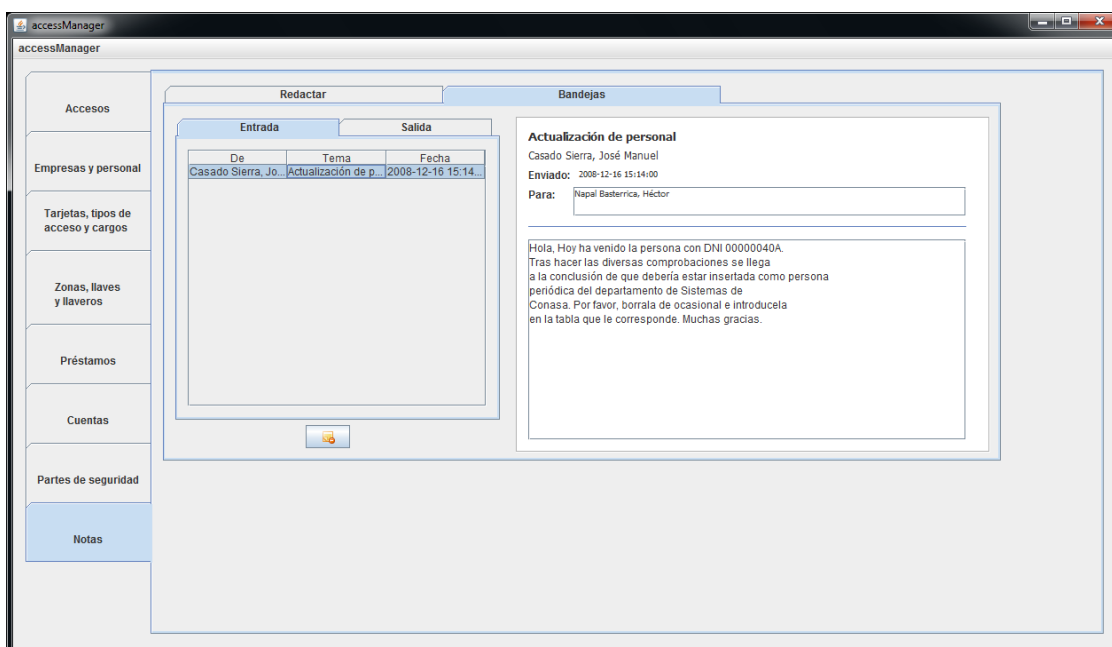
Interfaz de la pestaña 'Notas': Redactar - Antes

This screenshot is identical to the previous one, showing the 'Notas' tab in 'accessManager' before the 'Enviar' button is clicked. The 'Redactar' tab is active, and the 'Enviar' button is visible in the 'Contactos' section.

Interfaz de la pestaña 'Notas': Redactar - Después



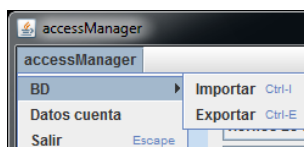
Interfaz de la pestaña 'Notas': Bandejas - Antes



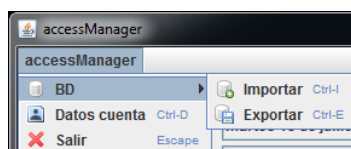
Interfaz de la pestaña 'Notas': Bandejas - Después



5.10. Menú 'accessManager'



Interfaz del menú 'accessManager' - Antes



Interfaz del menú 'accessManager' - Después

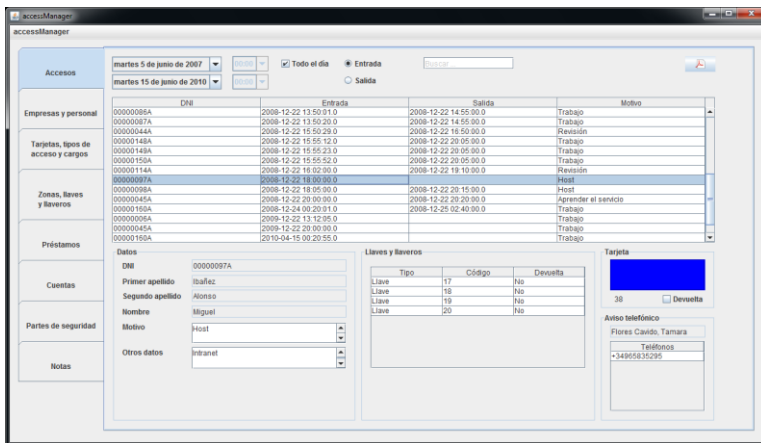


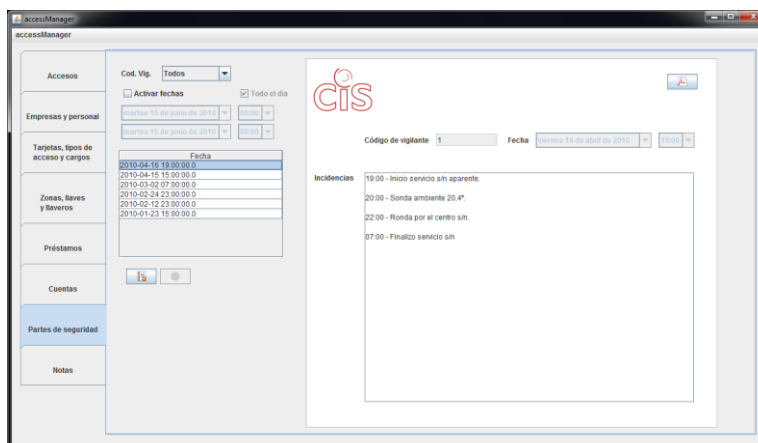
6. Pruebas

6.1. De unidad

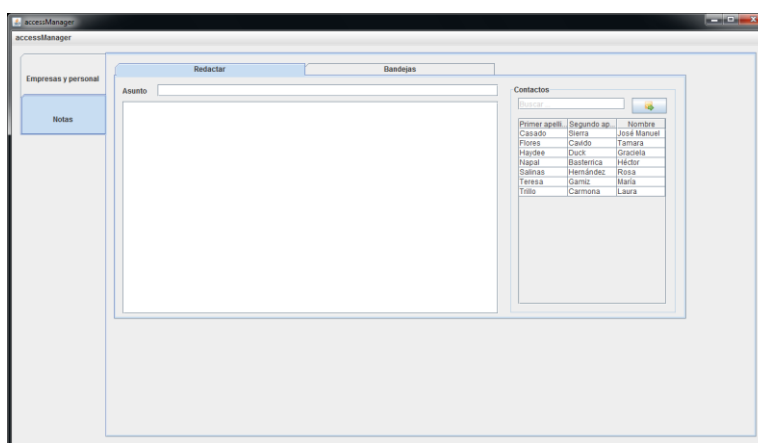
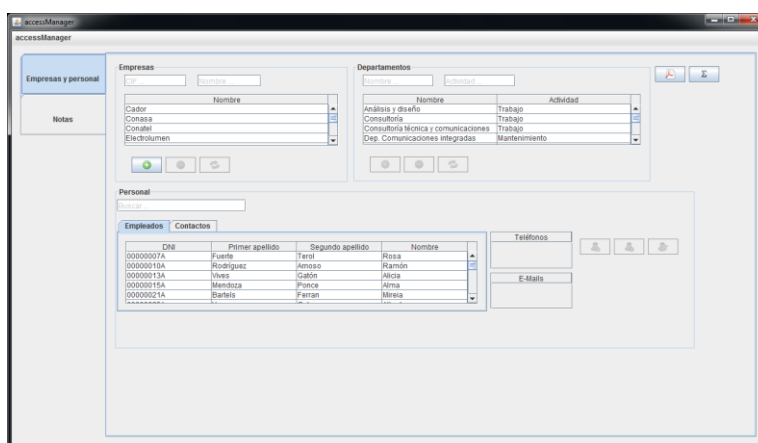
Los dos únicos sucesos que debemos probar son:

- La aplicación debe rediseñarse en función del tipo de usuario que la maneje.
- El contenido que una pestaña muestra debe reflejar el estado actual de la base de datos.

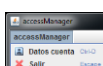
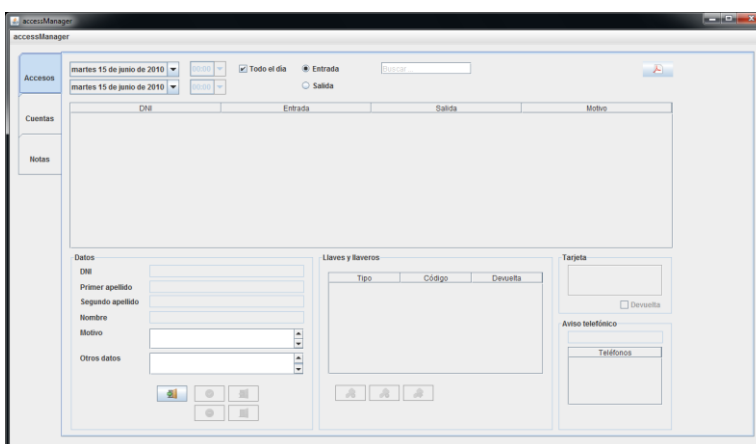
Caso de prueba	12.01 - Inicialización de la aplicación según el tipo de usuario															
Objetivo	Comprobar que accessManager despliega los componentes gráficos correctos en función del tipo de usuario que loguea.															
Actividades	Acceder al sistema con cuentas pertenecientes a diversos tipos de usuarios y comprobar que siempre se despliegan los componentes debidos.															
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. 															
Datos de entrada	<p>Las cuentas que emplearemos son:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Tipo de usuario</th> <th>Usuario</th> <th>Pass</th> </tr> </thead> <tbody> <tr> <td>Administrador</td> <td>hnapalb</td> <td>cambiar</td> </tr> <tr> <td>Autorizador</td> <td>lcarmona</td> <td>cambiar</td> </tr> <tr> <td>Vigilante</td> <td>ghaydeed</td> <td>cambiar</td> </tr> <tr> <td>RP</td> <td>evane</td> <td>cambiar</td> </tr> </tbody> </table>	Tipo de usuario	Usuario	Pass	Administrador	hnapalb	cambiar	Autorizador	lcarmona	cambiar	Vigilante	ghaydeed	cambiar	RP	evane	cambiar
Tipo de usuario	Usuario	Pass														
Administrador	hnapalb	cambiar														
Autorizador	lcarmona	cambiar														
Vigilante	ghaydeed	cambiar														
RP	evane	cambiar														
Salida esperada	La aplicación deberá desplegarse de tal modo que se cumplan siempre las condiciones impuestas en la tabla 'Relación de pestañas y funcionalidad accedida según el tipo de usuarios' de la sección 4.1 de este documento.															
Salida observada	<p>La observada. Tenemos:</p> <ul style="list-style-type: none"> • Administrador: No puede manipular el panel de accesos ni añadir o modificar partes de seguridad. 															



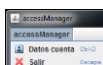
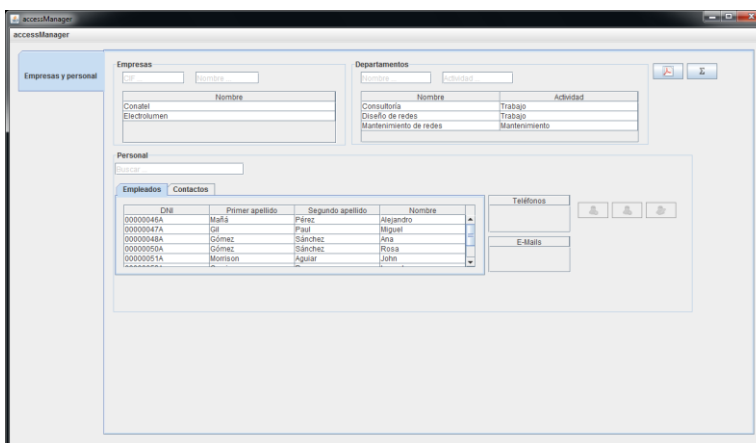
- Autorizador: Solo puede acceder a las pestañas 'Empresas y personal' y 'Notas' y al menú 'Datos cuenta'.



- Vigilante: No tiene acceso a las pestañas de empresas, tarjetas, zonas, préstamos y cuentas, así como al menú que permite manipular la base de datos.



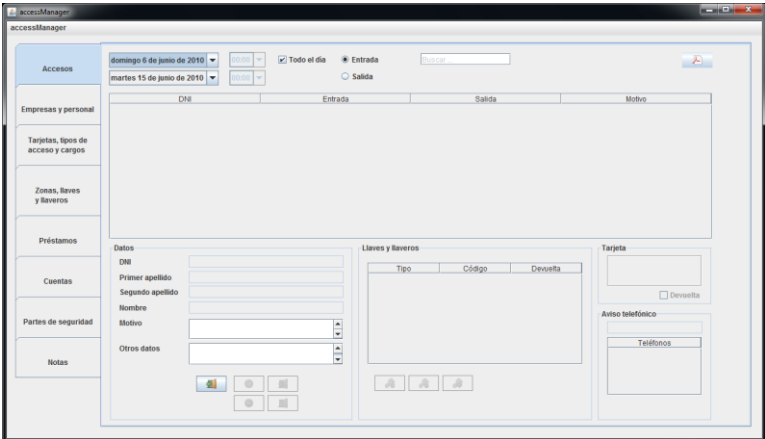
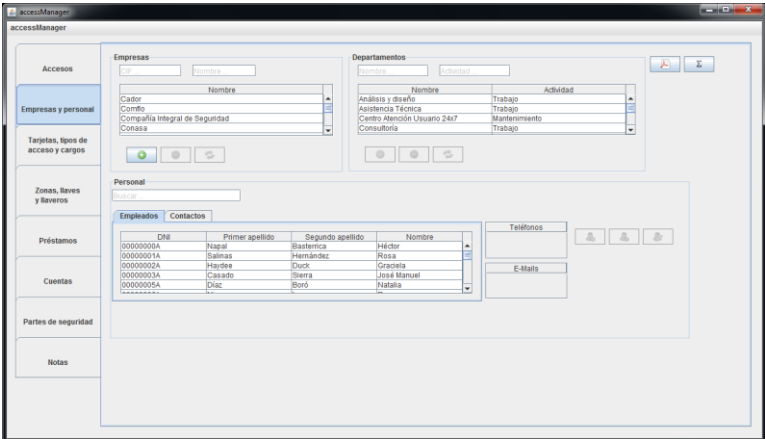
- RP: Única y exclusivamente puede manejar el personal de las empresas-departamentos de las que es responsable y actualizar los datos de su cuenta.

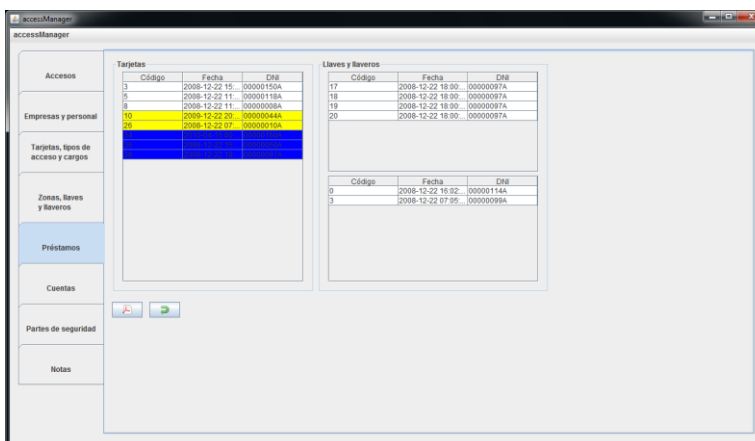
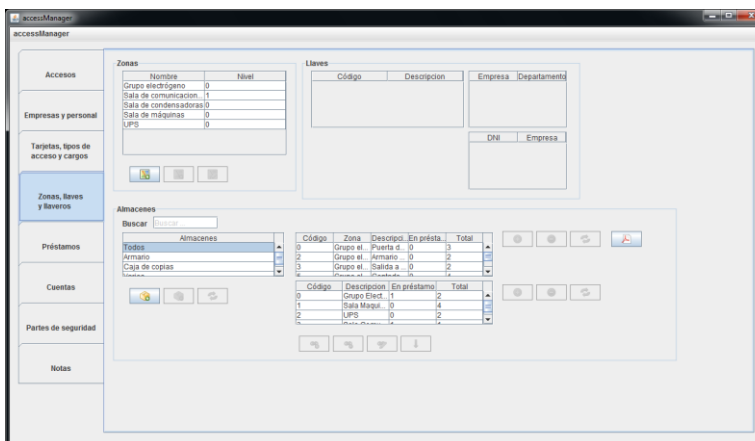
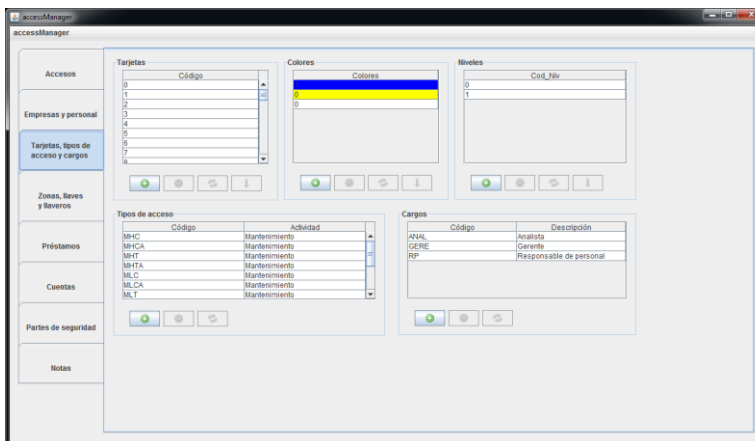


**Evaluación/
conclusión**

La inicialización de la aplicación según el tipo de usuario que accede a ella se realiza siempre satisfactoriamente.

Caso de prueba para la inicialización de la aplicación según el tipo de usuario

Caso de prueba	12.02 - Refresco de los datos de la aplicación
Objetivo	Verificar que todas las pestañas de accessManager siempre muestran la información existente en la base de datos en el momento de ser seleccionadas (desplegadas).
Actividades	Desactivar la función visualización (solo durante este caso de uso, para así poder acceder a todas las pestañas a la vez), desplegar accessManager y comprobar el estado de todas las pestañas. Acto seguido introducir un nuevo elemento en cada pestaña, de modo que al volver a visionarlas éstos deberán constar en ellas.
Técnicas y herramientas	<ul style="list-style-type: none"> • Aplicación accessManager. • Base de datos con población de prueba. • Utilidad phpMyAdmin.
Datos de entrada	<p>Este es el estado inicial de cada una de las pestañas...</p>  





Cuentas

DNI	Tipo	Usuario	Contraseña
00000000A	AUTORIZADOR	hnapab	*****
00000000A	ADMINISTRADOR	hnapab	*****
00000001A	WIGLANTE	hnapab	*****
00000002A	WIGLANTE	ghuades	*****
00000003A	WIGLANTE	casados	*****
00000188A	AUTORIZADOR	mgamg	*****
0000028A	AUTORIZADOR	lcarmona	*****
0000043A	AUTORIZADOR	lcarab	*****
0000063A	RFP	lgarrap	*****
0000068A	IGG	infantocruz	*****

Tipos de usuario

Tipo	Descripción
ADMINISTRADOR	Administrador de sistema
AUTORIZADOR	Autorizador de departamentos
RFP	Responsable de personal de de
WIGLANTE	Vigilante del centro

Historial de accesos al sistema

Desde: Fecha: martes 15 de junio de 2010

Usuario	Tipo	DNI	Fecha
sjmenest	RFP	00001900A	2010-06-15 14:59:42.0
hnapab	RFP	00000188A	2010-06-15 14:57:43.0
hnapab	RFP	00000188A	2010-06-15 14:58:49.0
hnapab	RFP	00000188A	2010-06-15 15:01:06.0
hnapab	RFP	00000188A	2010-06-15 15:16:16.0
hnapab	ADMINISTRADOR	00000000A	2010-06-15 16:19:40.0
hnapab	ADMINISTRADOR	00000000A	2010-06-15 16:17:35.0

Accesos

Cod. Vig. Tipos

Activar fechas Todo el día

Desde: martes 15 de junio de 2010

Hasta: martes 15 de junio de 2010

Fecha
2010-04-16 19:00:00.0
2010-04-15 15:00:00.0
2010-03-02 07:00:00.0
2010-02-24 23:00:00.0
2010-02-12 23:00:00.0
2010-01-23 15:00:00.0

Redactor

Entrada	Salida	
De	Tema	Fecha
Cesado Sienta Ju	Actualización de p...	2008-12-16 15:14

... y éstos los registros nuevos que introduciremos:

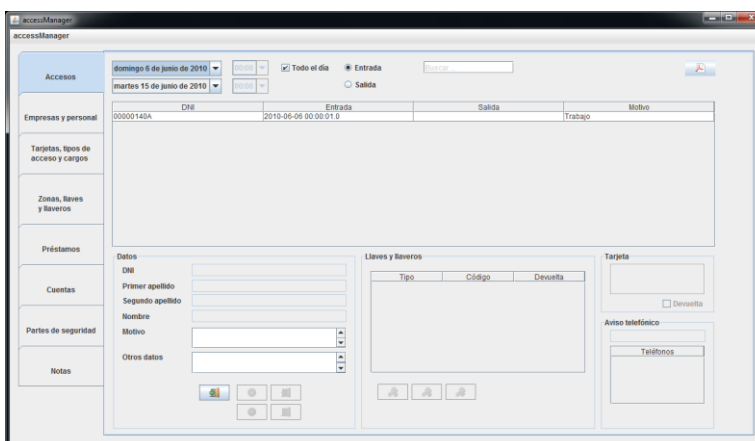
```
Ejecutar la(s) consulta(s) SQL en la base de datos bd:
INSERT INTO Acceso VALUES ('00000140A', '2010-06-06 00:00:01', null, 'Trabajo', 0, '2010-01-01 00:00:01', null, '00000001A');
INSERT INTO Empresa VALUES ('PRU', '99999999A', 'Empresa de prueba', 'Dirección de prueba');
INSERT INTO Tarjeta VALUES (99, CURDATE(), null, -1, '2010-01-01 00:00:01', null);
INSERT INTO Zona VALUES ('PRU', CURDATE(), 'Zona de prueba', 0, '2010-01-01 00:00:01');
INSERT INTO Llave Prestada VALUES ('2010-06-06 00:00:01', 6, '2010-01-01 00:00:01', false);
INSERT INTO Cuenta VALUES ('usuario', 'pass', 'ADMINISTRADOR', '00000005A');
INSERT INTO Parte_Seguridad VALUES (CURDATE(), 0, 'Incidencias de prueba');
INSERT INTO Nota VALUES ('2010-06-06 00:00:01', '00000000A', 'Tema de prueba', 'Texto de prueba', false);
INSERT INTO Destinatario VALUES ('2010-06-06 00:00:01', '00000000A', '00000000A', false, false);
```



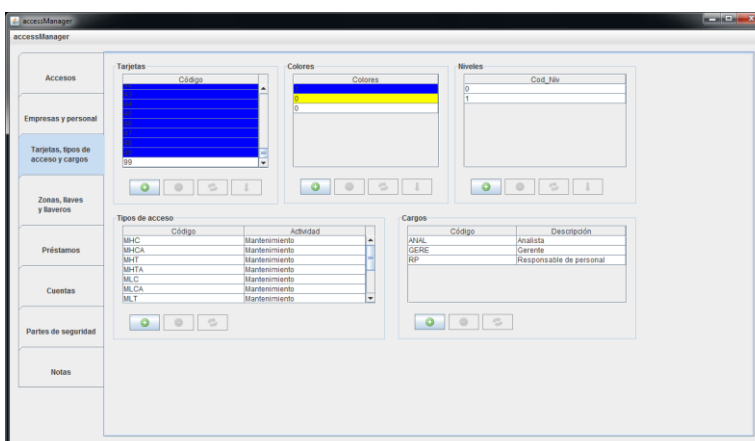
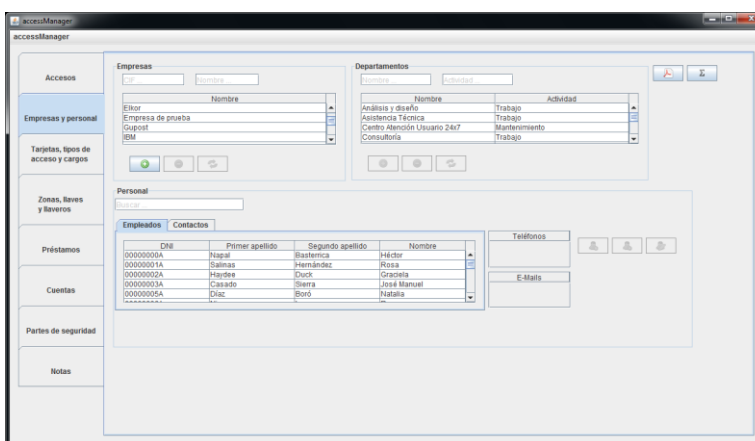
Salida esperada

Sin cerrar la aplicación volveremos a clicar cada pestaña, y todas ellas deberán poseer la misma información que la del punto de partida además de los registros insertados en el punto anterior.

La esperada. Veamos los resultados obtenidos pestaña a pestaña:



Salida observada





The screenshot shows the 'accessManager' application interface. On the left is a navigation menu with options: Accesos, Empresas y personal, Tarjetas, tipos de acceso y cargos, Zonas, llaves y llaveros, Préstamos, Cuentas, Partes de seguridad, and Notas. The main content area is divided into three sections:

- Zonas:** A table with columns 'Nombre' and 'Nivel'. It lists: Grupo electrológico (0), Sala de comunicación (1), Sala de conmutadoras (0), Sala de máquinas (0), LPS (0), and Zona de prueba (0).
- Llaves:** A table with columns 'Código', 'Descripción', 'Empresa', and 'Departamento'. It includes fields for 'DNI' and 'Empresa'.
- Almacenes:** A search bar and two tables. The first table has columns 'Código', 'Zona', 'Descripción', 'En préstamo', and 'Total'. The second table has columns 'Código', 'Descripción', 'En préstamo', and 'Total'.

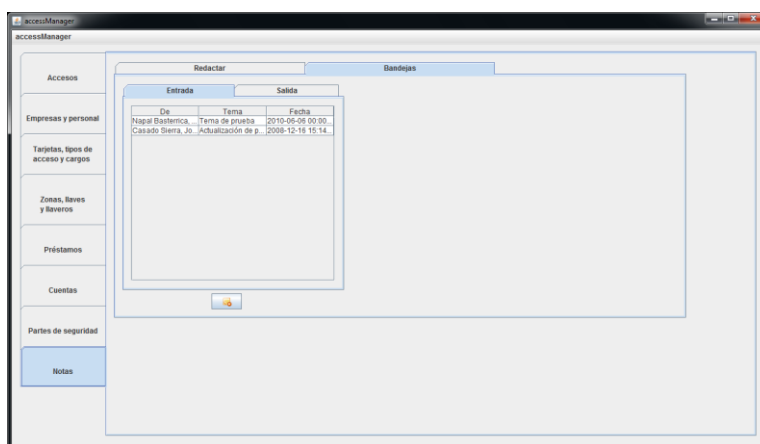
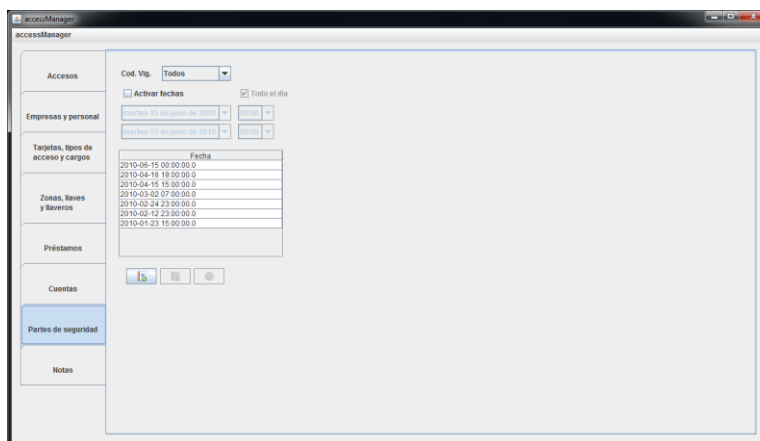
The screenshot shows the 'accessManager' application interface. The navigation menu is on the left. The main content area has two sections:

- Tarjetas:** A table with columns 'Código', 'Fecha', and 'DNI'. It lists several entries with dates ranging from 2010-05-05 to 2008-12-22.
- Llaves y llaveros:** A table with columns 'Código', 'Fecha', and 'DNI'. It lists several entries with dates ranging from 2010-05-05 to 2008-12-22.

The screenshot shows the 'accessManager' application interface. The navigation menu is on the left. The main content area has two sections:

- Cuentas:** A table with columns 'DNI', 'Tipo', 'Usuario', and 'Contraseña'. It lists various user roles like 'VIGILANTE', 'ADMINISTRADOR', 'AUTORIZADOR', 'scarmena', 'Icavido', 'jgarcap', 'adominguez', 'craigas', 'mgonzalez', and 'juchan'.
- Tipos de usuario:** A table with columns 'Tipo', 'Descripción', and 'Descripción'. It lists roles like 'ADMINISTRADOR', 'AUTORIZADOR', 'RP', and 'VIGILANTE' with their respective descriptions.

Below these sections is a 'Historial de accesos al sistema' section with filters for 'Fecha' (martes 15 de junio de 2010) and 'Hora' (martes 15 de junio de 2010). A table below shows a list of system access logs with columns 'Usuario', 'Tipo', 'DNI', and 'Fecha'.



**Evaluación/
conclusión**

El refresco de los datos mostrados en la aplicación se realiza siempre satisfactoriamente.

Caso de prueba para el refresco de los datos mostrados en la aplicación

6.2. De integridad

No se estima oportuno realizar ninguna prueba de integridad ya que los componentes del sistema no han variado en su estructura, aunque sí mínimamente en su composición (dimensión).

6.3. De sistema

Tampoco cabe decir gran cosa al respecto. Se ha modificado mínimamente la aplicación para que se inicialice de modo que ofrezca a los usuarios solo las funcionalidades que deben poder manejar y para que muestre los datos siempre actualizados y de una forma más “amigable”. Tras aprobar las pruebas de unidad no existen elementos que puedan incitar a que cualquier caso de uso ya no se ejecute, con que consideramos que el funcionamiento sigue siendo correcto. En cuanto a la documentación creemos que este documento es suficiente ya que explica todos los elementos generados o modificados en esta iteración.



7. Evaluación de la iteración

En estos momentos sí que podemos concluir que disponemos del sistema que el cliente nos encargó en su momento terminado. Todos los casos de uso han sido resueltos de manera satisfactoria, tal y como hemos venido explicando en todos los artefactos anteriores, la apariencia y usabilidad de la aplicación es buena y los datos que muestra se actualizan con bastante frecuencia. A falta de quizá algún pequeño retoque que pueda surgir en la reunión que tenemos con el cliente, creemos que es el momento de pasar a la fase de Transición, en la que prepararemos la release, instalaremos el sistema en el entorno del cliente y acometeremos tareas de entrenamiento y futuro soporte y mantenimiento.

8. Despliegue

Llegamos a la última reunión de despliegue dentro de la fase de Construcción, en la que mostramos al cliente los cambios realizados sobre la aplicación. La verdad es que no existen entes sobre los que opinar en esta iteración, salvo la apariencia de la interfaz gráfica. Se le pregunta al cliente por ésta y por si estima oportuno realizar algún cambio en el sistema ya que nos disponemos a dar por finalizado su desarrollo siendo éste el momento de realizar los últimos retoques. El cliente responde que está muy satisfecho con el producto que le hemos elaborado y que no aprecia ningún elemento que deba ser modificado. Convenimos pues de mutuo acuerdo dar por concluida la fase de Construcción, pactando que el producto software está finalizado y que en la próxima fase se podrán realizar mínimos retoques pero que los de mayor alcance entrarán a formar parte de la etapa de mantenimiento.

Pamplona, a 6 de abril de 2010.

Héctor Napal Basterrica
Responsable del proyecto

Pedro Martínez Pérez
*Jefe de área de producción de
Informática Navarra S.A.*

Capítulo 3

Transición

3.1. Objetivos de la fase

Aterrizamos en la fase de Transición tras tener implementado el producto software encargado por el cliente. Ésta fase debe servir para asegurar que accessManager está listo para ser utilizado por los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios para su uso y proveer el soporte técnico necesario.

3.2. Iteración T1: Preparación de la release final

La primera tarea en la liberación del producto software será la de prepararlo, creando todo el material necesario para su transmisión al cliente. Crearemos la población real inicial que residirá en un principio en el sistema y completaremos toda la documentación generada hasta el momento, asegurando que sea idónea y acompañándola de un manual de usuario¹ que pueda servir en un futuro para el aprendizaje del uso del sistema.

3.3. Iteración T2: Entrega

La segunda iteración de esta fase tiene como objetivo la entrega del producto para que el usuario final pueda comenzar a darle un uso real. Se realizará la instalación y configuración de todas las máquinas necesarias, además de las pruebas de aceptación y concurrencia, en las que evaluaremos conjuntamente con el cliente el producto en su totalidad y su capacidad para responder ante situaciones de concurrencia máxima.

3.4. Iteración T3: Entrenamiento, soporte y mantenimiento

El entrenamiento, soporte y mantenimiento son actividades de suma importancia y que no pueden ser pasadas por alto. Se planificarán reuniones con todos los grupos de usuarios para proceder con el entrenamiento necesario para que puedan manejar a la perfección el sistema, además de explicarles toda la documentación que se les proporciona y entregarles la aplicación en soporte CD.

Mantendremos una reunión con Informática Navarra S.A. con el fin de acordar quién será el encargado del futuro mantenimiento del sistema. En caso de que vayan a ser ellos, se

¹ No incluido en esta memoria.

reunirá al cliente y responsables del mantenimiento para explicarles detalladamente toda la composición y funcionamiento del sistema. En el otro caso, concertaremos con el cliente un presupuesto y modo de actuación que guíe el mantenimiento en el resto del ciclo de vida de accessManager. Sea cual sea el resultado, el nuevo sistema deberá convivir con el viejo (registros en papel) hasta que quede clara la bondad del primero.



accessManager
Desarrollo de un sistema de control de accesos a un centro

Iteración T2: Entrega

Fecha de entrega: 13 de abril de 2010
Editor: Héctor Napal Basterrica



Tabla de contenidos

1. Introducción

- 1.1. Propósito
- 1.2. Audiencia
- 1.3. Definiciones, acrónimos y abreviaciones
- 1.4. Estructura del documento

2. Estado del proyecto

3. Análisis

- 3.1. Planificación de la iteración
- 3.2. Gestión de riesgos

4. Instalación hardware

5. Instalación y configuración software

6. Pruebas

- 6.1. De aceptación
- 6.2. De concurrencia

7. Evaluación de la iteración



1. Introducción

1.1. Propósito

Redactamos este artefacto con la intención de recoger todas las tareas realizadas en la segunda iteración de la fase de Transición, en la que realizaremos la instalación y configuración del sistema, así como las pruebas restantes.

1.2. Audiencia

El presente documento se desarrollará a lo largo de toda la iteración y será incluido en la documentación entregada en ésta, de modo que pase a disposición del cliente para su futura consulta.

1.3. Definiciones, acrónimos y abreviaciones

- | | | |
|-------|--------------------------|------------------------------|
| • JRE | Java Runtime Environment | Entorno de ejecución de Java |
| • JVM | Java Virtual Machine | Máquina virtual de Java |
| • SSL | Secure Sockets Layer | Capa de conexiones seguras |

1.4. Estructura del documento

En la segunda sección realizamos una breve aproximación al estado del proyecto y qué objetivos buscaremos con esta iteración. A continuación realizamos un análisis de todo el trabajo a desarrollar mediante la planificación de la iteración y el estudio de qué riesgos podrían acecharnos. Seguimos con los capítulos 4 y 5, donde explicamos cómo realizamos la instalación hardware y cómo le dotamos a éste de todo el software necesario. Ideamos el apartado 6 como una breve explicación de las pruebas realizadas en esta iteración (de aceptación y concurrencia) y damos fin a este artefacto con una breve evaluación de la iteración en la séptima sección.

2. Estado del proyecto

Tenemos el producto software desarrollado y documentado, de modo que se encuentra en perfectas condiciones para comenzar con la liberación del mismo.



3. Análisis

3.1. Planificación de la iteración

Estas son las actividades que desarrollaremos en los tres próximos días:

PLAN DE ITERACIÓN T2: ENTREGA																																																																																						
ASPECTO	DESCRIPCIÓN																																																																																					
Objetivos	Instalar, configurar y probar el sistema de modo que éste pueda comenzar a ser utilizado por el usuario final.																																																																																					
Artefactos de entrada	<ul style="list-style-type: none"> Planificación temporal y costes detallados del proyecto. Documentación generada hasta el momento en papel y soporte electrónico (CD). Aplicación accessManager, que consta de todo el código implementado junto con el archivo ejecutable de la aplicación (.jar) y el Java Runtime Environment, que permita la ejecución de ésta a través de la Java Virtual Machine (CD). 																																																																																					
Artefactos modificados o creados	<ul style="list-style-type: none"> Plan de pruebas. El presente documento. 																																																																																					
Calendarización detallada	Análisis	09/04/10	<ul style="list-style-type: none"> Planificación de la iteración y gestión de riesgos. Planificación de las pruebas. Inicio de la documentación. 																																																																																			
	Instalación + Configuración	10/04/10	<ul style="list-style-type: none"> Instalación hardware. Instalación y configuración software. 																																																																																			
	Prueba de aceptación	11/04/10	<ul style="list-style-type: none"> Realización de las pruebas de aceptación y concurrencia con el cliente. Finalización y entrega de toda la documentación. 																																																																																			
		13/04/10																																																																																				
<table border="1"> <thead> <tr> <th>Nombre de tarea</th> <th>Duración</th> <th>Comienzo</th> <th>Fin</th> <th>0</th> <th>vie 09 abr</th> <th>dom 11 abr</th> <th>mar 13 abr</th> </tr> </thead> <tbody> <tr> <td>Estudio de oportunidad (inception)</td> <td>7 días</td> <td>jue 07/01/10</td> <td>vie 15/01/10</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Elaboración</td> <td>16 días</td> <td>lun 18/01/10</td> <td>lun 08/02/10</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Construcción</td> <td>41 días</td> <td>mar 09/02/10</td> <td>mar 06/04/10</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Transición</td> <td>8 días</td> <td>mié 07/04/10</td> <td>vie 16/04/10</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Iteración T1: Preparación de la release final</td> <td>2 días</td> <td>mié 07/04/10</td> <td>jue 08/04/10</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Iteración T2: Entrega</td> <td>3 días</td> <td>vie 09/04/10</td> <td>mar 13/04/10</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Despliegue</td> <td>3 días</td> <td>vie 09/04/10</td> <td>mar 13/04/10</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Iteración T3: Entrenamiento, soporte y mantenimiento</td> <td>3 días</td> <td>mié 14/04/10</td> <td>vie 16/04/10</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Liberación del producto</td> <td>0 días</td> <td>vie 16/04/10</td> <td>vie 16/04/10</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>							Nombre de tarea	Duración	Comienzo	Fin	0	vie 09 abr	dom 11 abr	mar 13 abr	Estudio de oportunidad (inception)	7 días	jue 07/01/10	vie 15/01/10					Elaboración	16 días	lun 18/01/10	lun 08/02/10					Construcción	41 días	mar 09/02/10	mar 06/04/10					Transición	8 días	mié 07/04/10	vie 16/04/10					Iteración T1: Preparación de la release final	2 días	mié 07/04/10	jue 08/04/10					Iteración T2: Entrega	3 días	vie 09/04/10	mar 13/04/10					Despliegue	3 días	vie 09/04/10	mar 13/04/10					Iteración T3: Entrenamiento, soporte y mantenimiento	3 días	mié 14/04/10	vie 16/04/10					Liberación del producto	0 días	vie 16/04/10	vie 16/04/10				
Nombre de tarea	Duración	Comienzo	Fin	0	vie 09 abr	dom 11 abr	mar 13 abr																																																																															
Estudio de oportunidad (inception)	7 días	jue 07/01/10	vie 15/01/10																																																																																			
Elaboración	16 días	lun 18/01/10	lun 08/02/10																																																																																			
Construcción	41 días	mar 09/02/10	mar 06/04/10																																																																																			
Transición	8 días	mié 07/04/10	vie 16/04/10																																																																																			
Iteración T1: Preparación de la release final	2 días	mié 07/04/10	jue 08/04/10																																																																																			
Iteración T2: Entrega	3 días	vie 09/04/10	mar 13/04/10																																																																																			
Despliegue	3 días	vie 09/04/10	mar 13/04/10																																																																																			
Iteración T3: Entrenamiento, soporte y mantenimiento	3 días	mié 14/04/10	vie 16/04/10																																																																																			
Liberación del producto	0 días	vie 16/04/10	vie 16/04/10																																																																																			
Recursos	Temporales		3 días																																																																																			
	Humanos		12 personas																																																																																			
	Software	Sistema operativo		Microsoft Windows 7																																																																																		
		Ofimática		Microsoft Word 2010 Adobe Acrobat Professional 8																																																																																		
		Diseño		Microsoft Visio 2007 Visual Paradigm 4.2																																																																																		
		Codificación		NetBeans 6.8																																																																																		
Hardware		11 PCs																																																																																				

Planificación de la iteración

accessManager · Iteración T2: Entrega · Página 4 de 10



3.2. Gestión de riesgos

Una instalación exitosa es una condición necesaria para el funcionamiento de cualquier software. El equipo y software que debemos instalar es sencillo, pero nunca viene mal hacer un pequeño estudio de los riesgos de la iteración:

IDENTIFICACIÓN DE RIESGOS		
TIPO	EXISTENCIA	DESCRIPCIÓN
Tecnología y herramientas	No se observan	La máquina servidor a instalar es nueva, equipada con Windows 7 y en la que deberá correr un servidor web, otro de bases de datos, la aplicación desarrollada y la utilidad phpMyAdmin. accessManager solo necesitará de la JVM, mientras que el resto de componentes son aplicaciones muy asentadas en el mercado y de máximas garantías, con un proceso de instalación sencillo. Para el resto de máquinas clientes únicamente se estima oportuno instalar el JRE en caso de que no dispongan de éste. Se trata de un proceso de instalación y configuración muy sencillo en el que apreciamos un grado de complejidad muy bajo.
Requerimientos	No se observan	El cliente ha sido partícipe en el desarrollo y prueba de todo el software elaborado a lo largo de la fase de Construcción. Ésta fase debería ser realizada con vistas a hacerle ver que el software es el que demandaba y el adecuado para sus pretensiones, no estando previsto que puedan sugerirse cambios, o que si surgen sean de carácter mínimo.
Estimación	No se observan	Se designan tres días para la instalación, configuración y prueba de todo el sistema. Se trata de un tiempo más que prudente para realizar la instalación y prueba exhaustiva de éste.

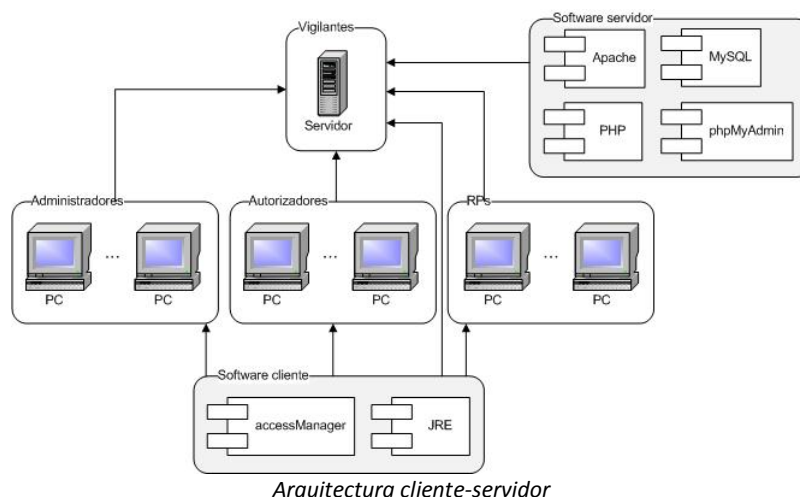
Identificación de riesgos

4. Instalación hardware

En el estudio de la arquitectura del sistema se estableció que este dispondría de una arquitectura cliente-servidor en la que una máquina diese servicio al resto. Los usuarios que emplearán accessManager ya disponen de ordenadores conectados a la red Internet, con que el único componente hardware que deberemos instalar será el servidor, que a su vez será usado como máquina cliente para los vigilantes.



Tras desembalar el ordenador adquirido, conectarlo a la red y ponerlo en marcha disponemos de la siguiente disposición/jerarquía de máquinas:



5. Instalación y configuración software

Una vez que disponemos de la arquitectura hardware creada podemos pasar a instalar todo el software necesario.

En lo que respecta a los clientes, es de suponer que todos los ordenadores dispongan del JRE ya instalado, pues se trata de una herramienta muy utilizada. En caso de no disponer de ella su instalación es trivial ya que se trata de un proceso guiado en el que no se debe configurar ningún parámetro. accessManager, como aplicación Java, no requerirá de instalación y será ejecutada siempre y cuando tengamos asociados los archivos *.jar* a la JVM.

Del lado del servidor nos encontraríamos ante una actividad mucho más compleja de no existir los instaladores WAMP, que instalan Apache, PHP, MySQL y phpMyAdmin en un solo proceso de instalación guiado. Existen varios, aunque nosotros optaremos por el de mayor prestigio: WAMPServer en su versión 2.0i. Proporcionamos dos instaladores junto con el resto del software, uno con WAMPServer y otro con la extensión de MySQL necesaria para disponer de la más reciente y poder ejecutar disparadores. Ambas también son descargables desde el sitio web <http://www.wampserver.com/en/download.php>.



El proceso de instalación resulta trivial al ser en ambos casos guiado. Las únicas configuraciones que deberemos realizar son:

1. Permitir las conexiones desde el exterior en el programa antivirus y los ficheros de configuración de Apache y phpMyAdmin

(C:/WAMP/bin/apache/Apache2.2.11/conf/httpd.conf...

```
<Directory />
  Options FollowSymLinks
  AllowOverride None
  Order deny,allow
  Deny from all
</Directory>

<Directory />
  Options FollowSymLinks
  AllowOverride None
  Order deny,allow
  Allow from all
</Directory>

<Directory "c:/WAMP/www/">
  Options Indexes FollowSymLinks
  AllowOverride all
  Order Deny,Allow
  Deny from all
  Allow from 127.0.0.1
</Directory>

<Directory "c:/WAMP/www/">
  Options Indexes FollowSymLinks
  AllowOverride all
  Order Deny,Allow
  Deny from all
  Allow from all
</Directory>
```



... y C:/WAMP/alias/phpmyadmin.conf):

```
<Directory "c:/WAMP/apps/phpmyadmin3.2.0.1/">
  Options Indexes FollowSymLinks MultiViews
  AllowOverride all
  Order Deny,Allow
  Deny from all
  Allow from 127.0.0.1
</Directory>
```

```
<Directory "c:/wamp/apps/phpmyadmin3.2.0.1/">
  Options Indexes FollowSymLinks MultiViews
  AllowOverride all
  Order Deny,Allow
  Deny from all
  Allow from all
</Directory>
```

2. **Configurar un nombre de usuario y contraseña para phpMyAdmin**, de modo que solo puedan acceder a ésta utilidad aquellas personas que que dispongan de privilegios para ello. La forma más sencilla y rápida de llevar a cabo esta operación es abrir el explorador web, ir a <http://localhost/phpMyAdmin/> y clicar en 'Privilegios', donde editaremos los siguientes usuarios:

	Usuario	Servidor	Contraseña	Privilegios globales 1	Conceder
<input type="checkbox"/>	cualquiera	%	--	USAGE	No
<input type="checkbox"/>	cualquiera	localhost	No	USAGE	No
<input checked="" type="checkbox"/>	root	127.0.0.1	No	ALL PRIVILEGES	Si
<input checked="" type="checkbox"/>	root	localhost	No	ALL PRIVILEGES	Si

Pestaña 'Privilegios' con los usuarios a editar

Cambio de contraseña

Sin contraseña

Contraseña: Debe volver a escribir:

Hashing de la contraseña: MySQL 4.1+ Compatible con MySQL 4.0

Generar la contraseña

Sección de modificación de contraseña del usuario

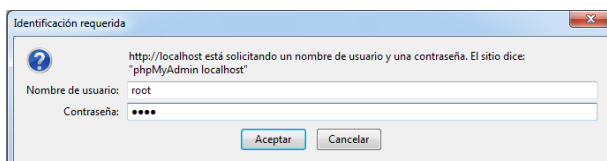


Además debemos modificar el fichero *config.inc.php* ubicado en *C:/WAMP/apps/phpmyadmin3.2.0.1* haciendo que el logueo sea vía HTTP y no automática (configurado por defecto):

```
$cfg['Servers'][$i]['auth_type'] = config;  
$cfg['Servers'][$i]['user'] = 'root';  
$cfg['Servers'][$i]['password'] = '';  
$cfg['Servers'][$i]['AllowNoPassword'] = true;
```

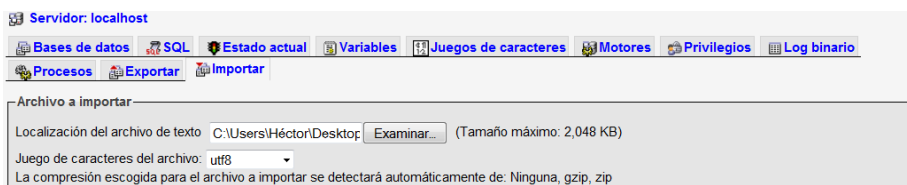
```
$cfg['Servers'][$i]['auth_type'] = 'http';  
$cfg['Servers'][$i]['user'] = 'root';  
$cfg['Servers'][$i]['password'] = '';  
$cfg['Servers'][$i]['AllowNoPassword'] = false;
```

A partir de ahora siempre será necesario introducir usuario y contraseña para acceder a phpMyAdmin:



Ventana de identificación desplegada al intentar acceder a phpMyAdmin

Para terminar, solo restaría importar la base de datos desde la pestaña 'Importar' de phpMyAdmin, disponiendo así del sistema completo instalado y configurado.



Importación de una base de datos a través de phpMyAdmin

Tabla	Acción	Registros	Tipo	Cotejamiento	Tamaño	Residuo a depurar
acceso		32	InnoDB	utf8_spanish_ci	16.0 KB	-
acceso_sistema		80	InnoDB	utf8_spanish_ci	32.0 KB	-
almacen		3	InnoDB	utf8_spanish_ci	32.0 KB	-
almacen_llave		58	InnoDB	utf8_spanish_ci	48.0 KB	-
almacen_llavero		4	InnoDB	utf8_spanish_ci	48.0 KB	-
cargo		3	InnoDB	utf8_spanish_ci	16.0 KB	-
codigo		1,000	InnoDB	utf8_spanish_ci	48.0 KB	-
color		3	InnoDB	utf8_spanish_ci	32.0 KB	-
composicion_llavero		15	InnoDB	utf8_spanish_ci	48.0 KB	-

Base de datos importada



6. Pruebas

Una vez terminamos de instalar todo el hardware y software necesario, podemos proceder a realizar las últimas pruebas necesarias para dar por finalizado el proceso de elaboración del software: la de aceptación del sistema, junto con el cliente, y la de concurrencia, con la ayuda de varios operadores.

6.1. Prueba de aceptación

Destinada a probar los requisitos de usuario, en ella evaluamos conjuntamente con el cliente el producto, haciéndole ver que es el apropiado para sus pretensiones. Todos los casos de uso han vuelto a ser probados con resultados positivos, quedando el cliente muy satisfecho con el software desarrollado.

6.2. Prueba de concurrencia

Para concluir se ejecutan varios de los casos de uso en situaciones de máxima concurrencia (dos administradores, tres autorizadores y seis responsables de personal), comprobando que el sistema es robusto y que responde bien ante situaciones límite.

7. Evaluación de la iteración

Damos por finalizada la iteración 'T2 - Entrega', en la que hemos desplegado y probado todo el sistema con resultados positivos. Además, se realiza la entrega de todo el material desarrollado, tanto software como documentación en papel y formato electrónico (CD).

A falta de realizar la última iteración (en la que el producto software no será modificado) podemos dar por concluida la implementación del sistema encargado habiendo conseguido un gran grado de satisfacción en el cliente, con lo que todos nuestros objetivos han sido cumplidos.

Pamplona, a 13 de abril de 2010.

Héctor Napal Basterrica
Responsable del proyecto

accessManager · Iteración T2: Entrega · Página 10 de 10

accessManager

Desarrollo de un sistema de control de accesos a un centro

Héctor Napal Basterrica
Edurne Barrenechea Tartas
Pamplona, 24 de junio de 2010

Veremos

- Ámbito y empresa acometida
- Elección del proceso de desarrollo
- Desarrollo del proyecto
- Demostraciones
- Ruegos y preguntas

Veremos

- **Ámbito y empresa acometida**
- Elección del proceso de desarrollo
- Desarrollo del proyecto
- Demostraciones
- Ruegos y preguntas

Ámbito y empresa acometida

- **PIN S.L.: Introducción**

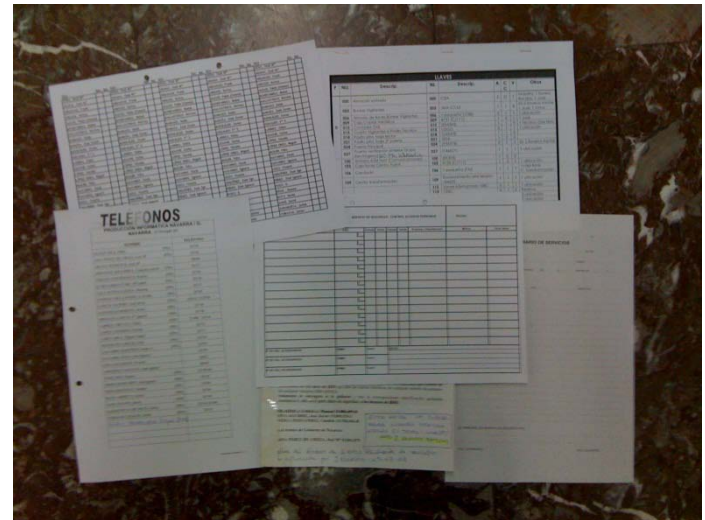
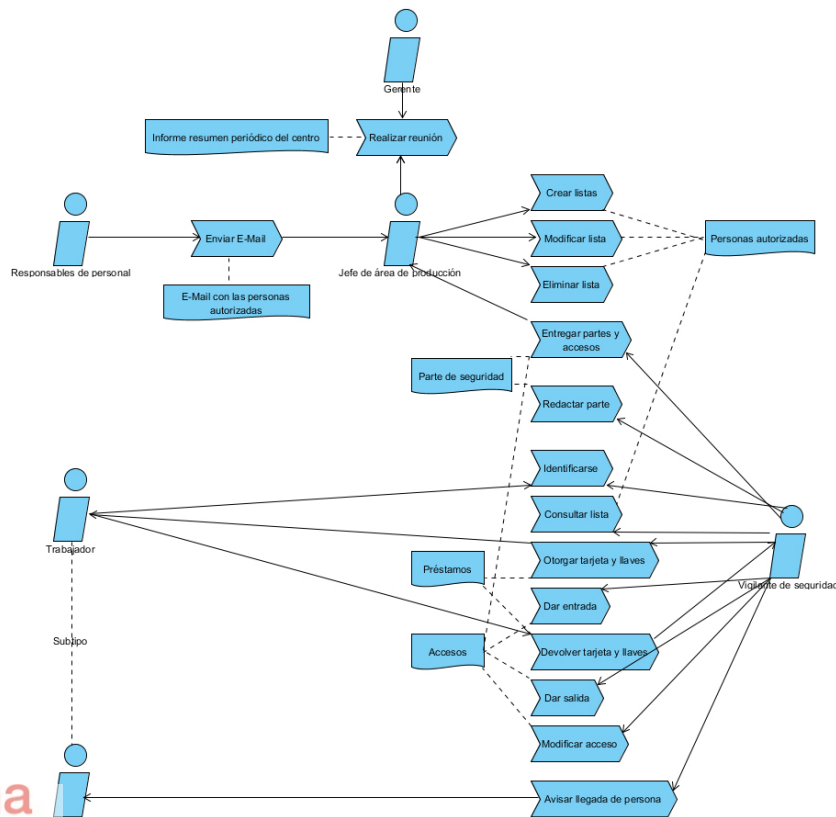
Empresa pública perteneciente al Gobierno de Navarra, creada para proveer al mismo y a los organismos e instituciones que dependen de él, de sistemas de información y de la explotación de los mismos.

PiN Producción Informática de Navarra S.L.



Ámbito y empresa acometida

- PIN S.L.: Modelo de negocio



Ámbito y empresa acometida

- **Objetivo**

Analizar, diseñar, implementar, implantar, configurar y mantener todo el software necesario que provea las siguientes funcionalidades principales:

- *Control de accesos al centro, con el consecuente tratamiento de toda la información necesaria.*
- *Redacción, visualización y almacenamiento de los partes de vigilancia del centro.*
- *Mensajería entre usuarios.*

iN Informática Navarra

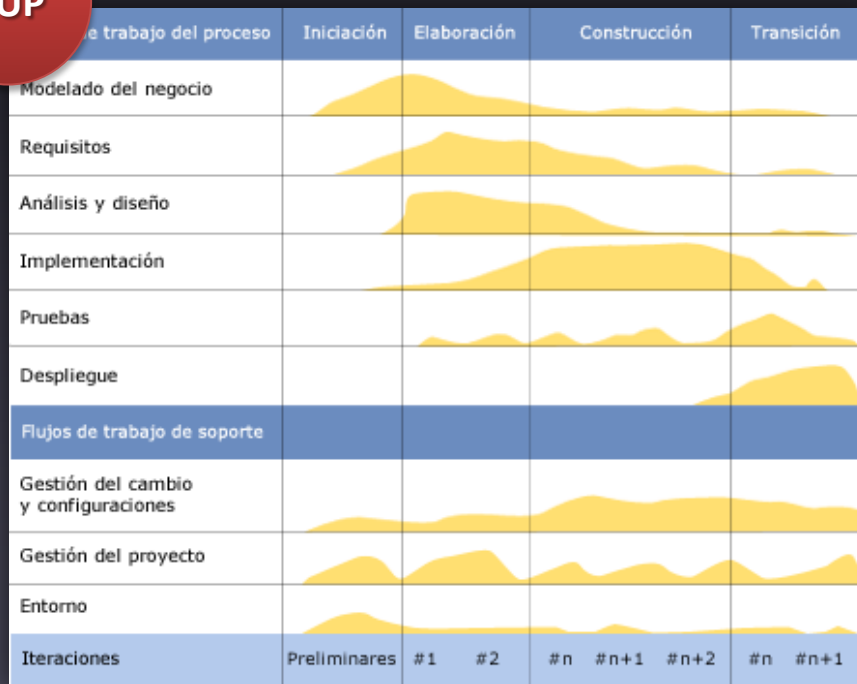
Software  Solutions

Veremos

- **Ámbito y empresa acometida**
- **Elección del proceso de desarrollo**
- Desarrollo del proyecto
- Demostraciones
- Ruegos y preguntas

Elección del proceso de desarrollo

RUP



- Iterativo e incremental
- Dirigido por los CdUs

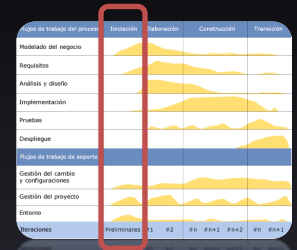
Veremos


- **Ámbito y empresa acometida**
- Elección del proceso de desarrollo
- **Desarrollo del proyecto**
- Demostraciones
- Ruegos y preguntas

Desarrollo del proyecto

- **Iniciación**

- Estudio inicial de la viabilidad del proyecto software.





Software Solutions

accessManager
 Desarrollo de un sistema de control de acceso a un centro

Estudio inicial de la viabilidad del proyecto software

Fecha de entrega: 15 de enero de 2010
 Editor: Jaan Hernández Córdoba



3.3. Planificación temporal
 El ciclo de vida RUP presenta cuatro fases divididas en iteraciones, cada una de las cuales desarrollará en menor o mayor medida las disciplinas o flujos de trabajo fundamentales característicos.

Se estima la siguiente distribución de tiempos:

FASE	DURACION
Estudio de oportunidad (Inception)	1,5 semanas
Elaboración	3 semanas
Construcción	7,5 semanas
Transición	1,5 semanas
TOTAL	13,5 semanas

Estimación temporal de la fase RUP

3.4. Estimación económica
 Los característicos del proyecto nos hacen prever los siguientes costes:

MANO DE OBRERA			
CARGO	TIEMPO	SALARIO/SEMANA	TOTAL
Director del proyecto, Ingeniero de desarrollo y soporte	13,5 semanas	375 €	5063 €

MATERIA PRIMA			
MATERIA	CANTIDAD	VALOR / \$ AÑOS	TOTAL
PC's ambiente de desarrollo	1	400 €	25.274 €
Licencia software	1	4000 €	220.184 €
Servidor	1	450 €	450 €
TOTAL			221.058 €

Costes estimados económicos del desarrollo del proyecto

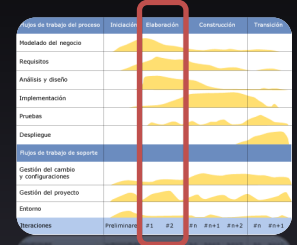
4. Conclusiones
 La empresa encargada resulta ser, en un principio, un proyecto convencional en la sociedad que nos concierne, no requiriendo una avanzada tecnología y que podrá ser desarrollada perfectamente bajo las condiciones iniciales impuestas por el cliente.
 Desde el punto de vista coste-beneficio, resulta ser un sistema esencial en un centro de tales características y una inversión muy atractiva para los intereses de la empresa y los ventajas que proporciona.

accessManager - Estudio inicial de la viabilidad del proyecto software - Página 5 de 5

Desarrollo del proyecto

- **Elaboración**

- Estudio detallado del problema y modelado de negocio.



accessManager
Desarrollo de un sistema de control de acceso a un centro

Estudio detallado del problema software

Fecha de entrega: 19 de enero de 2010
Editor: Néstor Najar Basteriza

empresas y departamentos, siendo necesario que sean accesibles en algunos de los accesos que realizan sus trabajadores.

El sistema posee de varias zonas enlazadas en diferentes niveles de acceso, de modo que un trabajador pueda acceder a una o varias en función del nivel de acceso que le otorgue la tarjeta asignada en su acceso al centro. Las tarjetas están elaboradas en función del nivel de acceso que poseen, pudiendo corregirlos cuando a un mismo nivel.

El acceso permitido a una zona no implica que se pueda acceder a toda ella, quedando para tal fin, (impedidos) y elementos de ella (ej. armarios) pendientes de que el trabajador disponga de la clave que permita su apertura. Todos los datos de acceso almacenados pueden ser identificados mediante perfiles por parte del sistema de seguridad.

2.2. Es personal y salas

El acceso a la sala consta de dos puertas, una de apertura manual y otra automática, que solo puede ser desbloqueada al alguno de los lectores de tarjetas que la componen (por zona y entrada), para entrar y para salir. Se recomienda permanentemente una tarjeta como válida. En este lugar se dibuja en media el control de acceso al centro, como principio de desarrollo quedará resumido en la siguiente tabla:

AREA	SECTOR	TIPO	ACCESO	TIPO	USUARIO	Ejemplo	Límite
NO1	SI	NO	X	SI	SI	Presión tarjeta y Bases	Desarrollar tarjetas y bases
NO2	SI	NO	X	SI	NO	Presión tarjeta	Desarrollar tarjetas
NO3	SI	NO	X	SI	SI	Por presión accesos	Desarrollar tarjetas
NO4	SI	SI	SI	SI	NO	Presión tarjeta y Bases	Desarrollar tarjetas y bases
SI	SI	SI	SI	SI	SI	Presión tarjeta y Bases	Desarrollar tarjetas y bases
SI	X	---	NO	NO	---	Puede acceder	Desarrollar Bases
SI	X	---	NO	---	---	Presión Bases	Desarrollar Bases
SI	NO	---	SI	SI	SI	No puede acceder	Desarrollar tarjetas
SI	SI	---	SI	NO	---	Presión tarjeta	Desarrollar tarjetas
SI	SI	---	SI	SI	SI	Presión tarjeta y Bases	Desarrollar tarjetas y bases

(Continúa en otro documento)

Nota: Toda petición será un requisito, en caso de no ser así confirmará por correo de correo.

accessManager - Estudio detallado del problema software - Página 4 de 6

3. Modelo de casos de uso y objetos del negocio principales

Para no extenderse en exceso y simplificar el modelado de negocio, a continuación resumimos los dos modelos que lo componen, estudiando los procesos más importantes y complejos a través de sus respectivos diagramas, junto con unas breves explicaciones de los mismos.

Diagrama de casos de uso y objetos del negocio - Central

accessManager - Especificación del modelo de negocio - Página 4 de 6

2.1. Continuar lista

Las listas que continúan todas las personas autorizadas a entrar al centro son elaboradas (consultas, inserciones y eliminaciones) por el jefe de zona de producción a partir de los 5 Mails recibidos por parte de los responsables del personal de los Departamentos de las empresas que trabajan en la sede de Informática Huskara I.A.

Diagrama de casos de uso y objetos del negocio - Continuar lista

2.2. Entrada de una persona

La llegada al centro de una persona implica todo el proceso de acceso. Esta se dirige al vigilante identificándose mediante su DNI. El vigilante comprueba que coincide en alguna de las listas, si el encuentro ocurre de no horas laborales se le muestra una autorización telefónica. Tras realizar las comprobaciones necesarias se otorga la tarjeta o tarjeta de acceso que necesite, registrando su entrada en una hoja de acceso y dándole su libre.

Diagrama de casos de uso y objetos del negocio - Entrada de una persona

2.3. Salida de una persona

A la salida del centro el trabajador devuelve la tarjeta a Bases y/o Basteriza que le hacen salir personal. A continuación, el vigilante de seguridad registra la devolución. Se crea en la hoja de acceso la impresión de salida y se permite su salida.

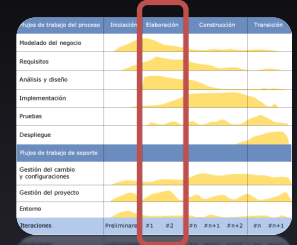
Diagrama de casos de uso y objetos del negocio - Salida de una persona

accessManager - Especificación del modelo de negocio - Página 5 de 6

Desarrollo del proyecto

- **Elaboración**

- Estudio detallado del problema y modelado de negocio.
- Captura de requisitos.



accessManager
Desarrollo de un sistema de control de accesos a un centro

Especificación inicial de los casos de uso

Fecha de entrega: 28 de enero de 2008
Edición: 16 de febrero de 2008

1. Introducción

1.1. Propósito
A lo largo de las siguientes páginas realizamos la especificación inicial de los casos de uso que deberán ser implementados por el nuevo sistema. Este es el documento clave que deberá ser leído en cuenta y reflejado a lo largo de todo el desarrollo software.

1.2. Audiencia
Se trata de un documento de alto nivel, destinado a ser aprobado por el cliente y servirá de base para el trabajo del equipo de desarrollo.

1.3. Estructura del documento
En la segunda sección identificamos y describimos los actores que usarán el nuevo sistema. A continuación, como finalización del documento, tenemos a cabo una especificación de los principales casos de uso, detallando qué funcionalidad proporcionará y las relaciones que existen entre ellos.

2. Actores del sistema
Definimos actores y entornos estableciendo los siguientes actores:

CAMPO	DESCRIPCIÓN
Nombre	Administrador
Descripción	Administrador del sistema.
Principales usos del sistema:	<ul style="list-style-type: none"> Realizar tareas de mantenimiento de la información, basadas en una navegación por menú. Obtener toda la información necesaria para realizar los labor. Comunicarse con los demás usuarios del sistema.
Nombre	Vigilante
Descripción	Vigilante de seguridad del centro.
Principales usos del sistema:	<ul style="list-style-type: none"> Usar un control eficiente de los accesos realizados al centro a de los visitantes que conforman. Realizar los planes de seguridad de los servicios que realiza. Comunicarse con los demás usuarios del sistema.

CAMPO	DESCRIPCIÓN
Caso de uso	01. Visualizar información de Bases
Funcionalidad	Administrador
Caso de uso	02. Visualizar información de Bases
Funcionalidad	Administrador
Caso de uso	03. Añadir información de Bases
Funcionalidad	Administrador
Caso de uso	04. Borrar información de Bases
Funcionalidad	Administrador
Caso de uso	05. Modificar información de Bases
Funcionalidad	Administrador
Caso de uso	06. Consultar información de Bases
Funcionalidad	Administrador

Diagrama de flujo de datos - Sistema de Bases

Desarrollo del proyecto

• Elaboración

- Estudio detallado del problema y modelado de negocio.
- Captura de requisitos.
- Definición de la arquitectura básica.



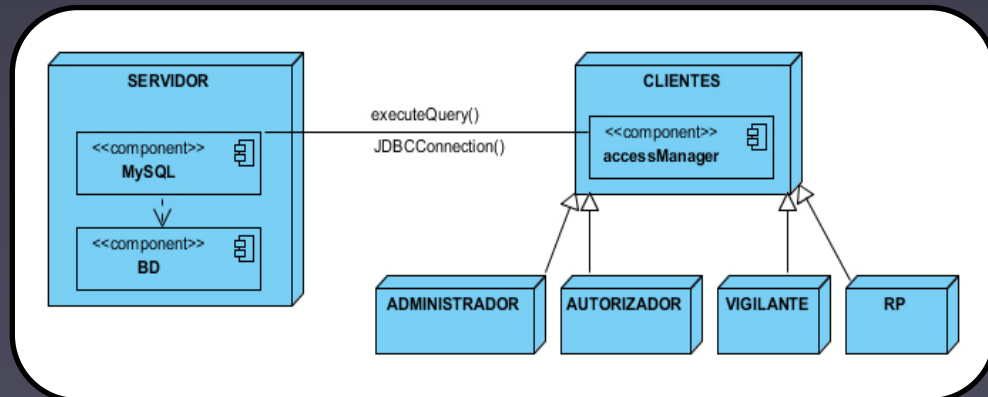
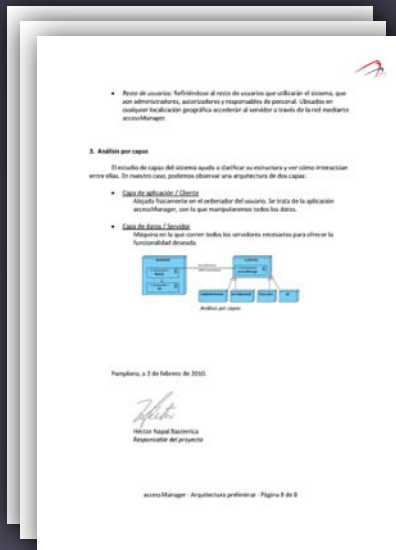
The image shows three overlapping pages from a technical document. The top page is the cover page, the middle page is page 4, and the bottom page is page 8. The document is titled 'Software Solutions' and 'accessManager Desarrollo de un sistema de control de accesos a un centro'. The middle page (page 4) contains the text '2.3. Elección del modelo de la base de datos' and discusses the choice of a distributed database model. The bottom page (page 8) contains the text '3. Análisis por capas' and discusses the analysis of the system's layers, including client and server components. The document is dated '3 de febrero de 2010' and signed by 'Mikel Izuel Basterrika'.

Desarrollo del proyecto



- **Elaboración**

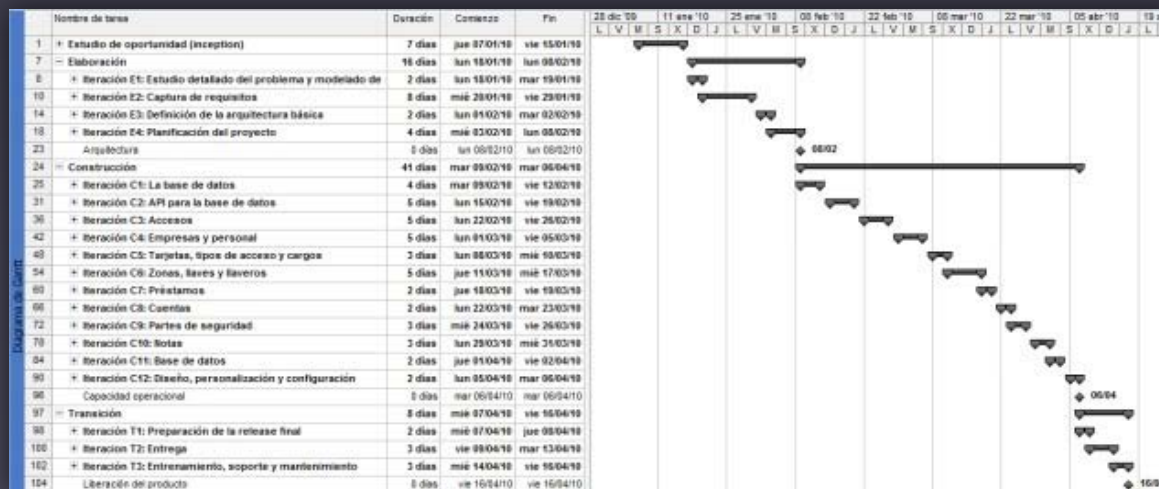
- Estudio detallado del problema y modelado de negocio.
- Captura de requisitos.
- Definición de la arquitectura básica.



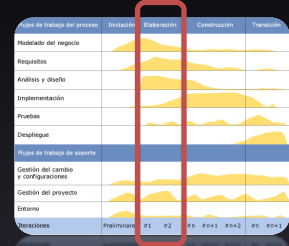
Desarrollo del proyecto

- **Elaboración**

- Estudio detallado del problema y modelado de negocio.
- Captura de requisitos.
- Definición de la arquitectura básica.
- Planificación del proyecto.



Desarrollo del proyecto



- **Elaboración**

- Estudio detallado del problema y modelado de negocio.
- Captura de requisitos.
- Definición de la arquitectura básica.
- Planificación del proyecto.
- Plan de pruebas inicial.

The image shows three overlapping pages from a test plan document for 'accessManager'. The top page is the cover page, the middle page is the '5. concreción de las pruebas' section, and the bottom page is the '4. Recursos' section.

Page 1: Cover Page

Software Solutions

accessManager
Desarrollo de un sistema de control de acceso a

Plan de pruebas inicial

Fecha de entrega: 29 de enero de 2003
Editor: Hector Navea Basterra

Page 2: 5. concreción de las pruebas

Una vez establecidas qué pruebas se deben realizar, con los recursos necesarios, pasamos a concretarlas en los apartados siguientes.

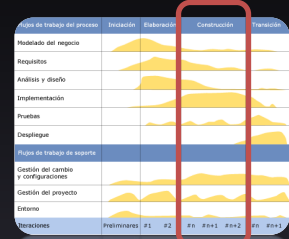
DESCRIPCIÓN	RECURSOS	PRUEBAS	UNIDAD
C1	Implementación de la base de datos	De unidad.	
C2	Implementación de la API para la base de datos.	De unidad.	
C3	Todas las cosas de más referidas a los accesos al sistema.	De unidad, integración y sistema.	
C4	Usari, con usuarios y personal.	De unidad, integración y sistema.	
C5	Usari, con tarjetas, cadenas, usuarios, tipos de acceso y control.	De unidad, integración y sistema.	
C6	Usari, con control, bases de datos.	De unidad, integración y sistema.	
C7	Usari, con protocolos.	De unidad, integración y sistema.	
C8	Usari, con cuentas.	De unidad, integración y sistema.	
C9	Usari, con parámetros de seguridad.	De unidad, integración y sistema.	

Page 3: 4. Recursos

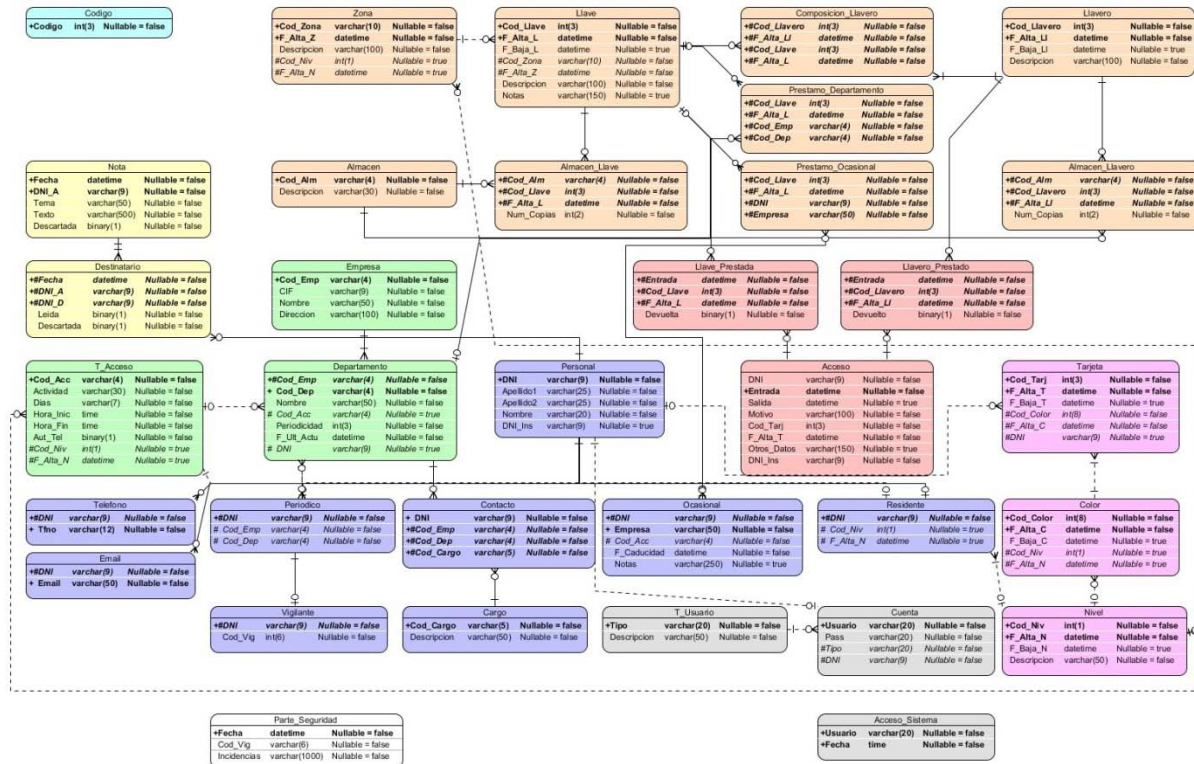
Trabaja con los recursos necesarios para llevar a cabo todo el desarrollo de las pruebas:

RECURSO	CANTIDAD	TIPO DE RECURSO	DESCRIPCIÓN
PC	1	Trabajo	Computadora con la que poder realizar el diseño e implementación de las pruebas.
PC	11	De sistema	Ordenadores para poder realizar las pruebas 'load de todo el desarrollo en una configuración de concurrencia estándar.

Desarrollo del proyecto

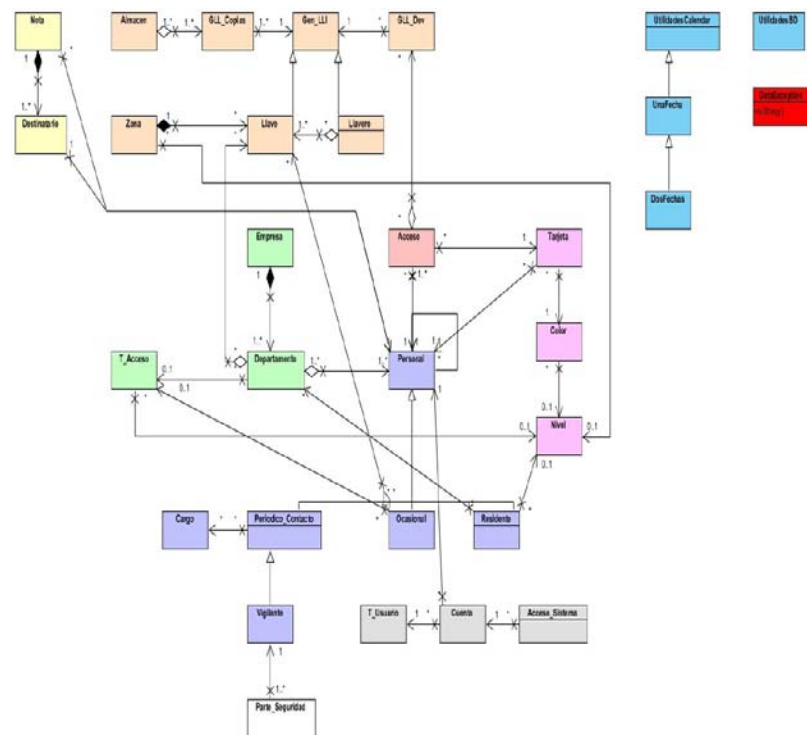
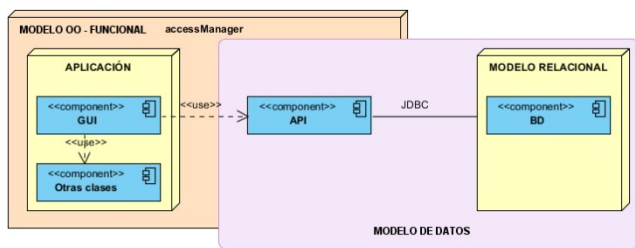


- Construcción – C1: La base de datos



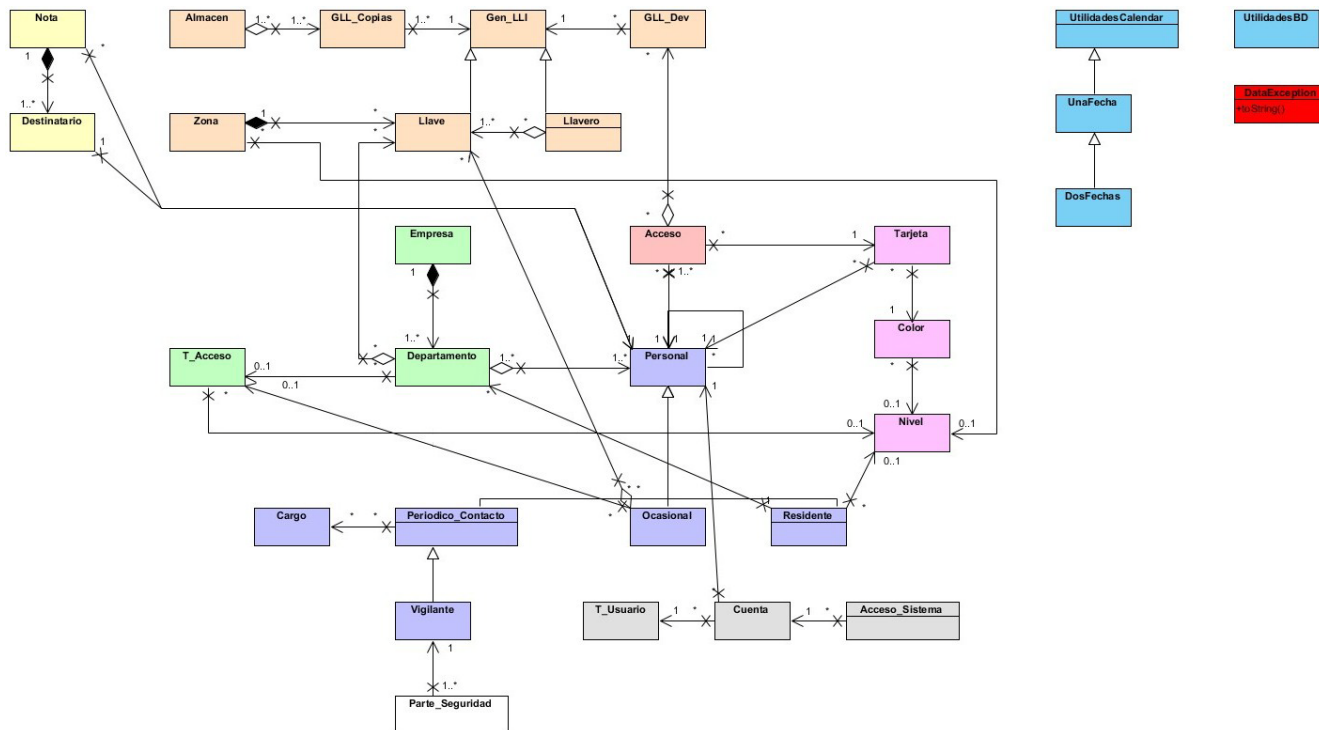
Desarrollo del proyecto

- Construcción – C2: API inicial



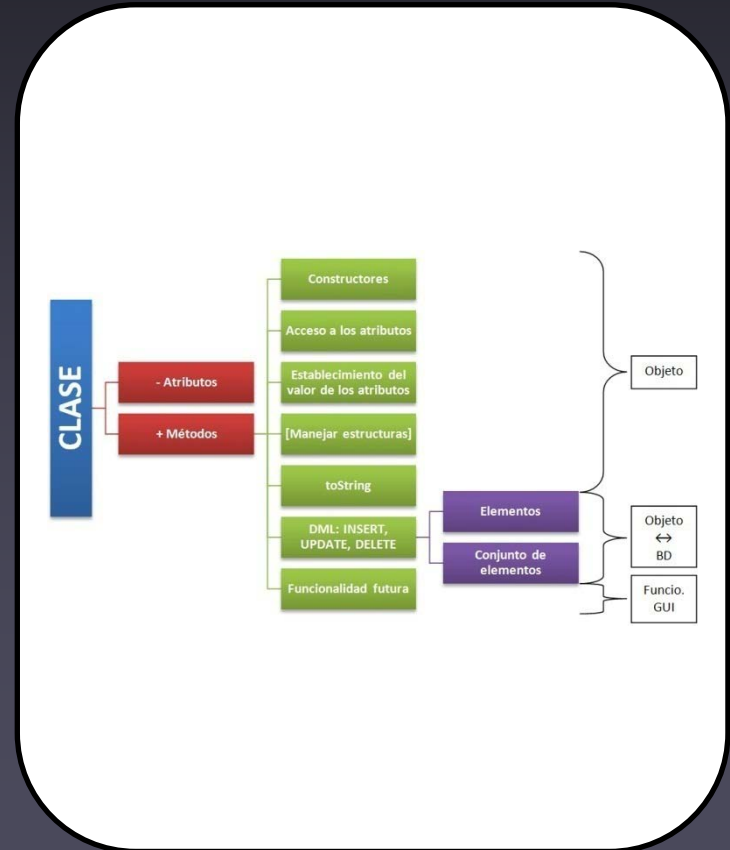
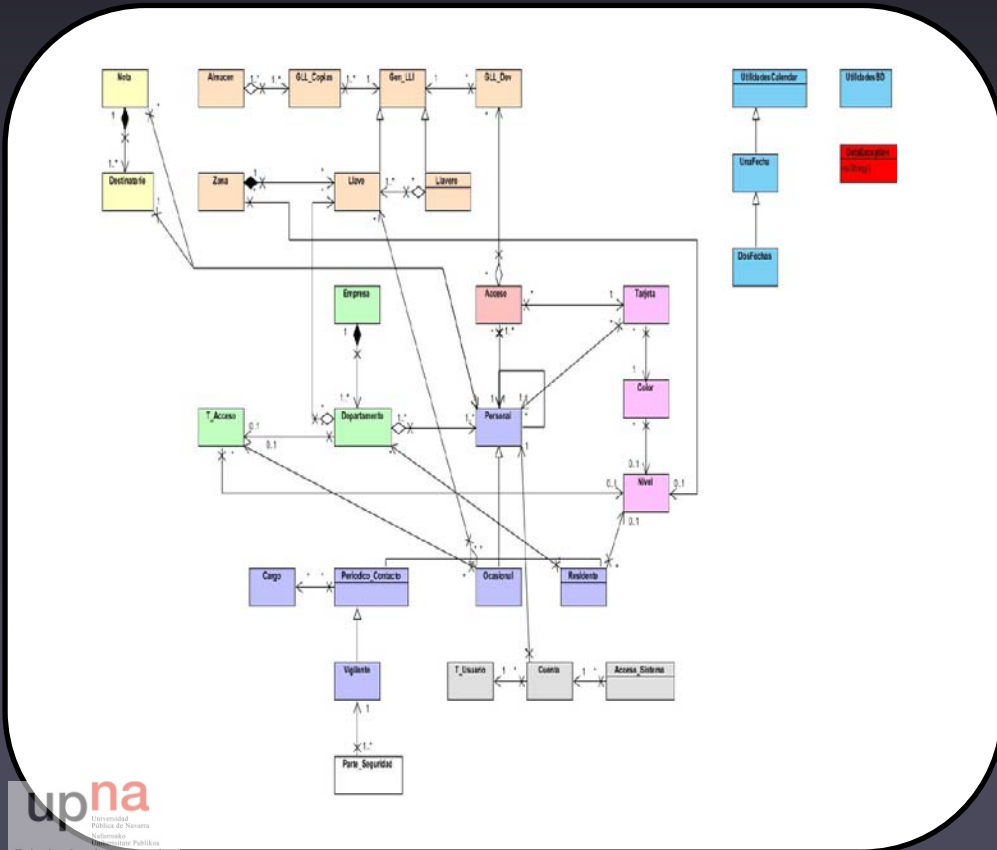
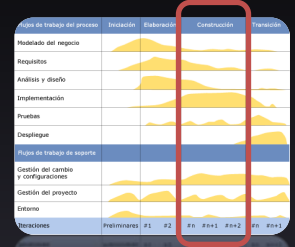
Desarrollo del proyecto

- Construcción – C2: API inicial



Desarrollo del proyecto

- Construcción – C2: API inicial



Desarrollo del proyecto

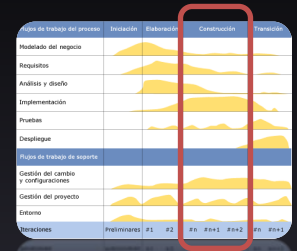
• Construcción – C3: Accesos

○ Análisis

○ Planificación de la iteración

- Objetivos
- Artefactos
- Calendarización detallada
- Recursos
- Casos de uso y escenarios implementados

○ Análisis de los casos de uso y escenarios



Software Solutions

Actividad	Inicio	Fin
Análisis	23/02/10	23/02/10
Calendario	23/02/10	23/02/10
Pruebas	24/02/10	24/02/10
Calendario	24/02/10	24/02/10

Fecha de entrega: 01/01/10

Actividad principal: Administrar y vigilar

Pre-condiciones: El usuario se encuentra logueado en el sistema y desea consultar los accesos realizados al centro.

Acciones principales:

- El administrador o vigilante desea consultar los accesos realizados en el centro en el día en que se encuentra en el centro.
- El sistema comprueba que las fechas programadas son correctas y realiza la búsqueda de accesos según los parámetros establecidos, mostrando el resultado a través de una tabla.

Post-condiciones: El usuario debe consultar los accesos realizados entre días y horas específicas.

Alternativas (A):

- Modificar los parámetros que definen la búsqueda y el punto 2.

Post-condiciones: El usuario dispone para consultar los accesos realizados al centro entre los días y horas indicados.

Caso de uso 3: Visualizar accesos

Descripción: Analizar el acceso puede ser los datos característicos de todos aquellos accesos que concierne con los parámetros establecidos, también debe poder visualizar todos los datos del acceso que desea.

Acciones principales: Administrar y vigilar

Pre-condiciones: El usuario se encuentra logueado en el sistema y desea consultar todos los datos de uno de los accesos que muestra en pantalla.

Acciones principales:

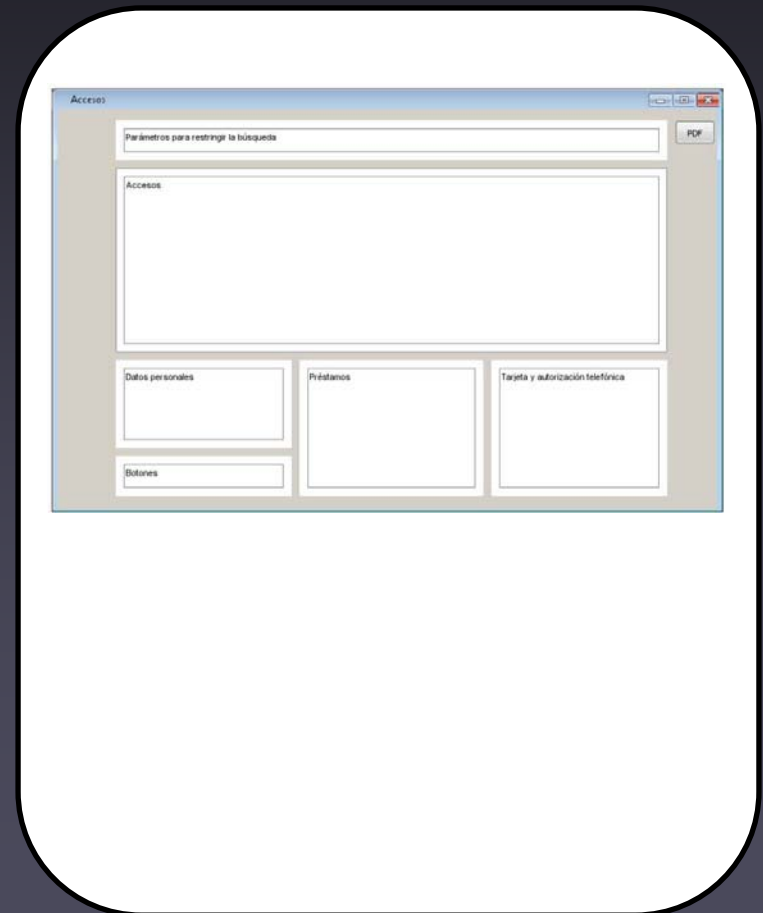
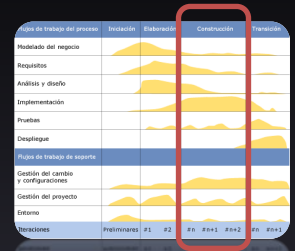
- El usuario selecciona un acceso en pantalla.
- El sistema se encarga de mostrar todos los datos referentes a ese acceso y mostrarlos.

Post-condiciones: El usuario dispone para consultar todos los datos pertenecientes al acceso seleccionado.

SoftwareManager - Iteración C3: Accesos - Página 7 de 55

Desarrollo del proyecto

- **Construcción – C3: Accesos**
 - **Diseño (CdU a CdU)**
 - Interfaz gráfica necesaria.

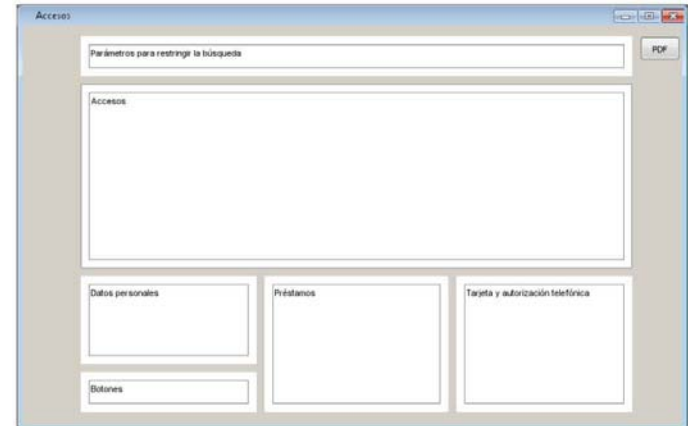
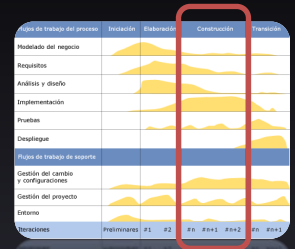


Desarrollo del proyecto

- **Construcción – C3: Accesos**

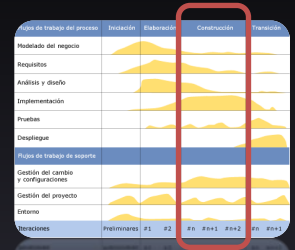
- **Diseño (CdU a CdU)**

- Interfaz gráfica necesaria.
 - Estrategias / procesos a seguir.



1. Se genera un JFileChooser que permita indicar al usuario la ruta donde crear el reporte.
 - a. Si el usuario elige una ruta
 - Creamos un objeto Generadorinformes con la ruta en la que debe dejar el informe. Le pedimos que lo genere con el título y texto indicados y que extraiga toda la información de la tabla suministrada.
 - El objeto Generadorinformes realiza su tarea de forma satisfactoria y el caso de uso finaliza.

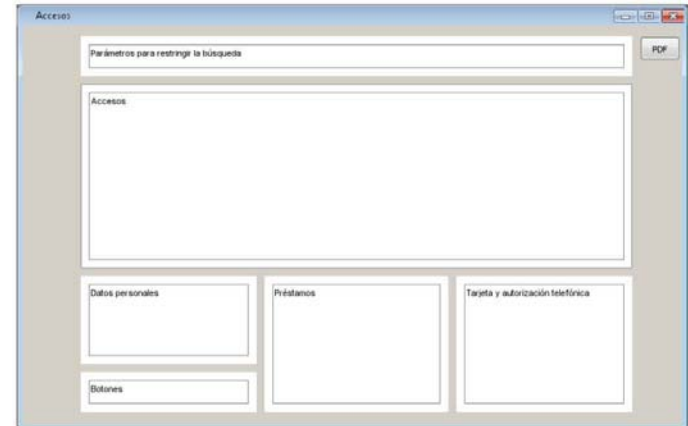
Desarrollo del proyecto



- **Construcción – C3: Accesos**

- **Diseño (CdU a CdU)**

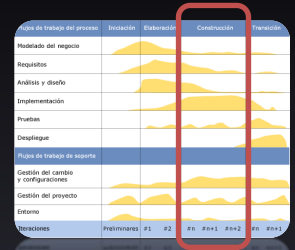
- Interfaz gráfica necesaria.
 - Estrategias / procesos a seguir.
 - Refinamiento de la API inicial.



1. Se genera un JFileChooser que permita indicar al usuario la ruta donde crear el reporte.
 - a. Si el usuario elige una ruta
 - Creamos un objeto `GeneradorInformes` con la ruta en la que debe dejar el informe. Le pedimos que lo genere con el título y texto indicados y que extraiga toda la información de la tabla suministrada.
 - El objeto `GeneradorInformes` realiza su tarea de forma satisfactoria y el caso de uso finaliza.

MÉTODO	FUNCIONALIDAD
<code>GeneradorInformes</code>	Constructor que crea un objeto <code>GeneradorInformes</code> tomando como parámetro de entrada la ruta en la que queremos que los almacene.
<code>generarInforme</code>	Método que genera un informe tomando como datos de entrada el título, texto y tabla a reportar.

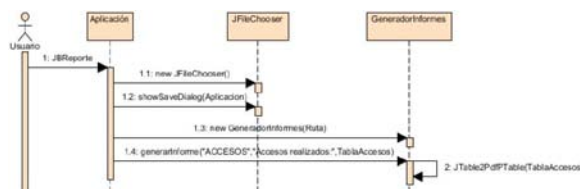
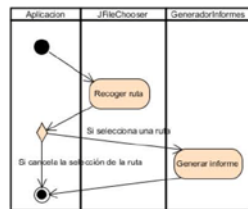
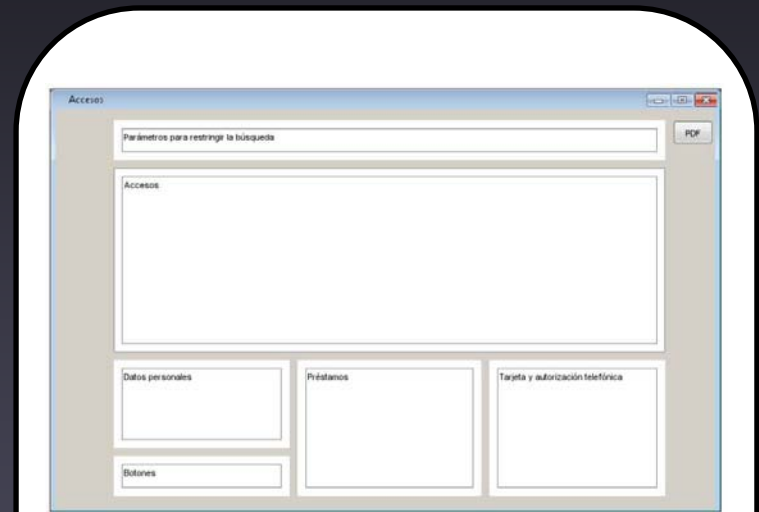
Desarrollo del proyecto



- **Construcción – C3: Accesos**

- **Diseño (CdU a CdU)**

- Interfaz gráfica necesaria.
 - Estrategias / procesos a seguir.
 - Refinamiento de la API inicial.
 - Diagramas de actividades y de secuencia.

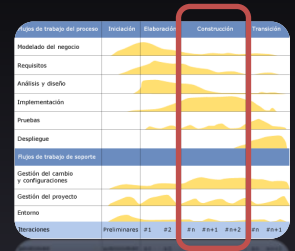


1. Se genera un JFileChooser que permita indicar al usuario la ruta donde crear el reporte.
 - a. Si el usuario elige una ruta
 - Creamos un objeto GeneradorInformes con la ruta en la que debe dejar el informe. Le pedimos que lo genere con el título y texto indicados y que extraiga toda la información de la tabla suministrada.
 - El objeto GeneradorInformes realiza su tarea de forma satisfactoria y el caso de uso finaliza.

MÉTODO	FUNCIONALIDAD
<i>GeneradorInformes</i>	Constructor que crea un objeto GeneradorInformes tomando como parámetro de entrada la ruta en la que queremos que los almacene.
<i>generarInforme</i>	Método que genera un informe tomando como datos de entrada el título, texto y tabla a reportar.

Desarrollo del proyecto

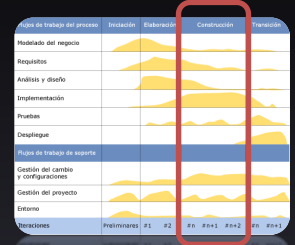
- Construcción – C3: Accesos
 - Codificación e integración



The screenshot displays the 'accessManager' web application interface. The main area is titled 'Accesos' and shows configuration options for access. The date is set to 'martes 18 de mayo de 2010' and the time is '00:00'. There are radio buttons for 'Entrada' (selected) and 'Salida'. A search box is present with the text 'Buscar...'. Below this is a table with columns for 'DNI', 'Entrada', 'Salida', and 'Motivo'. The table is currently empty. To the left is a sidebar menu with options: 'Empresas y personal', 'Tarjs., cols., nivs. y tipos de acc.', 'Zonas, llaves y llaveros', 'Préstamos', 'Cuentas', 'Partes de seguridad', and 'Notas'. Below the table are several form sections: 'Datos' (with fields for DNI, Primer apellido, Segundo apellido, Nombre, Motivo, and Otros datos), 'Llaves y llaveros' (with a table for Tipo, Código, and Devuelta), 'Tarjeta' (with a field for Devuelta), and 'Aviso telefónico' (with a field for Teléfonos). Navigation buttons (+, -, →, ←, →+) are located at the bottom of the form sections.

Desarrollo del proyecto

- **Construcción – C3: Accesos**
 - Pruebas (Unidad + Integración + Sistema).
 - Evaluación de la iteración.
 - Despliegue con el cliente.



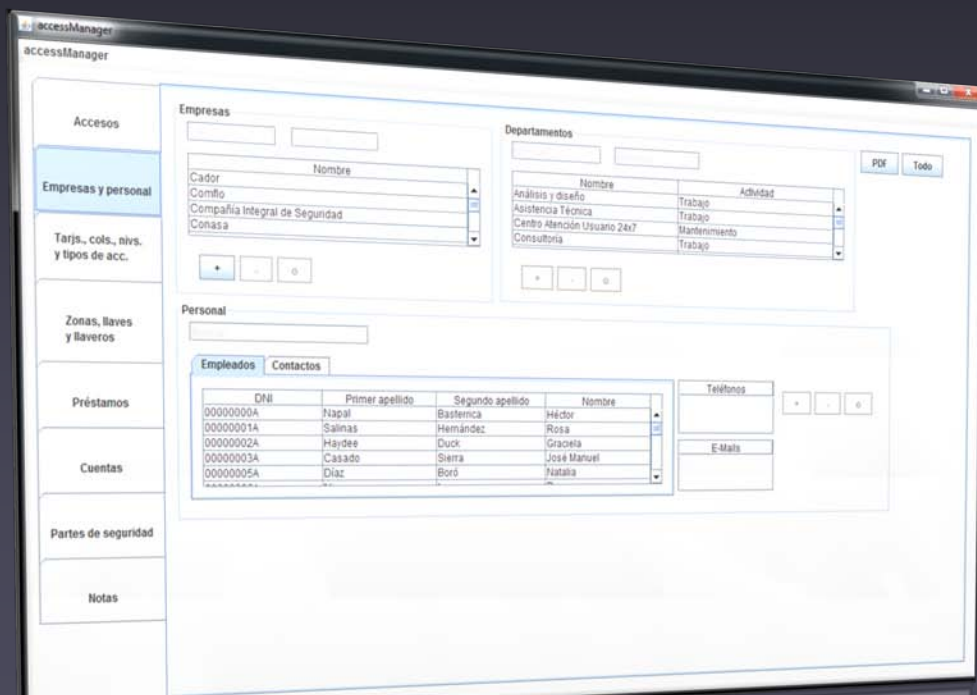
Caso de uso 11. Listar accesos
Objetivo: Verificar que la aplicación permita listar los accesos visualizados en el estado de la tabla de accesos.
Actividades: Analizar la aplicación y probar que se generen listados con la tabla de accesos, roles y datos.
Herramientas: Aplicación accessManager, Base de datos con población de prueba.
Datos de entrada: Tabla de accesos roles, Tabla de accesos roles.
Salida esperada: En el primer caso un informe que indique que los datos y en el segundo un informe con todos los datos de la tabla.
Salida observada: En el caso segundo caso.

Caso de uso 12. Dar acceso automático
Objetivo: Comprobar que el sistema realice el mantenimiento de la tabla Accesos en el día, otorgando roles a todos aquellos accesos realizados en el día anterior a que no dispongan de ellos.
Actividades: Introducir tres accesos del día anterior con fecha de caducidad, borrar la población de la tabla que recoge el historial de accesos al sistema (roles, sistema) e intentar como regular, verificando que se le asigna roles a los accesos del día.
Herramientas: Aplicación accessManager, Base de datos con población de prueba, Unidad gráfica.
Datos de entrada: Accesos de prueba, uno por cada tipo de persona.
Salida esperada: Los accesos mostrados deben tener las siguientes horas de caducidad: 23:00:00, 23:00:00, 23:59:59.
Salida observada: [Table with columns: Fecha, Hora, Persona, Rol, Estado, Fecha Caducidad].
Evaluación/ conclusión: El caso de uso '12. Dar acceso automático' ha sido realizado satisfactoriamente.

E.8. De sistema
Objetivo: Llegado al final de la iteración debemos hacer una breve recapitulación, comprobando que hemos implementado todos los casos de uso que nos propusimos al inicio de la misma, que todo funcione correctamente y que la documentación que hemos elaborado es la correcta.
7. Evaluación de la iteración: Contribuir a la parte de desarrollo correspondiente a la iteración 'C3 - Accesos', dando por desarrollado de forma preliminar los casos de uso en los que el manejo de accesos se refiere. Creemos que hemos realizado un buen trabajo, creando una interfaz sencilla, que recoge toda la información que desea el cliente y proporciona una correcta funcionalidad a tener en los casos de prueba especificados.
8. Despliegue: Con este documento y el software desarrollado hasta el momento nos reunimos con el cliente para mostrarle el trabajo realizado. Tras explicarle el funcionamiento del prototipo y probarlo con varios ejemplos, Pedro Martínez se muestra bastante satisfecho con el diseño, usabilidad y funcionalidad que el software proporciona, dándonos vía libre para proseguir con nuestro trabajo.
Fecha: 20 de febrero de 2010.
Hector Ángel Rodríguez, Responsable de desarrollo.
Pedro Martínez Pérez, Jefe de área de producción de informática Navarra S.A.

Desarrollo del proyecto

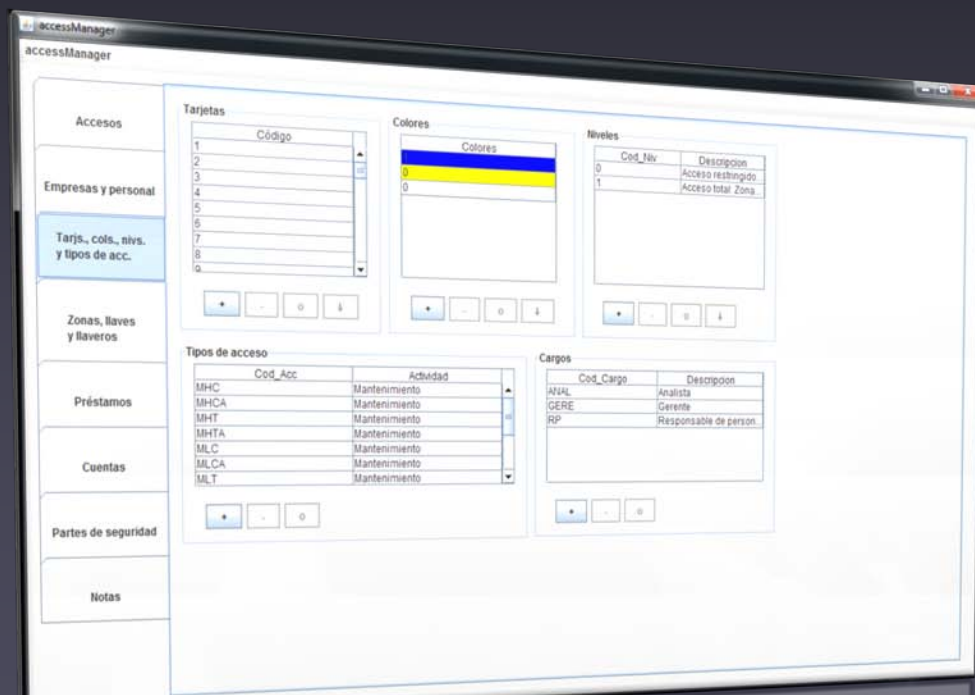
- Construcción – C4: Empresas y personal



- 13 - Visualizar empresas
- 14 - Visualizar empresa
- 15 - Añadir empresa
- 16 - Borrar empresa
- 17 - Modificar empresa
- 18 - Listar empresas
- 19 - Visualizar deps.
- 20 - Visualizar dep.
- 21 - Añadir dep.
- 22 - Borrar dep.
- 23 - Modificar dep.
- 24 - Listar dep.
- 25 - Visualizar personal
- 26 - Visualizar persona
- 27 - Añadir persona
- 28 - Borrar persona
- 29 - Modificar persona
- 30 - Añadir teléfono
- 31 - Borrar teléfono
- 32 - Añadir E-Mail
- 33 - Borrar E-Mail
- 34 - Añadir cargo
- 35 - Borrar cargo
- 36 - Listar personal

Desarrollo del proyecto

- Construcción – C5: Tarjetas, tipos de acceso y cargos

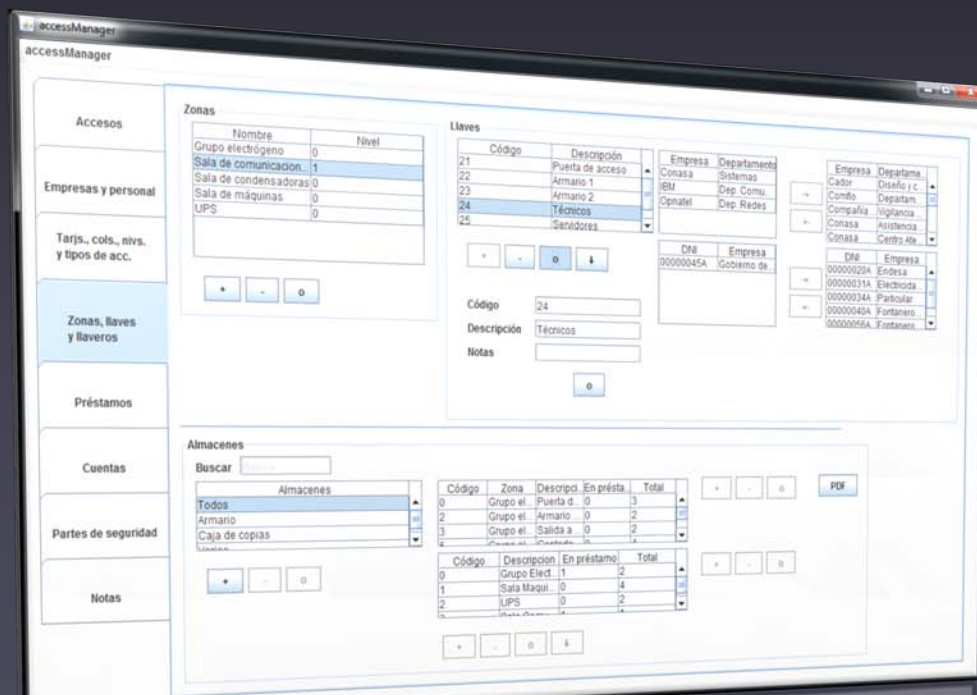


- 37 - Visualizar niveles
- 38 - Visualizar nivel
- 39 - Añadir nivel
- 40 - Borrar nivel
- 41 - Modificar nivel
- 42 - Dar de baja nivel
- 43 - Visualizar colores
- 44 - Visualizar color
- 45 - Añadir color
- 46 - Borrar color
- 47 - Modificar color
- 48 - Dar de baja color
- 49 - Visualizar tarjetas
- 50 - Visualizar tarjeta

- 51 - Añadir tarjeta
- 52 - Borrar tarjeta
- 53 - Modificar tarjeta
- 54 - Dar de baja tarjeta
- 55 - Visu. tipos de acceso
- 56 - Visu. tipo de acceso
- 57 - Añadir tipo de acceso
- 58 - Borrar tipo de acceso
- 59 - Modif. tipo de acceso
- 60 - Visualizar cargos
- 61 - Visualizar cargo
- 62 - Añadir cargo
- 63 - Borrar cargo
- 64 - Modificar cargo

Desarrollo del proyecto

- Construcción – C6: Zonas, llaves y llaveros

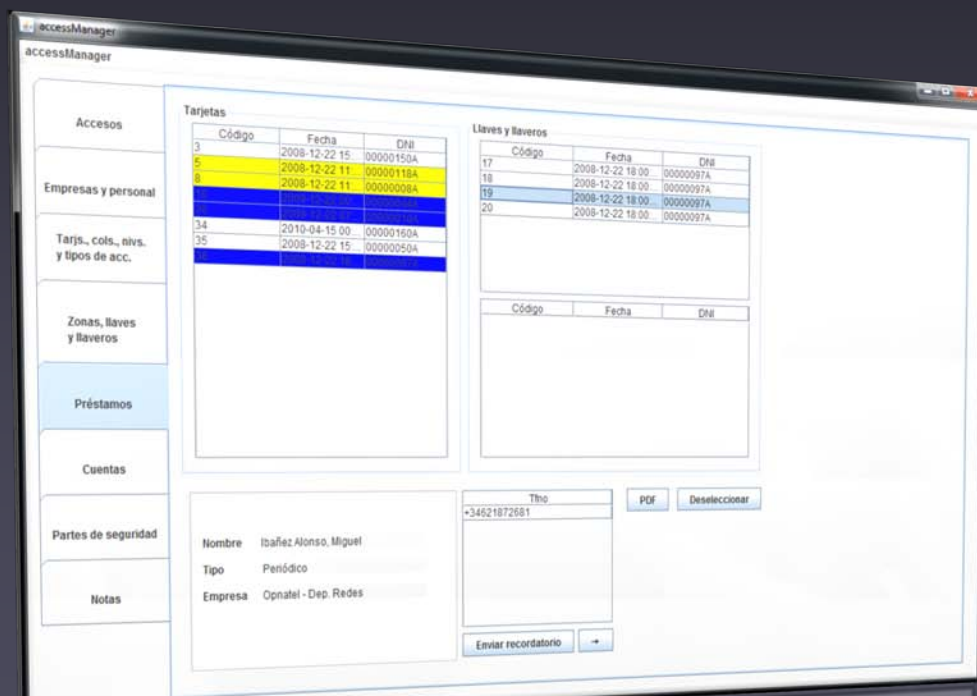


- 65 - Visuali. zonas
- 66 - Visualizar zona
- 67 - Añadir zona
- 68 - Borrar zona
- 69 - Modificar zona
- 70 - Visualizar llaves
- 71 - Visualizar llave
- 72 - Añadir llave
- 73 - Borrar llave
- 74 - Dar de baja llave
- 75 - Modificar llave
- 76 - Listar llaves
- 77 - Visualizar llaveros
- 78 - Visualizar llavero
- 79 - Añadir llavero
- 80 - Borrar llavero
- 81 - Dar de baja llavero

- 82 - Modificar llavero
- 83 - Añadir llave
- 84 - Borrar llave
- 85 - Listar llaveros
- 86 - Visu. alms. de llaves
- 87 - Visu. alm. de llaves
- 88 - Añadir alm. de llaves
- 89 - Borrar alm. de llaves
- 90 - Modif. alm. de llaves
- 91 - Añadir llave
- 92 - Borrar llave
- 93 - Modificar copias llave
- 94 - Añadir llavero
- 95 - Borrar llavero
- 96 - Modif. copias llavero
- 97 - Listar alms. de llaves

Desarrollo del proyecto

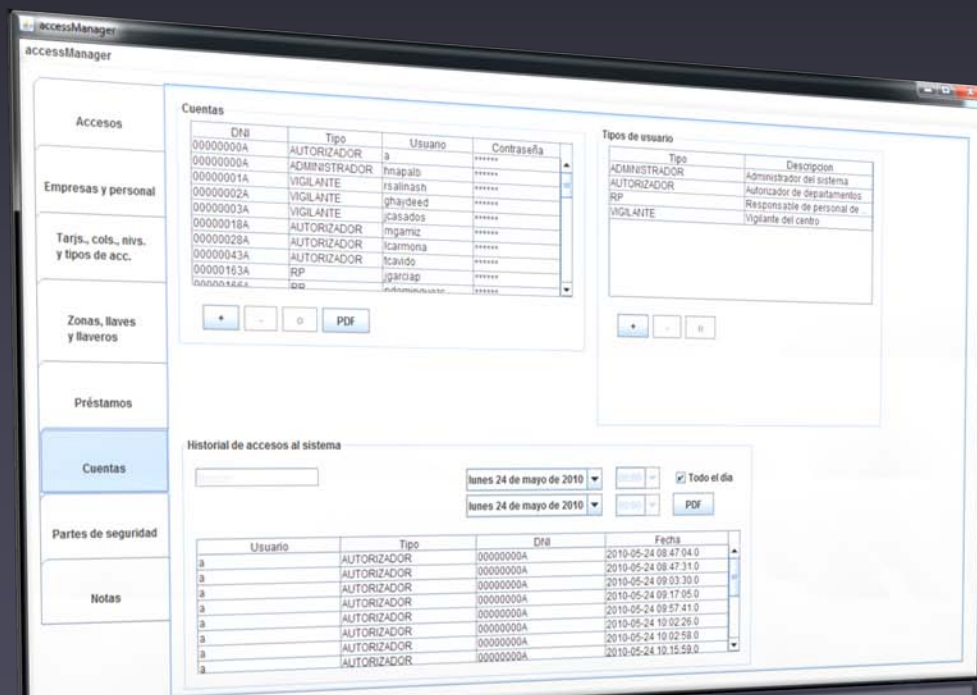
- Construcción – C7: Préstamos



- 98 - Visualizar préstamos
- 99 - Visualizar datos del prestatario
- 100 - Devolver préstamo
- 101 - Enviar recordatorio vía E-Mail
- 102 - Listar préstamos pendientes

Desarrollo del proyecto

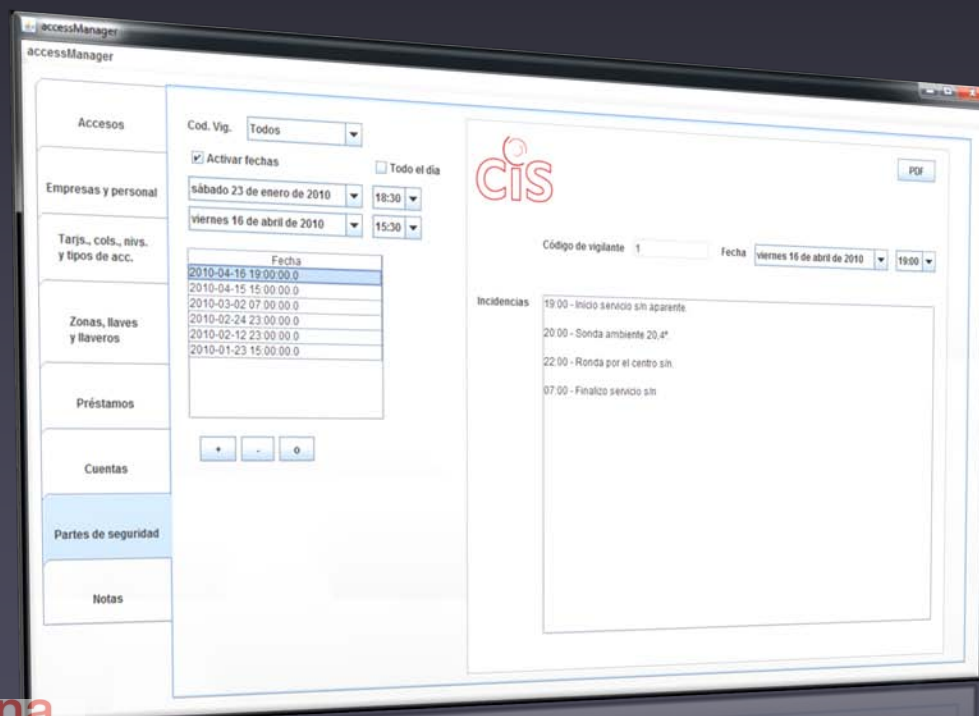
- Construcción – C8: Cuentas



- 103 - Visualizar cuentas
- 104 - Visualizar cuenta
- 105 - Añadir cuenta
- 106 - Borrar cuenta
- 107 - Modificar cuenta
- 108 - Listar cuentas
- 109 - Visualizar tipos de usuario
- 110 - Visualizar tipo de usuario
- 111 - Añadir tipo de usuario
- 112 - Borrar tipo de usuario
- 113 - Modificar tipo de usuario
- 114 - Visualizar accesos al sistema
- 115 - Listar accesos al sistema
- 116 - Modificar datos de cuenta

Desarrollo del proyecto

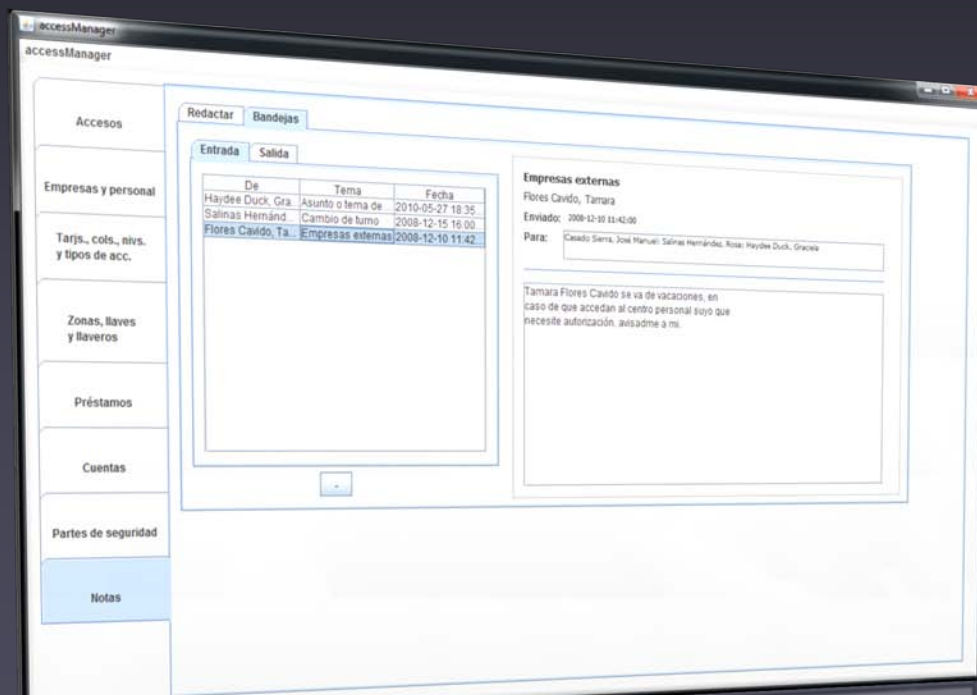
- Construcción – C9: Partes de seguridad



- 117 - Recordar personal a punto de caducar
- 118 - Recordar préstamos pendientes
- 119 - Borrar datos obsoletos
- 120 - Visualizar partes de seguridad
- 121 - Visualizar parte de seguridad
- 122 - Añadir parte de seguridad
- 123 - Borrar parte de seguridad
- 124 - Modificar parte de seguridad
- 125 - Listar parte de seguridad

Desarrollo del proyecto

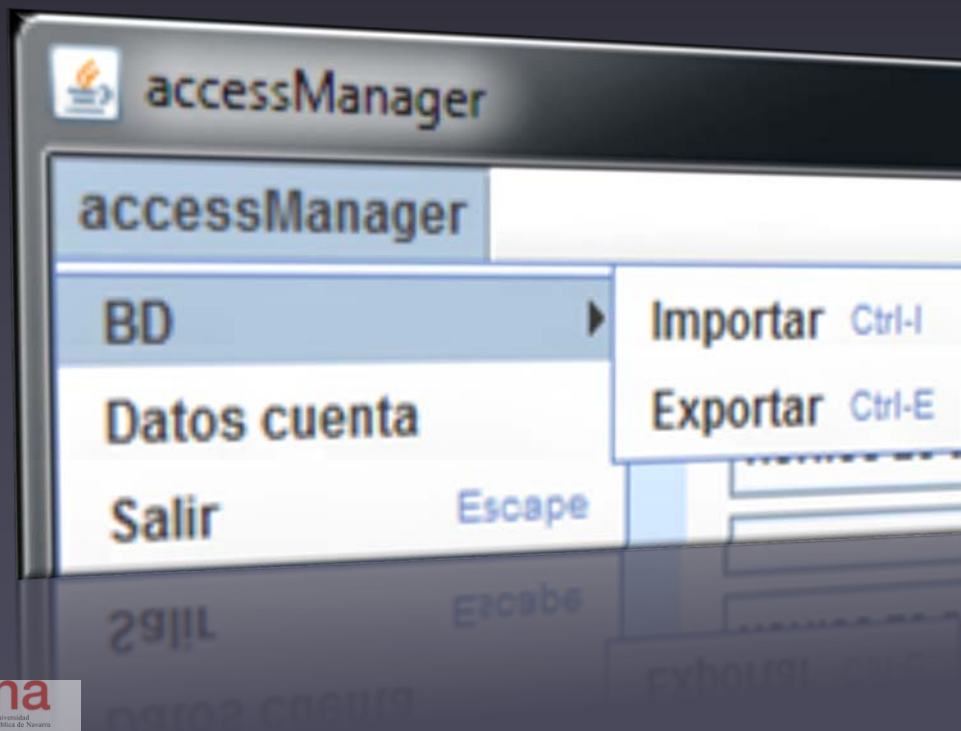
- Construcción – C10: Notas



- 126 - Redactar nota
- 127 - Visualiza notas recibidas
- 128 - Visualizar nota recibida
- 129 - Borrar nota recibida
- 130 - Visualizar notas enviadas
- 131 - Visualizar nota enviada
- 132 - Eliminar nota enviada

Desarrollo del proyecto

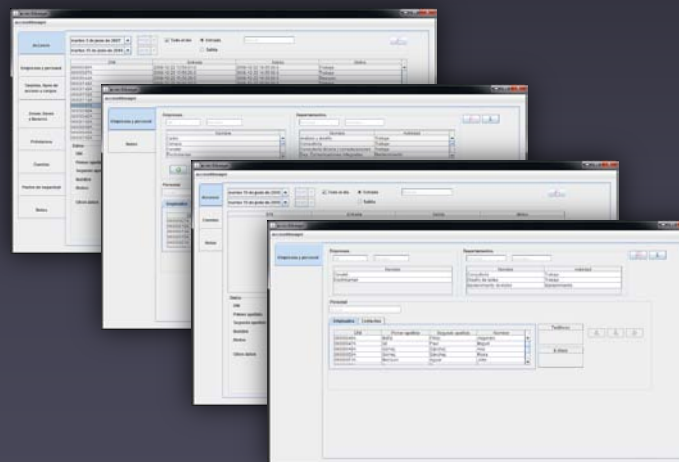
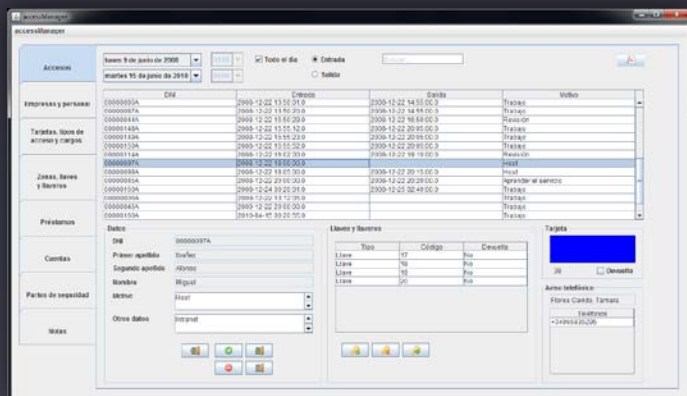
- Construcción – C11: Base de datos



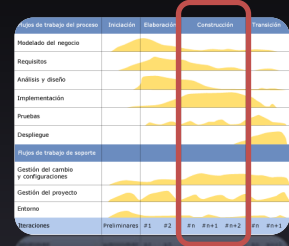
133 - Importar base de datos
134 - Exportar base de datos

Desarrollo del proyecto

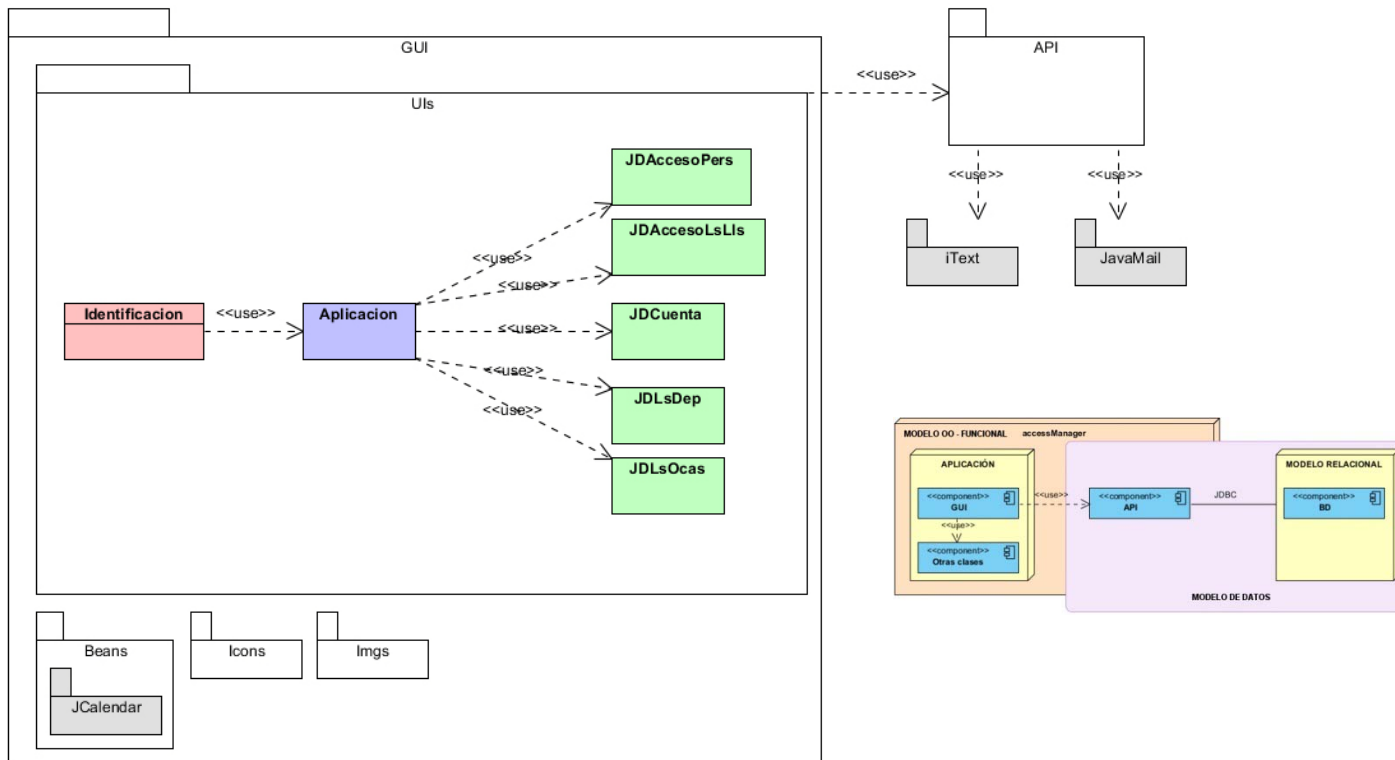
- Construcción – C12: Diseño, personalización y configuración



Desarrollo del proyecto



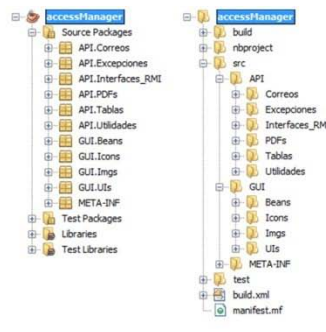
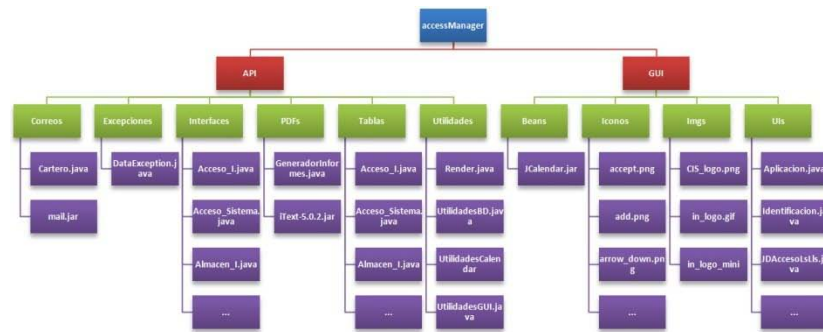
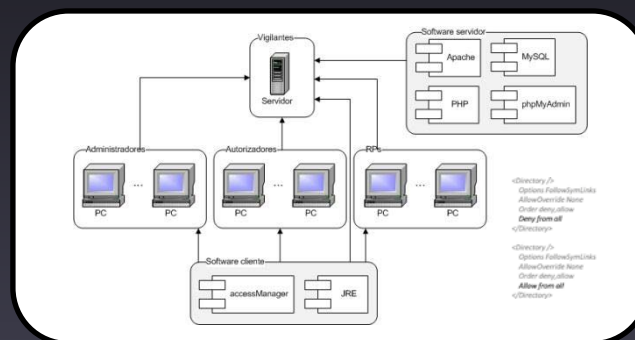
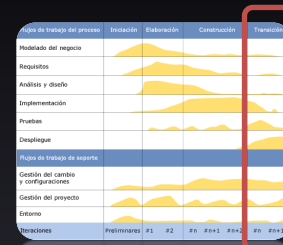
- Construcción – Estructura de la aplic.



Desarrollo del proyecto

- **Transición**

- T1: Preparación de la release final.
- T2: Entrega.
- T3: Entrenamiento, soporte y mantenimiento.



Código fuente + JAR + ...

Doc. + Javadoc + ...

Veremos

- **Ámbito y empresa acometida**
- Elección del proceso de desarrollo
- Desarrollo del proyecto
- **Demostraciones**
- Ruegos y preguntas

Veremos

- Ámbito y empresa acometida
- Elección del proceso de desarrollo
- Desarrollo del proyecto
- Demostraciones
- Ruegos y preguntas

Veremos

- Ámbito y empresa acometida
- Elección del proceso de desarrollo
- Desarrollo del proyecto
- Demostraciones
- Ruegos y preguntas

