



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO EN INFORMÁTICA

Título del proyecto:

“DESARROLLO DE UNA APLICACIÓN WEB PARA LA
GESTIÓN DE CURSOS DE FORMACIÓN”

Usue Napal Martínez

Tutor: David Benito

Pamplona, Julio de 2011

ÍNDICE

1. INTRODUCCIÓN	7
2. OBJETIVOS	9
3. CONFIGURACIÓN INICIAL.....	11
3.1 Instalación de Java	11
3.2 Instalación de Tomcat	12
3.3 Instalación de MySQL	13
3.4 Instalación y configuración de Eclipse.....	14
3.5 Instalación de Postfix	17
4. ANÁLISIS.....	19
4.1 Requisitos funcionales.....	19
4.2 Requisitos no funcionales.....	24
4.3 Diagramas de casos de uso	26
Iteración 0	26
Iteración 1	27
Iteración 2	28
Iteración 3	30
Iteración 4	36
5. DISEÑO	39
5.1 Diagramas de clases y diagramas de secuencia.....	39
Iteración 1	39
Iteración 2	43
Iteración 3	45
Iteración 4	63
5.2 Identificación de entidades a definir	66
5.3 Diagrama Entidad/Relación	69
6. DESARROLLO E IMPLEMENTACIÓN	71
6.1 Aplicación basada en Servlets	71
6.2 Uso de formularios para la creación de cursos.....	73
6.3 Acceso mediante registro de usuarios	74
6.4 Uso de sesiones para cada conexión.	76
6.5 Creación de correo interno.	77
6.6 Envío de mails de confirmación e información a los usuarios participantes.	79
6.7 Recuperación de contraseña	80
6.8 Administración.	81
6.9 Sistema de ficheros propio.	82
6.10 Internacionalización del lenguaje de la aplicación.....	83
7. CONCLUSIONES	85
8. BIBLIOGRAFÍA.....	87

9. ANEXO: MANUAL DEL USUARIO.....	89
9.1 CONSULTA OFERTA DOCENTE.....	89
9.2 REGISTRO.....	91
9.3 ACCESO.....	92
9.3.1 MIS CURSOS.....	93
9.3.2 MIS INSCRIPCIONES.....	94
9.3.3 BUSCAR.....	95
9.3.4 CORREO INTERNO.....	98
9.3.5 MI CUENTA.....	100
9.3.6 AGENDA.....	101
9.3.7 VER APUNTADOS.....	102
9.3.8 EDITAR CURSO.....	103
9.3.9 ESPACIO CURSO.....	104
9.3.10 CREAR CURSO.....	106
9.3.11 RECUPERAR CONTRASEÑA.....	107
9.4 ADMINISTRACIÓN.....	108
9.4.1 MANTENIMIENTO.....	109
9.4.2 ESTADÍSTICAS.....	112

1. INTRODUCCIÓN

Hoy en día, las nuevas tecnologías se encuentran en auge, y cada vez su uso es mayor, más extenso, y abarca toda clase de perfiles entre sus usuarios. Más concretamente, el crecimiento en la aparición y el uso de las redes sociales es cada vez más notorio. Éstas han pasado a convertirse en parte esencial de la vida de los usuarios, que generalmente, son parte de las generaciones más jóvenes.

Este creciente uso, obviamente, tiene sus pros y sus contras, pero claramente, son tecnologías fuertemente arraigadas en el día a día de nuestras vidas. Por lo tanto, debido a la incidencia de éstas, sobre todo en las personas más jóvenes, sería correcto poder utilizarlas para fines no sólo dedicados al ocio, si no también a la educación y culturización de la sociedad.

De todo esto, las personas más jóvenes, serán precisamente las que puedan sacar mayor partido a sus posibilidades. Así pues, la idea principal de este proyecto parte de estos dos puntos: la educación, y las nuevas tecnologías.

El objetivo principal será mezclar ambos conceptos, logrando así desarrollar una aplicación Web, de carácter dinámico, y dirigida a la creación y gestión de la docencia. Esta aplicación podrá ser usada dentro del ámbito universitario o escolar, para la gestión de cursos educativos que puedan impartirse en estas comunidades, tanto para el profesorado como para el alumnado.

El hecho de que cada persona pueda gestionar sus propios cursos, hace que la comunidad educativa generada no sea centralizada, si no que cada usuario pueda aportar su parte al sistema, así como hacer uso de la docencia impartida por el resto de miembros de la comunidad.

De esta forma, aprovechando la interacción y funcionamiento de las redes sociales, bien conocidas por sus usuarios, se logrará obtener un mayor partido de este proyecto, haciendo que el enfoque educativo sea más atractivo para sus usuarios. Todo esto, será facilitado mediante un interfaz sencillo, intuitivo y de fácil manejo para todo tipo de usuarios, independientemente del nivel de conocimientos tecnológicos del mismo.

Las características más importantes de esta aplicación se resumen a continuación.

En primer lugar, la aplicación se basará en facilitar la gestión de la docencia por medio de una 'comunidad' Web. Su acceso será controlado por medio de autenticación, donde sólo será permitido el acceso a aquellos usuarios registrados. Además, cada usuario tendrá opción de editar su propia información privada.

Así mismo, cada usuario tendrá la posibilidad de crear una serie de cursos, los cuales impartirá y gestionará él mismo. Todo esto, estará basado en formularios Web, donde se introducirán los datos más relevantes de dichos cursos, como lugar, una breve descripción, horario, plazas disponibles, etc.

Por otro lado, y también como usuario registrado, es posible inscribirse a aquellos cursos que sean de interés personal para cada usuario. Cualquier cambio que se produzca en dichas inscripciones será notificado al usuario vía email.

La aplicación, además, permitirá llevar un seguimiento de aquellas personas que se inscriban en la docencia de cada usuario, permitiendo incluso la comunicación directa con ellos por medio del correo interno de la aplicación.

Finalmente, cada curso creado dispondrá de un apartado propio para almacenar información acerca de la docencia, que pueda servir de ayuda a sus asistentes. Desde la exposición de contenido del curso, hasta una zona docente común a todos aquellos usuarios inscritos en dicho curso, que permita mejorar la información existente así como añadir nuevos datos, opiniones, etc. de la misma, tanto por parte del docente como por parte de los asistentes.

2. OBJETIVOS

Como ya se ha explicado en el punto anterior, se pretende construir una aplicación Web dedicada a la docencia particular de los usuarios registrados en ella. Si cada usuario puede impartir una serie de cursos al resto de usuarios pertenecientes a la 'comunidad', se logra obtener un sistema de docencia variada y no centralizada, donde cada persona tiene la oportunidad de gestionar sus propios cursos.

Por lo tanto el objetivo principal que se persigue con este proyecto es el de conseguir desarrollar una aplicación que permita esta serie de requisitos. Para ello se programará en lenguaje java mediante el programa destinado al desarrollo de aplicaciones 'eclipse'.

Así pues, si el objetivo a conseguir implica el uso del lenguaje java y así mismo, la herramienta de desarrollo eclipse, existirán también una serie de objetivos implícitos, además de los objetivos que se pretende lograr con la creación de la aplicación en sí. Uno de ellos será el correcto manejo de ambos, pretendiendo tener un dominio generalizado de ambas herramientas, que permita desarrollar la aplicación de forma sencilla.

Sin embargo, para implementar la aplicación es necesario preparar y configurar correctamente un servidor donde ésta pueda funcionar. Es decir, antes de implementar la aplicación será necesario crear y configurar un servidor donde estará corriendo la aplicación docente. Así pues, otro de los objetivos principales de este proyecto será éste, configurar un servidor para el correcto funcionamiento de la aplicación Web.

Por último, podrá desarrollarse el código de dicha aplicación. No sin antes realizar un estudio de los requisitos que especifica la herramienta, así como un análisis del proyecto que va a realizarse. Una vez se tengan estos pasos completados, se pasará a realizar un diseño del prototipo, que finalmente se implementará para lograr obtener una aplicación que cumpla todos estos requisitos.

Una vez que la aplicación Web esté implementada se deberá probar su funcionamiento. En caso de existir errores se corregirán y se harán las pruebas necesarias hasta que se verifique su total y correcto funcionamiento. En el momento que la aplicación esté instalada y funcionando correctamente podrá darse por finalizado el proyecto.

Así pues, podemos concluir que los objetivos básicos de este proyecto son 3:

- 1) Manejo adecuado del lenguaje java y la herramienta eclipse.
- 2) Configuración de un servidor para una aplicación Web dinámica.
- 3) Análisis, diseño e implementación de la aplicación Web en sí (objetivos propios del sistema).

- Características principales de la aplicación:

- Acceso mediante registro de usuarios.
- Posibilidad de personalización de los datos de perfil de usuario.
- Uso de sesiones para cada conexión.
- Creación de correo interno.
- Envío de mails de confirmación e información a los usuarios participantes.
- Uso de formularios para la creación de cursos.
- Posibilidad de inscripción de usuarios a cursos docentes.

- Búsqueda de cursos por parte de los usuarios.
- Información de las inscripciones de otros usuarios a cursos creados por el usuario.
- Información de las inscripciones de un usuario a cursos docentes.
- Internacionalización del lenguaje de la aplicación.

Por tanto, a lo largo de este proyecto, se irán explicando las diferentes partes del mismo, es decir, los objetivos descritos en el párrafo anterior.

En primer lugar se centrará en la configuración de servidor que permita el funcionamiento de la aplicación. Seguidamente, se describirá el esquema de la aplicación Web para la docencia, cuál será su estructura y su funcionamiento.

Y finalmente se llevará a cabo el desarrollo del código fuente de la aplicación en sí. Así como el seguimiento del funcionamiento mediante una serie de pruebas y corrección de fallos para lograr una aplicación completa y de correcto funcionamiento.

Finalmente, y como anexo a todo el proceso, podrá realizarse un manual de usuario, que facilite a éste el aprendizaje de la herramienta y pueda aprender de forma sencilla todas las opciones y ventajas que ésta le proporciona.

3. CONFIGURACIÓN INICIAL

Inicialmente, la aplicación que se va a construir estará instalada y en funcionamiento en una máquina de la Universidad Pública de Navarra, para el uso local e interno de la misma. Aunque el hecho de crear una aplicación Web permita en el futuro a los usuarios descargarla y crearse su propia aplicación local, las configuraciones se centraran en dejar preparada la máquina que será utilizada por la aplicación desde la universidad, siendo extensibles dichas configuraciones a cualquier servidor que posean los usuarios.

En primer lugar, y antes de comenzar con la programación, es necesario realizar una serie de configuraciones para que el ordenador donde se va a trabajar esté preparado para ello.

A continuación se describirán los pasos que hay que seguir para completar correctamente dicha configuración. Estos pasos del desarrollo inicial están basados en la configuración de un PC con sistema operativo Linux (Ubuntu 7.04, en concreto), por lo que si desean seguirse al pie de la letra en otro sistema operativo, es posible que sean necesarios algunos cambios en la ejecución.

En primer lugar, ya que la aplicación estará basada y desarrollada en lenguaje Java, será necesario configurar la máquina para su correcto funcionamiento en este aspecto.

3.1 Instalación de Java

Lo primero será descargar el binario de java desde la Web oficial. A continuación, será necesario convertir el .bin en un .deb para proceder a su instalación. Para que esto pueda llevarse a cabo es necesario instalar a su vez dos aplicaciones, java-package y fakeroot.

Se descargan mediante el comando apt-get:

```
# sudo apt-get install java-package  
# sudo apt-get install fakeroot
```

Una vez instalados ambos, ya es posible convertir el binario a archivo .deb, y proceder a la instalación de java:

```
# fakeroot make-jpkg jdk-1_5_0_18-linux-i586.bin
```

Así se crea el archivo sun-j2sdk<version>.deb, que se ejecutará de la siguiente forma.

```
# sudo dpkg -i sun-j2sdk1.5_1.5.0+update06_i386.deb  
[Nombre del paquete generado en cada caso.]
```

Una vez realizado esto, se moverá el directorio creado a aquel que se desee. En este caso resulta conveniente moverlo al directorio /opt, en cual se dispone de permisos suficientes y será el directorio en el cual se instalen el resto de aplicaciones que se utilizarán en el desarrollo de la aplicación final. Por lo tanto, dentro de /opt se puede crear una carpeta llamada 'java'. Así, el comando a ejecutar es el siguiente.

```
# sudo mv /usr/lib/j2sdk1.5-sun /opt/java/
```

Con esto finaliza la instalación y configuración de java. Ahora se pasará a instalar el servidor Web dinámico Tomcat, que permitirá desplegar la aplicación Web para que sea accesible.

3.2 Instalación de Tomcat

La herramienta que se va a generar es una aplicación Web, que permitirá la creación de cursos docentes por parte del usuario, y la inscripción a dichos cursos por parte del resto de usuarios. Por lo tanto, dicha aplicación estará basada en el desarrollo de servlets, que permitan toda esta serie de operaciones. Para una aplicación Web normal, bastaría con la instalación de apache, que sirviera como portal para ésta. Sin embargo, tratándose de servlets, se pasa a hablar de aplicaciones dinámicas, que el servidor apache no soporta, y por tanto es necesario instalar el servidor tomcat, también de apache. Así pues, a continuación se explicará paso a paso, la instalación y configuración de dicho servidor.

Como siempre, en primer lugar, se descargará de la Web oficial de apache el paquete comprimido de tomcat(tar.gz), versión 5.5.

Una vez descargado, se descomprime y se mueve al directorio donde se están instalando todas las aplicaciones que se van a necesitar. Los comandos para ello son los siguientes.

```
# tar -xvzf apache-tomcat-5.5.27.tar.gz
# mv /apache-tomcat-5.5.27 /opt/
```

A continuación se otorgarán permisos de ejecución a todos los ficheros sh del directorio.

```
# chmod +x /opt/apache-tomcat-5.5.27/bin/*.sh
```

Finalmente se deberá adecuar la configuración de tomcat a las necesidades del proyecto y el lugar de trabajo. Por ello se abrirá el fichero server.xml para su modificación.

```
# gedit /opt/apache-tomcat-5.5.27/conf/server.xml
```

Se debe añadir la frase 'URIEncoding="UTF-8"' al final de la tag <Connector>, quedando finalmente como se especifica a continuación. Los puertos y demás configuraciones se mantienen sin modificaciones.

```
<Connector port="8080" maxHttpHeaderSize="8192"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" redirectPort="8443" acceptCount="100"
connectionTimeout="20000" disableUploadTimeout="true" URIEncoding="UTF-8" />
```

Así queda configurado el servidor tomcat. Sin embargo, hay que definir las variables de entorno necesarias para la ejecución de éste. Por ello, y para mayor comodidad, se crearán unos scripts que permitirán arrancar tomcat de una forma automática.

El primer script se llamará "entorno.sh", y cargará todas las variables de entorno, el directorio de instalación de tomcat, el de java, etc. El segundo cargará este script e iniciará el tomcat, mostrando por pantalla el log de éste, por si se produjese algún fallo a la hora de arrancar el servidor.

El primer script, “entorno.sh”, tendría el siguiente contenido:

```
export JAVA_HOME=/opt/java/j2sdk1.5-sun
export JAVA_OPTS="-server -Xms768m -Xmx768m -XX:PermSize=256m
-XX:MaxPermSize=256m -XX:NewSize=192m -XX:MaxNewSize=384m
-XX:+UseParallelGC -Djava.awt.headless=true -Duser.language=es -Duser.region=ES"
export CATALINA_HOME=/opt/apache-tomcat-5.5.27
export PATH=$PATH:$JAVA_HOME/bin/:$CATALINA_HOME/bin
```

El segundo script, “arr_tomcat.sh”, llamará al script anterior y al script de inicio de tomcat.

```
. $HOME/entorno.sh
$CATALINA_HOME/bin/startup.sh
tail -f $CATALINA_HOME/logs/catalina.out
```

Mediante la ejecución del último comando se está indicando que el registro del log de tomcat, almacenado en el fichero catalina.out, se muestre por pantalla, para observar los detalles del inicio de tomcat.

Para comprobar que funciona correctamente basta con ejecutar el script “arr_tomcat.sh” y desde el explorador Web escribir:

<http://localhost:8080/>

Con esto finaliza la instalación de apache tomcat. A continuación se describen los pasos para instalar MySQL, que servirá para la creación de la base de datos sobre la cual actuará la aplicación. Desde el registro de usuarios hasta el almacenamiento de su información personal, cursos creados e inscripciones realizadas.

3.3 Instalación de MySQL

Para poder manejar los datos de la aplicación será necesario disponer de una base de datos, que se mantendrá desde la misma aplicación. Esta base de datos estará basada en MySQL, que deberá ser instalado en el PC desde el que se esté trabajando para poder tener acceso a ella. Se puede hacer fácilmente mediante la descarga automática de paquetes de apt-get. Se ejecutarán los siguientes comandos:

```
# sudo apt-get install mysql-client
# sudo apt-get install mysql-server
```

A continuación se establece la contraseña para el usuario root.

```
# mysqladmin -u root password "*****"
```

Por último se arranca mysql y se crean la base de datos que se utilizará y los usuarios necesarios, para evitar acceder con permisos de root y que haya algún problema.

```
# mysql -u root -p*****
```

Los siguientes comandos sirven para crear la base de datos, luego crear el usuario *info* y otorgarle los privilegios necesarios para poder manejar la base de datos.

```
mysql> create database InscripcionCSIE default character set utf8;
mysql> grant all on InscripcionCSIE.* to sinfo@'localhost' identified by '*****';
mysql> exit
```

Por último, y para acabar con la instalación de mysql, será necesario descargar el mysql-connector que nos permitirá trabajar conjuntamente con la base de datos y la aplicación desarrollada. Se descargará el paquete comprimido en tar.gz y se descomprimirá. A continuación, se moverá el paquete descomprimido desde el directorio actual hasta el directorio de librerías de tomcat.

```
# tar -xvzf mysql-connector-java-5.0.8.tar.gz
# mv mysql-connector-java-5.0.8-bin.jar /opt/apache-tomcat-5.5.25/shared/lib/.
```

Así queda completada la instalación y configuración de mysql. Más adelante se verá cómo crear las bases de datos necesarias. De forma opcional y para facilitar el manejo de mysql puede instalarse la herramienta phpmyadmin o herramientas similares, que permiten al usuario una interacción gráfica con la base de datos.

3.4 Instalación y configuración de Eclipse

En primer lugar será necesario instalar y configurar el eclipse para poder trabajar con él. Para descargarlo basta con acceder a la Web oficial y descargarse el paquete comprimido con la aplicación.

Una vez se disponga del tar.gz con el código de la aplicación, se moverá a la carpeta donde se quiera instalar. Por ejemplo, la carpeta /opt. la cual otorga todo tipo de permisos al usuario. Si acaso fuera necesario otorgar dichos permisos, por ejemplo, sobre esta carpeta, se ejecutaría el siguiente comando:

```
# sudo chmod 777 /opt
```

Una vez este todo resuelto, desde el directorio donde está situado el paquete con eclipse, se ejecutará el comando siguiente, para descomprimirlo.

```
# tar -xvzf <nombre_del_archivo>.tar.gz
```

Así queda instalada la aplicación, para arrancarla bastará con realizar los siguientes pasos:

```
# cd /opt/eclipse
# ./eclipse
```

En caso de que la instalación se haga con permisos de root, el ejecutable de eclipse solo quedaría accesible para dicho usuario, siendo necesario ejecutar la aplicación mediante el comando sudo. Sin embargo esto puede causar problemas a la hora de usar eclipse o ejecutar aplicaciones que se estén desarrollando mediante él, por ello sería conveniente otorgar los permisos necesarios a la carpeta completa, para que el usuario normal, que será el que generalmente utilice eclipse para el desarrollo de aplicaciones, tenga acceso y dominio total sobre el programa.

Una vez completada la instalación se inicia eclipse. Nada más arrancarlo, aparecerá una ventana de selección de workspace. Aquí se introducirá el nombre del workspace (el directorio de trabajo para el proyecto a realizar, que contendrá todos los datos y meta datos acerca del espacio de trabajo) en el que se trabajará.

Ahora ya se ha iniciado eclipse, y se puede pasar a la configuración del workspace desde el que se trabajará. En primer lugar se deberán descargar los plugins necesarios para construir una aplicación basada en servlets. Para obtenerlos, basta con descargarse el paquete comprimido de plugins wtp (Web Tools Platform) para eclipse, que se puede encontrar en su página oficial. Una vez descargado y descomprimido, se combinará la carpeta 'plugins' creada con la carpeta ya existente en el directorio /opt donde se ha instalado eclipse. Una vez se reinicie eclipse ya estarán los plugins listos para usar.

En segundo lugar se configurará desde el eclipse las opciones del servidor con el que se va a trabajar. Esto permite hacer cambios en el código de la aplicación y publicarlos simultáneamente en el servidor, desde el mismo eclipse. Así como también permite arrancar y parar el servidor cuando se desee. Para ello se inicia el eclipse y se seleccionan las siguientes opciones:

Window → Preferences → Server → Runtime Environment → Add → Apache v.5.5 → Next

En el campo 'name' se escribirá el nombre que se quiera dar al servidor, por ejemplo, 'Tomcat'. Y el campo 'path' se rellenará con el directorio donde se encuentra instalado el servidor, en este caso '/opt/apache-tomcat-5.5.25/'. Se pulsa 'OK' y así finaliza la configuración de opciones del servidor. Ahora le toca el turno a la configuración del servidor desde eclipse.

Para ello lo primero será cambiar a la vista de servidor. Basta con hacer click en Window -> Open perspective -> Debug. En la pestaña server se pulsa botón derecho y se crea un nuevo servidor. Haciendo doble click sobre él se abrirá otra pestaña con las opciones a configurar.

El nombre del servidor puede ser 'Tomcat (server)', por ejemplo. A continuación se selecciona la opción de usar configuración de Tomcat, para que los cambios del workspace se desplieguen en el servidor, y se especifican los directorios para ello. Esta parte es la más importante.

Hay un apartado llamado 'Launch configuration'. Se deberá clickar en éste y configurarlo, de modo que quede listo para poder desplegar las aplicaciones desarrolladas al servidor. En la segunda pestaña, 'Arguments' se debe cambiar el argumento 'start' por 'jps start'. Y el resto de los argumentos deberán quedar de la siguiente manera:

```
Dcatalina.base="/opt/apache-tomcat-5.5.25" -Dcatalina.home="/opt/apache-tomcat-5.5.25"  
  
Dwtp.deploy="/opt/apache-tomcat-5.5.25/Webapps/" -Djava.endorsed.dirs="/opt/apache  
tomcat-5.5.25/common/endorsed"
```

Los dos primeros especifican los directorios de instalación del tomcat, y -Dwtp.deploy guarda el path donde se desplegará la aplicación. Para terminar con la configuración del servidor, en la pestaña 'Environment' se añadirán dos variables de entorno necesarias para desplegar la aplicación en tomcat desde eclipse.

```
JPDA_ADDRESS=8000  
JPDA_TRANSPORT=dt_socket
```

Esto además, conlleva un pequeño cambio en los scripts para lanzar tomcat. Al inicio del fichero '/opt/apache-tomcat-5.5.25/bin/startup.sh' deberán añadirse estas dos líneas, para indicar la exportación de las variables de entorno utilizadas por eclipse para el manejo del servidor:

```
export JPDA_ADDRESS=8000
export JPDA_TRANSPORT=dt_socket
```

Así queda completada la configuración del servidor desde eclipse. Ahora será necesario crear el nuevo proyecto desde el cual se desarrollará la aplicación de gestión de cursos. Para ello se hace click en:

File → New → Other → Web → Dynamic Web Project

En el campo 'Name' se escribe el nombre que se quiera dar al proyecto, y en 'Target Runtime' se seleccionará la configuración del tomcat realizada anteriormente. Por último pulsar 'Finish' y eclipse estará listo para usarse. El sistema de ficheros de la aplicación quedará de la siguiente forma:

```
src/      → es.upna.csie/      → <clases>.java
                                     → ....
web/      → Administracion/        → <ficheros-prottegidos>
                                     → css/                    → <ficheros-css>

                                     → imag/                   → <imagenes-usadas>
                                     → META-INF/              → context.xml
                                     → Web-INF/                → Web.xml

                                     → index.jsp
                                     → <archivos-jsp> ...
```

Una vez que ésta se compile, el sistema de ficheros se desplegará en el tomcat, en una carpeta con el nombre del proyecto dentro de el directorio '/opt/apache-tomcat-5.5.25/Webapps/', generándose entonces los ficheros con extensión .class creados en la compilación de las clases creadas y que serán los que la aplicación utilice.

El context.xml, contiene información del esquema de la aplicación y de la conexión a la base de datos. Si se quiere configurar el acceso a la base de datos debe especificarse ahí, de la siguiente manera:

```
<Resource name="jdbc/MySql" auth="Container" type="javax.sql.DataSource"
maxActive="100" maxIdle="30" maxWait="10000"
username="sinfo" password="*****" driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost:3306/InscripcionCSIE-PRUEBAS"/>
```

La variable name es nombre que se le dará a está conexión en concreto. Username y password son el usuario y contraseña con el que la aplicación debe conectarse a la base de datos, y la url consta de 'ip:puerto/<NombreDeLaBaseAConectar>'

Como último paso, y antes de pasar a desarrollar la aplicación en sí, se deberá proceder a la instalación de Postfix, de manera que la aplicación pueda tener un servidor de correo en el lugar donde se encuentre corriendo, y pueda funcionar correctamente.

3.5 Instalación de Postfix

Lo primero será instalar el paquete de postfix. Para ello desde el terminal se escribirán los comandos correspondientes:

```
# sudo apt-get install postfix
```

Una vez instalado el servidor de postfix habrá que configurarlo. Los archivos de configuración del mismo se encuentran en el directorio /etc, por tanto se accederá a ellos para definir los parámetros que se consideren oportunos para el servidor.

```
# vim /etc/postfix/main.cf
```

Al final del fichero, se encuentran las líneas que se deben modificar, correspondientes al nombre del host que tiene el servidor de correo así como el del dominio.

```
# myhostname      correo.usue.com      (Este nombre debe de ser el configurado en el DNS)
# mydomain        usue.com             (Dominio en el que se recibirán los correos)
# mydestination   usue.com
# mynetworks      192.168.1.0/24, 127.0.0.0/8    (Dirección de red)
```

Ahora se reinicia el servidor para comprobar que no existan errores, y para que cargue la última configuración realizada.

```
# /etc/init.d/postfix restart
```

A continuación será necesario tratar de conectarse al servidor de correo telnet mediante el puerto 25. Para ello se ejecutarán las siguientes órdenes:

```
# telnet 127.0.0.1 25
Trying to 127.0.0.1.escape character is '^]'. 220 correo.usue.com ESMTP postfix (ubuntu)
```

Para conectar con el servidor y comprobar que está corriendo correctamente, basta con “mandar” el saludo de bienvenida al mismo:

```
# < ehlo HOLA
```

Con esto queda totalmente configurado el servidor de correo, y listo para usar.

4. ANÁLISIS

Para realizar un correcto análisis y posterior diseño de la aplicación Web, se deberán analizar y concretar en primer lugar los requisitos esenciales que ha de presentar la aplicación en sí. Como se ha comentado anteriormente, se han predefinido una serie de características que dicha aplicación deberá cumplir, y a partir de las cuales podrán obtenerse los requisitos del sistema Web.

A continuación, se pasará a repasar dichas características propias de la aplicación, de forma que se obtengan los requisitos en base a las mismas.

- Acceso mediante registro de usuarios.
- Posibilidad de personalización de los datos de perfil de usuario.
- Uso de sesiones para cada conexión.
- Creación de correo interno.
- Envío de mails de confirmación e información a los usuarios participantes.
- Creación de cursos.
- Posibilidad de inscripción de usuarios a cursos docentes.
- Búsqueda de cursos por parte de los usuarios.
- Información de las inscripciones de otros usuarios a cursos creados por el usuario.
- Información de las inscripciones de un usuario a cursos docentes.
- Internacionalización del lenguaje de la aplicación.

A partir de aquí se pueden obtener los siguientes requisitos para el análisis y posterior diseño de la aplicación.

4.1 Requisitos funcionales

R1) Login de usuarios

En primer lugar, y siendo una de las características base del proyecto, está el hecho de que el acceso a la Aplicación Web se encontrará basado en la autenticación de usuarios. Es decir, es un sistema con acceso restringido, únicamente para aquellos usuarios que estén registrados. Esto a su vez conlleva el desarrollo de una herramienta que permita la recuperación de contraseñas a usuarios que hayan podido olvidarla.

R1.1) Sistema de recuperación de contraseña.

En la página inicial de la aplicación se incluirá la opción de acceder al sistema en caso de estar registrado, o por el contrario, la opción de registrarse para poder hacer uso del servicio. Este registro será almacenado en la base de datos, sobre la cual se hará la comprobación correspondiente cada vez que se produzca un acceso a la aplicación por parte de uno de los usuarios.

R1.2) Sistema de registro para usuarios no registrados.

Por otra parte, existirá igualmente la posibilidad de acceso como administrador, opción de la cual tan sólo podrá hacer uso el propio Administrador de la aplicación. En este apartado se dispondrá de dos tipos de acciones, por una parte la gestión y mantenimiento de datos del sistema, y por otro lado, la posibilidad de realizar y obtener estadísticas del funcionamiento y uso de la aplicación.

Sería positivo que en éste último apartado, se diera la posibilidad al administrador de obtener listados informativos acerca de la aplicación, como por ejemplo, la obtención de datos de usuarios asistentes a determinados cursos, número de usuarios formados por curso, o listados con los cursos ofrecidos a lo largo de un año.

Por otro lado, en la parte de mantenimiento, sería necesario controlar la parte de usuarios, cursos e inscripciones, para que el administrador pueda acceder a los mismos y modificarlos en base a diferentes situaciones que puedan darse en la gestión del sistema.

R1.3) Administración de la Aplicación Web permitida únicamente al administrador del mismo.

R2) Personalización de datos de perfil de usuario

Dentro de la aplicación, es decir, accesible a aquellos usuarios registrados, existirá la opción de configurar y personalizar la información de cada usuario. Cada uno de ellos podrá editar la información de su cuenta y sus datos personales, como nombre apellidos o contraseña.

Sin embargo, el nombre de usuario, es decir, el email, permanecerá intacto, ya que será el dato único e inequívoco por el cual se distinguirá a cada usuario del resto de personas registradas. Siguiendo el patrón de acceso en la Universidad Pública de Navarra tendría la siguiente estructura:

nombreDeUsuario@e.unavarra.es

R2.1) Nombre de usuario en el sistema no modificable.

R3) Correo Interno

En siguiente lugar, y continuando con los usuarios, cabe destacar que estos tendrán la opción de comunicarse entre ellos mediante un sistema de correo interno a la aplicación. Un sistema de mensajes privados de usuario a usuario, que permitirá la comunicación entre ellos, de forma que se produzca una interacción entre usuarios de toda la aplicación, que ayude a mejorar ciertas partes y aspectos de la docencia de cada uno de ellos.

Esto permitirá que usuarios inscritos en ciertos cursos puedan aportar sus puntos de vista al responsable de dicha docencia. Así como un intercambio de opiniones, información, etc. que hagan más rica y variada esa gestión de la docencia.

R4) Agenda

Así mismo, la aplicación tendrá una herramienta en la que se gestione la actividad de cada usuario. De esta forma, cada persona podrá acceder y ver a qué eventos está apuntado, y cuales de ellos darán comienzo en cierto periodo de tiempo.

Aquellas inscripciones realizadas por el usuario, aparecerán en la agenda en la fecha correspondiente a su inicio, como recordatorio para el usuario, y de forma que éste pueda visualizar de una forma general los compromisos educativos que tiene pendientes, o que están actualmente llevándose a cabo.

El calendario de eventos será de formato mensual, permitiendo al usuario navegar de un mes a otro, de forma que se muestre con claridad todos los datos citados anteriormente.

R5) Creación de cursos

Otra de las opciones características de la aplicación y a la cual todo usuario registrado tendrá acceso, es a la creación que cursos docentes propios. Es decir, cursos formativos que el mismo usuario impartirá, seleccionando él mismo el lugar, fecha y contenidos del curso en sí. De esta forma, podrá crear la formación que desee y dispondrá de la posibilidad de publicarla para que el resto de usuarios tengan acceso a dicho curso, matriculándose en el mismo en caso de ser de su interés.

Para ello, el usuario dispondrá en su página inicial de la aplicación de un acceso a la creación de cursos. Esto, le llevará hasta un formulario que deberá rellenar, y seguidamente, una vez haya terminado de completarlo, de forma sencilla el curso será creado, y publicado para que el resto de usuarios se inscriban.

El usuario podrá decidir cómo cumplimentar el formulario, especificando el lugar y fecha que desee, así como otros datos necesarios. Y de la misma forma, se le permitirá a dicho usuario decidir si desea publicar inmediatamente dicho curso para el resto de usuario, o si desea hacerlo más adelante.

Esto lleva a otro requisito fundamental, que es la posibilidad de edición de cursos ya creados por un usuario, que le permitirá modificar datos del mismo, así como eliminarlo en caso de así desearlo. Si esto ocurriese, se notificaría a aquellos usuarios matriculados de los cambios efectuados.

R5.1) Edición y eliminación de cursos.

Una vez que se cree un curso por un usuario, y éste lo haga público, el resto de usuarios del sistema podrán comenzar a inscribirse en el mismo. Por ello, será necesario que cada usuario pueda acceder a ver las inscripciones realizadas sobre sus cursos, de forma que le permita también una comunicación directa con aquellos usuarios que vayan a acudir a sus sesiones de formación.

R5.2) Ver los usuarios inscritos en los cursos creados por el usuario.

R5.3) Permitir una comunicación con los usuario inscritos.

R6) Inscripción de usuarios a cursos docentes

Todo usuario registrado que tenga acceso a la herramienta podrá hacer una búsqueda de los cursos disponibles para inscripción. En dicha búsqueda aparecerán todos los cursos disponibles y la información principal de cada uno de ellos.

Así el usuario podrá elegir si le interesa alguno de ellos. Si es así, bastará con seleccionar el curso y hacer click sobre un botón. De esta forma quedará inscrito en dicho curso.

Además, una vez hecha la inscripción al usuario se le mostrarán todos los datos de ésta, como el lugar del curso, horarios, etc. Así mismo se le informará de si existen plazas libres o si por el contrario se encuentra en la lista de espera.

R6.1) Muestra de información de las inscripciones de usuarios a cursos.

De la misma forma, todo usuario inscrito en un curso, podrá acceder a un espacio reservado únicamente para éste, en el cual se expondrá más detalladamente la descripción del curso, y además permitirá colaborar a los usuarios inscritos en el mismo, haciendo comentarios al respecto o adjuntando archivos que puedan ayudar o ser de interés para el resto de usuarios.

Se permitirá por tanto, que todo usuarios inscrito a un curso pueda hacer uso del sistema de ficheros de la aplicación, y tenga la opción de subir documentación del curso al espacio reservado para éste, o de la misma forma, que pueda descargarse aquellos ficheros o documentos que otros usuarios hayan subido al sistema.

R6.2) Creación de un espacio reservado para información y documentación de cada curso.

Por cada inscripción el usuario inscrito recibirá un correo electrónico con toda esta información necesaria de matrícula. Lo mismo ocurrirá en caso de que dicho usuario decida desapuntarse del curso.

R6.3) Eliminar inscripción a un curso.

R7) Búsqueda de cursos

Todo usuario registrado y que haya accedido a la aplicación tras logearse, tendrá la posibilidad también de realizar búsquedas de cursos de formación, de los cuales podrá ver una pequeña descripción detallada del mismo, y tendrá la opción de decidir si apuntarse o no al mismo. En caso de hacerlo, será de forma sencilla, seleccionando el curso al que desee acudir y haciendo click sobre un botón de inscripción.

Para esta búsqueda existirá una interfaz especial, en la cual podrá elegirse cómo realizar dicha búsqueda, permitiéndose tres tipos diferentes. En primer lugar, se podrá buscar cursos por el usuario que la imparte, de forma que se pueda agrupar la formación disponible según su tutor, y permita al usuario de una forma sencilla buscar cursos que puedan ser de temática similar, ya que el responsable de esa docencia será experto en ciertas materias.

Por otro lado, podrán realizarse búsquedas por nombre de curso. De esta forma los resultados quedarán ordenados alfabéticamente, y permitirán al usuario encontrar la formación determinada que está buscando de una forma más sencilla y directa.

Y finalmente, para juntar ambas características anteriores, podrán realizarse búsquedas mediante la introducción de una palabra clave, de forma que los resultados sean más acotados y puedan centrarse en un tema especificado por el usuario.

Por último, existirá una opción, válida también para usuarios no registrados, de forma que se permita ver la oferta docente que existe en el calendario, sin necesidad de haberse logeado previamente. Desde la página de inicio de la aplicación existirá un enlace a este apartado, de forma que sea accesible para todo el mundo.

R7.1) Consulta de oferta docente externa a la aplicación.

R8) Mails de confirmación desde la aplicación

La aplicación podrá enviar mails de confirmación o información a los usuarios. Se realizará esta acción cada vez que un usuario se inscriba en un curso, avisándole vía mail de su correcta inscripción a la docencia correspondiente. Así mismo, se informará de sus datos de matrícula y de si dispone de una plaza para el curso, o por el contrario debe mantenerse en lista de espera hasta que queden plazas libres.

En este último caso, al producirse una modificación y ser admitido en la lista de asistentes recibirá de nuevo un mail de aviso. Así mismo se hará con todos aquellos usuarios apuntados a un curso, el cual deje de encontrarse disponible o sea borrado por el usuario que imparte dicho curso.

Todo esto será posible gracias a un servidor de mail que deberá instalarse en el servidor que albergue la aplicación Web. Se conseguirá así mediante la instalación de postfix en el servidor, explicado en apartados anteriores. Éste funcionará así también como servidor de mail, haciendo posible el envío de correos desde la aplicación a los usuarios que deban ser informados.

4.2 Requisitos no funcionales

RN 1) Sencillez y consistencia: Uso sencillo y consistente del interfaz de usuario

Debido a que la aplicación está dirigida a todo tipo de usuarios, una parte importante del diseño y desarrollo de la misma es crear un interfaz de interacción con el usuario que muestren una sencillez y claridad de uso para dichos usuarios.

La aplicación Web deberá ser visualmente clara, de forma que el usuario pueda intuir de forma sencilla que opciones presenta en cada momento cada pantalla, y cómo llevarlas a cabo. Un diseño sencillo y directo será necesario para conseguir esto. De la misma forma, no se querrá que las pantallas sean inconsistentes, de forma que la manera de operar para acciones similares en diferentes pantallas sea igual en todas ellas, y no existan inconsistencias en la usabilidad y manejabilidad de la herramienta en diferentes partes de la misma.

El diseño visual podrá ser discutido y modificado, pero se requiere que posea un aspecto visual directo, y que no exista una mezcla cargante de colores u objetos en cada pantalla. De forma que el usuario detecte de forma directa y visual las acciones que desea o tiene la opción de realizar en cada momento de su interacción con el sistema.

RN 2) Accesibilidad: Internacionalización de la aplicación

La aplicación se diseñará de forma que quedé internacionalizada. Es decir, el idioma en el que se muestre la aplicación dependerá del idioma que el usuario configure como principal en su navegador. La aplicación tomará estos datos y desde sus archivos internos de propiedades seleccionará los datos exactos a mostrar en cada momento y para cada usuario.

Esto implicará también que los ficheros de propiedades de cada idioma se encuentren bien traducidos para que la internacionalización sea del todo correcta. En principio la aplicación se presentará con dos idiomas, el castellano como lengua por defecto, y la posibilidad de ver la aplicación en inglés.

En caso de que la aplicación se desee mostrar en algún otro idioma bastaría tan solo con crear un nuevo archivo de propiedades para el idioma deseado, con su correcta traducción. Esto último sería tarea de un traductor especializado, que asegure que la traducción final de la aplicación se realice de forma adecuada y sin errores, de forma que el usuario disfrute de una perfecta interacción con la herramienta.

RN 3) Seguridad: Uso de sesiones para cada conexión

Tratándose de una aplicación Web es importante que cada sesión de un usuario diferente se identifique inequívocamente. De esta forma se evitan problemas en su funcionamiento, y se consigue que cada usuario y sesión estén unidas, de forma que los datos de éste para cada conexión permanezcan accesibles mientras la sesión esté activa.

De la misma forma, se garantiza la seguridad y privacidad de los datos de cada usuario y de la sesión que se encuentra activa mientras el usuario utiliza de la aplicación.

Así pues, cada vez que un usuario inicie sesión, los datos de dicho usuario y sesión se mantendrán activos, haciendo posible que mientras el usuario hace uso de la aplicación, ésta pueda utilizar esos datos cuando le sea necesario. Por ejemplo, mantendría el nombre de usuario, el inicio de la conexión, y otros datos que puedan servir de interés y que sean necesarios en el ámbito de la programación interna de la aplicación.

RN 4) Robustez: Acceso simultáneo de usuarios

La definición de robustez indica que es la capacidad de los sistemas software de reaccionar de forma apropiada ante situaciones de condiciones excepcionales. Es decir, se debe garantizar que el sistema responda de forma correcta en situaciones en las que el contexto de desarrollo de la herramienta no sea el esperado en un funcionamiento normal de la aplicación.

En este caso, al tratarse de un sistema Web, habrá que garantizar el correcto funcionamiento de la herramienta en casos en los que el acceso de usuarios sea demasiado alto, y el servidor pueda no responder correctamente. Lo primero será establecer el número de usuarios que puedan usar la herramienta a lo largo de un día, y lo que es más importante, el número de usuarios que se espera usen la herramienta de forma simultánea.

La ventaja para establecer estos límites se encuentra en el hecho de que la aplicación será usada dentro del ámbito de la Universidad Pública de Navarra, luego esto ya establece un límite de usuarios, principalmente usuarios del Servicio Web, y de la misma forma, garantiza que el acceso simultáneo a la herramienta no va a ser nunca mayor a éste número de usuarios de la herramienta.

Por tanto, será necesario instalar la herramienta en un servidor que sea capaz de albergar toda esta acción. Es posible que un sólo servidor, aunque sea potente, no sea suficiente para el uso de la herramienta. En caso de ser así, podría desplegarse en dos servidores, de forma que haya un balanceo de carga y se evite la saturación de la herramienta, garantizando un correcto funcionamiento de la misma.

Sistemas actuales de la Universidad Pública de Navarra funcionan actualmente de esta manera, por tanto, existiendo un referente de correcto funcionamiento en la misma situación de accesibilidad sería posible aplicar dicha configuración también a este servicio.

RN 5) Fiabilidad: Fiabilidad del sistema

Con la fiabilidad del sistema se desea conseguir que la herramienta se comporte y responda de forma adecuada a las diferentes especificaciones del uso de la misma. Para ello la fase de pruebas deberá comprobar que todas las especificaciones definidas a continuación se llevan a cabo de forma correcta y tal y cómo se define que debe ser su comportamiento final.

Este requisito estará estrechamente ligado al de robustez, de forma que una vez se asegure la robustez del sistema, en condiciones tanto normales como extremas, tan sólo será necesario que cada una de las funciones del sistema lleven a cabo sus funciones de forma correcta.

4.3 Diagramas de casos de uso

Conociendo ya los requisitos que se desea cumpla el sistema, podrá pasarse a analizar los casos de uso que puedan darse en la interacción con la aplicación. Se comenzará desde la iteración más genérica, donde se presentará el uso de la herramienta Web por los diferentes usuarios, profundizando una a una en las actividades que éstos puedan realizar sobre el sistema.

Se establecerán los flujos principales y alternativos de cada caso de uso, así como los requisitos necesarios en cada uno de ellos. Se especificará así mismo el resultado final esperado para cada opción. De esta forma, se logrará un análisis detallado del funcionamiento y requisitos del sistema, que ayudará a realizar un posterior diseño e implementación, ahondando en el funcionamiento y diseño de las clases que crearan el servicio final.

Iteración 0

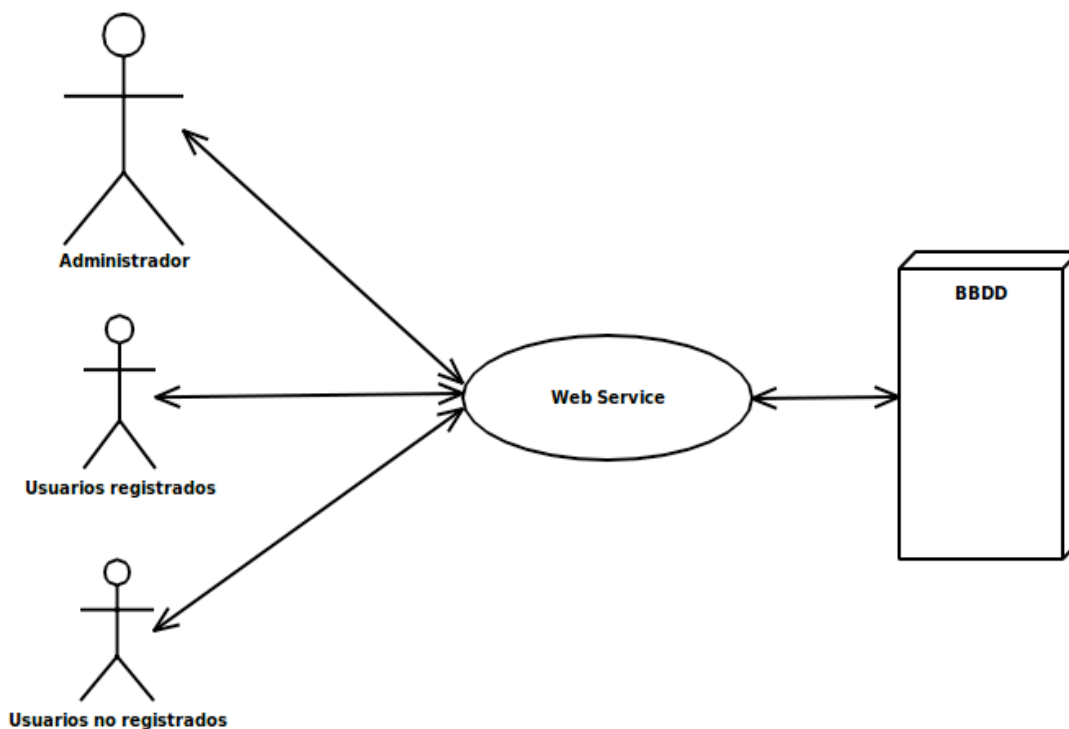


Figura 1 – Casos de uso, iteración 0.

Caso de uso	Esquema general del sistema.
Actores	El administrador del sistema, los usuarios registrados, y los usuarios no registrados que deseen acceder a la oferta académica de la aplicación.
Precondición	La página de la herramienta Web debe encontrarse disponible.

Iteración 1

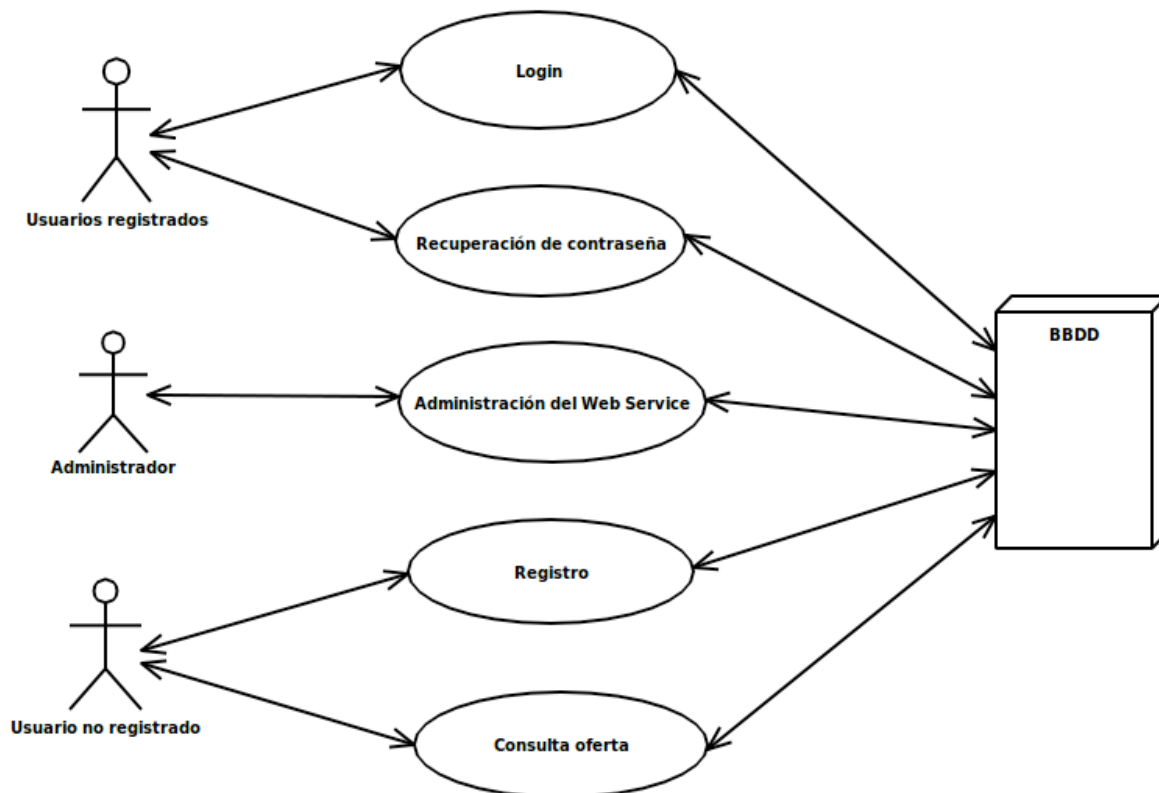


Figura 2 – Casos de uso, iteración 1.

Caso de uso	Uso general de la aplicación Web.
Actores	El administrador del sistema, los usuarios registrados, y los usuarios no registrados que deseen acceder a la oferta académica de la aplicación.
Precondición	La página de la herramienta Web debe encontrarse disponible.
Flujo principal	El usuario puede registrarse, loguearse, o consultar la oferta académica de la aplicación. El administrador podrá acceder al mantenimiento.
Flujo alternativo	Error en el servicio de la aplicación.
Postcondición	Es usuario quedará registrado, logueado, o finalizará la consulta de oferta académica. El administrador ejecutará las acciones de mantenimiento.

Iteración 2

Login:

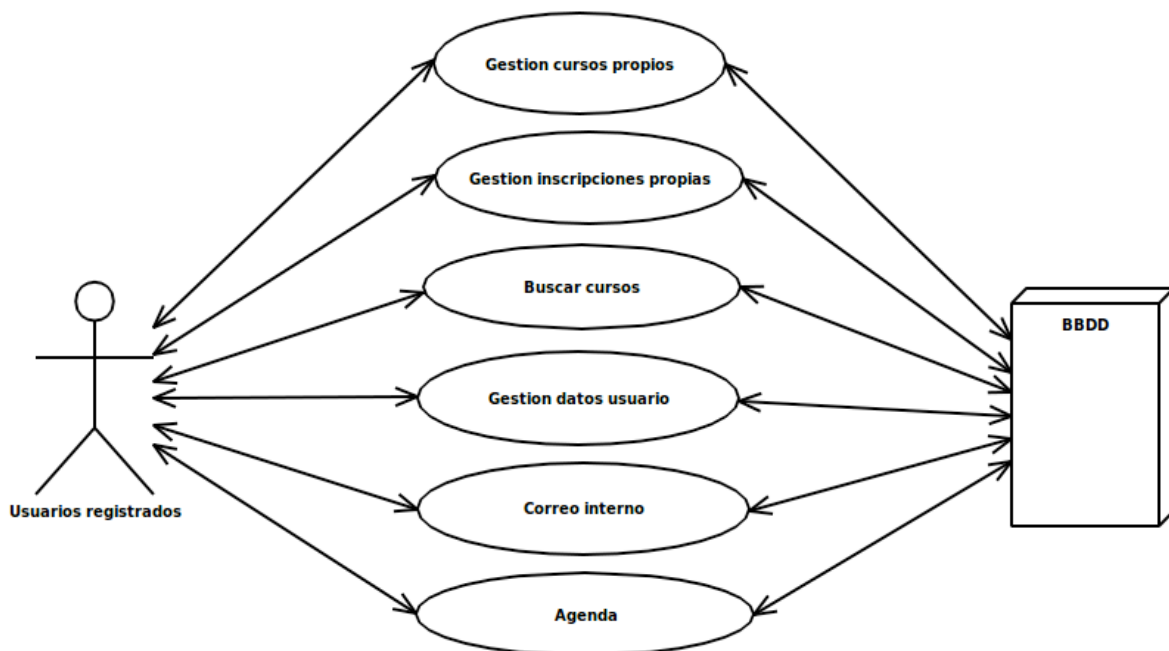


Figura 3 – Casos de uso, iteración 2.1.

Caso de uso	Login
Actores	Usuarios registrados.
Precondición	Acceder como usuario registrado.
Flujo principal	El usuario gestiona sus cursos, inscripciones, datos personales o correo interno. También puede realizar búsquedas de cursos o visualizar su agenda.
Flujo alternativo	Login incorrecto.
Postcondición	El usuario realiza las acciones deseadas de forma correcta.

Administración de la aplicación Web:

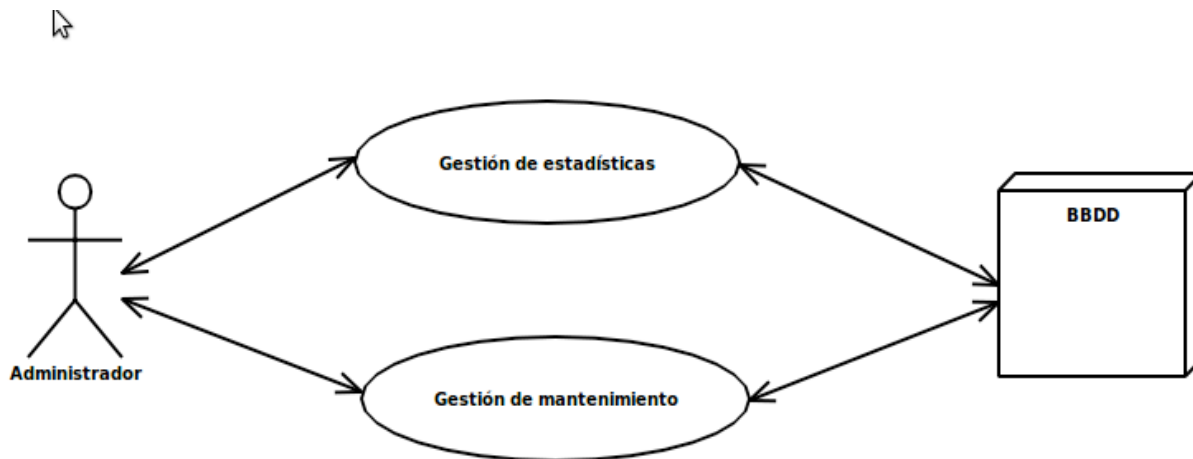


Figura 4 – Casos de uso, iteración 2.2.

Caso de uso	Administración de la aplicación Web.
Actores	El administrador del sistema.
Precondición	Ser el administrador y conocer las claves de acceso.
Flujo principal	Gestionar estadísticas de la formación o llevar a cabo el mantenimiento del sistema.
Flujo alternativo	Error en las acciones realizadas o identificación incorrecta.
Postcondición	Estadísticas obtenidas correctamente o mantenimiento realizado sin problemas.

Iteración 3

Gestión cursos propios:

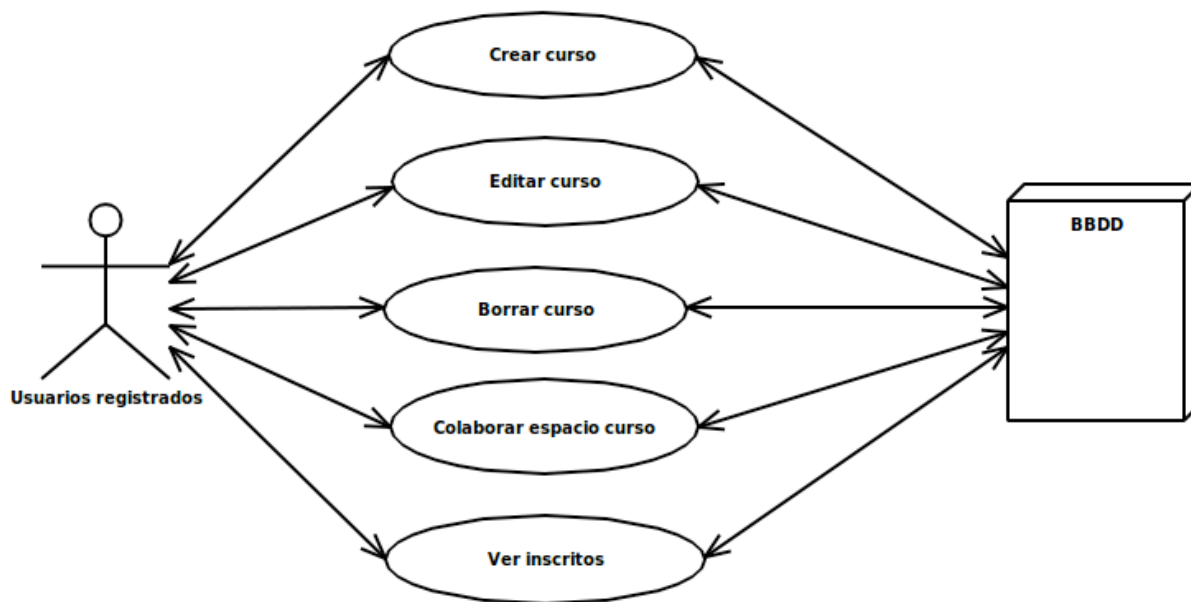


Figura 5 – Casos de uso, iteración 3.1.

Caso de uso	Gestión de cursos propios.
Actores	Usuarios registrados.
Precondición	Acceder como usuario registrado. Tener cursos creados para su edición, borrado, colaboración o lista de inscripciones.
Flujo principal	Crear, editar o borrar cursos del usuario, colaborar en el espacio de un curso, o acceder a la lista de inscripciones en sus cursos.
Flujo alternativo	Error en el login. Error al crear un curso.
Postcondición	Acciones realizadas correctamente.

Gestión inscripciones propias:

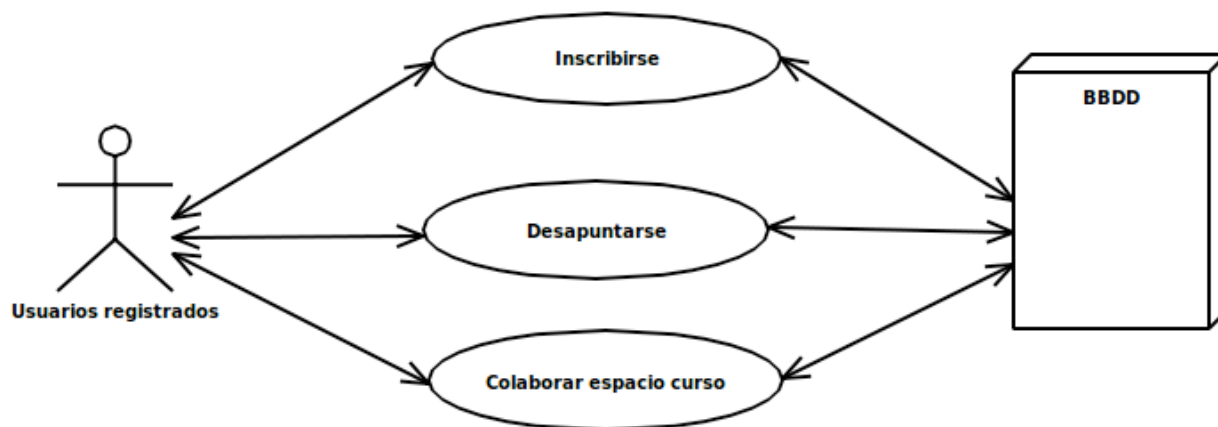


Figura 6 – Casos de uso, iteración 3.2.

Caso de uso	Gestión de inscripciones propias.
Actores	Usuarios registrados.
Precondición	Acceder como usuario registrado.
Flujo principal	Inscripción a un curso. Desapuntarse de un curso. Colaborar en el espacio de un curso.
Flujo alternativo	Error en el login. Error en la inscripción. Inscripción a un curso en lista de espera. Error al colaborar en el espacio de un curso.
Postcondición	Inscripción correcta con plaza en el curso. Desapuntarse correctamente. Colaborar sin errores en el espacio de un curso.

Gestión datos de usuario:

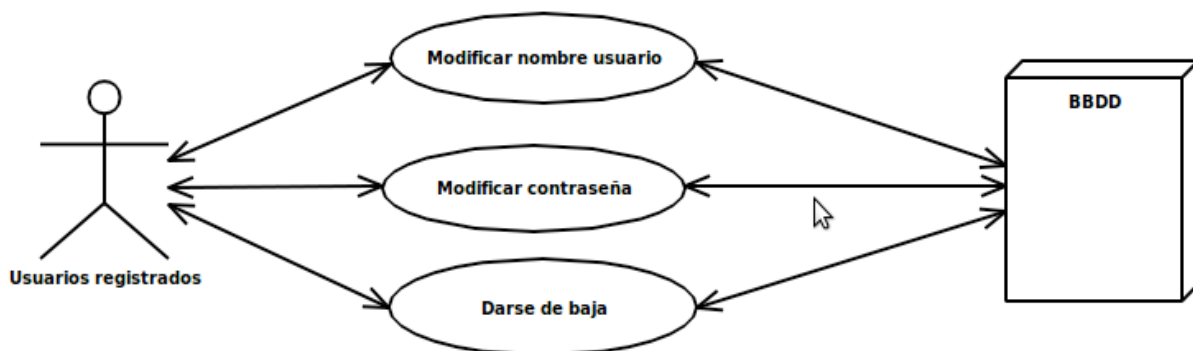


Figura 7 – Casos de uso, iteración 3.3.

Caso de uso	Gestión datos de usuario.
Actores	Usuarios registrados.
Precondición	Poseer una cuenta de acceso al sistema.
Flujo principal	Modificación nombre de usuario. Modificación contraseña de usuario. Eliminación cuenta de acceso al sistema.
Flujo alternativo	Error en la modificación de datos. Error en la baja de usuario.
Postcondición	Modificación de datos sin errores. Correcto funcionamiento de la cuenta tras la modificación.

Correo interno:

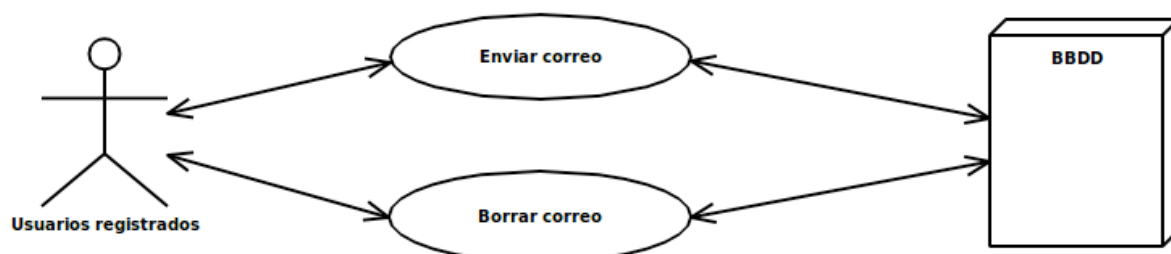


Figura 8 – Casos de uso, iteración 3.4.

Caso de uso	Correo interno.
Actores	Usuarios registrados.
Precondición	Acceder como usuario registrado.
Flujo principal	Enviar un correo. Borrar correo.
Flujo alternativo	Error en el login. Error al enviar un correo. Error al borrar un correo.
Postcondición	Envío correcto. Borrado correcto.

Gestión de mantenimiento:

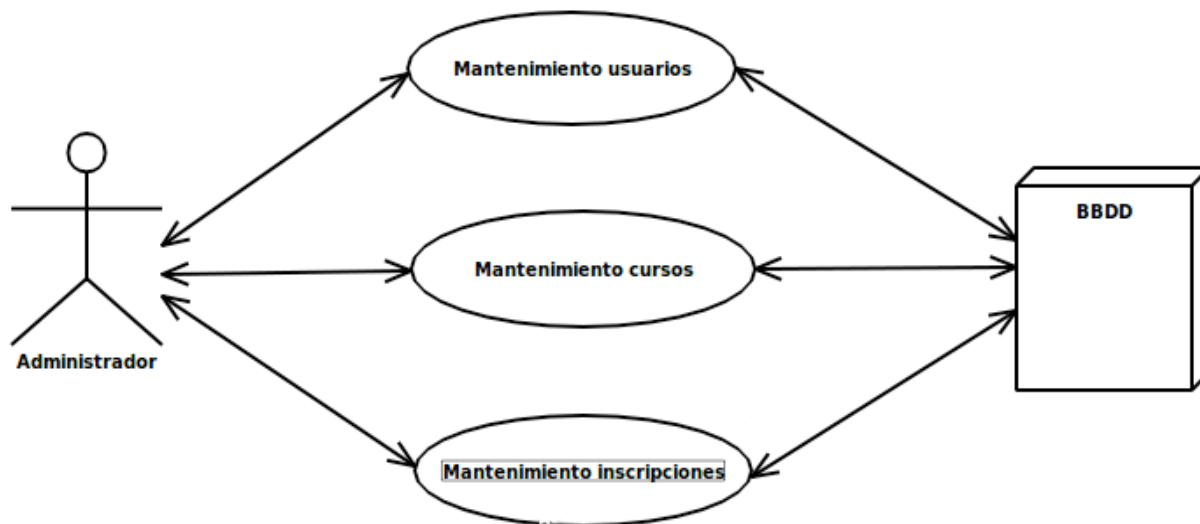


Figura 9 - - Casos de uso, iteración 3.5.

Caso de uso	Gestión de mantenimiento.
Actores	Administrador.
Precondición	Acceder como administrador del sistema.
Flujo principal	Crear, modificar o borrar usuarios. Crear, modificar o borrar cursos. Crear, modificar o borrar inscripciones.
Flujo alternativo	Acceso incorrecto como administrador. Error al crear, modificar o borrar.
Postcondición	Mantenimiento sin errores.

Gestión de estadísticas:

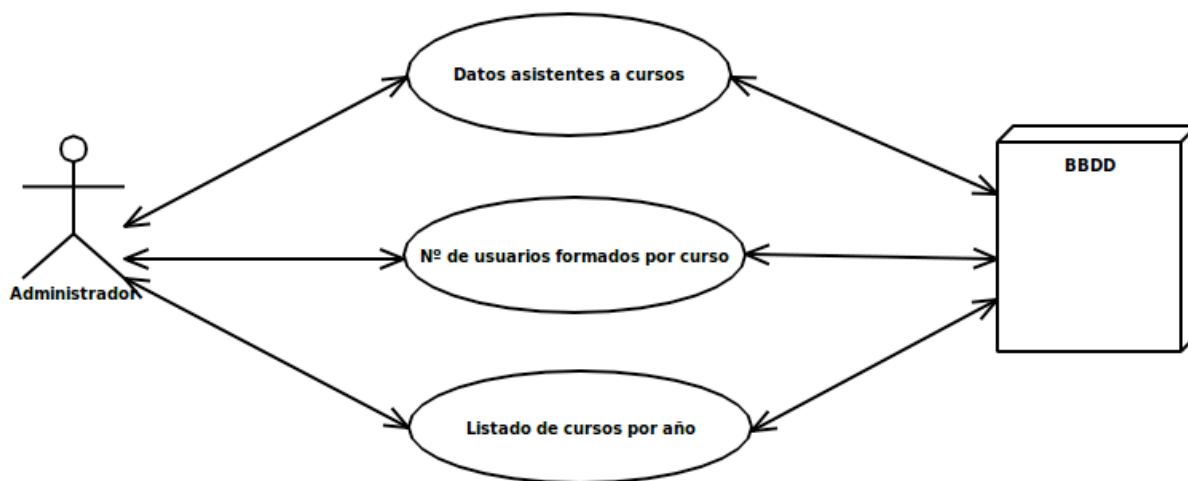


Figura 10 – Casos de uso, iteración 3.6.

Caso de uso	Gestión de estadísticas.
Actores	Administrador.
Precondición	Acceder como administrador del sistema.
Flujo principal	Obtener datos de los asistentes a cursos. Obtener el número de usuarios formador por curso académico. Obtener listados de cursos ofrecidos por año.
Flujo alternativo	Acceso incorrecto como administrador. Error en la obtención de datos estadísticos.
Postcondición	Obtención de estadísticas correctamente.

Iteración 4

Colaborar espacio curso:

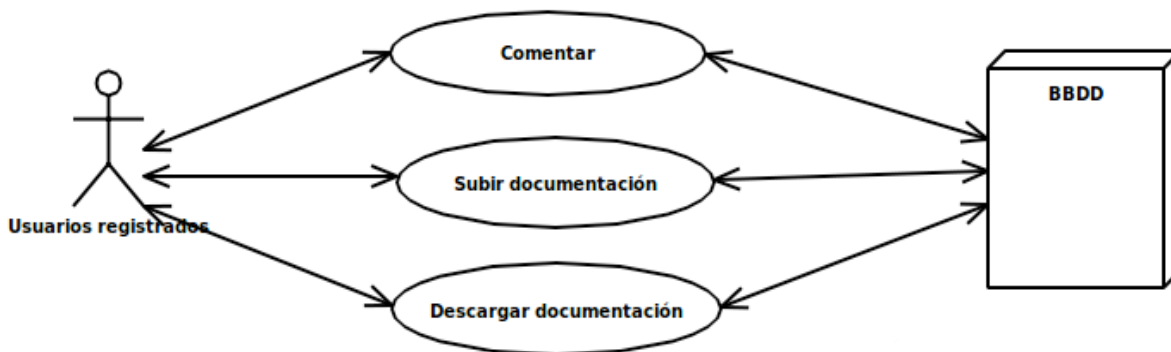


Figura 11 – Casos de uso, iteración 4.

Caso de uso	Colaborar espacio curso.
Actores	Usuarios registrados.
Precondición	Estar matriculado en un curso o ser el profesor del mismo.
Flujo principal	Comentar acerca de un curso. Subir documentación referente al curso. Descargarse documentación referente al curso.
Flujo alternativo	Error en la descarga o subida de ficheros. Error al comentar.
Postcondición	Subida correcta de archivos al espacio del curso. Obtención de ficheros de forma correcta. Comentarios correctos.

5. DISEÑO

Una vez realizada la fase de análisis del proyecto que logra definir los objetivos y requisitos esenciales que deberán llevarse a cabo a lo largo del proyecto, se obtiene una visión más clara y fundamental a la hora de comenzar con la fase de diseño. Conociendo los requisitos que debe cumplir el sistema, se pasará a diseñar el mismo, de forma que pueda especificarse la estructura de las clases de datos y el esqueleto relacional que va a utilizarse para desarrollar e implementar de forma definitiva la aplicación.

En primer lugar, se identificarán los diferentes subsistemas de la aplicación y se especificarán los diagramas de clases y secuencia, que desarrollen los diferentes casos de uso generados en la fase anterior del proyecto, de forma que la estructura del sistema queda diseñada de manera clara y pueda pasarse a la posterior implementación.

A continuación, y conociendo las necesidades de la aplicación, se dará paso a la definición de las entidades y primitivas necesarias para el futuro desarrollo, así como la identificación de sus responsabilidades y la relación entre ellas.

5.1 Diagramas de clases y diagramas de secuencia

Iteración 1

SUBSISTEMA: Recuperación de password

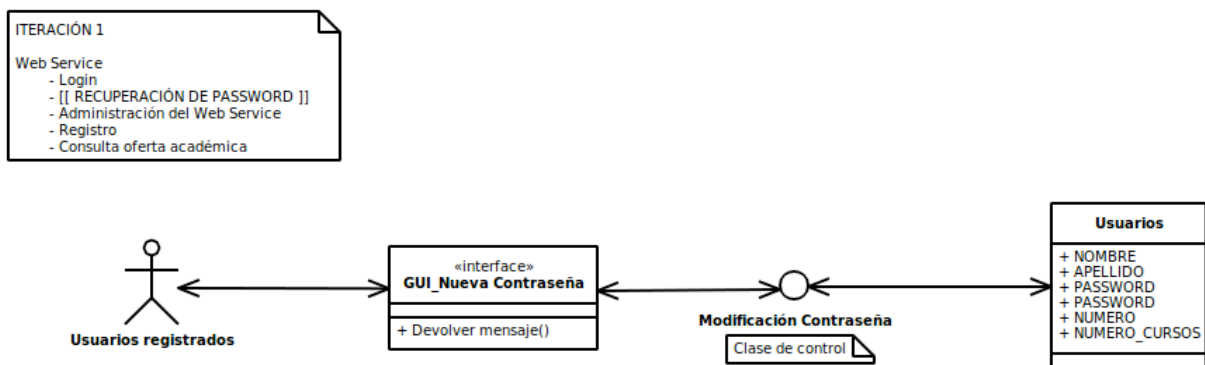


Figura 12 – Diagrama de clases, iteración 1.1.

Todo usuario registrado, en la pantalla de inicio de la herramienta, dispondrá de un enlace del cual podrá hacer uso en caso de olvidar la contraseña que permita su acceso a la aplicación. Bastará con introducir el nombre de usuario y el sistema creará una nueva contraseña que se notificará al usuario mediante un mensaje vía email.

De la misma forma, estos cambios se realizarán en la base de datos del sistema, actualizando la información de los datos de usuario con la nueva contraseña generada.

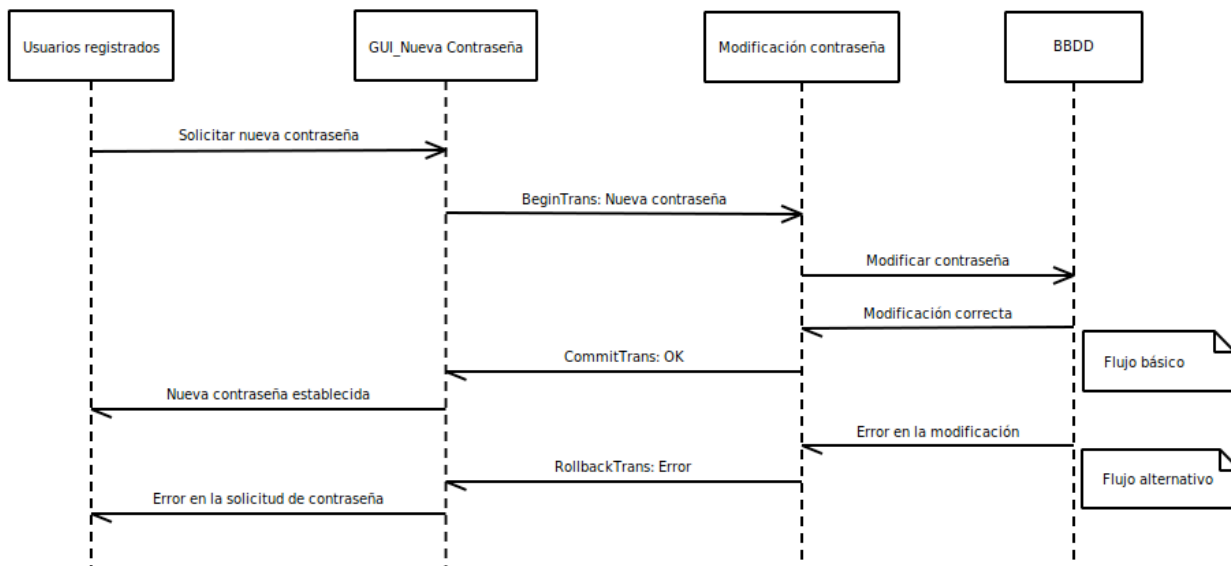


Figura 13 – Diagrama de secuencia, iteración 1.1.

SUBSISTEMA: Registro

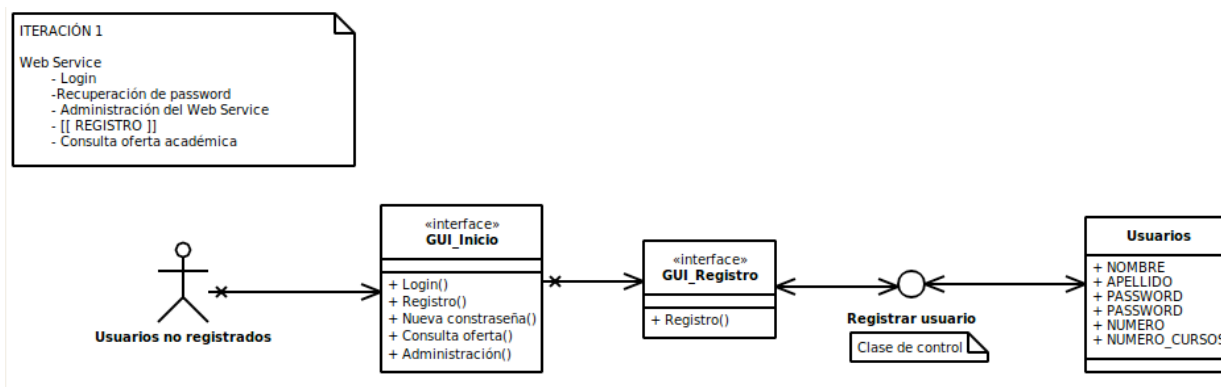


Figura 14 – Diagrama de clases, iteración 1.2.

Todo usuario que acceda a la página inicial de la herramienta Web, se encontrará con la opción de poder registrarse en el sistema y así tener acceso a la aplicación y hacer uso de ella.

Para registrarse bastará con introducir la información que se pida en el formulario, siendo ésta aquellos datos que comprenden la definición del usuario como entidad del sistema. Uno datos personales, como nombre, apellido e email con el que se tendrá acceso a la herramienta, y una contraseña que permita el acceso restringido del usuario a la aplicación Web.

Tras acabar con la operación se mostrará por pantalla al usuario un mensaje correspondiente al correcto, o por el contrario, incorrecto funcionamiento de la misma.

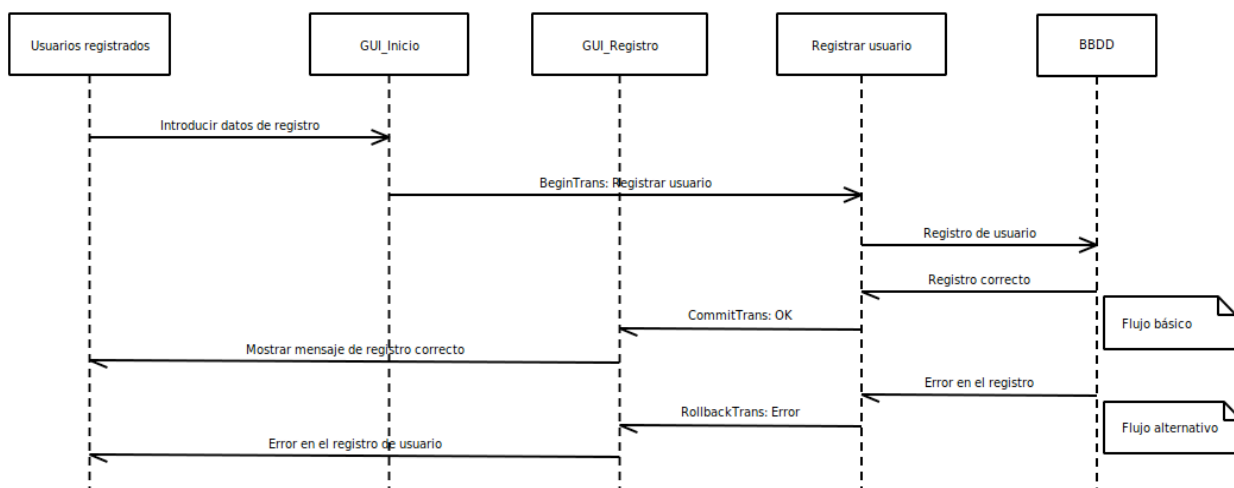


Figura 15 – Diagrama de secuencia, iteración 1.2.

SUBSISTEMA: Consulta de oferta académica

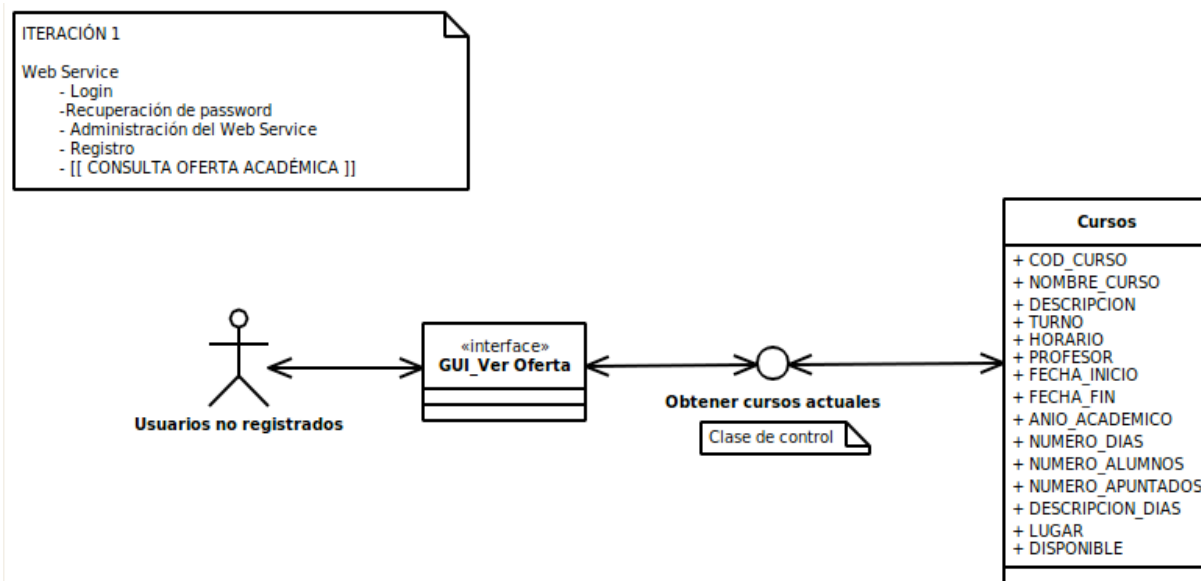


Figura 16 – Diagrama de clases, iteración 1.3.

Todos aquellos usuarios no registrados, o por el contrario, usuarios registrados que no hayan accedido a la herramienta por medio de la autenticación, tendrán acceso a la oferta docente del momento. En un enlace de la página inicial, se podrá hacer clic, y la aplicación conducirá al usuario a otra interfaz con un listado y descripción de los cursos ofrecidos en el momento.

Únicamente se trata de visualizar la oferta actual y sus características más importantes, sin embargo, en caso de querer acceder a la misma o inscribirse en uno de los cursos, se deberá iniciar sesión o crear una nueva cuenta que permita al usuario acceder a la herramienta y todas sus utilidades.

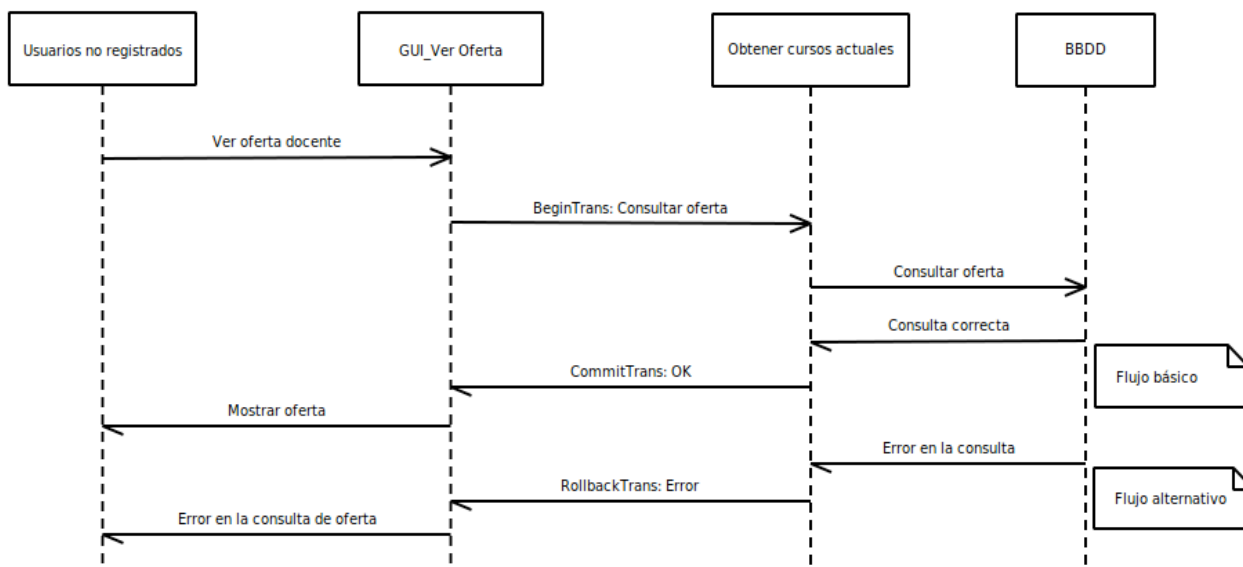


Figura 17 – Diagrama de secuencia, iteración 1.3.

Iteración 2

SUBSISTEMA: Buscar cursos

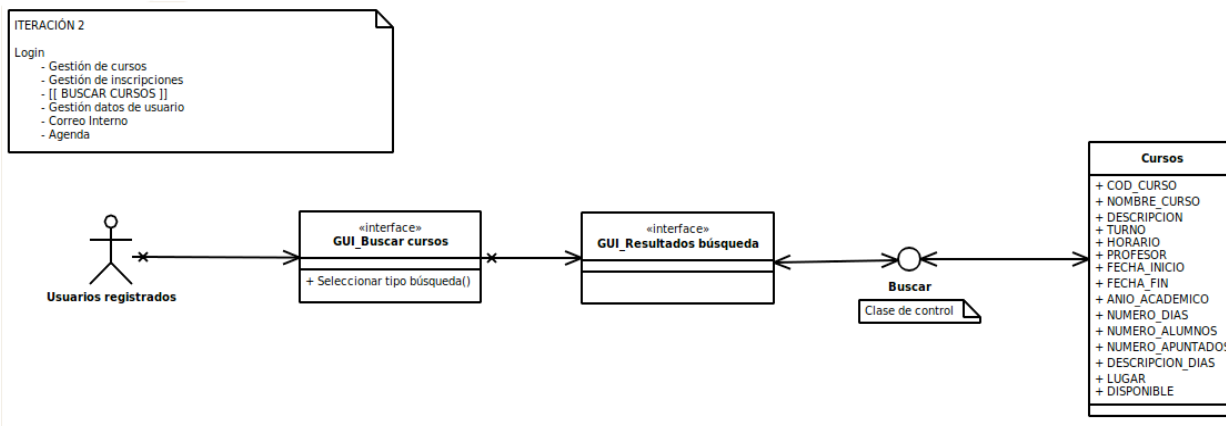


Figura 18 – Diagrama de clases, iteración 2.1.

Todo usuario registrado podrá realizar búsquedas para localizar qué oferta de cursos docentes existe en el momento. Desde la aplicación se podrá acceder a un interfaz de búsqueda en el cual se podrá seleccionar el tipo de filtro que se desea para la misma: será posible filtrar una búsqueda por curso, o de la misma forma, será posible obtener una lista de cursos ordenados por el docente que los imparte.

Tras realizar las acciones pertinentes por parte del sistema, se obtendrán los resultados de la búsqueda, que se mostraran al usuario por pantalla. Una vez se conozca la oferta, un usuario podrá inscribirse en el curso deseado (Iteración 3: Gestión de inscripciones).

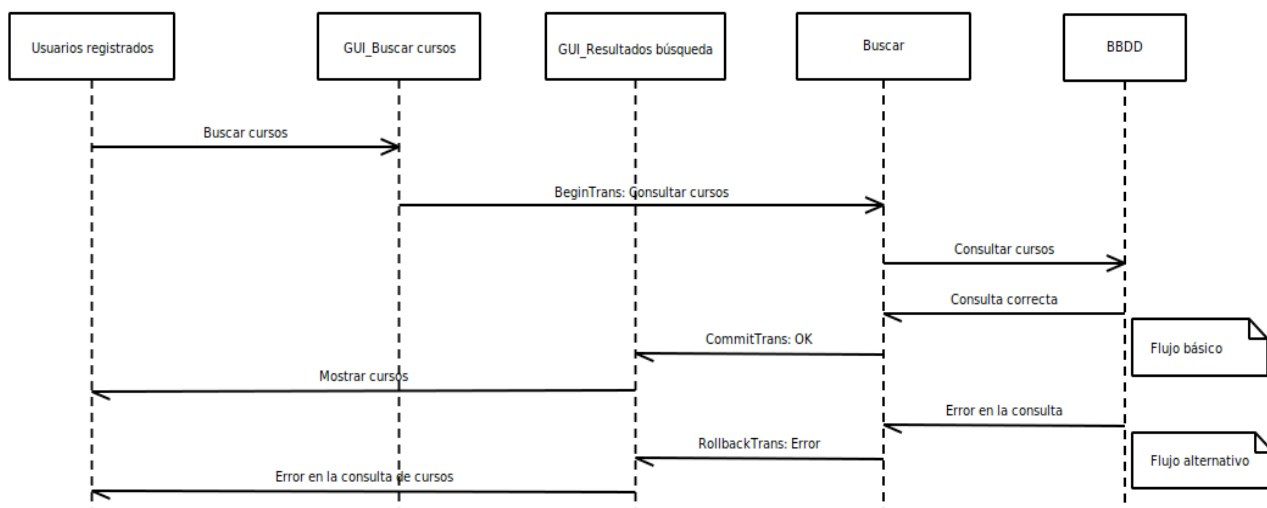


Figura 19 – Diagrama de secuencia, iteración 2.1.

SUBSISTEMA: Agenda

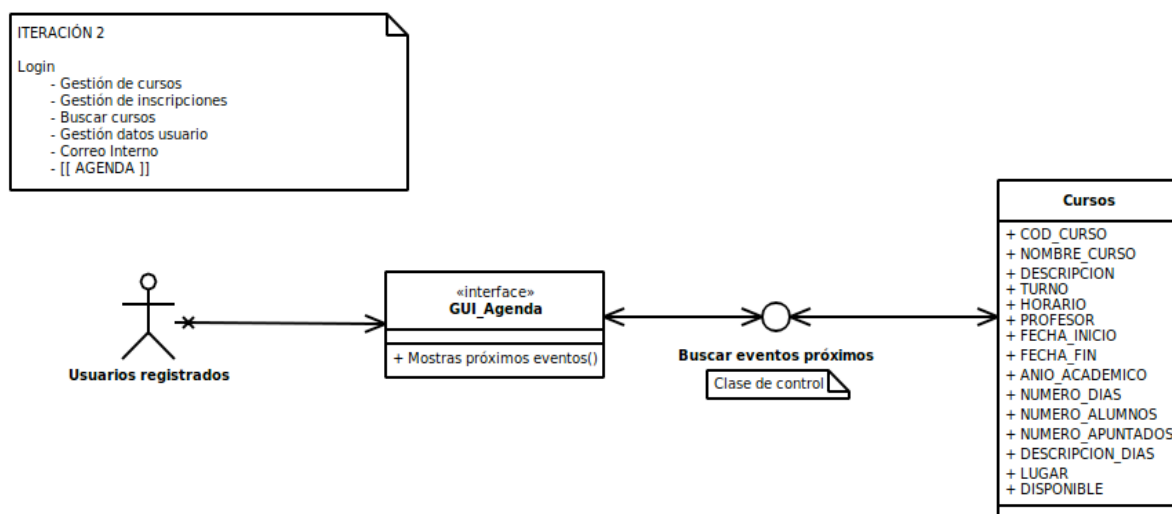


Figura 20 – Diagrama de clases, iteración 2.2.

Todo usuario registrado, una vez acceda a la aplicación, y desde cualquier parte de ésta, tendrá la opción de navegar directamente hasta una interfaz que muestre su agenda personalizada. En ella aparecerán todos aquellos eventos que tendrán lugar próximamente, o que por el contrario están ya celebrándose.

El calendario de eventos será de formato mensual, permitiendo al usuario navegar de un mes a otro, de forma que se muestre con claridad y de manera visual los cursos que el usuario tiene programados próximamente.

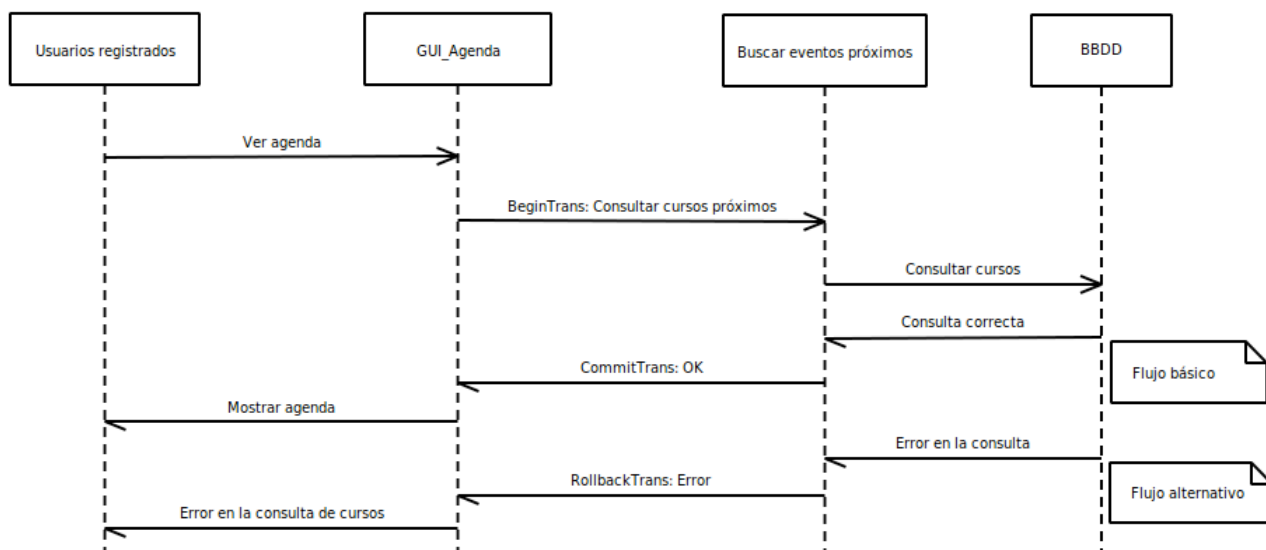


Figura 21 – Diagrama de secuencia, iteración 2.2.

Iteración 3

SUBSITEMA: Gestión de cursos propios

CREAR CURSO

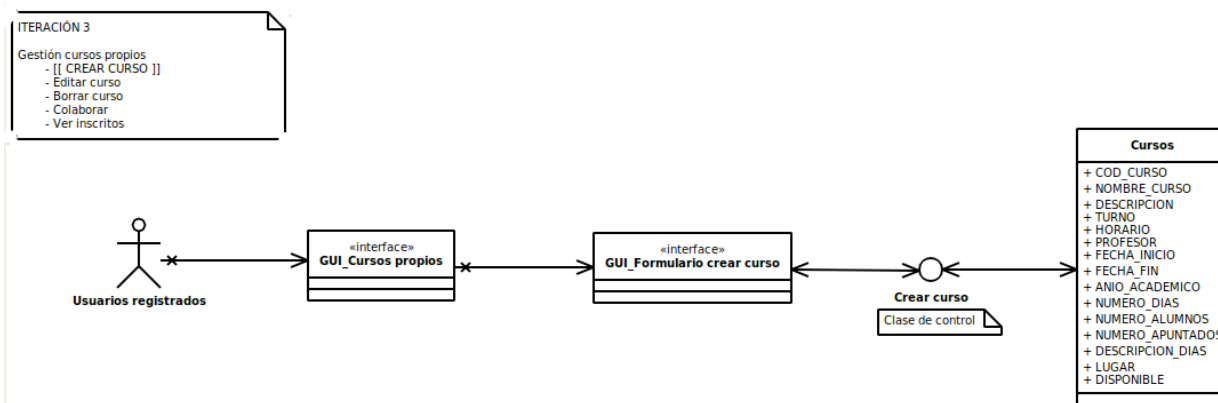


Figura 22 – Diagrama de clases, iteración 3.1.

Todo usuario registrado accederá primeramente a la interfaz donde se encuentra la gestión de cursos propios. En él aparecerán aquellos cursos creados por el mismo usuario. En este interfaz, dispondrá de la opción de crear un curso nuevo.

Si decidiese crear un curso, la aplicación le llevaría a la interfaz para crearlo. Éste es un formulario, en el cual el usuario deberá rellenar todos los campos, como nombre del curso, descripción del mismo, turno, horario, etc. Una vez realizada está operación, se creará el curso, introduciéndose en la base de datos. Si se marca como curso disponible, automáticamente después de su creación el resto de usuario tendrán ya la opción de localizarlo e inscribirse a él.

Tras la creación cualquier mensaje de error, correcto funcionamiento de la operación o cualquier otra opción será mostrada al usuario por pantalla.

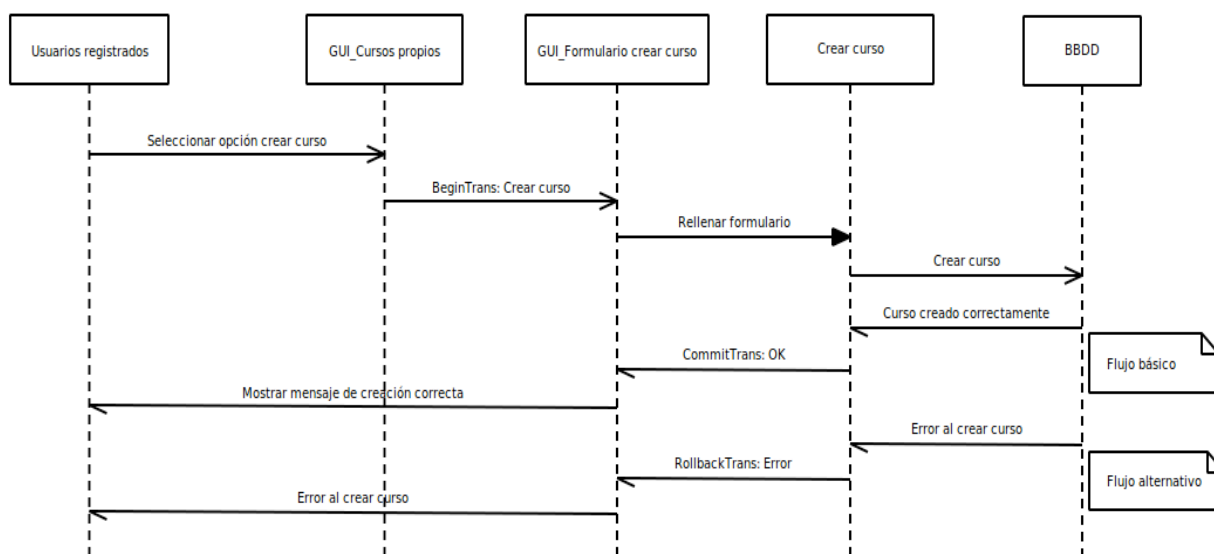


Figura 23 – Diagrama de secuencia, iteración 3.1.

SUBSISTEMA: Gestión de cursos propios

EDITAR CURSO

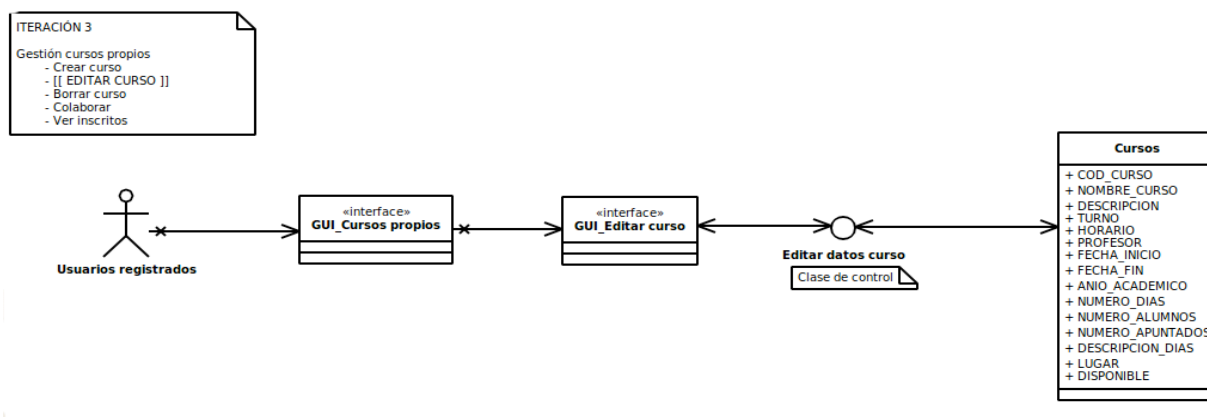


Figura 24 – Diagrama de clases, iteración 3.2.

Todo usuario registrado, en la página inicial de la herramienta dispondrá de una lista con los cursos impartidos por él mismo. Por tanto, existirá en cada uno de ellos la opción de editar la información del curso en sí. Bastará con hacer clic sobre el botón que permita dicha acción, y la herramienta se desplazará hasta el interfaz de edición de cursos.

En este interfaz, aparecerá un formulario similar al de la creación de cursos, donde se le permitirá al usuario modificar aquellos datos del curso que consideré oportunos. Una vez realizados los cambios, se producirán las actualizaciones pertinentes en la base de datos, y se mostrará al usuario un mensaje con el estado del cambio en los datos, si se ha realizado de forma correcta, o si por el contrario ha ocurrido algún tipo de error en el proceso.

Sin embargo, aunque se permita la edición de ciertos datos, habrá algunos que se consideren no modificables, siendo por ejemplo el nombre del profesor, que obligatoriamente deberá ser el mismo usuario, o el código de curso establecido por el sistema.

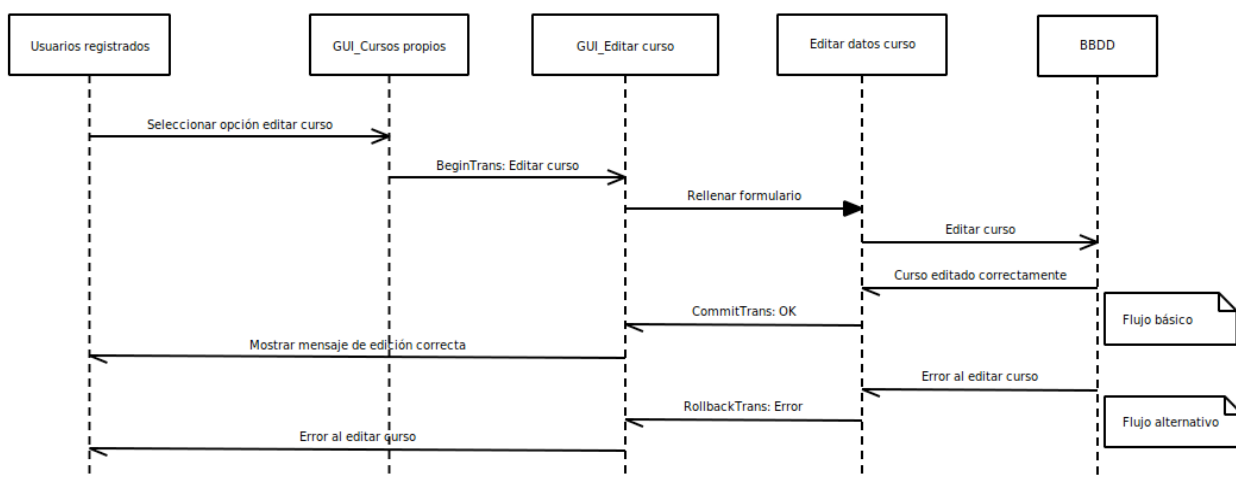


Figura 25 – Diagrama de secuencia, iteración 3.2.

SUBSISTEMA: Gestión de cursos propios

BORRAR CURSO

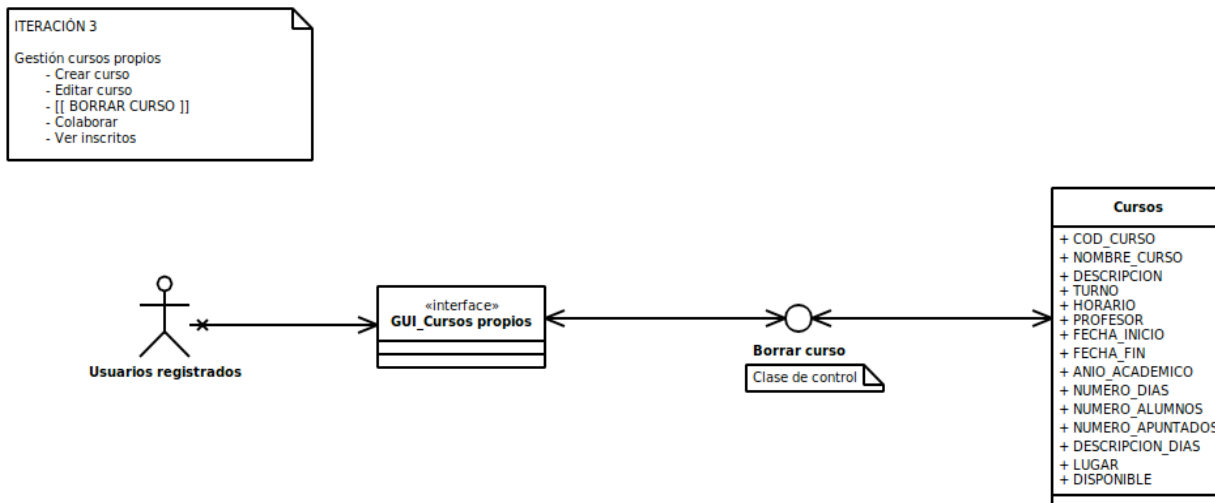


Figura 26 – Diagrama de clases, iteración 3.3.

Así como en la página de gestión de cursos propios se permite el crear o editar un curso, también deberá existir la opción de eliminar cursos. Para ello existirá un botón que haga dichos cambios, y se informe al usuario en caso de algún error en la actualización de los datos en el sistema.

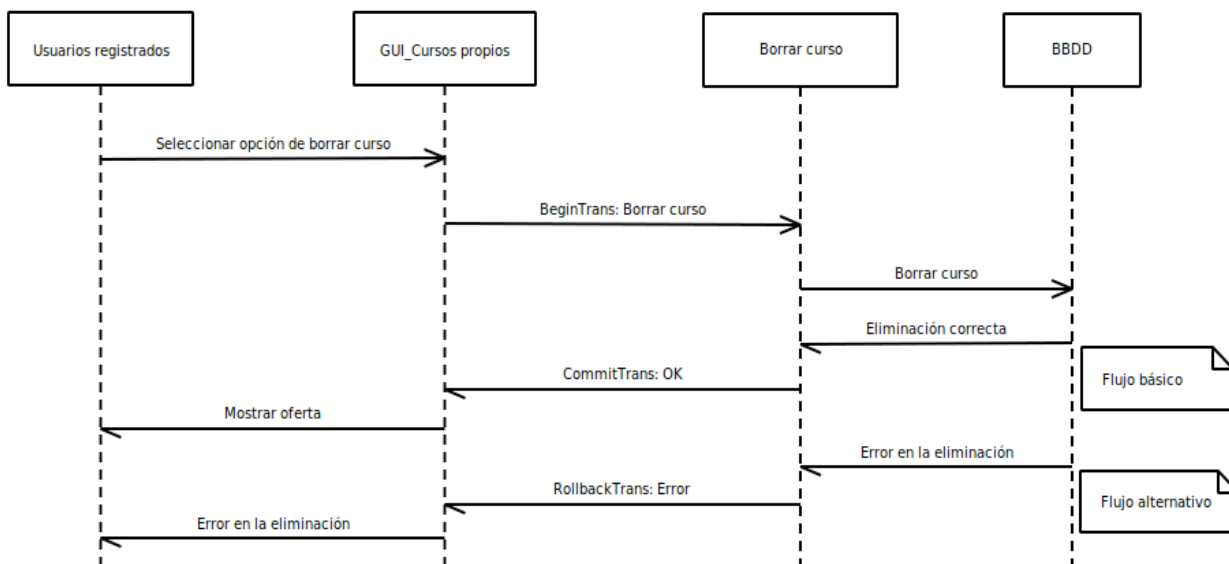


Figura 27 – Diagrama de secuencia, iteración 3.3.

SUBSISTEMA: Gestión de cursos propios

VER INSCRITOS

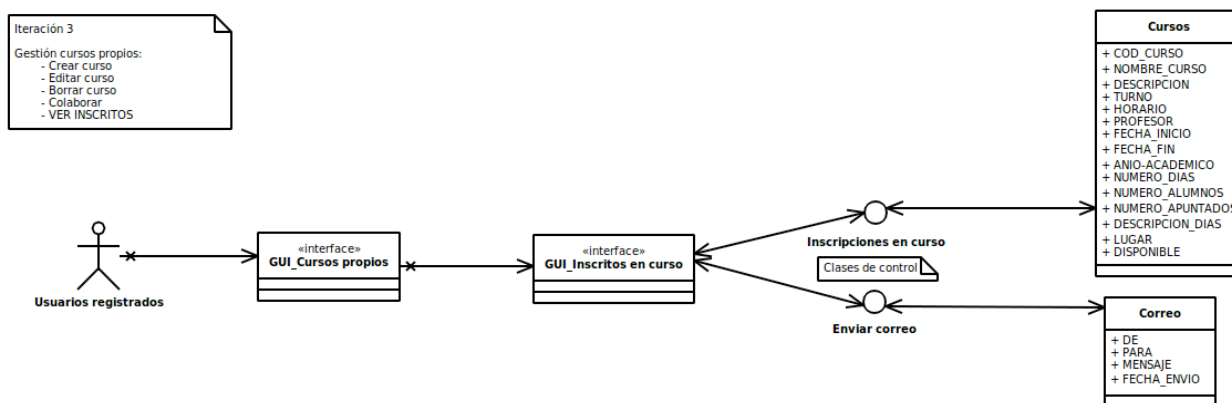


Figura 28 – Diagrama de clases, iteración 3.4.

Todo usuario registrado que sea el encargado de impartir un curso dispondrá de la opción de ver que otros usuarios se encuentran inscritos a su curso. Desde la interfaz de gestión de cursos propios se tendrá acceso a esta opción para cada uno de los cursos que el usuario posea como docente.

Si se elige esta opción, el sistema conducirá al usuario a un interfaz donde realicen las acciones pertinentes y se muestre por pantalla al usuario actual todos aquellos otros usuarios de la herramienta Web que se encuentren inscritos en su curso.

Como el requisito R5.2 indica, habrá de permitir que el usuario docente pueda comunicarse con todos aquellos usuarios inscritos en su curso. Por tanto, en el interfaz que muestre los usuarios inscritos al curso, el docente titular de éste dispondrá de la opción de comunicarse con ellos vía correo interno de la aplicación.

Para esta última opción se obtiene el siguiente diagrama de secuencia, mostrado en la figura 30.

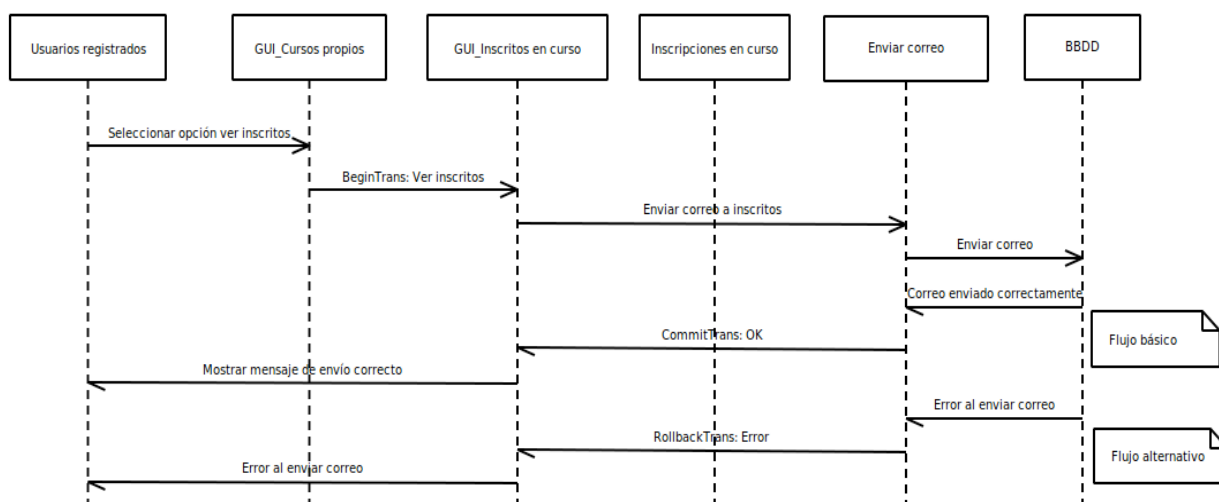


Figura 29 – Diagrama de secuencia, iteración 3.4.1.

Si lo que se desea es únicamente obtener una lista de todos aquellos usuarios inscritos en el curso, el diagrama de secuencia que se obtiene es el siguiente.

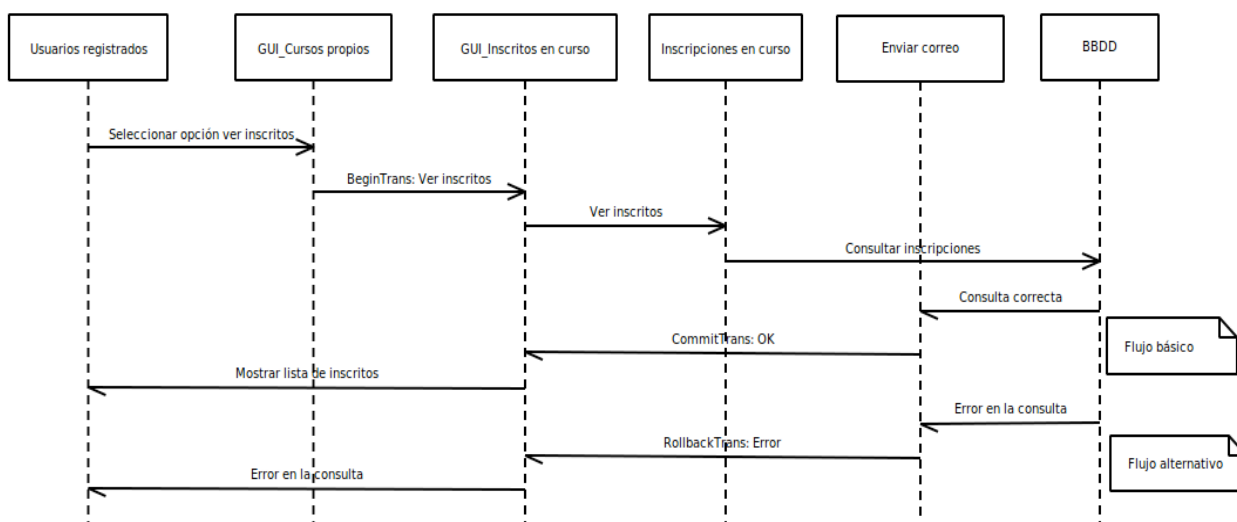


Figura 30 – Diagrama de secuencia, iteración 3.4.2

SUBSISTEMA: Gestión de inscripciones

INSCRIBIRSE

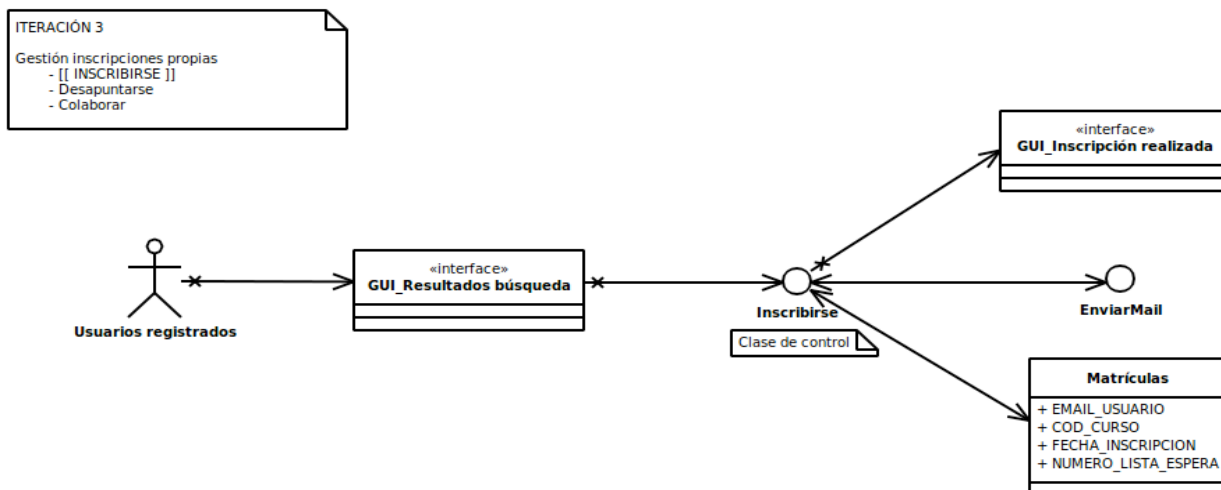


Figura 31 – Diagrama de clases, iteración 3.5.

Todo usuario registrado, tras realizar la búsqueda de cursos según sus intereses, si alguno de ellos le resulta de interés, podrá inscribirse, solicitando una plaza en el mismo. Para ello, bastará con ejecutar la búsqueda y seleccionar el curso deseado. Una vez realizada esta acción, el sistema procederá a inscribir al usuario en dicho curso.

Una vez hecha la inscripción, se mostrará al usuario un mensaje con los datos de su inscripción, incluyendo si dispone de plaza para el curso, o si por el contrario se encuentra en lista de espera. De la misma forma, se le enviará un mail que contenga dichos datos. También se permitirá que desde el interfaz de inscripciones propias el usuario pueda ver estos datos siempre que lo desee.

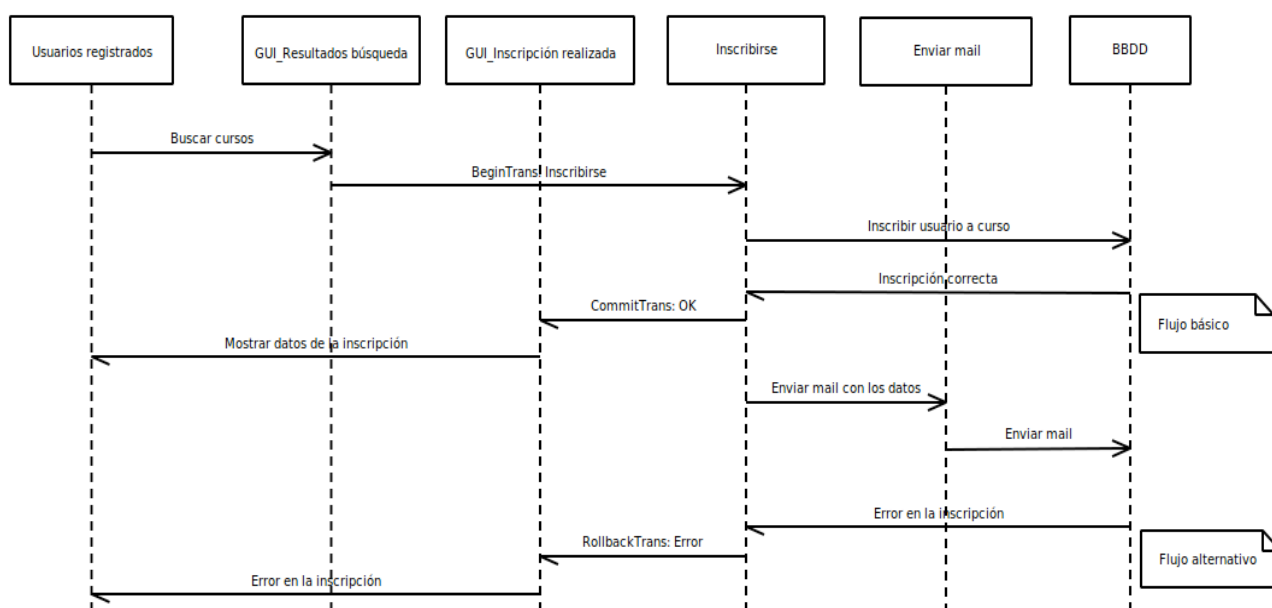


Figura 32 – Diagrama de secuencia, iteración 3.5.

SUSBSISTEMA: Gestión de inscripciones

DESAPUNTARSE

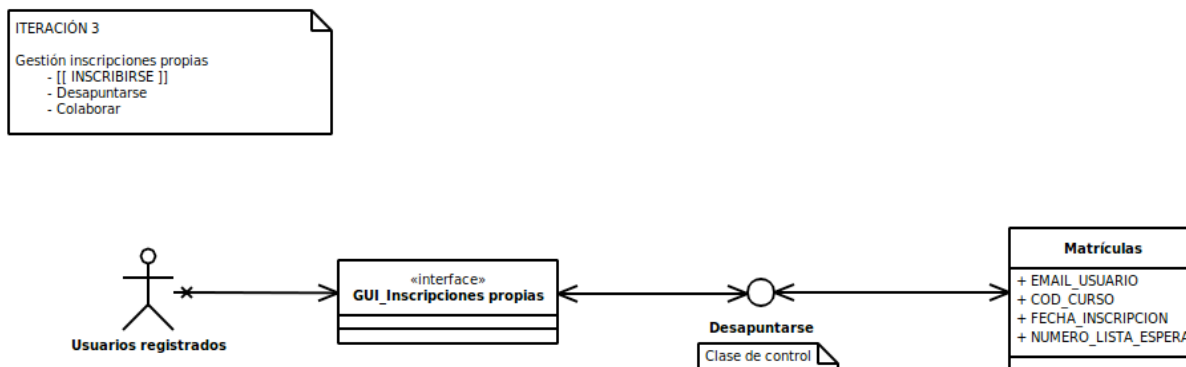


Figura 33 – Diagrama de clases, iteración 3.6.

Todo usuario registrado, podrá comprobar sus inscripciones a cursos docentes desde el interfaz de “Inscripciones propias”. Para cada curso, además de ver los detalles del mismo, como se ha comentado anteriormente, se permitirá al usuario la opción de desapuntarse del curso.

Si escoge dicha opción, el sistema ejecutará los cambios necesarios y mostrará al usuario su nueva situación de inscripciones.

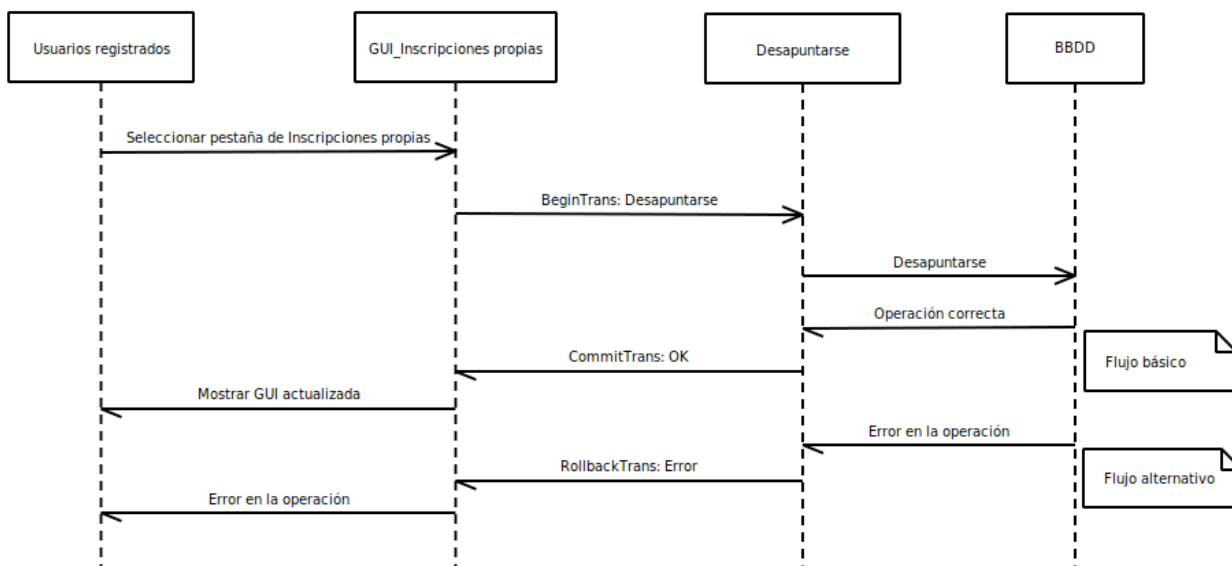


Figura 34 – Diagrama de secuencia, iteración 3.6.

SUBSISTEMA: Gestión datos de usuario

MODIFICAR NOMBRE

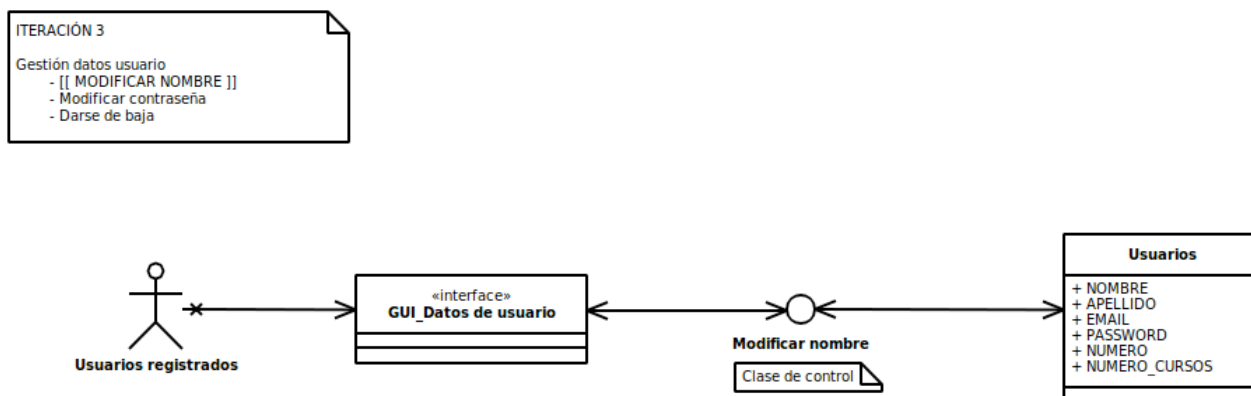


Figura 35 – Diagrama de clases, iteración 3.7.

Todo usuario registrado tendrá la opción de acceder a los detalles de su información privada y modificarlos según sus preferencias. Podrá tanto modificar su nombre y apellido, como la contraseña de acceso al sistema. Sin embargo, no se dará la opción de modificación de email, ya que es el dato clave que identifica a cada usuario, en el caso de la Universidad Pública de Navarra, con una estructura como la siguiente: *nombreDeUsuario@e.unavarra.es*.

Debido a esto, el dato de nombre de usuario (email), no será modificable en ninguno de los casos. Una vez el sistema realice la operación los cambios se aplicaran inmediatamente en el funcionamiento del sistema.

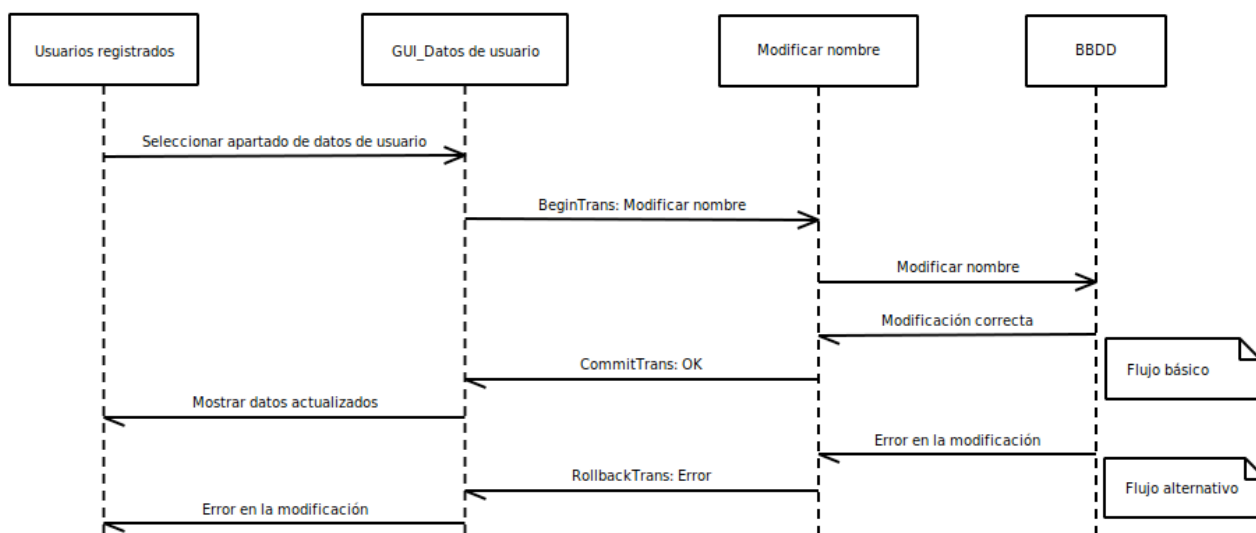


Figura 36 – Diagrama de secuencia, iteración 3.7.

SUBSISTEMA: Gestión datos de usuario

MODIFICAR CONTRASEÑA

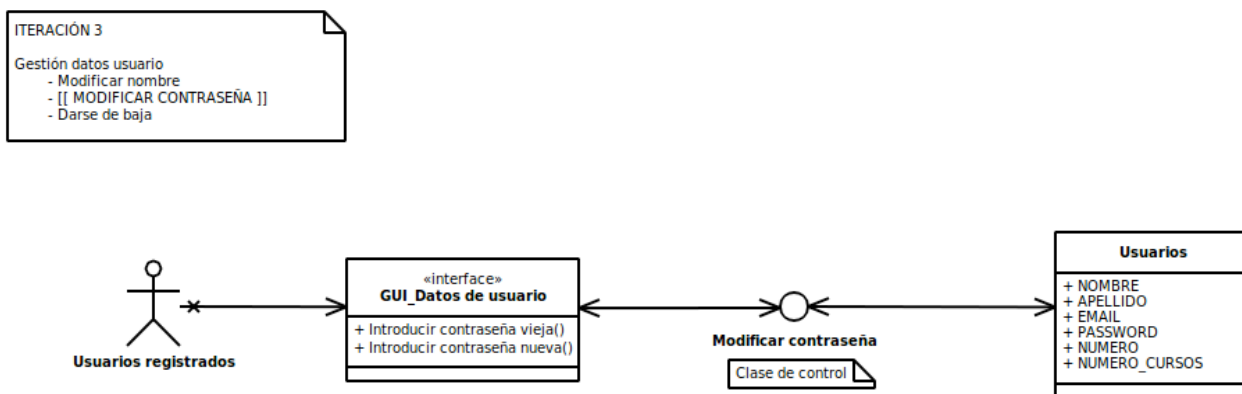


Figura 37 – Diagrama de clases, iteración 3.8.

Todo usuario registrado tendrá la opción de acceder a los detalles de su información privada y modificarlos según sus preferencias. Como se ha comentado en el caso anterior, éste podrá tanto modificar su nombre y apellido, como la contraseña de acceso al sistema.

Para esto último, deberá ingresar en el formulario del interfaz de datos de usuario su contraseña actual, y la nueva que desea establecer. Pulsando el botón de modificar datos, el sistema validará que la contraseña actual introducida por el usuario coincide con la contraseña registrada por el sistema en la base de datos, y si esto es así, se procederá a hacer el cambio y guardar las actualizaciones. En caso contrario se mostrará al usuario el mensaje de error correspondiente.

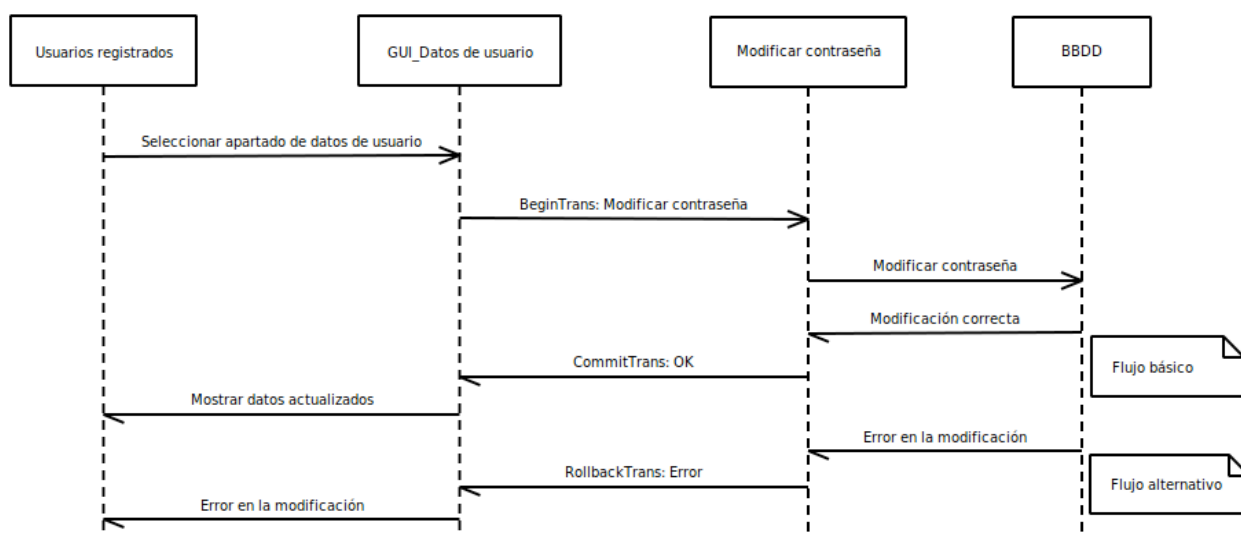


Figura 38 – Diagrama de secuencia, iteración 3.8.

SUBSISTEMA: Gestión datos de usuario

DARSE DE BAJA

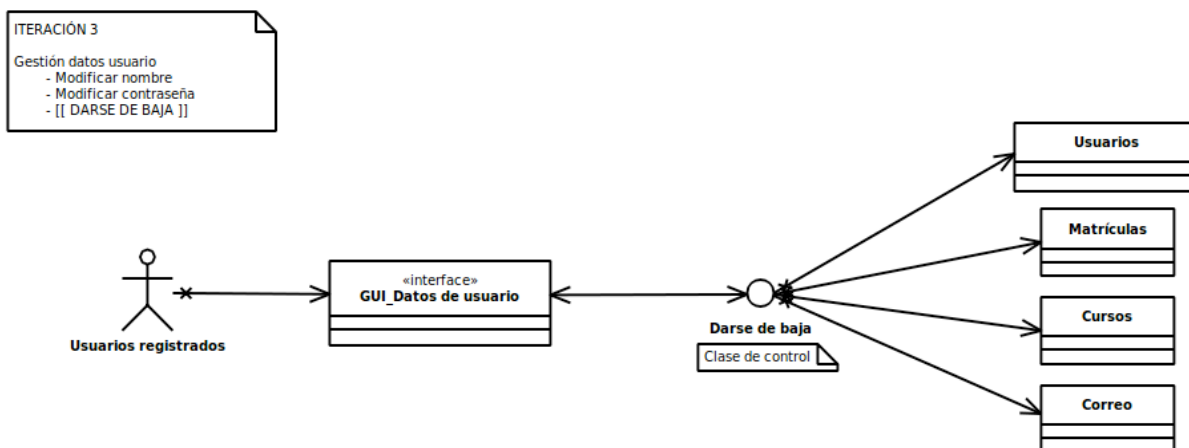


Figura 39 – Diagrama de clases, iteración 3.9.

Todo usuario registrado podrá darse de baja y eliminar su cuenta. Mediante esta acción, el sistema eliminará toda la información existente en relación al usuario. Tanto datos privados, como docencia impartida, cursos asistidos, y correo existente en la aplicación.

En caso de producirse algún error se comunicaría al usuario inmediatamente.

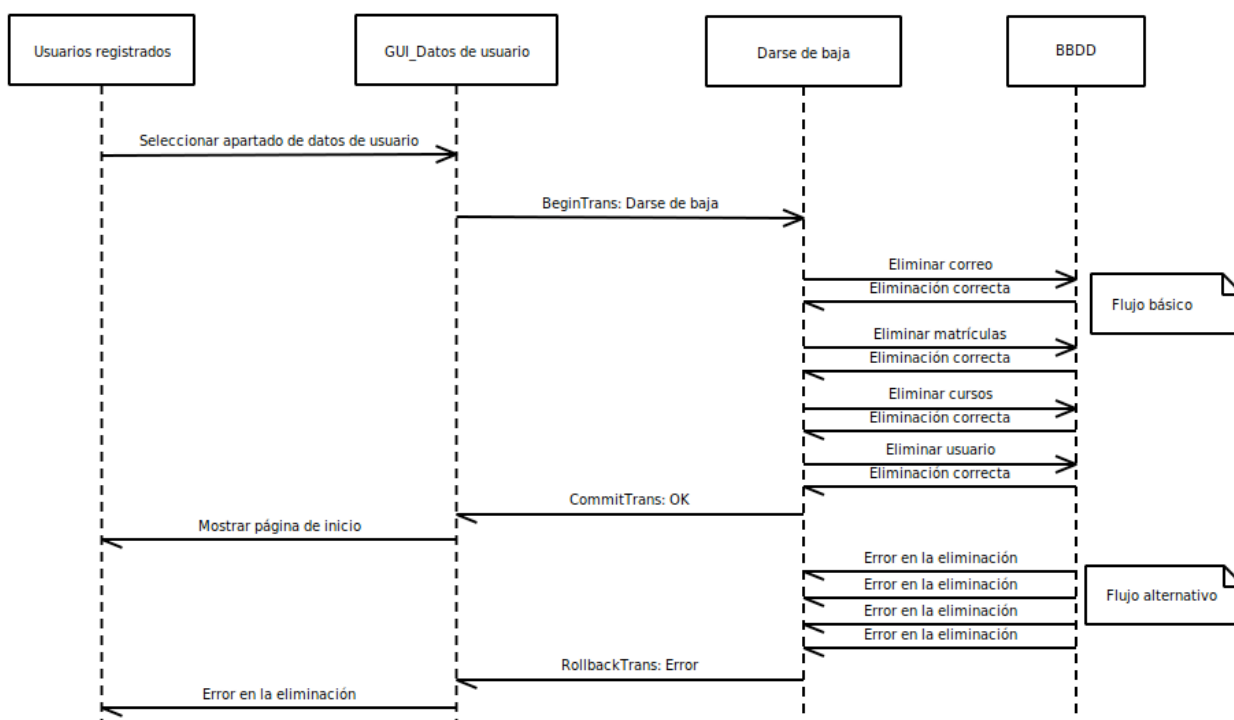


Figura 40 – Diagrama de secuencia, iteración 3.9.

SUBSISTEMA: Correo interno

ENVIAR

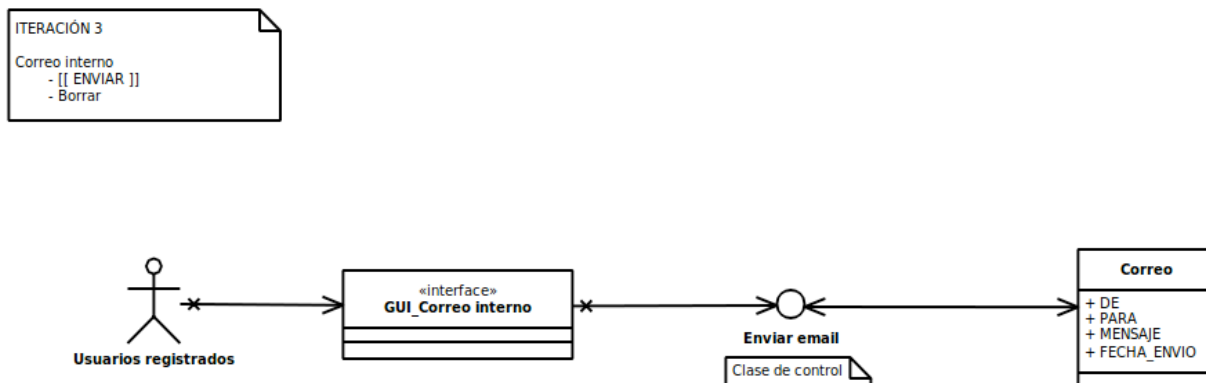


Figura 41 – Diagrama de clases, iteración 3.10.

Todo usuario registrado tendrá la opción de enviar correos a otros usuarios de la aplicación para comunicarse entre ellos. Desde el interfaz de correo podrá enviar mensajes privados a otros miembros de la comunidad y de la misma manera, gestionar su correo.

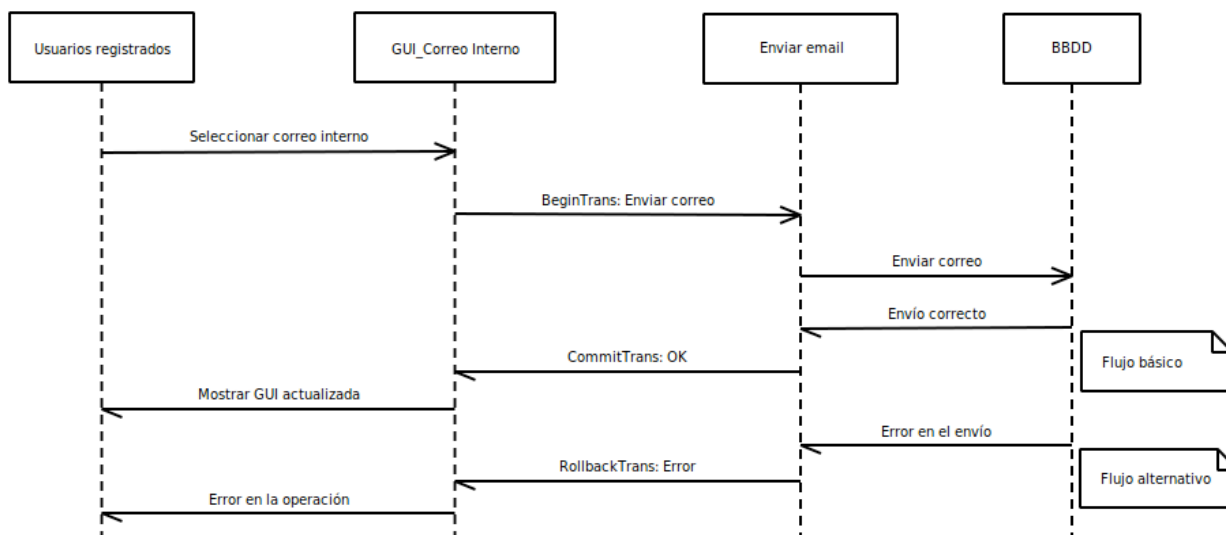


Figura 42 – Diagrama de secuencia, iteración 3.10.

SUBSISTEMA: Correo interno

BORRAR

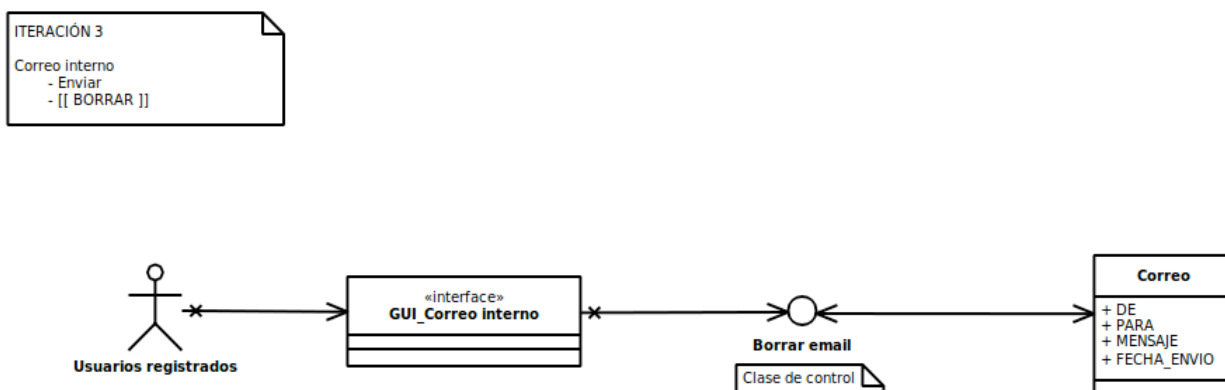


Figura 43 – Diagrama de clases, iteración 3.11.

Todo usuario registrado tendrá la opción de enviar correos a otros usuarios y de gestionar su correo, borrando aquellos mensajes que desee. Bastará con seleccionar aquellos que desee eliminar, o seleccionar la opción de eliminar todos, y el sistema realizará las acciones especificadas.

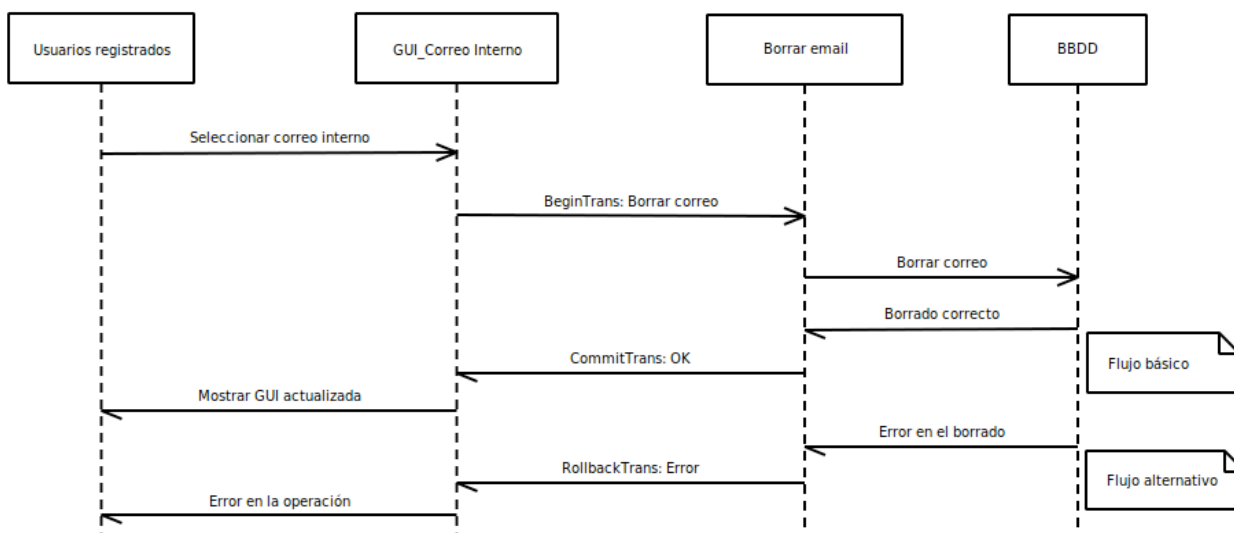


Figura 44 – Diagrama de secuencia, iteración 3.11.

SUBSISTEMA: Gestión de mantenimiento

MANTENIMIENTO DE USUARIOS

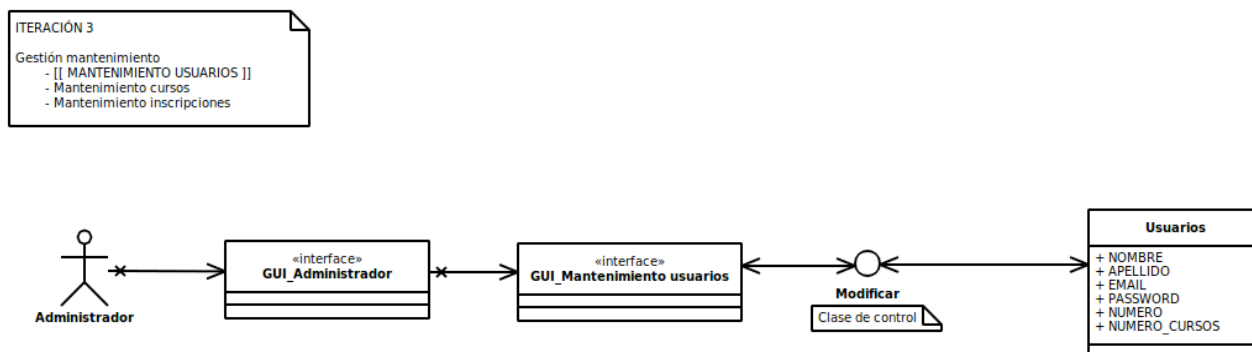


Figura 45 – Diagrama de clases, iteración 3.12.

El usuario Administrador, tendrá la opción de gestionar y mantener el sistema desde el interfaz creado para ello. Accediendo a éste, podrá llevar, por ejemplo, un mantenimiento de los usuarios participantes en el sistema. Podrá modificar sus datos, o eliminarlos del todo.

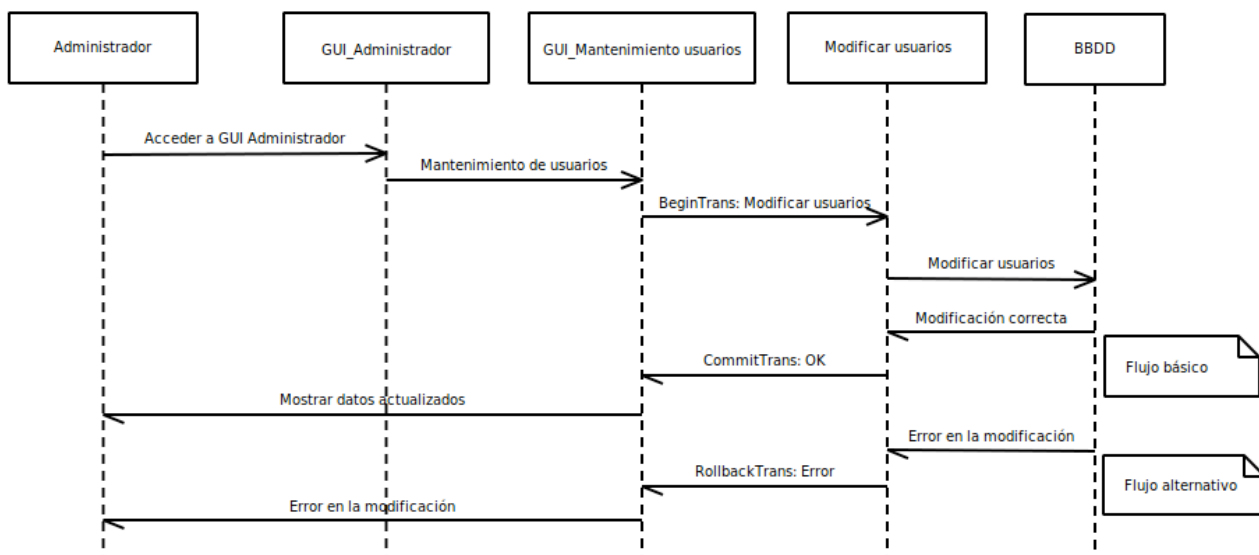


Figura 46 – Diagrama de secuencia, iteración 3.12.

SUBSISTEMA: Gestión de mantenimiento

MANTENIMIENTO DE CURSOS

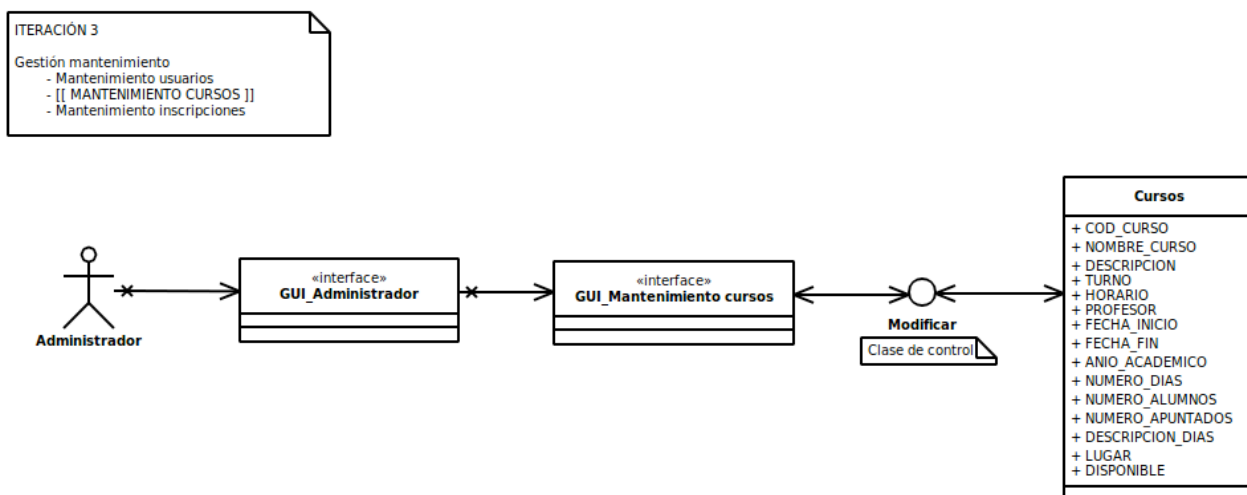


Figura 47 – Diagrama de clases, iteración 3.13.

De la misma forma, el Administrador del sistema podrá realizar el mantenimiento y gestión de los cursos registrados en el sistema. Bastará con acceder a la parte de mantenimiento de cursos, desde el interfaz de Administrador, y gestionar los cursos que se crean oportunos, así como la inserción o eliminación de los mismos.

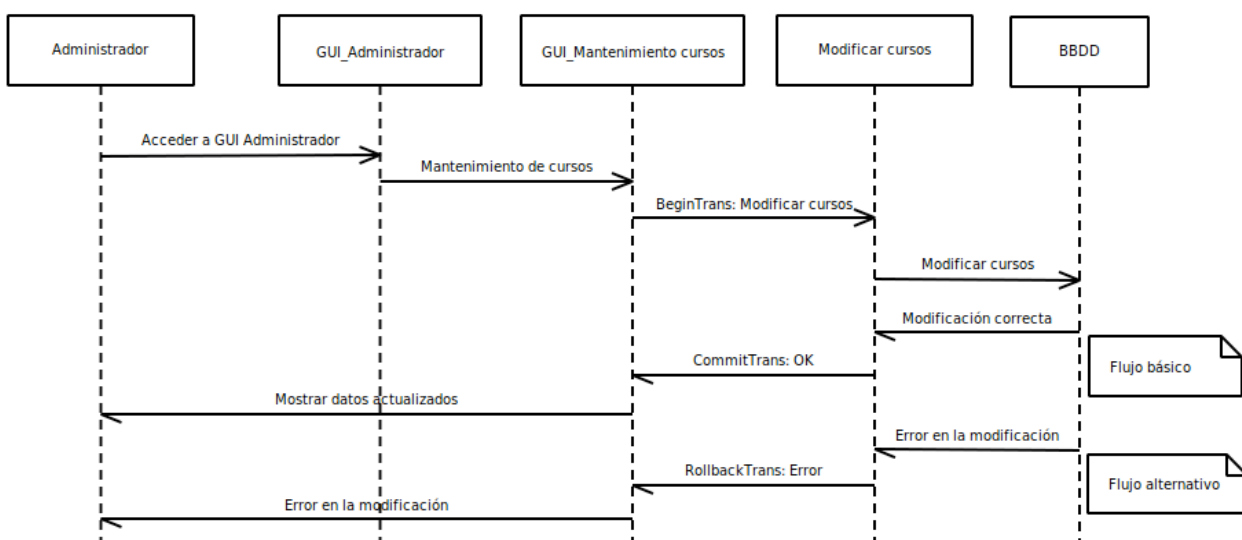


Figura 48 – Diagrama de secuencia, iteración 3.13.

SUBSISTEMA: Gestión de mantenimiento

MANTENIMIENTO DE INSCRIPCIONES

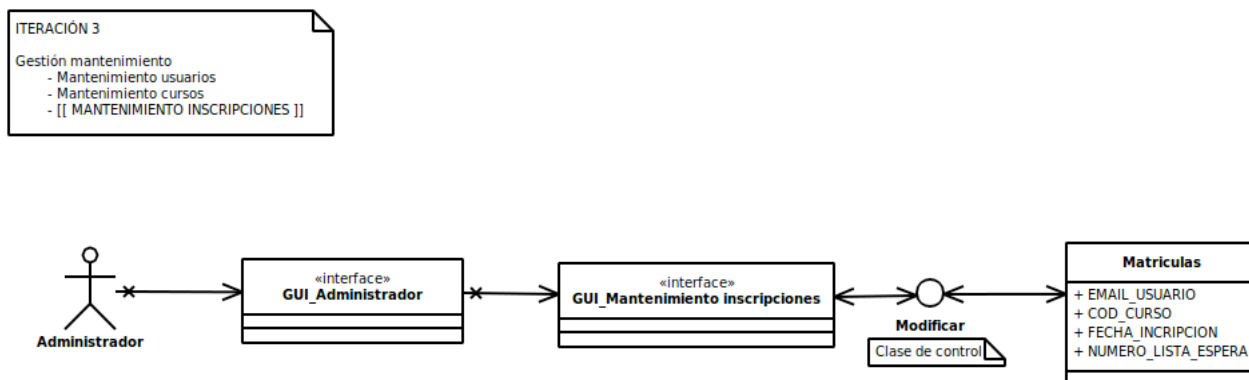


Figura 49 – Diagrama de clases, iteración 3.14.

Así mismo, el Administrador del sistema será capaz de llevar el mantenimiento y gestión de las diferente inscripciones realizadas a los cursos registrados en el sistema. Bastará con acceder a la parte de mantenimiento de inscripciones, desde el interfaz de Administrador, y gestionar aquellas que se consideren oportunas.

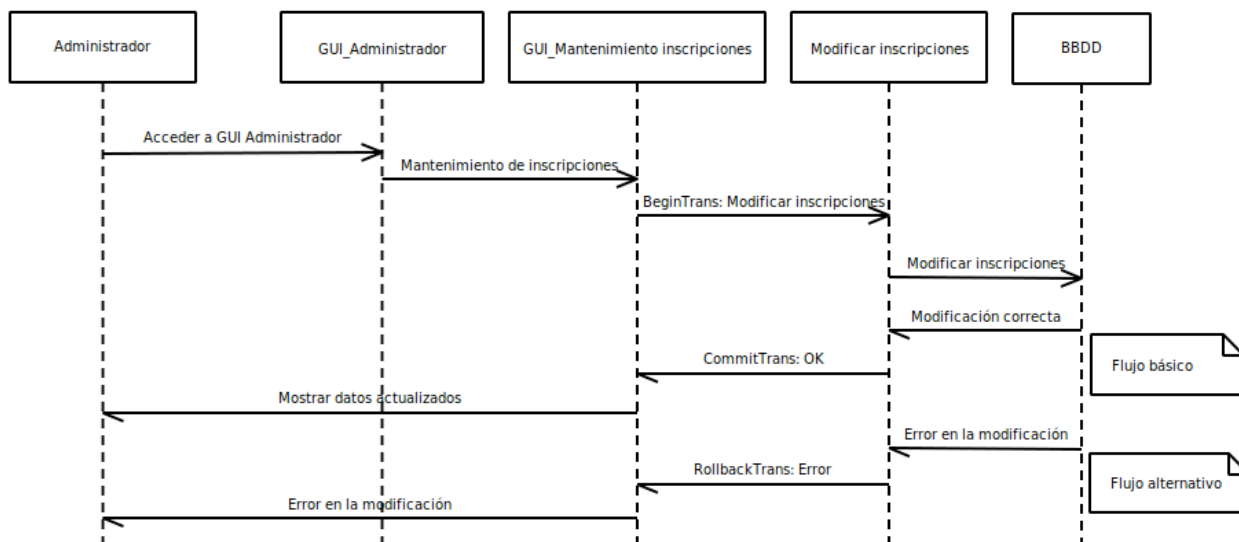


Figura 50 – Diagrama de secuencia, iteración 3.14.

SUBSISTEMA: Gestión de estadísticas

DATOS ASISTENTES A CURSOS

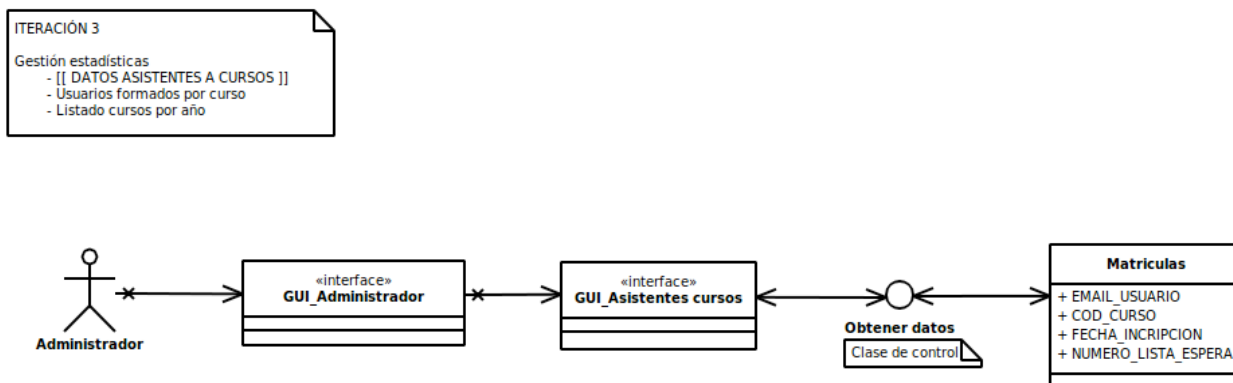


Figura 51 – Diagrama de clases, iteración 3.15.

Por otra parte, el usuario Administrador, tendrá la opción de gestionar y obtener diferentes estadísticas del sistema desde el interfaz de administración. Accediendo a éste, podrá llevar a cabo la obtención de, en este caso, los datos de los usuarios asistentes a los diferentes cursos.

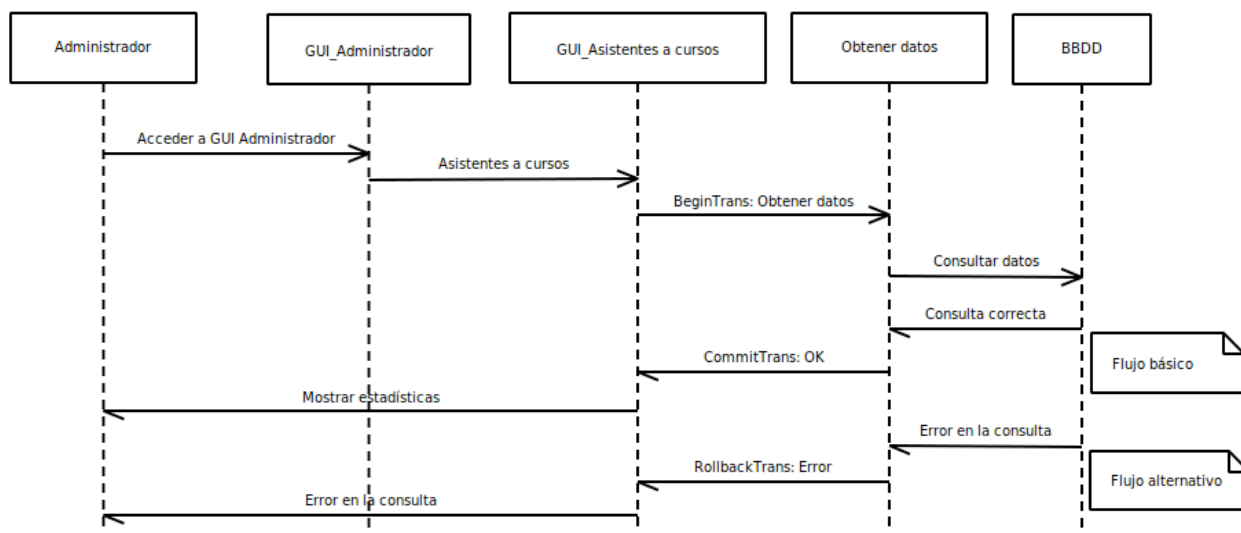


Figura 52 – Diagrama de secuencia, iteración 3.15.

SUBSISTEMA: Gestión de estadísticas

USUARIOS FORMADOS POR CURSO

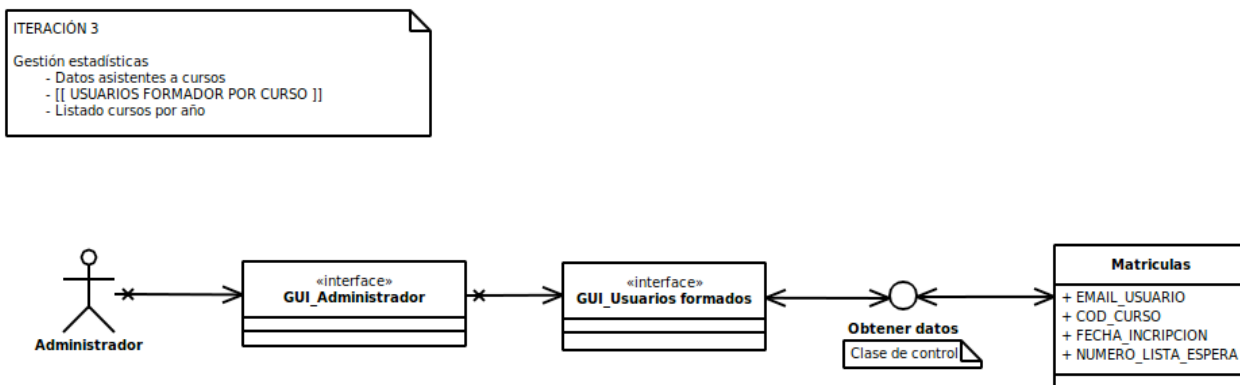


Figura 53 – Diagrama de clases, iteración 3.16.

Así mismo, el Administrador del sistema será capaz de obtener una lista de usuarios formados en cada curso. Bastará con acceder a la parte de estadísticas del sistema de administración, y obtener los datos que se consideren necesarios.

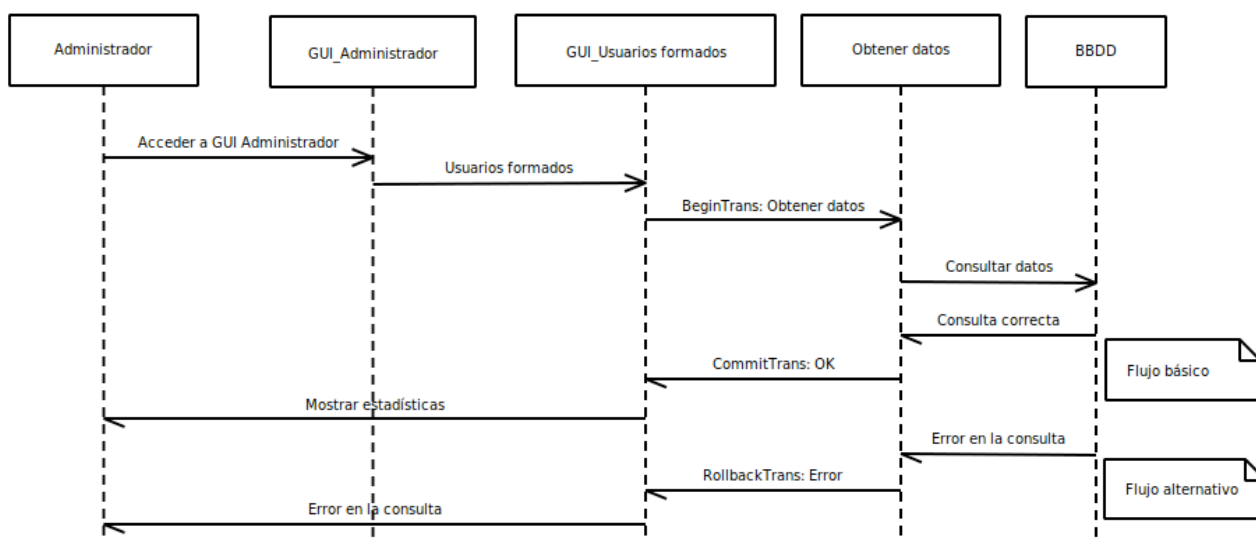


Figura 54 – Diagrama de secuencia, iteración 3.16.

SUBSISTEMA: Gestión de estadísticas

LISTADO DE CURSOS POR AÑO

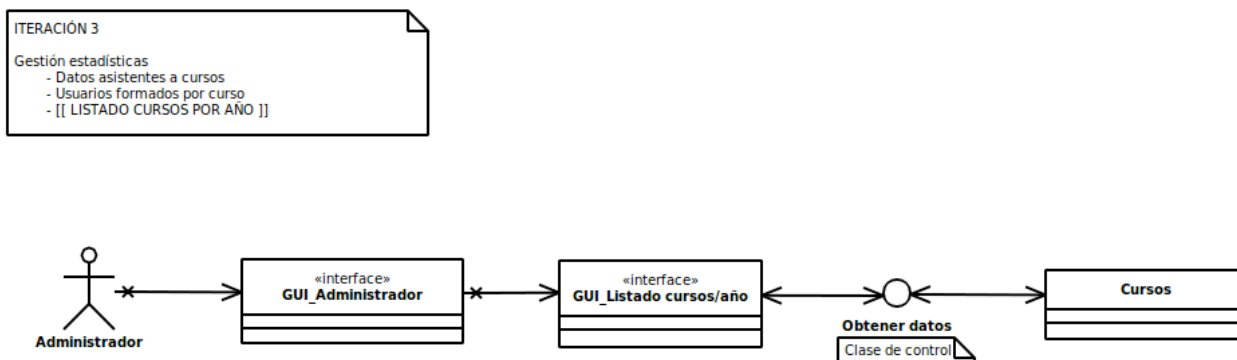


Figura 55 – Diagrama de clases, iteración 3.17.

De la misma forma, el Administrador del sistema será capaz de obtener listados de cursos impartidos por año, para llevar un recuento de los mismos. Bastará con acceder a la parte de estadísticas del sistema de administración, y obtener los datos oportunos.

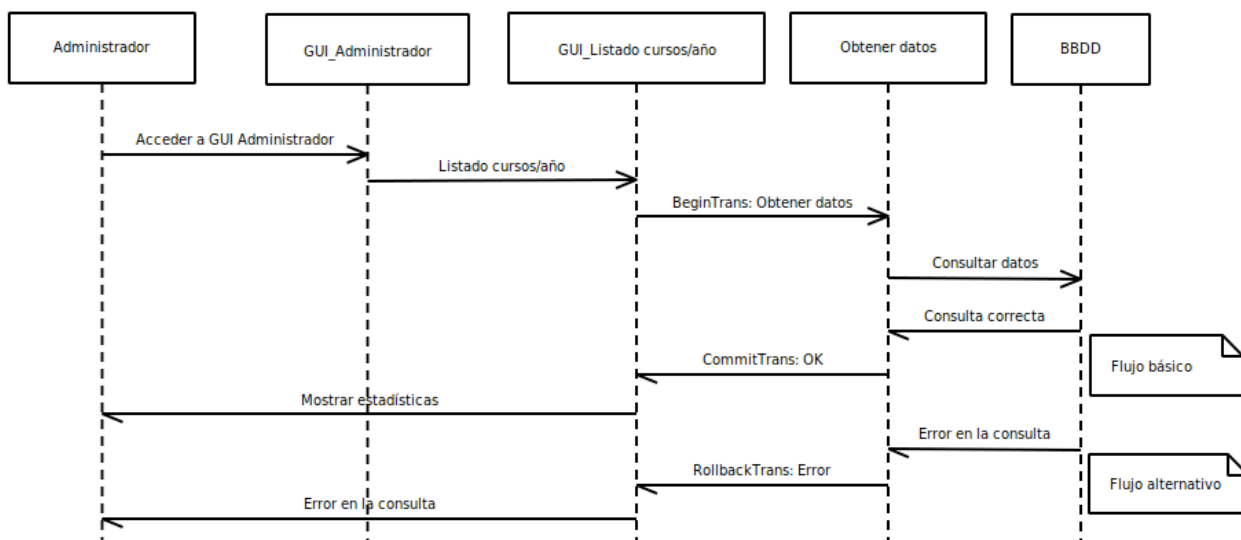


Figura 56 – Diagrama de secuencia, iteración 3.17.

Iteración 4

SUBSISTEMA: Colaborar espacio curso

COMENTAR

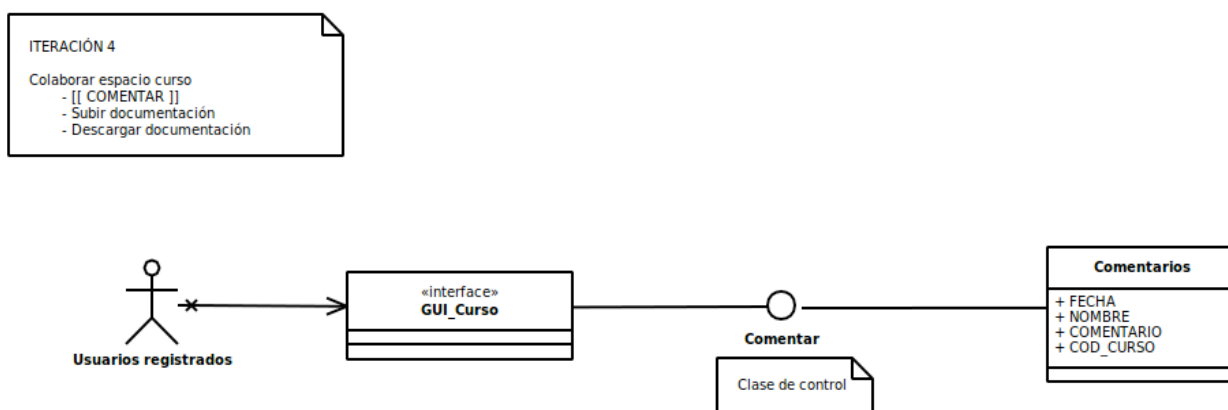


Figura 57 – Diagrama de clases, iteración 4.1.

Todo usuario registrado e inscrito en un curso determinado, podrá colaborar en el espacio reservado para éste. Bastará con acceder al sistema, y desde el interfaz de información del curso acceder al espacio reservado para éste. En él, el usuario podrá escribir los comentarios que considere oportunos, para valorar, dar su opinión u ofrecer datos y conocimientos al resto de usuario matriculados en dicho curso.

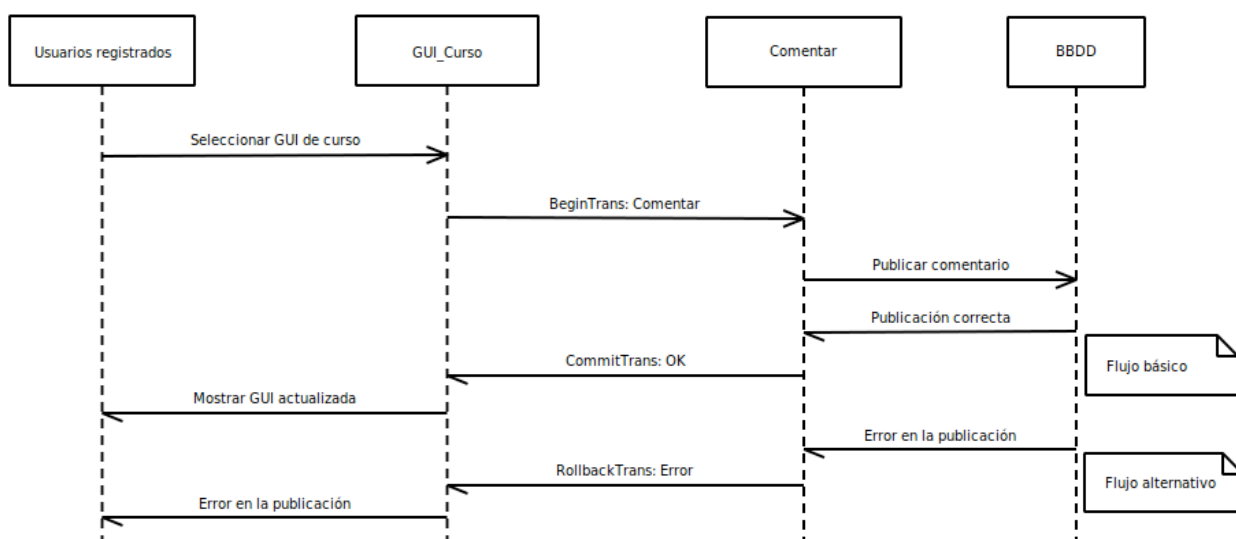


Figura 58 – Diagrama de secuencia, iteración 4.1.

SUBSISTEMA: Colaborar espacio curso

SUBIR DOCUMENTACIÓN

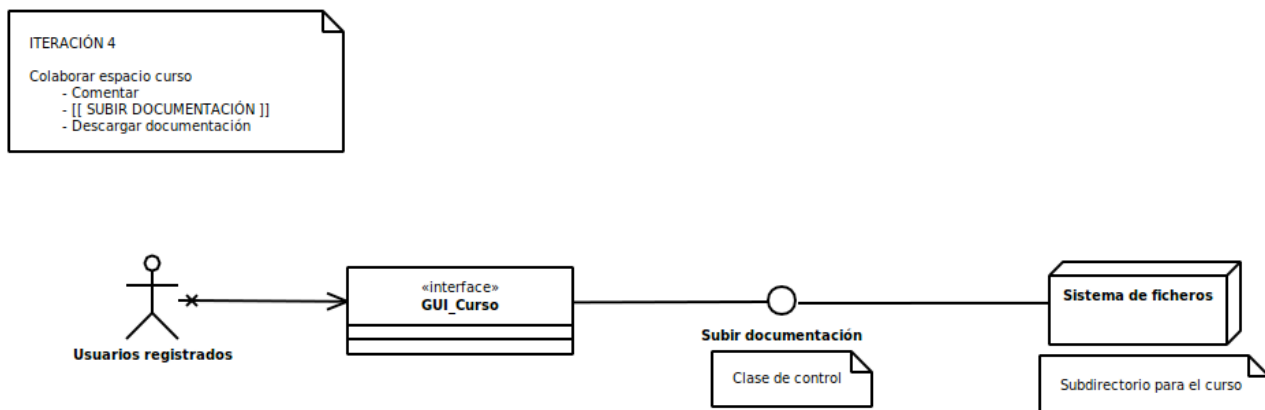


Figura 59 – Diagrama de clases, iteración 4.2.

Todo usuario registrado e inscrito en un curso determinado, podrá colaborar en el espacio reservado para éste teniendo la opción de subir la documentación que se considere necesaria o de ayuda al resto de usuarios inscritos. Esta opción podrá ser manejada tanto por el creador del curso, que suba su documentación para impartir el curso de forma organizada, así como por el resto de usuario inscritos, que podrán colaborar con documentación de interés en relación con la materia del curso.

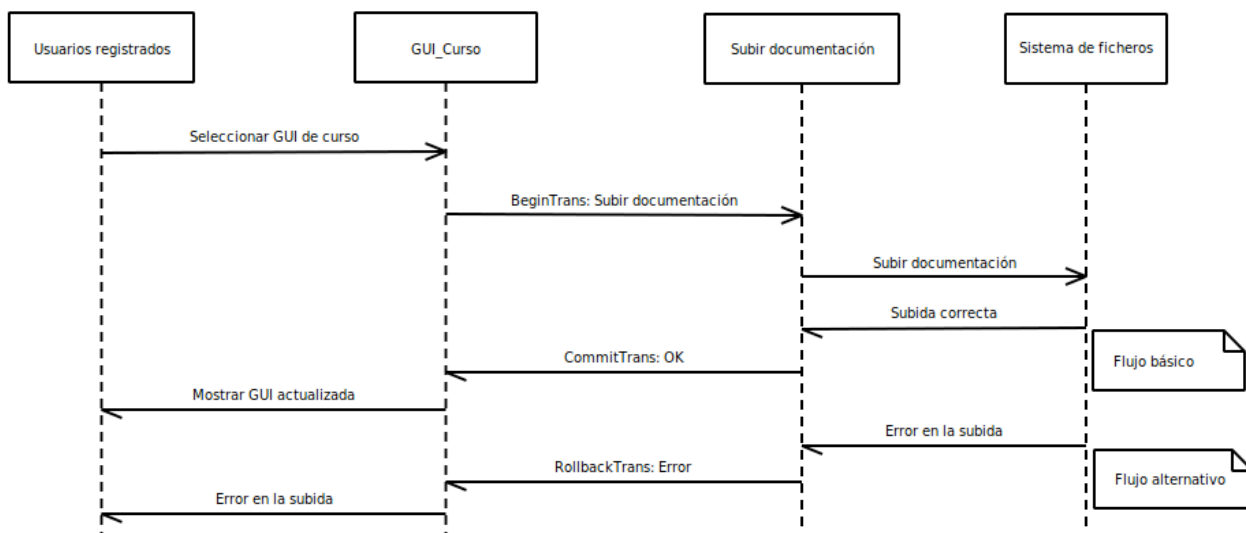


Figura 60 – Diagrama de secuencia, iteración 4.2.

SUBSISTEMA: Colaborar espacio curso

DECARGAR DOCUMENTACIÓN

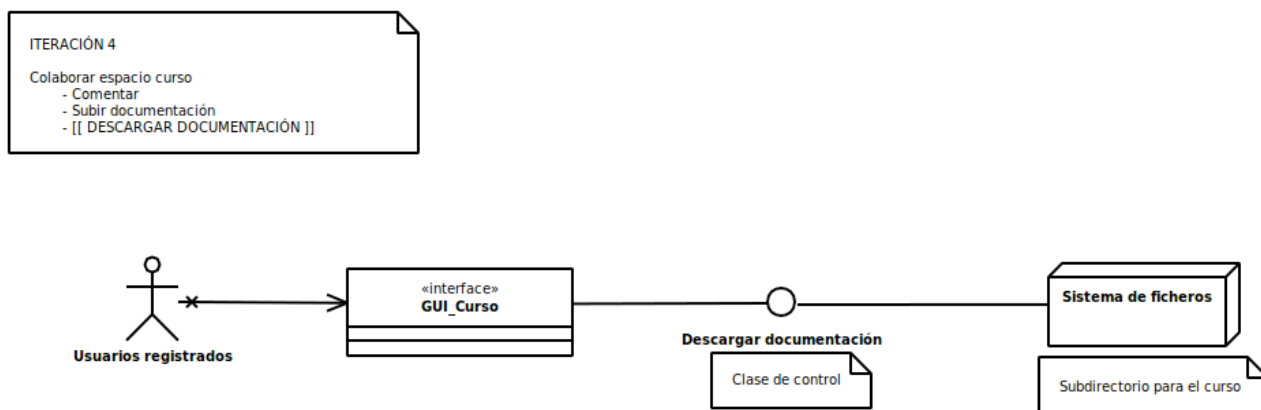


Figura 61 – Diagrama de clases, iteración 4.3.

Así mismo, todo usuario inscrito en un curso, podrá hacer uso de dicha documentación añadida al espacio del curso, de forma que el sistema le permitirá descargarla en el momento que desee.

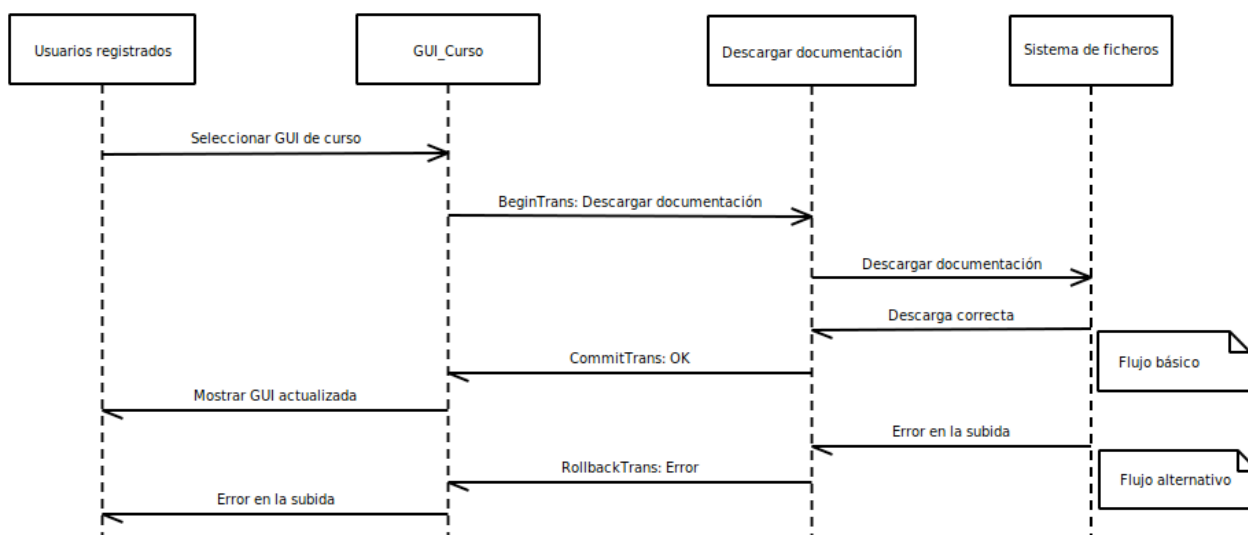


Figura 62 – Diagrama de secuencia, iteración 4.3.

5.2 Identificación de entidades a definir

Identificación de clases

Básicamente, serán necesarias seis tipos de entidades diferentes, según los requisitos especificados en la fase de análisis anterior. Mediante estas entidades, se permitirá un futuro diseño de los diagramas de clases y del esquema de entidad-relación necesarios en el proyecto. Estas seis entidades son:

1. Curso
2. Matrículas
3. Usuarios
4. Correo
5. Comentarios
6. BBDD

La entidad “Curso” dispondrá de los elementos necesarios para poder definir bien dicha clase, de forma que se puedan cumplir los requisitos y objetivos especificados con anterioridad. Será necesario en primer lugar definir un atributo que distinga los diferentes cursos entre ellos. Es decir, será básico el definir un código de curso exclusivo para cada uno.

Además, deberá tener de la misma forma, una serie de atributos que sirvan de descripción del curso, y que lo identifiquen, para después también poder mostrar dentro de la aplicación dicha información. Estos atributos serán aquellos campos del formulario que el usuario deba rellenar para la creación de sus cursos. Se especifican a continuación los definidos en una primera etapa. En una situación futura podría verse ampliada dicha lista, en caso de querer añadir datos extra o en caso de resultar necesario el aportar más especificaciones referentes a los cursos de cada usuario.

Cada curso dispondrá por tanto de los siguientes detalles:

Nombre, descripción del curso, turno en el que se llevará a cabo la formación, horario y profesor, fechas de duración del curso, año en el que se desarrollará la actividad, número de plazas y de alumnos inscritos al mismo, y finalmente lugar donde se impartirá la docencia.

Por otro lado, la entidad “Usuario”, describirá a cada usuario que se registre en la aplicación. Como se ha comentado anteriormente, cada usuario vendrá identificado inequívocamente por su email, atributo principal por tanto. Además, se especificó que cada usuario podría cambiar sus datos personales, como nombre, apellido o contraseña utilizada en el login de la aplicación. Así pues, estos tres campos entraran también dentro de la entidad “Usuarios”. Finalmente, se añadirá un número que identifique el número de usuario registrado, y la cantidad de cursos que imparte como docente.

Finalmente, disponemos de las entidades “Matrículas”, “Correo” y “BBDD”. Esta última implementará todas aquellas gestiones que se lleven a cabo con la base de datos del sistema, que se especificaran a lo largo del desarrollo de la aplicación.

Sin embargo, la clase “Matrículas” permite definir su estructura en este momento del diseño. Al describir las inscripciones de los usuarios en los diferentes cursos, será necesario que posea los atributos de email de usuario y código de curso, que identificará la tupla que se cree. Además, para informar al usuario de su estado se añadirá el atributo fecha de inscripción y número de lista de espera, en caso de que dicho usuario no haya obtenido plaza, y se encuentre en la lista de espera.

Por otro lado, la entidad “Correo” será necesario que almacene la información correspondiente al emisor y receptor del mensaje, así como la fecha de envío del mismo y su contenido.

Por último, la entidad “Comentarios” deberá almacenar el curso al cual pertenece el comentario realizado, el nombre de la persona que lo realizó, la fecha del momento del comentario, y por su puesto, el texto del mismo.

Así, tras definir las entidades básicas necesarias, obtenidas tras el análisis de los requisitos del sistema, se tendría el esquema representado en la siguiente imagen, de forma que sea posible avanzar en la fase de diseño de la aplicación.

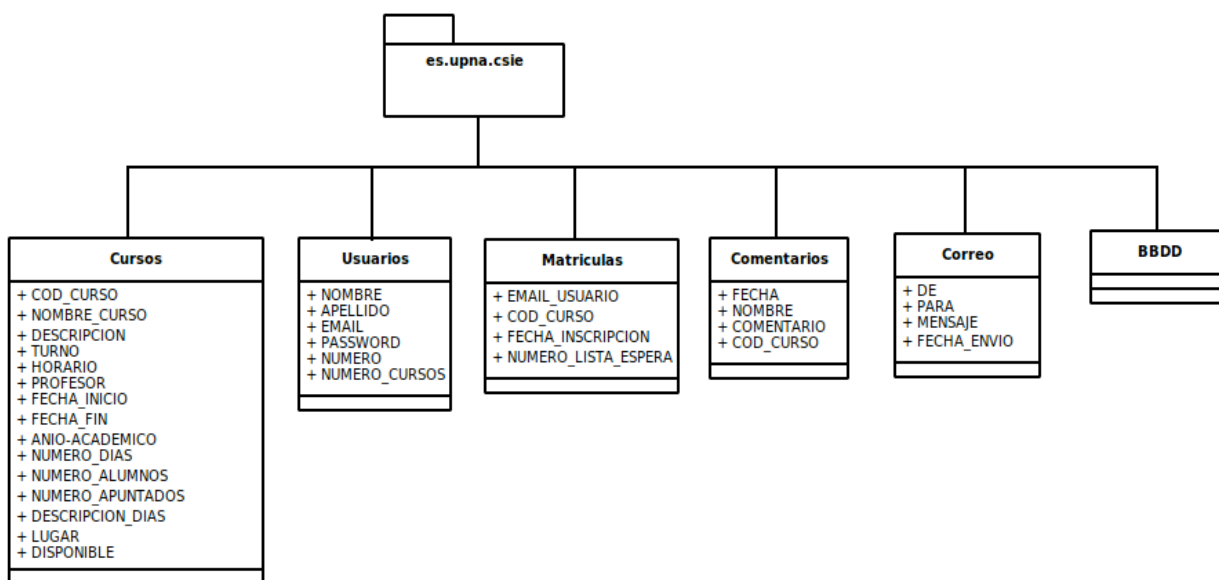


Figura 63 – Clases del sistema.

Asociación y generalización de clases

Principalmente se encuentran las clases “Usuarios” y “Cursos”, que son las relaciones principales del sistema. En primer lugar, un usuario puede crear todos los cursos que desee, y éstos tendrán una única persona responsable de su impartición. Por otro lado, todo usuario puede inscribirse a los cursos que desee.

Además, cuando se produzcan nuevas inscripciones a cursos se almacenarán como un nuevo objeto matrícula, que además de registrar qué usuario se inscribe a qué curso, registrará la fecha y el número de lista de espera en el que se encuentra el usuario para acceder a ese curso, en caso de no obtener plaza para el mismo.

Por otro lado, se encuentra el sistema de correo interno, que permite a los usuarios comunicarse entre ellos dentro del sistema. Un usuario puede enviar y recibir todos los correos que desee, y un correo puede enviarse a una única persona, o por el contrario, es posible enviar el mismo correo a todos los usuarios inscritos en un mismo curso. Esto será una opción del usuario que imparta dicha formación.

Finalmente, cada curso tendrá un apartado propio para documentación y valoraciones del mismo, donde podrán dejarse comentarios o valoraciones anónimas acerca de la visión o estado del curso de formación.

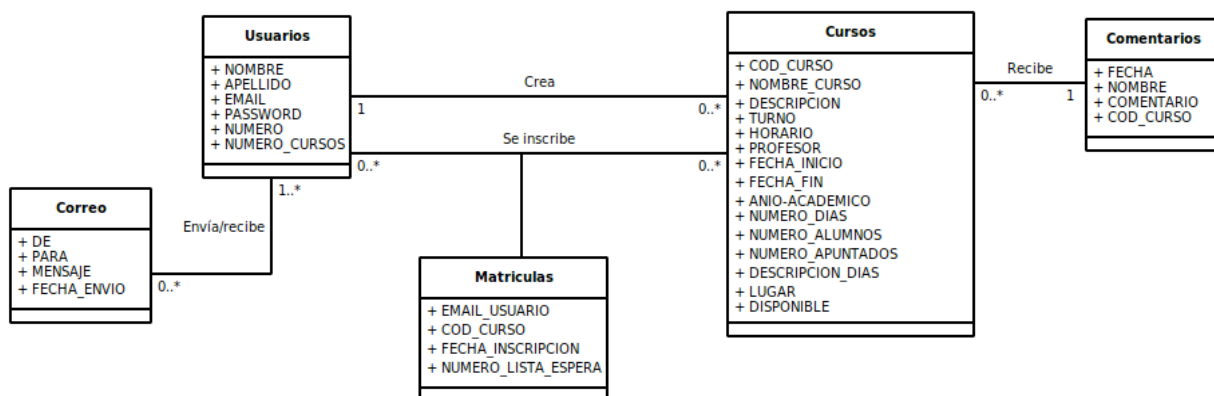


Figura 64 - Asociación de clases.

Por último, puede obtenerse una generalización del mismo esquema, donde el Usuario puede jugar el papel único de usuario registrado, o por el contrario poseer también privilegios de administrador.

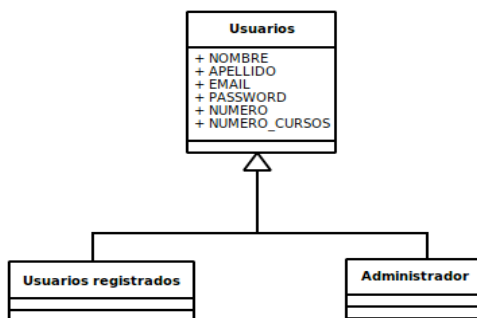


Figura 65 – Generalización de clases

5.3 Diagrama Entidad/Relación

A continuación, en la figura 67 se muestra el modelo final de entidad relación desarrollado para implementar de forma correcta la base de datos.

Su desarrollo se basa en los apartados anteriores, en los que se han descrito las clases a implementar y sus atributos, añadiendo alguno más que haya podido resultar necesario a la hora de la implementación del modelo, o en base a posibles usos futuros no especificados en el funcionamiento de la actual aplicación.

La entidad “BBDD” no se implementa en el modelo E/R, ya que será una entidad que posea únicamente aquellos métodos necesarios para la comunicación y manejo de la base de datos en sí, pero no será necesaria su representación en la misma como una tabla más.

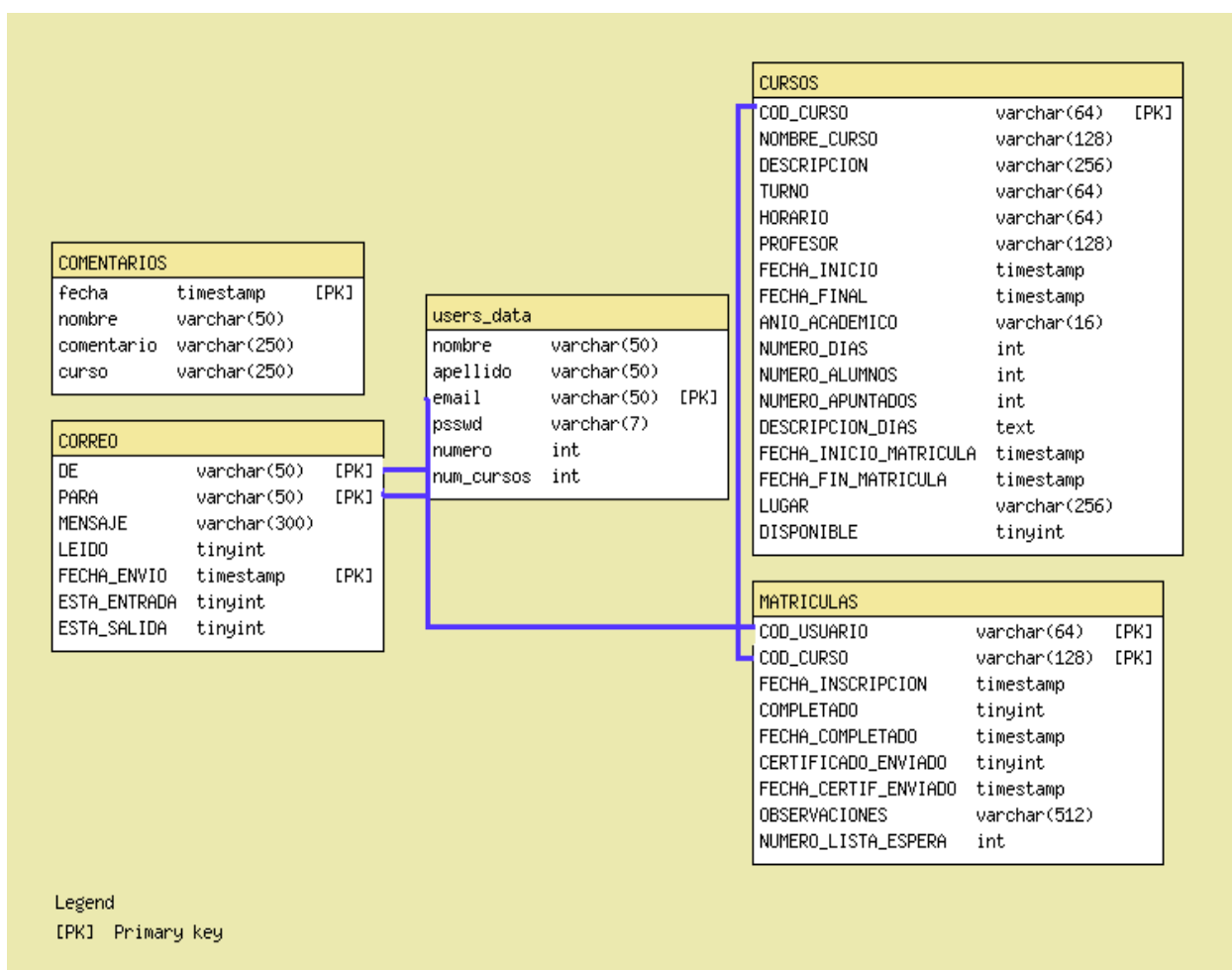


Figura 66 – Diagrama Entidad-Relación del sistema

6. DESARROLLO E IMPLEMENTACIÓN

El código de la aplicación es muy extenso como para explicarlo todo paso a paso. Dado que se basa en tecnologías bastante conocidas, como la programación en lenguaje java o el uso de bases de datos (MySQL), no resultaría tampoco excesivamente interesante conocer todos los detalles de la implementación de la aplicación en sí.

Por tanto, en este apartado se explicarán los puntos más importantes del desarrollo de la herramienta. Esto se resume en diez puntos, expuestos más detalladamente a continuación.

6.1 Aplicación basada en Servlets

La aplicación está desarrollada mediante el uso de Servlets en lenguaje java. Los servlets y Java Server Pages (JSPs) son métodos de creación de páginas web dinámicas alojadas en un servidor, y usan el lenguaje Java.

Cada servlet se ejecuta en su propio contexto, pero no se ejecuta cada vez que recibe una petición, sino que persiste de una a la siguiente. De esta manera, no se pierde tiempo en invocarlo. Además, un servlet puede manejar múltiples peticiones de clientes. Su persistencia le permite también hacer una serie de acciones de forma eficiente, como son la conexión a bases de datos y el manejo de sesiones.

Debido a que estas dos operaciones son muy requeridas en el funcionamiento de la aplicación, el sistema se basa en el desarrollo de servlets y archivos jsp, que son un modelo específico de servlets. Son archivos que contienen código html, pero que además permiten incorporar código java.

El ciclo de vida del funcionamiento de un servlet es el siguiente.

1. El cliente solicita una petición a un servidor vía URL.
2. El servidor recibe la petición.
 1. Si es la primera, se utiliza el motor de Servlets para cargarlo y se inicializa.
 2. Si no lo es, la petición se convierte en un nuevo hilo.
3. Se procesa la petición del cliente devolviéndole a éste el resultado correspondiente.
4. Cuando se apaga el motor del servlet, éste se destruye.

Un servlet implementa el interfaz de java javax.servlet.Servlet o hereda de la clase “HttpServletRequest”. Al implementar el interfaz, un servlet es capaz de interpretar aquellos objetos de tipo “HttpServletRequest” y “HttpServletResponse”, que contienen toda la información de la página que invocó al servlet.

Además, es necesario sobrescribir los métodos:

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response);  
protected void doGet(HttpServletRequest request, HttpServletResponse response);  
protected void doPost(HttpServletRequest request, HttpServletResponse response);
```

Las ventajas de trabajar con servlets son bastantes, y más respecto a métodos más antiguos que han quedado en desuso, como CGI (*Common Gateway Interface*), implementados habitualmente en perl, c o visual basic.

Eficiencia.

Los Servlets se ejecutan en la máquina Virtual Java, que una vez se inicializa, permanece arrancada, y cada petición es manejada por un thread Java de poco peso. De esta forma similar, si hay N peticiones simultáneas, hay a su vez N threads, pero sólo una copia del servlet solicitado. Además, permite mantener sesiones y por tanto, conexiones a la base de datos dentro de éstas.

Conveniencia.

Hoy en día, se tiene bastante conocimiento de lenguaje java, y se encuentra bastante desarrollado. Por tanto, los Servlets tienen una gran infraestructura para análisis y decodificación de datos de formularios HTML, para leer y seleccionar cabeceras HTTP, manejar sesiones, y muchas otras opciones.

Potencia.

Así mismo, los servlets pueden comunicarse directamente con el servidor Web. Esto simplifica las operaciones que implican la búsqueda de imágenes y archivos almacenados en situaciones standards. También pueden mantener información de solicitud en solicitud, simplificando el manejo de sesiones y manteniendo el caché de cálculos anteriores.

Portable.

Los Servlets están implementados en lenguaje Java y siguen un API estandarizado, por tanto, están soportados en la mayoría de los servidores Web.

Económico.

Hay un número de servidores Web gratuitos y de correcto funcionamiento. Sin embargo, con la excepción de Apache, que es gratuito, la mayoría de los servidores Web comerciales son relativamente caros.

Por otro lado, existen también desventajas de los servlets (clases .java) respecto a otro tipo de servlets, como por ejemplo los jsp. En principio, ambos pueden ofertar lo mismo, sin embargo, la mayor ventaja de un jsp frente a un servlet .java, es básicamente que te permite escribir, y sobre todo, modificar, el código HTML de una manera mucho más sencilla, y sin tener que incorporar miles de sentencias “println” para generar dicho código.

Por otro lado, en el campo del diseño es más eficiente, ya que pueden separarse el formato del contenido, y mientras éste se desarrolla por una parte, simultáneamente por otra, puede desarrollarse el diseño de la página Web.

6.2 Uso de formularios para la creación de cursos

La creación de cursos, el registro o el login de usuarios, está implementado mediante formularios web. Como se ha comentado, cada servlet hereda de la clase `HttpServlet`, y por tanto, es capaz de procesar las peticiones GET y POST que se realizan en la aplicación.

Por ejemplo, a la hora de solicitar una nueva contraseña, se utiliza también un formulario web. Su implementación es la siguiente.

```
out.println("<br><form name='nuevaPsswd' id='nuevaPsswd' action='NuevaPasswordServlet'  
method='POST' >"+  
    "<p>" + msg.getString("nuevaPassword1") + "</p>"+  
    "<p><input name='email' type='text' ></input></p>"+  
    "<p><input name='enviar' type='submit'  
value='"+msg.getString("enviar")+"'></p>"+  
    "</form>");
```

Esta petición se envía al servlet “NuevaPasswordServlet”, que procesa la información enviada por el formulario para ejecutar las acciones correspondientes. Para ello, en el método “processRequest” de la clase “HttpServlet”, se manejan los objetos “request” y “response” para obtener la información enviada a través del html.

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)  
{  
    ...  
    //Obtener el parámetro del campo "email" del formulario.  
    String nameUser=request.getParameter("email");  
    ...  
}
```


6.3 Acceso mediante registro de usuarios

Lo primero es mencionar que el sistema funciona como una comunidad privada, por tanto, todo usuario que decida pertenecer a ella, deberá registrarse para poder acceder. Por ello, habrá que crear un sistema que evite que usuario sin permisos accedan a la herramienta.

Para ello se hace uso de métodos de autenticación y comprobación de usuarios. Cuando un usuario accede al sistema mediante el formulario de acceso a la aplicación, se hace una consulta a la base de datos comprobando que existe ese usuario, y que la contraseña introducida en el campo correspondiente es igual a la que dicho usuario posee en el registro de la base de datos.

```
//Comprobar validez del usuario y psswd.
correcto=miBBDD.CompruebaUsuario(usuario, psswd);
//Registrar los datos de numero y nombre de usuario.
this.numUser=miBBDD.DevuelveNumUsuario(usuario);
this.nameUser=usuario;
//Atributo de la sesión.
session.setAttribute("nombreUsuario", this.nameUser);
```

Así mismo, se comprueba que los campos de acceso no sean nulos.

```
//USUARIO Y PSSWD son parametros obligatorios.
if ((usuario=this.CogeParametro("usuario", request))==null)
parametroVacio=true;
if ((psswd=this.CogeParametro("psswd", request))==null)
parametroVacio=true;
```

Además se comprueba en cada página de cual procede el usuario, y si tiene permisos para ello. Cada parte del sistema puede proceder de una serie de páginas, en caso de que no lo haga así, o que la dirección sea escrita a mano en la barra de direcciones, para intentar acceder ilegalmente, el sistema lo detectará y producirá un mensaje de error.

```
if( (request.getHeader("Referer")==null)||
(!request.getHeader("Referer").equals("DIR_PAGINAS_PRECEDENTES")))
{
    urlMal=true;
}
```

Por otro lado, a la hora de registrarse en el sistema, por seguridad, se tiene en cuenta que quien intenta realizar el registro es una persona. Por tanto, se ha creado un sistema de creación de captchas, que se imprime en la pantalla de inicio junto con el formulario de registro, y que el usuario debe introducir correctamente.

Este sistema se basa en crear una imagen, mediante números y letras, y colocar una serie de rayas encima de la imagen creada, de forma que se haga ilegible, y sólo una persona pueda reconocer los caracteres representados en la imagen.

En la pantalla de inicio se llama al fichero que crea la imagen del captcha, integrado en el código del formulario de acceso.

```
<p align="center">
  </img>
</p>
```

El sistema que crea los captchas, en primer lugar crea el string principal, después crea la imagen y le añade las letras y números, estableciendo los colores de todos ellos. Por último dibuja líneas por encima de la imagen para hacerla ilegible. Finalmente la devuelve imprimiéndola como una imagen en el formulario de acceso.

Además guarda el valor real de números y letras como un atributo de sesión para cotejarlo más adelante una vez que el usuario envíe los datos del formulario.

```
//Random para elegir aleatoriamente números y letras.
Random rdm=new Random();
int rl=rdm.nextInt();
String hash1 = Integer.toHexString(rl);
String capstr=hash1.substring(0,5);//Tamaño captcha=5

//Color de fondo, fuente.
Color background = new Color(255,255,255);
Color fbl = new Color(111,55,222);
Font fnt=new Font("Comic Sans MS",1,25);

//Se crea la imagen.
BufferedImage cpimg = new BufferedImage(width,height,BufferedImage.TYPE_INT_RGB);
Graphics g = cpimg.createGraphics();

//Se le añade el color de fondo, las letras y lo números.
g.setColor(background);
g.fillRect(0,0,width,height);
g.setColor(fbl);
g.setFont(fnt);
g.drawString(capstr,15,20);
g.setColor(background);
//Se dibuja líneas por encima.
g.drawLine(50,17,80,17);
g.drawLine(50,22,80,22);
[...]
g.drawLine(40,0,40,50);

//Crea sesión nueva, y guarda los caracteres del captcha en el atributo "key".
HttpSession nueva_session = request.getSession(true);
nueva_session.setAttribute("key",capstr);
```

6.4 Uso de sesiones para cada conexión.

Al tratarse de una comunidad web, restringida a aquellos usuarios registrados en el sistema, es necesario mantener una sesión de usuario mientras éste se encuentre activo. Por ello, se trabaja mediante sesiones, que además de permitir verificar la veracidad del perfil de usuario, permite almacenar datos de sesión necesarios para el correcto funcionamiento interno del sistema.

La primera vez que se accede al sistema, se crea una sesión nueva, que se irá utilizando posteriormente en el resto de pantallas, según el usuario vaya interactuando con el sistema.

```
//Crear una 'session' nueva para almacenar los datos de sesión del usuario  
HttpSession session=request.getSession(true);
```

En el resto de servlets del sistema, simplemente se obtendrá la sesión activa en el momento, para, más adelante, guardar y usar sus datos si esto fuese necesario.

```
//Obtener 'session' actual  
session=request.getSession();
```

Por otro lado, cada vez que se carga una nueva página, se cargan los atributos necesarios para el funcionamiento del sistema, generalmente el nombre de usuario, que permitirá cargar todas sus configuraciones personales, desde los cursos, inscripciones, hasta el correo o la agenda.

```
//Si la URL de acceso es correcta, se almacena el nombre de usuario  
if(!urlMal){  
    nameUser=session.getAttribute("nombreUsuario").toString();  
}
```

Además, como ya se ha visto, se utiliza para almacenar el valor del captcha mostrado en el registro de usuarios, para que, una vez enviados los datos del formulario, el sistema pueda comprobar si la cadena alfanumérica introducida por el usuario coincide con la almacenada en la sesión.

Así mismo, la sesión tiene un periodo determinado de validez, si se sobrepasa el mismo, ya sea porque el usuario está inactivo o por cualquier otra razón, la sesión caduca y se produce un error al intentar navegar de nuevo. En este caso está establecida en 20 minutos, ya que teniendo en cuenta el tipo de uso que va a darse a la aplicación, se considera suficiente para invalidar la sesión un periodo inactivo de 20 minutos.

```
session.setMaxInactiveInterval(1200);
```

Finalmente, al salir de la aplicación, se elimina la sesión.

6.5 Creación de correo interno.

En primer lugar, para lograr que funcione el sistema de correo interno en la aplicación, es necesario crear una tabla en la base de datos que se encargue de almacenar los datos correspondientes.

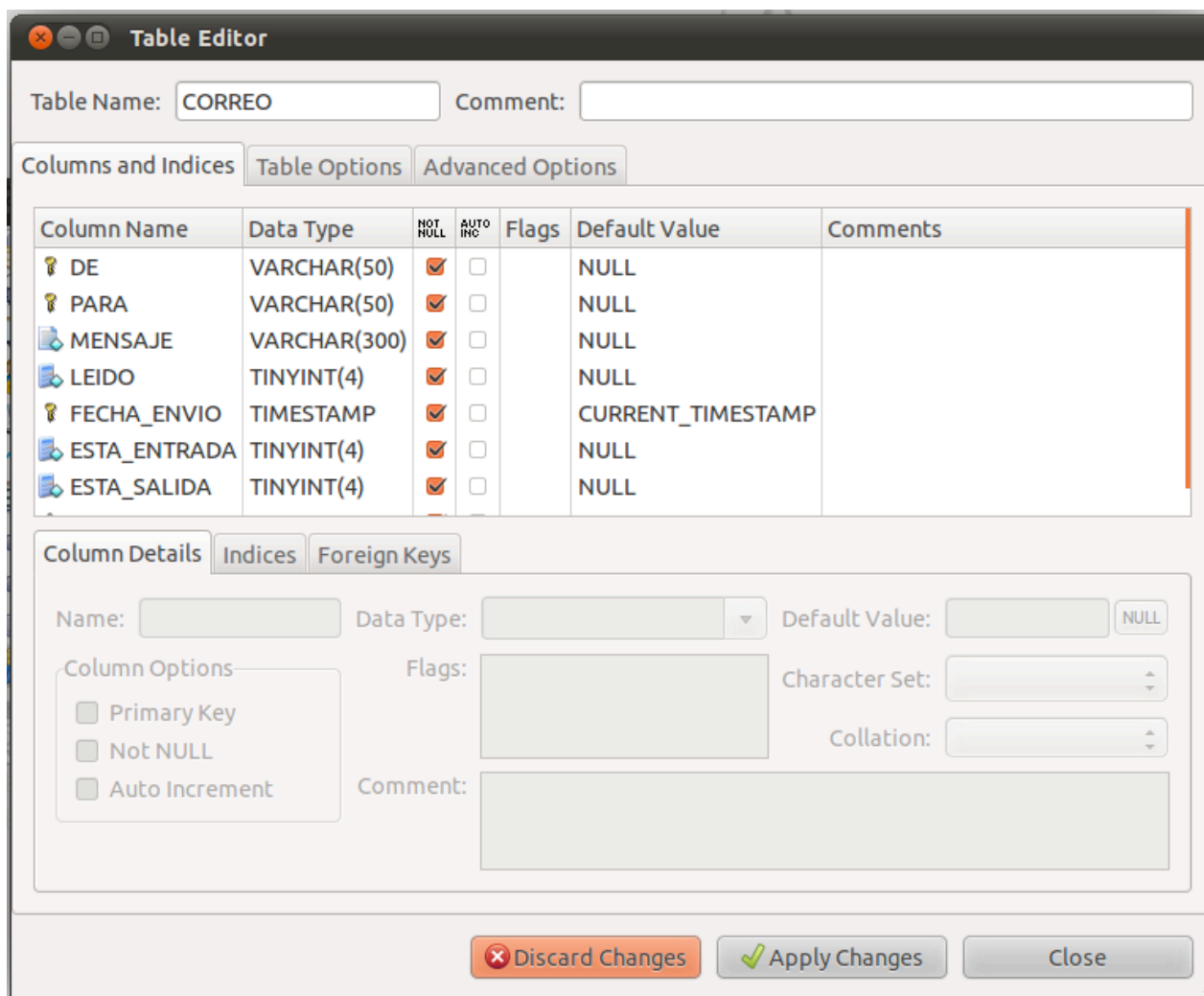


Figura 67 – Tabla ‘Correo’ de la base de datos.

Como se observa en la figura anterior, en dicha tabla se almacenan el emisor, el receptor, el mensaje y la fecha de envío, y tres variables de carácter booleano, que indican si el mensaje está leído por el receptor y si se encuentra en la bandeja de entrada, y por último, si se encuentra en la bandeja de salida del emisor.

De esta forma, se establecen unos indicadores para, más adelante, indicar al usuario si tiene correos nuevos o correos sin leer. Además, lo que se logra con estos datos, es saber si el mensaje debe mostrarse en la bandeja de entrada o salida del usuario, o si por el contrario, el usuario ya lo ha borrado con anterioridad.

Los mensajes permanecerán en la base de datos hasta el momento en que ambos usuarios, tanto el emisor como el receptor, hayan borrado el mismo de sus correspondientes bandejas. De este modo se evita una sobrecarga de la base de datos, permitiendo una limpieza automática, siempre que los usuarios lo permitan.

Por otro lado, centrándose en la parte visual, la programación del efecto de ocultar y mostrar el contenido de los mensajes, dependiendo de la acción del usuario, se realiza mediante dos sencillas funciones en javascript. Si el usuario presiona la imagen con el símbolo “+”, se mostrará el contenido del correo, o si por el contrario éste ya está mostrado en pantalla, se ocultará.

```
<script language='JavaScript' type='text/javascript'>
    function hideMails(){
        var lista=document.getElementsByName('hide');
        for (var i=0; i<lista.length; i++){
            lista[i].style.display='none';
        }
    }
    function aparecer(s){
        if(document.getElementById(s).style.display=='none'){
            document.getElementById(s).style.display='';
            var lista=document.getElementsByName('hide');
            for (var i=0; i<lista.length; i++){
                if(lista[i]!=document.getElementById(s))
                    lista[i].style.display='none';
            }
        }
        else{
            document.getElementById(s).style.display='none';
        }
    }
</script>
```

Cuando se presiona sobre un mensaje en el símbolo de “+”, este ejecuta la función “aparecer”. Si el estilo de ese mensaje indica que está oculto, de toda la lista de mensajes, los que están ocultos los deja igual, y el mensaje que se desea visualizar se muestra. Por otro lado, si se ha presionado el símbolo de mostrar/ocultar, y el mensaje se encuentra en pantalla, se oculta.

La función hideMails() se encarga de ocultar todos los mails al inicio de carga de la página, de forma que al entrar al correo, siempre se encuentren los mensajes ocultos.

El contenido de la celda donde se imprime el mensaje es el siguiente. Así, mediante la etiqueta 'hide' se identifica a los elementos que podrán ocultarse o mostrarse posteriormente.

```
<tr><td colspan='4'>
    <div id="" +id+"" name='hide' style='display:'>
    ...
</tr>
```

6.6 Envío de mails de confirmación e información a los usuarios participantes.

Para que el sistema de emails de la aplicación funcione de forma correcta se ha instalado postfix en el servidor, como se explica en la sección de configuraciones iniciales. La implementación para permitir enviar correos desde el sistema es muy sencilla. En primer lugar es necesario incluir el .jar de javax.mail.

Para que la programación resulte más eficaz y ordenada, se crea una clase para todos los modos relacionados con los emails que se enviarán en la aplicación. Esta clase tendrá como atributos la sesión de usuario, usuario y password del correo (en caso de que se quiera especificar, aunque en este caso no es necesario), un objeto de la base de datos para obtener los datos del usuario, y el nombre y email del usuario al que se enviará.

Además, los métodos de la clase variarán dependiendo del objetivo del mail, inscripción en un curso, cambio de contraseña, o eliminación o cancelación de la inscripción a un curso.

```
public class EnviarMail{  
  
    private final Properties properties = new Properties();  
    private Session session;  
    String user="", password="";  
    BBDD miBBDD;  
    String nombre,mail;
```

Los objetos de la clase se inicializan con los datos del servidor y del emisor. Se cargan las propiedades del protocolo de email y se especifican el host y puerto del servidor.

```
public EnviarMail(){  
    properties.setProperty("mail.transport.protocol", "smtp");  
    properties.put("mail.smtp.host", "127.0.0.1");  
    properties.put("mail.smtp.port",25);  
    //Añadir esta línea si se desea autenticación  
    //En nuestro caso no es necesario  
    //properties.put("mail.smtp.auth", "true");  
  
    session = Session.getDefaultInstance(properties,null);  
}
```

Para enviarlo habrá que crear el mensaje que se quiera en cada caso y mandar el email de la siguiente forma, especificando emisor, receptor y asunto.

```
MimeMessage message = new MimeMessage(session);  
  
//Creamos un InetAddress para añadir el 'Emisor' del email.  
//La dirección de correo es la especificada pero se mostrará como emisor "GESTDOC"  
InetAddress nia=new InetAddress("csie@csie-server.unavarra.es","GESTDOC");  
message.setFrom(nia);  
  
//Receptor, InetAddress con el email del receptor.  
message.addRecipient(Message.RecipientType.TO, new InetAddress(mail));  
//Asunto y texto del mensaje.
```

```
message.setSubject(msg.getString("asunto_mail"));
message.setText("CUERPO DEL MENSAJE");

//Transport a partir de la sesión actual, se conecta, envía y se desconecta.
Transport t = session.getTransport("smtp");
t.connect(properties.getProperty("mail.smtp.host"),user,password);
t.sendMessage(message, message.getAllRecipients());
t.close();
```

6.7 Recuperación de contraseña

Derivado del apartado anterior se crea éste. Si el usuario olvida la contraseña, desde el apartado de recuperación de contraseña, introducirá su dirección de correo, donde recibirá un email con la nueva contraseña, generada de manera aleatoria. Para generar dicha contraseña se siguen los pasos indicados a continuación.

```
public String enviarPassword(String email,HttpServletRequest request){
try
{
    String newPassword=null;
    Date d=new Date();
    long l;

    StringTokenizer st=new StringTokenizer(email);
    //La fecha actual en segundos
    l=d.getTime();
    //Nombre del email
    newPassword=st.nextToken("@");
    //Se recorta hasta 8 caracteres, si el token los sobrepasa.
    if(newPassword.length(>8)
        newPassword=newPassword.substring(0,8);
    //Se añaden caracteres especiales y el valor de la fecha actual
    newPassword=newPassword.concat("!-").concat(String.valueOf(l));
    ...
}
}
```

Las configuraciones de correo para enviar el mensaje son idénticas a las explicadas en el apartado anterior. En este caso variará tan sólo el contenido del mensaje. La generación de la contraseña es bastante simple, pero por otro lado, la fecha con la que se crea es la fecha exacta del momento de la petición, por lo que sería difícil duplicarla. Aún así, se recomienda al usuario que esa contraseña sea únicamente para el primer acceso al sistema, y se modifique acto seguido.

Una vez se haya generado la nueva contraseña, el sistema debe modificarla instantáneamente en la base de datos, para que no se produzcan errores en la futura autenticación de usuario.

```
//Cambiar en la BBDD la contraseña
miBBDD.ConectaMySQL();
//Nombre y apellido del user
al=miBBDD.DevuelveUsers_data(nameUser,-1,false,null,false);
miBBDD.modificaUsuario(nameUser,...,psswd,null);
miBBDD.DesconectaMySQL();
```

6.8 Administración.

La parte de Administración está restringida al uso del Administrador de la aplicación. Por ello, su acceso está regulado mediante el acceso por roles. Es decir, se crea un determinado rol, al que se dará acceso mediante autenticación al sistema de Administración.

Para ello, hay que seguir dos puntos muy sencillos. En primer lugar, establecer el rol, y asignarlo a aquellos usuarios que se desee tengan esos privilegios. En segundo lugar, habrá que especificar en el contexto del servidor, que parte o partes de la aplicación estarán restringidos, y de la misma forma, especificar a que roles pertenecen estas restricciones.

La primera parte, se debe llevar a cabo en el fichero del servidor “tomcat-users.xml”, y deberá tener la siguiente forma.

```
<tomcat-users>
  <role rolname="Administracion"/>
  <user username="usu" password="usupass" roles="Administracion"/>
</tomcat-users>
```

El rol de “Administración” se le asigna al usuario “usu” con contraseña “usupass”. Y el fichero “context.xml” especifica a qué roles se da permisos de acceso, las partes con acceso restringido y el tipo de autenticación.

```
<security-role>
  <role-name>Administracion</role-name>
</security-role>
<security-constraint>
  <display-name>Administracion</display-name>
<web-resource-collection>
  <web-resource-name>Administracion</web-resource-name>
  <description/>
  <url-pattern>/Administracion/*</url-pattern>
  <http-method>GET</http-method>
  <http-method>POST</http-method>
  <http-method>HEAD</http-method>
  <http-method>PUT</http-method>
  <http-method>OPTIONS</http-method>
  <http-method>TRACE</http-method>
  <http-method>DELETE</http-method>
</web-resource-collection>
<auth-constraint>
  <description>Administracion</description>
  <role-name>Administracion</role-name>
</auth-constraint>
</security-constraint>
<login-config>
  <auth-method>BASIC</auth-method>
  <realm-name>Administracion</realm-name>
</login-config>
```


6.9 Sistema de ficheros propio.

El sistema de ficheros está implementado mediante el módulo de subida y descarga de ficheros propio de apache, *org.apache.commons.fileupload*. Mediante dos formularios web se llevan a cabo la subida y descarga de archivos, que se almacenan en el directorio local del proyecto, creando una carpeta para cada curso.

```
//DECARGA FICHEROS – listaF = lista de ficheros en el directorio del curso.
"<form name='descargar' action='DescargaFicherosServlet?&cn="+codCurso+" method='POST'>";
    int x=0;
    if(listaF!=null){
        while (x<listaF.length){//Imprimir ficheros.
            String fich=listaF[x].getName();
            out.println("<p><input type='radio' name='recurso'
                value='"+fich+"'></input><img src='imag/FileGreen.png' height='20'
                weight='20' >"+fich+"</p>");
            x++;
        }
        out.println("<br><div><p><input name='descargar' type='submit'
            value='"+msg.getString("descargar")+"'></p></div>");//Botón descarga.
    }
    out.println("</form>"+
        "<br><hr></hr><br>"+
        //SUBIDA FICHEROS
        "<form name='subirficheros' id='subirFicheros' method='POST' enctype='multipart/form-data'
            accept='text/plain;image/jpeg' /*Texto e imágenes*/ action='SubirFicherosServlet' ></p>"+
            "<p>"+msg.getString("selectFile")+"</p>"+
            "<p><input type='file' id='fichero' name='fichero'>"+
            "<p><br><input type='submit' name='enviar' value='"+msg.getString("enviar")+"'>"+
        "</form>");
```

Para descargar un fichero, el usuario selecciona qué fichero desea, y mediante un `BufferedInputStream`, se lee del fichero correspondiente que se encuentra en el servidor, y a la vez, se crea un fichero nuevo en el directorio actual del cliente, y mediante un `BufferedOutputStream`, se forma ese fichero con el contenido que se está leyendo.

Por otro lado, para subir un fichero, se ejecuta el siguiente código.

```
//construimos el objeto que es capaz de parsear la petición
DiskFileUpload fu = new DiskFileUpload();

// maximo numero de bytes
fu.setSizeMax(1024*512);

// 512 K
// tamaño por encima del cual los ficheros son escritos directamente en disco
fu.setSizeThreshold(4096);

// Directorio en el que se escribirán los ficheros con tamaño superior al
//soportado en memoria
fu.setRepositoryPath("/tmp");
```

```

// ordenamos procesar los ficheros
List fileItems = fu.parseRequest(request);
if(fileItems == null)
{
    System.out.println("La lista es nula");
    return 0;
}
// Iteramos por cada fichero
Iterator i = fileItems.iterator();
FileItem actual = null;
while (i.hasNext())
{
    actual = (FileItem)i.next();
    if(actual!=null){
        String fileName = actual.getName();
        File f=new File(codCurso);
        if (!f.isDirectory()) f.mkdir();
        // construimos un objeto file para recuperar el trayecto completo
        File fichero = new File(fileName);
        // nos quedamos solo con el nombre y descartamos el path
        fichero = new File(codCurso+"/"+fichero.getName());
        // escribimos el fichero colgando del nuevo path
        actual.write(fichero);
    }
}

```

6.10 Internacionalización del lenguaje de la aplicación.

La aplicación está diseñada para poder trabajar en 2 idiomas, Castellano e Inglés. El idioma se detecta según las configuraciones locales del usuario en el navegador.

```

//Traducción
Locale currentLocale = request.getLocale();
ResourceBundle msg = ResourceBundle.getBundle("MessageBundle",currentLocale);

```

Una vez que se detecta el idioma que tiene configurado el usuario, se carga el archivo de propiedades correspondiente, con título, en este caso, “MessageBundle”, y con extensión el idioma correspondiente en cada caso. Para ello se sigue el estándar especificado para “Locales”, primero se indica el idioma de forma obligatoria, y a continuación el territorio, por ejemplo, en el caso de España sería: “_es_ES”, español de España, y en caso de Argentina, sería “_es_AR”.

Tras detectar el idioma y cargar los archivos de propiedades, cada vez que se desee cargar una palabra almacenada en dichos archivos, se utilizará el objeto “msg”, accediendo al string deseado.

```

msg.getString("mi_cuenta");

```

En caso de que el idioma principal no esté establecido, o que éste sea otro diferente al Castellano o Inglés, se cargará el idioma por defecto, en este caso es Castellano. Éste idioma por defecto se encuentra en el archivo MessageBundle.properties, sin especifica la extensión del idioma.

7. CONCLUSIONES

Aprovechando el avance y extensión de las redes sociales en nuestra sociedad hoy en día, es posible sacar partido a todas sus prestaciones, de forma que además de calar en las personas más jóvenes en el desarrollo dentro de su ámbito social, pueda aprovecharse también para el desarrollo en el educativo.

De esta forma, vemos que puede camuflarse en la interacción con una red social ciertos aspectos formativos, que se conviertan así en objetivos más apetecibles para sus usuarios. Ya que la mayoría de los jóvenes conocen el funcionamiento de estas redes sociales, y las aprovechan al máximo, se puede lograr que el atractivo de dicho formato, sea aplicado a la formación.

Este proyecto, implementa una aplicación Web desde este punto de vista, de forma que no sólo sea una aplicación instalada en una determinada localización, si no que pueda implementarse en cualquier lugar, y sin ningún tipo de restricciones, pudiéndose adaptar en cada caso al enfoque de la formación que se desee.

Sus usuarios podrán interactuar al igual que en una red social, y comunicarse entre ellos, así como aportar detalles a cursos docentes de otros usuarios, o de la misma forma, ofertando sus propias formaciones. Todos ellos poseerán los mismos privilegios dentro de la comunidad, haciendo que está no esté centralizada, y que sean los propios usuarios los que gestionen el sistema, ocupándose cada uno de su formación.

De esta forma, se logra que no haya niveles de interacción y que cada usuario asuma sus propias responsabilidades dentro del sistema. Es decir, podríamos resumir la repercusión de este proyecto en los siguientes puntos.

- ⤴ Desarrollo formativo.
- ⤴ Enfoque docente adaptable.
- ⤴ Igualdad de acción dentro del sistema.
- ⤴ Comunicación entre usuarios.
- ⤴ Interacción entre usuarios.
- ⤴ Responsabilidad.

8. BIBLIOGRAFÍA

1. *THINK IN JAVA*. Bruce Eckel, Prentice Hall, 2000.
2. *JAVA SERVLET PROGRAMMING*. Jason Hunter. O'Reilly, 2001.
3. *INTRODUCCIÓN A LA INGENIERÍA DEL SOFTWARE*. F. Alonso, L. Martínez, F.J. Segovia. Delta Publicaciones, 2005.
4. *THE PRAGMATIC PROGRAMMER*. Andrew Hunt, David Thomas. Addison-Wesley Professional, 1999.
5. *ECLIPSE COOKBOOK*. Steve Holzner. O'Reilly, 2004.
6. *JAVA THREADS*. Scott Oaks, Henry Wong. O'Reilly, 2001.
7. *BASES DE DATOS CON JAVA*. K. Mukhar, T. Lauinger, J. Carnell. Anaya, 2002.

REFERENCIAS

- ⤴ Web oficial de Tomcat [2011] <http://tomcat.apache.org/>
- ⤴ Tutorial Servlets [2011]
 http://www.programacion.net/tutorial/servlets_basico/
- ⤴ Java [2011]
 - Tutoriales java <http://www.programacion.net/java/>
 - Api de java <http://download.oracle.com/javase/1.5.0/docs/api/>
 - Java mail api
 <http://www.oracle.com/technetwork/java/javamail/index.html>

9. ANEXO: MANUAL DEL USUARIO

Como ayuda al futuro usuario de la aplicación Web, se ha decidido incluir esta sección, de forma que permita mejorar y hacer más sencillo el uso y manejo de la misma, de cara a sus usuarios, cuyas habilidades tecnológicas se encontraran en rangos bastante dispares.

Se describirá a continuación cada una de las secciones y posibilidades que ofrece la herramienta, acompañadas cada una por una imagen que ilustre y facilite la comprensión de cada una de ellas. En primer lugar, se explicará la consulta de oferta docente, acción no ligada al hecho de estar registrado como usuario del sistema. Seguidamente, se describirán las acciones que todo usuario registrado puede llevar a cabo, finalizando con las posibilidades que se ofrecen en el ámbito de la administración, restringidas al Administrador del sistema.

A todos estos espacios puede accederse a través de la página de inicio de la aplicación Web.

9.1 CONSULTA OFERTA DOCENTE

upna Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

INICIO Mis Cursos Mis Inscripciones Buscar

Acceso a GestDoc

Acceda a GestDoc

Usuario:

Contraseña:

Acceder

[¿Olvidaste tu contraseña?](#)

Registrarse

Regístrate para acceder.

Nombre:

Apellido:

Email:

Password:

a0336

Introduzca los caracteres de la imagen

Registrarse

*La contraseña debe tener una longitud máxima de 6 caracteres

Bienvenido a GestDoc, su gestión docente

¿Qué es la gestión docente?

La gestión docente es una herramienta de uso privado, que permite a sus usuarios diferentes posibilidades a la hora de gestionar su propia docencia.

¿Por qué usar GESTDOC?

Registrándose y accediendo a la comunidad, podrá crear sus propios cursos de manera que sólo usted elija la forma en que se impartirán, lugar, horario, etc.

Además, el resto de usuarios podrán ver la oferta docente y tendrán así mismo, la posibilidad de inscribirse a ésta, y poder acceder a los diferentes cursos.

Esta herramienta permite además la comunicación entre usuarios, creando una mayor interacción y haciendo que la gestión docente sea más sencilla para todos.

Oferta docente

Listado de cursos [Ver...](#)

Admin [...]

Copyright © 2010 GestDoc. All Rights Reserved. Designed by Usue Napal
Contacto: usuenapal@gmail.com

Figura 68 – Pantalla de Inicio.

La consulta de oferta es independiente del usuario que la consulte, tanto miembros de la comunidad como usuario no registrados podrán acceder a ella desde la página inicial del sistema (parte inferior derecha).

En ella podrán ver que cursos se ofertan actualmente, y si lo desean, acceder para inscribirse en los mismos. Además, podrán registrarse como usuarios del sistema si aún no lo son, y de la misma forma, acceder a dicha inscripción.



upna
Universidad
Pública de Navarra
Nafarroako
Unibertsitate Publikoa

Oferta docente

Cocina Moderna
10:00-13:00
2011-08-17
Parroquia de Mutilva
10 plazas
Profesor: usuenapal@gmail.com
[Acceder](#)

Curso de batería
17:00-19:00
2011-07-04
Edificio El Sario (UPNA)
2 plazas
Profesor: lus86@hotmail.com
[Acceder](#)

Latín
9:00-11:00
2011-08-15
Universidad Pública de Navarra
5 plazas
Profesor: lus86@hotmail.com
[Acceder](#)

[Inicio](#)

Figura 69 – Pantalla de oferta docente del sistema.

9.2 REGISTRO

Para formar parte del sistema y poder disfrutar de todas las ventajas de la aplicación será necesario registrarse para ser un miembro más del sistema, y tener acceso al mismo mediante autenticación. Registrarse en el sistema es una acción muy sencilla y que tan sólo requiere un par de minutos para completarse.

En la página de inicio existe un formulario, en la parte izquierda inferior, que bastará con rellenar y pulsando sobre “Registrarse” se completará el proceso, de forma rápida y sencilla.

upna Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

INICIO Mis Cursos Mis Inscripciones Buscar

Acceso a GestDoc

Acceda a GestDoc

Usuario:

Contraseña:

Acceder

[¿Olvidaste tu contraseña?](#)

Registrarse

Regístrate para acceder.

Nombre:

Apellido:

Email:

Password:

5063b

Introduzca los caracteres de la imagen

Registrarse

*La contraseña debe tener una longitud máxima de 6 caracteres

Bienvenido a GestDoc, su gestión docente

¿Qué es la gestión docente?

La gestión docente es una herramienta de uso privado, que permite a sus usuarios diferentes posibilidades a la hora de gestionar su propia docencia.

¿Por qué usar GESTDOC?

Registrándose y accediendo a la comunidad, podrá crear sus propios cursos de manera que sólo usted elija la forma en que se impartirán, lugar, horario, etc.

Además, el resto de usuarios podrán ver la oferta docente y tendrán así mismo, la posibilidad de inscribirse a ésta, y poder acceder a los diferentes cursos.

Esta herramienta permite además la comunicación entre usuarios, creando una mayor interacción y haciendo que la gestión docente sea más sencilla para todos.

Oferta docente

Listado de cursos [Ver...](#)

Admin [...]

Copyright © 2010 GestDoc. All Rights Reserved. Designed by Usue Napal
Contacto: usuenapal@gmail.com

Figura 70 – Registro de usuarios.

9.3 ACCESO

El acceso se llevará a cabo también desde la página de inicio. Al igual que en el registro, en el menú izquierdo de la página inicial se encuentra un pequeño formulario. Bastará con completarlo, indicando el email con el que el usuario se ha registrado, y su contraseña, y pulsando sobre el botón “Acceder” se accederá a la aplicación.

The screenshot displays the GestDoc web application interface. At the top left is the 'upna' logo (Universidad Pública de Navarra / Nafarroako Unibertsitate Publikoa). A navigation menu contains buttons for 'INICIO', 'Mis Cursos', 'Mis Inscripciones', and 'Buscar'. The main content area is divided into two columns. The left column features a 'Acceso a GestDoc' section with a login form (User: usuenapal@gmail.com, Password: masked) and an 'Acceder' button. Below it is a 'Registrarse' section with fields for Name, Surname, Email, and Password, and a 'Registrarse' button. A CAPTCHA image with the characters '4d013' is shown. The right column contains a 'Bienvenido a GestDoc, su gestión docente' section with introductory text, a '¿Por qué usar GESTDOC?' section with details about the tool, and an 'Oferta docente' section with a 'Listado de cursos Ver...' link. At the bottom, there is an 'Admin [...+]' link and a footer with copyright information: 'Copyright © 2010 GestDoc. All Rights Reserved. Designed by Usue Napal. Contacto: usuenapal@gmail.com'.

Figura 71 – Acceso de usuarios.

9.3.1 MIS CURSOS

Una vez que se ha accedido al sistema, en primer lugar se nos presenta la pestaña de “Mis Cursos”. En ella se mostrarán todos aquellos cursos que el usuario haya creado para impartir docencia. Dentro de cada uno, aparecerán algunos de los detalles básicos del curso indicados, como descripción, horario, etc.

Además, cada curso, permitirá acceder a otras opciones, como eliminar el curso, editar su información, ver los usuarios inscritos, o acceder a su espacio personal.

Por otro lado, en la misma pestaña se puede acceder al formulario para crear un nuevo curso, mediante el botón “Crear”, situado en la parte inferior de la pantalla.

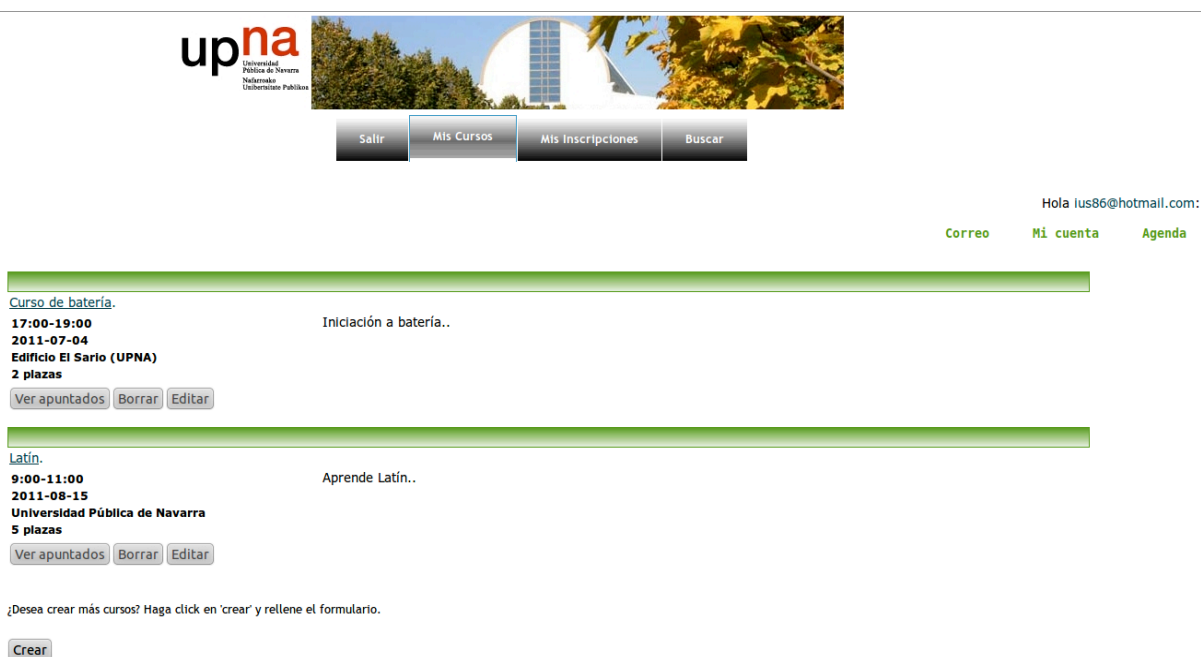


Figura 72 – Pantalla ‘Mis Cursos’

Por otro lado, y desde cada pantalla del sistema, se tendrá acceso al correo interno, la configuración de la cuenta de usuario y la agenda, que indica los eventos próximos en el calendario.

Estas tres secciones se explicarán con más detalle próximamente.

9.3.2 MIS INSCRIPCIONES

En la segunda pestaña del sistema, se encuentra la sección “Mis inscripciones”, donde al igual que en la primera se muestran una serie de cursos y sus datos y acciones básicas. En este caso se trata de aquellos cursos en los que el usuario ha realizado una inscripción.



Figura 73 – Pantalla ‘Mis inscripciones’

Al igual que en al pestaña de “Mis Cursos”, se observa los cursos en pantalla, con su información básica, pero además en este caso se añade la información referente a plazas vacantes del curso. Es decir, si el usuario inscrito ha obtenido una plaza en el curso, o por el contrario, se encuentra en lista de espera.

En caso de tratarse de este último caso, el aviso aparecerá en color rojo, indicando también la posición de la lista de espera en la que se encuentra. Si en un futuro ese estado cambiase, el usuario recibiría un email de confirmación, y en mensaje mostrado en esta pantalla variaría para indicar que el usuario dispone de una plaza vacante en el curso.

Por otro lado, como puede observarse, existe un botón en la parte inferior de la impresión del curso, que permitirá al usuario desapuntarse del mismo. Bastará con seleccionar el curso, y pulsar el botón. La página solicitará entonces la confirmación por parte del usuario para realizar la operación, y en caso de ser confirmada, se llevará a cabo.

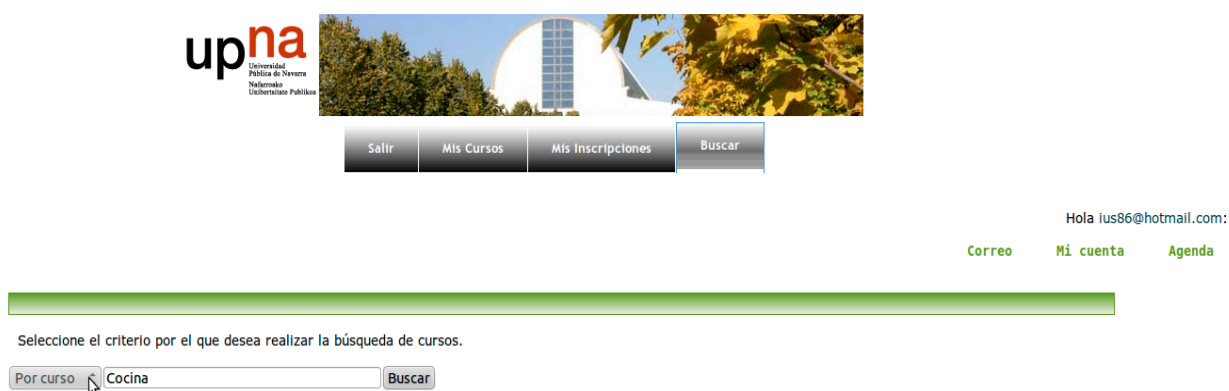
Finalmente, y de la misma forma que en la sección de “Mis Cursos”, podrá accederse al espacio personal del curso.

9.3.3 BUSCAR

Esta sección permite al usuario buscar cursos. Puede ordenar los resultados por nombre del curso, o bien, agruparlos por profesor. De esta forma, si está interesado especialmente en la formación de un determinado docente, podrá obtener un resultado con todos sus cursos de forma ordenada.

Una vez obtenida la lista de cursos disponibles, el usuario se podrá inscribir en aquel curso que sea de su interés, en un paso muy sencillo: seleccionar dicho curso y pulsar sobre el botón “Inscribirse”.

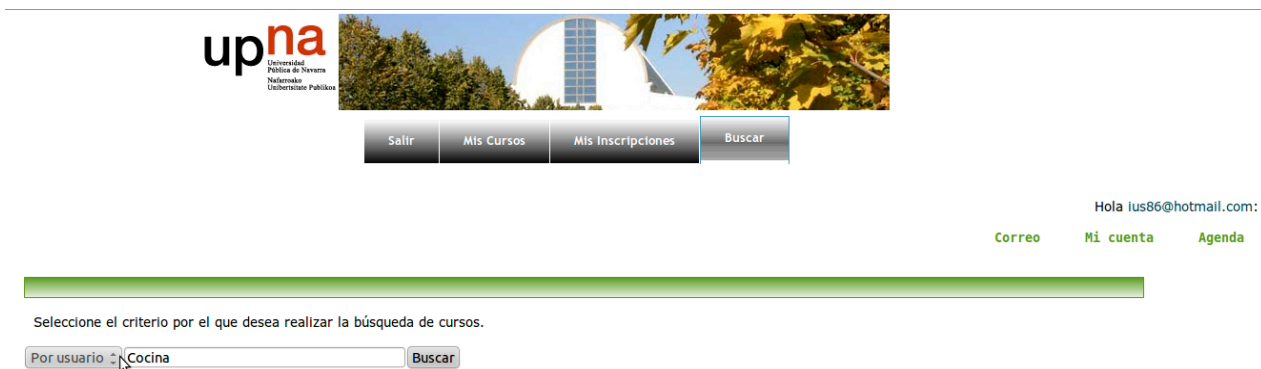
Listado ordenado por curso:



The screenshot shows the top navigation bar of the UPNA website. On the left is the UPNA logo (Universidad Pública de Navarra). To its right is a banner image of a building. Below the banner are four buttons: "Salir", "Mis Cursos", "Mis Inscripciones", and "Buscar". On the right side of the page, there is a user greeting "Hola ius86@hotmail.com:" and three links: "Correo", "Mi cuenta", and "Agenda". Below this is a green horizontal bar. Underneath the bar, the text "Seleccione el criterio por el que desea realizar la búsqueda de cursos." is displayed. Below this text is a search form with a dropdown menu set to "Por curso", a text input field containing "Cocina", and a "Buscar" button.

Figura 74 – Búsqueda de cursos por nombre.

Listado ordenado por profesor:



This screenshot is identical to the one above, showing the same navigation bar and user information. However, the search form below the green bar has a dropdown menu set to "Por usuario" instead of "Por curso", and the text input field still contains "Cocina".

Figura 75 – Búsqueda de cursos por docente.

Resultado de búsqueda por profesor:



The screenshot shows the UPNA web application interface. At the top left is the UPNA logo (Universidad Pública de Navarra) and a navigation bar with buttons for 'Salir', 'Mis Cursos', 'Mis Inscripciones', and 'Buscar'. On the right, there are links for 'Correo', 'Mi cuenta', and 'Agenda', along with the user's email 'Hola ius86@hotmail.com:'. Below the navigation bar, a search result is displayed for a course: 'Curso impartido por usuenapal@gmail.com'. The course details are: 'Cocina Moderna', '10:00-13:00', '2011-08-17', 'Parroquia de Mutilva', '10 plazas', and 'Profesor: usuenapal@gmail.com'. An 'Inscribirse' button is located below the course details.

Figura 76 – Resultado búsqueda por docente.

Inscripción correcta a un curso:



The screenshot shows the UPNA web application interface displaying a successful registration confirmation. At the top, there is a green banner with the text 'Inscripción realizada con éxito'. Below this, a message states: 'La inscripción en el curso ha tenido un resultado satisfactorio. Los datos de matrícula del curso son los siguientes:'. A table lists the course details:

Nombre del curso:	Cocina Moderna
Descripción:	Introducción a la cocina moderna
Turno:	Mañana
Horario:	10:00-13:00
Profesor:	usuenapal@gmail.com
Fecha de inicio del curso:	2011-08-17 00:00:00.0
Fecha de finalización del curso:	2011-09-16 00:00:00.0
Año académico:	2011
plazas:	10
Número de días del curso:	30
Observaciones:	
Fecha de inicio de matrícula:	2011-07-02 00:00:00.0
Fecha de fin de matrícula:	2011-07-02 00:00:00.0
Lugar:	Parroquia de Mutilva
Usuario:	josu.napal@gmail.com

Below the table, a message states: 'Le informamos de que actualmente hay plazas disponibles y le hemos reservado una.'

Figura 77 – Pantalla de inscripción correcta.

Inscripción correcta (en lista de espera):

upna Universidad Pública de Navarra Nafarroako Unibertsitatea Publikoa

Salir Mis Cursos Mis Inscripciones Buscar

Inscripción realizada con éxito
La inscripción en el curso ha tenido un resultado satisfactorio.
Los datos de matrícula del curso son los siguientes:

Nombre del curso:	Curso de batería
Descripción:	Iniciación a batería.
Turno:	Tarde
Horario:	17:00-19:00
Profesor:	ius86@hotmail.com
Fecha de inicio del curso:	2011-07-04 00:00:00.0
Fecha de finalización del curso:	2011-08-01 00:00:00.0
Año académico:	2011
plazas:	2
Número de días del curso:	20
Observaciones:	
Fecha de inicio de matrícula:	2011-07-02 21:05:26.0
Fecha de fin de matrícula:	2011-07-02 21:05:26.0
Lugar:	Edificio El Sario (UPNA)
Usuario:	josu.napal@gmail.com

Actualmente las plazas disponibles están ya asignadas. Usted se encuentra el número 1 en lista de espera .

En caso de que en el futuro se liberen plazas será debidamente notificado.

Figura 78 – Pantalla de inscripción en lista de espera.

9.3.4 CORREO INTERNO

Mediante el correo interno todo usuario podrá ponerse en contacto con cualquier otro usuario del sistema, de forma que se establezca una comunicación interna. Mediante cualquier pantalla, y pulsando sobre “Correo”, en la parte superior derecha de la pantalla, se accederá a dicha sección. En ella se podrá enviar correos y recibirlos, así como llevar la gestión de los mismos.

Figura 79 – Pantalla de correo interno.

En primer lugar, en la parte superior de la pantalla, se encuentra la bandeja de entrada, donde aparecerán los mensajes recibidos por el usuario. Inicialmente se mostrará de quien procede el mensaje, y cuándo ha sido enviado. Para acceder al contenido del mismo, se pulsará sobre la cruz blanca situada en la parte derecha de la caja que contiene el mensaje. Una vez desplegada, se mostrará también la opción de responder, iniciándose automáticamente la escritura de un correo nuevo al usuario que acaba de escribirnos.

A continuación, se encuentra la bandeja de salida, que funciona de la misma forma. Además, podrá seleccionarse el mensaje que se desee y pulsar sobre borrar, o por el contrario, pulsa el botón “Borrar todos”, que vaciará la bandeja de salida.

Finalmente se encuentra la sección que permite componer nuevos correos.

En caso de tener correos nuevos, el sistema avisará, imprimiendo entre paréntesis, el número de correos nuevos de los que dispone el usuario.



Figura 80 – Aviso de nuevos correos.

Para que dicho aviso desaparezca, habrá que marcar como leído el mensaje, desde la bandeja de entrada.

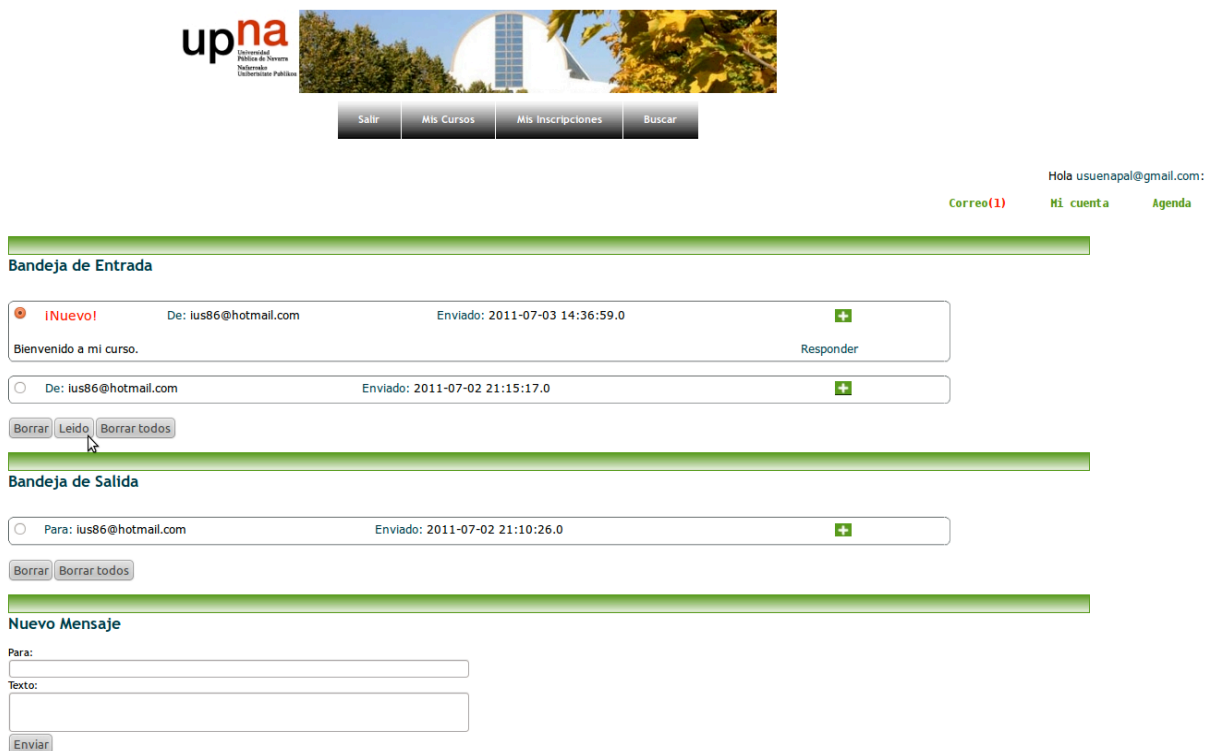


Figura 81 – Eliminar aviso de nuevos correos.

9.3.5 MI CUENTA

La configuración de cuenta permite al usuario configurar sus datos personales, tanto nombre y apellido, como la contraseña mediante la cual accederá al sistema. Para hacer cualquier cambio, deberá introducir su contraseña, para verificar que realmente se trata del usuario en sí.

Además podrá optar a eliminar su cuenta por completo. En este caso, se avisará por email a aquellos usuarios inscritos en sus cursos, indicando que el curso al que se habían inscrito ya no se encuentra disponible.

upna
Universidad
Pública de Navarra
Nafarroako
Unibertsitate Publikoa

Salir Mis Cursos Mis Inscripciones Buscar

Hola ius86@hotmail.com:
Correo Mi cuenta Agenda

Configuraciones ius86@hotmail.com

Nombre : Ius
Apellido : Honorarium
Password:
Nueva Password:

Modificar

*La contraseña debe tener una longitud máxima de 6 caracteres

Eliminar cuenta

Figura 82 – Pantalla de datos de usuario.

El email del usuario no podrá ser modificado ya que éste es el identificador inequívoco del usuario dentro del sistema.

9.3.6 AGENDA

En la agenda se muestra al usuario los eventos próximos que tiene en su calendario. Se especificará la fecha de inicio de cada uno de ellos. Podrá navegarse por las fechas mediante las flechas de navegabilidad que se encuentran a ambos lados del mes mostrado.

The screenshot shows the user interface of the 'upna' web application. At the top left is the 'upna' logo (Universidad Pública de Navarra / Nafarroako Unibertsitate Publikoa). Below it are navigation buttons: 'Salir', 'Mis Cursos', 'Mis Inscripciones', and 'Buscar'. On the right, there is a user greeting 'Hola ius86@hotmail.com:' and links for 'Correo', 'Mi cuenta', and 'Agenda'. A green horizontal bar separates the header from the main content area, which is titled 'PRÓXIMOS EVENTOS'. Below this is a calendar for August 2011. The calendar grid shows days from 1 to 31. The event 'Cocina Moderna' is listed on Wednesday, August 17th.

Agosto 2011						
LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SÁBADO	DOMINGO
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17 Cocina Moderna	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Figura 83 – Agenda del sistema.

9.3.7 VER APUNTADOS

En la pantalla de ver apuntados, el usuario que impartirá un determinado curso, podrá ver qué usuarios se han inscrito. Además de mostrar una lista con los datos de estos usuarios y la posición en lista de espera que ocupan, tendrá la opción de ponerse en contacto con ellos mediante el correo interno.

Si desea enviar un mensaje a todos los usuario apuntados a su curso, podrá escribir el mensaje en la parte inferior de la pantalla, en el cuadro de texto indicado. Bastará con pulsar el botón de enviar y cada uno de los usuarios inscritos recibirá en su bandeja de entrada dicho mensaje.

The screenshot shows the 'upna' logo (Universidad Pública de Navarra) and a navigation menu with buttons for 'Salir', 'Mis Cursos', 'Mis Inscripciones', and 'Buscar'. The user is logged in as 'ius86@hotmail.com'. The main content area is titled 'Inscripciones en 'Curso de batería'' and contains a table of users:

Usuario	Nombre	Apellido	Número lista espera
<input type="radio"/> josu.napel@gmail.com	Josu	Napel	1
<input type="radio"/> l.mar.alon@gmail.com	Laura	Martínez	0
<input type="radio"/> usuenapel@gmail.com	Usue	Napel	0

Below the table, there is a button 'Enviar mensaje' and a text box for sending a message to all users. The text box contains the message: 'Bienvenidos al curso de batería. Para cualquier duda/sugerencia podéis escribirme.' Another 'Enviar mensaje' button is located below the text box.

Figura 84 – Pantalla de muestra de usuarios inscritos.

Por otra parte, si tan sólo desea ponerse en contacto con uno de los usuarios, o varios, en concreto, podrá pinchar sobre el usuario deseado y a continuación pulsar el botón de enviar, que le trasladará directamente a la sección de correo interno, colocando automáticamente el nombre del destinatario.

9.3.8 EDITAR CURSO

Desde la pestaña de “Mis cursos”, se puede acceder a editar un curso de forma muy sencilla. Si se pulsa sobre el botón “Editar”, se accederá a la siguiente pantalla.

upna
Universidad
Pública de Navarra
Nafarroako
Unibertsitate Publikoa

Salir Mis Cursos Mis Inscripciones Buscar

Hola ius86@hotmail.com:
Correo Mi cuenta Agenda

Código curso: 02CodigoCurso
 Nombre del curso: Curso de batería
 Descripción: Iniciación a batería.
 Turno: Tarde
 Horario: 17:00-19:00
 Profesor: ius86@hotmail.com
 Año académico: 2011
 Número de días del curso: 20
 plazas: 2
 Lugar: Edificio El Sario (UPNA)

Descripción (opcional) de los días:

Disponible: Si No

Fechas del curso (Formato: YYYY-mm-dd)

Fecha de inicio del curso: 2011-07-04
 Fecha de finalización del curso: 2011-08-01

Editar

Figura 85 – Pantalla de edición de cursos.

En ella se muestran los datos del curso. En caso de querer modificar alguno, bastará con editar aquellos que se desee, y a continuación se pulsará sobre “Editar”.

9.3.9 ESPACIO CURSO

Para acceder al espacio privado de un curso, bastará pinchar sobre el nombre del mismo. Sólo se podrá acceder a este espacio si el usuario que lo intenta se encuentra inscrito en dicho curso, o por otro lado, si el usuario es el que va a impartir la formación.

En él se puede encontrar la descripción del curso, así como una sección en la que todos los participantes, tanto docentes como alumnos, podrán participar dejando sus comentarios y opiniones a cerca de la formación, de una forma anónima. Sin embargo, para ello deberán estar también inscritos en el curso.

Hola ius86@hotmail.com:

[Correo](#) [Mi cuenta](#) [Agenda](#)

Cocina Moderna

Introducción a la cocina moderna

Curso impartido por: usuenapal@gmail.com

Comentarios

Espero que el curso sea de vuestro agrado. Si tenéis alguna duda o sugerencia podéis poneros en contacto conmigo, mediante el correo interno o escribiéndome a: usuenapal@gmail.com Un saludo, Usue.
2011-07-03 00:08: por Usue N. M.

¡¡Muchas gracias por la receta!! Para ir empezando está muy bien.
¡A mí me ha salido estupendo! :D
Saludos.
2011-07-03 00:13: por Antonio

Realizar un comentario

Nombre :

Comentario:

Recursos

brownie.doc

Figura 86 – Espacio personal de cursos.

Finalmente, y en la parte inferior de la pantalla se encuentra el apartado de documentación, donde los usuarios podrán subir ficheros que vayan a ser de utilidad para el resto de participantes, o por el contrario, descargar aquella documentación que otros participantes hayan subido al sistema de ficheros. Todo de forma absolutamente sencilla.

Para poder subir un fichero, se seleccionará este del sistema de ficheros local del usuario, y se pinchará en el botón “Subir”. Para descargar un documento, se seleccionará aquel que se desee descargar y se pulsará sobre el botón “Descargar”.

Recursos del curso:

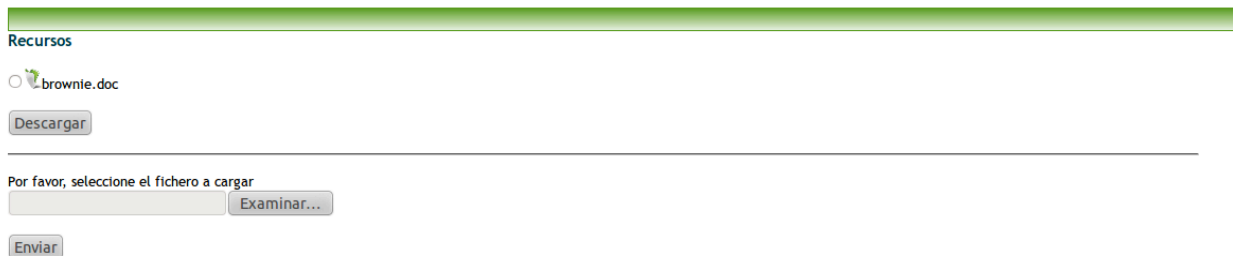


Figura 87 – Descarga y subida de ficheros.

Por otro lado, si el usuario que se encuentra en el espacio del curso, es aquel que impartirá la docencia, tendrá la opción de moderar los comentarios realizados. Para ello, en la parte superior de cada uno de ellos, se mostrará una pequeña cruz, que pulsando sobre ella permitirá al usuario propietario del curso eliminarlo.

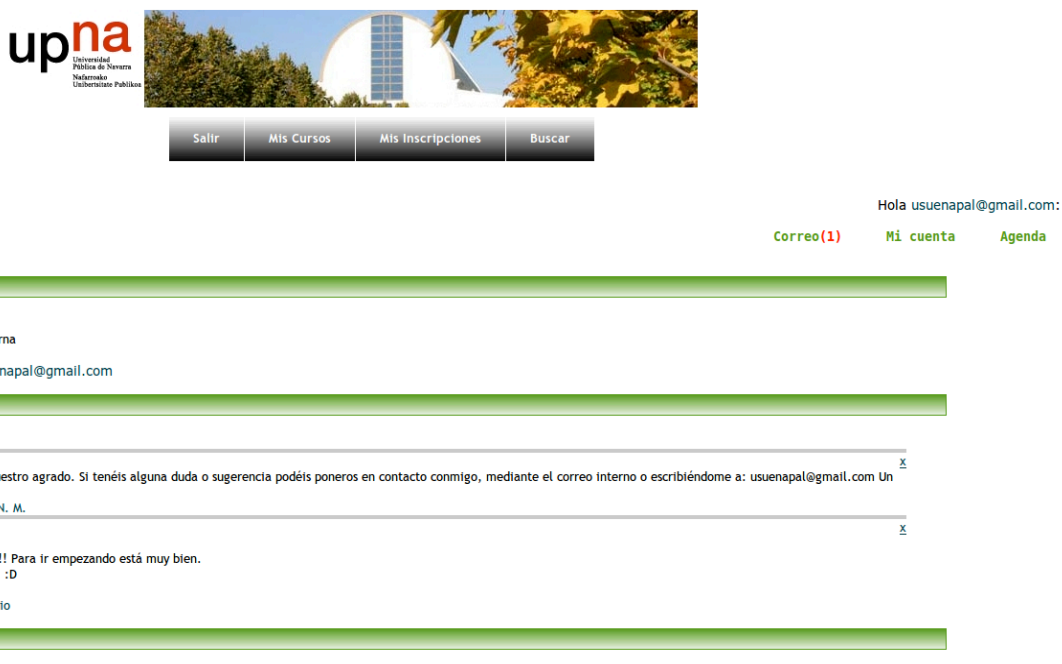


Figura 88 – Comentarios del curso.

9.3.10 CREAR CURSO

Para crear un curso, desde la pestaña de “Mis cursos” se pulsará sobre el botón “Crear”, y se accederá a una pantalla que contiene el formulario específico para crear cursos. Bastará con rellenarlo con los datos adecuados y pulsar sobre “Crear”.

En caso de querer borrar todos los datos y comenzar a rellenar el formulario desde cero, podrá pincharse sobre el botón “Reset”, que reseteará los datos del formulario.


Figura 89 – Pantalla de creación de cursos.

Para que el curso esté disponible y puedan verlo el resto de usuario se deberá seleccionar “Si” en la opción “Disponible”. Si se selecciona “No” el curso no será visible para el resto de usuarios, pero el propietario del mismo podrá modificar esta característica más adelante, para publicarlo cuando crea necesario.

Los campos en negrita son obligatorios, y los campos de “Código curso” y “Profesor” se rellenan automáticamente, de forma que no se pueden editar.

9.3.11 RECUPERAR CONTRASEÑA

Desde la página de inicio se puede solicitar una nueva contraseña, en caso de que el usuario no recuerde la suya. Para ello, debajo del formulario de acceso, se pinchará sobre la pregunta “¿Olvidaste tu contraseña?”. Esto llevará al usuario a otra pantalla, donde con sólo insertar su correo electrónico, se le enviará al mismo su nueva contraseña para acceder al sistema.



Por favor, introduzca su correo electrónico para solicitar una nueva contraseña.

Figura 90 – Pantalla de recuperación de contraseña.

9.4 ADMINISTRACIÓN

El Administrador será el único usuario que tenga acceso a esta parte de la aplicación. De forma que se necesitará autenticación por parte del que intente acceder a ella, solicitada por parte del sistema.

upna Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

INICIO Mis Cursos Mis Inscripciones Buscar

Acceso a GestDoc

Acceda a GestDoc

Usuario:

Contraseña:

Identificación requerida

http://localhost:8080 está solicitando un nombre de usuario y una contraseña. El sitio dice: "Administracion"

Nombre de usuario:

Contraseña:

Cancelar Aceptar

Bienvenido a GestDoc, su gestión docente

¿Qué es la gestión docente?

La gestión docente es una herramienta de uso privado, que permite a sus usuarios diferentes posibilidades a la hora de gestionar su propia docencia.

Nombre:

Apellido:

Email:

Password:

33905

Introduzca los caracteres de la imagen

Registrarse

*La contraseña debe tener una longitud máxima de 6 caracteres

Oferta docente

Listado de cursos [Ver...](#)

Admin [...+]

Copyright © 2010 GestDoc. All Rights Reserved. Designed by Usue Napal
Contacto: usuenapal@gmail.com

Figura 91 – Acceso a administración.

Una vez dentro se podrá acceder a datos estadísticos así como al mantenimiento de los datos del sistema. Por otro lado, pulsando sobre el enlace de inicio, se regresa a la página de inicio del sistema.

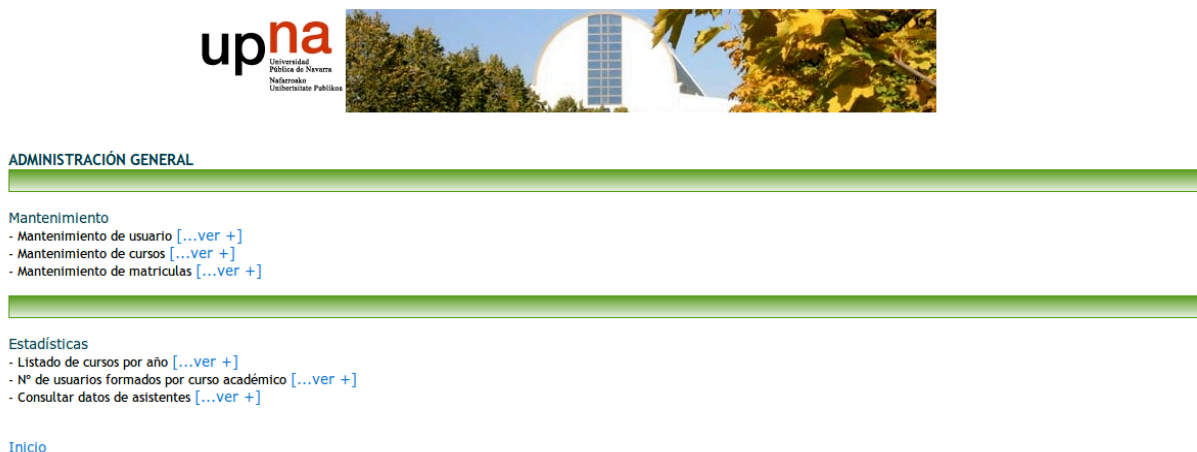


Figura 92 – Menú de administración.

9.4.1 MANTENIMIENTO

En la parte de mantenimiento existen tres secciones diferentes, el mantenimiento de usuario, de cursos y de matriculas.

Mantenimiento de usuarios:

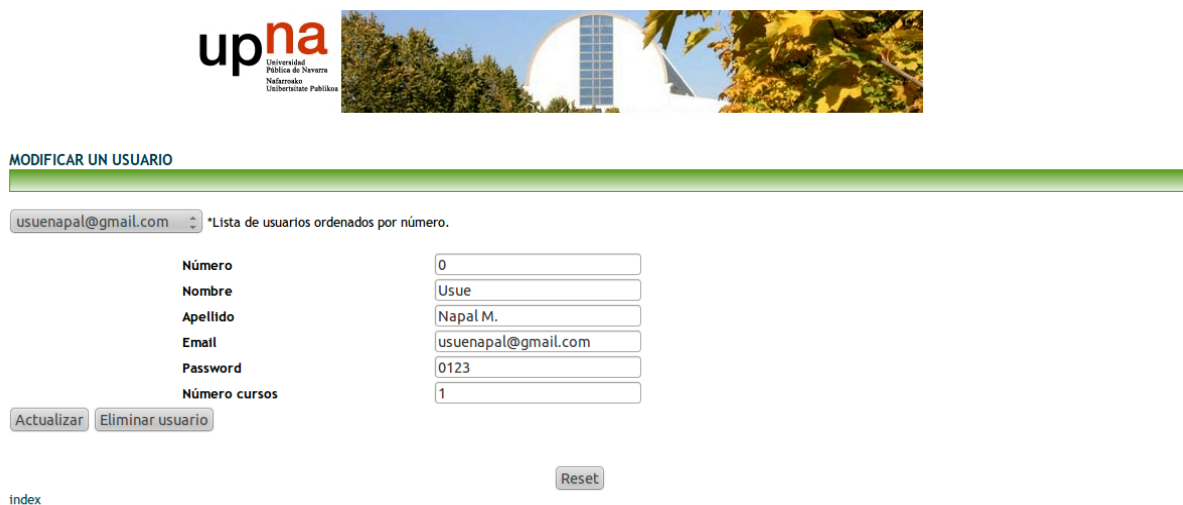


Figura 93 – Mantenimiento de usuarios.

Mediante el mantenimiento de usuarios se puede modificar los datos de cualquiera de los usuarios registrados en el sistema, o por otro lado, eliminar su cuenta.

Mediante el select situado en la parte superior derecha de la pantalla se selecciona el usuario que se desea modificar. Éstos están ordenados por el número de usuario e identificados por su correo electrónico. Bastará con seleccionar el usuario a modificar y realizar los cambios deseados.

Finalmente, una vez realizadas las modificaciones, se pulsará sobre el botón de “Actualizar”, y se mostrarán los datos actualizados con el correspondiente mensaje del estado de la actualización.

Mantenimiento de cursos:

MODIFICAR UN CURSO

Seleccione un año académico y un curso por favor.

2011

02CodigoCurso

Código	02CodigoCurso
Nombre	Curso de batería
Descripción	Iniciación a batería.
Turno	Tarde
Horario	17:00-19:00
Profesor	ius86@hotmail.com
Inicio	2011-07-04
Fin	2011-08-01
Año académico	2011
Número Días	20
Plazas	2
Descripción días	
Inicio matrícula	2011-07-02
Fin matrícula	2011-07-02
Lugar	Edificio El Sario (UPNA)
Disponible	true

Actualizar Eliminar curso

*Formato fecha: aaaa-mm-dd
*Disponible: Si/No

Reset Crear curso

[index](#)

Figura 94 – Mantenimiento de cursos.

Mediante el mantenimiento de cursos se puede modificar los datos de cualquiera de los cursos disponibles en el sistema, o por otro lado, eliminarlos o crear uno nuevo.

Mediante el select situado en la parte superior derecha de la pantalla se selecciona el año sobre el que se desea trabajar, y acto seguido el curso que se desea modificar. Éstos están ordenados por el código de curso. Bastará con seleccionar un curso, realizar los cambios deseados y pulsar sobre el botón de “Actualizar”.

Mantenimiento de matrículas:

upna
Universidad
Pública de Navarra
Nafarroako
Unibertsitate Publikoa

MODIFICAR UNA MATRÍCULA

Matriculas por curso

Seleccione un año académico y un curso por favor.

2011

02CodigoCurso

josu.napal@gmail.com Borrar

Lmar.alon@gmail.com Borrar

usuenapal@gmail.com Borrar

Reset

[index](#)

Figura 95 – Mantenimiento de matrículas.

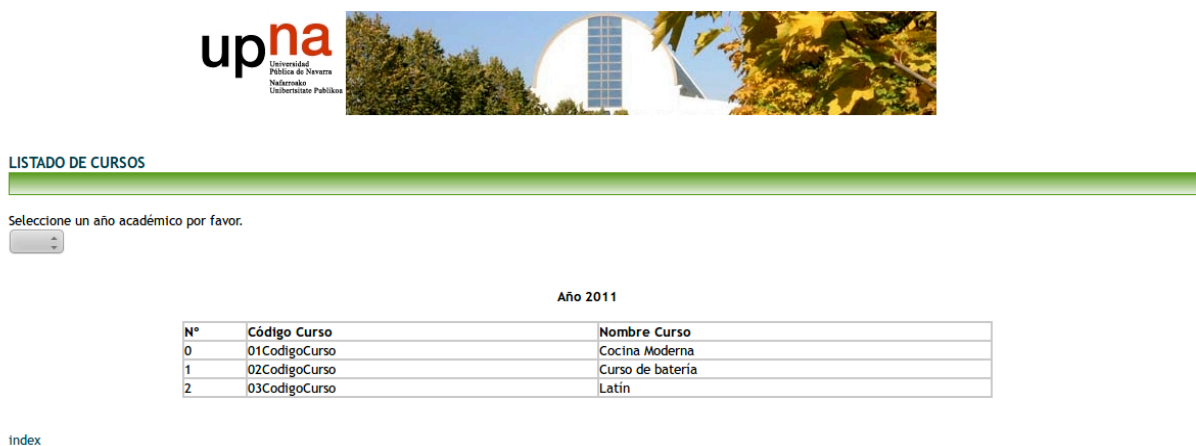
Para las matrículas, habrá que seleccionar el año al que pertenece el curso, y el código del mismo. Con ello se mostrarán por pantalla las inscripciones a dicho curso, y podrán borrarse si se desea.

Los botones de reset en estas pantallas indican un reset del curso o usuario seleccionado para modificar. De la misma forma, en todas las pantallas, el link “Inicio” envía al usuario a la pantalla inicial del Administrador.

9.4.2 ESTADÍSTICAS

En la parte de estadísticas existen tres secciones diferentes, por un lado el listado de cursos por año, por otro el nº de usuarios formados por curso académico y finalmente la consulta de datos de asistentes a cursos.

Listado de cursos por año:



LISTADO DE CURSOS

Seleccione un año académico por favor.

Año 2011

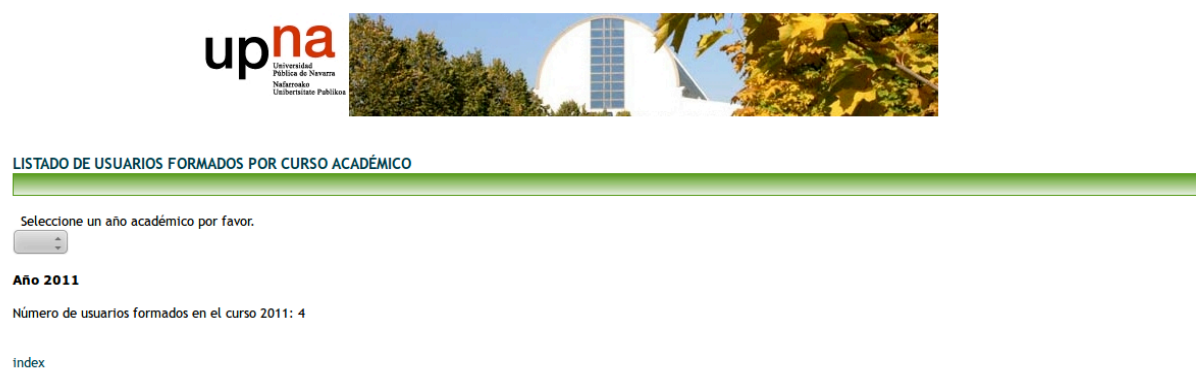
Nº	Código Curso	Nombre Curso
0	01CodigoCurso	Cocina Moderna
1	02CodigoCurso	Curso de batería
2	03CodigoCurso	Latin

[Index](#)

Figura 96 – Listados de cursos por año académico.

El usuario selecciona un año del select situado en la parte superior izquierda de la pantalla, y seguidamente se muestran los resultados por pantalla.

Nº de usuarios formados por curso académico:



LISTADO DE USUARIOS FORMADOS POR CURSO ACADÉMICO

Seleccione un año académico por favor.

Año 2011

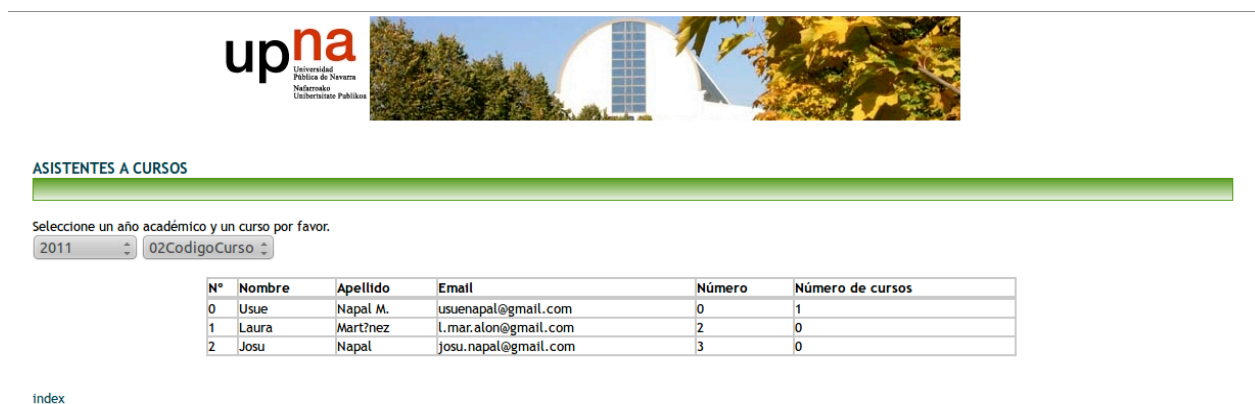
Número de usuarios formados en el curso 2011: 4

[Index](#)

Figura 97 – Listado de usuarios formados.

Basta con que el usuario seleccione el año académico del que desea obtener las estadísticas y éstas se mostrarán por pantalla.

Consulta de datos de asistentes a cursos:



ASISTENTES A CURSOS

Seleccione un año académico y un curso por favor.

2011 02CodigoCurso

Nº	Nombre	Apellido	Email	Número	Número de cursos
0	Usue	Napal M.	usuenapal@gmail.com	0	1
1	Laura	Martínez	l.mar.alon@gmail.com	2	0
2	Josu	Napal	josu.napal@gmail.com	3	0

[index](#)

Figura 98 – Listado de asistentes a un curso.

El usuario seleccionará el año en el cual desea trabajar, acto seguido seleccionará el curso del que desea obtener las estadísticas y éstas se mostrarán en pantalla.

NOTA:

Todas las pantallas del sistema que realizan algún tipo de variación en el estado de los datos del sistema, ya sea añadiendo, eliminando o modificando los mismos, remiten por pantalla un mensaje del estado de la operación, tanto si es correcto, como si no lo es. En caso de producirse algún error se especificará cual ha sido para que el usuario pueda proceder a arreglarlo.

