



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO TÉCNICO EN INFORMÁTICA DE GESTIÓN

Título del proyecto:

“APLICACIÓN SILVERLIGHT PARA LA GESTIÓN DE UN
ALMACÉN DE SUMINISTROS INDUSTRIALES”

Alumno: Héctor González Liñán

Tutor: Jesús Villadangos Alonso

Pamplona, 9 de Septiembre del 2011

Agradecimientos:

A Jesús Villadangos, el cual me ha ido resolviendo cualquier duda a lo largo del desarrollo del proyecto.

A mi familia, amigos y especialmente a Sara.

A mis compañeros de clase, especialmente a David y Jorge.

INDICE

1. Resumen.....	4
2. Introducción.....	5
2.1. Introducción a la empresa.....	5
2.2. Antecedentes y objetivos.....	6
2.3. Etapas de desarrollo.....	7
3. Herramientas y tecnologías.....	8
4. Análisis y Diseño.....	15
4.1. Especificación de requisitos y casos de uso.....	15
4.2. Modelo Entidad-Relación.....	21
4.3. Modelo Relacional.....	24
5. Diseño de interfaces	26
6. Implementación.....	39
7. Pruebas.....	56
8. Conclusiones.....	58
9. Líneas futuras.....	59
10 Bibliografía.....	60

1. RESUMEN

En este documento se recoge la memoria del Proyecto de Fin de Carrera para la obtención del título de Ingeniero Técnico en Informática de Gestión en la Universidad Pública de Navarra.

Este proyecto que lleva por título “Aplicación Silverlight para la gestión de un almacén de suministros industriales” ha sido realizado por Héctor González Liñán.

En líneas generales, el proyecto tiene como objetivo desarrollar una aplicación que mejore la gestión de la empresa Fercos Suministros Industriales. Construyendo un interface que muestre, de una manera atractiva, procesada y clasificada esta información, con el propósito de mejorar la atención a los clientes y facilitar ciertas tareas a los empleados del almacén.

Para ello se ha utilizado Silverlight, el lenguaje C#, así como SQL SERVER para la gestión de la base de datos.

Para llevar a cabo este proyecto, se han seguido todas las fases del ciclo de vida de un producto: especificación de requisitos, análisis, diseño, implementación, pruebas e implantación.

2. INTRODUCCIÓN

2.1 Introducción a la empresa.



Ferco Suministros Industriales, fundada en 1985 en Corella (Navarra), se dedica al comercio de suministros para la industria, la construcción y la agricultura.

Actualmente dispone de unas instalaciones de 1000 metros cuadrados, dedicados a la exposición y venta de distintos productos para profesionales del sector.

Siendo uno de los más importantes almacenes para la adquisición de diferente material en la Ribera Navarra, su intención es actualizarse a las nuevas tecnologías, para mejorar considerablemente tanto el servicio ofrecido a sus clientes, como el rendimiento de la propia empresa de cara a las ventas.

Uno de los puntos fuertes de la empresa es su capacidad para plantear y desarrollar proyectos e iniciativas de mejora para superar esta situación de crisis y aumentar las ventas, introduciéndose en nuevos mercados, nuevos productos, colaboraciones con otras empresas del sector... ej.: creación de Navatools para la distribución de material especializado en el sector industrial y con un amplio catálogo, la venta por internet a través de EBay, etc.

Otro punto fuerte es la satisfacción del cliente: se valoran la calidad de los productos, la rapidez y la eficacia comercial.

2.2 Antecedentes y objetivos

Esta aplicación tiene su origen en dos carencias bien diferenciadas en la forma de trabajar en la empresa.

La primera es la utilización de formularios de papel realizados a mano, como forma de trabajar de la empresa, en los siguientes ámbitos:

-Gestión de los pedidos de cliente en el almacén. El motivo de que se trabaje de dicha forma radica en la imposibilidad de utilizar el programa de gestión (SQL PYME) que actualmente utiliza la empresa, para dicho fin.



FERCO Aragón FERCO rioja FERCO navarra FERCO aragón		
ARAGON FERCO Serradellada Industrial C/da del 100 50000 SERRADELLADA Tfno: 913 21 11 11 Fax: 913 21 11 11 E-mail: ferco@ferco.es	RIOJA Serradellada Rioja Norte C/da del 100 50000 SERRADELLADA Tfno: 913 21 11 11 Fax: 913 21 11 11 E-mail: ferco@ferco.es	NAVARRA Serradellada Rioja Norte C/da del 100 50000 SERRADELLADA Tfno: 913 21 11 11 Fax: 913 21 11 11 E-mail: ferco@ferco.es
FERCO		
PEDIDO 4389 FECHA: 07-08-2011		
CLIENTE: TOAX S.L.		
DESCRIPCION		
315	600 Set Profesional	4
26	Llave WKO 1/2" Cota	3
726	Balanzador de Cota	2

cliente

-Gestión de tareas de los empleados. Es necesario informatizar dicho escenario para llevar un control de las tareas de una forma más eficiente y ordenada.

La segunda es la necesidad de proporcionar a los clientes una información más detallada de los productos. Mediante la disposición de pantallas en el almacén, se podrá visualizar tanto las características de los productos como la localización dentro del almacén gracias a un modelado 3D del mismo. Incluye tanto los artículos que se dispone en dicho momento en el almacén, como la posibilidad de poder visualizar el amplio catálogo de cada uno de los fabricantes que se dispone.

Es por ello que los principales objetivos que se pretenden conseguir son:

Extender el servicio ofrecido a los clientes, ofreciendo una información detallada de los artículos disponibles en el almacén. En dicha información se incluirá la ubicación dentro del almacén.

Facilitar a los empleados la gestión de sus tareas diarias, a través de un gestor de tareas.

Mejorar la metodología utilizada para realizar pedidos por parte de los empleados.

2.3 Etapas de desarrollo

El proyecto ha seguido un desarrollo similar al modelo en Cascada o Clásico, a continuación se detallarán todas las etapas del ciclo de vida.

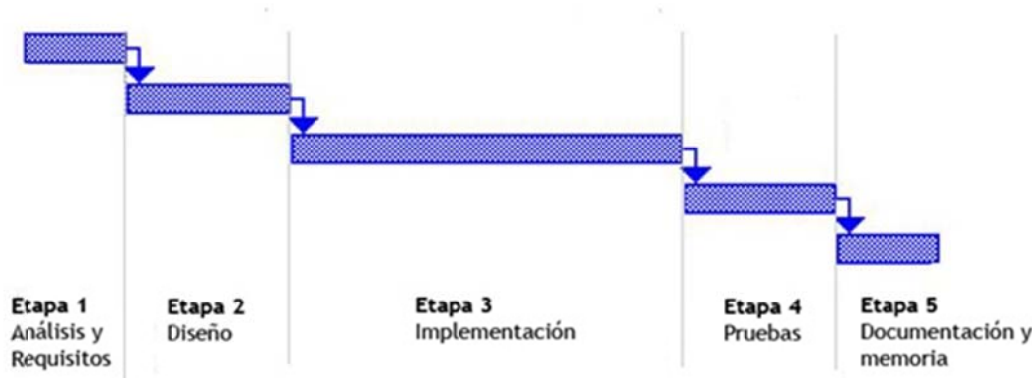


Figura2: Etapas de desarrollo del software

El proyecto se dividirá en las siguientes etapas:

- Etapa 1: Análisis de Silverlight como una estructura para aplicaciones Web, y especificación de todos los requisitos y funcionalidades que tendrá el proyecto. Se examinarán las diferentes herramientas (Visual Studio 2010, Microsoft Expression Blend, Expression Design, 3dsMax...), y en función de los requerimientos, se elegirán las más adecuadas.
- Etapa 2: Diseño de la estructura del sistema. En esta etapa se elaborara tanto el modelo de caso como el modelo Entidad Relación de la base de datos.
- Etapa 3: Implementación. A partir del diseño realizado en la tarea anterior se codificará la aplicación en Silverlight.
- Etapa 4: Periodo en el que se realizarán pruebas a nivel general del sistema, corrigiendo los errores y fallos que se vayan presentando.
- Etapa 5: Elaboración de la memoria del proyecto.

3. HERRAMIENTAS Y TECNOLOGIAS

En este apartado se describirán las distintas tecnologías y herramientas que se ha utilizado en el proceso de desarrollo de la aplicación. La elección de dichas tecnologías se debe a la intención de la empresa en desarrollar una aplicación web, y que a su vez permita el desarrollo de un interfaz amigable y sencillo.

Para ello se ha utilizado Visual Studio 2010 y Microsoft Expression para el desarrollo de Microsoft Silverlight 4 como estructura para aplicaciones web, implementado en código behind C#. SQL Server 2008 como gestor de bases de datos, y 3ds Max para la creación de un modelo 3D.

A continuación podrá encontrar una descripción de las distintas herramientas y tecnologías utilizadas:

3ds Max

Autodesk 3ds Max (anteriormente 3D Studio Max) es un programa de creación de gráficos y animación 3D desarrollado por Autodesk.



3ds Max es uno de los programas de animación 3D más utilizados. Dispone de una sólida capacidad de edición, una omnipresente arquitectura de plugins y una larga tradición en plataformas Microsoft Windows. 3ds Max es utilizado en mayor medida por los desarrolladores de videojuegos, aunque también en el desarrollo de proyectos de animación como películas o anuncios de televisión, efectos especiales y en arquitectura.

Incluye potentes herramientas integradas de modelado, animación, renderización y composición en 3D que multiplican rápidamente la productividad de los artistas y diseñadores. Ofrece herramientas y experiencias específicas a los desarrolladores de juegos, realizadores de efectos visuales y diseñadores gráficos, características especializadas para los arquitectos, diseñadores, ingenieros y especialistas en visualización.



Figura 3: Fotografía de las estanterías del almacén

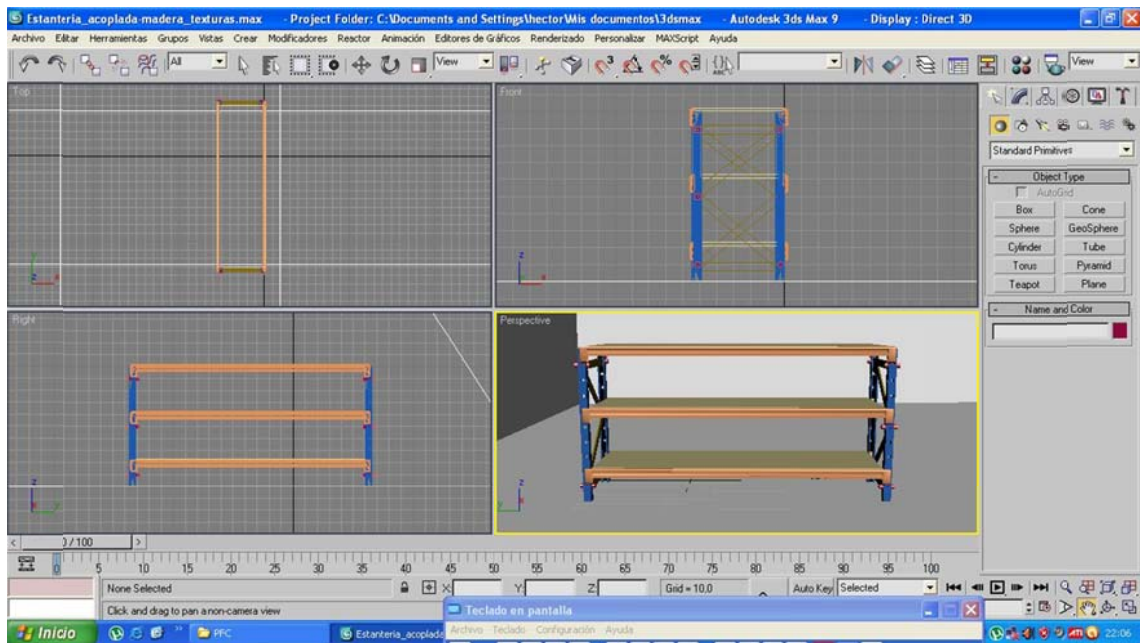


Figura 4: Modelado en 3DsMax de la estantería

Microsoft Visual Studio 2010



Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET. Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

Incluye .NET 4.0 Framework que es un modelo de programación integral y coherente de Microsoft para crear aplicaciones que proporcionen magníficas experiencias de usuario visuales, así como una comunicación eficaz y segura, y la capacidad de modelar una amplia gama de procesos empresariales. Incluye biblioteca ASP.NET AJAX para desarrollar aplicaciones web más eficientes, interactivas y altamente personalizadas que funcionen para todos los navegadores más populares y utilicen las últimas tecnologías y herramientas Web, incluyendo Silverlight.

Language-Integrated Query (LINQ) es un conjunto de características presentado en Visual Studio 2010 que agrega capacidades de consulta eficaces a la sintaxis de los lenguajes C# y Visual Basic. LINQ incluye patrones estándar y de fácil aprendizaje para consultar y actualizar datos, y su tecnología se puede extender para utilizar potencialmente cualquier tipo de almacén de datos. Visual Studio incluye ensamblados de proveedores para LINQ que habilitan el uso de LINQ con colecciones de .NET Framework, bases de datos SQL Server, conjuntos de datos de ADO.NET y documentos XML.

Microsoft Expression Studio 4



Introducción

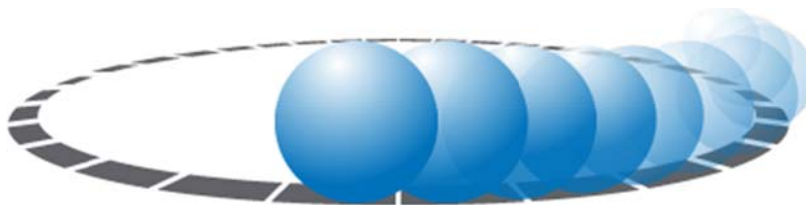
Microsoft Expression Studio es una herramienta de diseño profesional y contenido multimedia desarrollado por Microsoft, que proporcionan flexibilidad para elaborar potentes contenidos multimedia de manera sencilla.

Microsoft Expression Studio abre un nuevo mundo de posibilidades creativas. Sus herramientas de diseño profesional le dan la libertad para hacer su visión real, para diseñar sitios Web basados en normas-standards, experiencias ricas de escritorio, o Silverlight.

La versión de Expression Studio 4 incluyen los siguientes programas que funcionan de manera independiente:

- Expression Web: creación de sitios Web cumpliendo con los estándares.
- Expression Blend + SketchFlow: herramienta para el diseño de interacción en el Silverlight y .NET.
- Expression Design: diseño profesional para crear gráficos de contenido.
- Expression Media: visualización y catalogación de archivos.
- Expression Encoder: codificación de archivos.

A continuación se detallara la herramienta utilizada para el diseño de la aplicación Silverlight, Expression Blend 4.



Expression Blend 4

Expression Blend 4 es una herramienta profesional de diseño que le permite controlar la eficacia del XAML, .NET y Silverlight para Construir aplicaciones ricas y atractivas para el escritorio y la web.

SketchFlow, una característica de Expression Studio, te permite presentar ideas, flujos de iteración, pantallas y funcionalidad de la aplicación a tus clientes de forma rápida y con un prototipo efectivo. Transmite tus ideas de forma informal mediante dibujos y componentes 'esbozados' que ayudarán a tu cliente a centrarse en la funcionalidad del prototipo más que en el detalle, mientras que el reproductor de SketchFlow le dará a tu cliente la oportunidad de probar el prototipo y darte feedback, que puede ser visto directamente en contexto o en la superficie de diseño de Blend.

Expression Blend también incluye potentes importadores para Adobe Photoshop que permiten una sencilla integración. Durante la importación tienes la libertad de ver e importar archivos de Photoshop capa a capa. Las capas pueden agruparse fácilmente y los elementos mantienen sus formatos originales; capas, posición de capas, texto editable y vectores.

Expression permite crear animaciones. Una animación crea una ilusión mediante un ciclo rápido a través de una serie de imágenes, cada una de las cuales es ligeramente distinta de la anterior. El cerebro percibe la secuencia de imágenes como una sola escena que va cambiando. Las escalas de tiempo de animación se representan en XAML por medio de un elemento STORYBOARD. Puede controlar las escalas de tiempo de animación mediante programación en archivos de código subyacente usando C# o Visual Basic .NET.



Figura 5: Expression Blend 4 , Animación de la página principal.



Microsoft Silverlight 4

Microsoft Silverlight es un complemento de Microsoft que nos permite desarrollar aplicaciones enriquecidas para la web. Silverlight funciona sobre varias plataformas y múltiples exploradores y proporciona una nueva generación de experiencias de usuario basadas en .NET. Tras una descarga que se instala en segundos, Silverlight posibilita una nueva plataforma rica, segura y escalable.

Silverlight ofrece un modelo de programación flexible y coherente compatible con lenguajes .NET como Visual Basic, C# y F#, y otros como AJAX, Python, Ruby y que además se integra con las aplicaciones web existentes. Silverlight supone una nueva forma de aprovechar los elementos multimedia en los principales navegadores entre los que se incluyen Firefox, Safari e Internet Explorer tanto en MacOS como en Windows.

Experiencias de usuario atractivas para diferentes plataformas

- Proporciona experiencias multimedia enriquecidas (RIA) para la Web que incorporan vídeo, animaciones, interactividad e interfaces de usuario sensacionales.
- Pequeña instalación gracias a un complemento de unos 5Mb, el cual es fácil de instalar y que funciona en los principales navegadores.
- Crea experiencias web más completas y atractivas que aprovechan al máximo la capacidad del cliente para obtener el mejor rendimiento.
- Gráficos vectoriales, archivos multimedia, texto, animación y capas superpuestas permiten la integración perfecta de gráficos y efectos en cualquier aplicación web existente.

Un modelo de programación flexible con herramientas de colaboración

- Basadas en .NET Framework, Silverlight permite que los desarrolladores y diseñadores usen de una manera fácil las habilidades y herramientas existentes para proporcionar experiencias de medios y RIA para la Web.

- La integración sencilla con las tecnologías y los activos web existentes significa que Silverlight funciona con cualquier plataforma o tecnología web back-end. Silverlight se integra con su infraestructura y aplicaciones existentes, que incluyen Apache, PHP, así como JavaScript y XHTML en el cliente.
- Diferentes opciones de lenguajes de desarrollo que incluyen C#, Visual Basic.NET y F# además de JavaScript, Ruby, Python, etc.
- Expression Studio (diseñadores) para crear interfaces de usuario interactivas y experiencias de medios enriquecidas, preparar medios para su codificación y distribución, y crear sitios compatibles con estándares W3C mediante los nuevos XHTML, XML, XSLT, CSS y ASP.NET.
- Visual Studio (desarrolladores) para el desarrollo de código de cliente y servidor con la característica Intellisense completa, depuración eficaz en varias plataformas, compatibilidad con lenguaje enriquecido y mucho más.
- Modelo de presentación coherente con XAML, el lenguaje de presentación declarativo usado en aplicaciones de Windows 7. Los controles, diseños visuales, archivos multimedia y otros elementos se pueden presentar con total fidelidad de diseño en las aplicaciones de Silverlight y de Windows.

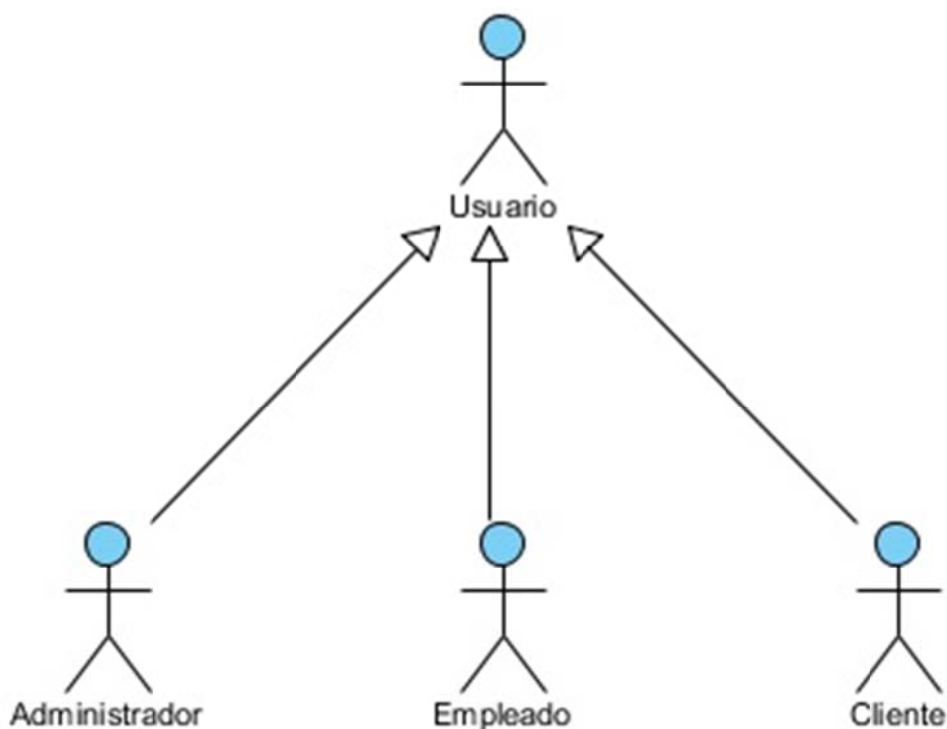
Totalmente multimedia

- El formato de archivos multimedia unificado abarca desde contenidos HD hasta formatos para dispositivos móviles mediante Vídeo de Windows Media (WMV), la implementación de Microsoft del estándar SMPTE VC-1 de video, al igual que compatibilidad con audio WMA y MP3.
- Soluciones flexibles de inserción de anuncios con vídeos y animaciones que incluyen la capacidad de proporcionar vídeo con estilo de emisión televisiva o anuncios animados sin pérdidas de fidelidad visual ni calidad de movimiento.

4. ANÁLISIS Y DISEÑO

4.1 Especificación de requisitos y casos de uso

La aplicación tendrá un sistema de creación de roles de usuario. Cada uno de ellos con unos permisos específicos. Los actores que participaran en el sistema son los siguientes: Un administrador del sistema que tendrá acceso a todas las funcionalidades, un empleado que tendrá ciertas restricciones con respecto al administrador, y por último el cliente. Hay que recalcar que el cliente no tiene que autenticarse para acceder a la aplicación.



A continuación se dividen los casos de uso según los actores involucrados, es decir, los casos de uso para el administrador, el empleado y el cliente.

ADMINISTRADOR DEL SISTEMA

Caso de Uso: Administración de usuarios por el administrador del sistema.

El administrador del sistema puede realizar la creación, modificación, eliminación y visualización de los usuarios.

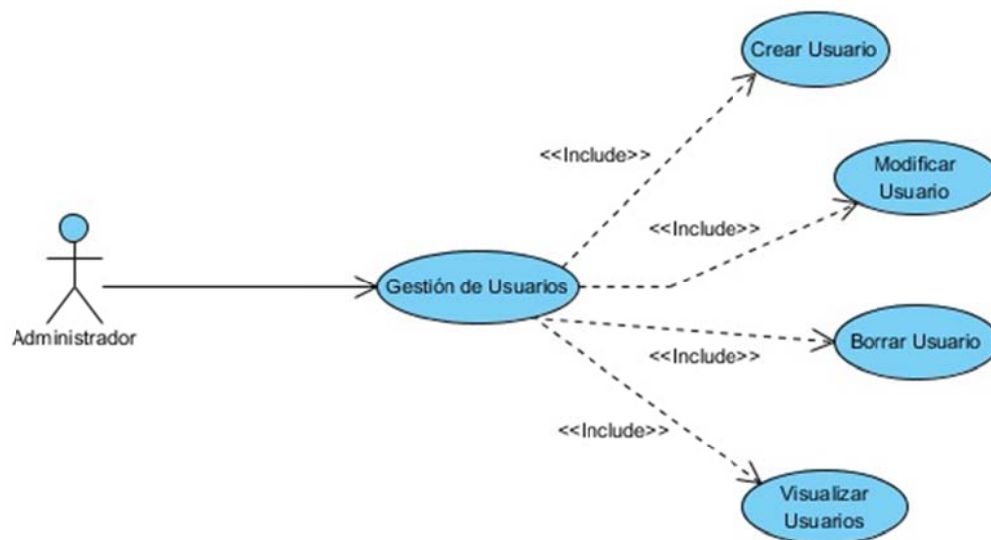


Figura 7: Gestión de usuario por el Administrador

Caso de uso: Gestión de tareas por el administrador del sistema.

El administrador del sistema podrá visualizar, crear, modificar y eliminar las tareas de cualquier usuario del sistema.

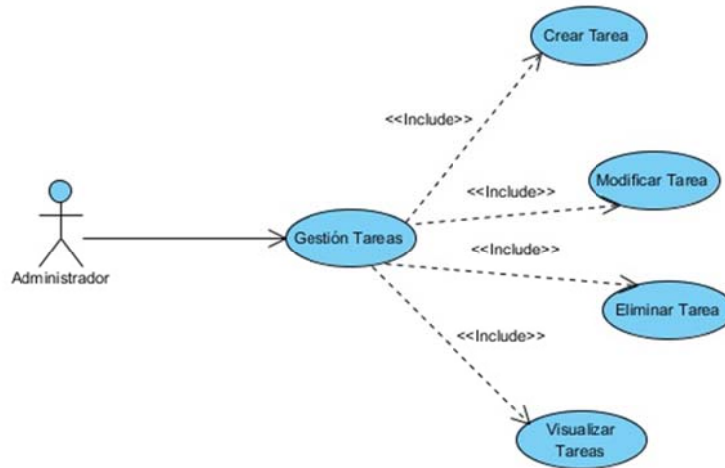


Figura 8: Gestión de Tareas por el Administrador

Caso de Uso: Gestión de pedidos por el administrador del sistema.

Crear, modificar, eliminar y visualización de pedidos. La creación de pedidos incluye añadir un artículo al pedido o eliminarlo del mismo. El administrador podrá gestionar los pedidos de todos los usuarios del sistema.

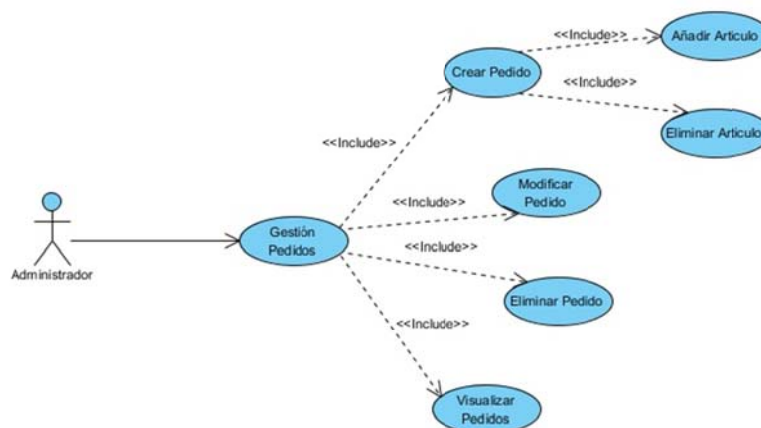


Figura 9: Gestión de Pedidos por el Administrador

Caso de Uso: Gestión de artículos por el administrador del sistema.

Puede visualizar un listado completo de los artículos, los fabricantes, las ofertas, y las diferentes familias de productos. A su vez, podrá visualizar los artículos según el Fabricante, y las Familias. Por último, permitirá vez la situación de los artículos dentro del almacén, y la modificación de la posición de los mismos.

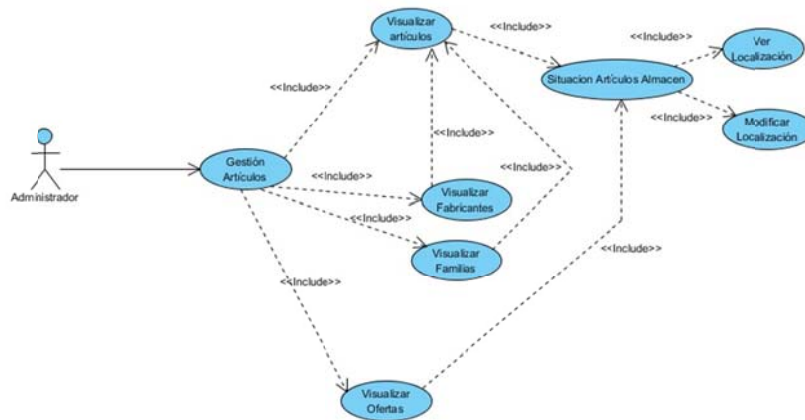


Figura 10: Gestión de artículos por el administrador

EMPLEADO

Caso de uso: Gestión de tareas por el empleado.

Crea, modifica, elimina, y visualiza tareas en el gestor de tareas.

El empleado únicamente podrá crear, visualizar, modificar y eliminar sus propias tareas.

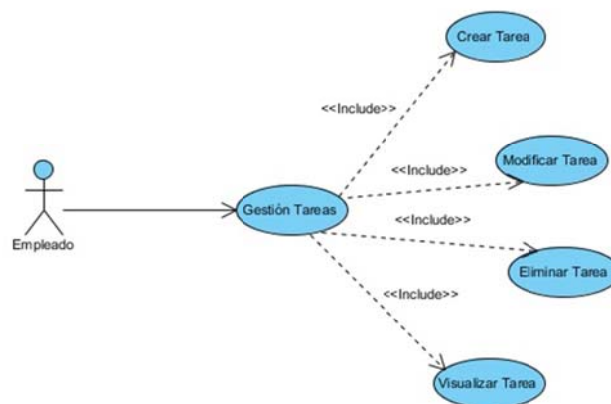


Figura 11: Gestión de Tareas por el Empleado

Caso de Uso: Gestión de pedidos por el empleado.

Crear, modificar, eliminar y visualización de pedidos. La creación de pedidos incluye añadir un artículo al pedido o eliminarlo del mismo.

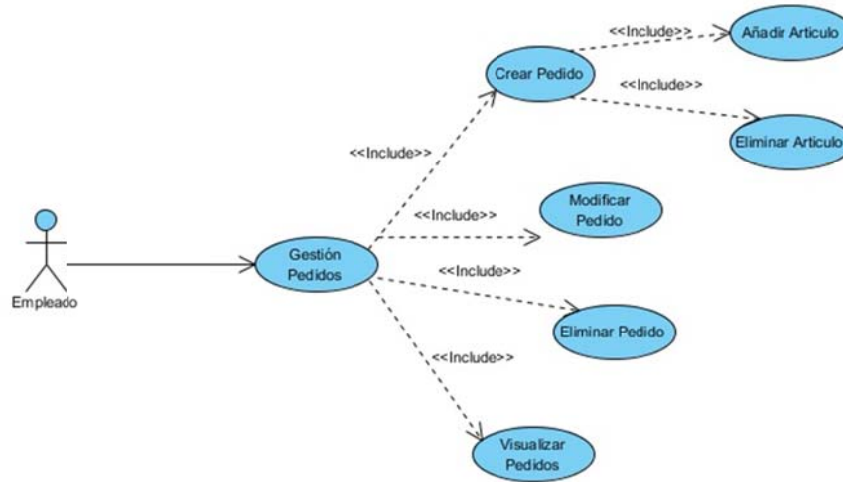


Figura 12: Gestión de pedidos por el empleado.

Caso de Uso: Gestión de artículos por el empleado.

Puede visualizar un listado completo de los artículos, los fabricantes, las ofertas, y las diferentes familias de productos. A su vez, podrá visualizar los artículos según el Fabricante, y las Familias. Por último, permitirá vez la situación de los artículos dentro del almacén, y la modificación de la posición de los mismos.

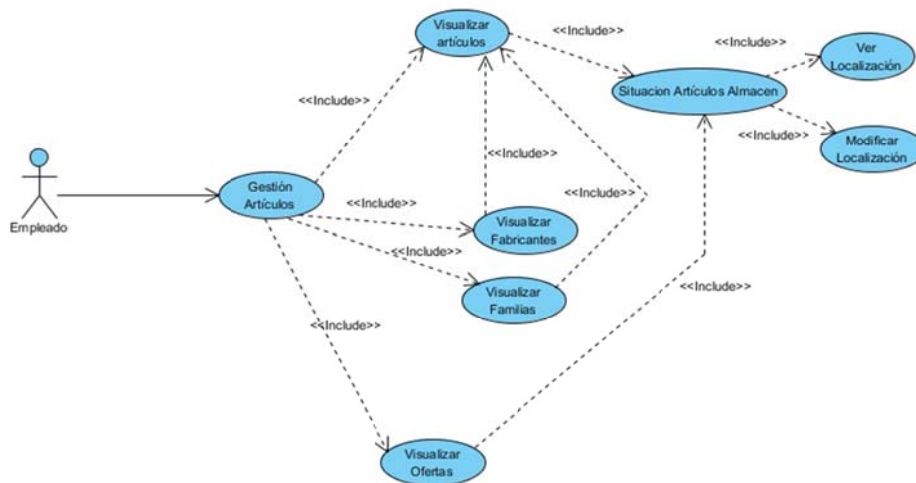


Figura 13: Gestión de artículos por el empleado.

CLIENTE

Caso de Uso: Gestión de artículos por el cliente.

Puede visualizar un listado completo de los artículos, los fabricantes, las ofertas, y las diferentes familias de productos. A su vez, podrá visualizar los artículos según el Fabricante, y las Familias. Por último, permitirá vez la situación de los artículos dentro del almacén.

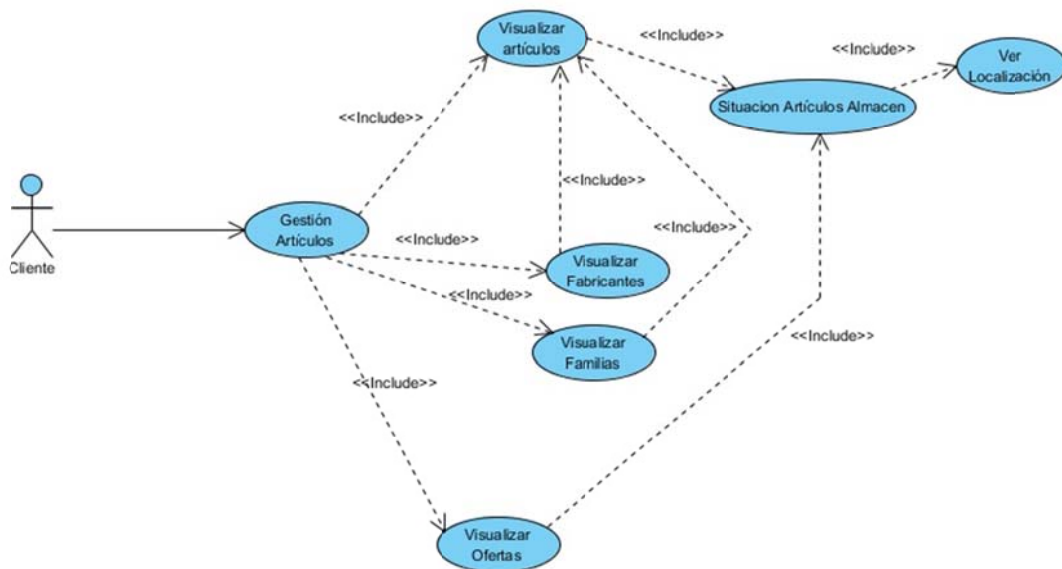
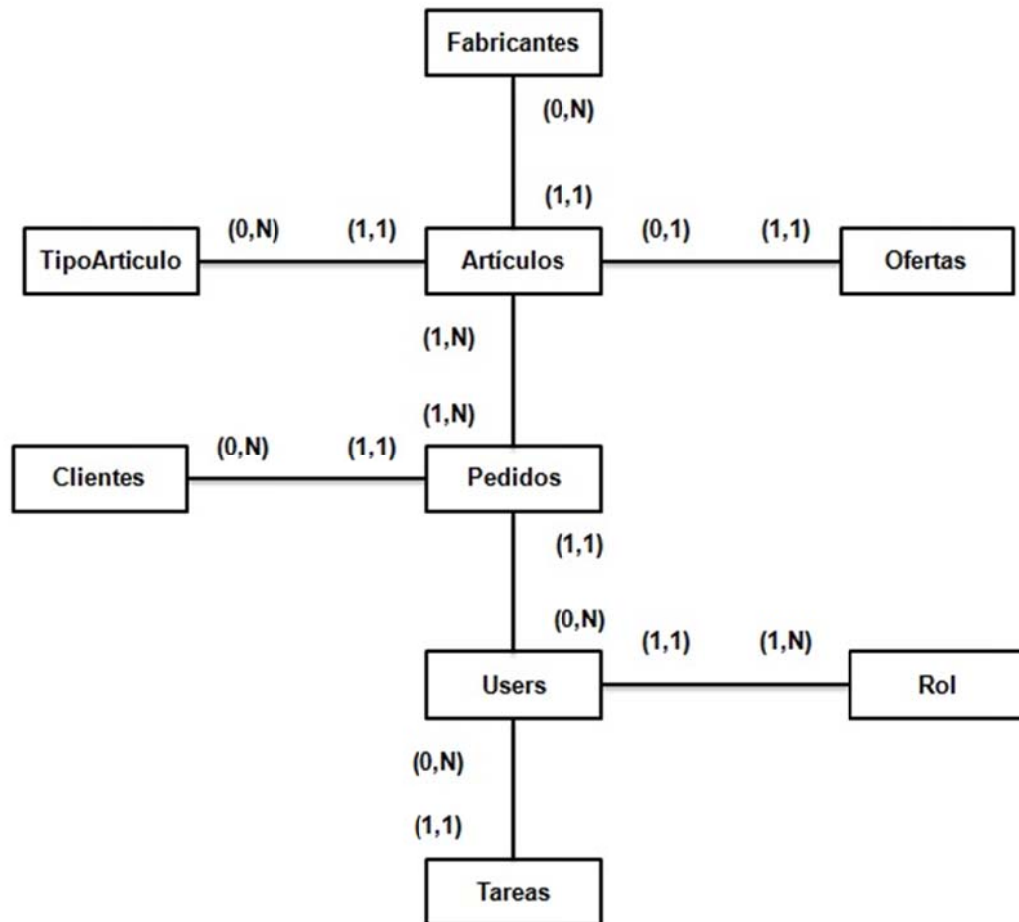


Figura 14: Gestión de artículos por el cliente.

4.2 Modelo Entidad-Relación



Las entidades principales del modelo E/R son:

Artículos: Entidad que representa los productos de los que se dispone para su venta, debemos conocer su código (que es único), nombre, descripción, referencia del proveedor, precio, stock, una imagen y video, y por último distintos campos que nos ayudaran tanto a posicionarlo como a representarlo gráficamente en el almacén (Numero de estantería, posición eje x, posición eje y, altura de la imagen y anchura de la misma).

Fabricantes: tendrá un código, nombre, descripción, una imagen que lo represente, su página Web y su catálogo.

Cientes: se les identifica por un código que es único, y su nombre.

Ofertas: se identificaran por un código y su precio.

Tipo de Artículo: tendrá un código, nombre, una breve descripción y una imagen que represente al conjunto de artículos de dicho tipo.

Users: los usuarios de la aplicación se identificaran por un código, nombre de usuario y una contraseña. Además tendrán un estado (activo o inactivo).

Rol: Identifica a cada usuario, y representa a un “Empleado” o al “Administrador”, se incluye por un código, un nombre y una descripción del mismo.

Tareas: cada usuario puede guardar una serie de notas o tareas, se identifican por un código, nombre, descripción, fecha de comienzo y fecha de finalización y por último un estado, que se clasificara como “Pendiente” o “Realizada”.

Pedidos: los pedidos de cliente se identificaran de la siguiente forma, un identificador, una descripción , el estado del pedido , que podrá ser “Pendiente Envío” “Enviado” y “Recibido” , la fecha de emisión del mismo , otra fecha que nos indica cuando se ha enviado al proveedor , y la ultima fecha que nos indica que el pedido ya se encuentra en el almacén .

Las relaciones del modelo E/R se detallan a continuación:

Artículos - Fabricantes: La relación existente es 1: N, es decir, un artículo pertenece a un único fabricante (1,1), y un fabricante puede tener ninguno o varios artículos (0, N).

Artículos – TipoArticulo: La relación es de 1: N, es decir, un artículo pertenece a un único tipo (1,1) y un Tipo puede tener ninguno o varios artículos (0, N).

Artículos – Ofertas: La relación es de 1:1, porque un artículo puede estar o no en oferta (1,0), y una oferta pertenece a un único artículo (1,1).

Artículos - Pedidos: La relación es de N: M, es decir, un artículo puede a uno o varios pedidos (1, N) y un pedido puede contener uno o varios artículos (1, N).

Pedidos – Clientes: La relación es de 1: N, ya que un pedido pertenece a un único cliente (1, 1) y un cliente puede realizar ninguno o varios pedidos (0, N).

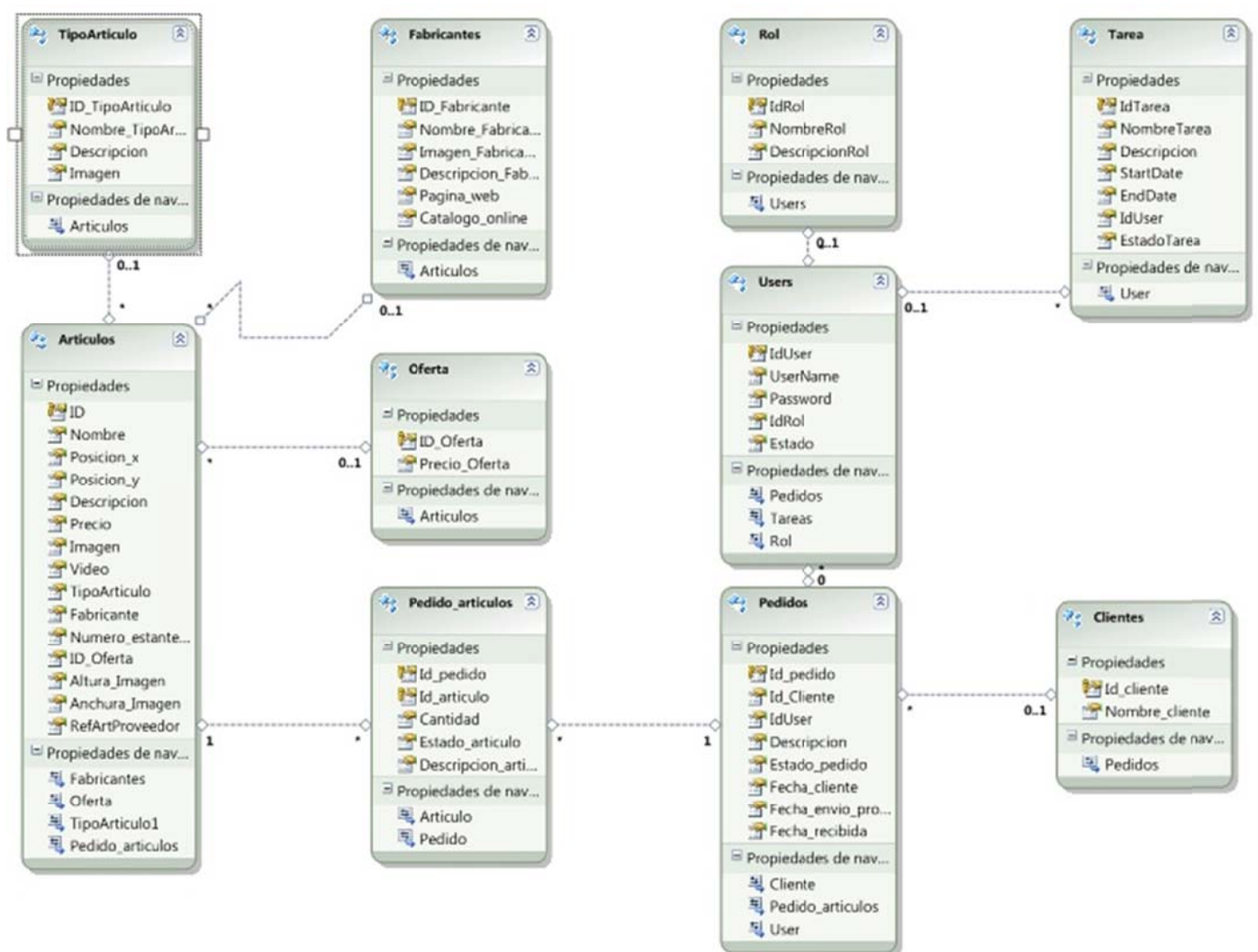
Pedidos – Users: La relación es de 1: N, porque un pedido es gestionado por un único usuario (1, 1), y un usuario puede gestionar ninguno o varios pedidos (0, N).

Users-Tareas: La relación es de 1: N, es decir, una tarea es realizada por único usuario (1,1), mientras que el usuario puede realizar ninguna o varias tareas (0, N).

Users-Rol: La relación es de 1: N, ya que un usuario pertenece a un único rol (1,1), y a un determinado rol pueden pertenecer ninguno o varios usuarios (0, N).

4.3 Modelo Relacional

Las tablas de la base de datos a las que da lugar el modelo son las siguientes:



Por último, es importante recalcar que la gestión de los artículos (esto incluye a las tablas artículos, fabricantes, ofertas, clientes, TipoArticulo) es realizada por el programa de gestión SQL PYME.

Nuestra aplicación únicamente se encargara de mostrar la información detallada anteriormente, añadiendo la gestión de los pedidos, tareas, y usuarios.

La siguiente imagen muestra el alta de un artículo dentro SQL PYME.

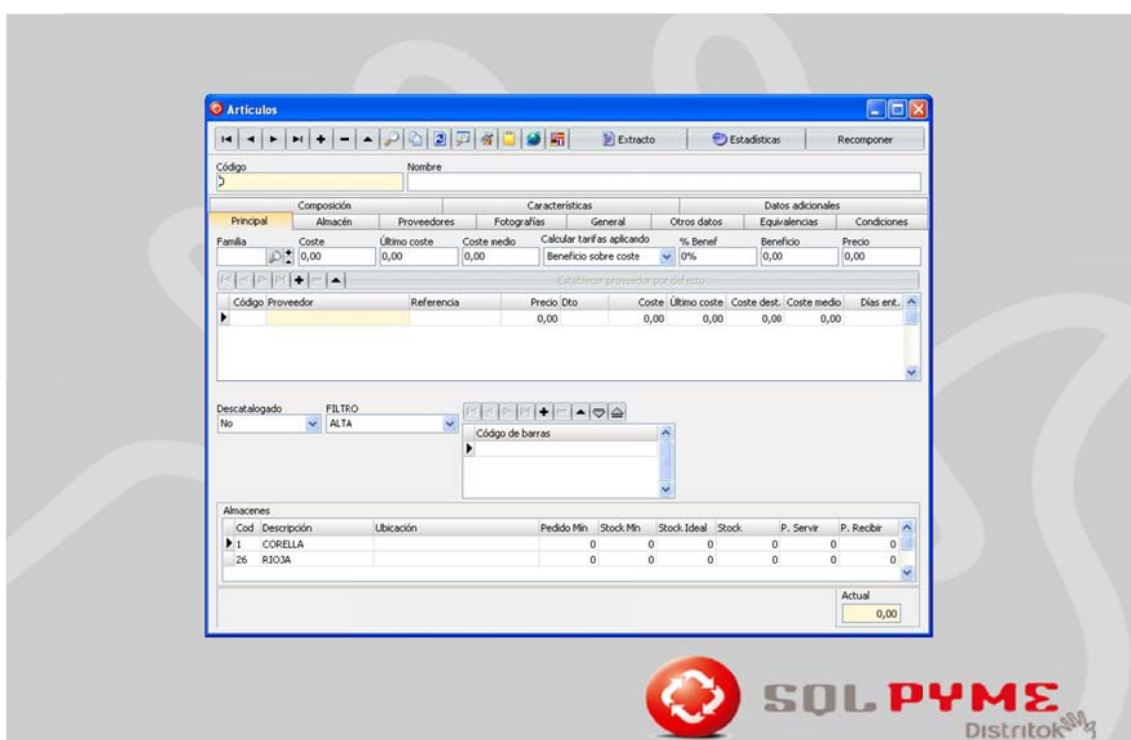


Figura 17: Alta de un artículo programa de gestión SQL PYME



5. DISEÑO DE INTERFACES

Los usuarios de un programa a nivel empresarial tienen ciertas expectativas sobre su apariencia. El programa debe dar la impresión de ser diseñado profesionalmente. El problema es que la mayoría de desarrolladores tienen poca experiencia en ese campo y escasos recursos como para contratar a un diseñador profesional. Con el fin de aliviar la situación el grupo a cargo de Silverlight ha invertido gran cantidad de tiempo en crear plantillas de proyectos para programas comerciales como la Business Silverlight Application. Las plantillas generan proyectos estructurados y programas de apariencia simple y agradable, fácil de adaptar al gusto inclusive por programadores.

Además la página Web de Silverlight nos proporciona de forma gratuita plantillas con determinados estilos, para el diseño de esta interfaz se ha elegido jetpack Theme.

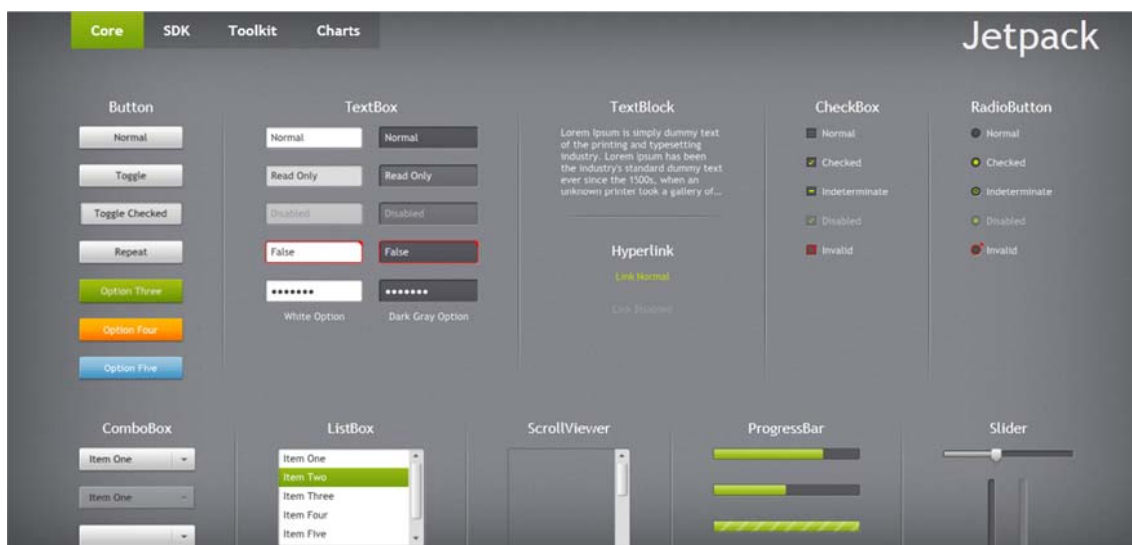


Figura 18: Ejemplo controles de la plantilla jetpack Theme.

A continuación se detallarán las principales pantallas de las que consta la aplicación:

ARTÍCULOS

No es necesario estar autenticado para acceder a la visualización de los artículos.

Se puede ver un listado completo de los artículos, una imagen de los mismos, una breve descripción y la localización del artículo dentro del almacén.

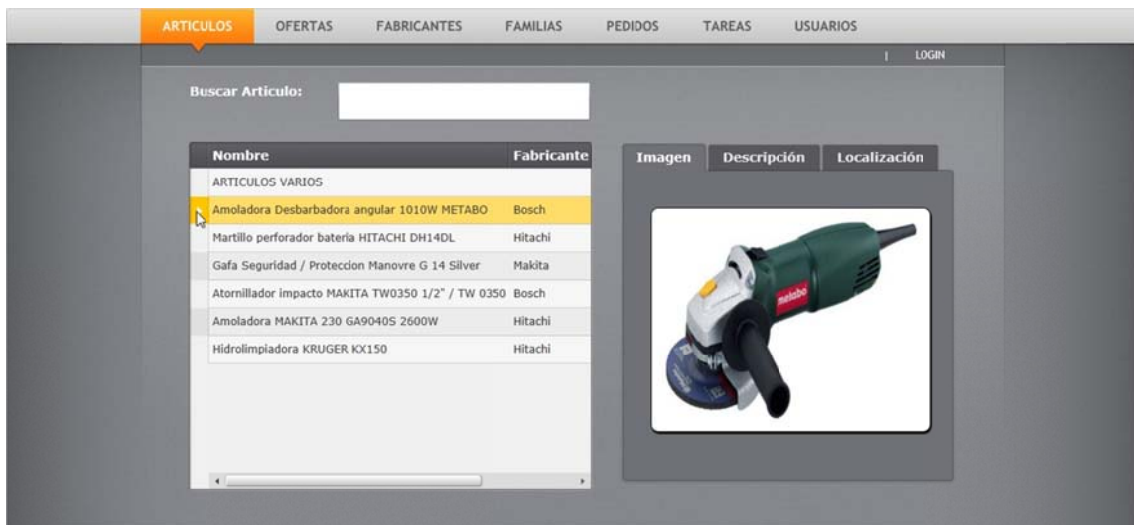


Figura 19: Listado de los artículos con visualización de la imagen



Figura 20: Listado de los artículos con visualización de la descripción

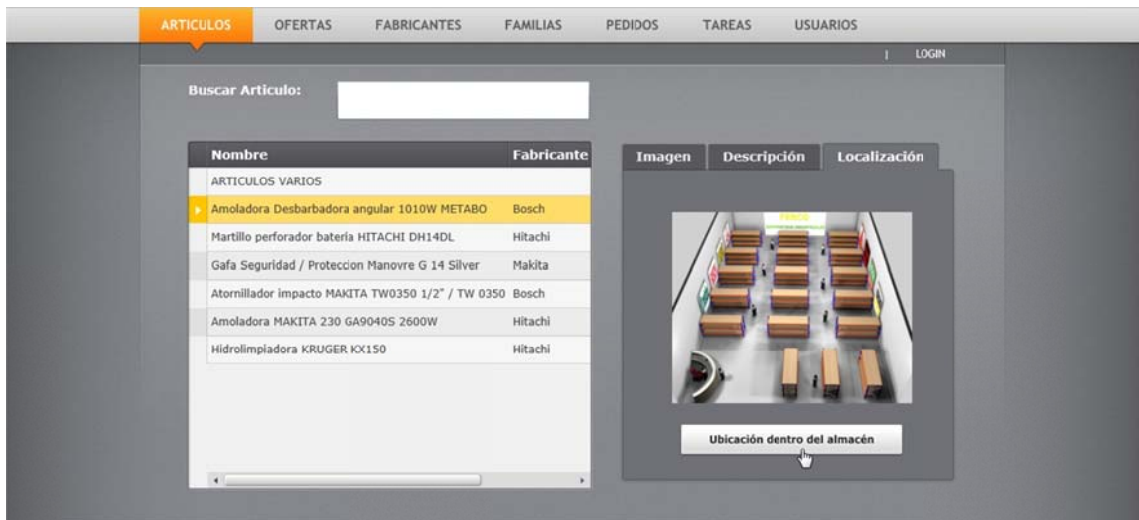


Figura 21: Listado de los artículos con la opción de verlos dentro del almacén

OFERTAS

Se detalla un listado de las ofertas y el precio anterior de cada una de ellas. No hay que estar autenticado.

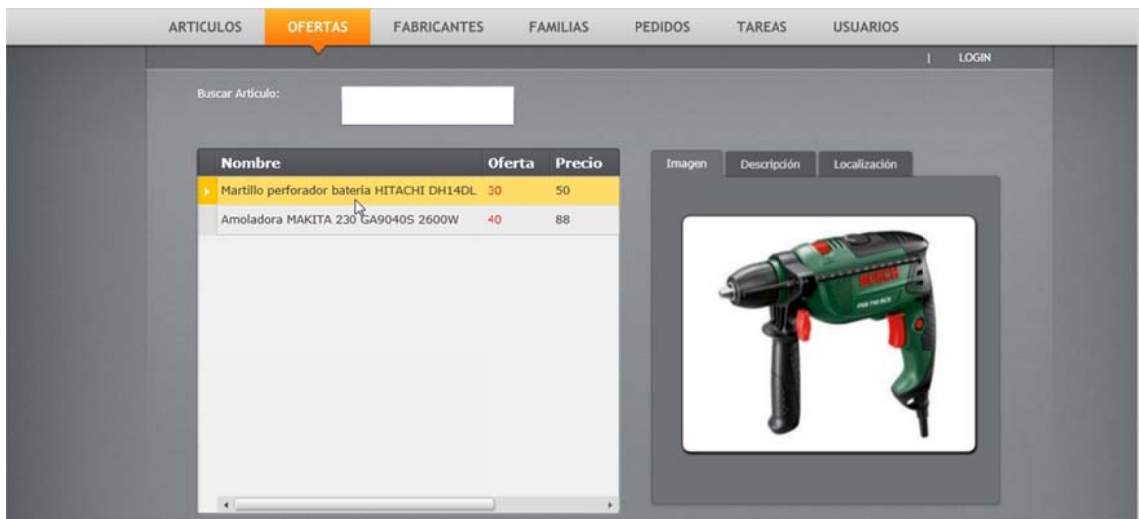


Figura 22: Visualización de las ofertas

FABRICANTES

Un listado de los fabricantes que proporcionan diferente material a la empresa. Se puede acceder a la página web y al catálogo online. Igualmente se podrán visualizar los artículos del fabricante elegido de los que se dispone en el almacén. No hay que estar autenticado.



Figura 23: Listado de los fabricantes

FAMILIAS

Se pueden obtener distinta información acerca de las distintas familias de productos. No es necesario autenticarse.



Figura 24: Listado de las diferentes familias de artículos

LOCALIZACIÓN EN EL ALMACÉN

A través de un sencillo diseño 3D se puede visualizar tanto la posición del artículo seleccionado dentro de la estantería, como la ubicación que ocupa dentro de la estantería del almacén.

Si usuario de la aplicación no está autenticado, únicamente podrá visualizar la pantalla, en cambio, como se ve en la siguiente imagen, si el usuario esta autenticado puede modificar el artículo de estantería, o moverlo dentro de su propia estantería con el ratón. Existe otra modificación a la que llamamos “Reorganizar almacén”, que puede verse en la figura 26, en la cual se puede modificar el tamaño de la imagen que representa a cada uno de los artículos. Además permite navegar por todas las estanterías para realizar las modificaciones oportunas.



Figura 25: Ubicación de un determinado artículo



Figura 26: Reorganización del almacén

GESTIÓN DE PEDIDOS DE CLIENTE

Para acceder a la sección de pedidos el usuario debe de estar autenticado.

Es decir, debe de ser empleado o administrador del sistema.

Una vez que accedemos a la pantalla de gestión de Pedidos, esta nos permite realizar:

Un nuevo pedido, la modificación de un pedido existente, la eliminación de los mismos, y su visualización.

Se puede observar en la parte superior derecha un listado de los pedidos.

En la parte superior izquierda podemos elegir que nos muestre los pedidos por sus diferentes estados (“Pendiente envío”, “Enviado”, Recibido”) y por diferentes fechas.

Abajo se muestra el listado de los artículos que pertenece a cada pedido.



Figura 27: Pantalla principal de la gestión de pedidos

Nuevo Pedido

Nos permite elegir el cliente que ha realizado el pedido y una breve descripción del mismo.

Después se procede a elegir los artículos que se han pedido. Para ello primero se busca el artículo, después se da una descripción del artículo y la cantidad.

Esta descripción del artículo es necesaria para determinados artículos. Existen unos artículos que se denominan “Artículos varios”, los cuales no están dados de alta dentro de la gestión, por lo general, son artículos nuevos. Es por ello que necesitamos de esta descripción para poder identificar al artículo.

En este pantallazo se puede observar que se ha seleccionado un “Artículo Vario” y dentro de la especificación del artículo se dice cual es la referencia del proveedor y el tipo.

Cuando hemos introducido la información correcta añadimos el artículo, también se permite eliminar un artículo del datagrid, por si se ha cometido algún fallo.

ARTICULOS

Cliente : Ergabia

Descripción : Llamar a Oscar . 694856049, cuando llegue la mercancía.

Pedido

Artículo : ARTICULOS VARIOS

Descripción : Ref.4543 Makita Atornillador

Artículo

Cantidad : 4

Idartículo	ReferenciaProveedArt	NombreArt	Cantidad
3	321321	Martillo perforador batería HITACHI DH14DL	3

Modificación de Pedido

Podemos marcar el estado de un pedido como Enviado o Recibido.

Con los pedidos de un determinado artículo también se puede modificar su estado, y además podemos añadir un artículo al pedido o borrarlo del mismo.



Figura 29: Modificación de pedido

Eliminación de Pedido

Por último, si deseamos borrar un pedido, nos muestra un mensaje de confirmación en el que se nos informa del pedido que deseamos borrar.



Figura 30: Mensaje de confirmación de eliminación de pedido

GESTIÓN DE TAREAS

Para acceder a la gestión de Tareas hay que estar registrado como Empleado o Administrador. En dicha pantalla se visualizan las tareas que el usuario tiene pendientes, así como una breve descripción de cada una de ellas. El usuario tendrá la opción de crear una nueva tarea, modificar una tarea, borrar la tarea, o ver un listado completo de todas las tareas que ha realizado.

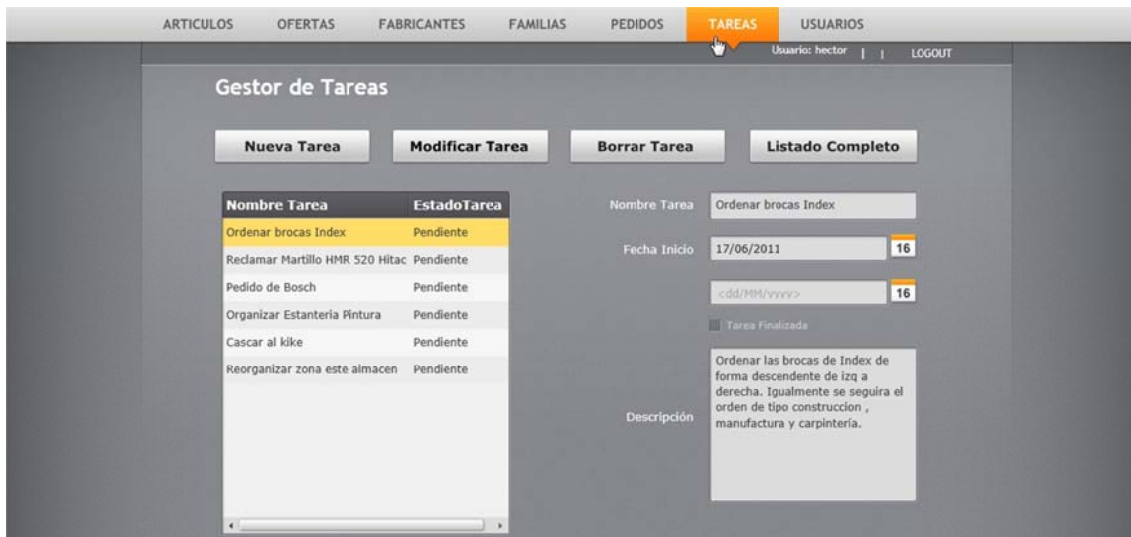


Figura 31: Gestor de Tareas

Creación de Tareas

El usuario podrá crear una nueva Tarea, en tal caso, se mostrara una ventana emergente con la información necesaria a cumplimentar.

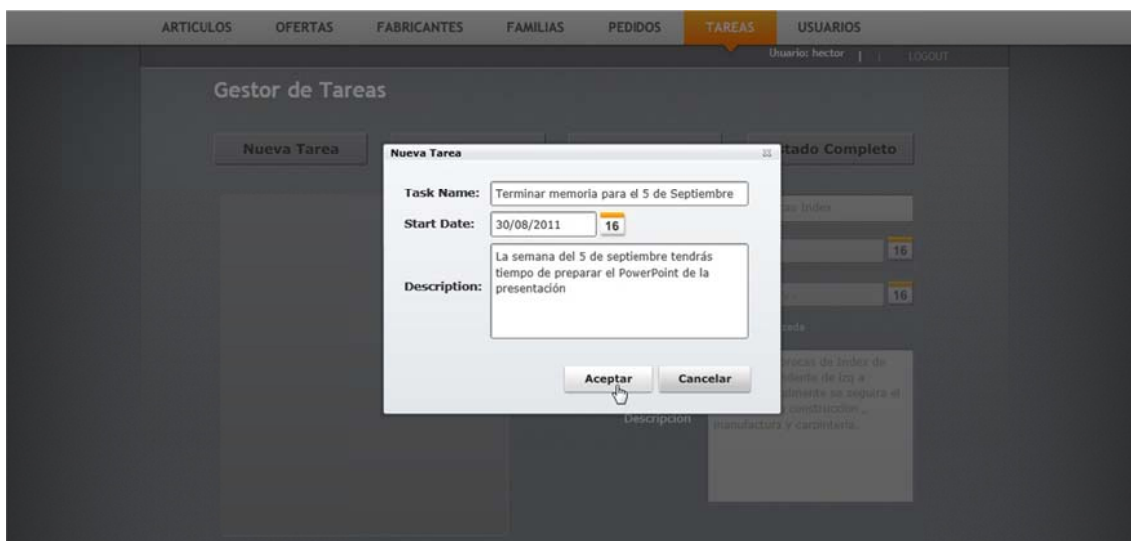


Figura 32: Pantalla creación de nueva tarea.

Modificación de Tareas

El usuario dispone de una lista de Tareas que puede seleccionar para su posterior modificación. Podrá modificar tanto el nombre como la descripción y en caso de que la tarea haya sido terminada podrá marcarse como tal.

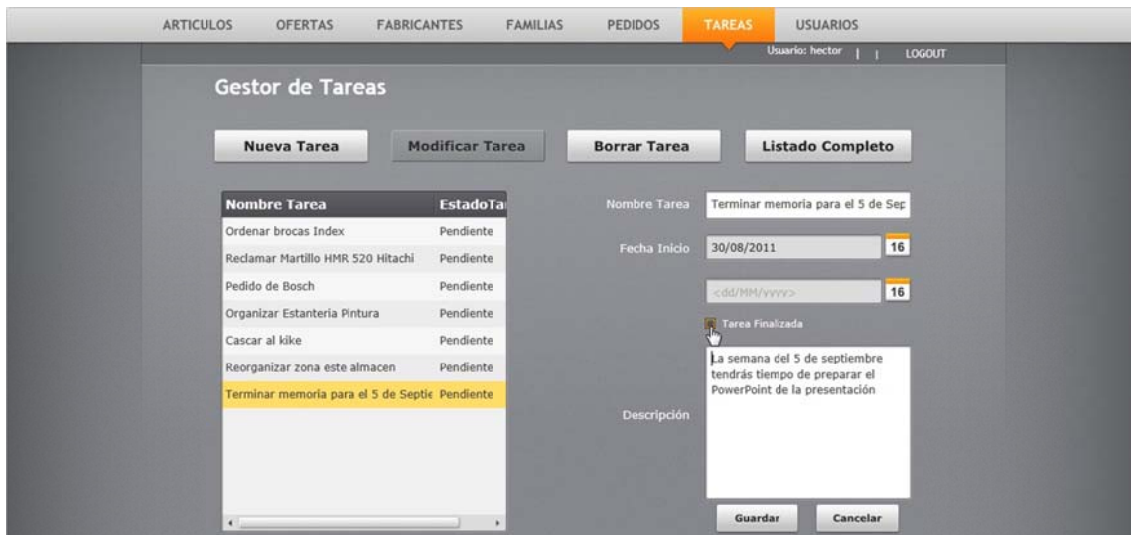
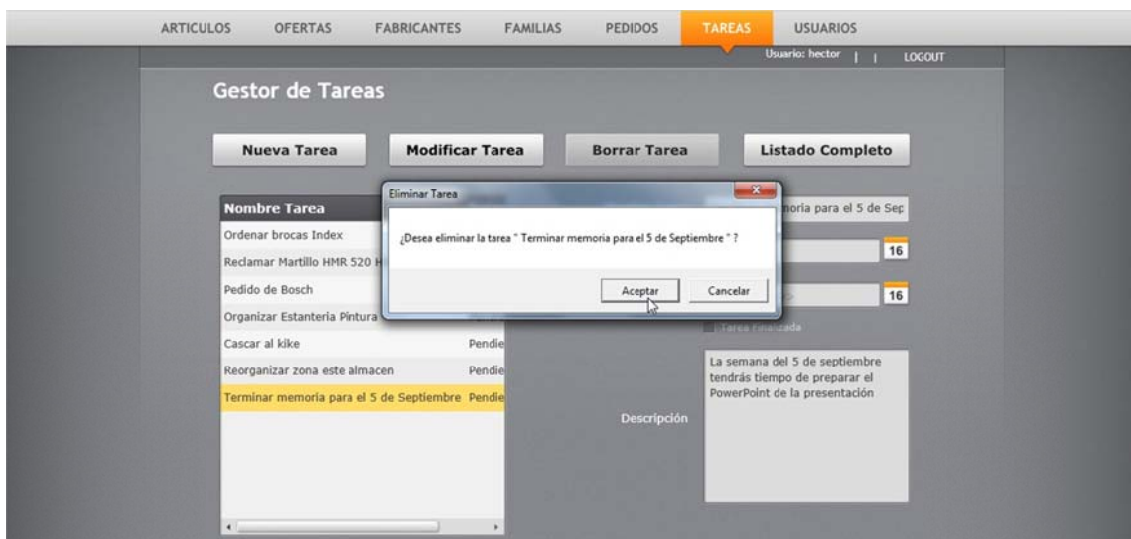


Figura 33: Modificación de una tarea seleccionada

Eliminación de Tareas

El sistema pregunta si se desea eliminar la tarea seleccionada.



Visualización del listado completo de Tareas

La pantalla inicial de Gestión de Tareas únicamente ofrece información de aquellas tareas que están pendientes, es decir, no han sido finalizadas.

Es por esto que necesitamos una pantalla que nos muestre un listado completo de las tareas del usuario, y que además permita organizarlos por fecha.



Figura 35: Listado de tareas que han finalizado entre el 15 de Julio y el 31 de Agosto

Gestión de Usuarios

Consulta y mantenimiento de los usuarios por parte de/los administradores. Se puede crear un usuario, modificarlo, darlo de baja, darlo de alta y eliminarlo.

Se ha hecho una pequeña modificación respecto al planteamiento inicial, en el cual un usuario podía ser borrado sin tener en cuenta si tenía pedidos o tareas bajo su responsabilidad. Como se quiere mantener un historial de todos los pedidos y tareas, aquel usuario que tenga alguno de estos bajo su nombre, no podrá ser borrado. Para ello se han introducido las opciones de alta de usuario, y baja de usuario.

Login de usuario: acceder a Pedidos y Tareas en el caso del Empleado. Y para la gestión de usuarios además de las dos anteriores por parte del administrador.





Figura 36: Pantalla principal del gestor de usuarios

Nuevo Usuario

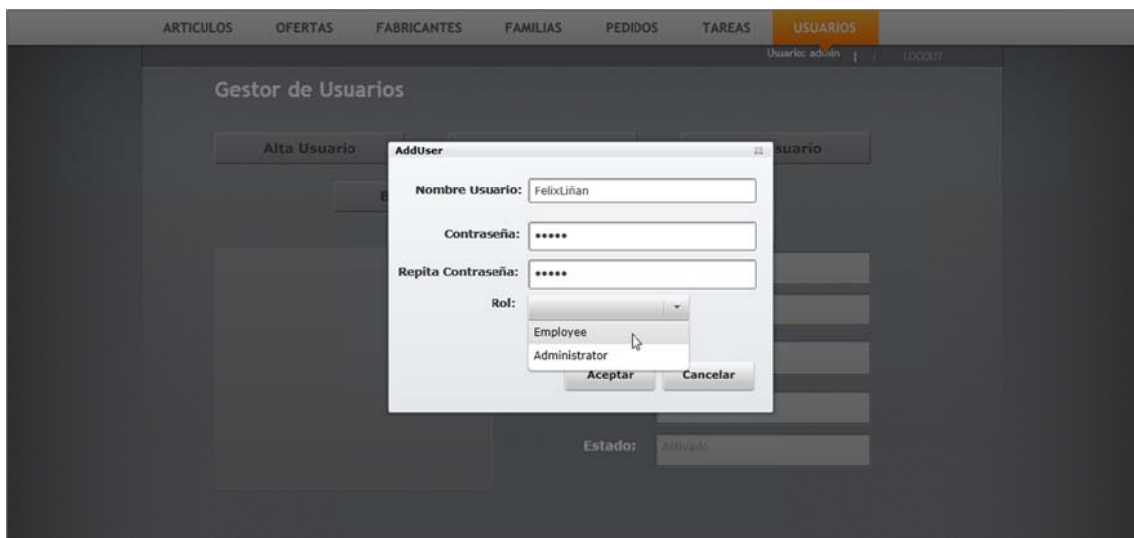


Figura 37: Creación de un nuevo usuario

Modificación Usuario

Únicamente se permite modificar el nombre de usuario, la contraseña y el Rol (“Empleado o Admin”).



Figura 38: Modificación de usuario

Alta, Baja y eliminación de Usuario

Si damos de baja a un usuario, este ya no tendrá acceso al sistema. Si queremos que pueda volver a acceder al sistema lo damos de alta. Y si queremos eliminarlo, no podrá tener tareas y pedidos bajo su nombre.

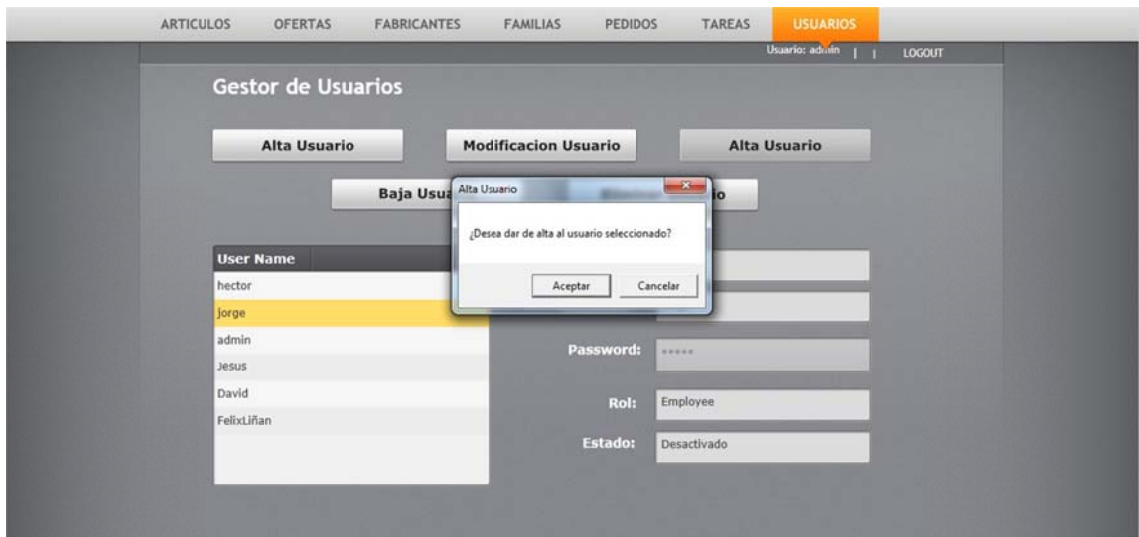


Figura 39: Aviso de alta de usuario

6. IMPLEMENTACIÓN

La mayoría de las aplicaciones giran en torno a la lectura y actualización de información en bases de datos. Para habilitar la integración de datos en aplicaciones distribuidas y escalables, Visual Studio proporciona diferentes recursos para integrar datos en las aplicaciones. Las aplicaciones basadas en Silverlight pueden obtener acceso a los datos mediante varias tecnologías. Para esta aplicación vamos a utilizar WCF Ria Services.

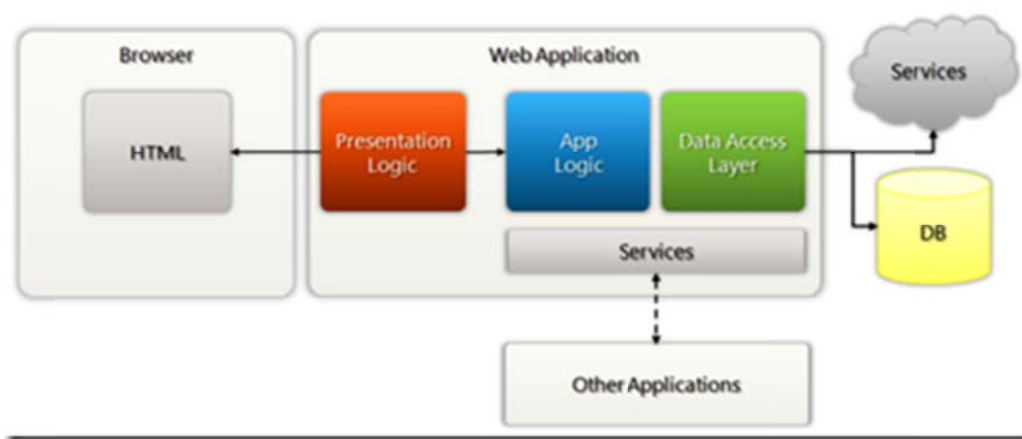
WCF RIA SERVICES

Introducción

WCF RIA Services simplifica el desarrollo de soluciones de n niveles para aplicaciones de Internet enriquecidas (RIA), como las aplicaciones de Silverlight. Un problema común a la hora de desarrollar una solución de RIA de n niveles es la coordinación de la lógica de aplicación entre el nivel intermedio y el nivel de presentación. Para crear la mejor experiencia de usuario, desea que el cliente de RIA Services conozca la lógica de aplicación que reside en el servidor pero no desea que se desarrolle ni se mantenga la lógica de aplicación tanto en el nivel de presentación como en el nivel intermedio. RIA Services soluciona este problema proporcionando componentes, herramientas y servicios de marco que ponen la lógica de aplicación del servidor a disposición del cliente de RIA Services sin que sea necesario duplicar manualmente la lógica de programación. Puede crear un cliente de RIA Services que conozca las reglas de negocios y sepa que el cliente se actualiza automáticamente con la última lógica de nivel intermedio cada vez que se recompila la solución.

RIA Services agrega a Visual Studio 2010 herramientas que permiten vincular proyectos de cliente y de servidor en una solución única y generar código para el proyecto de cliente a partir del código de nivel intermedio. Los componentes de marco admiten patrones preceptivos para escribir lógica de aplicación de modo que se pueda reutilizar en el nivel de presentación. Se proporcionan servicios para escenarios comunes, como administración de configuración de autenticación y de usuario, para reducir el tiempo de desarrollo.

Web tradicionales



Las webs tradicionales por un lado tienen el navegador basado exclusivamente en HTML alojado en el navegador y por el otro las diferentes capas de la aplicación (Presentación, lógica y de Acceso a Datos). Lo que conlleva pesados desarrollos del lado del servidor, además del retardo en el tiempo de respuesta a la hora de trabajar con los datos contenidos en el servidor.

Aplicaciones R.I.A

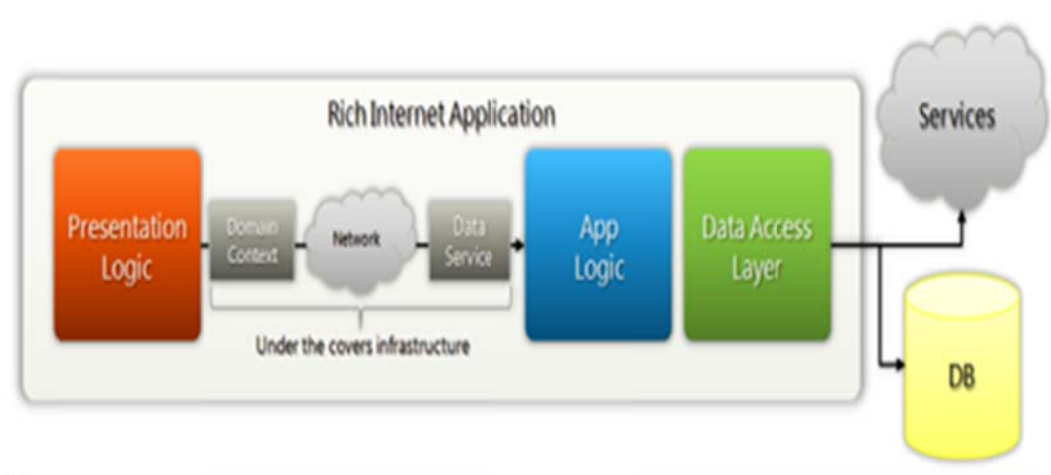


Figura 41: Estructura de una web RIA

Con la llegada de Silverlight la lógica de presentación se nos va al lado del cliente, consiguiendo con ello un cliente mucho más versátil, eso sí, también más pesado.

El objetivo de R.I.A Services es crear una capa intermedia entre la lógica de presentación y la de negocio. La mayor dificultad la encontramos en que las comunicaciones con las fuentes de datos no se realizan de manera directa, para ello nos ayudamos de WCF.

Integración de WCF

En RIA Services, se puede exponer datos del proyecto de servidor en el proyecto de cliente agregando servicios de dominio. El marco de RIA Services implementa cada servicio de dominio como un servicio de Windows Communication Foundation (WCF). Por lo tanto, se puede aplicar los conceptos de WCF Services a los servicios de dominio a la hora de personalizar la configuración.

Los servicios de dominio son servicios de Windows Communication Foundation (WCF) que encapsulan la lógica de negocios de una aplicación de WCF RIA Services. Un servicio de dominio expone un conjunto de operaciones relacionadas en forma de una capa de servicio. Cuando se define un servicio de dominio, se deben especificar las operaciones de datos permitidas a través del servicio de dominio.

Cuando se diseña un servicio de dominio, se debe pensar en el servicio de dominio como un conjunto de tareas relacionadas que se espera que los usuarios realicen en la aplicación. Normalmente, dichas tareas afectan a un grupo pequeño de entidades estrechamente relacionadas.

Uno de los aspectos principales de aplicaciones RIA es la transferencia de información del servidor al cliente y viceversa. Para demostrar los servicios RIA de .NET necesitamos entonces una fuente de información , que será la base de datos.

Utilizaremos un ADO.NET Entity Framework para crear un modelo de la base de datos.

ADO.NET Entity Framework

ADO.NET Entity Framework permite a los desarrolladores crear aplicaciones de acceso a datos programando con un modelo de aplicaciones conceptuales en lugar de programar directamente con un esquema de almacenamiento relacional. El objetivo es reducir la cantidad de código y el mantenimiento necesarios para las aplicaciones

orientadas a datos. Las aplicaciones de Entity Framework ofrecen las siguientes ventajas:

- Las aplicaciones pueden funcionar en términos de un modelo conceptual más centrado en la aplicación, que incluye tipos con herencia, miembros complejos y relaciones.
- Las aplicaciones están libres de dependencias de codificación rígida de un motor de datos o de un esquema de almacenamiento.
- Las asignaciones entre el modelo conceptual y el esquema específico de almacenamiento pueden cambiar sin tener que cambiar el código de la aplicación.
- Los desarrolladores pueden trabajar con un modelo de objeto de aplicación coherente que se puede asignar a diversos esquemas de almacenamiento, posiblemente implementados en sistemas de administración de base de datos diferentes.
- Se pueden asignar varios modelos conceptuales a un único esquema de almacenamiento.
- La compatibilidad con Language Integrated Query (LINQ) proporciona validación de la sintaxis en el momento de la compilación para consultas en un modelo conceptual.

Herramientas de ADO.NET Entity Data Model

Las herramientas de Entity Data Model están diseñadas para ayudar a compilar aplicaciones de Entity Framework. Con las herramientas de Entity Data Model puede crear un *modelo conceptual* a partir de una base de datos existente. Después, puede visualizar y modificar gráficamente dicho modelo. – O bien –, primero puede crear gráficamente un modelo conceptual y, a continuación, generar una base de datos que admita su modelo. En cualquier caso, puede actualizar el modelo automáticamente cuando la base de datos subyacente cambie y generar automáticamente el código de capa de objeto para la aplicación. La generación de bases de datos y la generación del código de capa de objeto son personalizables.

En la siguiente lista se describen las herramientas específicas que constituyen las herramientas de Entity Data Model:

- ADO.NET Entity Data Model Designer (Entity Designer) le permite crear y modificar visualmente entidades, asociaciones, asignaciones y relaciones de herencia. Además, Entity Designer también genera código de capa de objeto de C# o Visual Basic.
- El Asistente para Entity Data Model le permite generar un modelo conceptual a partir de una base de datos existente y agrega información de conexión de la base de datos a la aplicación.
- El Asistente para crear base de datos le permite crear primero un modelo conceptual y, a continuación, crear una base de datos que lo admita.
- El Asistente para actualizar modelo le permite actualizar el modelo conceptual, el modelo de almacenamiento y las asignaciones cuando se efectúen cambios en la base de datos subyacente.

Las herramientas generan o modifican un archivo .edmx, que contiene información que describe el modelo conceptual, el modelo de almacenamiento y las asignaciones entre ellos. Como puede verse en la siguiente imagen:

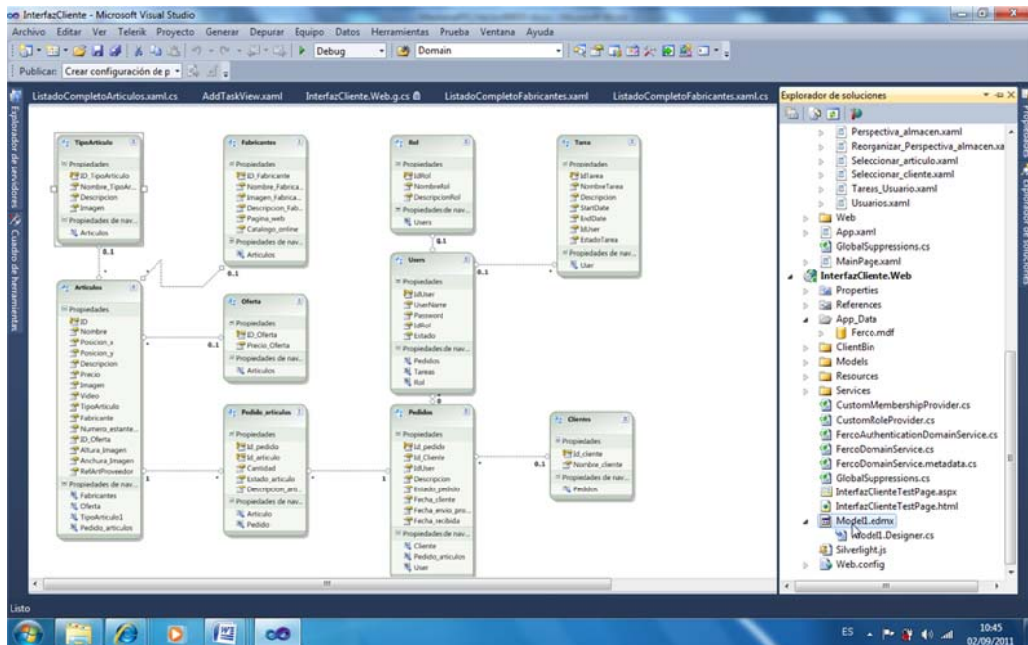


Figura 42: Entity Data Model

Servicios de dominio y sus orígenes de datos

La clase `DomainService` es la clase base para todas las clases que actúan como servicios de dominio. Para crear un servicio de dominio que enlaza a un objeto de datos personalizado, debe crear una clase que derive directamente de `DomainService`. Pero si tiene un servicio de dominio que enlaza un Entity Data Model de ADO.NET o que expone un vínculo a una base de datos SQL, puede utilizar en su lugar clases abstractas especiales que derivan de `DomainService`.

- Para crear un servicio de dominio que enlaza a un Entity Data Model de ADO.NET, debe crear una clase que derive de `LinqToEntitiesDomainService`. RIA Services proporciona la clase `LinqToEntitiesDomainService`.

Cuando se utiliza el cuadro de diálogo Agregar nueva clase de servicio de dominio para crear un servicio de dominio, el tipo de servicio de dominio que se crea está basado en las entidades que se exponen.

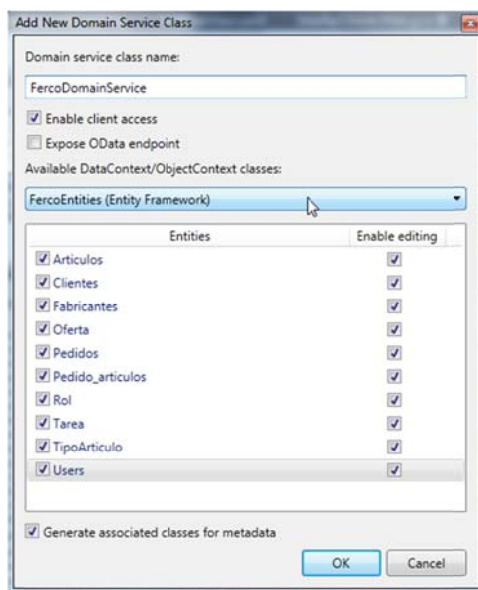


Figura 43: Creación de un servicio de dominio a partir de Entity Framework

Una clase de servicio de dominio se debe marcar con el atributo `EnableClientAccessAttribute` para que el servicio esté disponible para el proyecto de cliente. El atributo `EnableClientAccessAttribute` se aplica automáticamente al servicio de dominio cuando se activa la casilla Habilitar acceso de cliente del cuadro de diálogo

Cuando se aplica el atributo `EnableClientAccessAttribute` a un servicio de dominio, RIA Services genera las clases correspondientes para el proyecto de cliente. Por ejemplo, cuando se aplica el atributo `EnableClientAccessAttribute` a un servicio de dominio denominado `HRService` que expone una entidad denominada `Employee`, RIA Services genera un contexto de dominio denominado `HRContext` en el proyecto de cliente y una versión de cliente de la entidad `Employee`.

Nuestra clase `DomainService` se denomina `FercoDomainService`.

```
namespace InterfazCliente.Web
{
    using System;
    using System.Collections.Generic;
    using System.ComponentModel;
    using System.ComponentModel.DataAnnotations;
    using System.Data;
    using System.Linq;
    using System.ServiceModel.DomainServices.EntityFramework;
    using System.ServiceModel.DomainServices.Hosting;
    using System.ServiceModel.DomainServices.Server;

    // Implements application logic using the FercoEntities context.
    // TODO: Add your application logic to these methods or in additional methods.
    // TODO: Wire up authentication (Windows/ASP.NET Forms) and uncomment the following to disable anonymous access
    // Also consider adding roles to restrict access as appropriate.
    // [RequiresAuthentication]
    [EnableClientAccess()]
    public class FercoDomainService : LinqToEntitiesDomainService<FercoEntities>
    {
        // TODO:
        // Consider constraining the results of your query method. If you need additional input you can
        // add parameters to this method or create additional query methods with different names.
        // To support paging you will need to add ordering to the 'Articulos' query.
        [Query(IsDefault = true)]
        public IQueryable<Articulos> GetArticulos()
        {
            return this.ObjectContext.Articulos;
        }
    }
}
```

Figura 44: Clase `FercoDomainService`

WCF y servicios de dominio

Como servicio de Windows Communication Foundation (WCF), el servicio de dominio se basa en los conceptos de WCF. El servicio de dominio conserva las funciones siguientes:

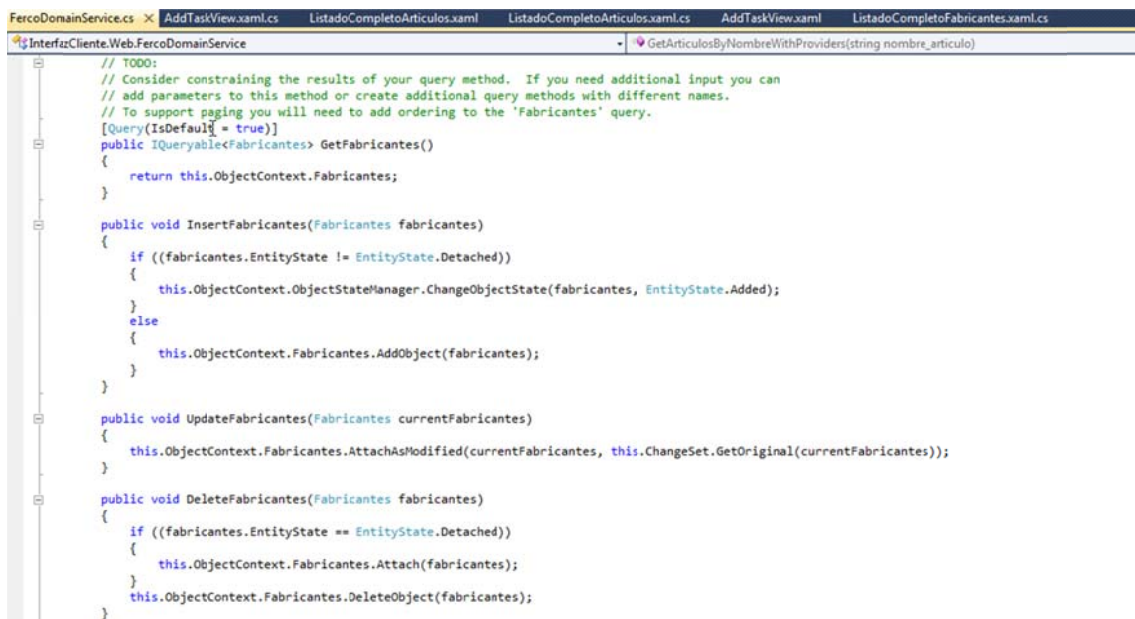
- Uso estándar de servicios de WCF
- Existencia de estructuras de modelos de programación de WCF, como contratos de operación, comportamientos de operación y comportamientos de servicio
- Capacidades de personalización de WCF estándar, como configuración de enlace, configuración de comportamiento e infraestructura de administración

El contexto de dominio se comunica con el servicio de dominio de RIA Services utilizando el elemento ChannelFactory de WCF para crear un canal y pasarle un contrato de servicio generado desde el servicio de dominio.

Operaciones de datos

Puede agregar métodos a un servicio de dominio que realicen la operación de datos que desee exponer. Por ejemplo, puede agregar métodos que realicen las siguientes operaciones:

- Query
- Update
- Insert
- Delete



```

FercDomainService.cs | AddTaskView.xaml.cs | ListadoCompletoArticulos.xaml | ListadoCompletoArticulos.xaml.cs | AddTaskView.xaml | ListadoCompletoFabricantes.xaml.cs
InterfazCliente.Web.FercDomainService | GetArticulosByNombreWithProviders(string nombre_articulo)
// TODO:
// Consider constraining the results of your query method. If you need additional input you can
// add parameters to this method or create additional query methods with different names.
// To support paging you will need to add ordering to the 'Fabricantes' query.
[Query(IsDefault = true)]
public IQueryable<Fabricantes> GetFabricantes()
{
    return this.ObjectContext.Fabricantes;
}

public void InsertFabricantes(Fabricantes fabricantes)
{
    if ((fabricantes.EntityState != EntityState.Detached))
    {
        this.ObjectContext.ObjectStateManager.ChangeObjectState(fabricantes, EntityState.Added);
    }
    else
    {
        this.ObjectContext.Fabricantes.AddObject(fabricantes);
    }
}

public void UpdateFabricantes(Fabricantes currentFabricantes)
{
    this.ObjectContext.Fabricantes.AttachAsModified(currentFabricantes, this.ChangeSet.GetOriginal(currentFabricantes));
}

public void DeleteFabricantes(Fabricantes fabricantes)
{
    if ((fabricantes.EntityState == EntityState.Detached))
    {
        this.ObjectContext.Fabricantes.Attach(fabricantes);
    }
    this.ObjectContext.Fabricantes.DeleteObject(fabricantes);
}

```

Figura 45: Operaciones Query,Insert,Update,Delete sobre la entidad Fabricantes. Dentro de FercDomainService

Para tener acceso a los datos se utiliza LINQ. Lo veremos más adelante.

Cuando se expone un servicio de dominio, se genera un objeto EntitySet en el contexto de dominio con propiedades que indican qué operaciones (inserción, actualización o

eliminación) se permiten en el cliente. Las modificaciones de datos se ejecutan modificando la colección de entidades y llamando después al método SubmitChanges.

En casi todos los escenarios, se deben usar operaciones de consulta en lugar de operaciones de invocación para la carga de datos. Los métodos de consulta devuelven un solo objeto Entity, un objeto IQueryable<Entity> o un objeto IEnumerable<Entity>. Los métodos de consulta forman parte integral del patrón de datos admitido por DomainService en el nivel intermedio y por DomainContext en el cliente. El marco de RIA Services genera entidades en el proyecto de cliente solo para las entidades devueltas por los métodos de consulta de un elemento DomainService.

LINQ

Language Integrated Query (LINQ) es un proyecto de Microsoft que agrega consultas nativas semejantes a las de SQL a los lenguajes de la plataforma .NET, inicialmente a los lenguajes Visual Basic .NET y C#. Muchos conceptos que LINQ ha introducido fueron originalmente probados en Cw, un proyecto de investigación de Microsoft.

LINQ define operadores de consulta estándar que permiten a lenguajes habilitados con LINQ filtrar, enumerar y crear proyecciones de varios tipos de colecciones usando la misma sintaxis. Tales colecciones pueden incluir vectores (arrays), clases enumerables, XML, conjuntos de datos desde bases de datos relacionales y orígenes de datos de terceros. El proyecto LINQ usa características de la versión 2.0 del .NET Framework, nuevos ensamblados relacionados con LINQ, y extensiones para los lenguajes C# y Visual Basic .NET. Microsoft ha distribuido una versión previa del LINQ, consistente de estas bibliotecas y compiladores para C# 3.0 y Visual Basic 9. Otros lenguajes, como F# y Nemerle, han anunciado brindar soporte preliminar.

El objetivo de crear LINQ es permitir que todo el código hecho en Visual Studio (incluidas las llamadas a bases de datos, datasets, XMLs) sean también orientados a objetos. Antes de LINQ, la manipulación de datos externos tenía un concepto más estructurado que orientado a objetos. Además LINQ trata de facilitar y estandarizar el acceso a dichos objetos.

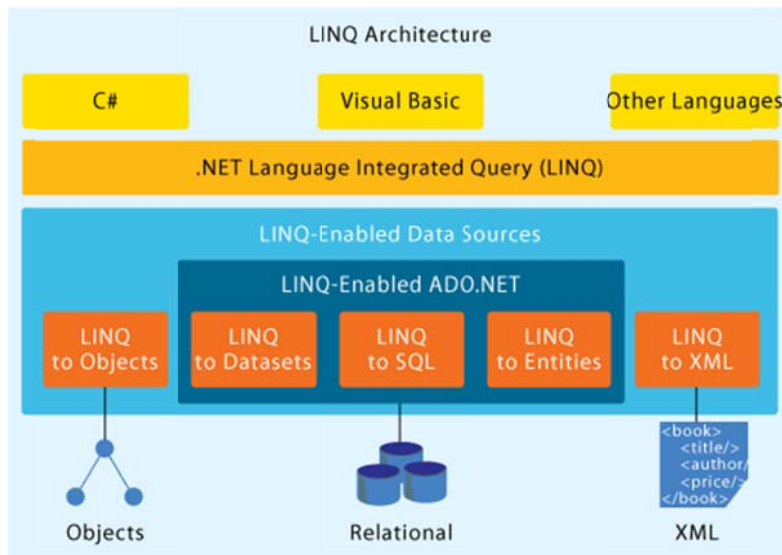


Figura 46:Arquitectura de LINQ

Nuestra forma de trabajar será con LINQ To Entities.

LINQ to Entities proporciona la capacidad de realizar consultas integradas en lenguajes (LINQ) que permite a los desarrolladores de software escribir consultas contra el modelo conceptual de Entity Framework mediante Visual Basic o Visual C#. Las consultas con Entity Framework se representan mediante consultas de árboles de comandos, que se ejecutan en el contexto del objeto. LINQ to Entities convierte las consultas de Language-Integrated Queries (LINQ) en consultas de árboles de comandos, ejecuta las consultas en Entity Framework y devuelve objetos que se pueden usar tanto en Entity Framework como en LINQ. A continuación se muestra el proceso para crear y ejecutar una consulta de LINQ to Entities.

1. Cree una instancia de ObjectQuery enObjectContext.
2. Cree una consulta de LINQ to Entities en C# o Visual Basic con la instancia de ObjectQuery.
3. Convierta los operadores y expresiones de consulta estándar de LINQ en árboles de comandos.

4. Ejecute la consulta, con representación de un árbol de comandos, en el origen de datos. Las excepciones producidas en el origen de datos durante la ejecución se pasan directamente al cliente.
5. Devuelva los resultados de la consulta al cliente.

A continuación veremos un ejemplo.

Consulta que nos devuelve aquellos artículos cuyo nombre contenga a la variable pasada como nombre_artículo. Además nos devolverá la información de los fabricantes de dicho artículo.

```
public IQueryable<Articulos> GetArticulosByNombreWithProviders(string
nombre_articulo)
{
    var query =
        from c in this.ObjectContext.Articulos.Include("Fabricantes")
        where c.Nombre.Contains(nombre_articulo)
        select c;

    return query;
}
```

DomainContext

Una clase DomainContext es una representación del lado del cliente de un servicio de dominio, que proporciona acceso a toda la funcionalidad del servicio.

Para cada servicio de dominio en el proyecto de servidor, WCF RIA Services genera una clase que se deriva de DomainContext. La clase DomainContext generada se utiliza para interactuar con el servicio de dominio. Los datos se recuperan si llama al método Load y pasa uno de los métodos de consulta generados como un parámetro. Los cambios se guardan en los datos si llama al método SubmitChanges. Todos los cambios de datos pendientes se cancelan y se revierten los datos a su estado anterior si llama al método RejectChanges. La clase generada contiene métodos de consulta que se corresponden con métodos de consulta en el servicio de dominio. De forma predeterminada, RIA Services usa una convención de nomenclatura para el contexto de dominios generado y sus métodos. Por ejemplo, un servicio de dominio en el proyecto de servidor denominado FercoDomainService tendrá una clase de contexto de dominios

en el proyecto de cliente denominada FercoDomainContext. Un método de consulta denominado GetFabricantes tiene un método correspondiente en el proyecto de cliente denominado GetCustomersFabricantes.

Por último, es importante destacar que cuando se ejecuta una operación de dominio, la operación se procesa de forma asincrónica. Por ello se utiliza el método Completed.

Código Xaml lo anterior queda implementado de la siguiente forma:

```
<riaControls:DomainDataSource AutoLoad="True" d:DesignData="{d:DesignInstance
my:Fabricantes, CreateList=true}" Height="0"
LoadedData="fabricantesDomainDataSource_LoadedData"
Name="fabricantesDomainDataSource" QueryName="GetFabricantesQuery" Width="0">
    <riaControls:DomainDataSource.DomainContext>
        <my:FercoDomainContext />
    </riaControls:DomainDataSource.DomainContext>
</riaControls:DomainDataSource>
```

Para mostrar los datos en un datagrid, por ejemplo:

```
<sdk:DataGrid AutoGenerateColumns="False" Height="435" HorizontalAlignment="Left"
ItemsSource="{Binding ElementName=fabricantesDomainDataSource, Path=Data}"
Margin="30,48,0,0" Name="fabricantesDataGrid"
RowDetailsVisibilityMode="VisibleWhenSelected" VerticalAlignment="Top"
Width="226" SelectionChanged="fabricantesDataGrid_SelectionChanged"
HeadersVisibility="All" GridLinesVisibility="Horizontal" IsReadOnly="True"
FontSize="14" FontWeight="Black">
    <sdk:DataGrid.Columns>
        <sdk:DataGridTextColumn x:Name="nombre_FabricanteColumn"
Binding="{Binding Path=Nombre_Fabricante}" Header="Nombre Fabricante"
Width="SizeToHeader" FontSize="12"/>
    </sdk:DataGrid.Columns>
</sdk:DataGrid>
```

Si queremos hacerlo en code-behind C#:

```
public static class AppState
{
    public static FercoDomainContext ctxFerco;
}

AppState.ctxFerco = new FercoDomainContext();

EntityQuery<Fabricantes> query = AppState.ctxFerco.GetFabricantesQuery();
LoadOperation<Fabricantes> loadOp =
AppState.ctxFerco.Load<Fabricantes>(query);
loadOp.Completed += (sen, eventArgs) =>
{
    fabricantesDataGrid.ItemsSource = loadOp.Entities.ToList();
};
```

AUTENTICACIÓN Y ROLES

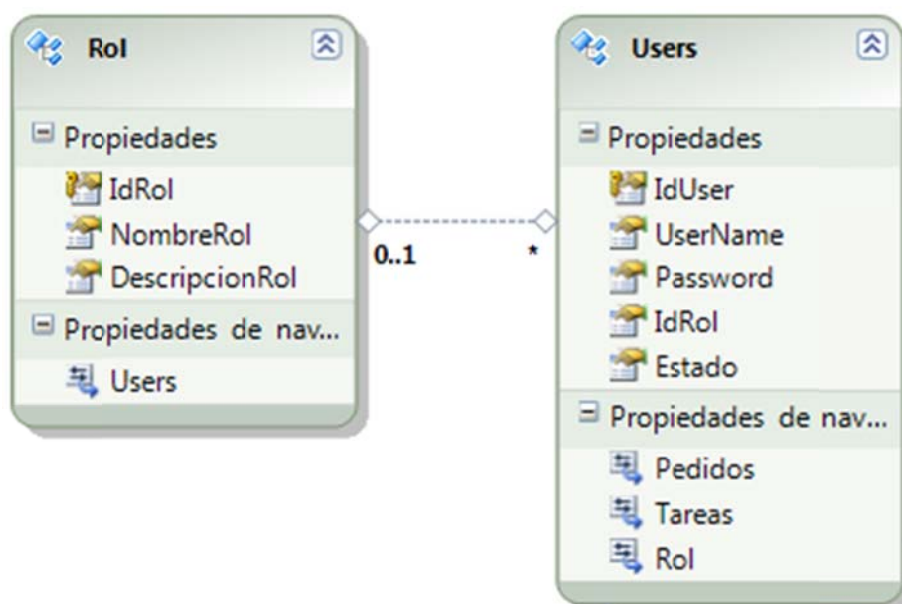
La autenticación usando WCF Ria Services se puede implementar de dos maneras:

- Uso de autenticación por defecto, proporcionada por la plantilla de Negocios de Silverlight (uso de la tabla de ASP.NET “ASPNETDB”).
- Uso de autenticación personalizada (utilizando nuestro propio modelo y base de datos).

En nuestra aplicación utilizaremos una autenticación personalizada, ya que de esta forma mantenemos una cohesión y simplicidad de nuestra base de datos.

(La tabla users está relacionada con tareas y pedidos, si tuviera las tablas en distintas bdd esta simplicidad se perdería).

La tabla Users de nuestra base de datos ,va a almacenar la información básica de un usuario. Además la tabla Rol tendrá los diferentes Roles que permite nuestra aplicación, “Administrator” o “Employee”.



Se ha agregado un servicio de dominio de autenticación a nuestra solución de WCF RIA Services para comprobar las credenciales de un usuario, restringir el acceso para ciertas operaciones y conservar las propiedades para cada usuario desde su proyecto de cliente. RIA Services se basa en el marco de pertenencia a ASP.NET exponiendo el marco de pertenencia a los clientes de Internet enriquecidos a través del servicio de dominio de autenticación. Después de agregar un servicio de dominio de autenticación, puede habilitar las siguientes funciones:

- Autenticación: para comprobar las credenciales de un usuario y marcarlo como que ha iniciado sesión o ha cerrado sesión.
- Roles: para agrupar a los usuarios por responsabilidades y conceder permisos de recurso a los miembros autenticados de un grupo.
- Perfiles: para conservar las propiedades para los usuarios autenticados y recuperar esas propiedades en la aplicación.

A continuación se explica mas a fondo cómo utilizar autenticación, roles y perfiles en una solución de RIA Services.

Servicio de dominio de autenticación.

Creo un membershipProvider personalizado, que me permitirá obtener los usuarios de mi base de datos que estén dados de alta (“Activos”).

```
public class CustomMembershipProvider : MembershipProvider
{
    public override bool ValidateUser(string username, string password)
    {
        using (FercoEntities context = new FercoEntities())
        {
            var user = context.Users.Where(u => u.UserName == username && u.Password == password && u.Estado=="Activado").FirstOrDefault();
            return user != null;
        }
    }

    public override string ApplicationName
    { get { return "FERCO"; } set { } }
}
```

Figura 48: Clase CustomMemberShipProvider

RIA Services proporciona la plantilla DomainService de autenticación para facilitar el acceso a autenticación, roles y perfiles en el nivel de presentación.

Agregamos un servicio de dominio de autenticación, llamado `FercoAuthenticationDomainService` y se agregan automáticamente dos clases al proyecto de servidor. La clase que representa el servicio de autenticación deriva de la clase `AuthenticationBase`. La clase que representa al usuario deriva de la clase `UserBase`. La clase de usuario contiene las propiedades de perfil para un usuario autenticado.

Cuando se genera la solución, RIA Services genera automáticamente una clase `WebContext` en el proyecto de cliente. La clase `WebContext` permite el acceso al servicio de dominio de autenticación y al usuario en el proyecto de cliente. Puede utilizar la propiedad `Current` para recuperar la instancia actual de `WebContext`. La clase `WebContext` deriva de `WebContextBase`.

Autenticación.

RIA Services proporciona clases que permiten implementar fácilmente la autenticación mediante formularios o la autenticación de Windows en la solución.

Después de configurar los proyectos de cliente y de servidor, puede hacer que los usuarios inicien sesión asincrónicamente desde la aplicación de Silverlight llamando al método `Login` en el objeto `WebContext` :

```
WebContext.Current.Authentication.Login(LoginParameters);
```

De la misma forma podemos cerrar sesión asincrónicamente de un usuario autenticado:

```
WebContext.Current.Authentication.Logout(LoginParameters);
```

Podemos obtener acceso al estado del usuario, por ejemplo , el nombre de usuario:

```
WebContext.Current.Authentication.User.Identity.Name
```

Roles.

Después de haber implementado la autenticación, puede configurar la solución para utilizar roles. Con roles, puede asignar usuarios a grupos.

Para restringir el acceso a determinadas páginas utilizamos:

`WebContext.Current.Authentication.User.Identity.IsAuthenticated`

Restringe el acceso a los usuarios que están registrados.



Figura 49: Acceso denegado en la gestión de pedidos

Si queremos ser más restrictivos podemos utilizar:

`WebContext.Current.Authentication.User.IsInRole("Administrator")`

En este caso , únicamente podrán acceder a la página el usuarios que sea tenga un rol de Administrator.

Para especificar que una operación de dominio determinada solo esté disponible para los miembros de ese rol , se puede aplicar los atributos siguientes al servicio de dominio completo o a las operaciones individuales en el servicio. Cuando se aplica un atributo al servicio completo, se aplica a todas las operaciones.

- `RequiresAuthenticationAttribute`: especifica que solo los usuarios con credenciales de autenticación válidas pueden tener acceso a la operación.
- `RequiresRoleAttribute`: especifica que solo los usuarios autenticados que pertenecen a los roles especificados pueden tener acceso a la operación

Por ejemplo, queremos que únicamente el usuario que este autenticado pueda acceder al servicio de dominio de Users:

```
[RequiresAuthentication]
public IQueryable<Users> GetUsers()

{

return this.ObjectContext.Users.Include("Rol");

}
```

7. PRUEBAS

En este apartado se detallarán las distintas pruebas que se han realizado, para identificar los distintos fallos en la implementación de la aplicación. Para ello se han utilizado las llamadas pruebas de validación, en las que un posible usuario final ha revisado el sistema para comprobar que cumple con las especificaciones dadas.

A continuación se detallarán un par de las distintas pruebas que se ha realizado.

Pruebas en la gestión de usuarios.

La gestión de usuarios permite eliminar a determinados usuarios. Un usuario puede tener asignados tareas o pedidos. Si intentamos eliminar un usuario que tiene asignados tareas o pedidos (Ver figura 50) Nos mostrara un mensaje, denegándonos dicha solicitud (Ver Figura 51).

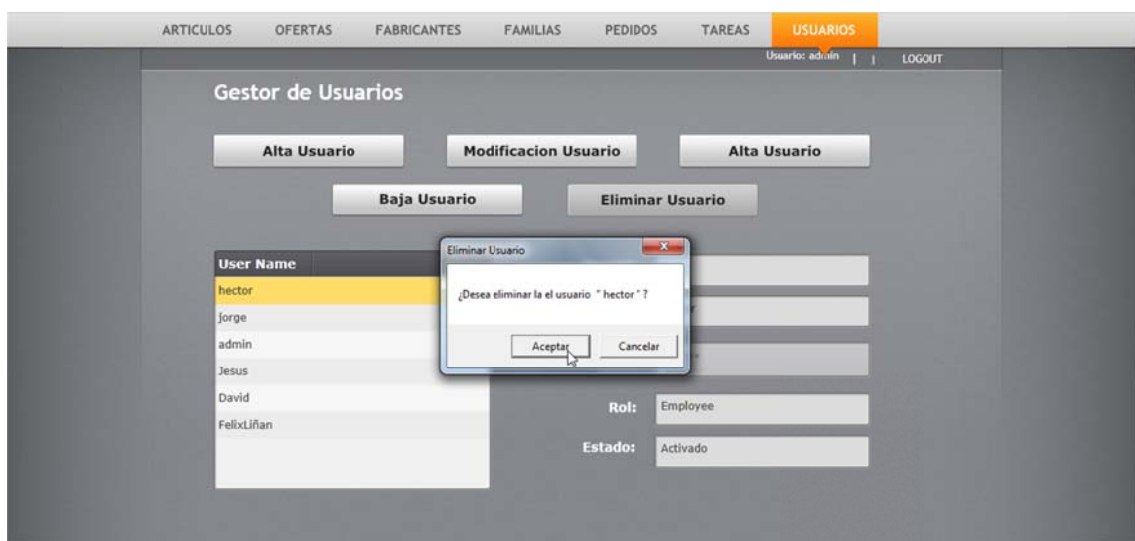


Figura 50: Mensaje confirmación eliminación de usuario

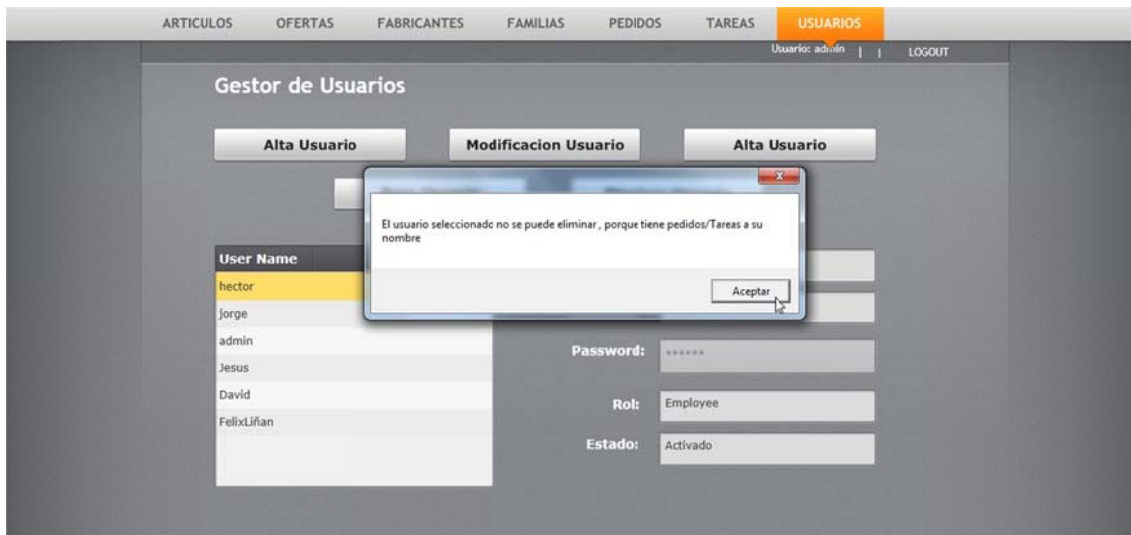


Figura 51: Mensaje denegándonos el borrado de un usuario

Pruebas en la gestión de pedidos.

Un pedido, al igual que un artículo de un pedido. Tiene los siguientes estados (“Pendiente Enviar”, “Enviado”, “Recibido”). Si el estado de un pedido es “Recibido” e intentamos modificarlo al estado “Enviado”, en el sistema saltara un mensaje denegándonos dicha solicitud.



Figura 52 : Mensaje de aviso

8. CONCLUSIONES

En general estoy bastante satisfecho con el resultado de este proyecto. Especialmente si se tiene en cuenta que todas estas tecnologías son nuevas para mí., ya que nunca había programado en C# con Visual Studio, y no conocía Silverlight.

La principal conclusión que he obtenido ha sido que desarrollar una aplicación web, es mucho más complicado que desarrollar una aplicación de escritorio.

Por ejemplo, la separación entre cliente y servidor que propone Silverlight, el tema de la seguridad , la incompatibilidad con ciertos navegadores, etc. .

La utilización de nuevas tecnologías como WCF RIA Services, ha llevado la mayor parte del proyecto, ya que primero hay que entender cómo funciona para poder implementarla, y luego para que funcionase de forma correcta suponía otro gran esfuerzo.

9. LINEAS FUTURAS

Una vez realizado el proyecto, vamos a comentar las posibles líneas de trabajo en el futuro que se podrían hacer en esta aplicación.

Implementar la aplicación para una versión en Windows Phone 7.

Utilización de la quinta versión de Silverlight, la cual está actualmente disponible en la versión beta. Esta versión nos proporciona bastantes novedades, una de las cuales es la posibilidad de utilizar objetos de 3D. Con esta posibilidad se mejoraría considerablemente la parte de la aplicación que se refiere a la representación del almacén y los artículos.

Por último, una posible ampliación del número de funcionalidades de la aplicación.

10. BIBLIOGRAFIA

Microsoft MSDN. “Autenticación, Roles y perfiles.WCF Ria Services”. Disponible en: [http://msdn.microsoft.com/es-es/library/ee707361\(v=VS.91\).aspx](http://msdn.microsoft.com/es-es/library/ee707361(v=VS.91).aspx) .

Microsoft MSDN. “ADO.NET Entity Framework”. Disponible en: [http://msdn.microsoft.com/es-es/library/bb399572\(v=VS.100\).aspx](http://msdn.microsoft.com/es-es/library/bb399572(v=VS.100).aspx)

Microsoft MSDN. “Herramientas de ADO.NET Entity Data Model”. Disponible en: <http://msdn.microsoft.com/es-es/library/bb399249.aspx>

Microsoft MSDN. “Conozca Silverlight”. Disponible en: <http://msdn.microsoft.com/es-es/silverlight/bb187401>

Noyes, Brian. “WCF RIA Services Part 7 - Authentication and Authorization” . Disponible en: <http://www.silverlightshow.net/items/WCF-RIA-Services-Part-7-Authentication-and-Authorization.aspx> .

Viale, Pablo. “Cómo crear las tablas del Membership y Role Provider en SQL Server. Disponible en: <http://www.viale.mx/2008/02/22/como-crear-las-tablas-del-membership-y-role-provider-en-sql-server/>

Patnaik, Manas. “Authentication against custom table in Silverlight 4 using RIA Service”. Disponible en : <http://www.c-sharpcorner.com/UploadFile/3cf852/6256/> .

Latorre, Jose Luis. “Novedades en Silverlight 5”. Disponible en: <http://www.desarrolloweb.com/articulos/novedades-silverlight-5-parte-2-dotnet.html>

Microsoft Showcase, “Creating storyboard animation”. Disponible en: <http://www.microsoft.com/showcase/en/us/details/933faef8-d99b-4ce5-864b-3bfeaf072cce>

Microsoft Silverlight Team, “Animations” . Disponible en: [http://www.silverlight.net/learn/creating-ui/animation-and-easing/animations-\(silverlight-quickstart\)](http://www.silverlight.net/learn/creating-ui/animation-and-easing/animations-(silverlight-quickstart))

