

TFC – J2EE: Desarrollo de una tienda virtual

(Plan de trabajo)

Estudiante: Asier Moreno Pascuas
Titulación: ITIS
Consultor: Oscar Escudero Sanchez
Fecha: 12 de Marzo de 2007

Agradecimientos

A mi familia, por apoyarme siempre en todos los proyectos que he emprendido, fueran descabellados o no.
A Iraide, mi pareja, por estar a mi lado tanto en los buenos como en los malos momentos, y por haber padecido estos cinco largos años de estudios que finalizan con el presente TFC.
A Markel, mi recién nacido hijo.

Índice

1. Introducción

- 1.1 Resumen del proyecto.
- 1.2 Motivación
- 1.3 Descripción del proyecto
- 1.4 Planificación

2. Recogida y documentación de requisitos

- 2.1 Información inicial
- 2.2 Glosario
- 2.3 Modelo del dominio
- 2.4 Diagrama de casos de uso
- 2.5 Documentación textual de los casos de uso
- 2.6 Requisitos de la interfaz de usuario

3. Análisis

- 3.1 Identificación de las clases de entidades y sus atributos
- 3.2 Diagramas de estados
- 3.3 Diagramas de actividades
- 3.4 Modelo de datos

4. Diseño técnico

- 4.1 El concepto de patrón
- 4.2 El patrón Service To Worker
- 4.3 Problemas con las típicas aproximaciones Servlet
- 4.4 La aproximación XSLT + Servlet
- 4.5 Uso del patrón Service To Worker y XSLT

5. Implementación

- 5.1 Herramientas de desarrollo y pruebas
- 5.2 Proceso de instalación

6. Conclusiones

7. Bibliografía

1. Introducción

1.1 Resumen del proyecto

Con la elaboración de este TFC, pretendía adquirir una base sólida de conocimiento en el desarrollo de aplicaciones web mediante el uso de patrones y tecnologías de servidor como J2EE. Buscaba entender el funcionamiento de este tipo de aplicaciones así como desarrollar las capacidades necesarias para diseñarlas y desarrollarlas en el ámbito profesional.

Concretamente, pretendía profundizar en el patron Modelo Vista Controlador, muy utilizado actualmente en el mundo empresarial. En principio pensé en utilizar jsp para desarrollar las vistas que propone el modelo, pero tras investigar las nuevas tendencias en arquitectura web, opté por implementarlas con XSLT. Algo parecido ocurrió con el modelo de datos. En principio pensaba utilizar la base de datos MySQL, pero me decidí por un modelo de datos basado en XML.

Con la realización de este TFC he descubierto la importancia de la separación de los datos y la lógica de la aplicación, y me he introducido en las nuevas tendencias en desarrollo web, como la web semántica.

En cuanto al software, he utilizado el IDE Eclipse 3.2 con el plugin XML Spy para XML y XSLT proporcionado por Altova en versión gratuita para uso no profesional.

En cuanto al servidor, me he decantado por usar Tomcat, que goza de gran popularidad tanto en el mundo empresarial como en la comunidad de desarrolladores.

1.2 Motivación

Por un lado me motivó a desarrollar este TFC la necesidad de aprender la tecnología J2EE para mejorar mi situación en el ámbito laboral ya que deseaba y deseo orientar mi carrera profesional hacia el desarrollo web.

Por otro lado, el proyecto se basaba en una necesidad real. Desarrollar una aplicación que sirva de base al futuro desarrollo de un negocio real en el que estoy trabajando con mi hermana, que es diseñadora de joyería.

1.3 Descripción del proyecto

El proyecto consiste en la creación de una tienda virtual de venta al por mayor de artículos de joyería. Sólo los usuarios que tengan una cuenta de usuario (otorgada por el administrador), podrán acceder al site, ver los productos, hacer pedidos, etc.

De modo que el site tiene dos partes claramente diferenciadas:

- El site dinámico al que sólo tendrán acceso los usuarios registrados, donde podrán consultar el histórico de sus pedidos y comprobar el estado de los nuevos pedidos que hagan (recibido, en proceso, enviado,...). También podrán gestionar los datos de su cuenta. Por supuesto, podrán realizar nuevos pedidos, añadir productos a esos pedidos y pagarlos.
- Una parte de administración, sólo accesible por parte del administrador, por los encargados de preparar los pedidos (manufacturadores) y por los representantes de la empresa

El administrador podrá dar de alta nuevas colecciones, nuevos productos, cambiar precios, organizar cómo aparecen los productos en las páginas, añadir textos descriptivos, seleccionar qué productos son públicos y cuales no, etc. También podrá dar de alta y de baja a los clientes del negocio, y dispondrá de toda la información relativa a los clientes, sus pedidos, facturación, etc. También se encargará de dar de alta y baja a los empleados de la empresa, es decir,

manufacturadores, representantes y administradores.

Aunque existe la figura de un administrador general del site, éste podrá delegar tareas en otros empleados que también serán de tipo administrador, pero con privilegios limitados.

Los manufacturadores recibirán los pedidos via email. Cuando comiencen a preparar un pedido, se lo comunicarán a la aplicación, para que el cliente pueda conocer su estado y para que otro manufacturador no comience a preparar el mismo pedido. Cuando el pedido esté terminado, también se lo comunicarán al sistema, para que el cliente lo pueda saber. También podrán consultar información sobre los clientes.

El administrador se encargará de dar de alta en el sistema las diferentes colecciones de productos que estarán a la venta. También dará de baja las colecciones que estime oportuno.

Cada colección se compone de una serie de productos que introducirá el administrador. En todo momento podrá cambiar la información relativa a los mismos: nombre, fotos, descripción, precios,... y también podrá eliminarlos.

Debe quedar claro que en realidad se trata de dos aplicaciones web. Por un lado tenemos una tienda virtual, y por otro un administrador del site, desde el que se administran los contenidos, usuarios, etc, y además otros aspectos importantes dentro del negocio, como son la preparación de pedidos, control de stocks, etc. Por estos motivos debería ser una intranet.

1.4 Planificación

Puede verse la planificación del proyecto en el siguiente diagrama de Gant:

Nombre de la tarea	Inicio	Final	Duración (días)	marzo 07					abril 07					mayo 07					junio 07			
				26/2	5/3	12/3	19/3	26/3	2/4	9/4	16/4	23/4	30/4	7/5	14/5	21/5	28/5	4/6	11/6	18/6		
Plan de Trabajo	28/02/2007	12/03/2007	12	[Gantt bar from 28/02 to 12/03]																		
PEC1	28/02/2007	12/03/2007	12	[Gantt bar from 28/02 to 12/03]																		
Analisis	12/03/2007	16/04/2007	34	[Gantt bar from 12/03 to 16/04]																		
PEC2	12/03/2007	16/04/2007	34	[Gantt bar from 12/03 to 16/04]																		
Diseño	16/04/2007	21/05/2007	35	[Gantt bar from 16/04 to 21/05]																		
PEC3	16/04/2007	21/05/2007	35	[Gantt bar from 16/04 to 21/05]																		
Implementacion	21/05/2007	18/06/2007	28	[Gantt bar from 21/05 to 18/06]																		
Memoria	21/05/2007	18/06/2007	28	[Gantt bar from 21/05 to 18/06]																		

2. Recogida y documentación de requisitos

2.1 Información inicial

Como ya se ha explicado anteriormente, el proyecto consiste en la creación de una tienda virtual de venta al por mayor de artículos de joyería.

Aunque sea una tienda online, no será publica , sólo los usuarios que tengan una cuenta abierta, podrán hacer uso del site y realizar pedidos.

Como ya se ha explicado, el site consta de dos partes claramente diferenciadas:

- Una parte privada a la que sólo tendrán acceso los usuarios registrados, donde podrán consultar el histórico de sus pedidos y comprobar el estado de los nuevos pedidos que hagan (recibido, en proceso, enviado,...). También podrán gestionar los datos de su cuenta. Por supuesto, podrán realizar nuevos pedidos, añadir productos a esos pedidos y pagarlos.

- Una parte de administración, sólo accesible por parte del administrador, los encargados de preparar los pedidos (manufacturadores) y los representantes.

En el sistema hay 4 tipos de usuarios: administrador, clientes, representantes y manufacturadores. El administrador tiene permisos para efectuar todas las operaciones que permite el sistema. Sus labores consisten en gestionar las cuentas de los usuarios (dar de alta o de baja, modificar datos, ...), gestionar los productos y colecciones que se venden en la tienda (dar de alta y baja productos y colecciones,...) y gestionar las páginas que se visualizan en la tienda virtual, es decir, los textos y fotos que aparecerán en la home, en las páginas relativas a las colecciones, etc.

Podrá crear otros usuarios de tipo administrador, otorgándoles los permisos que considere oportunos, para que también puedan realizar tareas de administración, incluso sería interesante que el propio administrador general, se creara una segunda cuenta de administrador, restringiendo permisos para efectuar tareas críticas como el borrado de usuarios, productos, etc. De esta forma evitaría el borrado de datos accidental.

Hay cuatro tipos de permisos: crear, borrar, modificar y eliminar. También hay cinco tipos de objetos dentro de la aplicación: usuarios, colecciones, productos, pedidos y facturas. De modo que cada usuario tiene asociada una tabla de permisos que representa los valores de los permisos que tiene sobre cada tipo de objeto. Sólo un administrador puede modificar los permisos de un usuario. El administrador también se encarga de crear las colecciones y los productos que se venden en la tienda. Cada producto deberá ir asociado a una colección y también será clasificado por tipo (collares, pulseras, pendientes,...), por estilo (noche, fiesta, boda) y por clasificación (mujeres, hombres, niños, ...), de esta forma los usuarios podrán utilizar criterios de búsqueda avanzada para encontrar los productos que necesitan.

Tanto las colecciones como los productos pueden tener estado activo o inactivo. De esta forma, el administrador puede ir preparando colecciones que no deben estar a la venta aún, y cambiar su estado a activo en el momento de lanzamiento. Si la colección esta activa, puede decidir si un producto concreto está inactivo, por ejemplo, porque está agotado.

Los clientes sólo pueden acceder a la parte privada del sistema cuando tengan una cuenta. Desde el site pueden solicitar la creación de una cuenta de usuario. Si el administrador aprueba sus solicitud, les dará de alta en el sistema.

Existe otra forma de darse de alta. Los representantes poseen unos códigos de activación únicos, que podrán dar a los clientes de su zona. Así, si un usuario introduce un código válido en el proceso de registro, será validado automáticamente.

Los representantes tienen una zona asignada, y se encargan de hacer nuevos clientes en dicha zona. Cuando un cliente tiene su cuenta puede hacer pedidos añadiendo productos. Se puede guardar un pedido durante 24 horas antes de ser tramitado, así el cliente no tiene que hacerlo de una vez. Cuando termina de hacer el pedido, lo paga, y si todo va bien, recibirá su factura y el pedido quedará registrado en el sistema.

Desde su cuenta, el cliente podrá modificar sus datos personales, consultar sus pedidos y sus facturas.

Los manufacturadores son los encargados de preparar los pedidos y enviárselos a los clientes.

Cuando un manufacturador quiere preparar un pedido, consulta la lista de pedidos que se han recibido. Los pedidos tienen distinta prioridad en función de la fecha de entrada (baja, media, alta, crítica). Cuando decide preparar un pedido, lo selecciona, de forma que queda bloqueado para que ningún otro manufacturador pueda comenzar a prepararlo. Tanto el administrador como los manufacturadores pueden consultar el estado de los pedidos recibidos (recibido, en curso, pausado, pendiente de envío,...). Cuando el manufacturador ha entregado un pedido a la empresa transportista, lo avisa al sistema para que conste como enviado.

Cuando el cliente recibe su pedido, tiene siete días para decidir si está conforme o si quiere devolverlo. También puede ocurrir que algún producto no sea lo que esperaba o que esté defectuoso,

etc. En ese caso, podrá devolverlo sin coste adicional y se le abonará la diferencia o se le cambiará el producto.

2.2 Glosario

Administrador (Administrator): Es el usuario del sistema que tiene plenos privilegios sobre el resto de usuarios y sobre todos los objetos que lo componen. Su misión es gestionar las cuentas de los usuarios y hacer las modificaciones que considere oportunas. También se encarga de gestionar las colecciones y los productos.

Manufacturador (Manufacturer): Se encarga de preparar los pedidos que hacen los clientes y de enviárselos.

Representante (Salesman): Su misión es hacer nuevos clientes. Tiene asignada una o varias zonas, en las que desarrollará su trabajo. Cuando haga un nuevo cliente le proporcionará un código de activación para que pueda registrarse en el sistema sin necesidad de solicitarlo. El uso de este código servirá para contabilizar los clientes que hace mensualmente cada representante y así poder calcular sus comisiones.

Cliente (Client): Es el comprador de los productos que se venden en la tienda. Sólo podrá darse de alta en el sistema si tiene un código de activación, bien proporcionado por el administrador (previa solicitud) o por un representante. Efectuará las compras haciendo pedidos que deberá abonar por adelantado. Podrá conocer el estado de sus pedidos en todo momento. También podrá consultar los pedidos que haya hecho así como sus facturas correspondientes.

Pedido (Order): Es el conjunto de productos que compra un cliente. Los fabricantes se encargan de prepararlos y enviarlos.

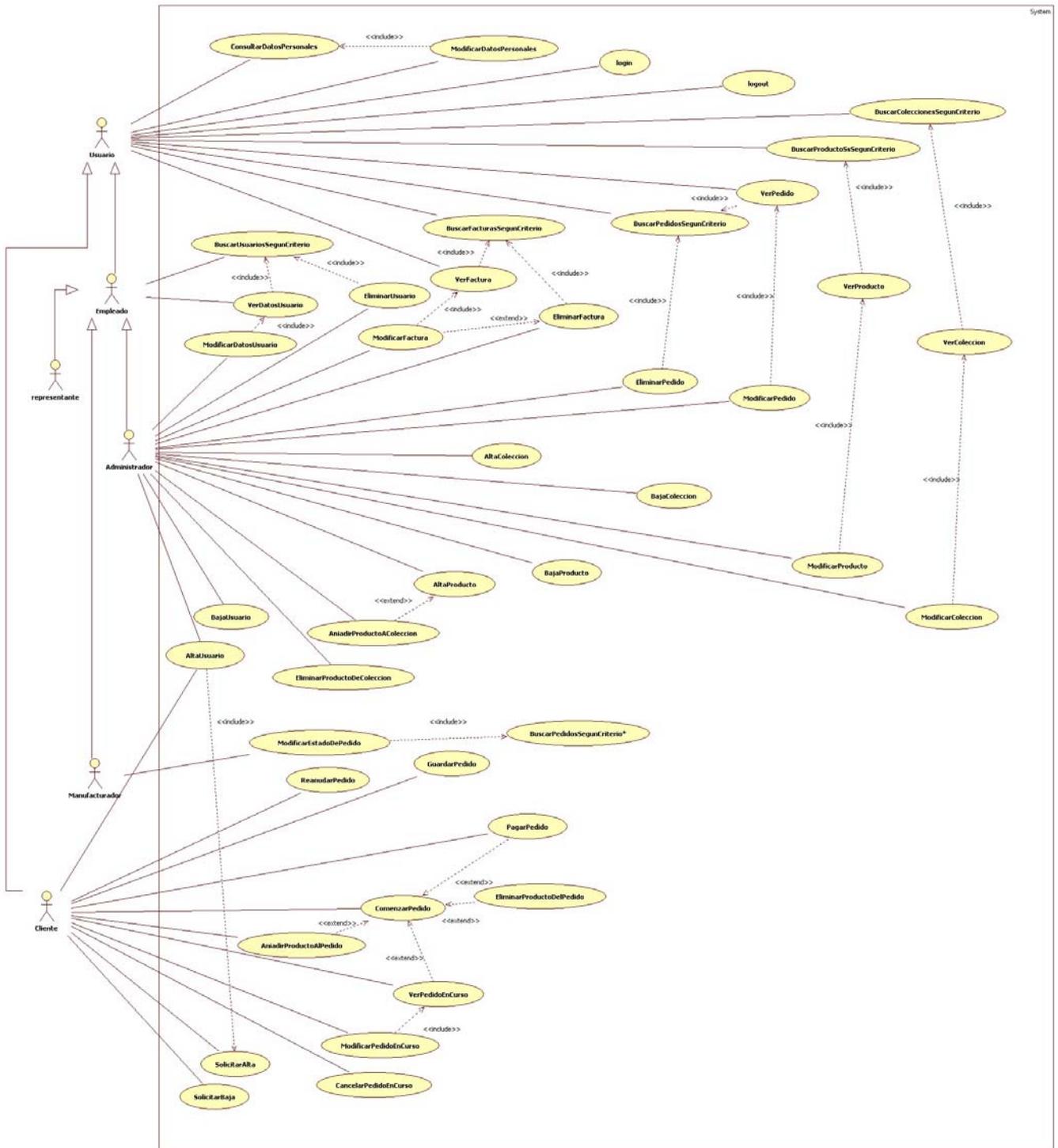
Factura (Invoice): Facturas generadas al pagar los pedidos.

Colección (Collection): Es la forma en que se organizan los productos en el sistema. Las crea el administrador, y decide qué productos las componen.

Producto (Product): Son los artículos que se venden en la tienda. Deben pertenecer a una única colección.

2.3 Modelo del dominio

A partir de la información recopilada, pueden identificarse las clases u objetos: administrador, cliente, representante, fabricante, pedido, factura, colección, zona... Las clases identificadas, con algunos de sus atributos, se recogen en el siguiente diagrama de clases:



2.5 Documentación textual de los casos de uso

Caso de uso número 1: “login”

Resumen de la funcionalidad: identifica a un usuario en el sistema, comenzando una nueva sesión.

Papel en el trabajo del usuario: todos los usuarios deben identificarse para acceder a la parte privada del site o a la parte de administración.

Actores: Administrador, cliente, fabricante, representante.

Casos de uso relacionados: "logout"

Precondición: El usuario está registrado en el sistema (tiene una cuenta) y no tiene una sesión abierta.

Poscondición: El usuario está identificado en el sistema y tiene una sesión abierta.

Descripción: El usuario introduce su nick y password en un formulario y si se encuentra en la base de datos, se le identifica en el sistema y se abre una sesión.

Alternativas de proceso y excepciones: Si se introduce un nick y/o password erróneo, se le comunica al usuario, y se le avisa de que si no está registrado no puede acceder a la parte privada del site.

Otras cuestiones: Habrá un login en el site, es decir en la parte que pueden ver los clientes, y otro en la parte de administración, a la que sólo pueden acceder el administrador y el fabricante, previa identificación.

Caso de uso número 2: "logout"

Resumen de la funcionalidad: cierra la sesión de un usuario.

Papel en el trabajo del usuario: los usuarios deben cerrar su sesión para evitar posibles usos indebidos de sus cuentas de usuario.

Actores: Administrador, cliente, fabricante, representante.

Casos de uso relacionados: "login"

Precondición: El usuario se ha identificado y tiene una sesión abierta.

Poscondición: La sesión se ha cerrado. Para acceder a la parte privada del site, el usuario debe volver a identificarse.

Descripción: El usuario pulsa el botón "logout" y el sistema se encarga de cerrar la sesión y de redirigirle a la parte pública del site.

Alternativas de proceso y excepciones: La sesión se cerrará de forma automática tras un cierto periodo de inactividad o cuando se cierre el navegador del usuario.

Otras cuestiones: Habrá un logout para el área de clientes y otro para la parte de administración.

Caso de uso número 3: "consultar datos personales"

Resumen de la funcionalidad: Da información de los datos de un usuario registrado.

Papel en el trabajo del usuario: Todos los usuarios pueden consultar sus datos personales

Actores: Administrador, fabricante, cliente, representante.

Casos de uso relacionados: "modificar datos personales".

Precondición: Existe la cuenta del usuario.

Poscondición: Ninguna.

Descripción: Un usuario consulta sus datos personales.

Alternativas de proceso y excepciones: Ninguna.

Otras cuestiones: Ninguna.

Caso de uso número 4: "modificar datos personales"

Resumen de la funcionalidad: permite a un usuario registrado modificar sus datos personales.

Papel en el trabajo del usuario: los usuarios registrados pueden modificar sus datos personales cuando lo deseen.

Actores: Administrador, cliente, fabricante, representante.

Casos de uso relacionados: "enviar mensaje"

Precondición: El usuario está registrado.

Poscondición: El usuario ha modificado sus datos personales.

Descripción: Los usuarios pueden cambiar sus datos personales como el número de teléfono, dirección, nick, password,.. . Cuando se produzca el cambio, se dará el caso de uso “envío de mensaje” en el que el sistema enviará un email al usuario notificándole que se ha producido el cambio con éxito.

Alternativas de proceso y excepciones: Si se produce un error en el proceso de cambio, se le comunica al usuario para que vuelva a intentarlo en otro momento.

Otras cuestiones: los datos introducidos serán validados para verificar su corrección.

Caso de uso número 5: “buscar pedidos según criterio”

Resumen de la funcionalidad: Permite mostrar un listado de pedidos según un criterio determinado, por ejemplo, por fecha, o por cliente,....

Papel en el trabajo del usuario: Tanto el administrador como los fabricantes tienen acceso a todos los pedidos y pueden consultarlos. Lo más importante de este caso de uso es que permite a los usuarios visualizar los pedidos según los criterios que más les convengan.

El cliente podrá consultar sus pedidos por fechas.

Actores: Administrador, cliente, fabricante, representante.

Casos de uso relacionados: “consultar pedido”, “modificar pedido”.

Precondición: Existen pedidos.

Poscondición: Se muestra un listado de pedidos.

Descripción: Se obtendrán listados de pedidos según diferentes criterios: por fecha, por usuario y por estado (nuevo, en proceso, preparado, enviado, recibido). Cuando el usuario obtenga el listado que ha solicitado, podrá consultar los pedidos obtenidos, dándose así el caso de uso “ver pedido”. En el caso de los clientes, sólo podrán acceder a sus propios pedidos, y el listado será únicamente por fechas.

Alternativas de proceso y excepciones: Aviso en caso de que no se hayan encontrado pedidos según el criterio seleccionado.

Otras cuestiones: Ninguna.

Caso de uso número 6: “ver pedido”

Resumen de la funcionalidad: Da información de un pedido de un cliente.

Papel en el trabajo del usuario: El administrador tiene acceso a todo el sistema, así que también a los pedidos de los usuarios. Los fabricantes necesitan consultar los pedidos para prepararlos, y los clientes deben poder consultar sus propios pedidos. El administrador podrá asignar un pedido a cualquier fabricante. Un fabricante sólo podrá asignarse un pedido así mismo.

Actores: Administrador, cliente, fabricante, representante.

Casos de uso relacionados: “consulta global de pedidos”, “modificar pedido”.

Precondición: Existe el pedido.

Poscondición: Ninguna.

Descripción: Un usuario consulta los datos de un pedido de otro usuario (en el caso del administrador y de los fabricantes) o de uno de sus propios pedidos, y obtiene información detallada del mismo.

Alternativas de proceso y excepciones: Aviso en caso de que el pedido no exista.

Otras cuestiones: Ninguna.

Caso de uso número 7: “buscar facturas según criterio”

Resumen de la funcionalidad: Permite mostrar un listado de facturas según un criterio determinado, por ejemplo, por fecha, o por cliente,....

Papel en el trabajo del usuario: Tanto el administrador como los fabricantes tienen acceso a todas las facturas y pueden consultarlas. Lo más importante de este caso de uso es que permite a los usuarios visualizar las facturas según los criterios que más les convengan.

El cliente podrá consultar sus facturas por fechas.

Actores: Administrador, cliente, fabricante, representante.

Casos de uso relacionados: “consultar factura”, “modificar factura”.

Precondición: Existen facturas.

Poscondición: Se muestra un listado de facturas.

Descripción: Se obtendrán listados de facturas según diferentes criterios: por fecha, por usuario...

Cuando el usuario obtenga el listado que ha solicitado, podrá consultar las facturas obtenidas, dándose así el caso de uso “ver factura”.

En el caso de los clientes, sólo podrán acceder a sus propias facturas, y el listado será únicamente por fechas.

Alternativas de proceso y excepciones: Aviso en caso de que no se hayan encontrado facturas según el criterio seleccionado

Otras cuestiones: Ninguna.

Caso de uso número 8: “ver factura”

Resumen de la funcionalidad: Da información de una factura de un cliente.

Papel en el trabajo del usuario: El administrador, los fabricantes y los representantes tienen acceso a las facturas de los clientes. Los clientes sólo pueden ver sus propias facturas.

Actores: Administrador, cliente, fabricante, representante.

Casos de uso relacionados: “consulta global de facturas”, “modificar factura”.

Precondición: Existe la factura.

Poscondición: Ninguna.

Descripción: Un usuario consulta los datos de una factura y obtiene información detallada de la misma.

Alternativas de proceso y excepciones: Aviso en caso de que la factura no exista.

Otras cuestiones: Ninguna.

Caso de uso número 9: “buscar colecciones según criterio”

Resumen de la funcionalidad: Permite mostrar un listado de colecciones según un criterio determinado.

Papel en el trabajo del usuario: Todos los usuarios pueden consultar las colecciones.

Actores: Administrador, fabricante, cliente, representante.

Casos de uso relacionados: “consultar colección”, “modificar colección”.

Precondición: Existen colecciones.

Poscondición: Se muestra un listado de colecciones.

Descripción: Se obtendrán listados de colecciones según diferentes criterios: por temporada, por nombre...

Cuando el usuario obtenga el listado que ha solicitado, podrá consultar las colecciones obtenidas, dándose así el caso de uso “ver colección”.

Alternativas de proceso y excepciones: Aviso en caso de que no se haya encontrado ninguna colección según el criterio indicado.

Otras cuestiones: Ninguna.

Caso de uso número 10: “ver colección”

Resumen de la funcionalidad: Da información de una colección.

Papel en el trabajo del usuario: Todos los usuarios pueden ver la información de una colección

Actores: Administrador, cliente, fabricante, representante.

Casos de uso relacionados: “consulta global de colecciones”, “modificar colección”.

Precondición: Existe la colección.

Poscondición: Ninguna.

Descripción: Un usuario consulta los datos de una colección y obtiene información detallada de la misma.

Alternativas de proceso y excepciones: Aviso en caso de que la colección no exista.

Otras cuestiones: Ninguna.

Caso de uso número 11: “buscar productos según criterio”

Resumen de la funcionalidad: Permite mostrar un listado de productos según un criterio determinado, por ejemplo, colección, material, estilo,...

Papel en el trabajo del usuario: Todos los usuarios pueden obtener listados de los productos.

Actores: Administrador, fabricante, cliente, representante.

Casos de uso relacionados: “consultar producto”, “modificar producto”.

Precondición: Existen productos.

Poscondición: Se muestra un listado de productos.

Descripción: Se obtendrán listados de productos según diferentes criterios: por colección, por precio, por estilo... Cuando el usuario obtenga el listado que ha solicitado, podrá consultar los productos obtenidos, dándose así el caso de uso “ver producto”.

Alternativas de proceso y excepciones: Aviso en caso de que no haya productos según el criterio seleccionado

Otras cuestiones: Ninguna.

Caso de uso número 12: “ver producto”

Resumen de la funcionalidad: Da información de un producto concreto.

Papel en el trabajo del usuario: Todos los usuarios pueden consultar los datos de un producto.

Actores: Administrador, fabricante, cliente, representante.

Casos de uso relacionados: “consulta global de productos”, “modificar producto”.

Precondición: Existe el producto.

Poscondición: Ninguna.

Descripción: Un usuario consulta los datos de un producto y obtiene información detallada del mismo.

Alternativas de proceso y excepciones: Aviso en caso de que el producto no exista.

Otras cuestiones: Ninguna.

Caso de uso número 13: “buscar usuarios según criterio”

Resumen de la funcionalidad: Permite mostrar un listado de usuarios según un criterio determinado, por ejemplo, nombre, zona...

Papel en el trabajo del usuario: Los empleados, es decir, el administrador, los fabricantes y los representantes pueden hacer una búsqueda de usuarios del sistema.

Actores: Administrador, fabricante, representante.

Casos de uso relacionados: “ver datos usuario”.

Precondición: Existen usuarios.

Poscondición: Se muestra un listado de usuarios.

Descripción: Se obtendrán listados de usuarios según el criterio seleccionado. Cuando se seleccione un usuario del listado, se dará el caso de uso “ver datos de usuario”.

Alternativas de proceso y excepciones: Aviso en caso de que no haya productos según el criterio seleccionado

Otras cuestiones: Ninguna.

Caso de uso número 14: “ver datos usuario”

Resumen de la funcionalidad: Da información de un usuario concreto.

Papel en el trabajo del usuario: Todos los empleados pueden ver los datos de otros usuarios del sistema.

Actores: Administrador, manufacturador, representante.

Casos de uso relacionados: “buscar usuarios según criterio”, “modificar datos de usuario”, “eliminar usuario”

Precondición: Existe el usuario.

Poscondición: ninguna.

Descripción: Un empleado consulta los datos de un usuario y obtiene información detallada del mismo. En el caso del administrador, podrá modificar los datos (caso de uso “modificar datos de usuario”) y eliminarlo (caso de uso “eliminar usuario”).

Alternativas de proceso y excepciones: Aviso en caso de que el usuario no exista.

Otras cuestiones: Ninguna.

Caso de uso número 15: “modificar datos de usuario”

Resumen de la funcionalidad: Modifica los datos de un usuario registrado.

Papel en el trabajo del usuario: Permite la modificación de los datos de un usuario por parte del administrador.

Actores: Administrador.

Casos de uso relacionados: “buscar usuarios según criterio”, “ver datos usuario”, “enviar mensaje”.

Precondición: Existe la cuenta del usuario.

Poscondición: Ninguna.

Descripción: El administrador puede modificar todos los datos de todos los usuarios del sistema. Cuando los datos de un usuario hayan sido modificados, se dará el caso de uso “enviar mensaje”, en el que se enviará un mensaje al usuario informándole de los cambios que se han producido.

Alternativas de proceso y excepciones: Aviso en caso de error en el proceso de modificación.

Otras cuestiones: Ninguna.

Caso de uso número 16: “eliminar usuario”

Resumen de la funcionalidad: Da de baja a un usuario del sistema.

Papel en el trabajo del usuario: El administrador es el único que puede dar de baja al resto de usuarios del sistema.

Actores: Administrador.

Casos de uso relacionados: “enviar mensaje”.

Precondición: Existe la cuenta del usuario.

Poscondición: La cuenta del usuario ya no existe, aunque sus datos no se borran.

Descripción: Cuando interese dar de baja a un usuario, el administrador lo hará. En realidad, la cuenta no desaparece sino que pasa al estado “desactivado”. De este modo, siguen estando accesibles los datos del usuario, su histórico de pedidos, sus facturas, etc.

Cuando la cuenta haya sido dada de baja, se dará el caso de uso “envío de mensaje” en el que el sistema enviará al usuario un mensaje notificándole que ha sido dado de baja en el sistema.

Alternativas de proceso y excepciones: El sistema deberá avisar al administrador de todos los

errores que se produzcan en el proceso de baja.

Otras cuestiones: Ninguna.

Caso de uso número 17: “modificar factura”

Resumen de la funcionalidad: Permite modificar una factura.

Papel en el trabajo del usuario: El administrador puede cambiar los datos de una factura.

Actores: Administrador.

Casos de uso relacionados: “buscar facturas según criterio”, “ver factura”, “enviar mensaje”.

Precondición: Existe la factura.

Poscondición: La factura se ha modificado.

Descripción: El administrador podrá cambiar los datos de una factura aunque este caso de uso se dará en raras ocasiones, es necesario dejar abierta esta posibilidad para poder solucionar posibles errores.

Alternativas de proceso y excepciones: Aviso en caso de error en la modificación.

Otras cuestiones: Ninguna.

Caso de uso número 18: “eliminar factura”

Resumen de la funcionalidad: Permite eliminar una factura.

Papel en el trabajo del usuario: El administrador puede eliminar una factura.

Actores: Administrador.

Casos de uso relacionados: “buscar facturas según criterio”, “ver factura”, “enviar mensaje”.

Precondición: Existe la factura.

Poscondición: La factura se ha eliminado..

Descripción: El administrador podrá eliminar una factura aunque este caso de uso se dará en raras ocasiones, es necesario dejar abierta esta posibilidad para poder solucionar posibles errores.

Alternativas de proceso y excepciones: Aviso en caso de error en la eliminación.

Otras cuestiones: Ninguna.

Caso de uso número 19: “modificar pedido”

Resumen de la funcionalidad: Permite modificar un pedido.

Papel en el trabajo del usuario: El administrador puede cambiar los datos de un pedido.

Actores: Administrador.

Casos de uso relacionados: “buscar pedidos según criterio”, “ver pedido”, “enviar mensaje”.

Precondición: Existe el pedido

Poscondición: El pedido se ha modificado.

Descripción: El administrador podrá cambiar los datos de un pedido aunque este caso de uso se dará en raras ocasiones, es necesario dejar abierta esta posibilidad para poder solucionar posibles errores.

Alternativas de proceso y excepciones: Aviso en caso de error en la modificación.

Otras cuestiones: Ninguna.

Caso de uso número 20: “eliminar pedido”

Resumen de la funcionalidad: Permite eliminar un pedido.

Papel en el trabajo del usuario: El administrador puede eliminar un pedido.

Actores: Administrador.

Casos de uso relacionados: “buscar pedidos según criterio”, “ver pedido”, “enviar mensaje”.

Precondición: Existe el pedido.

Poscondición: El pedido se ha eliminado..

Descripción: El administrador podrá eliminar un pedido aunque este caso de uso se dará en raras ocasiones, es necesario dejar abierta esta posibilidad para poder solucionar posibles errores.

Alternativas de proceso y excepciones: Aviso en caso de error en la eliminación.

Otras cuestiones: Ninguna.

Caso de uso número 21: “alta colección”

Resumen de la funcionalidad: Da de alta una colección.

Papel en el trabajo del usuario: El administrador puede introducir nuevas colecciones de productos en el sistema.

Actores: Administrador.

Casos de uso relacionados: “baja colección”, “añadir producto a colección”.

Precondición: La colección no existe.

Poscondición: La colección ha sido dada de alta.

Descripción: El administrador da de alta una nueva colección introduciendo los datos necesarios de la misma. Durante el proceso de alta, el administrador podrá añadir productos a la colección.

Alternativas de proceso y excepciones: Avisar en caso de que la colección exista.

Otras cuestiones: Ninguna.

Caso de uso número 22: “baja colección”

Resumen de la funcionalidad: Da de baja una colección.

Papel en el trabajo del usuario: El administrador puede dar de baja una colección existente en el sistema.

Actores: Administrador.

Casos de uso relacionados: “alta colección”.

Precondición: La colección existe.

Poscondición: La colección ha sido dada de baja.

Descripción: El administrador da de baja una colección existente.

Alternativas de proceso y excepciones: Avisar en caso de error en el proceso de eliminación.

Otras cuestiones: Ninguna.

Caso de uso número 23: “alta producto”

Resumen de la funcionalidad: Da de alta un producto.

Papel en el trabajo del usuario: El administrador puede introducir nuevos productos en el sistema.

Actores: Administrador.

Casos de uso relacionados: “baja producto”.

Precondición: El producto no existe.

Poscondición: El producto ha sido dado de alta.

Descripción: El administrador da de alta un nuevo producto introduciendo los datos necesarios del mismo.

Alternativas de proceso y excepciones: Avisar en caso de que el producto exista.

Otras cuestiones: Ninguna.

Caso de uso número 24: “baja producto”

Resumen de la funcionalidad: Da de baja un producto.

Papel en el trabajo del usuario: El administrador puede dar de baja un producto existente en el sistema.

Actores: Administrador.

Casos de uso relacionados: “alta producto”.

Precondición: El producto existe.

Poscondición: El producto ha sido dado de baja.

Descripción: El administrador da de baja un producto existente.

Alternativas de proceso y excepciones: Avisar en caso de error en el proceso de eliminación.

Otras cuestiones: Ninguna.

Caso de uso número 25: “añadir producto a colección”

Resumen de la funcionalidad: Añade un producto a una colección.

Papel en el trabajo del usuario: El administrador puede añadir productos a las colecciones.

Actores: Administrador.

Casos de uso relacionados: “eliminar producto de colección”.

Precondición: El producto no está asociado a la colección. La colección y el producto existen.

Poscondición: El producto ha sido añadido a la colección

Descripción: El administrador añade un producto a una colección.

Alternativas de proceso y excepciones: Avisar en caso de que el producto ya haya sido añadido a la colección..

Otras cuestiones: Ninguna.

Caso de uso número 26: “eliminar producto de colección”

Resumen de la funcionalidad: Elimina un producto de una colección.

Papel en el trabajo del usuario: El administrador puede eliminar productos de las colecciones.

Actores: Administrador.

Casos de uso relacionados: “añadir producto a colección”.

Precondición: El producto está asociado a la colección. La colección y el producto existen.

Poscondición: El producto ha sido eliminado de la colección

Descripción: El administrador elimina un producto de una colección.

Alternativas de proceso y excepciones: Avisar en caso de que se produzca algún error.

Otras cuestiones: Ninguna.

Caso de uso número 27: “modificar colección”

Resumen de la funcionalidad: Modifica una colección.

Papel en el trabajo del usuario: El administrador puede modificar una colección existente en el sistema.

Actores: Administrador.

Casos de uso relacionados: “ver colección”, “alta colección”, “baja colección”.

Precondición: La colección existe.

Poscondición: La colección ha sido modificada.

Descripción: El administrador modifica los datos de una colección existente.

Alternativas de proceso y excepciones: Avisar de errores en la modificación de la colección.

Otras cuestiones: Ninguna.

Caso de uso número 28: “modificar producto”

Resumen de la funcionalidad: Modifica un producto

Papel en el trabajo del usuario: El administrador puede modificar un producto existente en el sistema.

Actores: Administrador.

Casos de uso relacionados: “ver producto”, “alta producto”, “baja producto”.

Precondición: El producto existe.

Poscondición: El producto ha sido modificado.

Descripción: El administrador modifica los datos de un producto existente.

Alternativas de proceso y excepciones: Avisar de errores en la modificación del producto.

Otras cuestiones: Ninguna.

Caso de uso número 29: “alta de usuario”

Resumen de la funcionalidad: Da de alta un usuario en el sistema.

Papel en el trabajo del usuario: El administrador es el único que puede crear nuevas cuentas de usuario para fabricantes y representantes. Los clientes podrán darse de alta siempre que introduzcan el código de validación correcto.

Actores: Cliente, administrador.

Casos de uso relacionados: “baja de usuario”, “enviar mensaje”, “solicitar alta”.

Precondición: No existe la cuenta de un usuario.

Poscondición: La cuenta del usuario ha sido creada.

Descripción: Cuando intente dar de alta a un cliente nuevo, el administrador se encargará de crear su cuenta, introduciendo sus datos. Podrá hacer lo mismo con los fabricantes. Sólo el administrador podrá llevar a cabo esta tarea.

Cuando la cuenta haya sido creada, se dará el caso de uso “envío de mensaje” en el que el sistema enviará al usuario un mensaje notificándole que ha sido dado de alta en el sistema, e informándole de su nick y password.

Alternativas de proceso y excepciones: El sistema deberá avisar al administrador de todos los errores que se produzcan en el proceso de alta. Debe comprobar si el usuario ya estaba registrado, si el nick elegido ya está en uso, etc.

Otras cuestiones: Ninguna.

Caso de uso número 30: “baja de usuario”

Resumen de la funcionalidad: Da de baja a un usuario del sistema.

Papel en el trabajo del usuario: El administrador es el único que puede dar de baja al resto de usuarios.

Actores: Administrador.

Casos de uso relacionados: “alta de usuario”, “enviar mensaje”.

Precondición: Existe la cuenta del usuario.

Poscondición: La cuenta del usuario ya no existe, aunque sus datos no se borran.

Descripción: Cuando intente dar de baja a un usuario, el administrador lo hará. En realidad, la cuenta no desaparece sino que pasa al estado “desactivado”. De este modo, siguen estando accesibles los datos del usuario, su histórico de pedidos, sus facturas, etc.

Cuando la cuenta haya sido dada de baja, se dará el caso de uso “envío de mensaje” en el que el sistema enviará al usuario un mensaje notificándole que ha sido dado de baja en el sistema.

Alternativas de proceso y excepciones: El sistema deberá avisar al administrador de todos los errores que se produzcan en el proceso de baja.

Otras cuestiones: Ninguna.

Caso de uso número 31: “modificar estado de pedido”

Resumen de la funcionalidad: Permite a un fabricante modificar el estado de un pedido dentro del sistema.

Papel en el trabajo del usuario: Los fabricantes deben actualizar el estado de los pedidos en el sistema para poder efectuar un correcto seguimiento de los mismos.

Actores: Fabricante.

Casos de uso relacionados: “buscar pedidos según criterio”, “ver pedido”.

Precondición: Existe el pedido.

Poscondición: El estado del pedido ha sido modificado

Descripción: Cuando un pedido llega al sistema se encuentra en estado “recibido”. Cuando un fabricante va a prepararlo, debe cambiar su estado a “en proceso”. Si por algún motivo para procesarlo, deberá ponerlo en estado “pausado”. Cuando lo termine, cambiará su estado a “pendiente de envío”. Finalmente, cuando lo envíe, lo marcará como “enviado”.

Alternativas de proceso y excepciones: ninguna.

Otras cuestiones: Ninguna.

Caso de uso número 32: “solicitar alta”

Resumen de la funcionalidad: Permite a un cliente solicitar el alta en el sistema.

Papel en el trabajo del usuario: Para que un cliente pueda interactuar con el sistema debe estar dado de alta.

Actores: Cliente.

Casos de uso relacionados: “baja de usuario”, “enviar mensaje”, “solicitar baja”.

Precondición: No existe la cuenta de un cliente.

Poscondición: El cliente ha solicitado ser dado de alta.

Descripción: El cliente rellena un formulario mediante el que solicita ser dado de alta. Si el administrador considera que debe darle de alta, enviará al cliente un código de validación para que pueda darse de alta en el sistema.

Alternativas de proceso y excepciones: ninguna.

Otras cuestiones: Ninguna.

Caso de uso número 33: “solicitar baja”

Resumen de la funcionalidad: Permite a un cliente solicitar darse de baja en el sistema.

Papel en el trabajo del usuario: Permite a un cliente darse de baja.

Actores: Cliente.

Casos de uso relacionados: “baja de usuario”, “enviar mensaje”, “solicitar alta”.

Precondición: Existe la cuenta de un cliente.

Poscondición: El cliente ha solicitado ser dado de baja.

Descripción: El cliente rellena un formulario mediante el que solicita ser dado de baja. El sistema se encargará de darle de baja y de enviarle un mensaje confirmando que se ha dado de baja con éxito.

Alternativas de proceso y excepciones: ninguna.

Otras cuestiones: Ninguna.

Caso de uso número 34: “comenzar pedido”

Resumen de la funcionalidad: comienza un nuevo pedido.

Papel en el trabajo del usuario: Es el primer paso a la hora de hacer un pedido

Actores: Cliente.

Casos de uso relacionados: “modificar pedido actual”, “añadir producto al pedido”, “eliminar pedido”.

Precondición: El pedido no existe.

Poscondición: El pedido existe.

Descripción: Cuando un cliente hace click en el botón “comenzar pedido”, se crea un nuevo pedido listo para añadir productos.

Alternativas de proceso y excepciones: avisar en caso de que ya haya un pedido en curso.

Otras cuestiones: ninguna.

Caso de uso número 35: “añadir producto al pedido”

Resumen de la funcionalidad: Añade un producto al pedido en curso.

Papel en el trabajo del usuario: En el proceso de realización de un pedido por parte de un cliente, añade un producto al mismo.

Actores: Cliente.

Casos de uso relacionados: “modificar pedido actual”.

Precondición: El producto no ha sido añadido previamente.

Poscondición: Se ha añadido el producto al pedido.

Descripción: Cuando un cliente hace click en el botón “añadir al pedido” de un producto, éste queda añadido al pedido en curso.

Alternativas de proceso y excepciones: Aviso en caso de que el producto no esté disponible o de que ya haya sido añadido al pedido.

Otras cuestiones: Al lado del botón “añadir al pedido” aparecerá por defecto un uno, que indica el número de unidades del producto. El cliente podrá cambiar este número por tantas unidades como desee (siempre que haya stock).

Caso de uso número 36 “modificar pedido en curso”

Resumen de la funcionalidad: Modifica el pedido en curso.

Papel en el trabajo del usuario: En el proceso de realización de un pedido por parte de un cliente, permite eliminar productos del mismo o modificar el número de unidades.

Actores: Cliente.

Casos de uso relacionados: Ninguno.

Precondición: El pedido existe.

Poscondición: El pedido ha sido modificado.

Descripción: Cuando un cliente consulta el estado del pedido que está efectuando y hace click en el botón “eliminar” correspondiente a un producto que ha añadido previamente, éste queda eliminado del pedido. El usuario también puede modificar el número de unidades de un producto. Para que el pedido se modifique realmente, el usuario debe pulsar el botón “guardar cambios”.

Alternativas de proceso y excepciones: Ninguna.

Otras cuestiones: Ninguna.

Caso de uso número 37: “ver pedido en curso”

Resumen de la funcionalidad: Consulta el pedido que se está realizando.

Papel en el trabajo del usuario: En el proceso de realización de un pedido por parte de un cliente, permite consultar los productos que ha añadido.

Actores: Cliente.

Casos de uso relacionados: “modificar pedido actual”, “procesar pedido actual”.

Precondición: El producto ha sido añadido previamente.

Poscondición: Se muestra la información del pedido.

Descripción: Cuando un cliente consulta el estado del pedido que está efectuando aparece la información relativa al mismo, y ofrece la posibilidad de modificarlo, continuar añadiendo productos o confirmar el pedido y efectuar el pago.

Alternativas de proceso y excepciones: Avisar en caso de error en la consulta del pedido.

Otras cuestiones: Ninguna.

Caso de uso número 38: “pagar pedido”

Resumen de la funcionalidad: Efectúa el proceso de pago del pedido.

Papel en el trabajo del usuario: El usuario paga el pedido con el medio de pago que elija..

Actores: Cliente.

Casos de uso relacionados: “enviar mensaje”.

Precondición: El pedido está completo.

Poscondición: El pedido ha sido pagado.

Descripción: Cuando el cliente acepta las condiciones del servicio, selecciona el medio de pago e introduce los datos necesarios, el sistema efectúa el cargo en su tarjeta de crédito. Si el proceso concluye con éxito, envía un mensaje al usuario confirmándole la compra, y adjuntándole la factura correspondiente. Por otro lado, envía un mensaje al administrador y a los fabricantes informándoles de que ha entrado un pedido nuevo.

Alternativas de proceso y excepciones: Avisar en caso de error en el pago.

Otras cuestiones: Ninguna.

Caso de uso número 39: “guardar pedido en curso”

Resumen de la funcionalidad: guarda el pedido en curso.

Papel en el trabajo del usuario: En el proceso de realización de un pedido por parte de un cliente, permite guardar el pedido en curso para poder retomarlo más adelante.

Actores: Cliente.

Casos de uso relacionados: Ninguno.

Precondición: El pedido existe.

Poscondición: El pedido ha sido guardado.

Descripción: Un cliente puede guardar el pedido en curso para retomarlo más adelante, pero sólo se guardará por un periodo de 24 horas.

Alternativas de proceso y excepciones: Ninguna.

Otras cuestiones: Ninguna.

Caso de uso número 40: “reanudar pedido”

Resumen de la funcionalidad: Recupera un pedido guardado.

Papel en el trabajo del usuario: Permite a un cliente recuperar un pedido que haya guardado previamente.

Actores: Cliente.

Casos de uso relacionados: Ninguno.

Precondición: El pedido está guardado.

Poscondición: El pedido se ha recuperado.

Descripción: Un usuario puede retomar un pedido que haya guardado con anterioridad para seguir añadiendo productos, modificarlos o pagarlo.

Alternativas de proceso y excepciones: Avisar en caso de que hayan transcurrido más de 24 horas desde que se guardó el pedido.

Otras cuestiones: Ninguna.

Caso de uso número 41: “enviar mensaje”

Resumen de la funcionalidad: Envía un email informativo al usuario.

Papel en el trabajo del usuario: Sirve para comunicar algunos eventos a los usuarios, como la entrada de un pedido a los fabricantes, el cambio de contraseña a un usuario, etc.

Actores: Sistema.

Casos de uso relacionados: “cambio de contraseña y/o nick”, “alta de usuario”, “baja de usuario”, “modificar datos de usuario”, “modificar pedido”, “modificar factura”, “pagar pedido actual”.

Precondición: El usuario al que se le envía el email existe en el sistema y tiene estado activo.

Poscondición: El mensaje ha sido enviado.

Descripción: Al producirse ciertos eventos que requieren la comunicación de información a los usuarios, el sistema les envía un mensaje informativo según el tipo de evento que se produzca.

Alternativas de proceso y excepciones: Ninguna.

Otras cuestiones: Ninguna.

2.6 Requisitos de la interfaz de usuario

El sistema se compone de una parte de administración, que es la que verán los empleados (administrador, fabricantes y representantes) y la tienda virtual.

En la parte de administración se prestará mayor atención a la usabilidad mientras que el diseño quedará relegado a un segundo plano. Sin embargo en la tienda virtual, se prestará igual atención al diseño y a la usabilidad.

Aunque los empleados están habituados al uso de ordenadores, disponen de conocimientos muy básicos de entornos administrativos, así que la interfaz de la parte de administración deberá ser muy intuitiva.

En cuanto a los clientes, utilizarán el sistema como una tienda virtual, con las convenciones típicas de este tipo de site, así que no tendrán problemas a la hora de interactuar. De todas formas, se buscará un diseño de interfaz intuitiva y accesible.

3. Análisis

3.1 Identificación de las clases de entidades y sus atributos

Usuario: tipoUsuario (int), nick(String), password(String), nombre(String), apellidos(String), documentoIdentificativo(DocumentoIdentificativo), sexo (int), direccion(String), codigoPostal(String), idPais(int), idProvincia(int), telefono(String),

telefonoMovil(String), skype(String), estado(int), fechaNacimiento(GregorianCalendar), email(String), web(String), tablaPermisos(HashMap<int, permiso>), idioma(int).

Representante: codigoAutorizador(String), observaciones(String), zona(int).

Cliente: codigoAutorizacion(String), nombreComercial(String), descripcionComercio(String), comoNosHaConocido(String), moneda(int), zona(int).

Manufacturador: observaciones(String).

Administrador

Pedido: idPedido(String), estado(int), prioridad(Prioridad), idCliente(String), estadoCierre(int).

Factura: numeroDeFactura(String), idPedido(String), fecha(GregorianCalendar), estado(int), importe(float), formaPago(FormaPago).

Coleccion: id(String), nombre(String), descripcion(String), estado(int), privacidad(int), imagenes(String[]).

Producto: id(String), nombre(String), descripcion(String), estado(int), privacidad(int), imagenes(String[]), idColeccion(String), tipo(tipoProducto), estilo(estiloProducto), clasificacion(clasificacionProducto), materiales(String), unidadesDisponibles(int), precio(float), precioRebajado(float), precioRebajado(float), precioDetalle(float).

3.2 Diagramas de estados

Diagrama de estados de la clase Pedido:

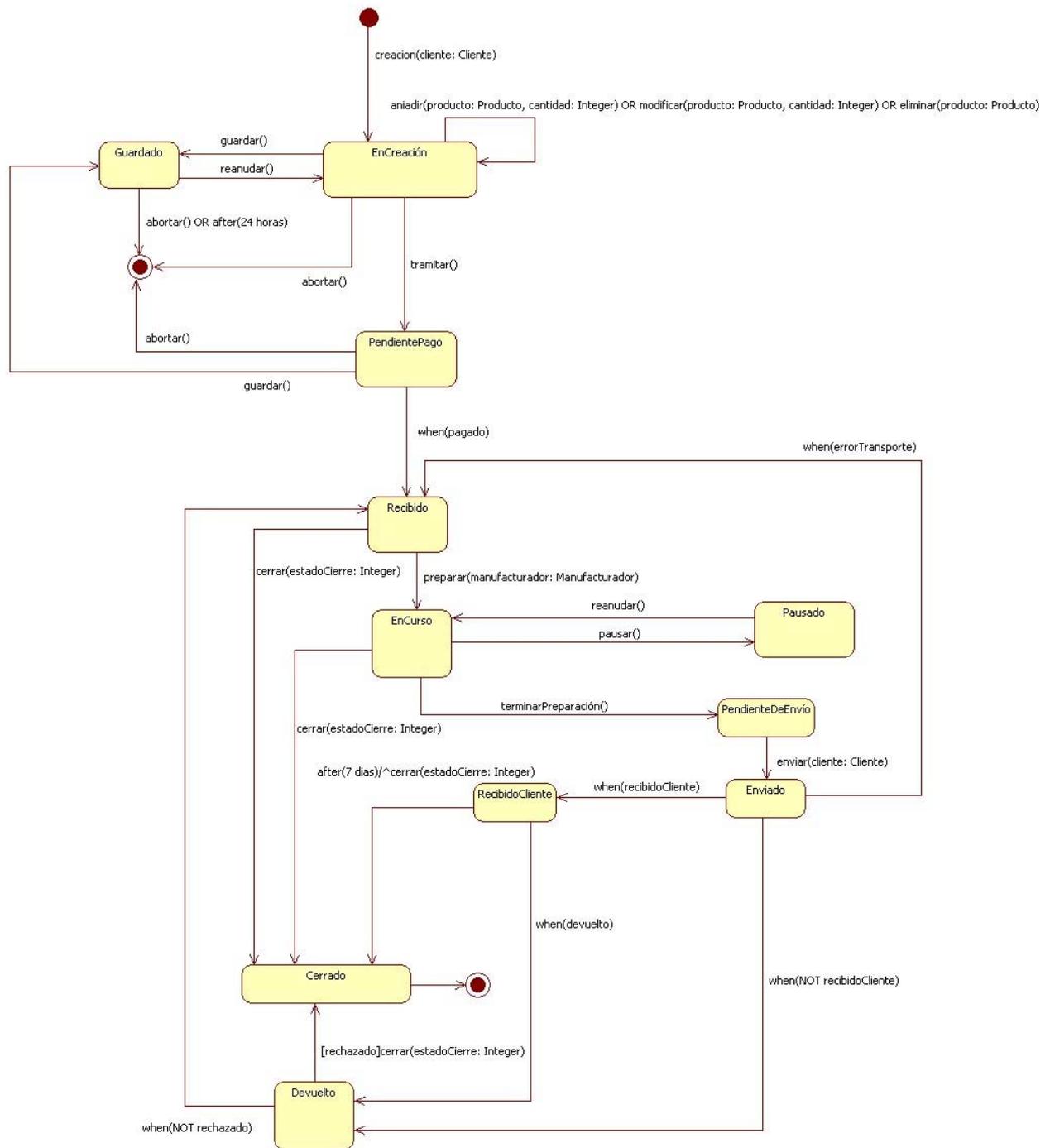


Diagrama de estados de la clase Factura:

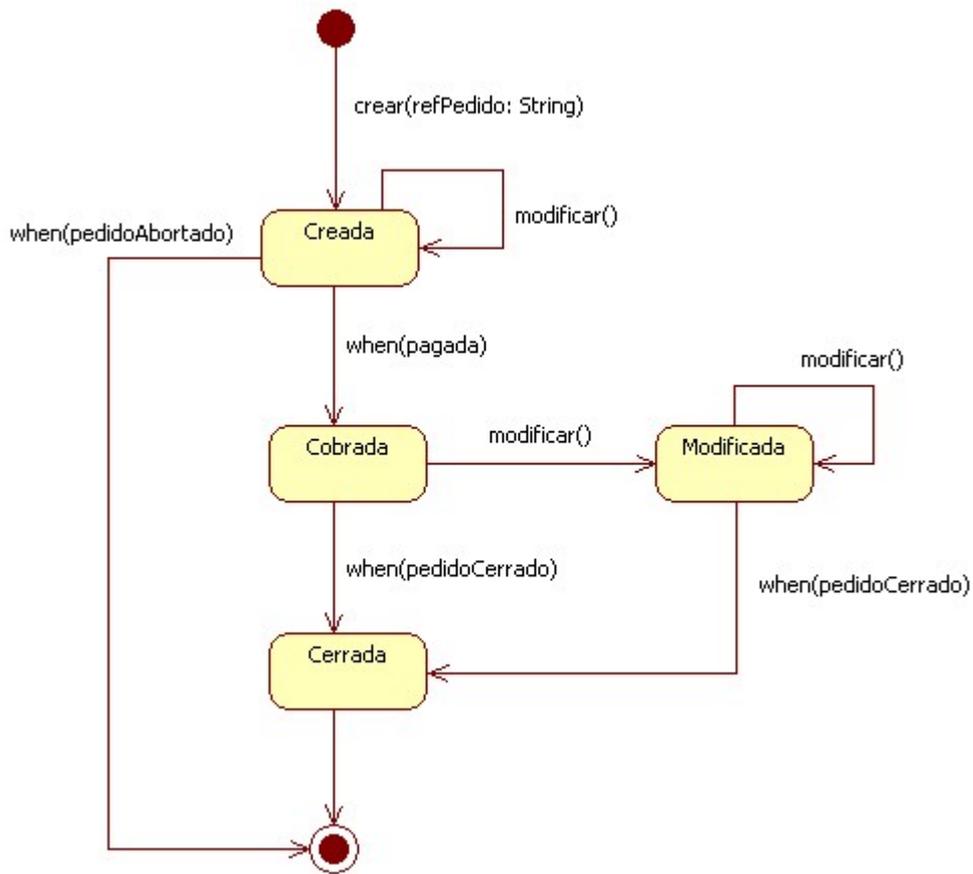


Diagrama de estados de los usuarios:

Diagrama de estados de los Clientes.

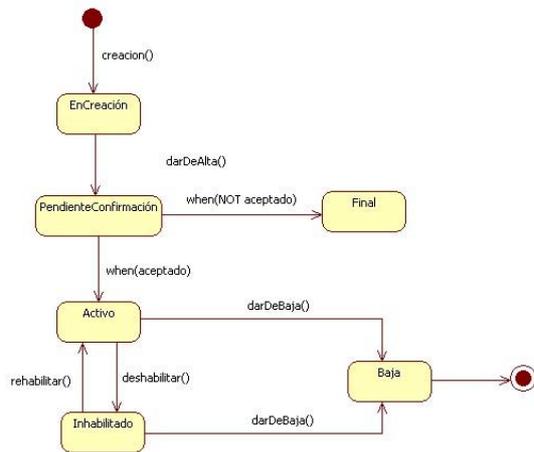


Diagrama de estados de los Manufacturadores y Representantes.

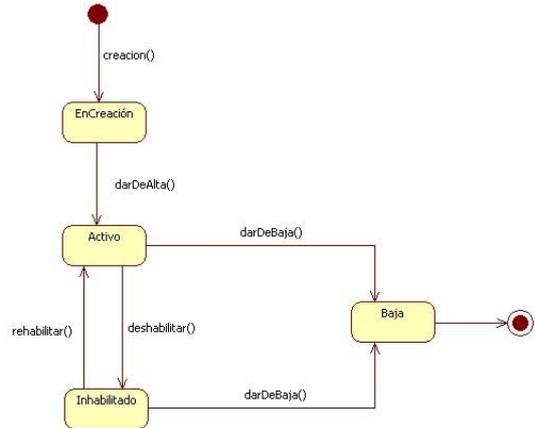
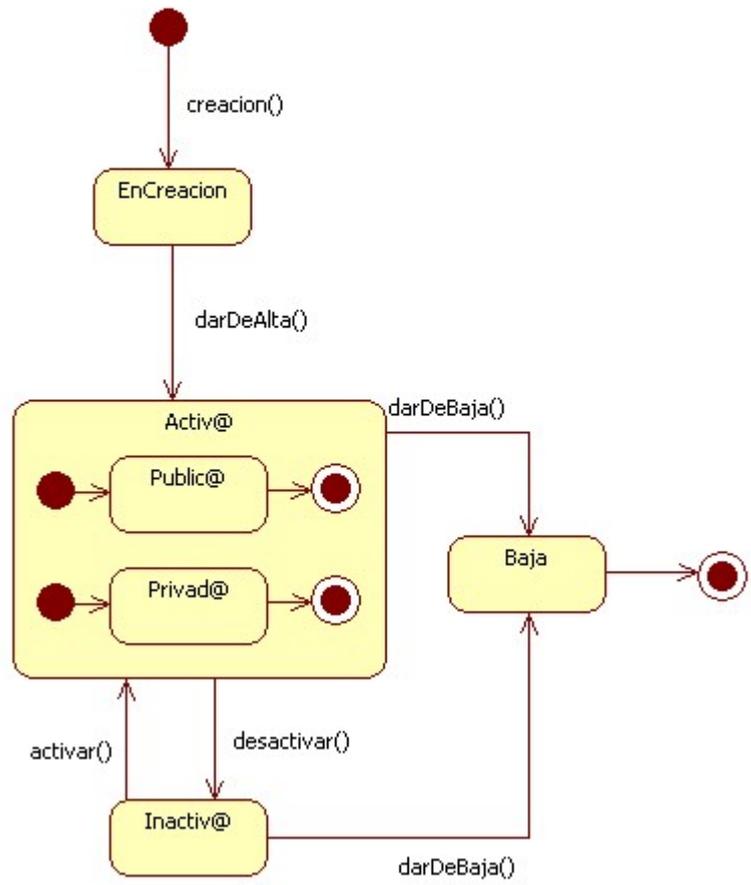


Diagrama de estados de las colecciones y los productos:

Diagrama de estados válido para Colecciones y Productos.



3.3 Diagramas de actividades

Diagrama de actividades correspondiente a la clase Pedido:

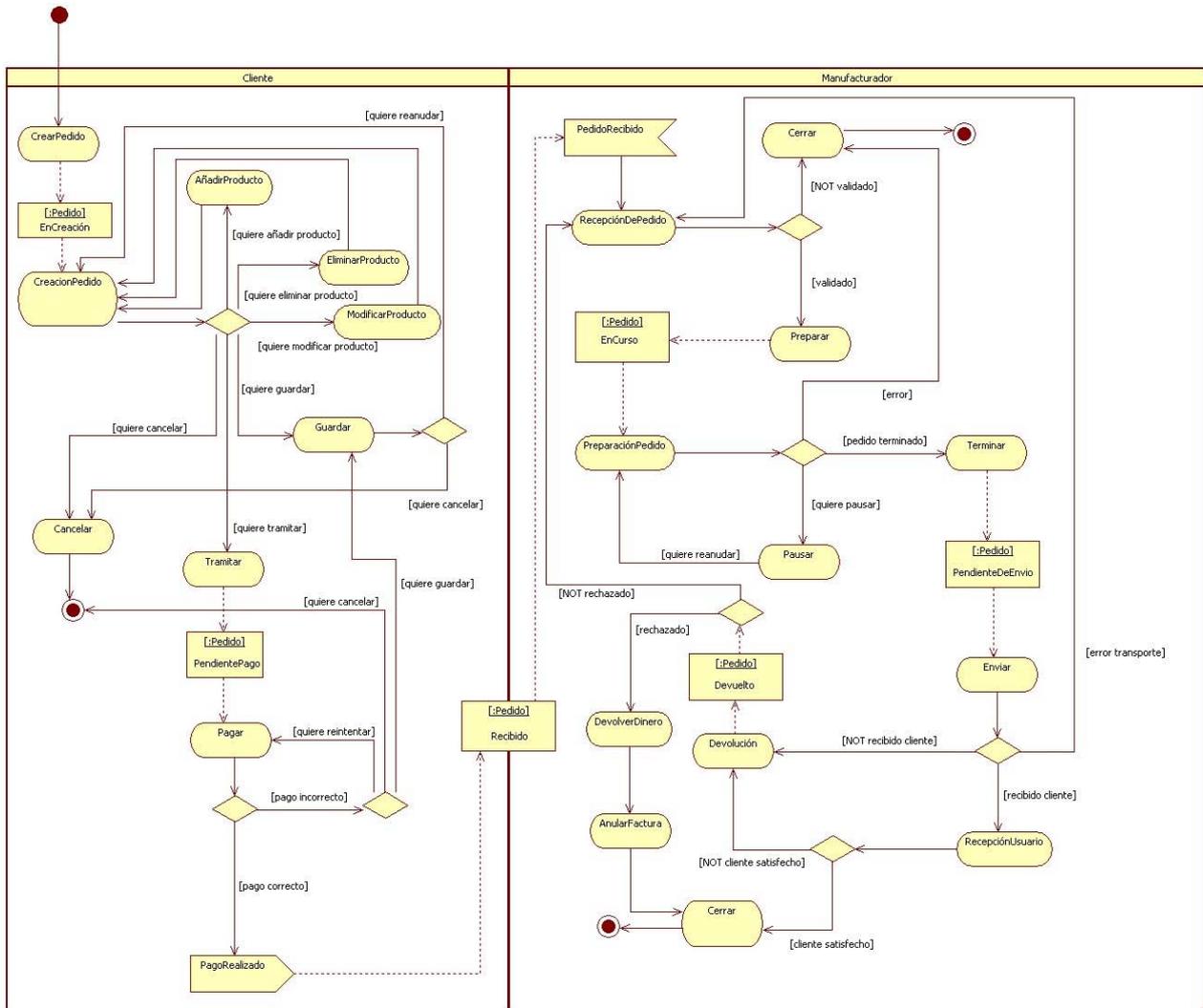
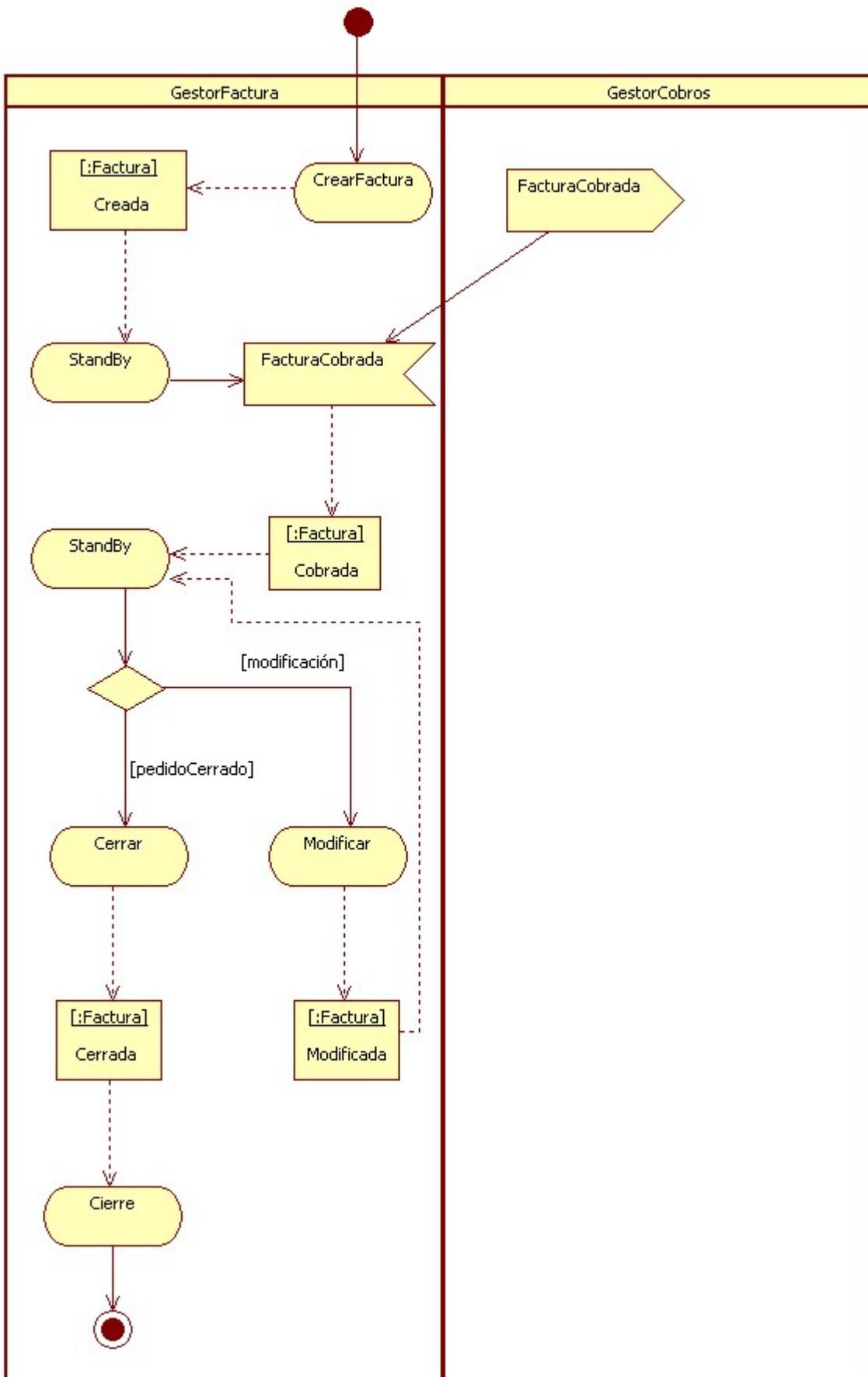


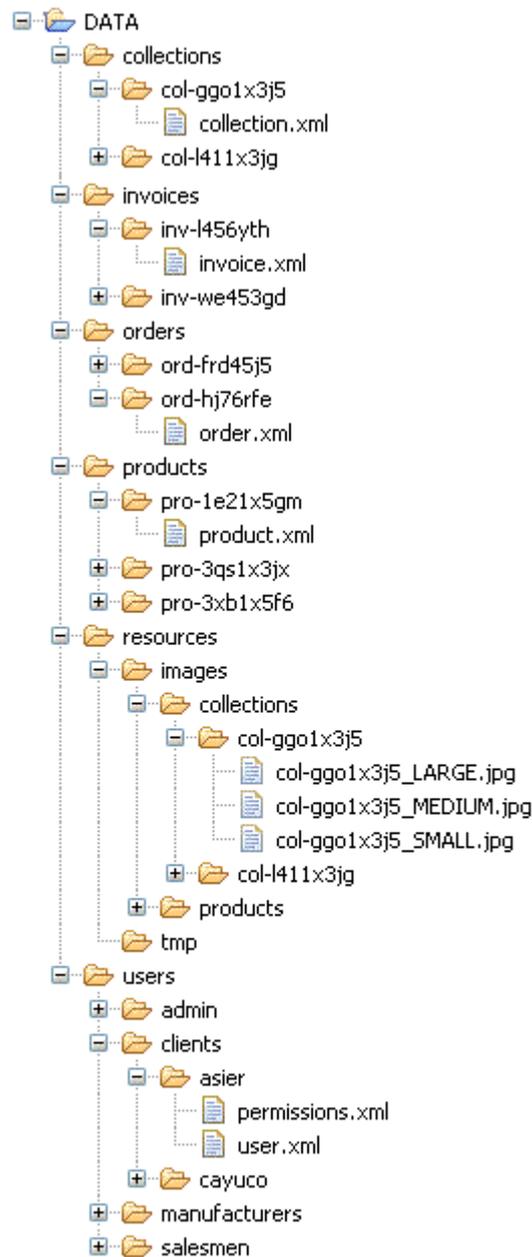
Diagrama de actividades correspondiente a la clase Factura:



3.4 Modelo de datos

Se ha optado por el almacenamiento de los datos en formato xml. Para ello, se ha organizado una estructura de directorios en la que se almacenarán los ficheros relativos a los diferentes objetos que intervienen en la aplicación.

A continuación puede verse cómo es la estructura de directorios utilizada.



Cada uno de los objetos principales de la aplicación (usuarios, colecciones, productos, pedidos y facturas) tienen un directorio asignado (users, collections, products, invoices y orders). En el caso de los usuarios hay un subdirectorio por cada tipo de usuario.

En los directorios correspondientes a los objetos, se crea un subdirectorio cuyo nombre es el id del objeto, y dentro de éste va un fichero xml con los datos del objeto.

En el caso de los usuarios, además se guarda la lista de permisos que tiene asociados.

Hay otro directorio, resources, en el que se guardan las imágenes de las colecciones y los productos, y también podrían guardarse videos, etc. Dentro del subdirectorio images, hay otros dos, uno para las colecciones y otro para los productos. Dentro de éstos, se guardan las imágenes en un subdirectorio cuyo nombre es el id del objeto correspondiente. Las imágenes se nombran con el id del objeto y los sufijos SMALL, MEDIUM y LARGE, para cada uno de los tamaños posibles.

A continuación se muestra un ejemplo de los ficheros xml relativos a un usuario.

Fichero user.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<user>
  <data language="0" sex="0" status="1" type="3"/>
  <nick><![CDATA[minick]]></nick>
  <password><![CDATA[1111]]></password>
  <name><![CDATA[pedro]]></name>
  <surnames><![CDATA[lopez mora]]></surnames>
  <doc-identifier id="0" value="449312357A"/>
  <location country="17" province="48"/>
  <address><![CDATA[c/ Las Heras, 24 bajo dch.]]></address>
  <post-code><![CDATA[48543]]></post-code>
  <phone-number><![CDATA[944343234]]></phone-number>
  <mobile><![CDATA[607345676]]></mobile>
  <web><![CDATA[www.mitienda.com]]></web>
  <skype><![CDATA[skype]]></skype>
  <email><![CDATA[pedro@lopez.edu]]></email>
  <birth-date dd="3" mm="7" yyyy="1975"/>
  <birth-date-millis><![CDATA[173574000000]]></birth-date-millis>
  <regdate><![CDATA[2007-05-28]]></regdate>
  <leave-date dd="00" mm="00" yyyy="00"/>
  <leave-date-millis><![CDATA[0]]></leave-date-millis>
  <auth-code><![CDATA[1111]]></auth-code>
  <zone id="4"/>
  <commercial-name><![CDATA[bisutex]]></commercial-name>
  <salesman><![CDATA[Manolo Pérez]]></salesman><currency id="1"/>
  <how-knew-us id="3"/>
  <description><![CDATA[Tienda de alta jolleria en la Gran Vía]]></description>
</user>
```

Fichero permissions.xml

```
<?xml version="1.0" encoding="iso-8859-1"?>
<permissions nick="asier" user-type="3">
  <user change="0" create="0" delete="0" view="0" />
  <order change="0" create="1" delete="0" view="1" />
  <invoice change="0" create="0" delete="0" view="1" />
  <collection change="0" create="0" delete="0" view="1" />
  <product change="0" create="0" delete="0" view="1" />
</permissions>
```

4. Diseño técnico

4.1 Concepto de patrón

Los patrones de diseño son soluciones a problemas comunes, obtenidas a partir de la experiencia. Su objetivo es aprovechar el conocimiento que han adquirido diseñadores expertos y documentar las soluciones obtenidas para problemas concretos, con el fin de que el resto de diseñadores puedan utilizarlos.

Un patrón de diseño consta de los siguientes elementos :

Nombre del patrón: para referirnos a él.

El Problema: describe cuándo debe aplicarse.

La Solución: describe los elementos que componen el diseño y cómo se relacionan.

Las Consecuencias: describe las consecuencias que tiene su aplicación

4.2 El patrón Service To Worker

Contexto

Los Sistemas manejan solicitudes Web. Los procesos de la capa de presentación requieren la generación de una vista basada en una plantilla y un modelo dinámico.

Problema

Los cambios de la capa de Presentación ocurren a menudo y son difíciles de desarrollar y mantener, debido a la combinación de la lógica de negocio para el acceso a los datos y la lógica de formateo de la presentación, esto hace que el sistema sea menos flexible, menos reutilizable, y menos adaptable a los cambios.

Las porciones de lógica de negocio que son mezcladas con las vistas deben ser adaptadas a un modelo intermedio para su visualización.

Mezclar la lógica de negocio con el procesamiento de las vistas también reduce la modularidad y disminuye la separación de roles a la hora de asignar tareas a los miembros de un equipo de desarrollo de software o de aplicaciones web.

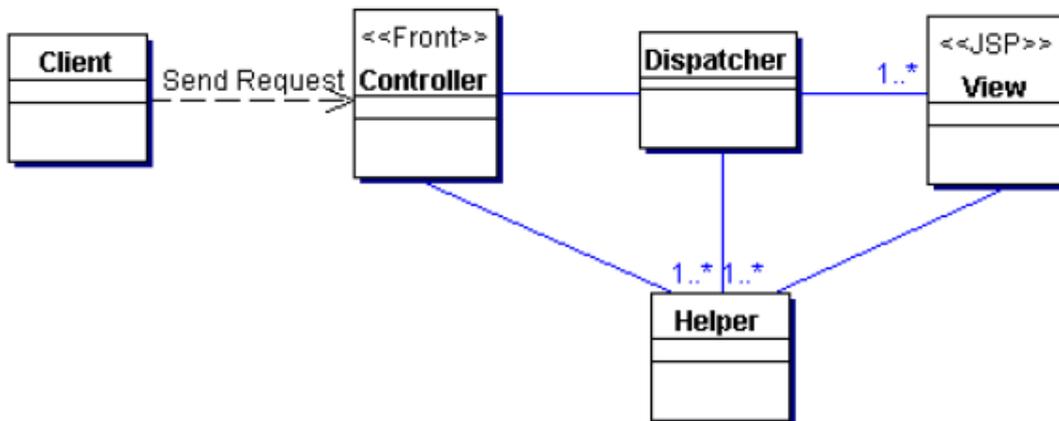
Fortalezas

- Necesidad de generación dinámica de contenido
- El contenido requiere que los datos de negocio sean extraídos dinámicamente.
- Los servicios de procesamiento comunes del sistema, tales como autenticación y chequeo de permisos deben realizarse por cada solicitud entrante.
- Crear puntos de decisión para la recuperación y manipulación de contenido
- Usar múltiples vistas para responder a solicitudes similares.
- Encapsular la lógica de negocio en otros componentes distintos de las Vistas.

Solución

Combinar un despachador con vistas y ayudantes para manejar las solicitudes de los clientes y preparar una presentación dinámica como respuesta. Un despachador es responsable por el manejo de vistas y la navegación, y puede ser o no encapsulado dentro de un controlador, o puede ser un componente independiente que trabaje de forma sincronizada con los componentes internos.

A continuación se muestra el diagrama de clases que representan el patrón Service To Worker:



Los componentes del diagrama de clases se especifican a continuación:

Controlador

El Controlador es el punto de contacto inicial para el manejo de una solicitud. Este trabaja con un Despachador para completar el manejo de vistas y la navegación.

Despachador

Un Despachador es responsable del manejo de Vistas y navegación, ya que este se encarga de seleccionar la próxima vista a ser presentada al Usuario, además de proveer un mecanismo para el control vectorizado de un recurso.

Vista

Una vista representa y visualiza información para los clientes. La información que es usada en una vista es tomada de un modelo.

Ayudante

Un ayudante es responsable de contribuir a completar el procesamiento que realiza una vista o un Controlador. Así, los ayudantes tienen numerosas responsabilidades, incluyendo la recolección de datos requeridos por una vista y adaptarlos al modelo de datos usado por la vista. Los ayudantes pueden atender las solicitudes de datos de una vista con el simple hecho de proveerle a la misma acceso a los metadatos o formateando los datos como contenido Web.

Una vista puede trabajar con cualquier número de ayudantes, los cuales son implementados típicamente como JavaBeans y etiquetas personalizadas. Adicionalmente, un ayudante puede representar un objeto de tipo Command, Delegate o también un objeto de tipo XSLT, el cual es usado en combinación con una hoja de estilo para adaptar y convertir el modelo a una forma apropiada.

4.3 Problemas con las típicas aproximaciones Servlet

La separación de los datos, la presentación y la lógica de programación siempre ha sido un problema para los desarrolladores de servlets. La primera generación de servlets se diseñó para reemplazar a los programas CGI, y normalmente contenían código que se parecía a este:

```
PrintWriter pw = response.getWriter();
pw.println("<html>");
pw.println("<head><title>Home Page</title></head>");
pw.println("<html>");
pw.println("<h1>Welcome to our web site</h1>");
```

Esta aproximación es tediosa, propensa a errores, y difícil de mantener. Bloquea completamente la autoría Web para los no-programadores, y requiere un esfuerzo significativo para cambiar el look de una site Web generada de esta forma. Aunque generalmente escribiremos el código en fragmentos modulares de código HTML, la aproximación aún requiere que se modifique el código fuente ante cualquier nuevo requerimiento.

Java Server Pages (JSP) proporciona una solución a este problema. Desafortunadamente, JSP sólo invierte el problema. En lugar de embeber código HTML en servlets, normalmente terminamos embebiendo código Java en nuestras páginas HTML. Las etiquetas JSP personalizadas nos permiten eliminar todo el código Java, en teoría, pero no sin la dificultad de crear las etiquetas o encontrar una librería existente que cubra nuestras necesidades. Ejemplo:

```
<%@ taglib uri="/mi_taglib" prefix="abc" %>
<html>
  <head>
    <title>Ejemplo de tag library</title>
  </head>
  <body>
    <abc:cabecera/>
    <abc:logo/>

    <h1>titulares recientes</h1>
    <abc:titulares filtro="recientes"/>

    <abc:pie/>
  </body>
</html>
```

Incluso con etiquetas personalizadas es virtualmente imposible eliminar todo el código Java en un solución JSP pura. Específicamente, la aproximación JSP pura se viene abajo cuando se envían complejos formularios HTML y la página JSP tiene que realizar validaciones.

4.4 La Aproximación XSLT + Servlet

Una arquitectura basada en XSLT+Servlets permite separar completamente el código java del HTML y supone una mejora frente a la separación de datos y lógica descrita en el modelo vista controlador.

Pueden utilizarse diferentes hojas de estilo para diferentes dispositivos y formatos de salida, no limitándonos al HTML. Por ejemplo, en el caso de la aplicación del proyecto, podría utilizarse una hoja de estilos específica para generar catálogos de productos en pdf.

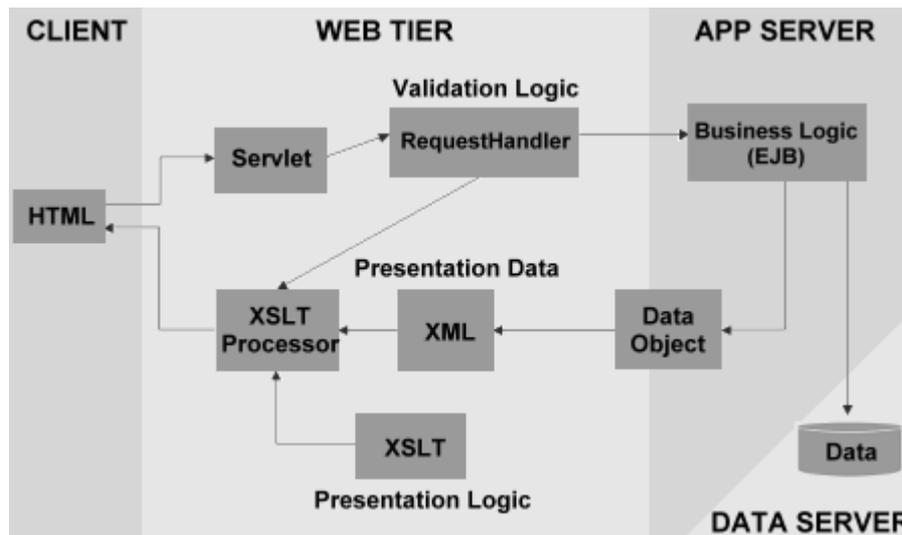
Entre los tipos de aplicaciones que se benefician de este tipo de aproximación se incluyen:

- Sitios Web que necesitan dirigirse a muchos navegadores incompatibles o quizás dispositivos sin cable;
- Aplicaciones Web que necesitan proporcionar datos XML además de interfaces XML;
- Complejas aplicaciones Web donde tiene sentido el diseño de una forma muy modular; forzando a una separación limpia entre datos, presentación y lógica de programación; y
- Aplicaciones Web que necesitan dirigirse a diferentes conjuntos de idiomas y juegos de caracteres desde el mismo modelo de datos.

Es difícil de imaginar una situación donde XML y XSLT no podrían funcionar, incluso más elegantemente que las aproximaciones JSP o puro servlet.

La principal desventaja de la aproximación XSLT podría ser el rendimiento en tiempo de ejecución. Cada petición al servlet requiere una transformación XSLT. Para la mayoría de las aplicaciones el rendimiento es suficientemente rápido; unas pocas técnicas de optimización pueden mejorar la situación donde se requiere la máxima potencia. Primero, las hojas de estilo XSLT son simples ficheros XML estáticos. Puede cargarse una vez en memoria y reutilizarlos una y otra vez. Esto mejora substancialmente el rendimiento porque el XML sólo tiene que ser analizado una vez.

Ejemplo típico de arquitectura:



4.5 Uso del patrón Service To Worker y XSLT

Como consecuencia de lo expuesto anteriormente, me decidí a desarrollar la aplicación web siguiendo el patrón Service To Worker, pero utilizando XSLT en lugar de el habitual JSP para las vistas.

Controlador: Es el servlet llamado WebappServlet, que recibe las peticiones del cliente. Debe recibir un parámetro llamado M, cuyo valor será el nombre de la vista que se debe mostrar.

Despachador : Clase WebApp, que se encarga de seleccionar la vista adecuada con ayuda del ayudante y de controlar el acceso a la misma.

Ayudante: Clase ServiceConfig, mapea las vistas a partir del fichero de configuración service.config. Transforma el XML que le pasa el despachador, en HTML mediante el parser XSLTTransformerPool.

Vistas: son las clases que extienden la interfaz AbstractServiceDataModule y están en los paquetes *amoreno.tfc.data* de las aplicaciones. En realidad la vista la componen estas clases y las hojas de estilo XSLT que se corresponden con cada una de ellas.

A *grosso modo*, el funcionamiento sería el siguiente:

El servlet WebappServlet (Controlador), en su método `init()`, crea una instancia de la clase ServiceConfig (Ayudante), quien cargará los datos necesarios para gestionar las vistas. Dichos datos los obtiene del fichero `service.config`. En dicho fichero se referencian las vistas de la siguiente forma:

```
<module name="CreateUser" dataclass="amoreno.tfc.data.users.CreateUser"/>
```

El atributo `name` indica el nombre de la vista, y `dataclass`, la clase correspondiente. La hoja de estilo XSLT que utiliza dicha vista, debe tener el mismo nombre que ésta. El nombre también se corresponde con el valor del parámetro M que recibe WebappServlet.

Cuando WebappServlet recibe una petición, crea una instancia de la clase WebApp (Despachador), y le pide que compruebe si el usuario puede acceder a la vista seleccionada. En caso afirmativo, WebApp pedirá a ServiceConfig que le devuelva la vista adecuada. Esta vista generará un

documento XML, que volverá a ser pasado a ServiceConfig para que devuelva el HTML correspondiente. Esta transformación la hará XSLTProcessorPool.

Una vez obtenido el código HTML, WebAppServlet se encargará de devolvérselo al navegador como respuesta a su petición.

5. Implementación

5.1 Herramientas de desarrollo y pruebas

Navegadores web Internet Explorer 6 y Mozilla Firefox 2.

Plataforma J2SE 1.5

IDE de desarrollo Eclipse 3.2

Plugin de Eclipse Altova XML Spy

Servidor Web Apache Tomcat 6

5.2 Proceso de instalación

Dentro del fichero amoreno.zip se encuentran todos los ficheros necesarios para hacer funcionar la aplicación.

Es necesario tener instalado el servidor Tomcat 5 o superior . Puede descargarse aquí:

<http://tomcat.apache.org/download-60.cgi>

También se precisa la instalación del JRE 1.5 o superior de Sun, que puede descargarse aquí:

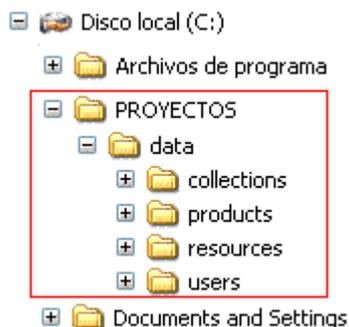
<http://java.sun.com/javase/downloads/index.jsp>

Hecho esto, puede procederse a la instalación de la aplicación.

Primero hay que descomprimir el fichero amoreno.zip. Dentro hay tres ficheros comprimidos en formato WAR: admin-lilyandlo, site-lilyandlo e imgserver y dos directorios, data y javalib.



El directorio llamado “data” contiene la estructura de directorios donde se almacenan los xml y correspondientes al modelo de datos de la aplicación, así como algunos datos de ejemplo. Hay que copiar este directorio en [C:/Proyectos](#), es decir, tendríamos el directorio [C:/Proyectos/data](#).

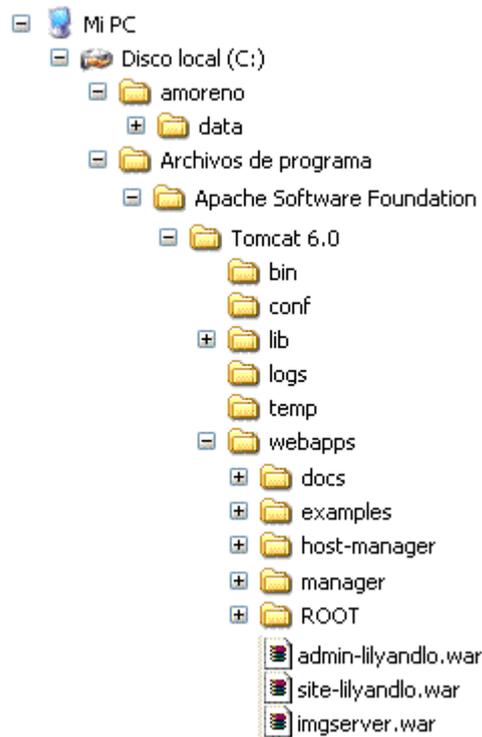


El directorios javalib contiene la librería generada durante la implementación del proyecto. Contiene clases de utilidad necesarias para el funcionamiento del site dinámico y del administrador. No es necesario hacer nada con ella, porque ya está incluida en los .war de las aplicaciones.

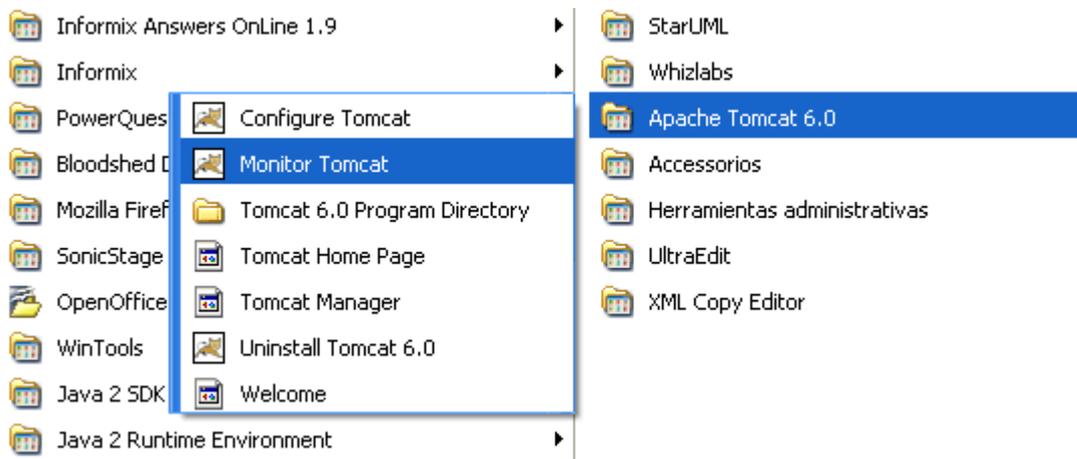
También se han incluido en el administrador (admin-lilyandlo) dos librerías necesarias para subir ficheros al servidor: Jakarta commons-fileupload-1.2.jar (http://jakarta.apache.org/site/downloads/downloads_commons-fileupload.cgi) y Jakarta commons-io-1.3.1.jar (http://jakarta.apache.org/site/downloads/downloads_commons-io.cgi)

El siguiente paso consiste en copiar los tres ficheros .war mencionados anteriormente, en el directorio *webapps* de Tomcat, que por defecto está en [C:\Archivos de programa\Apache Software Foundation\Tomcat 6.0](#)

Después hay que copiar los ficheros admin-lilyandlo.war, site-lilyandlo.war e imgserver.war en el directorio webapps, situado en el directorio de instalación de Tomcat.



Sólo queda arrancar Tomcat, para ello, hay que ejecutar Inicio > Todos los programas > Apache Tomcat 6 > Monitor Tomcat



Aparecerá un nuevo icono en la barra de herramientas.



Hay que situarse con el puntero sobre el icono, pulsar botón derecho y seleccionar “Start service” en el menú contextual que aparece.



Ya se puede empezar a usar las aplicaciones.

La url del administrador del site es <http://localhost:8080/admin-lilyandlo>

La del site es <http://localhost:8080/site-lilyandlo>

El nick del administrador es “admin” y su password, “admin”.

6. Conclusiones

La plataforma J2EE es ideal para el desarrollo de aplicaciones distribuidas en entornos empresariales. Su uso, junto con la aplicación de los patrones de diseño han contribuido a mejorar notablemente la reutilización de código y de la calidad de las aplicaciones que se generan.

Por otro lado, la utilización de xml para transmitir y almacenar datos mejora la portabilidad y la flexibilidad de las aplicaciones, e introduce grandes mejoras en la capa de presentación. El uso de parsers XSLT y hojas de estilos amplía las capacidades de J2EE en dicha capa y permite desarrollar aplicaciones web independientes del dispositivo en el que se mostrarán los datos, ya sea un teléfono móvil, un pc, una televisión un dispositivo para personas invidentes....

Dada la gran extensión de estas tecnologías, en el proyecto he podido aplicar una mínima parte de éstas, pero me ha servido para tener una visión global de su uso, conocer patrones, tendencias, frameworks, etc.

La falta de tiempo ha sido el gran enemigo con el que he tenido que luchar. El proyecto era ambicioso en cuanto a extensión, porque se compone de dos aplicaciones, que aunque comparten arquitectura, no dejan de implicar doble trabajo. Además se pretendía aportar un gran número de funcionalidades, que, desgraciadamente han quedado pendientes de implementar.

La conclusión general es muy positiva. La investigación en este area del desarrollo de software ha sido muy grata, y me ha ayudado a decidir que será hacia donde oriente mi carrera profesional.

7. Bibliografía

- Core J2EE Patterns: Best Practices and Design Strategies
(<http://www.corej2eepatterns.com/Patterns2ndEd/index.htm#>)
- The J2EE Architect's Handbook, DVT Press
- Servlets.com
<http://www.servlets.com>
- Designing Enterprise Applications with the J2EETM Platform, Second Edition, Addison-Wesley
- Documentación de Sun, (<http://java.sun.com/javase/5/docs/tutorial/doc/index.html>).
- XML programming bible, Wiley
- J2EE and XML development, Manning
- Mastering XSLT transformations, O'Reilly