

Restaurant-OnLine

Juan Carlos Serra Diaz

ETIG

Verónica Peña Pastor

13 de enero de 2009

Dedicatoria

Ante todo destacar que aunque este TFC haya sido fruto de mi esfuerzo, éste no hubiese podido llevarse a cabo sino fuese por el apoyo recibido en todo momento por las personas que me rodean, tanto en el hogar como en el trabajo. Por ello, deseo agradecer la paciencia demostrada por mis compañeros de trabajo cuando expresaba pensamientos en voz alta o realizaba explicaciones de lo que iba realizando a lo largo de mis estudios.

Mención especial merece mi esposa Geli y mis hijos, Virginia, David y Carlos que han tenido que padecer mis horas de ausencia dedicadas a la carrera y, sobre todo, a mis cambios de humor ante las dificultades de los trabajos realizados durante todos estos años, especialmente, durante este último semestre. Por ello, gracias a todos por vuestra comprensión y paciencia.

Resumen

La presente memoria pretende dar una visión general de los pasos seguidos, los componentes utilizados y las funcionalidades instaladas en la aplicación Restaurant-OnLine. Esta es una aplicación de comercio electrónico que, aunque no incluye la posibilidad de realizar el pago online, ofrece una alternativa a la realización de pedidos a establecimientos de restauración.

Para su elaboración se ha utilizado la arquitectura Java EE o Java Platform, Enterprise Edition (J2EE hasta la versión 1.4), muy apropiada para el desarrollo de aplicaciones web empresariales en lenguaje de programación Java.

Tras la finalización de este trabajo final de carrera se ha obtenido un importante conocimiento de esta plataforma, teniendo en cuenta que inicialmente se desconocía casi por completo. Por falta de tiempo no todos los conocimientos adquiridos han podido ser aplicados en Restaurant-OnLine, pero al menos han servido de punto de partida para empezar a disfrutar de esta tecnología.

Índice

Capítulo 1 Introducción.....	7
Justificación	7
Objetivos.....	7
Enfoque y método seguido.....	8
Planificación.....	9
Fase planificación	9
Fase especificación y análisis	9
Fase diseño	10
Fase implementación y testing	10
Fase documentación	10
Producto obtenido.....	11
Breve descripción de los otros capítulos de la memoria	11
Capítulo 2 Funcionamiento del sistema	12
Componentes de Restaurant-OnLine	14
Base de datos (capa persistencia)	14
Capa Modelo o lógica de negocio (Model)	14
Capa Controlador (Controller)	17
Capa presentación o Vista (View)	18
Capítulo 3 Análisis de las funcionalidades del sistema	20
Descripción de las funcionalidades	20
Identificación	20
Gestión de artículos.....	20
Gestión de pedidos.....	21
Gestión de usuarios	22
Casos de uso	22
Subsistema de Identificación	22
Subsistema de gestión de artículos.....	23
Subsistema de gestión de pedidos.....	24
Subsistema de gestión de usuarios.....	26
Diagramas de casos de uso.....	27
Casos de uso del subsistema de identificación.....	27
Casos de uso del subsistema de gestión de artículos.....	28
Casos de uso del subsistema de gestión de pedidos.....	28
Casos de uso del subsistema de gestión de usuarios.....	29
Capítulo 4 Diseño.....	30
Capa modelo.....	30

Clases correspondientes a los Transfer Object.....	30
Clases DAO	31
Capa Controlador	32
Clases comunes	32
Clases parte Cliente.....	32
Clases parte Empresa	32
Capa Vista o Presentación.....	34
Capítulo 5 Funcionamiento de Restaurant-OnLine.....	36
Inicio	36
Consultar artículos.....	37
Consultar nota de pedido.....	38
Realizar pedido	38
Confirmar pedido.....	39
Consultar pedidos tramitados.....	39
Registro de clientes	40
Menú de administración.....	40
Gestión de pedidos	41
Consultar pedidos	41
Admitir pedidos.....	42
Anular pedidos.....	42
Gestión de artículos	43
Consultar artículos	43
Modificar artículo.....	43
Alta artículos	44
Baja artículos	44
Gestión de usuarios	45
Consulta usuarios.....	45
Modificación de usuarios	46
Alta de usuarios	46
Baja de usuarios	47
Capítulo 6 Conclusiones	48
Capítulo 7 Glosario	49
Capítulo 8 Bibliografía.....	50

Índice de figuras

Capítulo 2 – Funcionamiento del sistema	
Figura 2.1. Arquitectura MVC-Modelo 2.....	13
Figura 2.2. Arquitectura Struts.....	14
Capítulo 3 – Análisis de las funcionalidades del sistema	
Figura 3.1. Diagrama de casos de uso del subsistema de identificación.....	28
Figura 3.2. Diagrama de casos de uso del subsistema de gestión de artículos.....	29
Figura 3.3. Diagrama de casos de uso del subsistema de gestión de pedidos.....	29
Figura 3.4. Diagrama de casos de uso del subsistema de gestión de usuarios.....	30
Capítulo 4 – Diseño	
Figura 4.1. Diagrama de clases Transfer Object.....	31
Figura 4.2. Diagrama de clases no persistentes.....	32
Figura 4.3. Diagrama de clases e interfaces DAO.....	32
Figura 4.4. Diagrama de clases e interfaces factoría de servicio.....	32
Capítulo 5 – Funcionamiento del Restaurant-OnLine	
Figura 5.1. Inicio de Restaurant-OnLine.....	37
Figura 5.2. Consulta categoría de artículos lado cliente.....	38
Figura 5.3. Consulta artículos lado cliente.....	38
Figura 5.4. Consulta nota de pedido lado cliente.....	39
Figura 5.5. Login de clientes.....	39
Figura 5.6. Confirmación nota pedido de cliente.....	40
Figura 5.7. Consulta pedidos pendientes por cliente.....	40
Figura 5.8. Registro de clientes.....	41
Figura 5.9. Menú administración lado empresa.....	41
Figura 5.10. Consulta pedidos lado empresa.....	42
Figura 5.11. Lista pedidos según criterio de consulta lado empresa.....	42
Figura 5.12. Admisión pedidos.....	43
Figura 5.13. Anulación pedidos.....	43
Figura 5.14. Selección criterio de consulta de artículos lado empresa.....	44
Figura 5.15. Modificación artículo.....	44
Figura 5.16. Alta artículos.....	45
Figura 5.17. Baja artículos.....	45
Figura 5.18. Selección criterio de consulta de usuarios.....	46
Figura 5.19. Lista tipo de usuarios.....	46
Figura 5.20. Modificación usuario.....	47
Figura 5.21. Alta usuario.....	47
Figura 5.22. Baja usuario.....	48

Capítulo 1 Introducción

Justificación

Las posibilidades que brinda la plataforma de Internet para la proliferación de transacciones comerciales son inmensas y han sido el motivo principal que me han impulsado a elegir como trabajo final de carrera (TFC) la creación de una aplicación de comercio electrónico.

Mi conexión con el mundo de la hostelería desde los primeros años de mi infancia han hecho que me decantara por la temática de la venta de las especialidades elaboradas por un establecimiento de restauración. He observado a lo largo de estos años que, en muchas ocasiones, los clientes desean realizar los pedidos con cierta antelación (con un día o, incluso, varios días), lo que implica la molestia de tener que personarse en el establecimiento o efectuar llamadas telefónicas que pueden producir cierta desconfianza en el establecimiento. Una aplicación que brinde la posibilidad de realizar estos mismos pedidos y que, además, confiera cierta seguridad a ambas partes, es la motivación principal de este trabajo.

Objetivos

Así, pues, el objetivo principal que se ha perseguido afrontando este proyecto ha sido la creación de una aplicación de comercio electrónico que posibilite la realización de pedidos a un establecimiento hostelero. Con ello no se pretende suplir, sino mas bien complementar, los canales existentes para realizar este tipo de operaciones (la telefónica o la de personarse en el establecimiento).

Para afrontar este reto la tecnología JEE puede ser muy útil, ya que actualmente se ha convertido en un estándar en el desarrollo de aplicaciones web empresariales. La idea inicial ha sido la utilización de una parte importante de JEE, la tecnología Servlet/JSP.

Al tratarse de una tecnología prácticamente desconocida para mi, se ha tenido que dedicar gran parte del tiempo disponible para el desarrollo del proyecto, al estudio de los diferentes componentes y herramientas existentes en el mercado para ello.

Finalmente, se ha pretendido obtener un producto que fuese robusto, de fácil crecimiento y mantenimiento, y que los cambios y mejoras futuros no obligasen a recomponer toda la aplicación. Por ello se ha querido utilizar el patrón de diseño para Servlets y JSP (Java Server Pages) más extendido, el

Modelo 2, también llamado Model-View-Control (MVC), popularizado por Jakarta Struts Framework.

Enfoque y método seguido

En primer lugar, se han buscado los componentes de la tecnología J2EE que pudiesen ser utilizados para afrontar con éxito un reto de estas características. Tras profundizar en el estudio de JEE, se ha optado, como se indicó anteriormente, en utilizar la tecnología de los Servlets y JSPs, ideal para este tipo de aplicaciones web.

A continuación, se ha visto la necesidad de utilizar patrones de diseño, que nos ofrecen como ventaja principal, el de reducir el tiempo de desarrollo, ya que ayudan a descomponer un sistema complejo en partes más manejables, a la vez que facilitan el posterior mantenimiento de la aplicación sobre todo cuando acaba teniendo unas dimensiones considerables.

Las mejores prácticas en este campo aconsejan utilizar el patrón de diseño MVC, mencionado antes. Este nos facilita una clara separación entre las capas intervinientes en una aplicación web, esto es, la de modelo de negocio (Model), la de presentación o vista (View) y la del controlador (Controller). Para ello, se ha optado por el Struts Framework que se ha convertido en un entorno ideal para el desarrollo de aplicaciones web que utilizan el MVC.

Además, también se ha visto conveniente utilizar el patrón Data Access Object (DAO), patrón que encapsula no sólo las fuentes de datos sino también la forma en la que accedemos a ellas.

Para poder utilizar estas tecnologías se ha tenido que seleccionar las herramientas y los componentes necesarios para ser instalados en el ordenador. Finalmente, tras instalar varias versiones de diferente software libre, la configuración final ha quedado como sigue:

- Eclipse, versión 3.3.2 como entorno de programación para Java, con diferentes plugins (los Web Tools Platform (WTP), los Data Tools Platform (DTP), y, sobre todo, JBossTools 2.1.2GA-ALL que ha permitido utilizar Struts 1.1 y JSP 2.3).
- Apache Tomcat, versión 5.5, como contenedor de servlets y servidor web.

- MySQL Server 5.0, como sistema de gestión de base de datos relacional. Para ayudar en la administración e interacción con la base de datos se ha utilizado MySQL Administrator y MySQL Query Browser.
- Driver mysql-connector-java 5.1.6.
- Macromedia Dreamweaver 8 como editor de páginas web.

Planificación

El TFC se ha planificado siguiendo las fases habituales del ciclo de vida del software (planificación, especificación y análisis, diseño, implementación y testing, y documentación), marcadas por las fechas de entrega de las diferentes PACs.

Fase planificación

1. Planificación	13 días	17/09/08	1/10/08
1.1. Presentaciones y estudio plan docente	3 días	17/09/08	20/09/08
1.2. Elaboración del plan de trabajo	11 días	21/09/08	30/09/08
1.2.1. Descripción general del proyecto	1 día	21/09/08	21/09/08
1.2.2. Objetivos	2 días	22/09/08	23/09/08
1.2.3. Descripción general funcionalidades	5 días	24/09/08	28/09/08
1.2.4. Planificación	2 días	29/09/08	30/09/08
1.3. Entrega Plan de trabajo (PAC 1)	1 día	1/10/08	1/10/08

Fase especificación y análisis

2. Especificación y Análisis	16 dies	1/10/08	16/10/08
2.1. Descripción detallada del projecte	1 día	1/10/08	1/10/08
2.2. Composición software	2 días	2/10/08	3/10/08
2.3. Descripción precisa funcionalidades	4 días	4/10/08	7/10/08
2.3.1. Subsistema Identificación	1 día	4/10/08	4/10/08
2.3.2. Subsistema Productos	1 día	5/10/08	5/10/08
2.3.3. Subsistema Gestión Pedidos	1 día	6/10/08	6/10/08
2.3.4. Subsistema Gestión Usuarios	1 día	7/10/08	7/10/08
2.4. Casos de uso: Identificación, relaciones y	8 días	8/10/08	15/10/08
2.4.1. Subsistema Identificación	2 días	8/10/08	9/10/08
2.4.2. Subsistema Productos	2 días	10/10/08	11/10/08
2.4.3. Subsistema Gestión Pedidos	2 días	12/10/08	13/10/08
2.4.4. Subsistema Gestión Usuarios	2 días	14/10/08	15/10/08
2.5. Glosario	1 día	16/10/08	16/10/08

Fase diseño

3. Diseño	15 días	17/10/08	31/10/08
3.1. Diagrama de clases de entidad (fichas CRC)	4 días	17/10/08	20/10/08
3.2. Diagramas de estado, secuencia y colaboración más	6 días	21/10/08	26/10/08
3.3. Diagrama de persistencia	2 días	27/10/08	28/10/08
3.4. Interficies gráficas	2 días	29/10/08	30/10/08
3.5. Consideraciones finales y entrega PAC 2	1 días	31/10/08	31/10/08

Fase implementación y testing

4. Implementación y Testing	46 días	02/11/08	17/12/08
4.1. Implementación	40 días	2/11/08	11/12/08
4.1.1. Subsistema Identificación	9 días	2/11/08	10/11/08
4.1.2. Subsistema Productos	9 días	11/11/08	19/11/08
4.1.3. Subsistema Gestión Pedidos	9 días	20/11/08	28/11/08
4.1.4. Subsistema Gestión Usuarios	9 días	29/11/08	7/12/08
4.1.5. Puesta a punto	4 días	8/12/08	11/12/08
4.2. Testing	6 días	12/12/08	17/12/08
4.2.1. Subsistema Identificación	1 día	12/12/08	12/12/08
4.2.2. Subsistema Productos	1 día	13/12/08	13/12/08
4.2.3. Subsistema Gestión Pedidos	1 día	14/12/08	14/12/08
4.2.4. Subsistema Gestión Usuarios	1 día	15/12/08	15/12/08
4.2.5. Puesta a punto y entrega PAC 3	2 días	16/12/08	17/12/08

Fase documentación

5. Documentación	28 días	17/12/08	14/01/09
5.1. Memoria TFC	16 días	17/12/08	2/01/09
5.2. Presentación TFC	10 días	3/01/09	12/01/09
5.3. Entrega final	2 días	13/01/09	14/01/09

Se ha intentado, y creo que conseguido, cumplir con la planificación prevista. La verdad es que ha faltado tiempo para poder repasar y pulir adecuadamente la finalización de cada una de las fases. Sobre todo en la fase de implementación, se ha trabajado a contrarreloj. No se ha podido, por falta de tiempo y experiencia, realizar el diseño web deseado. Finalmente se ha optado por dar prioridad al correcto funcionamiento de la aplicación, y a los objetivos propuestos para la aplicación: robustez, reusabilidad y escalabilidad.

Producto obtenido

El producto final que se ha obtenido ha sido una aplicación, el Restaurant-OnLine, que permite, por el lado cliente, dar a conocer los productos elaborados por un establecimiento hostelero a la vez que ofrece la posibilidad de hacer pedidos online, previo registro, y, por el lado empresa, gestionar los pedidos recibidos y administrar el sitio web. Para ello se distinguen cuatro tipos de usuarios:

- visitante, que puede acceder al restaurante y conocer sus especialidades, así como añadir artículos a la nota de pedido (similar al típico carrito de la compra), que finalmente, para ser tramitado necesita de un registro y de un acceso con nombre de usuario y contraseña.
- cliente, usuario que ya está registrado, y puede acceder al restaurante previa introducción de su nombre de usuario y contraseña, para realizar pedidos o consultar los ya realizados.
- empleado, usuario que también se le pide identificación, y que puede gestionar los pedidos y artículos presentes en la web.
- administrador, figura que, además de las funcionalidades inherentes al empleado, podrá gestionar los usuarios del sistema.

Breve descripción de los otros capítulos de la memoria

En los siguientes capítulos de la memoria se pretende dar una visión detallada del desarrollo de la aplicación Restaurant-OnLine. En el capítulo 2 se explica el funcionamiento general de la aplicación a través de describir los componentes de JEE utilizados a tal efecto. En el capítulo 3 se analizan con cierto detalle las funcionalidades implementadas. Las clases que se utilizan para el diseño son especificadas en el capítulo 4. En el siguiente capítulo se pretende con ayuda de varias capturas de pantalla describir el funcionamiento práctico de Restaurant-OnLine. Finalmente, los últimos capítulos lo componen las conclusiones, así como un glosario y la bibliografía utilizada.

Capítulo 2 Funcionamiento del sistema

JEE es una plataforma para ejecutar aplicaciones Java del lado servidor. Actualmente es muy utilizada a nivel empresarial para la creación de aplicaciones web. Una parte importante de esta tecnología la componen los siguientes componentes web: Servlets y JSP (Java Server Pages). En las arquitecturas cliente-servidor, el papel principal que realizan estos componentes es el de recibir peticiones del lado cliente (que vienen en la forma de peticiones HTTP) para posteriormente generar contenidos dinámicos o realizar alguna lógica de negocio, y devolver la respuesta al cliente.

Struts es un framework de código abierto usado para construir aplicaciones web en Java, basado en el patrón de diseño MVC y diseñado utilizando principalmente las tecnologías Java-Server Pages (JSP) y Servlets. Como ya se ha mencionado más arriba, la arquitectura del Modelo 2, MVC, facilita el desarrollo, el mantenimiento y la escalabilidad de la aplicación. Esto es así, principalmente, porque introduce entre el navegador y las JSP un servlet controlador que se encarga de atender las peticiones del lado cliente, como se muestra en la figura siguiente:

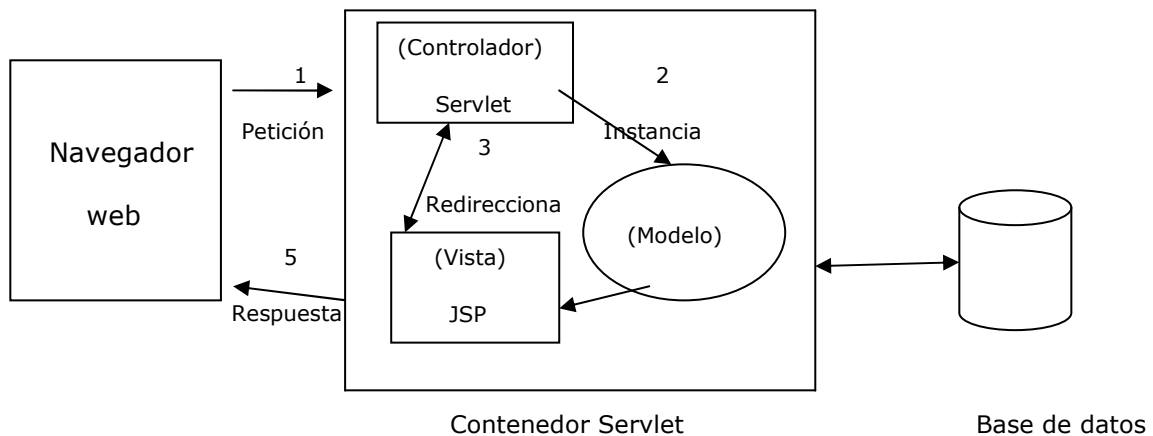


Figura 2.1: Arquitectura MVC-Modelo 2.

Struts dispone de un controlador central, que una vez inicializado, analiza un fichero de configuración (struts-config.xml) y lo utiliza para desplegar otros objetos de la capa de control. La Configuración Struts define, principalmente, los ActionMappings (mapeos de acciones) para la aplicación.

El Servlet Controlador de Struts consulta estos ActionMappings y envía las peticiones HTTP a otros componentes del framework, ya sean JavaServerPages u otras Action, atendiendo, así, las peticiones y

devolviendo la respuesta al cliente (generalmente un navegador web). Los mapeos ayudan al controlador a convertir las peticiones en acciones de la aplicación. Cada acción actúa como un adaptador entre la petición y la capa de modelo. Una vez la petición es tratada, el flujo vuelve al controlador y éste devuelve los resultados a una vista asignada, proporcionando, de esta forma, un ligero acoplamiento entre la capa Vista y la capa Modelo.

La siguiente figura muestra una visión general de la arquitectura Struts:

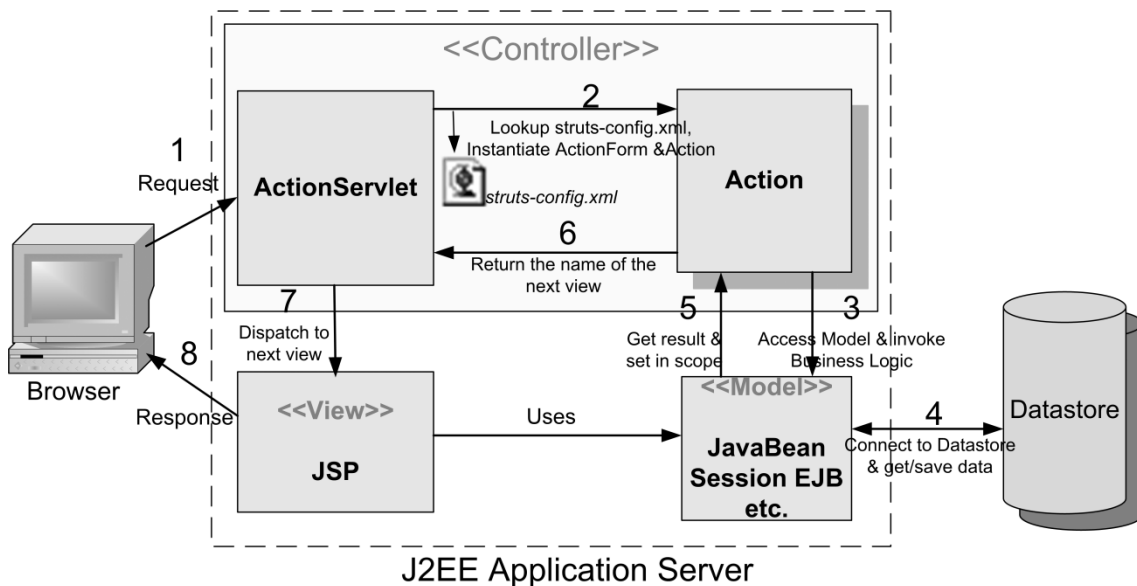


Figura 2.2: Arquitectura Struts

La aplicación de este TFC intenta hacer uso de prácticamente toda la tecnología inherente de Struts, puesto que utiliza:

- ❖ ActionForms , que representan los formularios HTML , usados para transferir los datos entre las capas Vista y Controlador.
- ❖ Acciones, tanto únicas (Action) como comunes (DispatchAction).
- ❖ ActionMappings que mapean las características de las acciones en el fichero xml central de Struts, struts-config.xml.
- ❖ Tiles Framework, que permite la creación de páginas web ensamblando diferentes tiles o plantillas.
- ❖ Struts Tags, librería de etiquetas propia de Struts.
- ❖ AccionsErrors para indicar los errores en la validación de formularios ActionForm.
- ❖ Validation Framework, que evita tener que crear nuestros propios métodos de validación de campos en formularios.
- ❖ Contenedor, que nos guarda toda la información de la sesión.
- ❖ Servicios, para conectarnos a la base de datos, en nuestro caso mediante un pool de conexiones.

También se ha intentado hacer uso de varias de las mejores prácticas en la implementación de Struts, como son:

- Creación de acciones comunes
- Evitar el doble "form submission", por ejemplo, en la página de registro, que puede ocurrir al hacer más de un click en aceptar antes de ser enviada la respuesta desde el servidor o al volver a la página desde los botones atrás o actualizar.
- uso de una sola ActionForm para formularios similares.
- evitar incluir lógica de negocio en los ActionForm.
- uso de patrones de diseño como Business Delegate, delegando a la capa de modelo las funcionalidades propias de esta capa, en lugar de implementarlo en las acciones (Action).
-

Componentes de Restaurant-OnLine

A continuación describiré con cierto detalle los diferentes componentes anteriormente mencionados que utiliza la aplicación.

Base de datos (capa persistencia)

Se utiliza una base de datos relacional implementado sobre el servidor MySQL. La conexión a esta base de datos es gestionada por el propio servidor Tomcat, así que nuestra aplicación únicamente instanciará una conexión existente.

Capa Modelo o lógica de negocio (Model)

TransferObject o ValueObject

El TransferObject encapsula una unidad de información de la fuente de datos. Viene a ser un bean de tabla (JavaBean) que, como tal, representa una tabla de la base de datos. Se utilizan los siguientes TransferObjects, con sus correspondientes getters y setters:

- BaseVO, una superclase de la cual descenderán otros.
- User para representar un usuario.
- Artículo para representar un artículo.
- Categoría que representa una categoría de artículos.
- Comanda para representar un pedido.

- `LiniaComanda` que representa el detalle del pedido, los artículos del cual se compone.

Además, disponemos de unas clases auxiliares que nos sirven para mostrar diferentes listas desplegables en nuestras páginas JSP, como son:

- `ListaCategoria` para poder acceder a la lista de categorías de artículos.
- `Hora`, clase que nos sirve para mostrar las horas de recepción del pedido.
- `Fecha`, análoga a la anterior.

Servicio

Para implementar el servicio que conectará con la base de datos y que contenga todos los métodos de negocio, se dispone de una interficie, `ICarritoService`. Esta será utilizada por parte del cliente como único nexo de unión con la lógica de negocio, y desde ella podrá invocar los diferentes métodos que la componen.

Los métodos implementados en `CarritoServiceImpl` serán los siguientes:

- `init()` que retorna la conexión con la base de datos, que se llama sólo una vez al instanciar la clase.
- `insertUser(user)`, que añade a la base de datos el usuario pasado como parámetro.
- `getListaUsuario(tipoUsuario)`, que devuelve una lista de un tipo de usuarios, pasado como parámetro.
- `findUser(nomUsuario)`, que devuelve un usuario a partir de su nombre de usuario.
- `findUserEmpresa(nomUsuario)`, que devuelve un usuario del tipo empleado o administrador a partir de su nombre.
- `validar(usuario, contraseña)`, que comprueba que el usuario existe y que la contraseña corresponde al usuario.
- `userUpdate(user)`, que actualiza los datos de un usuario con los datos nuevos pasados como parámetro.
- `deleteUser(nomUsuario)`, que borra de la base de datos el usuario cuyo nombre es pasado como parámetro.
- `findArticulo(identificador)`, que devuelve un artículo a partir de su identificador.
- `getListaArticulos(categoría, visible)`, que devuelve la lista de artículos de la categoría indicada como parámetro, y según se indique todos o sólo los que sean visible en la web.
- `getListaCatArt()`, que retorna la lista de categorías de artículos existentes.

- `articuloUpdate(articulo)`, que actualiza los datos de un artículo con las datos nuevos pasados como parámetro.
- `insertArticulo(articulo)`, que añade a la base de datos el artículo pasado como parámetro.
- `deleteArticulo(código)`, que borra de la base de datos el artículo cuyo identificador es pasado como parámetro.
- `insertComanda(comanda)`, que añade a la base de datos el pedido pasado como parámetro.
- `insertDetallePedido(pedido)`, que añade a la base de datos el detalle del pedido pasado como parámetro.
- `ultimoIdGenerado()`, que devuelve la clave primaria del último pedido introducido en la base de datos.
- `getListPedClienPend(nomUsuario)`, que devuelve una lista de los pedidos pendientes del usuario cuyo nombre se pasa como parámetro.
- `getPedClienPendDetall(identificador)`, que retorna el detalle del pedido pasado como parámetro.
- `getListPedidos(nomUsuario)`, que devuelve la lista de pedidos del usuario cuyo nombre pasamos como parámetro.
- `findIdComanda(identificador)`, que devuelve el pedido cuyo identificador pasamos como parámetro.
- `getListEstadoPed(estado)`, que retorna la lista de pedidos que se hayan en el estado que pasamos como parámetro.
- `getListFechaPed(inicio, fin)`, que devuelve una lista de pedidos que se hallan entre las fechas indicadas como parámetros.
- `admitirPedido(identificador)`, que pasa el pedido cuyo identificador se indica como parámetro del estado pendiente al estado admitido.
- `getListPedAnulables(nomUsuario)`, que devuelve una lista de pedidos pertenecientes al usuario cuyo nombre pasamos como parámetro, que se encuentran en estado pendiente o admitido.
- `anularPedido(identificador)`, que deja el pedido cuyo identificador se pasa como parámetro en estado anulado.
- `cambioContraseña(nombre, nuevaContraseña)`, que modifica la contraseña perteneciente al usuario cuyo nombre pasamos como parámetro.
- `articuloPedido(identificador)`, que indica si el pedido cuyo identificador es pasado como parámetro, se encuentra en los estados pendiente o admitido.

También disponemos de una interficie factoria de servicios y la clase que la implementa que permitirá al usuario crear el servicio anterior y hacer uso de él. La factoría de servicios se implementa a través de la posibilidad de extensión y personalización que ofrece Struts. Para la extensión se utiliza, aquí, el mecanismo de plug-in, esto es, una clase Java que se inicializa

cuando la aplicación empieza a ejecutarse y se destruye cuando la aplicación finaliza. El plug-in necesita declararse en el fichero de configuración de Struts bajo el elemento <plug-in>.

Capa Controlador (Controller)

La capa controlador forma parte del entorno Struts, por lo que haremos uso de su Api. En Struts sólo existe un componente controlador: el servlet ActionServlet. Existe una única instancia de este servlet por cada aplicación web y su función es la de procesar todas las peticiones que afectan al modelo de la aplicación. Para cada petición, el ActionServlet selecciona e invoca a la clase Action encargada de ejecutar la lógica de negocio asociada.

Veamos algunas las clases fundamentales que componen esta capa.

La clase org.apache.struts.action.ActionMapping

Un ActionMapping representa la información que el ActionServlet conoce acerca de una asignación-petición Action concreta. Esta información es pasada al método execute() de la clase Action, permitiendo, de esta manera, que esta clase pueda acceder directamente a esta información.

La clase org.apache.struts.action.Action

Un Action es un adaptador entre el contenido de una petición HTTP y la lógica de negocio que debería ejecutarse para procesar dicho contenido. Para cada petición, el ActionServlet selecciona la clase Action apropiada, crea una instancia de la misma (si no existe) y llama al método execute().

La clase org.apache.struts.action.ActionForward

Un objeto ActionForward identifica dónde debería reenviar el control el ActionServlet (por ejemplo a una JSP) para proporcionar la respuesta apropiada. La clase Action puede crear dinámicamente instancias de la clase ActionForward o utilizar las contenidas en el objeto ActionMapping.

En esta aplicación, para personalizar esta capa se ha extendido la clase org.apache.struts.action.Action creando una clase abstracta CustomBaseAction para disponer cada vez que invoquemos una acción en Struts una instancia del servicio que hemos implementado en la capa de modelo, almacenado en el contexto del ActionServlet. Lo mismo hacemos con la clase org.apache.struts.action. DispatchAction en CustomDispatchAction

Los métodos más importantes son los que crean el servicio para el cliente y el que permite obtener el contenedor del usuario con los datos de la sesión o, en caso de no existir todavía, crear este contenedor. La clase que implementa este contenedor es `UserContainer`, que contendrá la nota de pedido y, en caso de haber iniciado sesión, también la información del usuario.

Para trabajar con la nota de pedido en memoria, es decir, sin hacerlo persistente, lo incluimos dentro del contenedor. Para ello, utilizamos las clases `Pedido` y `ArticuloCestaPedido`. El primero implementa una lista que guardará los diferentes artículos que componen la nota de pedido, con sus métodos principales de añadir, borrar, actualizar u obtención del importe total. Y el segundo representará el artículo de la cesta, con sus métodos getters y setters correspondientes.

Capa presentación o Vista (View)

Es en la capa de presentación donde utilizaremos el potencial de Struts. Una vez implementadas todas las acciones de Struts, nos queda elaborar la parte de la interfaz con la que trabajará el usuario. Para ello, se utiliza la tecnología JSP. Además se utilizará parte de la librería de etiquetas, `Tags`, que nos proporciona Struts, esto es:

- `struts-html`
- `struts-logic`
- `struts-bean`
- `struts-tiles`

que nos permitirán realizar tareas relacionadas con el diseño HTML, la lógica (iteraciones, condiciones, etc) y trabajar con los beans en los `ActionForms`.

Se ha intentado, ante todo, evitar el uso de scriplets para no incluir código java embebido dentro de las páginas JSP. De esta forma, obtendremos un código más limpio y legible, a la vez que evitaremos que el creador de las páginas de presentación deba tener grandes conocimientos de programación.

También se ha utilizado, como se ha mencionado ya, el paquete `Tiles` de Struts para trabajar con plantillas de forma más fácil, y evitar que modificaciones en el diseño obliguen a hacer cambios en cada una de las páginas JSP afectadas. Para ello se utiliza el archivo `xml tiles-defs`, propio de Struts.

Finalmente, se ha de mencionar que se ha hecho uso del `Validation Framework` para la validación de formularios. Esto ha evitado tener que escribir código Java e implementar el método `validar()` en cada `ActionForm`. La lógica de la validación está configurada en una serie de

reglas de validación predefinidas en el archivo validator-rules.xml. Luego, en otro archivo, en validation.xml, se indica qué reglas serán utilizadas para cada ActionForm en particular.

En prácticamente toda la aplicación, se ha usado la validación del lado servidor, en contraposición a la del lado cliente. De esta forma, se ha evitado la incorporación de java-script, haciendo el sitio web más amigable y más acorde con las normas de accesibilidad de páginas web.

Capítulo 3 Análisis de las funcionalidades del sistema

Para facilitar el desarrollo del proyecto, se han distribuido el conjunto de funcionalidades de la aplicación en cuatro subsistemas:

- Subsistema de Identificación
- Subsistema de gestión de artículos
- Subsistema de gestión de pedidos
- Subsistema de gestión de usuarios

Descripción de las funcionalidades

Identificación

Esta funcionalidad permitirá la identificación del usuario como cliente, empleado o administrador.

El usuario introducirá su nombre de usuario y contraseña y el sistema validará estos datos. En el caso que el usuario esté dado de alta en el sistema y no tenga bloqueado el acceso, se le permitirá entrar. En caso contrario se le advertirá que sus datos no son correctos.

Además este subsistema incluirá una utilidad para efectuar el cambio de contraseña. Para ello se le pedirá que introduzca el nombre de usuario y la contraseña actual y, posteriormente, que introduzca dos veces la contraseña nueva, que deberá ser diferente a la actual.

Gestión de artículos

Consultar artículos

El visitante o el cliente dispondrán en el sitio web de unos enlaces que le permitirán consultar la lista de artículos que ofrece el restaurante accediendo a la categoría o familia de artículos específica.

En el caso de los empleados o administradores, éstos también podrán acceder desde su página de inicio a esta utilidad.

Dar de alta, de baja o modificar artículos

Estas funcionalidades serán exclusivas de los usuarios empleado y administrador, que les permitirá llevar un control de los artículos y de sus características en cada momento.

Gestión de pedidos

Realizar pedidos

El cliente tendrá la posibilidad de realizar pedidos. Para ello dispondrá de su 'nota de pedido', al estilo cesta de compra, que le permitirá visualizar en cada momento los artículos que ha elegido

Una vez la nota de pedido esté completada deberá tramitar el pedido, que será definitivamente aceptado una vez haya indicado los datos de envío, así como el día y hora deseada para su recepción.

Consultar pedidos

Por un lado, el cliente podrá consultar desde su navegador los pedidos que tiene tramitados y todavía están en estado pendiente, es decir, no admitidos por el restaurante.

Por otro lado, el empleado y administrador también podrán consultar los pedidos que se hallan en la base de datos. Para ello podrán filtrar la consulta seleccionando uno de los 4 criterios de búsqueda siguientes: el estado en el que se puede encontrar un pedido (esto es, pendiente, rechazado, admitido, o servido), el identificador de pedido, el nombre de usuario o rango de fechas.

Admitir pedidos

Los pedidos, una vez realizados por el cliente, son recibidos en la empresa. Los empleados o administradores deberán comprobar el pedido entrante y, una vez comprobado que no presenta ninguna irregularidad, proceder a su admisión. Internamente la empresa podrá, por ejemplo, imprimir el pedido y pasarlo a cocina para su elaboración. El pedido pasará de pendiente a admitido.

Anular pedidos

Esta opción, sólo accesible a empleados o administrador, permitirá pasar un pedido al estado de anulado, sin que ello suponga darlo de baja de forma definitiva de la base de datos.

Gestión de usuarios

Bajo este subsistema también se encuentran funcionalidades que afectan tanto el lado cliente como el lado empresa.

Registro

Del lado cliente, tenemos la utilidad del registro, trámite necesario para estar dado de alta como cliente y poder efectuar pedidos. Al usuario se le pedirán datos personales, datos de domicilio necesarios para los envíos y datos de acceso a la página (nombre de usuario y contraseña).

Consulta, alta, baja y modificación de usuarios

Del lado empresa dispondremos de las utilidades de consulta, alta, baja y modificación de usuarios.

Los empleados y administradores podrán consultar los usuarios existentes atendiendo a su nombre o al tipo de usuarios al que pertenece (cliente, empleado o administrador).

Únicamente a los administradores se les permite dar de alta a nuevos usuarios o modificar los datos de los mismos, y siempre que no sean de tipo cliente. En cambio, en las bajas no habrá esta limitación.

Casos de uso

De las funcionalidades anteriormente descritas se han elaborado los siguientes casos de uso, que también quedarán agrupados para mayor claridad según al subsistema al que pertenecen.

Subsistema de Identificación

Nombre:	Identificación
Funcionalidad general:	Verifica que el nombre de usuario y la contraseña que se haya introducido correspondan a algún usuario dado de alta en el sistema.
Actores:	Cliente, empleado y administrador.
Descripción:	El sistema solicita un nombre de usuario y una contraseña,

	comprueba que el usuario existe y que la contraseña es válida.
Casos de uso relacionados:	Cambio de contraseña

Nombre:	Cambio de contraseña
Funcionalidad general:	Permite que el usuario cambie su contraseña.
Actores:	Cliente, empleado y administrador.
Descripción:	El sistema solicita un nombre de usuario y una contraseña actual, comprueba que el usuario existe y que la contraseña es válida. Entonces solicita que se introduzca dos veces la nueva contraseña, para registrar el cambio.
Casos de uso relacionados:	Identificación.

Subsistema de gestión de artículos

Nombre:	Consultar artículos
Funcionalidad general:	Permite consultar los artículos que tiene la empresa a la venta.
Actores:	Visitante, cliente, empleado y administrador.
Descripción:	La aplicación realizará una consulta al sistema para mostrar en pantalla todos los artículos de la categoría seleccionada por el visitante o cliente, o, en caso de empleado o administrador, los artículos o artículo según el criterio de consulta seleccionado.
Casos de uso relacionados:	

Nombre:	Alta artículo
Funcionalidad general:	Permite dar de alta un nuevo artículo en el sistema.
Actores:	Empleado y administrador.
Descripción:	El usuario introduce un identificador de artículo y demás datos solicitados, y el sistema, tras comprobar que el identificador no existe, procede a incorporar el artículo a la base de datos.
Casos de uso relacionados:	

Nombre:	Baja artículo
Funcionalidad general:	Permite dar de baja un artículo existente en el sistema.
Actores:	Empleado y administrador.
Descripción:	El usuario introduce el identificador de artículo que desea borrar de la base de datos, y, tras una pantalla de confirmación, se procede a dar de baja el artículo deseado.
Casos de uso relacionados:	

Nombre:	Modificación artículo
Funcionalidad general:	Permite modificar un artículo existente en el sistema.
Actores:	Empleado y administrador.
Descripción:	El usuario introduce el identificador de artículo que desea modificar, tras lo cual podrá visualizar los detalles del mismo y proceder a modificarlo.
Casos de uso relacionados:	

Subsistema de gestión de pedidos

Nombre:	Realizar pedido
Funcionalidad general:	Permite realizar un pedido.
Actores:	Cliente.
Descripción:	El usuario, una vez incorporados los artículos a la nota de ha de confirmar éste para su tramitación. El sistema permitirá al usuario seleccionar la hora y día de envío y modificar, en su caso, la dirección de envío y/o teléfono de contacto.
Casos de uso relacionados:	Crear nota de pedido.

Nombre:	Crear nota de pedido
Funcionalidad general:	Permite crear una nota de pedido.
Actores:	Visitante y cliente.
Descripción:	El usuario selecciona los artículos deseados desde las diferentes categorías de artículos disponibles y los va incorporando a la nota de pedido.
Casos de uso relacionados:	Consultar artículos.

Nombre:	Consulta nota de pedido
Funcionalidad general:	Permite consultar los artículos de la nota de pedido.
Actores:	Visitante y cliente.
Descripción:	El usuario podrá visualizar en cualquier momento su nota de pedido permitiéndole borrar y añadir artículos o cambiar la cantidad de éstos.
Casos de uso relacionados:	Crear nota de pedido.

Nombre:	Consulta pedido
Funcionalidad general:	Permite consultar los pedidos que se hallan en la base de datos.
Actores:	Cliente, empleado y administrador.
Descripción:	En el caso del cliente, éste accederá al enlace 'Mis pedidos' y el sistema le mostrará los pedidos que tiene pendientes de envío. En el caso de los empleados y administrador, éstos podrán realizar la consulta según el criterio de búsqueda seleccionado (identificador, nombre de usuario, fecha o estado).
Casos de uso relacionados:	

Nombre:	Admitir pedido
Funcionalidad general:	Permite cambiar el estado de un pedido, pasándolo de pendiente a admitido.
Actores:	Empleado y administrador.
Descripción:	El usuario selecciona el pedido entrante, que se encuentra en estado pendiente, y, tras consultar los detalles, si verifica que todo es correcto y da el pedido como admitido.
Casos de uso relacionados:	Realizar pedido.

Nombre:	Anular pedido
Funcionalidad general:	Permite cambiar el estado de un pedido, pasándolo de pendiente o admitido a anulado.
Actores:	Empleado y administrador.
Descripción:	El usuario introduce el identificador de artículo que desea anular, tras lo cual podrá visualizar los detalles del mismo y proceder a anularlo. El

	sistema no dará de baja el pedido en la base de datos, sino sólo lo marcará como anulado.
Casos de uso relacionados:	Consulta pedido.

Subsistema de gestión de usuarios

Nombre:	Registro
Funcionalidad general:	Permite que el usuario se registre en el sistema.
Actores:	Visitante.
Descripción:	El usuario accede a la utilidad de registro he introduce los datos solicitados. El sistema da de alta al nuevo cliente, tras comprobar que todos los datos introducidos cumplen los criterios de validación de formato necesarios, que todos los campos de obligado relleno no están vacíos y que el usuario no existe en el sistema.
Casos de uso relacionados:	

Nombre:	Alta usuario
Funcionalidad general:	Permite dar de alta a un nuevo empleado o administrador en el sistema.
Actores:	Administrador.
Descripción:	El usuario introducirá un nombre y los demás datos solicitados. En el caso que el nombre de usuario no exista, el sistema incorporará el nuevo usuario a la base de datos.
Casos de uso relacionados:	

Nombre:	Baja usuario
Funcionalidad general:	Permite dar de baja a un usuario existente en el sistema.
Actores:	Administrador.
Descripción:	Se introducirá el nombre de usuario que se desee dar de baja. El sistema, tras solicitar confirmación, procederá a borrarlo de la base de datos.
Casos de uso relacionados:	

Nombre:	Modificar usuario
Funcionalidad general:	Permite modificar los datos de un usuario existente en el sistema
Actores:	Administrador.
Descripción:	Se introducirá el nombre de usuario cuyos datos se deseen modificar. El sistema mostrará los datos de este usuario con los campos susceptibles de ser modificados, y, tras los cambios realizados, procederá a guardarlos.
Casos de uso relacionados:	

Nombre:	Consultar usuario
Funcionalidad general:	Permite consultar los datos de un usuario existente en el sistema.
Actores:	Empleado y administrador.
Descripción:	La consulta podrá realizarse según el tipo de usuario o introduciendo un nombre de usuario.
Casos de uso relacionados:	

Diagramas de casos de uso

A continuación se mostrarán los diagramas de casos de uso detallados según el subsistema al que pertenecen.

Casos de uso del subsistema de identificación

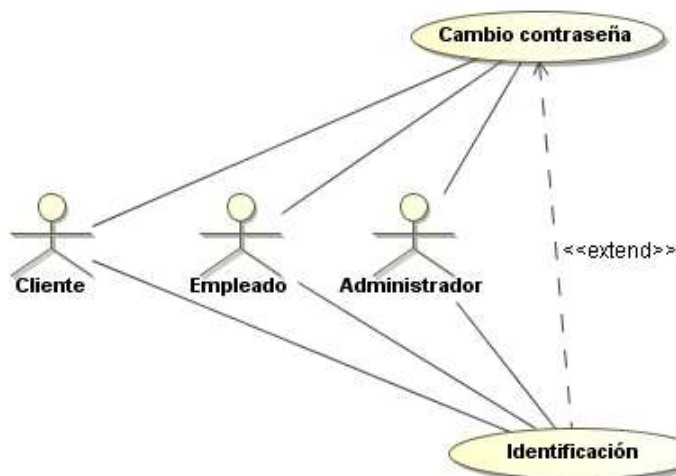


Figura 3.1: Diagrama de casos de uso del subsistema de identificación

Casos de uso del subsistema de gestión de artículos

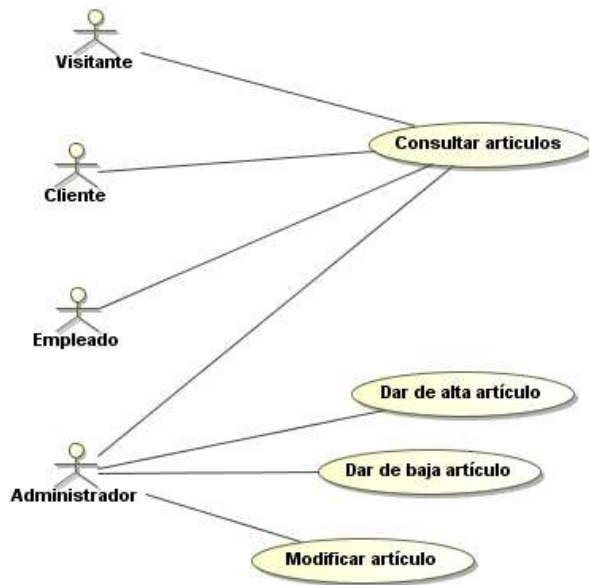


Figura 3.2: Diagrama de casos de uso del subsistema de gestión de artículos

Casos de uso del subsistema de gestión de pedidos

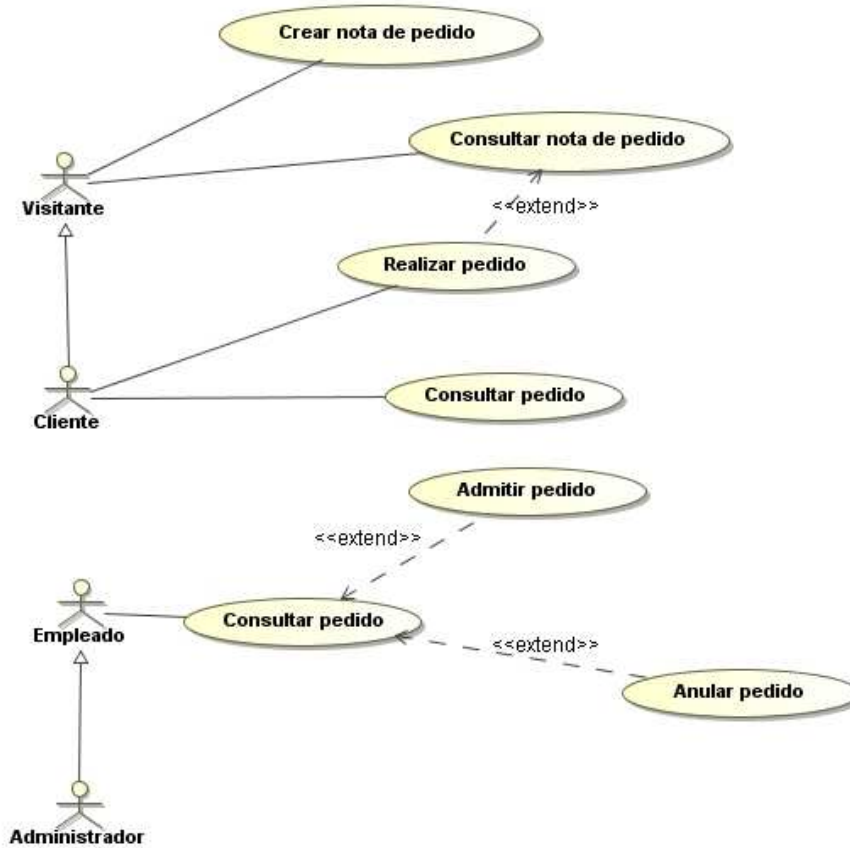


Figura 3.3: Diagrama de casos de uso del subsistema de gestión de pedidos

Casos de uso del subsistema de gestión de usuarios**Figura 3.4:** Diagrama de casos de uso del subsistema de gestión de usuarios

Capítulo 4 Diseño

A continuación se explicará los pasos seguidos en el diseño. En primer lugar, se han definido las clases necesarias para la aplicación, partiendo de las funcionalidades definidas en el capítulo anterior. Se hará una presentación de las clases agrupándolas según a la capa que pertenecen, siguiendo el patrón de diseño MVC.

Capa modelo

Clases correspondientes a los Transfer Object

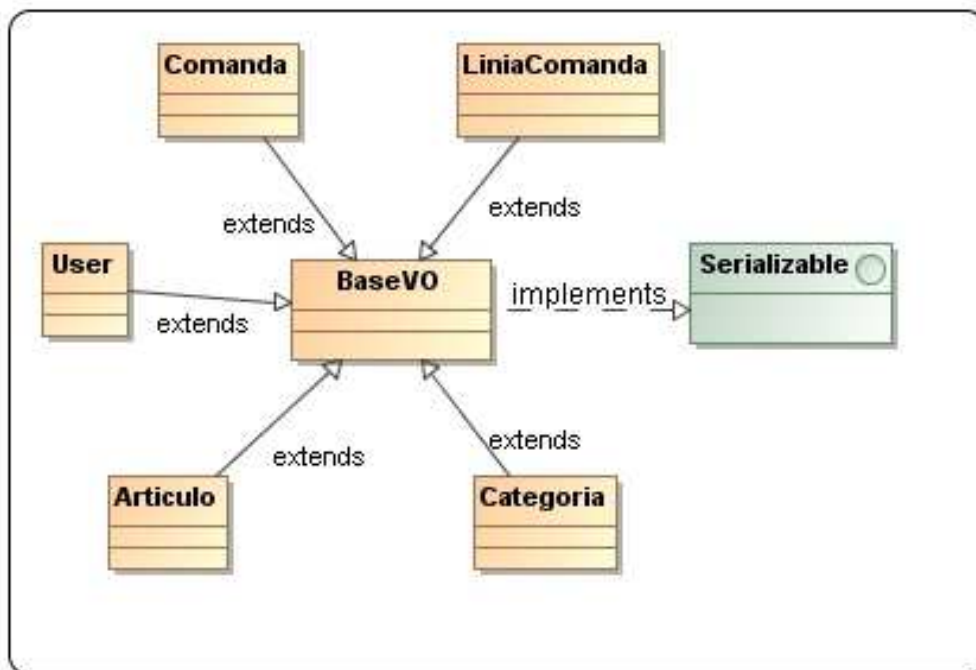


Figura 4.1: Diagrama de clases Transfer Object

Para mayor claridad no se incluyen los atributos ni las operaciones.

A continuación se muestran las clases que no hacemos persistentes, pero que necesitamos, tanto para guardar los datos de la sesión del usuario como para trabajar con la nota de pedido y los artículos que contendrán.

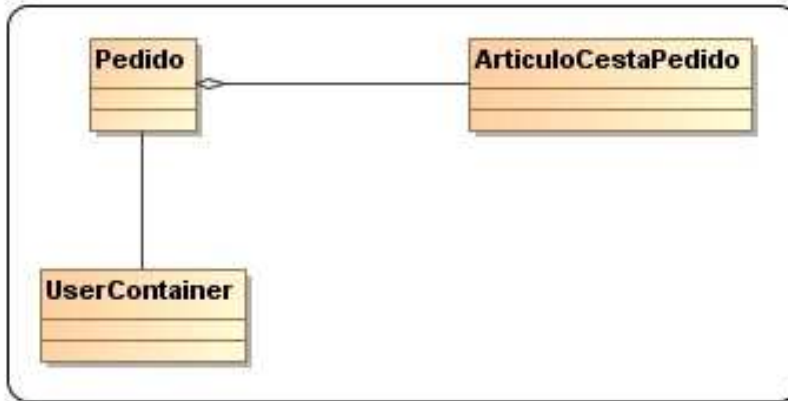


Figura 4.2: Diagrama de clases no persistentes

Clases DAO

Interficie y clase que encapsula e implementa la conexión a la base de datos y las diferentes operaciones de acceso a ella para obtener las vistas adecuadas.



Figura 4.3: Diagrama de clases e interfaces DAO

Interficie y clase permite la creación del servicio anterior.

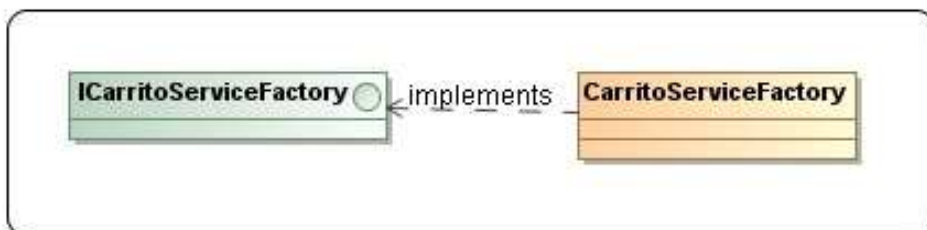


Figura 4.4: Diagrama de clases e interfaces factoría de servicio

El detalle de las operaciones se han explicado más arriba, en el apartado Componentes del Restaurante-OnLine.

Capa Controlador

Existen posturas enfrentadas en cuanto si la clase Action forma parte de la capa Controladora o del Modelo. Aquí seguimos el criterio de que forma parte de la capa controladora. Así pues, las clases que se implementan en este nivel se muestran en la siguiente figura, separando las acciones que pertenecen a la parte Cliente, a la parte Empresa y las comunes.

Clases comunes

Clase	Acción que realiza
CustomDispatchAction	Crea el servicio contenido en ICarritoService para usarlo con las acciones exclusivas de la nota de pedido
CustomBaseAction	Crea el servicio contenido en ICarritoService para usarlo con el resto de acciones
CambioContrasenaSaveAction DisplayCambioContrasenaForm	Ejecuta el cambio de contraseña Muestra el formulario para el cambio de contraseña
DisplayLoginForm	Muestra el formulario para iniciar sesión
LoginAction	Valida el inicio de sesión
LogoutAction	Ejecuta la finalización de la sesión

Clases parte Cliente

Clase	Acción que realiza
PedidoActions	Agrupar las clases usadas para la nota de pedido y tramitación del mismo.
DisplayNota	Muestra la nota de pedido
DisplayRegistrationForm	Muestra el formulario de registro
RegistrationSaveAction	Ejecuta el registro de clientes
GetListaPedClienteAction	Obtiene la lista de pedidos
GetPedClienteDetalleAction	Obtiene el detalle de un pedido

Clases parte Empresa

Gestión artículos

Clase	Acción que realiza
ArtAltaSaveAction	Ejecuta el alta de artículos
ArtBajaAction	Ejecuta la baja de artículos
ArtCategoriaAction	Obtiene las categorías de artículos
ArtModifSaveAction	Ejecuta la modificación de artículos
DisplayAltaArtCatAction	Obtiene las categorías para las altas
DisplayConArtCatAction	Obtiene las categorías para las

	consultas
GestBajaArtIdAction	Obtiene un artículo para las bajas
GestConArtCatAction	Obtiene una categoría de artículos
GestConArtIdAction	Obtiene un artículo para las consulta
GestModifArtIdAction	Obtiene un artículo para modificaciones

Gestión pedidos

Clase	Acción que realiza
AdmisionPedidoAction	Ejecuta la admisión de pedidos
DisplayGestAnulPed	Acción de redireccionamiento
GestAdmisionPedidosAction	Obtiene una lista de pedidos pendientes
GestAdmPedNomDetalle	Obtiene el detalle de un pedido para su admisión
GestAnulPedIdAction	Obtiene un pedido para su anulación
GestAnulPedNomDetalleAction	Obtiene el detalle de un pedido para su anulación
GestConPedEstAction	Obtiene una lista de pedidos según su estado
GestConPedFechaAction	Obtiene una lista de pedidos según un rango de fechas
GestConPedIdAction	Obtiene un pedido por su identificador
GestConPedIdDetalleAction	Obtiene el detalle del pedido anterior
GestConPedNomAction	Obtiene un pedido por nombre de usuario
GestConPedNomDetalleAction	Obtiene el detalle del pedido anterior
GestConsPedAction	Pasa el control a otra acción según el criterio de consulta de pedidos
PedAnulAction	Ejecuta la anulación de pedido

Gestión usuarios

Clase	Acción que realiza
DisplayGestAltaUsu	Acción de redireccionamiento
DisplayAltaUsuForm	Muestra formulario de alta
DisplayGesBajaUsu	Acción de redireccionamiento
DisplayGestConUsuAction	Acción de redireccionamiento
DisplayGestModifUsu	Acción de redireccionamiento
GestBajaUsuNomAction	Obtiene un usuario para la baja
GestConUsuAction	Pasa el control a otra acción según el criterio de consulta de usuarios
GestConUsuNomAction	Obtiene un usuario para la consulta
GestConUsuTipoAction	Obtiene una lista de usuarios según el tipo
GestConUsuTipoDetalleAction	Obtiene un usuario para la consulta
GestModifUsuNomAction	Obtiene un usuario para su

	modificación
UsuAltaSaveAction	Ejecuta el alta de usuarios
UsuBajaAction	Ejecuta la baja de usuarios
UsuModifSaveAction	Ejecuta la modificación de usuarios

Capa Vista o Presentación

Los componentes de la Vista son las páginas JSP y las clases ActionForm, que son beans asociados a los formularios de las páginas JSP.

Se definen los siguientes ActionForm:

ActionForm
AltaUsuarioForm
ArticuloForm
CambioContrasenaForm
CategoriaForm
ComandaForm
CriterioBusqueda
DateForm
LoginForm
ModifArticuloForm
RegisterForm

Las páginas JSP se agrupan, nuevamente, separando las de la parte Cliente de la parte Empresa.

Páginas JSP del lado Cliente	
contacto_body	verCategoria_1
quienSomos_body	notaPedido2_body
verProductos	confirmarPedido_body
verCategoria_1	displayPedidoConfirm_body
notaPedido2_body	cambioContrasenaCliente_body
iniciarSesion	errorLogin
registro_body	
registroError	
registroOK	
operacionOk	
error	
verProductos	
pedidosPendCliente_body	
pedidosPendClienteDetail_body	

Páginas JSP del lado empresa

cambioContrasena_body	gestPedConsIdDetalle_body
gestPedidos_body	gestPedConsNomDetalle_body
gestPedCons_body	gestPedModifNomDetalle_body
gestPedAdmision_body	gestPedConsEstadoDetalle_body
gestPedAdmNomDetalleUltimo_body	gestPedConsFechaDetalle_body
gestPedModif_body	gestPedConsNomDetalleUltimo_body
gestPedConsNom_body	gestPedAnulNomDetalleUltimo_body
gestPedModifNom_body	gestUsuarios_body
gestPedAnulId_body	gestUsuCons_body
gestPedAnulDetalle_body	gestUsuBajaNom_body
gestPedConsId_body	gestArtBajaId_body
gestPedConsEstado_body	gestUsuBajaNomDetalle_body
gestPedConsFecha_body	gestArtBajaIdDetalle_body
gestPedModifFecha_body	verSeleccion
gestUsuConsNom_body	gestUsuModifDetalle_body
gestArtConsId_body	gestArtModifDetalle_body
gestUsuAlta_body	gestUsuConsTipo_body
gestArtAlta_body	gestArtConsCat_body
gestUsuConsNomDetalle_body	gestUsuConsTipoDetalle_body
gestArtConsIdDetalle_body	gestArtConsCatDetalle_body
gestUsuConsTipoDetalleUltimo_body	gestArtCons_body
gestArticulos_body	gestUsuModifNom_body
gestArtModifId_body	usuarioError
contrasenaError	datoError
altaError	bajaOK
altaOK	altaUsuarioError
generalError	altaUsuarioOK
bajaUsuarioOK	gestUsuModifMensaje_body
errorLogin	

Capítulo 5 Funcionamiento de Restaurant-OnLine

En este capítulo se hará una explicación del funcionamiento de la aplicación, que se ilustrará con capturas de pantalla par un mejor entendimiento.

Inicio

La aplicación se inicia con la página welcome.jsp, que será su página de inicio.

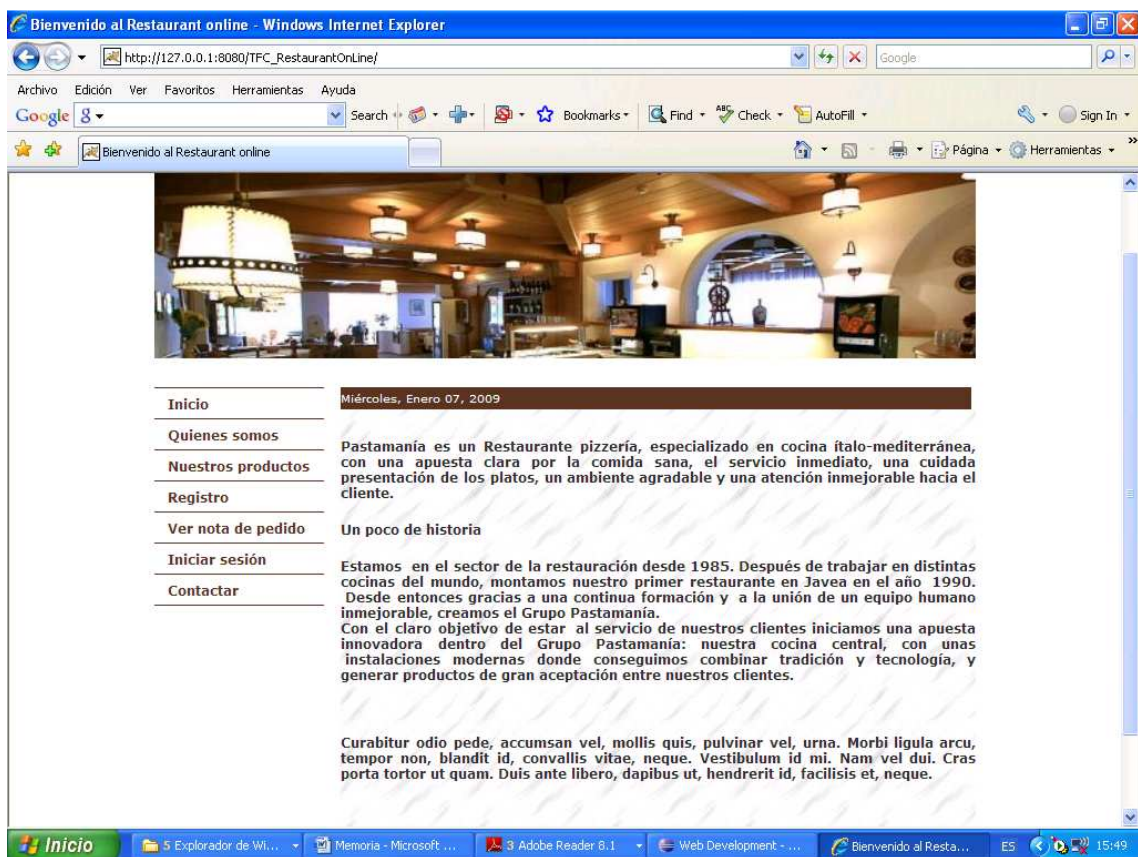


Figura 5.1: Inicio de Restaurant-OnLine

Desde esta página se pueden realizar diversas acciones. Principalmente las siguientes:

- permanecer como visitante y visualizar los productos que se ofrecen, incluso, usar la nota de pedido para añadir artículos a ésta.
- acceder a la página de registro, para hacerse cliente.
- iniciar sesión como cliente o como empleado de la empresa.

Consultar artículos

A través del enlace "Nuestros productos" accedemos a la siguiente página, que nos muestra las diferentes categorías de artículos que pueden consultarse.

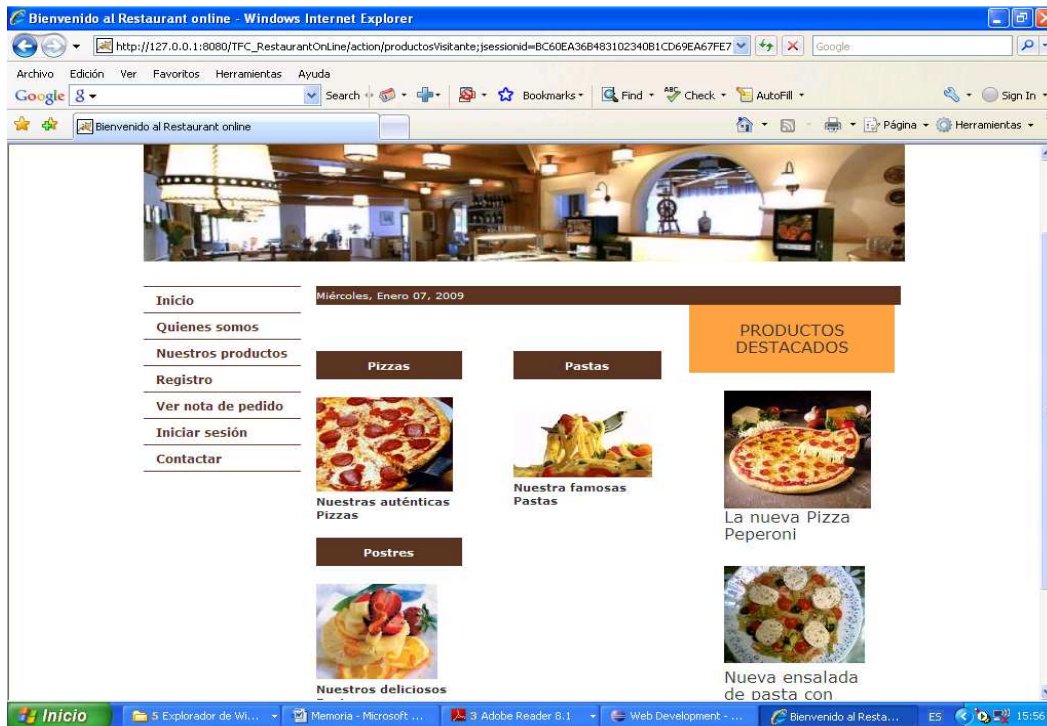


Figura 5.2: Consulta categoría de artículos lado cliente

Haciendo clic sobre uno de ellos se mostrará una lista de los artículos de esta categoría.

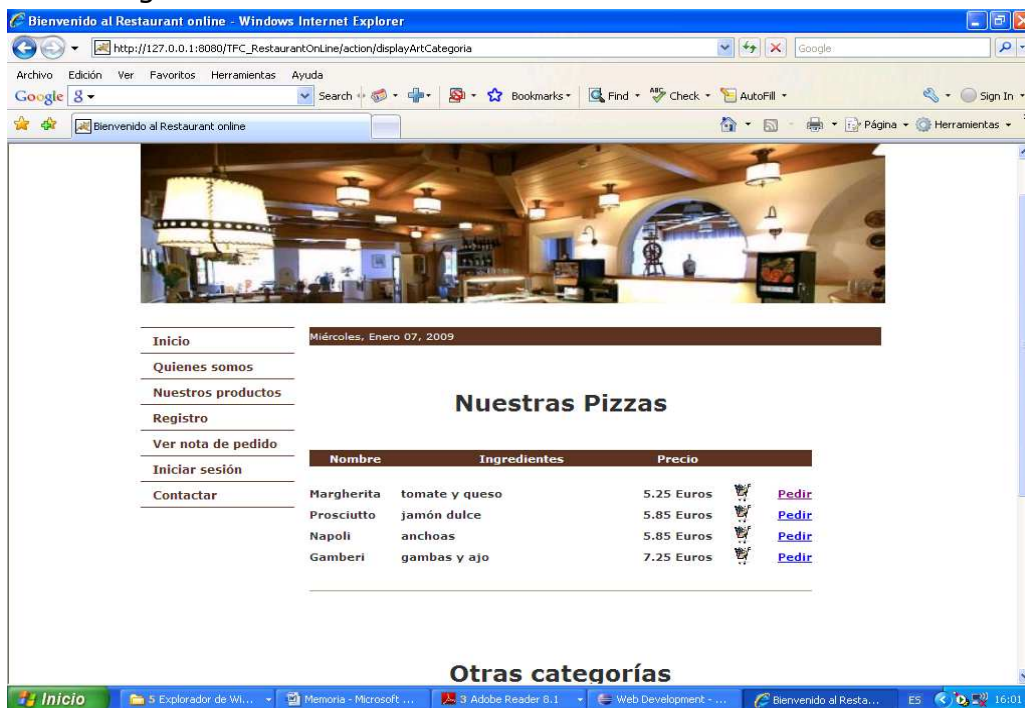


Figura 5.3: Consulta artículos lado cliente

Consultar nota de pedido

En cualquier momento podemos consultar la nota de pedido con los artículos que hasta el momento hemos incorporado a través de los enlaces de "Pedir", al final de cada línea de artículo.

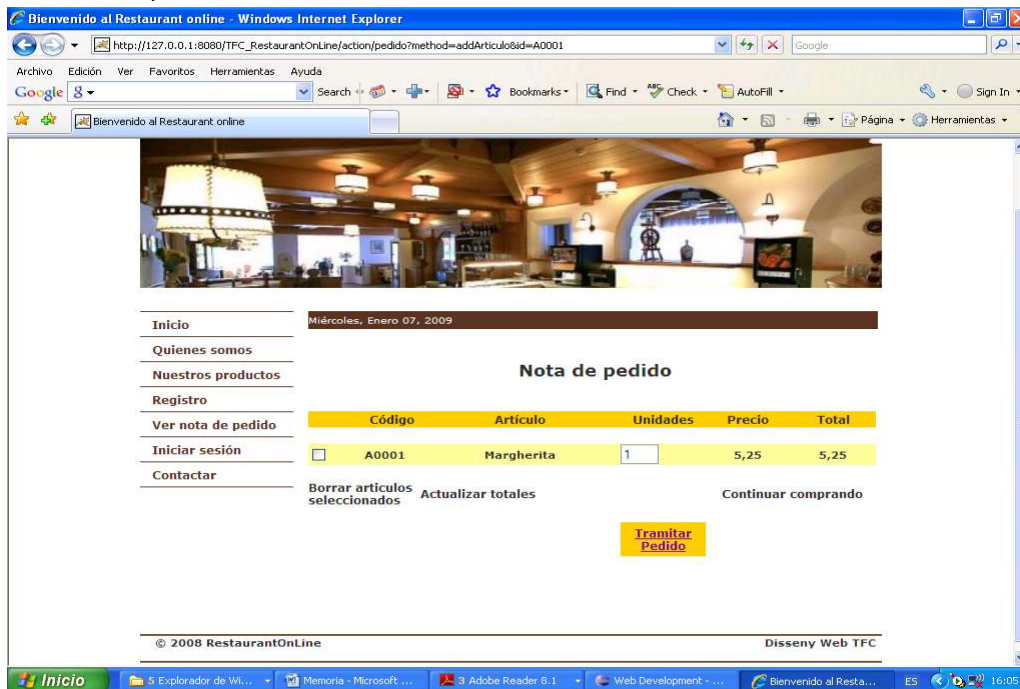


Figura 5.4: Consulta nota de pedido lado cliente

Realizar pedido

Una vez hayamos incorporado todos los artículos deseados a nuestra nota de pedido, podemos tramitarlo. Si somos cliente necesitaremos acceder al sistema con nuestro nombre y contraseña o, en caso de no estar registrado, efectuar dicho trámite.



Figura 5.5: Login de clientes

Confirmar pedido

Una vez iniciada sesión, podremos recuperar nuestra nota de pedido, para dar trámite definitivo al pedido. Para ello, se nos pedirá que confirmemos los datos de envío e indiquemos el día y hora deseados.

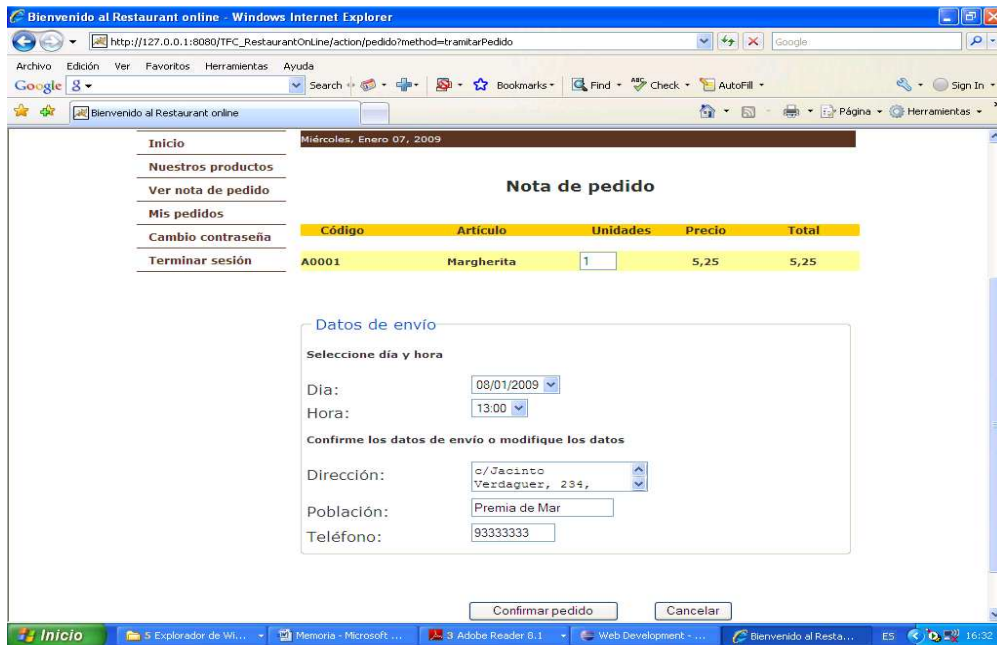


Figura 5.6: Confirmación nota pedido de cliente

Consultar pedidos tramitados

Iniciada la sesión, podremos consultar nuestros pedidos realizados y que todavía están pendientes de aceptar.

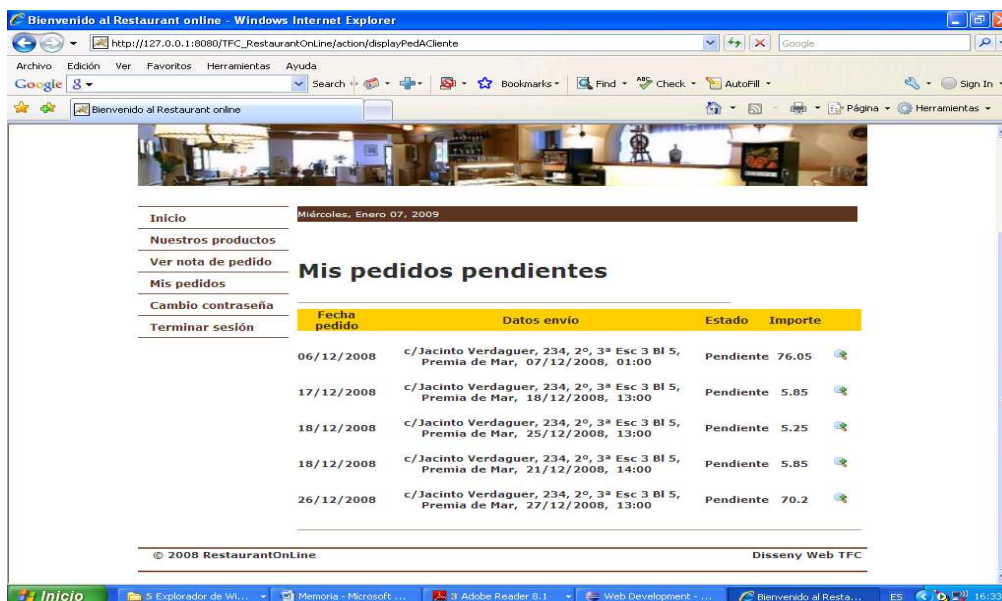


Figura 5.7: Consulta pedidos pendientes por cliente

Registro de clientes

Para registrarse en el sistema deberemos rellenar el siguiente formulario.

Bienvenido al Restaurant online - Windows Internet Explorer

http://127.0.0.1:8080/TFC_RestaurantOnline/action/displayRegistrationForm

Bienvenido al Restaurant online

Miércoles, Enero 07, 2009

Registro de clientes

Datos personales

Nombre*:

Primer apellido*:

Segundo apellido*:

Email*:

Datos domicilio

Dirección*:

Núm.*: Piso: Pta.: Esc.: Bl.:

Población*:

CP*:

Teléfono*:

Datos de acceso a Restaurant online

Nombre de usuario:

Contraseña*:

Confirmar contraseña*:

He leído las [condiciones generales de contratación](#)

Figura 5.8: Registro de clientes

Menú de administración

El usuario, ya sea empleado o administrador, deberá acceder a través de la misma página de login que los clientes. La página de inicio de estos usuarios será la que nos permite acceder a los tres pilares de la administración de Restaurant-OnLine, esto es, gestión de pedidos, gestión de artículos y gestión de usuarios.

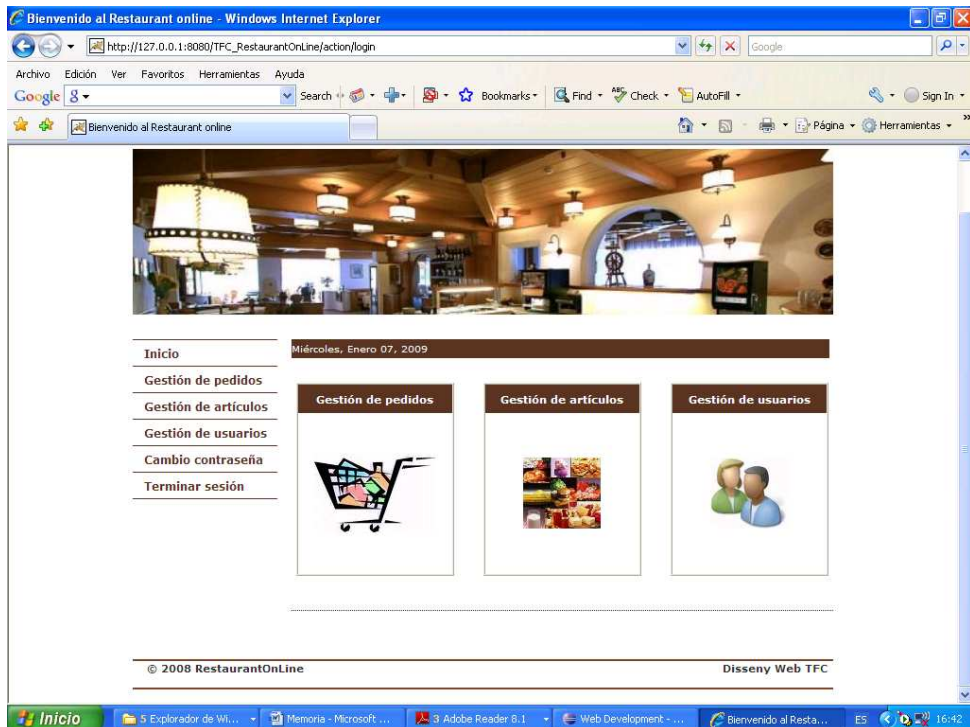


Figura 5.9: Menú administración lado empresa

Gestión de pedidos

En esta sección podremos acceder, básicamente, a las siguientes funciones:

Consultar pedidos

Aquí se nos ofrece la posibilidad de consultar los pedidos según diferentes criterios de consulta: estado, nombre de usuario, fecha o identificador.

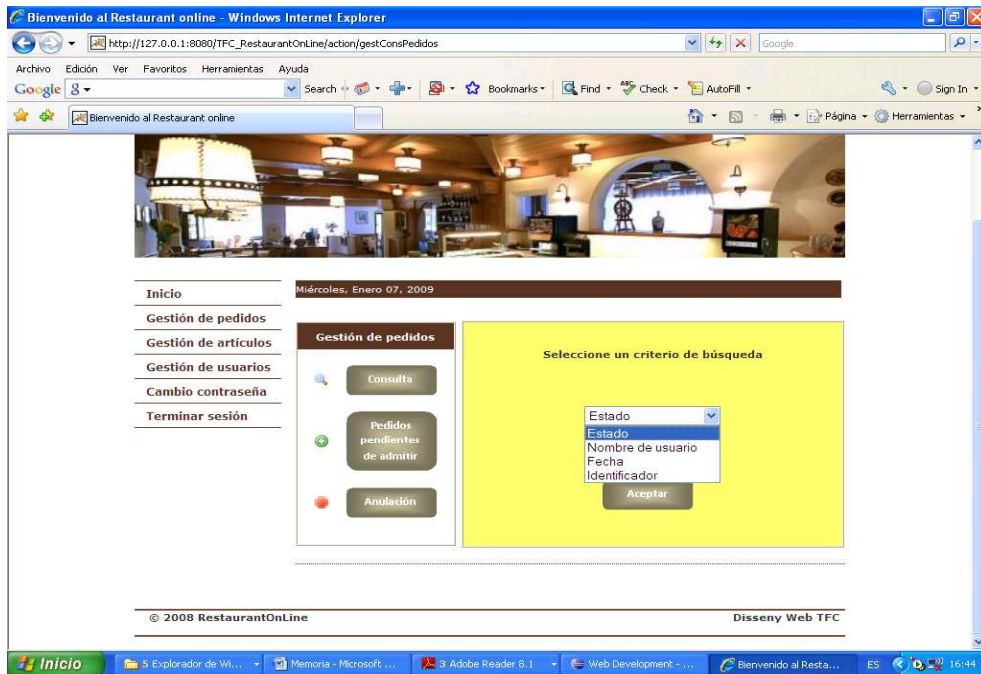


Figura 5.10: Consulta pedidos lado empresa

En los tres primeros casos se mostrará una lista de pedidos con la opción de acceder a los detalles.

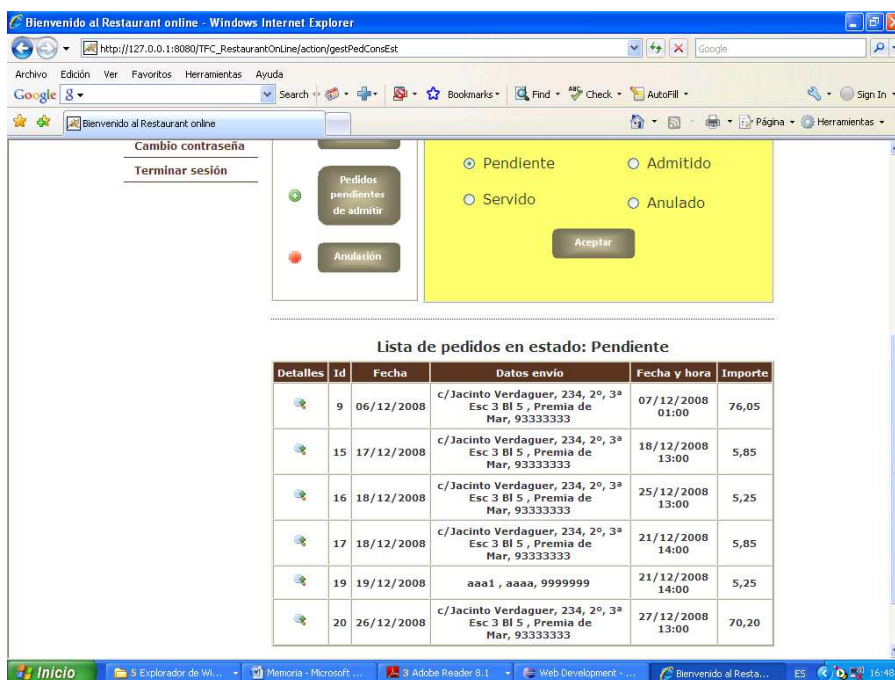


Figura 5.11: Lista pedidos según criterio de consulta lado empresa

En el caso de la consulta por identificador se mostrará, naturalmente, únicamente el pedido requerido, con la misma opción para los detalles del pedido.

Admitir pedidos

El botón "Pedidos pendientes de admitir" nos permitirá obtener la lista de pedidos pendientes con la opción de admitirlos, una vez marcado el botón de opción correspondiente.



Figura 5.12: Admisión pedidos

Anular pedidos

Por último, disponemos de la opción de anular pedido para lo cual deberemos introducir primero el identificador de pedido correspondiente.

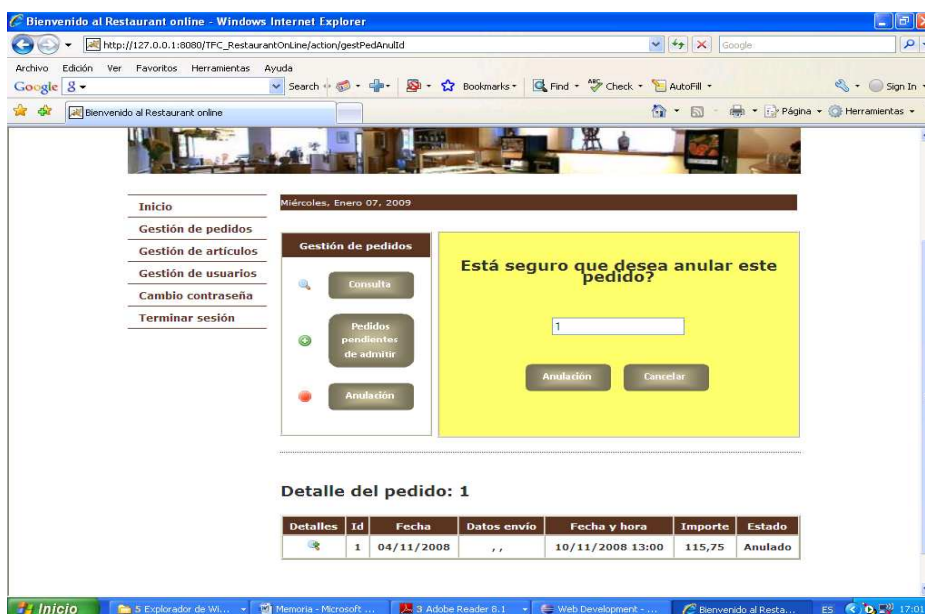


Figura 5.13: Anulación pedidos

Gestión de artículos

Consultar artículos

Podremos consultar los artículos de nuestra base de datos según dos criterios de consulta: categoría o código (identificador)

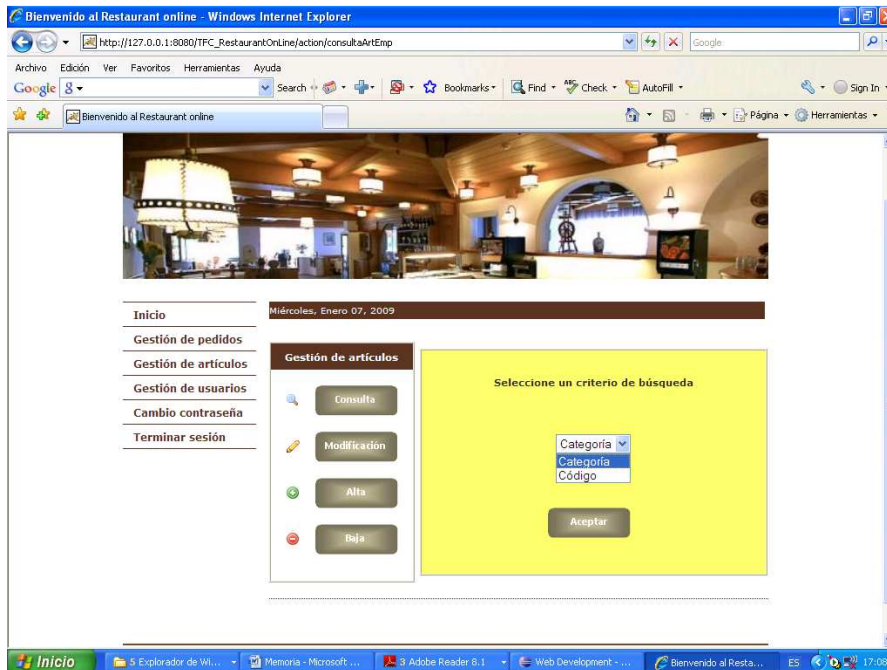


Figura 5.14: Selección criterio de consulta de artículos lado empresa

Modificar artículo

Tras introducir el código de artículo que se desea modificar accederemos a los campos disponibles, como se muestra a continuación.



Figura 5.15: Modificación artículo

Alta artículos

Para el alta deberemos rellenar una serie de campos que aparecen tras el enlace correspondiente.



Figura 5.16: Alta artículos

Baja artículos

Para efectuar una baja de artículo será necesario indicar el código y confirmar la operación.

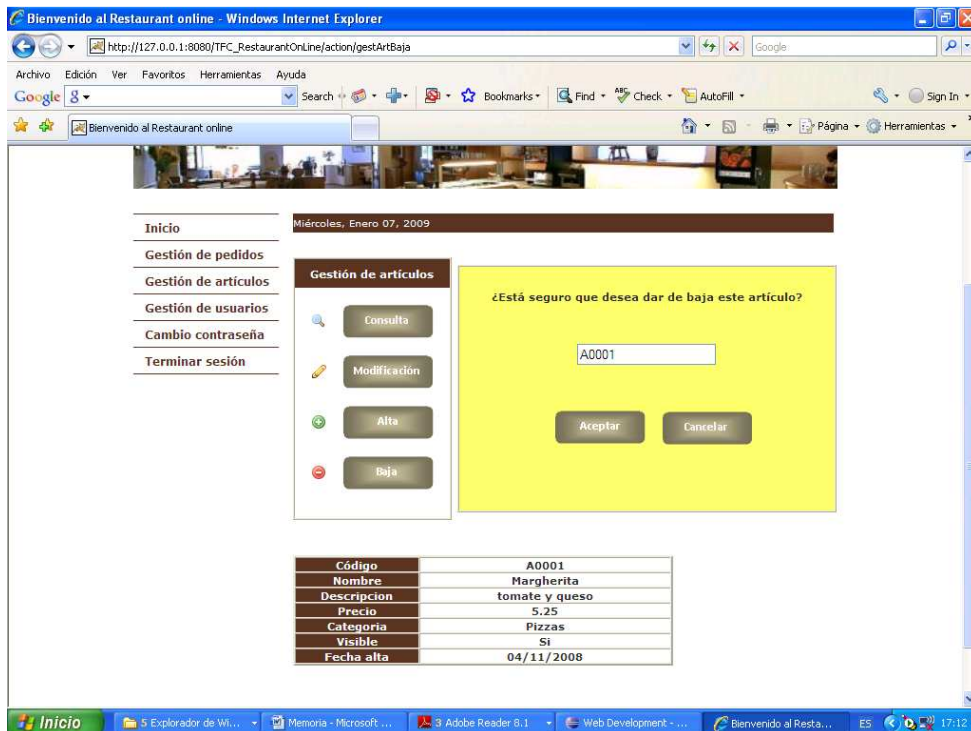


Figura 5.17: Baja artículos

Gestión de usuarios

En esta sección disponemos de las opciones de consulta, modificación, alta y baja. Será únicamente el administrador el que tendrá acceso a las tres últimas opciones.

Consulta usuarios

También en este caso podremos realizar la consulta según dos criterios diferentes: nombre de usuario o tipo de usuario.

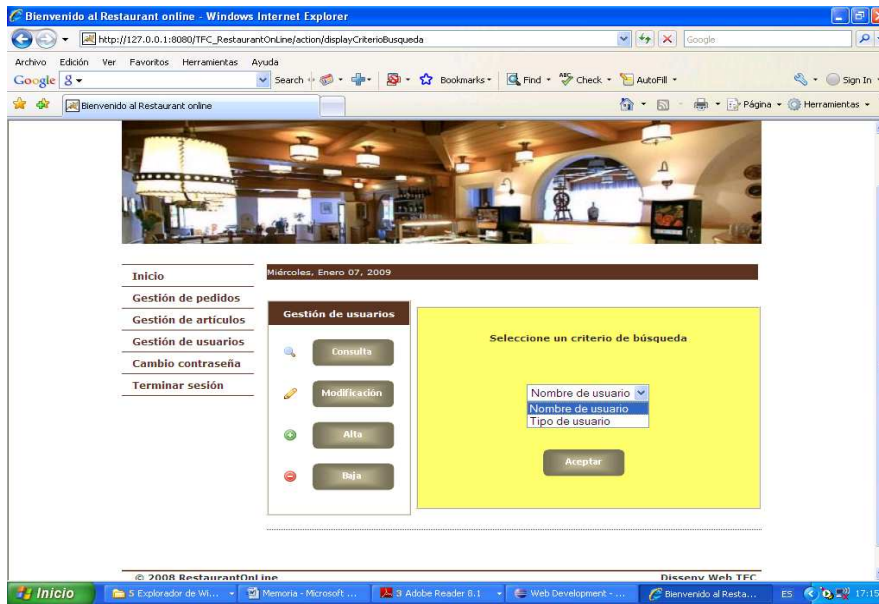


Figura 5.18: Selección criterio de consulta de usuarios

En caso de efectuar la consulta por tipo de usuario, tras seleccionarlo de una lista desplegable, se obtendrá un listado de usuarios, con la opción de acceder a la información cada uno de ellos.

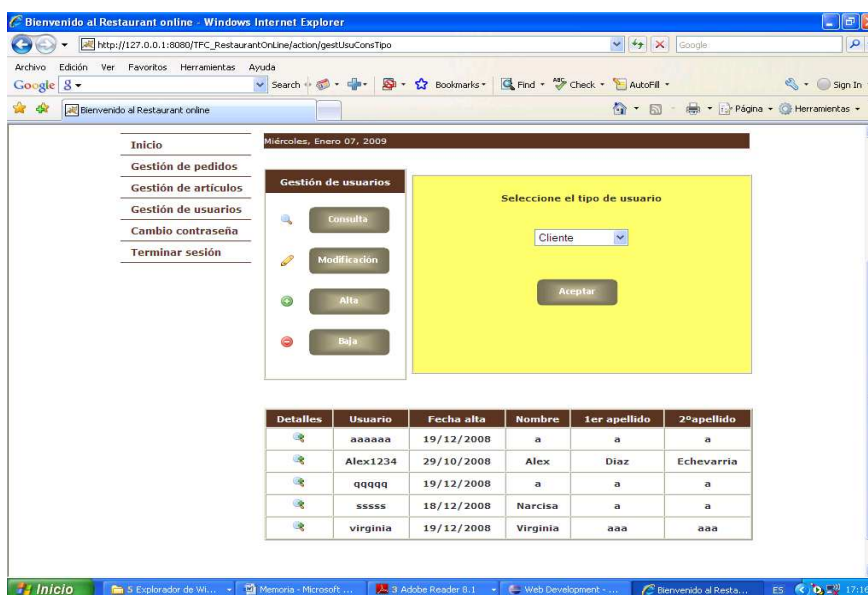


Figura 5.19: Lista tipo de usuarios

Modificación de usuarios

Indicando el identificador de usuario que se desea modificar, aparecerá una pantalla con los campos modificables.

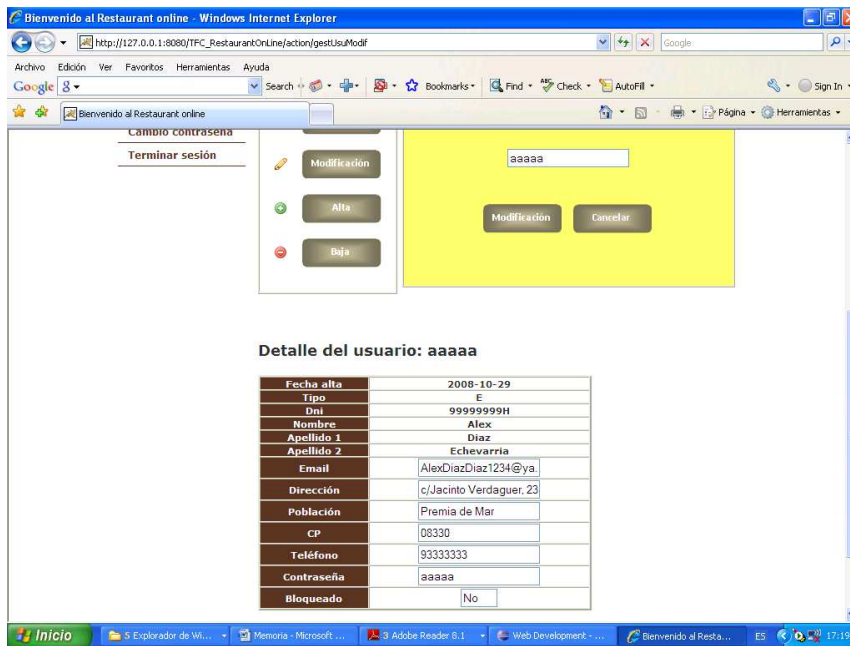


Figura 5.20: Modificación usuario

Alta de usuarios

En este enlace se nos presentará una serie de campos de obligado cumplimiento.

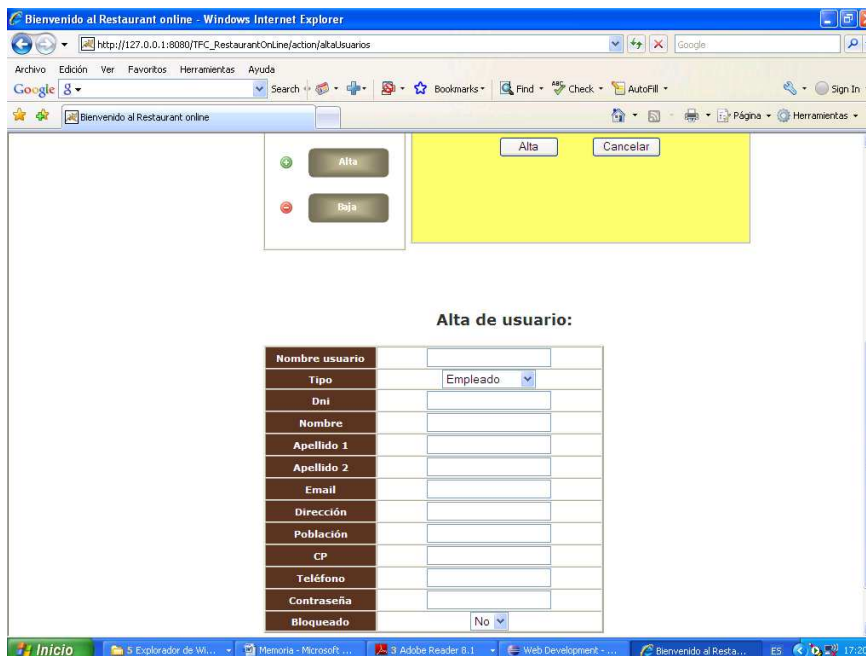


Figura 5.21: Alta usuario

Baja de usuarios

Para efectuar una baja de usuario será necesario indicar el identificador deseado y confirmar la operación.

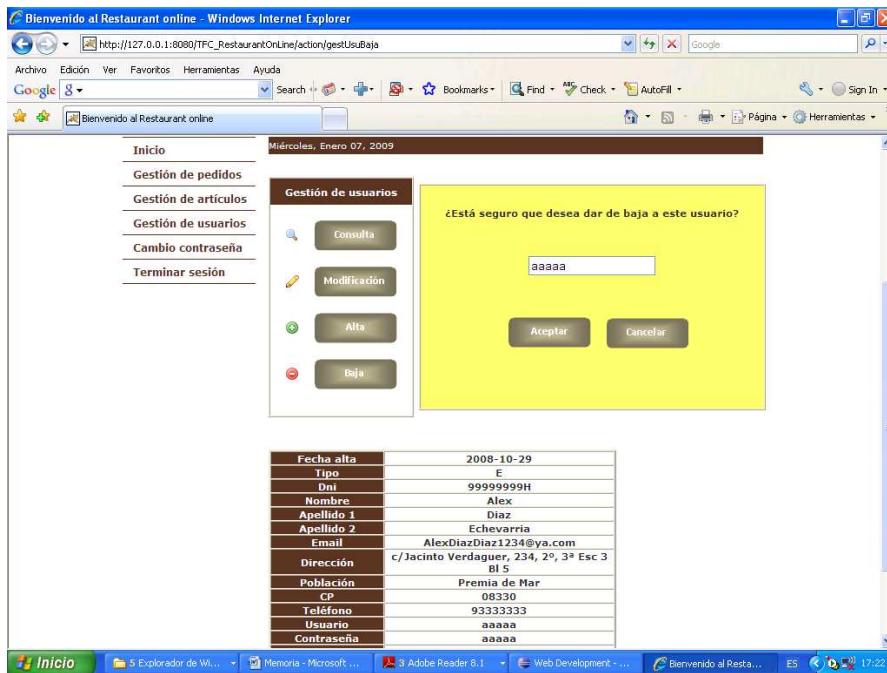


Figura 5.22: Baja usuario

Capítulo 6 Conclusiones

La elaboración de este TFC ha servido, principalmente, para conocer con cierta profundidad la plataforma JEE, sus componentes y alguna de sus tecnologías principales, Servlet y JSP.

Conforme se avanzaba en su estudio, por un lado, se ha hecho patente la necesidad de utilizar patrones de diseño, esto es, soluciones a problemas de diseño verificadas con anterioridad y reusables. Por otro, apoyarse en algún tipo de entorno que facilitara el desarrollo de una aplicación web en Java.

Ambos retos se han podido superar con el uso del Framework Struts. Éste nos ha sido de gran ayuda ya que nos ha proporcionado tanto la separación de la lógica de negocio, la lógica de control y el código de presentación como una estructura básica sobre la cual construir la aplicación.

Aunque al principio encajar todas las piezas del puzzle que se quería construir no ha sido tarea fácil, finalmente, la satisfacción de haber aprendido una tecnología prácticamente desconocida ha sido grande.

También se ha de reconocer que la falta de tiempo ha hecho imposible pulir, como se hubiese querido, la aplicación. Muchos aspectos son mejorables y otros no han sido tenidos en cuenta, como son un mayor control de acceso, tema de seguridad, diseño más elaborado, el manejo adecuado de excepciones, la internacionalización, uso de la biblioteca Log4j, y un largo etc. Pero, como en todo aprendizaje de una nueva tecnología, se necesita tiempo, que posteriormente se suple con la experiencia que se va adquiriendo, y que permite ir mejorando e introduciendo nuevos aspectos de diseño.

En resumen, se puede afirmar que la experiencia ha sido del todo positiva y ha despertado un mayor deseo del inicial de profundizar en la tecnología Java EE.

Capítulo 7 Glosario

Cliente: parte de la aplicación que interactúa con el visitante o cliente del restaurante.

Empresa: parte de la aplicación que interactúa con los empleados y administradores de la empresa, y sirve para la gestión y administración de la aplicación.

Patrón de diseño: son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de aplicaciones y otros ámbitos referentes al diseño de interacción o interfaces.

Struts: herramienta de soporte para el desarrollo de aplicaciones web Java EE bajo el patrón MVC.

Pool de conexiones: grupo de conexiones abiertas de una base de datos que pueden ser reusadas al realizar múltiples accesos a ésta.

Framework: estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

JavaBean: componente de software reusable para Java. En la práctica son clases escritas en Java que deben seguir una cierta convención.

Componente Web: componentes pertenecientes a la capa de presentación, como los Servlets y páginas JSP.

Contenedor Web: entorno de ejecución de los componentes web.

Capítulo 8 Bibliografía

Falkner, Jayson (2003): JAVA Servlets and JavaServerPages: the J2EE technology Web tier: Addison-Wesley.

Derek C. Ashmore (2004): The J2EE Architect's Handbook: DVT Press.

Richard Hightower (2004): Jakarta Struts LIVE: SourceBeat, LLC.

Srikanth Shenoy (2004): Struts Survival Guide: ObjectSource LLC.

Sun Microsystems: Core J2EE Pattern Catalog: Business Delegate, Transfer Object y Data Access Object.

Sang Shin (2008): Java EE (J2EE) Programming (with Passion!)
[www.javapassion.com/j2ee]

Adictos al Trabajo: Aplicación profesional con Struts – 2003
[<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=strutsb>]

Ramiro Lago (2007): Patrones de diseño software [http://www.proactiva-calidad.com/java/patrones/index.html#algunos_patrones]

The Middleware Company (2003): J2EE Best Practices
[<http://www.middleware-company.com/>]