Diseño de Aplicaciones J2EE

Aplicación de Patrones de Diseño

(Tienda de libros virtual)

Ingeniería Técnica Informática de Gestión
(Memoria)

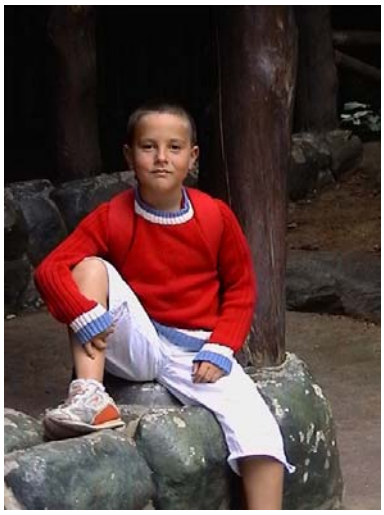
Autor: **Juan José Hidalgo Parrilla**
Consultor: **Javier Ferró García**
Entrega: **15 de Enero de 2007**

1. Dedicatoria y agradecimientos.

Aunque no tengo dudas de a quien dedicar este trabajo, creo que esta es la parte más difícil de toda la carrera. ¿Cómo dar las gracias a las personas que me han ayudado todo este tiempo?, pero no quiero de dejar intentarlo.

En primer lugar a mi esposa, amiga, compañera de estudios, consejera, etc., que me ha animado en todo momento, que nunca me ha dejado abandonar, y han sido muchos los momentos en los que he estado decidido a ello, y que ha puesto en ocasiones incluso más empeño que yo. Gracias, Chari, por aguantar junto a mí. ¡¡¡ Lo hemos conseguido!!!

Por supuesto a mis dos hijos, Cristina y Daniel, que han tenido que soportar, sin ser muy conscientes de porqué, largos fines de semana encerrados en casa, sin poder ir al campo o al cine, o sin que, simplemente, me pusiera a jugar con ellos a cualquier cosa cuando me lo pedían. A ellos les quiero pedir perdón y prometerles, que les devolveré todo el tiempo que les he robado.



¡GRACIAS!

2. Resumen.

Una de las tecnologías que mayor impacto ha tenido en el desarrollo de aplicaciones empresariales en los últimos tiempos ha sido J2EE (Java 2 Enterprise Edition). En este trabajo se aplican los principales conceptos de esta tecnología, así como los patrones de diseño que ayudan a su correcta implementación.

Como aplicación práctica se realiza el proceso de ingeniería completo para el desarrollo de una tienda virtual de libros, donde los usuarios registrados puedan consultar el catálogo de libros disponibles y realizar compras. Además, como valor añadido, los usuarios podrán introducir comentarios referentes a cualquiera de los libros, lo cual le servirá de orientación a los futuros clientes para decidir qué libro comprar.

El sistema dispondrá de las siguientes funcionalidades:

- Registro de usuarios: los usuarios del sistema se podrán registrar como clientes antes de poder comprar, hacer pedidos o introducir comentarios sobre libros.
- Carrito de compra: en él se incorporarán los libros que un usuario decida comprar.
- Entrada de comentarios: posibilitará la inclusión de comentarios relacionados con un determinado libro.
- Administración del sistema: un usuario especial podrá administrar el catálogo de libros, acceder a la lista de usuarios para realizar operaciones como bloqueos o envío de avisos y gestionar los pedidos.

.

3. Índice de contenidos.

1. DEDICATORIA Y AGRADECIMIENTOS. 2

2. RESUMEN. 3

3. ÍNDICE DE CONTENIDOS. 4

4. CUERPO DE LA MEMORIA, DIVIDIDO EN CAPÍTULOS. 6

4.1. CAPÍTULO 1: INTRODUCCIÓN. 6
4.1.1. Justificación del TFC y contexto en el que se desarrolla. 6
4.1.2. Objetivos del TFC. 7
4.1.3. Enfoque y método seguido. 7
4.1.4. Planificación del proyecto. 8
4.1.5. Productos obtenidos. 10
4.1.6. Descripción de los capítulos de la memoria. 11
4.2. CAPÍTULO 2: ANÁLISIS 12
4.2.1. Requisitos funcionales 12
4.2.2. Casos de uso 13
4.2.3. Diagramas de colaboración 20
4.2.4. Diagramas de secuencia 23
4.2.5. Diagrama estático de clases 24
4.2.6. Análisis de la interfaz de usuario 25
4.3. CAPÍTULO 3: DISEÑO 34
4.3.1. Diagrama estático de diseño 34
4.3.2. Diseño de la base de datos. 35
4.3.3. Diseño arquitectónico 36
4.3.4. Arquitectura de la aplicación Tienda Virtual 41
4.3.5. Despliegue del sistema. 46
4.4. CAPÍTULO 4: AMPLIACIONES DEL SISTEMA 47
4.5. CAPÍTULO 5: CONCLUSIONES. 48

5. GLOSARIO 49

6. BIBLIOGRAFÍA. 50

Índice de figuras

Ilustración 1: Planificación 9
Ilustración 2: Caso de uso usuarios 13
Ilustración 3: Caso de uso administradores 14
Ilustración 4: Diagrama de colaboración validación de usuarios 21
Ilustración 5: Diagrama de colaboración compra y confirmación 22
Ilustración 6: Diagrama de secuencia compra y confirmación 23
Ilustración 7: Diagrama estático de clases 24
Ilustración 8: Pantalla catálogo de libros 25
Ilustración 9: Pantalla ficha información de libros 26
Ilustración 10: Pantalla registro nuevo usuario 27
Ilustración 11: Pantalla Confirmación registro de usuario 28
Ilustración 12: Pantalla carrito de la compra 28
Ilustración 13: Pantalla buscador de libros 29
Ilustración 14: Pantalla modificación catálogo 30
Ilustración 15: Pantalla Datos de ofertas 31
Ilustración 16: Pantalla gestión de usuarios 32
Ilustración 17: Pantalla lista de avisos 33
Ilustración 18: Diagrama de clases 35
Ilustración 19: Arquitectura Struts 38
Ilustración 20: Patrón Business Delegate 39
Ilustración 21: Arquitectura tienda virtual 41
Ilustración 22: Diagrama de secuencia funcionamiento Struts 42
Ilustración 23: Actuación de los componentes de la aplicacion 43
Ilustración 24: funcionamiento Struts 44
Ilustración 25: Diagrama de clases de Tienda-Virtual 45
Ilustración 26: Despliegue del sistema 46

4. Cuerpo de la memoria, dividido en capítulos.

4.1. Capítulo 1: Introducción.

4.1.1. Justificación del TFC y contexto en el que se desarrolla.

Aunque el resultado del proyecto ha sido la implementación de una tienda virtual, no ha sido esta la meta principal de este trabajo, los objetivos que han motivado la elección de este proyecto han sido varios.

Por un lado profundizar en el desarrollo de aplicaciones en Java y la aplicación de las tecnologías J2EE. El lenguaje Java, aunque había tenido contacto con el a lo largo de la carrera, nunca había abordado un desarrollo desde cero y puesto que esta tecnología tiene una gran penetración en el mundo laboral, consideraba que era interesante afrontar el reto y descubrir sus ventajas e inconvenientes.

Por otro lado por mi experiencia en desarrollo, tenía cierto interés en comprobar la flexibilidad de esta tecnología a la hora de abordar un desarrollo medianamente complejo, es decir, un proyecto como el desarrollado donde intervienen, aunque a pequeña escala, una base de datos la cual puede sufrir modificaciones, interfaces de usuario más o menos complicadas que también pueden sufrir modificaciones y por supuesto una lógica de negocio donde nuevamente podemos tener que realizar cambios más o menos periódicos.

A estos objetivos se une otro, no menos importante y que, en definitiva, contribuirá a lograr los dos anteriores, como es la aplicación de patrones de diseño, como la forma de sistematizar la implementación, para garantizar un orden y forma de actuar, que faciliten el desarrollo organizando los elementos que componen el sistema y dirigiendo de alguna forma, las interacción entre todos ellos, dando como resultado un sistema bien formado, fácil de mantener y escalable.

Por último está la utilización de EJB's para el acceso a la base de datos. La tecnología EJB siempre me han resultado complicada de entender y un poco oscura en su funcionamiento, en parte por desconocimiento, por lo cual y con cierto miedo de no lograr el objetivo y tener que dar marcha atrás en mis pretensiones iniciales, decidí enfrentarme a ella para comprender su funcionamiento.

4.1.2. Objetivos del TFC.

El objetivo final de este proyecto es el seguimiento del proceso de ingeniería para el desarrollo de una tienda virtual donde se apliquen los conceptos de J2EE, que son las siglas de Java 2 Enterprise Edition.

Estas tecnologías comprenden un conjunto de especificaciones y funcionalidades para el desarrollo de aplicaciones, tomando como referencia las principales herramientas que actualmente se utilizan en la construcción de aplicaciones Java como struts, patrones de diseño y EJB's.

Los objetivos específicos que se persiguen con este proyecto son los siguientes:

- Estudio y aplicación de patrones de diseño. Se hará un estudio general de las ventajas del uso en el diseño de aplicaciones de los patrones y se profundizará en el Business Delegate o MVC (Modelo Vista Controlador), DTOs (Data Tranasfer Objects) entre otros.
- Implementación de Strutsb para controlar o normalizar la capa de Vista y la capa Controlador, dentro del modelo MVC.
- Funcionamiento y configuración del servidor de aplicaciones JBOSS-Tomcat. Actualmente es uno de los servidores más populares, por lo que se ha elegido como contenedor de EJBs y para la parte Web.
- Implementación de la capa de persistencia accediendo mediante EJBs a una base de datos; en este caso se ha elegido MySql.
- Uso de la herramienta de desarrollo NetBeans para la edición y compilación de clases java, JSP y XML.
- ANT como herramienta para la integración de todos los componentes del sistema.

4.1.3. Enfoque y método seguido.

Como ya se ha indicado el objetivo principal ha sido profundizar en la tecnología J2EE. La aplicación se ha diseñado respetando en todo momento un esquema de 3 capas haciendo una separación clara entre las capas de vista, capa de negocio y capa de datos.

Para lograrlo ha sido necesario en primer lugar un periodo de formación para la utilización de Struts y sobre todo de EJB, así como de diferentes patrones de diseño.

Por el lenguaje y los elementos a utilizar también ha sido importante identificar las herramientas con las que se va a trabajar, intentando seleccionar aquellas que proporcionen ayuda suficiente en la implementación de una de las partes.

La línea de trabajo seguida ha sido:

1. Identificación de información sobre las diferentes tecnologías a aplicar.
2. Identificación de herramientas. Herramienta de desarrollo, base de datos, servidor de aplicaciones.
3. Planificación de los principales hitos del proyecto
4. Durante el análisis y diseño ha sido necesario estudiar a fondo toda la documentación sobre la tecnología, para poder tomar las decisiones correctas de diseño y evitar así tener que dar marcha atrás a parte de los trabajos realizados en la fase de implementación.
5. Implementación. En esta fase se continua profundizando en la tecnología, además de la documentación se han localizado una serie de Web's donde poder consultar los posibles problemas o poder obtener recursos Web a incorporar en nuestro sistema.
6. Revisión pruebas y documentación del código.
7. Memoria y presentación final.

4.1.4. Planificación del proyecto.

Dado que el tiempo disponible para elaborar el proyecto era escaso y muchas las tareas a desarrollar, sobre todo de documentación, el planificar el tiempo a dedicar a cada una de ellas es vital para tener una visión global del proyecto, que te permita conocer en cada momento el plazo de que dispones para cada una de las tareas y en el caso de surgir nuevas, saber en que momento puedes realizarlas y cuales tendrás que acortar.

La planificación que he seguido ha sido la siguiente:

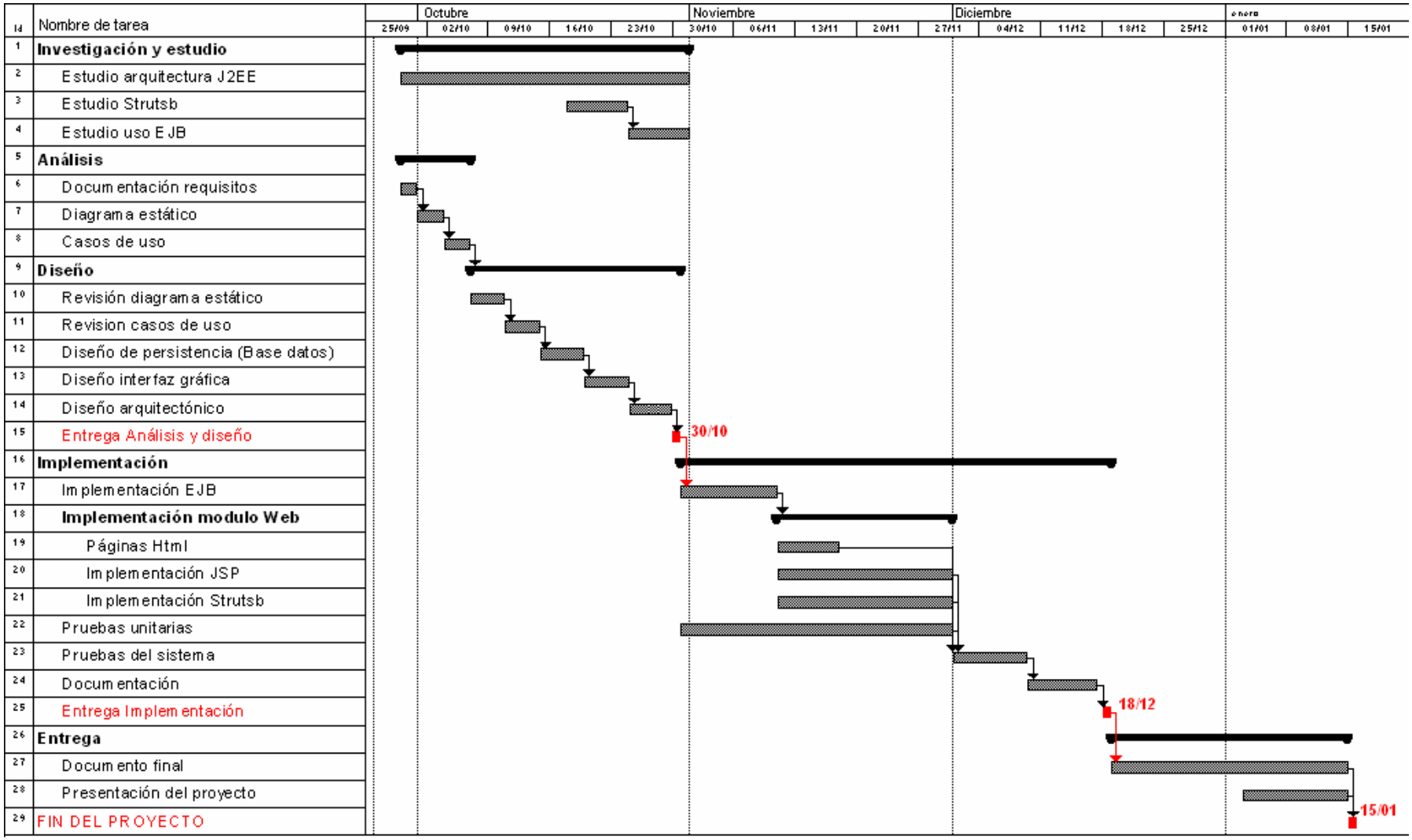


Ilustración 1: Planificación

4.1.5. Productos obtenidos.

Con este trabajo se han obtenido los siguientes productos que, aunque están todos integrados bajo un único proyecto, podrían ser reutilizados fácilmente en otros proyectos distintos. Estos productos son subproyectos dentro del proyecto general Tienda virtual y se detallan a continuación.

Tienda-Virtual

Este es el proyecto general y engloba a todos los elementos que finalmente compondrán la tienda virtual desarrollada.

Tienda

Este subproyecto contiene todos los componentes que implementan las funcionalidades de la tienda virtual, a excepción de la capa de acceso a la base de datos que se ha realizado en un subproyecto diferente. Con esto se consigue que la interfaz y lógica de negocio de la tienda puedan utilizarse con cualquier otro método de acceso a bases de datos, por ejemplo en este caso se ha utilizado EJB pero podría utilizarse JDBC, Hibernáte, etc.

La tienda implementa de una parte para administración y otro que implementará el funcionamiento general de la tienda (consultas y compras).

TiendaVirtual-EJBModule

Este es el subproyecto que se encarga de implementar el acceso a la base de datos. En este caso se ha utilizado EJB, a los que se accede mediante una clase de sesión que aísla del exterior las clases de entidad que representan la base de datos.

Base de datos

Como último producto se tiene la base de datos. En este caso se ha utilizado una base de datos MySql por su facilidad de utilización, es rápida y permite todos los tipos de datos y relaciones necesarias para este proyecto, pero sería fácilmente sustituible por otro gestor de base de datos gracias a la separación entre la capa de negocio y la de acceso a datos.

4.1.6. Descripción de los capítulos de la memoria.

En el resto de capítulos de esta memoria se da una visión más detallada de los aspectos más relevantes de las tecnologías utilizadas, y su aplicación concreta en el desarrollo del proyecto de tienda virtual.

En el **capítulo 2** se detalla el proceso de análisis que ayudará a tener una visión de que tipo de aplicación es la que se quiere obtener. Se hará un estudio de los requisitos iniciales y se identificarán y describirán los principales actores y escenarios.

En el **capítulo 3**, se aborda la fase de diseño, donde partiendo del compromiso de realizar un diseño en tres capas (presentación, negocio, acceso a datos) se presentan los diferentes patrones que se aplicarán en la implementación de cada una de las partes. En un primer punto se explican aquellos aspectos más interesantes de la tecnología y posteriormente su aplicación concreta dentro del proyecto de tienda virtual. También se describe el entorno donde funcionará la aplicación, servidor de aplicaciones, parámetros de configuración, etc.

En el **capítulo 4**, dedicado a las mejoras y ampliaciones, no se hace una lista de dichas mejoras o ampliaciones, que pueden ser muchas, sino que se expone la bondad de la tecnología utilizada para la implementación de estas mejoras.

Por último el **capítulo 5** está dedicado a exponer las principales conclusiones a las que he llegado al término de este proyecto.

4.2. Capítulo 2: Análisis

4.2.1. Requisitos funcionales

El software a construir ha de cumplir con los siguientes requisitos funcionales.

1. Permitir consultar el catálogo de libros.
2. Permitir buscar un libro o grupo de libros en el sistema por cualquiera de los datos.
3. Permitir ver el detalle de un determinado libro del catálogo.
4. Permitir consultar las ofertas disponibles.
5. Los usuarios han de estar registrados previamente en el sistema para poder realizar compras.
6. El usuario, una vez validado en el sistema, dispondrá de un “carrito de la compra” donde se irán acumulando las compras realizadas.
7. El usuario podrá eliminar compras del carrito.
8. Cualquier usuario previamente validado en el sistema podrá incorporar comentarios sobre los libros disponibles en el catálogo libros.
9. Existirá un usuario especial con perfil de administrador que podrá realizar operaciones de mantenimiento del sistema.
10. El administrador podrá acceder al catálogo de Libros para realizar labores de mantenimiento. Dispondrá de las opciones de incorporar nuevos libros, actualizar los datos de los libros.
11. El administrador podrá poner en oferta Libros, pudiendo indicar su precio de oferta y el periodo en el que permanecerá de oferta.
12. El administrador podrá consultar la lista de usuarios registrados y dispondrá de las opciones de modificar, eliminar, bloquear, poner avisos a usuarios y consultar las compras realizadas.
13. Un usuario que esté bloqueado no podrá acceder al sistema para realizar compras.

4.2.2. Casos de uso

Identificación de actores

En el sistema se identifican dos actores claramente diferenciados.

Usuario: Cualquier persona que acceda al sistema.

Administrador: Es un usuario con permisos para realizar labores de administración en el sistema. El usuario administrador tiene todas las características de los usuarios normales pero además tienen la propiedad administrador.

Descripción de casos de uso

En los siguientes diagramas simplificados, se muestran los escenarios principales que podemos identificar en el sistema. El primero de ellos corresponde a los escenarios para los usuarios generales del sistema y el segundo para el administrador del sistema.

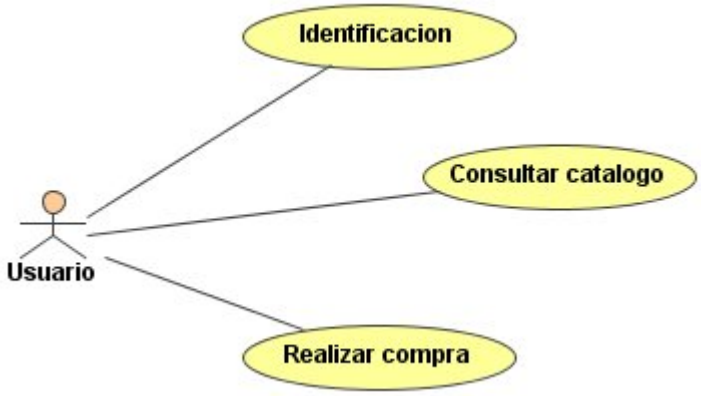


Ilustración 2: Caso de uso usuarios

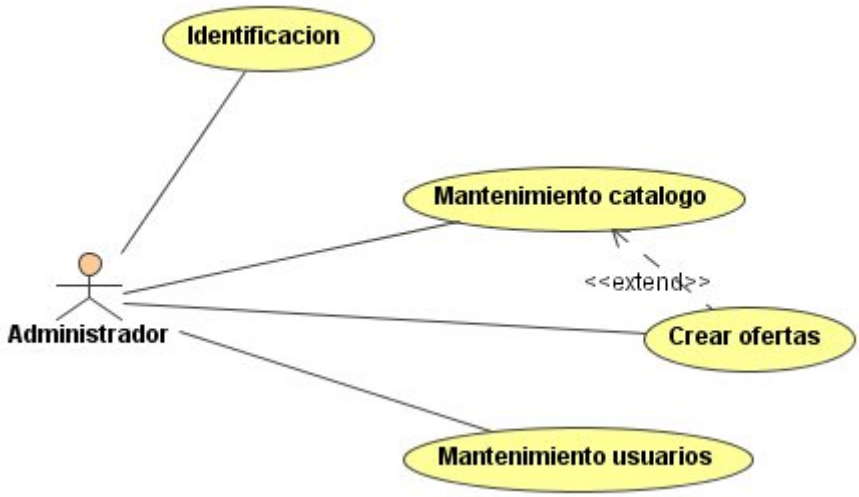


Ilustración 3: Caso de uso administradores

A continuación se describen de forma detallada los casos de uso de cada uno de los escenarios propuestos.

CASO DE USO: IDENTIFICACIÓN

Funcionalidad general: Verifica que el usuario y la contraseña introducidos correspondan con un usuario registrado en el sistema y que no esté dado de baja o bloqueado.
Actores principales: Administrador, usuario

Precondición: El usuario estará previamente registrado en el sistema.

Poscondición: El sistema ha validado el usuario y password y habilita las opciones propias del perfil del usuario validado.

Escenario principal:

1. Un usuario en cualquier momento introduce un usuario (email) y una password.
2. El sistema verifica la validez del usuario y password introducidos.
3. El sistema presentará los datos del usuario validado

Escenarios alternativos:

1. El usuario está bloqueado.
    a. No se validará en el sistema y se mostrará un error indicando que el usuario está bloqueado.
2. El usuario no existe o la password es incorrecta.

a. No se validará en el sistema y se mostrará el error.

CASO DE USO: CONSULTAR CATÁLOGO

Funcionalidad general: Presenta al usuario una lista de los libros disponibles en el catálogo pudiéndose realizar búsquedas por diferentes criterios o consultar las ofertas de libros.

Actores: Administrador, usuario

Casos de uso relacionados: Ver detalle

Precondición: Hay libros en el catálogo

Poscondición: Se presenta en pantalla una lista de los libros disponibles en el catálogo que cumplen con los criterios impuestos.

Escenario principal:

1. Cualquier usuario puede solicitar consultar el catálogo general de libros.
2. Se muestra una lista con todos los libros del catálogo.

Flujos alternativos:

1. Se quieren consultar sólo determinados libros.
    a. El usuario no solicita la consulta general del catálogo y solicita la búsqueda de libros, donde podrá introducir los criterios por los que se quieren buscar los libros.
    b. El sistema sólo presentará los libros que cumplan con los criterios introducidos.
2. Se quieren consultar sólo las ofertas.
    a. El usuario no solicita la consulta general del catálogo y solicita la consulta de ofertas.
    b. El sistema sólo presentará las ofertas disponibles.

CASO DE USO: REALIZAR COMPRA

Funcionalidad general: Seleccionar uno o varios libros del catálogo que se irán acumulando en un carrito de la compra y finalmente confirmar toda la compra.

Actores: Usuario

Casos de uso relacionados: Consulta de catálogo, Identificación

Precondición: Estar validado como usuario del sistema

Poscondición: El usuario adquiere uno o varios libros del catálogo.

Escenario principal:

1. El usuario se valida en el sistema.
2. Mediante la consulta del catálogo el usuario localiza el libro que quiere comprar.
3. Una vez localizado el libro, el usuario indica al sistema que quiere comprarlo.
4. El sistema lo incorporará al carrito de la compra del usuario.
5. El usuario repite los pasos 2 a 4 tantas veces como sea necesario.
6. Cuando no se quieran comprar otros libros, el usuario accederá a su carrito de la compra.
7. El usuario confirmará la compra efectuada.
8. El sistema registrará la compra y dejará vacío el carrito listo para poder realizar otra compra.

Flujos alternativos:

1. El usuario no está validado en el sistema.
    a. El usuario no podrá realizar compras y el sistema presentará el error.
2. El libro ya estaba incorporado al carrito.
    a. El sistema incrementará el numero de unidades adquiridas del libro
    b. El sistema recalculará el importe de la compra.
3. El usuario se arrepiente de la compra de un libro.
    a. El usuario accede a su carrito de la compra.
    b. Selecciona la compra a eliminar
    c. Pide al sistema que elimine la compra del carrito.
    d. Se elimina la compra del carrito y se recalcula el total.
4. El usuario quiere más de una unidad de uno de los libros comprados.
    a. El usuario accede a su carrito de la compra
    b. Modifica el numero de unidades del libro
    c. Se recalcula el importe total

CASO DE USO: MANTENIMIENTO DEL CATALOGO

Funcionalidad general: Permite realizar modificaciones en los datos de los libros existentes en el catálogo, eliminar libros del catálogo o añadir libros al catálogo.

Actores: Usuario administrador

Casos de uso relacionados: Consulta de catálogo, Identificación

Precondición: Estar validado como usuario del sistema y que el usuario sea administrador.

Poscondición: El usuario administrador podrá realizar modificaciones sobre el catálogo de libros.

Escenario principal:

1. Un usuario con perfil de administrador se valida en el sistema.
2. El usuario va a modificar los datos de un libro.
    a. Localiza el libro a modificar mediante la búsqueda de libros.
    b. Accede a los datos del libro
    c. Realiza las modificaciones en los datos
    d. Valida las modificaciones realizadas
3. El usuario va a eliminar un libro del sistema
    a. Localiza el libro a modificar mediante la búsqueda de libros.
    b. Pide al sistema que elimine el libro
    c. Si el libro no está incorporado en ninguna compra se elimina
    d. Si el libro está incorporado en alguna compra no se elimina y el sistema presenta el error.
4. El usuario quiere incorporar un libro en el catálogo.
    a. Pide al sistema incorporar un libro.
    b. Completa los datos del libro
    c. Valida los datos
    d. Si el libro no existe se incorpora al catálogo
    e. Si el libro existe no se vuelve a incorporar al catálogo y el sistema presenta el error.

CASO DE USO: CREAR OFERTA

Funcionalidad general: Permite poner como oferta a un libro del catálogo.

Actores: Usuario administrador

Casos de uso relacionados: Consulta del catálogo

Precondición: Se selecciona un libro del catálogo

Poscondición: El libro aparece en oferta en el catálogo.

Escenario principal:

1. Un usuario con perfil de administrador se valida en el sistema.
2. El usuario localiza el libro que quiere poner en oferta.
3. El usuario indica al sistema que va a poner el libro en oferta.
4. El usuario introduce en el sistema el precio de oferta y el periodo (fecha inicio y fecha fin) en la que el libro estará en oferta.
5. El sistema presentará el libro como oferta en las consultas al catálogo o de ofertas realizadas en el periodo indicado en la oferta.

Flujos alternativos:

1. El libro seleccionado ya está fijado como oferta.
    a. El sistema presenta los datos de los periodos y precios que tendrán en dichos periodos de oferta.
    b. El usuario modifica los datos de una oferta o introduce un nuevo periodo de oferta.
2. El usuario quiere eliminar un periodo de oferta de un libro.
    a. Localiza el libro a modificar mediante la búsqueda de libros.
    a. Accede a los datos de ofertas
    b. Selecciona el periodo de oferta a modificar.
    c. Solicita al sistema que elimine el periodo de ofertas.

CASO DE USO: MANTENIMIENTO DE USUARIOS

Funcionalidad general: Permite bloquear, eliminar o consultar las compras de un usuario.

Actores: Usuario administrador

Casos de uso relacionados: Identificación

Precondición: Estar validado como usuario del sistema y que el usuario sea administrador.

Poscondición: El usuario bloqueado o eliminado no podrá validarse en el sistema.

Escenario principal:

1. Un usuario con perfil de administrador se valida en el sistema.
2. El usuario administrador quiere consultar las compras realizadas por un determinado usuario.
    a. El usuario administrador localiza el usuario.
    f. El usuario administrador indica pide las compras realizadas.
    g. El sistema presenta la lista de compras realizadas
5. El usuario administrador quiere bloquear o borrar un usuario.
    a. El usuario administrador localiza el usuario.
    b. El usuario administrador localiza el usuario a bloquear o borrar.
    b. El usuario administrador indica al sistema que bloque o borre el usuario.
    c. El sistema eliminará o bloqueará el usuario seleccionado.
    d. El usuario eliminado o bloqueado no podrá validarse en el sistema

Flujos alternativos:

1. El usuario tiene compras asociadas y el usuario administrador quiere eliminarlo.
    a. El usuario no se eliminará del sistema
    b. El sistema presenta un mensaje de error.

    e. Localiza el libro a modificar mediante la búsqueda de libros.
    f. Accede a los datos del libro

   g. Realiza las modificaciones en los datos
   h. Valida las modificaciones realizadas
6. El usuario va a eliminar un libro del sistema
   a. Localiza el libro a modificar mediante la búsqueda de libros.
   b. Pide al sistema que elimine el libro
   c. Si el libro no está incorporado en ninguna compra se elimina
   d. Si el libro está incorporado en alguna compra no se elimina y el sistema presenta el error.
7. El usuario quiere incorporar un libro en el catálogo.
   a. Pide al sistema incorporar un libro.
   b. Completa los datos del libro
   c. Valida los datos
   d. Si el libro no existe se incorpora al catálogo
   e. Si el libro existe no se vuelve a incorporar al catálogo y el sistema presenta el error.

4.2.3. Diagramas de colaboración

A continuación se muestra el diagrama de colaboración de dos partes clave en el funcionamiento del sistema, por ser estas funcionalidades las de mayor uso.

La primera de ellas es la validación de los usuarios, que corresponde con el siguiente diagrama.

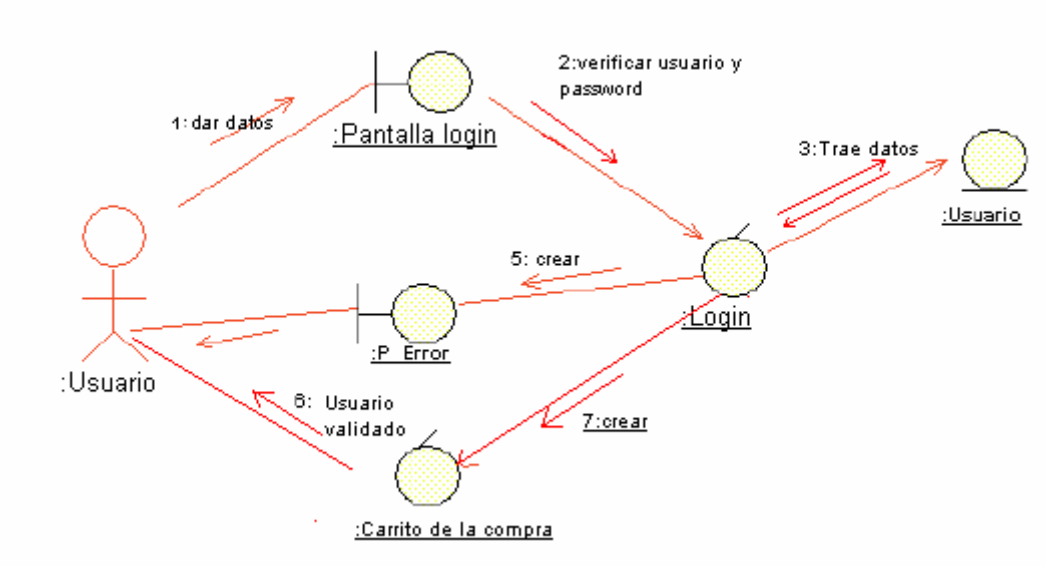


Ilustración 4: Diagrama de colaboración validación de usuarios

El usuario introducirá la cuenta de correo y una password. El sistema verificará que el usuario existe y es correcta su password accediendo a la base de datos de donde recuperará los datos del usuario. Si la cuenta y la password no son correctas se presentará un aviso de error, esto puede estar motivado porque el usuario no esté registrado en el sistema o porque exista pero se encuentre bloqueado. Si es correcta se informará al usuario que se ha validado correctamente y se creará un carrito de la compra (vacío), asociado al usuario validado.

En el siguiente diagrama de secuencia se representa la lógica detallada correspondiente a la compra de un libro, su incorporación al carrito de la compra del usuario y la confirmación final de las compras realizadas.

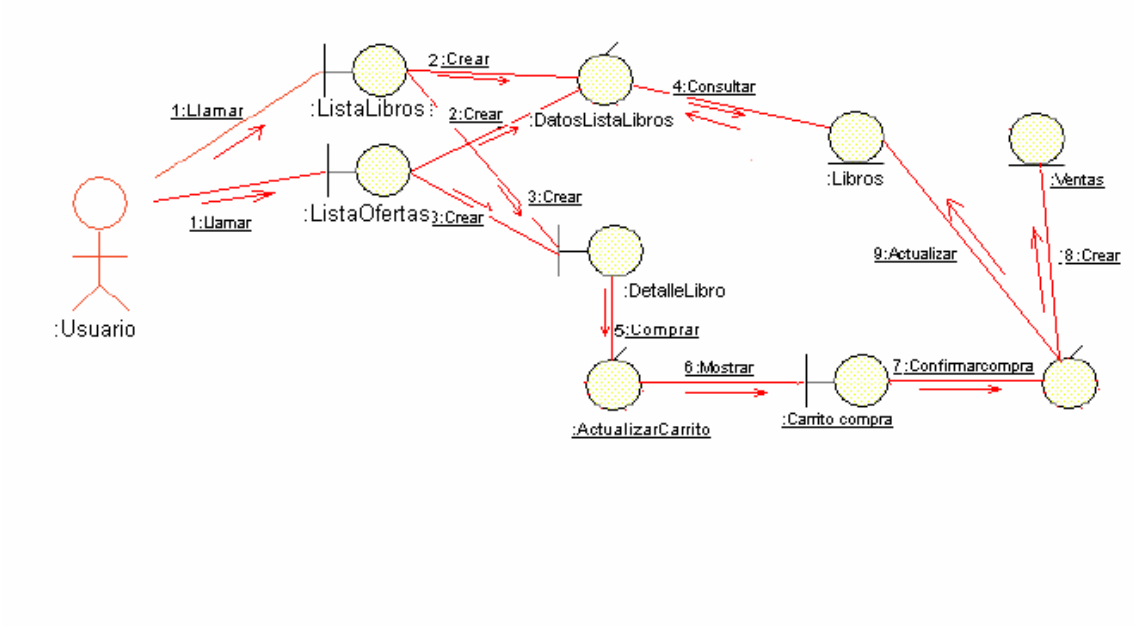


Ilustración 5: Diagrama de colaboración compra y confirmación

Un usuario selecciona un libro bien buscándolo en el catálogo general o bien consultando los libros en oferta. Una vez seleccionado el libro, el usuario pedirá que se muestre su detalle donde se podrá comprar el libro. Cuando se compra un libro se incorpora al carrito de la compra del usuario Desde el carrito de la compra el usuario podrá confirmar la compra. En ese momento se actualizará el stock del libro y se anotará la venta.

4.2.4. Diagramas de secuencia

En el siguiente diagrama de secuencia se representa la lógica detallada correspondiente a la compra de un libro, dado que es este el proceso más complejo del sistema.

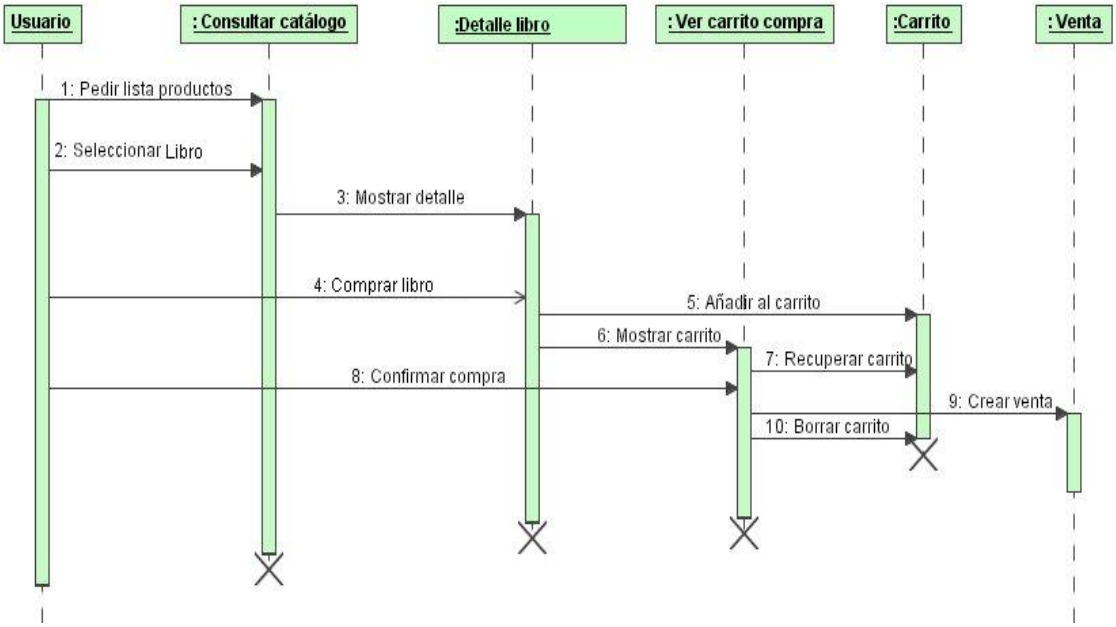


Ilustración 6: Diagrama de secuencia compra y confirmación

Los datos del carrito de la compra de cada usuario se mantendrán en su sesión (en memoria). No se materializarán en la base de datos hasta que el usuario confirme la compra. Esto se hace así dado que en aplicaciones de este tipo es muy común que los usuarios prueben a comprar un libro y posteriormente abandonen la compra sin previo aviso, por lo que se estarían gastando recursos inútilmente.

4.2.5. Diagrama estático de clases

El siguiente diagrama estático, muestra principales clases del sistema.

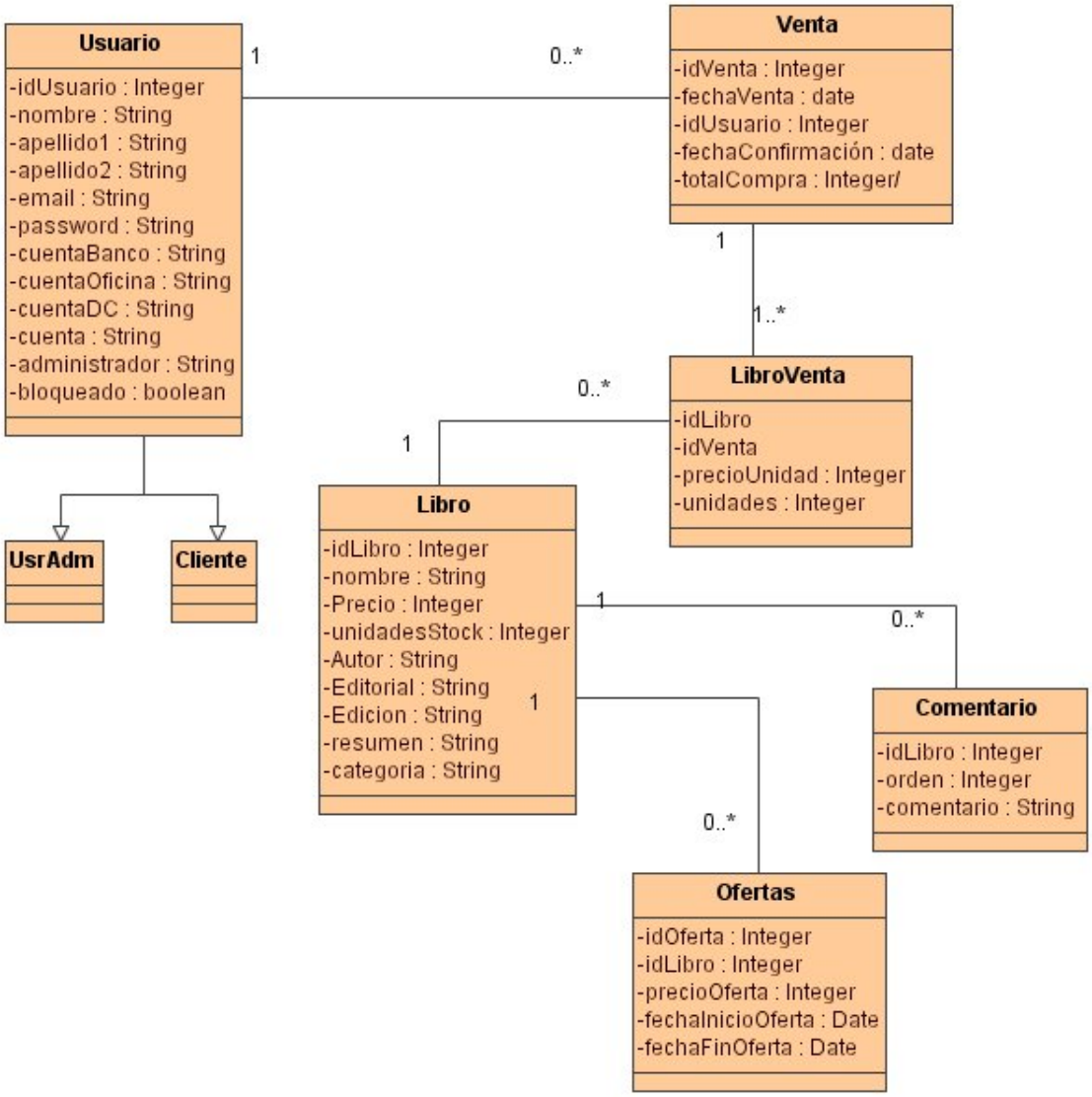


Ilustración 7: Diagrama estático de clases

4.2.6. Análisis de la interfaz de usuario

Pantalla catálogo de libros

Ilustración 8: Pantalla catálogo de libros

Esta configuración de pantalla será común para todos los casos en los que se muestre la lista de libros, tanto para el catálogo general de libros como para la lista de libros en oferta. Para cada uno de los libros presentados se dispondrá de los siguientes iconos:

Acceder a la ficha de detalle del libro

Incorporar al carrito de la compra

Indicador de que el libro está en oferta

En la esquina superior derecha se encuentra un cuadro donde los usuarios registrados podrán introducir su cuenta y su password para identificarse en el sistema y así poder comprar libros. También en este cuadro se dispondrá de una opción para que se pueda registrar un nuevo usuario.

Cuando un usuario esté validado en el sistema, aparecerá su nombre y apellidos en la línea de color rojo de la parte superior de la pantalla y se inhabilitarán los botones de "Validar" y de "Nuevo usuario" y se habilitará el botón de "terminar". Si no hay un usuario validado simplemente aparecerá "Usuario visitante" en la línea de información del usuario.

Información sobre libros

Ilustración 9: Pantalla ficha información de libros

Al pulsar el icono de información se presenta esta pantalla. Desde ella, además de poder consultar toda la información referente al libro se podrá "Comprar" el libro e incorporarlo al carrito o Introducir un comentario al libro mediante el enlace "Comentario".

Los iconos de comprar y comentarios, sólo estarán activos para usuarios registrados.

Registro de nuevo usuario

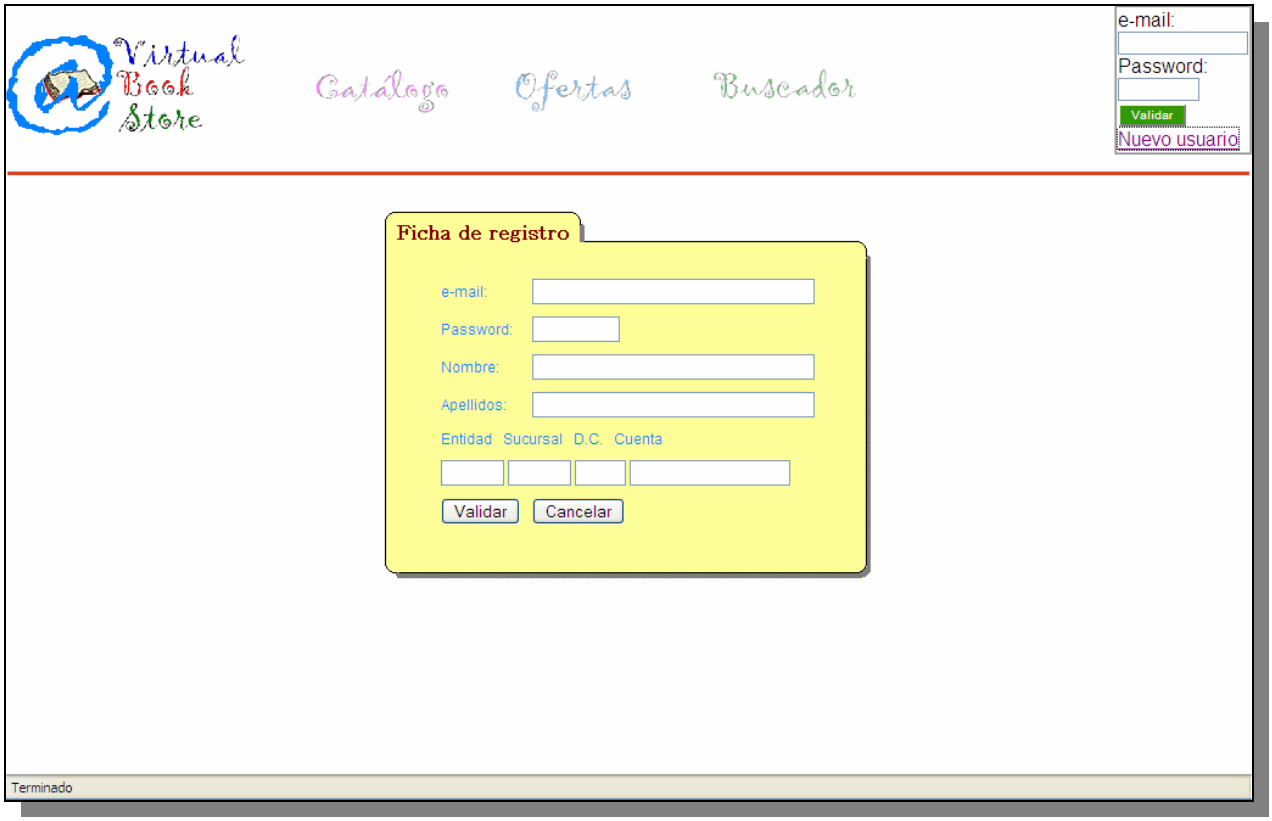


Ilustración 10: Pantalla registro nuevo usuario

Esta ventana se presentará cuando se pulse el botón de Nuevo usuario.

Se deberán completar todos los datos de registro. En caso contrario se presentará una pantalla de error.

Igualmente se presentará una ventana de error si el usuario ya existe.

Una vez completados los datos correctos del nuevo usuario, se registrarán en el sistema y se mostrará una pantalla de confirmación del registro.

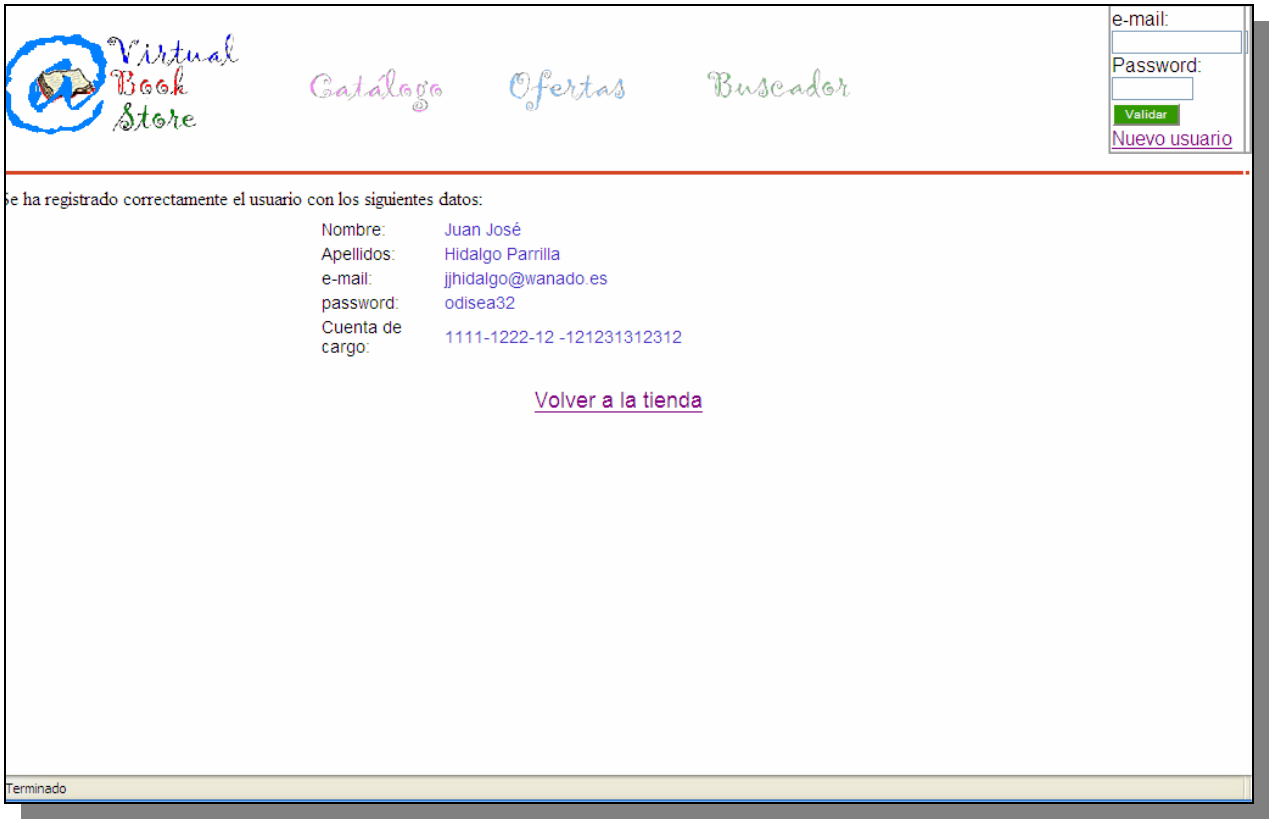


Ilustración 11: Pantalla Confirmación registro de usuario

Carrito de la compra

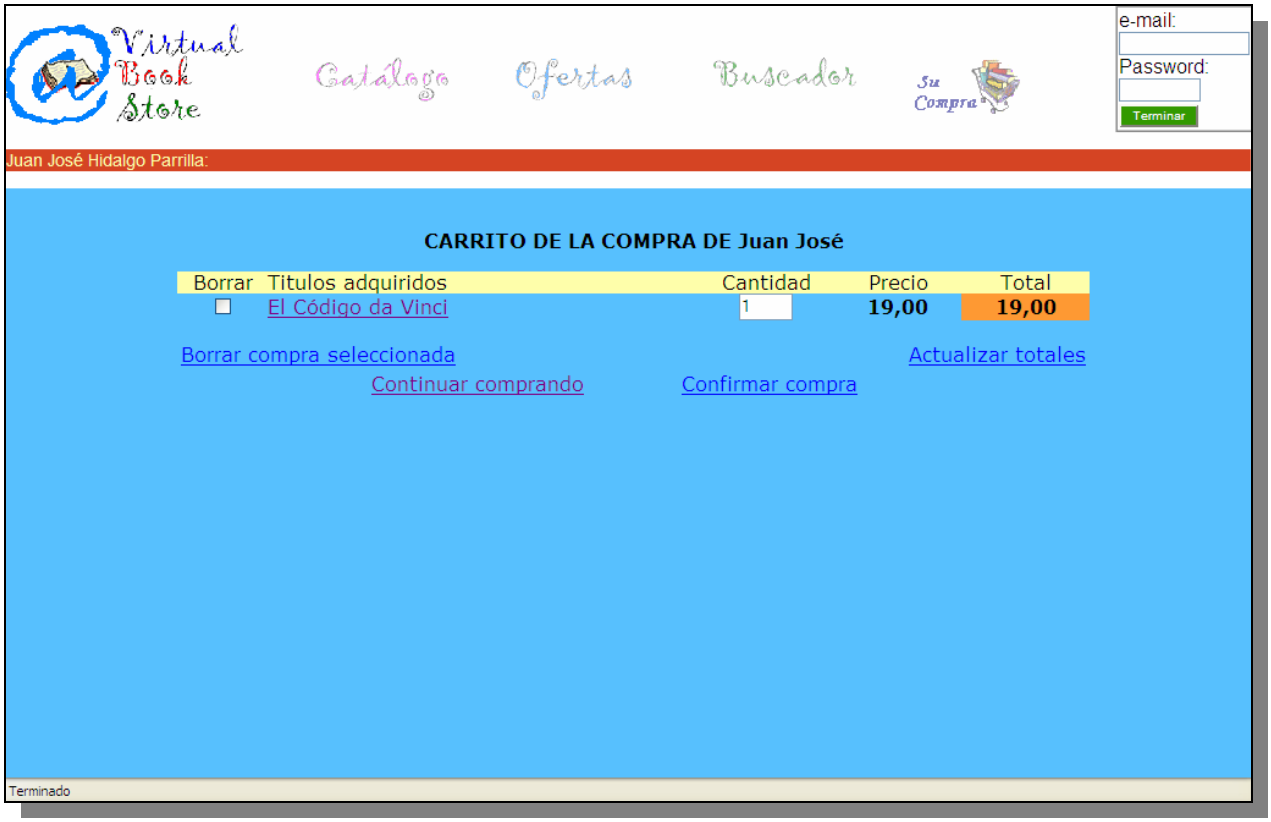


Ilustración 12: Pantalla carrito de la compra

En cualquier momento un usuario puede ver el estado de su carrito de la compra pulsando el botón . Su Compra

En esta pantalla podrá ver cada uno de los libros adquiridos, su precio por unidad y el total de cada compra.

Marcando el check de borrar, situado a la izquierda de cada compra y pulsando el enlace Borrar compra se eliminará la compra seleccionada del carrito.

Pulsando el enlace Continuar comprando volveremos al catálogo de libros, donde se podrá realizar otra compra.

Pulsando el enlace Confirmar compra concluiremos la compra.

Buscador de libros

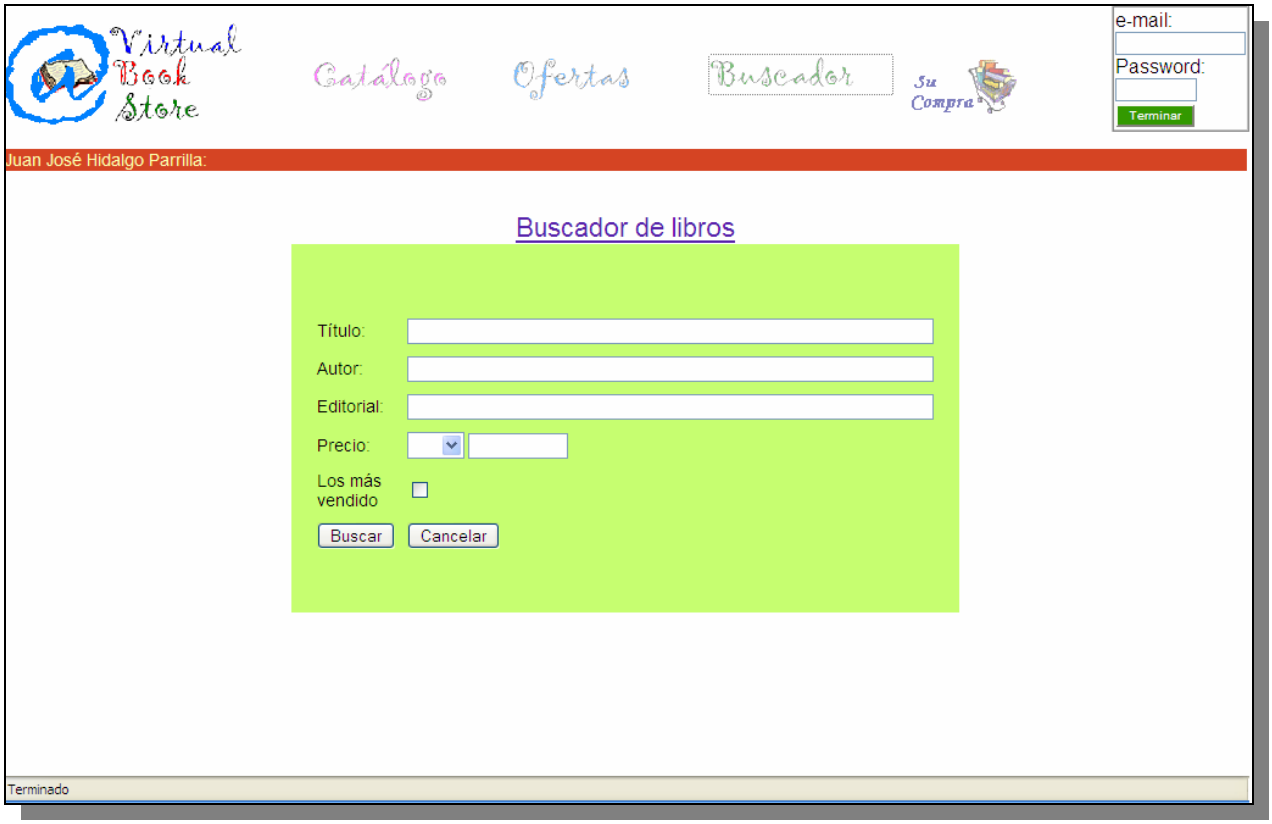


Ilustración 13: Pantalla buscador de libros

Pulsando la opción “Buscador”, presentará está pantalla donde se podrá introducir las características del libro a buscar.

El resultado con el libro o libros que cumplan con los criterios de búsqueda indicados se presentaran en pantalla al pulsar el botón de buscar.

Pantalla modificación catálogo

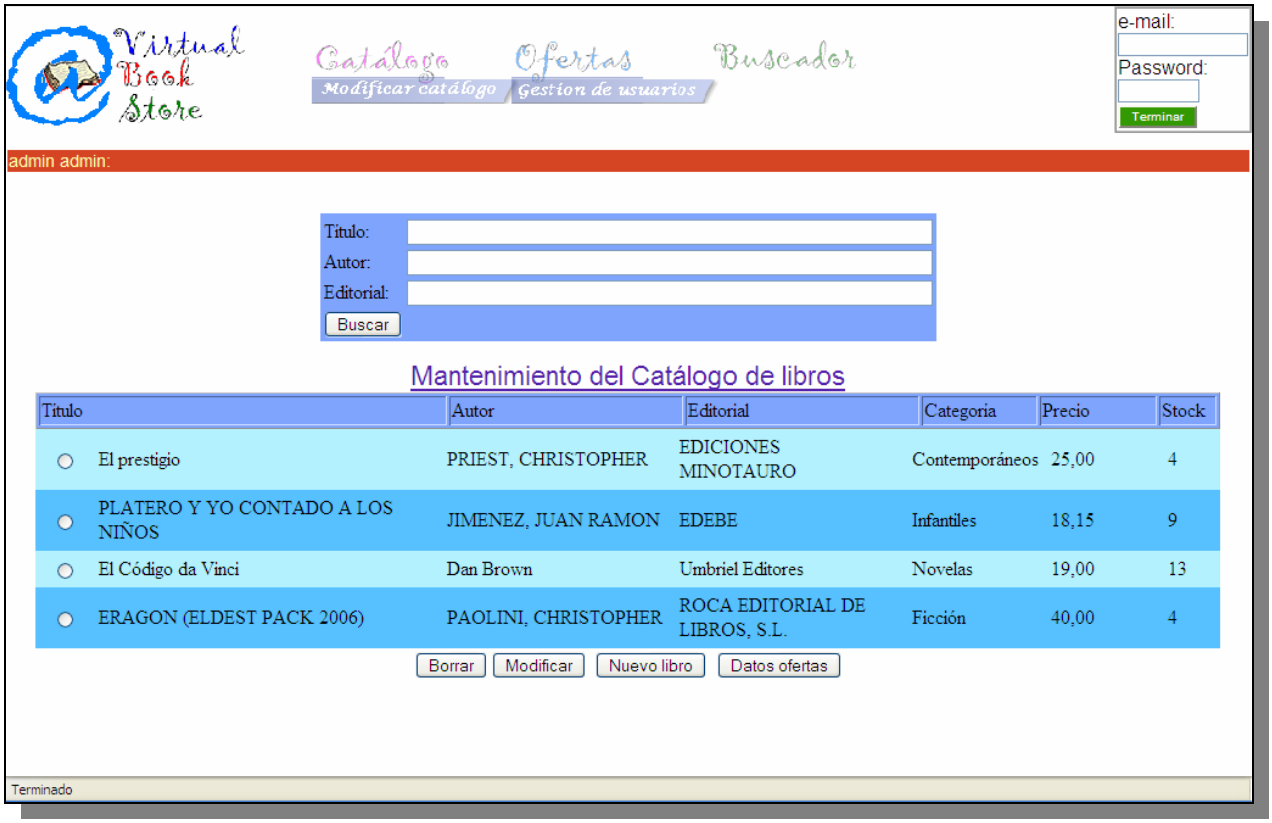


Ilustración 14: Pantalla modificación catálogo

Cuando el usuario validado sea del tipo administrador, en la pantalla se cambiará el menú habitual para los usuarios del sistema al menú de administración que se muestra en esta ventana, con las opciones de “Modificar catálogo” y “Gestión de usuarios”.

La ventana presentada corresponde a la opción “Modificar catálogo” desde la que se podrán modificar cualquiera de los datos de un libro accediendo a su ficha informativa (Ver pantalla más arriba), para lo cual se marcará el libro a modificar y se pulsará el botón Modificar.

Si lo que se quiere es borrar definitivamente el libro del sistema, se marcará el libro a borrar y se pulsará el enlace Borrar.

Pulsando el enlace Nuevo libro se presentará una ventana donde se podrá dar de alta un nuevo libro en el catálogo.

Pulsando el botón Datos Ofertas se presentará la lista de los periodos en los que el producto estará en oferta.

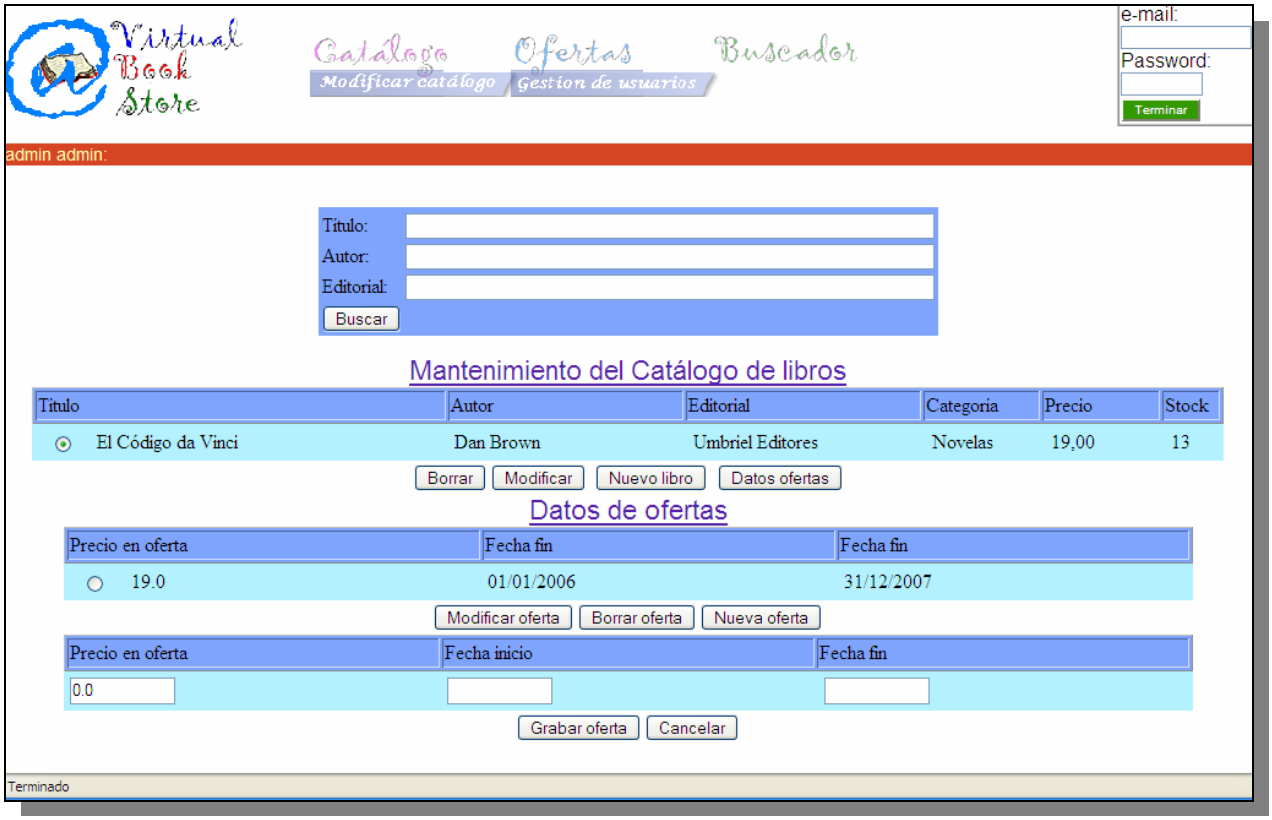


Ilustración 15: Pantalla Datos de ofertas

Bajo la lista de datos de ofertas se presentan una serie de botones para modificar, borrar o introducir una nueva oferta.

Al seleccionar una oferta y pulsar modificar se presentarán sus datos en la parte inferior de la pantalla y para crear una nueva oferta se presentará en la parte inferior de la pantalla los mismos campos que para modificar pero en blanco.

Mantenimiento de usuarios

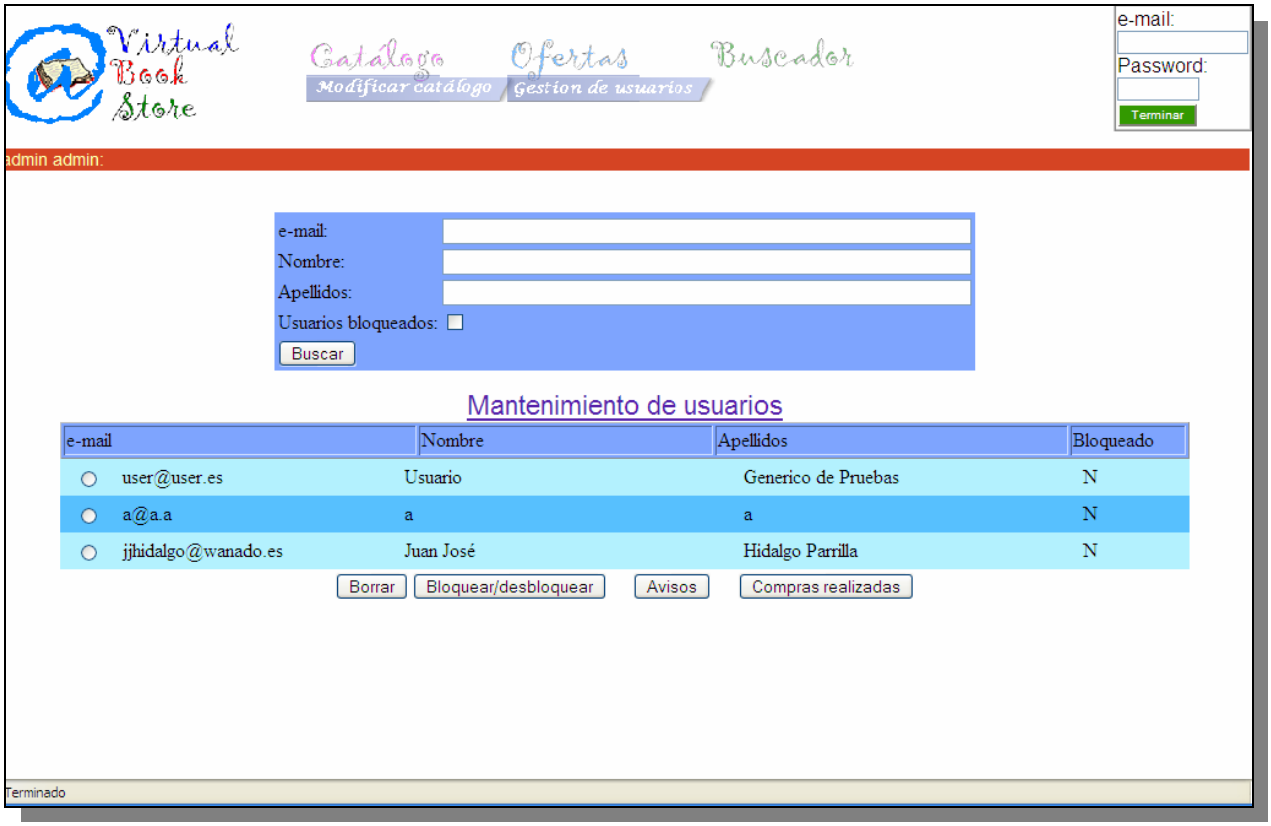


Ilustración 16: Pantalla gestión de usuarios

Esta pantalla corresponde con la opción de "Gestión de usuarios". El funcionamiento es similar a la pantalla de "Modificación catálogo". Se selecciona el usuario sobre el que se quiere actuar y se pulsa el enlace Borrar para eliminarlo, Bloquear/Desbloquear para bloquear el usuario pero sin borrarlo del sistema, o desbloquear si está bloqueado, Avisos para introducir un texto que le aparecerá al usuario al validarse en el sistema y Compras realizadas que presentará una lsita donde se podrán ver todas las compras efectuadas por el usuario.

En la siguiente pantalla se muestra el contenido en el caso de pulsar el botón de Avisos. Se presenta entonces la lista de avisos y los botones que permitirán borrar avisos o crear nuevos avisos.

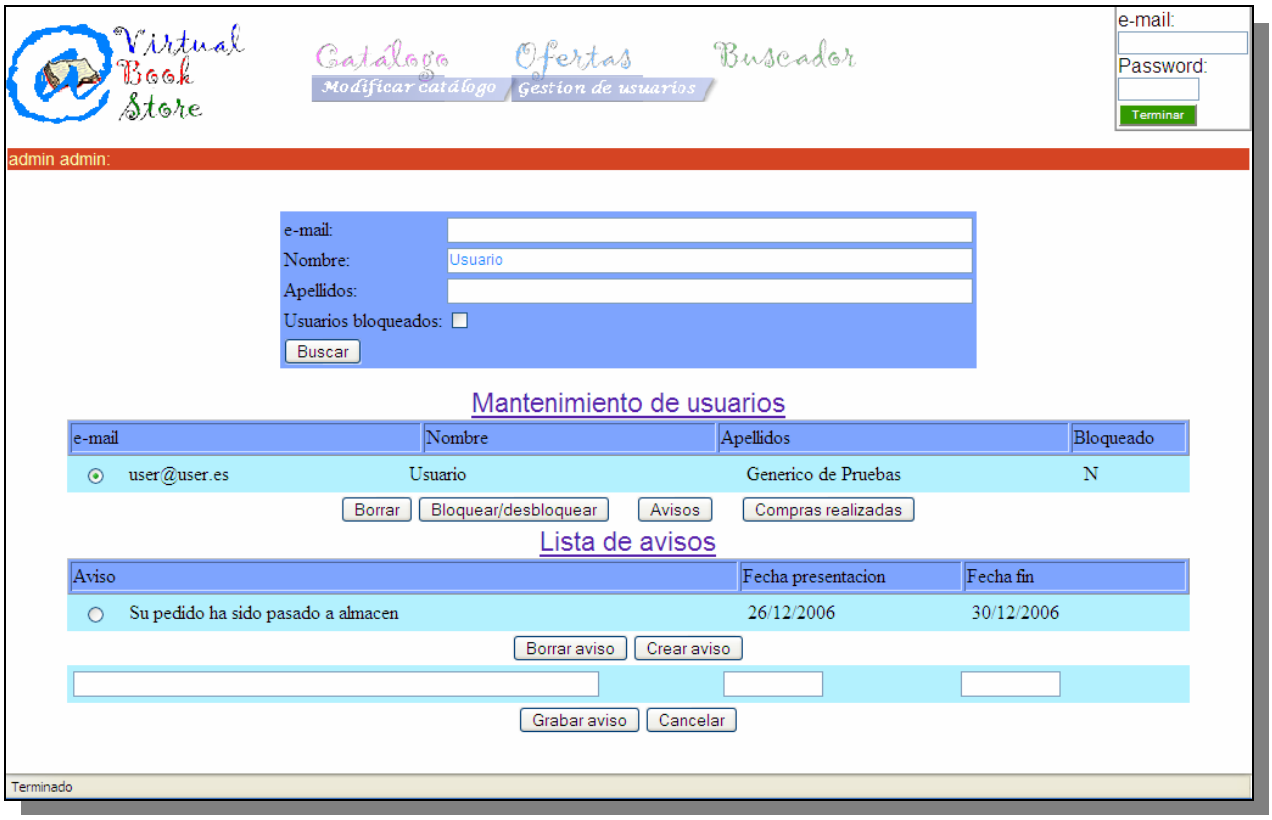


Ilustración 17: Pantalla lista de avisos

4.3. Capítulo 3: Diseño

4.3.1. Diagrama estático de diseño

En la revisión del diagrama estático de clases se han seguido las siguientes pautas:

1. Normalización de nombres. En este caso no ha sido necesario modificar ninguno de los nombres existentes en el diagrama.
2. Reconsideración de la necesidad de algunas clases.
   - La clase UsrAdm y Cliente tienen como objetivo diferenciar entre los usuarios con perfil de administrador del resto. Dado que no se prevé ningún atributo particular para cada una de ellas, se eliminan y se pone un nuevo atributo en la clase padre (administrador con valor 'S' cuando en usuario tenga este roll).
3. Adaptación de la herencia múltiple. No existe herencia múltiple.

4. Transformación ("reificación") de las clases asociativas. No es necesario

5. Eliminación de las asociaciones n-arias (n>2). No es necesario

El diagrama de clases quedará como sigue:

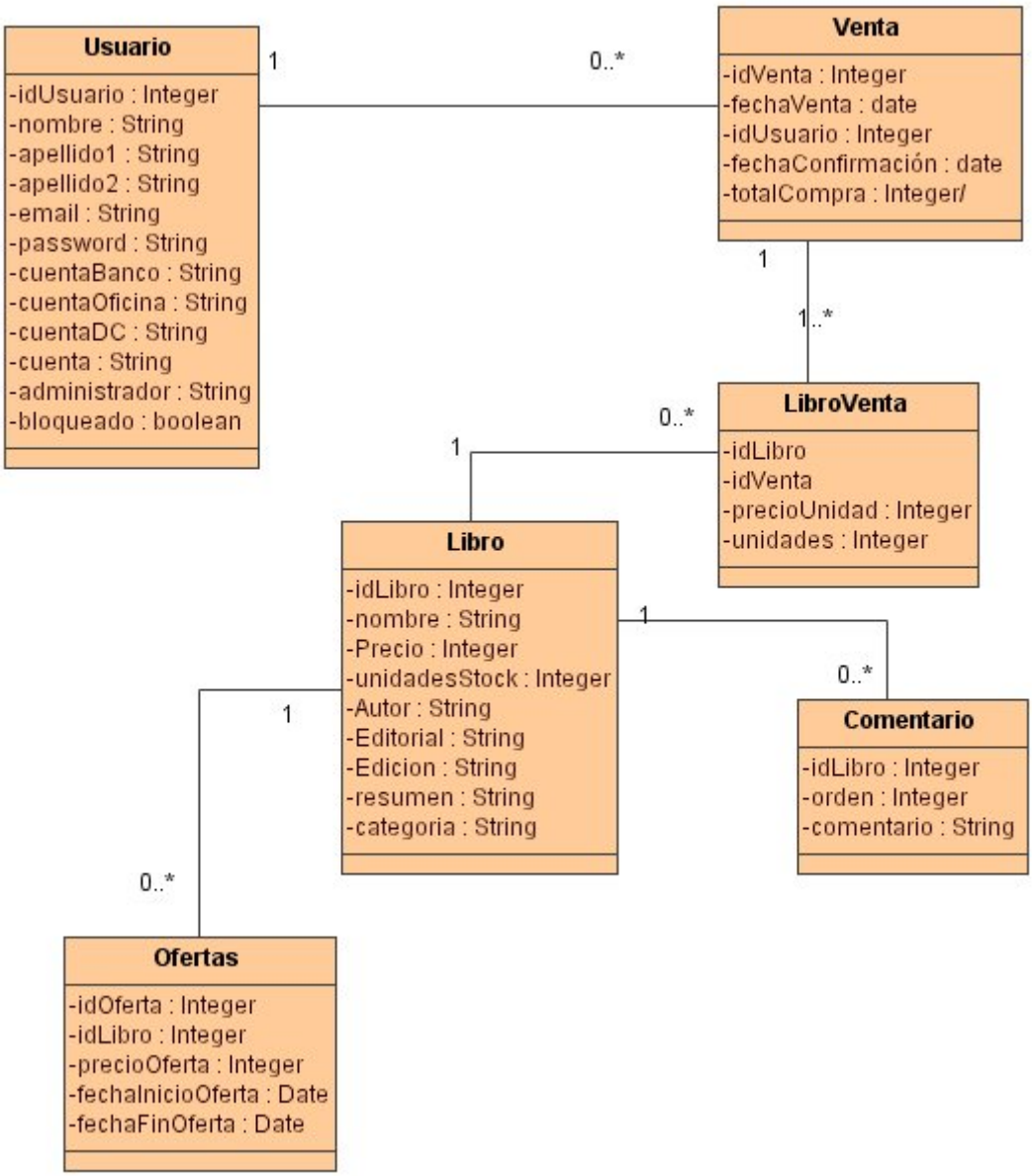


Ilustración 18: Diagrama de clases

4.3.2. Diseño de la base de datos.

Partiendo del diagrama estático de diseño, y dado que no existen casos a tratar de herencia, se obtienen las siguientes tablas:

En cada tabla aparecen subrayados los campos que componen la clave primaria y se indica las claves foráneas utilizadas. Las tablas resultantes son las siguientes:

Usuarios(idUsuario, nombre, apellidos, email, password, ctaEntidad,ctaSucursal,ctaDc,ctaCuenta,Administrador,Bloqueado)

Libros(**idLibro**, titulo, autor, editorial, edición, categoría, precio, resumen, foto, ventas)

ventas(**idVenta**, fechaVenta, idUsuario, fechaConfirmacion, totalCompra)
{idUsuario} es clave foránea de Usuario

LibrosVentas(**idLibro**, **idVenta**, precioUnidad, unidades)
{idVenta} es la clave foránea de Venta

Comentarios(**idLibro**, **orden**, comentario)
{idLibro} es la clave foránea de Libros

Ofertas(**idOferta**, **idLibro**, precioOferta, fechaInicioOferta, fechaFinOferta)
{idLibro} es la clave foránea en Libros

Categorias(**idCategoría,** Categoría)

Como sistema de gestión de base de datos se utilizará MySql.

MySql es un sistema de gestión de bases de datos relacionales multihilo, multiusuario , fácil de usar y es software libre.

4.3.3. Diseño arquitectónico

El sistema a desarrollar seguirá una arquitectura en tres capas donde estarán claramente separadas las capas de presentación o vista, la encargada de implementar la lógica del negocio y la capa encargada de la persistencia.

Comenzaremos con una explicación general de las diferentes soluciones tecnológicas que se aplicarán en la diseño de la tienda virtual y posteriormente veremos los detalles de su implementación.

Visión general de la arquitectura.

Se pretende conseguir un diseño bien estructurado basado en el modelo MVC (Modelo Vista Controlador) que nos permitan :

- Separar la lógica de negocio de la de presentación.
- Controlar el flujo de la navegación.
- Normalizar las acciones que procesa el sistema.

Respecto a la capa de negocio (Modelo) se utilizará un patrón de diseño conocido como DTO (Data Transfer Objects) también llamado Value Objects (VO). Los DTOs son vistas de los objetos que posteriormente se materializarán en la BD.

Por otro lado se separará la implementación de los procedimientos de acceso a la base de datos, aislando lo máximo posible nuestras clases de negocio de dicha implementación. Para ello se utilizará el patrón de diseño Business Delegate. Con este patrón, crearemos un servicio perteneciente a la capa de negocio, que servirá de nexo de unión con la capa de control.

Para la capa Vista se utilizará Struts. Struts es un framework para aplicaciones web java que implementa el modelo MVC. Se compone de un conjunto de clases y TAG-LIBS que conforman el Controlador, la integración con el Modelo (o lógica de negocio) y facilitan la construcción de vistas.

Capa de presentación: El modelo MVC y Struts

Para la capa de presentación, como ya se ha indicado se ha utilizado Struts como mecanismo que facilite la organización de los elementos que la componen y como modo de tener un control de la navegación a través del aplicativo. Todo esto se lleva a cavo mediante la aplicación del patrón de diseño MVC (Modelo vista controlador).

Este patrón de diseño consiste en dividir las aplicaciones en tres partes bien diferenciadas:

- Controlador: encargado de redirigir las peticiones. Define una especie de “mapa” de correspondencia entre peticiones y respuestas.

- Modelo: es la aplicación que corresponde a una determinada petición (lógica de negocio)
- Vista: es la parte de la aplicación encargada de presentar los resultados de las operaciones (por ejemplo en forma de formularios para que el usuario entre algún tipo de dato).

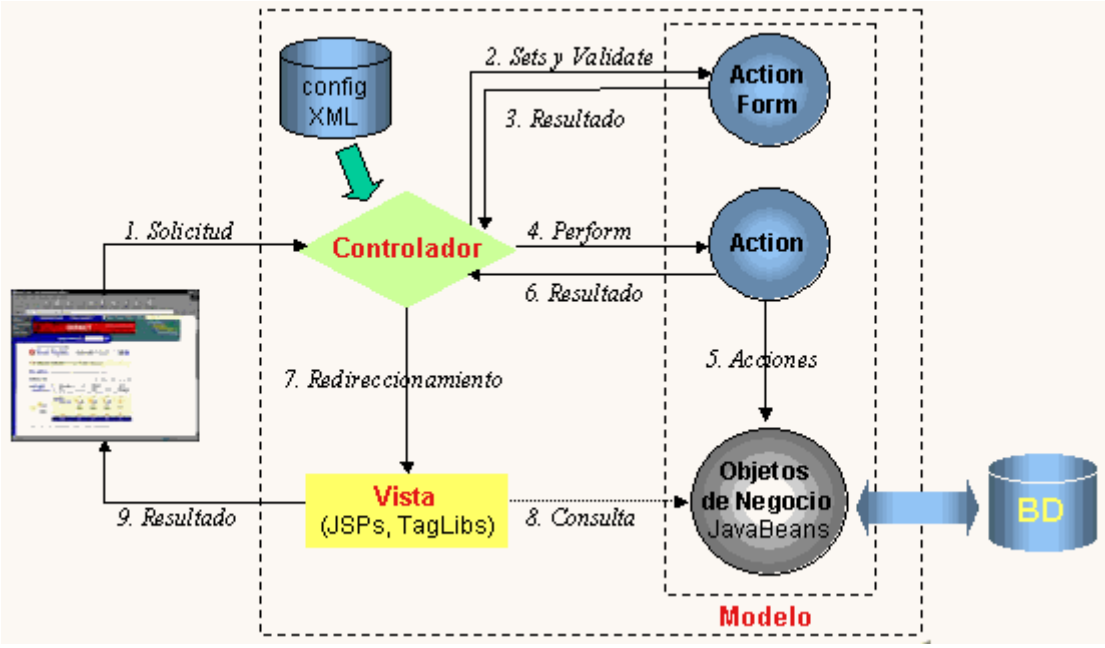


Ilustración 19: Arquitectura Struts

Como se puede ver en el dibujo, Struts necesita comunicarse con la capa Modelo. Esta comunicación debe ser independiente de la tecnología utilizada para la lógica de negocio. Para ello se utilizan DTOs (Value Objects). Con esto Struts recibirá información en forma de vistas. Estas vistas se solicitarán a un servicio perteneciente a la capa de negocio que servirá de nexo de unión con la capa de control.

Para implementar esta abstracción de la capa de negocio se utilizará el patrón Business Delegate.

Lógica de negocio: Patrón Business Delegate

Como se ha indicado el patrón Business Delegate se encarga de ocultar la implementación de los servicios del negocio, reduciendo el número de cambios necesarios en el cliente cuando cambie la implementación de la capa de negocio.

Es decir, permitirá que dicha implementación se pueda realizar utilizando diferentes mecanismos como JDBC, EJBs, HIBERNATE, etc., sin que sean necesarios muchos cambios en el resto del sistema.

La siguiente figura muestra el diagrama de clases que representa al patrón Business Delegate. El cliente solicita al BusinessDelegate que le proporcione acceso al servicio de negocio subyacente. El BusinessDelegate utiliza un LookupService para localizar el componente BusinessService requerido.

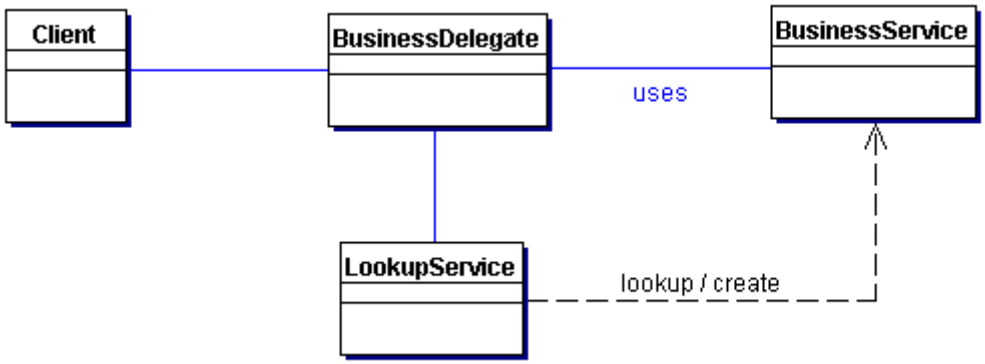


Ilustración 20: Patrón Business Delegate

Persistencia, acceso a datos: Enterprice Java Beans (EJB)

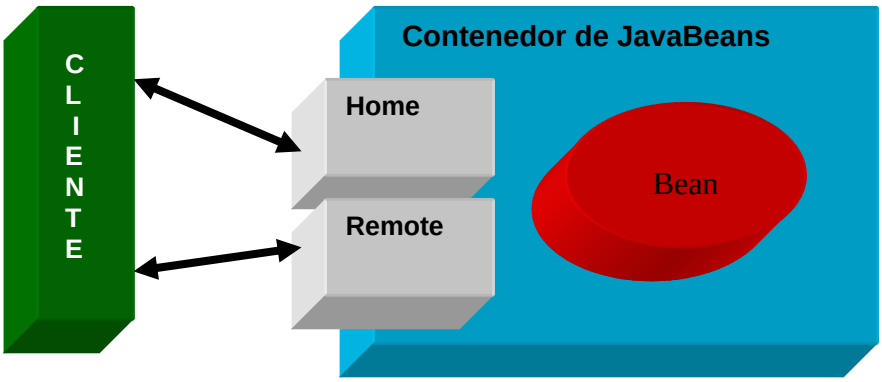
Por último para la capa de persistencia se utiliza EJBs (Enterprise Java Beans).

De forma resumida, se trata de una serie de Beans de entidad que representarán a cada una de las tablas de la base de datos con la que estamos interactuando y que implementan una serie métodos con los cuales se controlará cualquier acceso a la base de datos. Existen dos tipos de Beans de entidad:

- BMP (Persistencia manejada por el Bean): el programador es responsable de escribir en el bean todo el código para acceder a la base de datos.
- CMP (Persistencia manejada por el contenedor): el contenedor se encarga de controlar todos los acceso a la base de datos requeridos por el bean.

En este caso utilizaremos CMP.

Además son necesarios un conjunto de beans de sesión a través de los cuales los clientes accederán a los métodos que recuperarán los datos necesarios de la base de datos para lo cual utilizarán los métodos de accedo implementados en los entity bean.



Hasta este momento este ha sido uno de los métodos más utilizados en el desarrollo de aplicaciones Web con Java, sin embargo en los últimos tiempos va perdiendo terreno a favor de entre otros HIBERNATE.

4.3.4. Arquitectura de la aplicación Tienda Virtual

A nivel de clases el sistema se organizará en varios paquetes que se encargarán de contener las clases que implementarán las capas Modelo y Controlador. La capa vista estará compuesta por un conjunto de páginas JSPs y Html.

Los JSPs (Java Server Pages) se ejecutan en una máquina virtual Java, lo cual permite que, en principio, se puedan usar en cualquier tipo de ordenador, siempre que exista una máquina virtual Java para él. Cada JSP se ejecuta en su propia hebra, es decir, en su propio contexto; pero no se comienza a ejecutar cada vez que recibe una petición, sino que persiste de una petición a la siguiente, de forma que no se pierde tiempo en invocarlo (cargar programa + intérprete). Su persistencia le permite también hacer una serie de cosas de forma más eficiente: conexión a bases de datos y manejo de sesiones, por ejemplo.

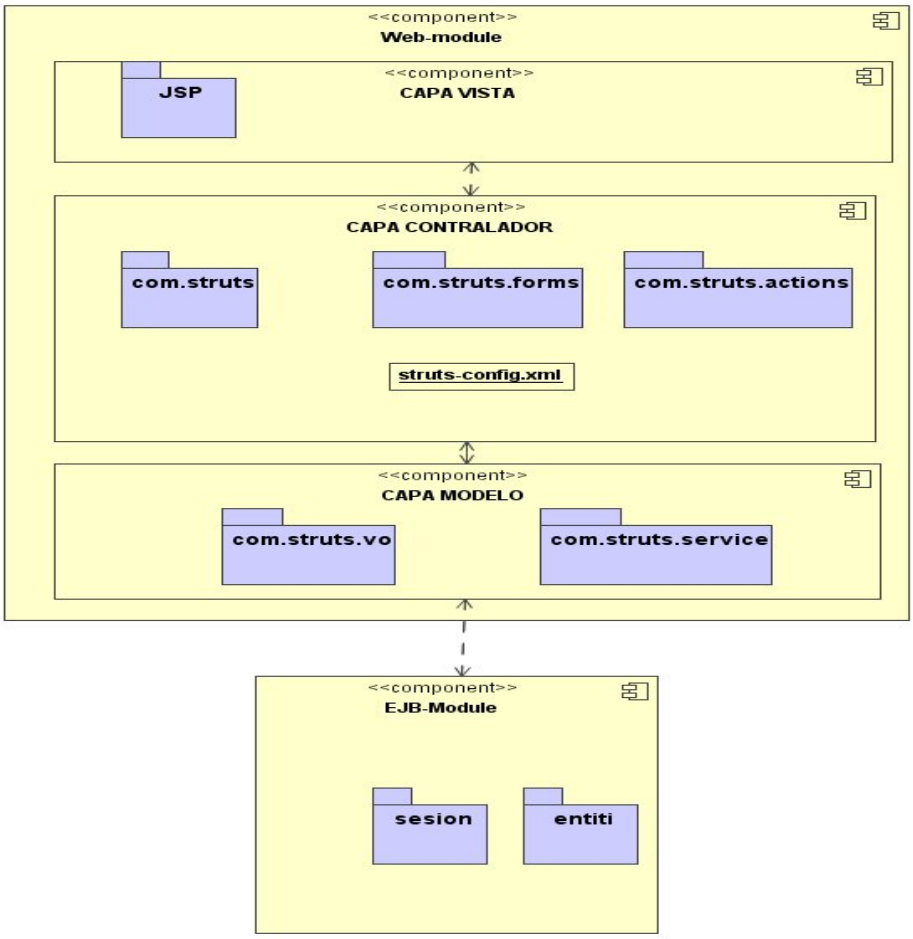


Ilustración 21: Arquitectura tienda virtual

El paquete Web-module estará compuesto por:

- Todos los elementos correspondientes a la parte de presentación (JSP, html, imágenes, etc).
- El paquete com.struts.forms y com.struts.actions se encargan de implementar la capa de Control (los actions de struts) junto con el fichero struts-config.xml donde estará definida toda la navegación del sistema.
- El paquete com.struts.service, com.struts.vo contendrán las clases encargadas de la lógica de negocio. En concreto en com.struts.vo se tienen las clases correspondientes a los DTOs (Value Objects), com.struts.service las clases encargadas de implementar el acceso a la base de datos, para lo cual accederán a las clases de sesion del módulo EJB-Module que implementarán una fachada de los Bean que no pueden ser utilizados directamente por los clientes.

Mediante los DTOs conseguimos una mayor eficiencia ya que se reduce el envío de mensajes por la red, dado que se obtiene una vez la información (por ejemplo del usuario conectado), se pasa a la clase VO (UsuarioVO) y a partir de ese momento para acceder a cualquiera de las propiedades del usuario no se atacará nuevamente a la base de datos, sino que se preguntará a UsuarioVO.

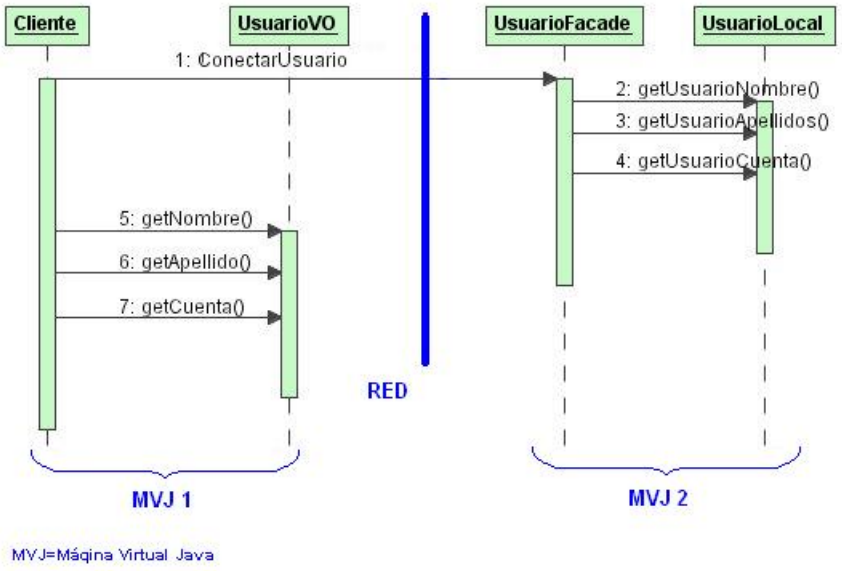


Ilustración 22: Diagrama de secuencia funcionamiento Struts

El siguiente gráfico resume como interactúan entre sí todos los componentes del sistema durante cualquier operación.

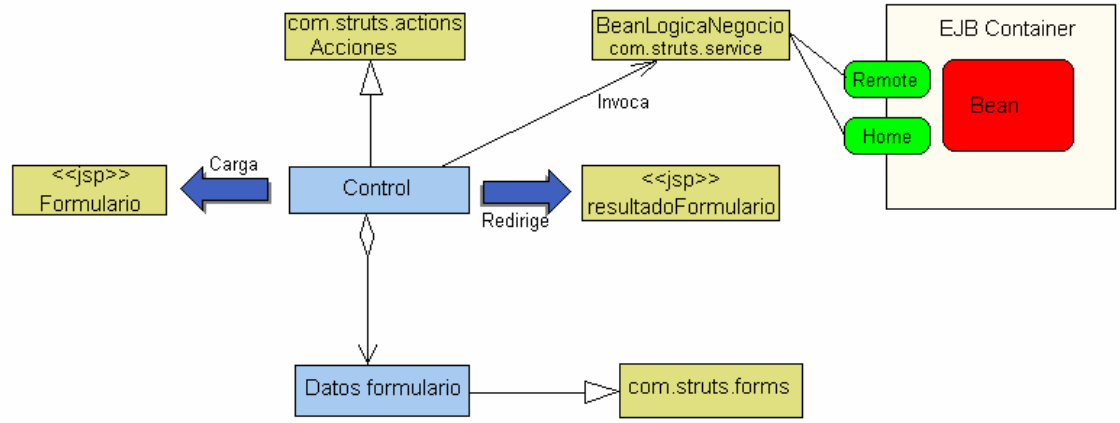


Ilustración 23: Actuación de los componentes de la aplicacion

Una parte importante para el diseño de una aplicación mediante struts es un fichero denominado struts-config.xml y que contendrá el "mapa" de la aplicación en cuanto a los mapeos entre acciones y hacia donde tiene que redirigirse.

En el siguiente diagrama de colaboración se representa el caso de uso correspondiente al registro de un nuevo usuario en el sistema. En el se observa el funcionamiento del sistema y como struts controla todo el flujo, decidiendo en cada caso que elemento es el que tiene que intervenir.

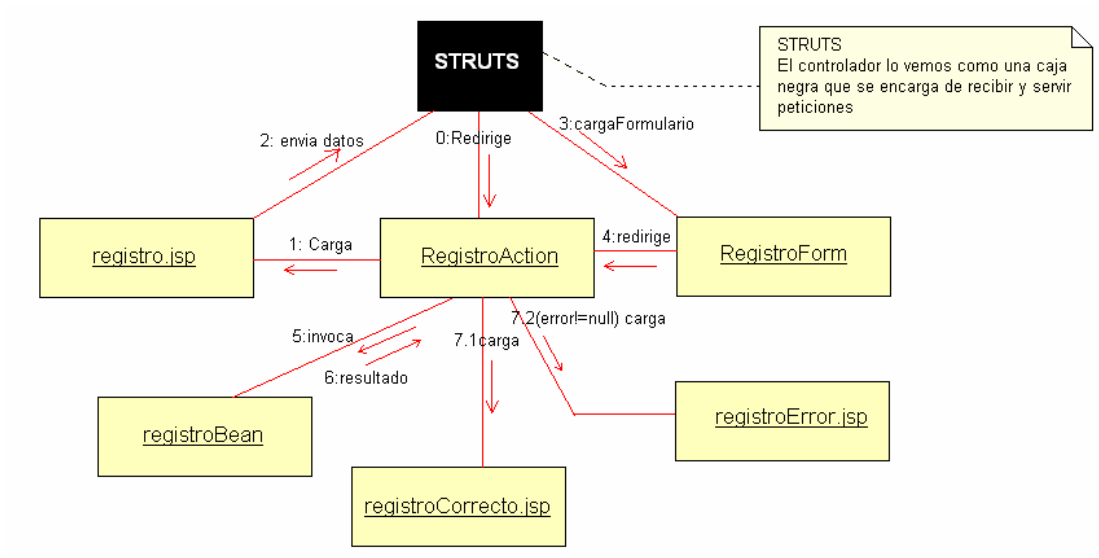


Ilustración 24: funcionamiento Struts

La implementación del carrito de la compra, no tendrá clases persistentes, en su lugar se creará un DTO para el carrito, con una serie de métodos para interactuar con el, otro DTO se encargará de guardar los datos de los libros que se incorporan al carrito, por último en un tercer DTO se mantendrán los datos de la sesión asociada a un usuario donde se tendrán además de los datos del usuario también se guardará la referencia a su carrito de la compra.

Para controlar y centralizar la creación de estos objetos se utilizará una factoría (factory pattern).

En el siguiente diagrama se ven el conjunto de las principales clases que compondrán el sistema. En alguna de ellas se han omitido los métodos get y set que dan acceso a sus propiedades para simplificar la representación.

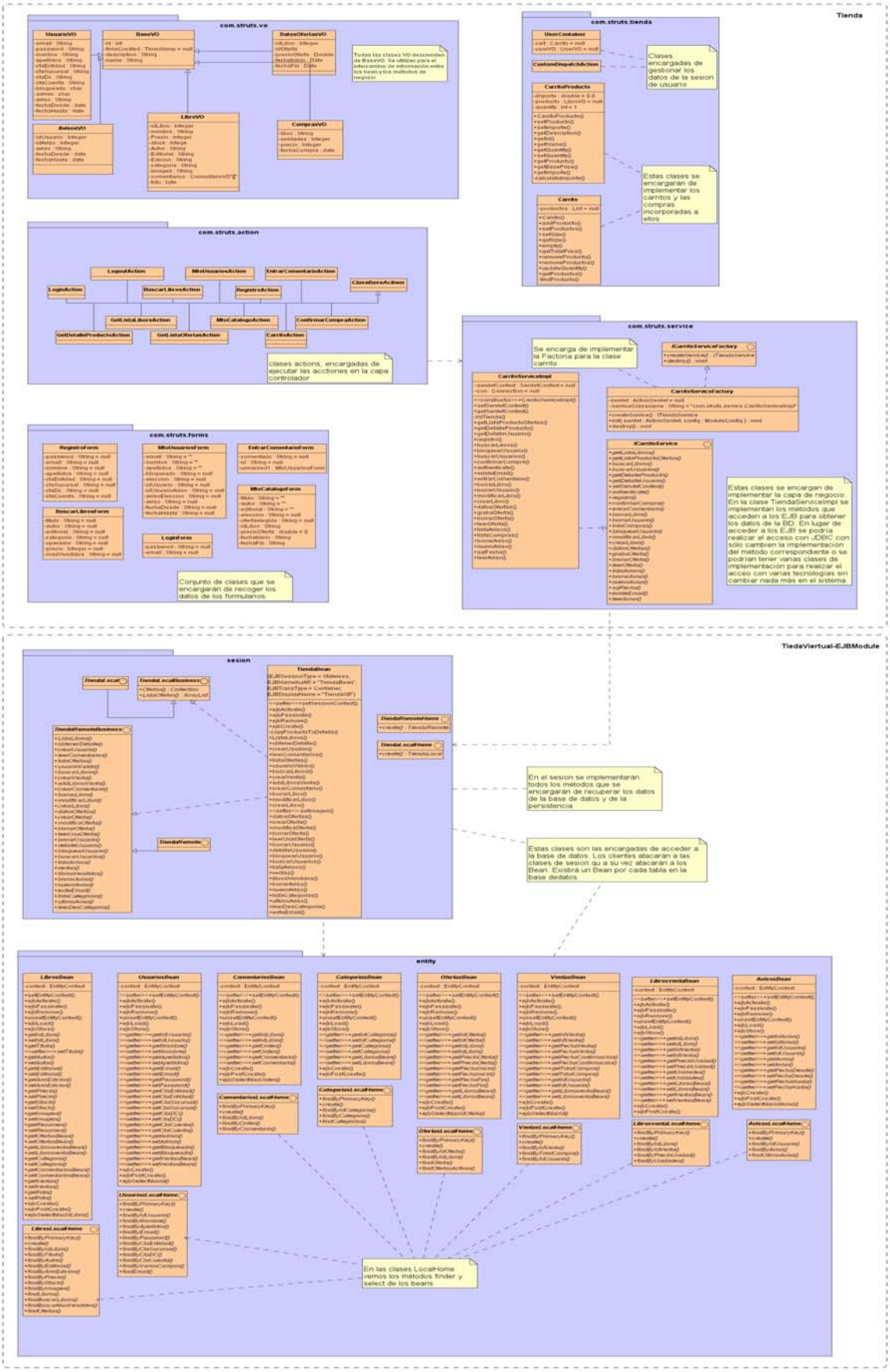


Ilustración 25: Diagrama de clases de Tienda-Virtual

4.3.5. Despliegue del sistema

Dado que vamos a usar Enterprise Java Bean's (EJBs) necesitamos un servidor que soporte esta tecnología.
El desarrollo se efectuará para ser desplegado en JBOSS. JBOSS es una implementación Open-Source de un "EJB Container"

Como se observa en la siguiente gráfica un "Java Application Server" se encuentra compuesto por dos partes: un "Servlet Engine" y un "EJB Engine", dentro del "Servlet Engine" se ejecutan exclusivamente las clásicas aplicaciones Web (JSP's y Servlets) , mientras el "EJB Engine(Container)" es reservado para aplicaciones desarrolladas alrededor de EJB's.

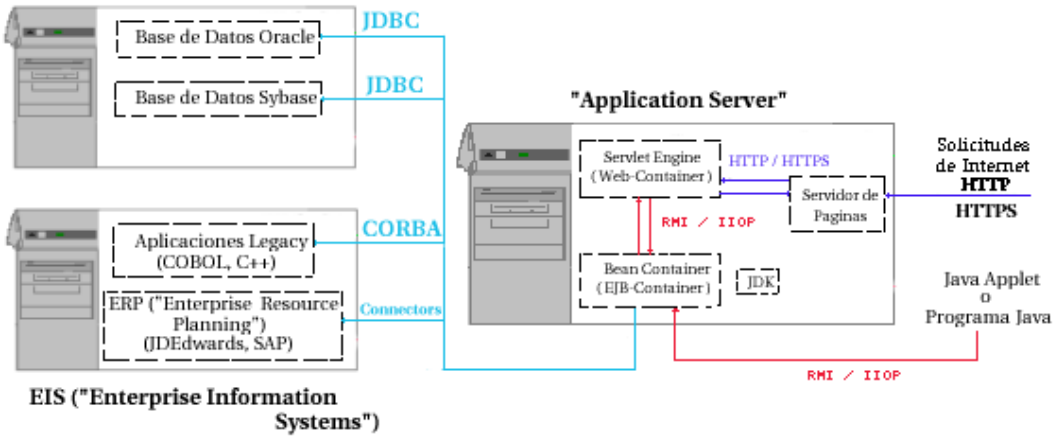


Ilustración 26: Despliegue del sistema

JBOSS incluye un TOMCAT que se encarga de desarrollar las funciones de servidor Web (Web-container).

Todos los componentes del sistema desarrollado estarán empaquetados en un ".ear" que se desplegará en el directorio deploy del JBOSS, además del fichero que se encarga de establecer la conexión con la base de datos mysql-ds.xml.

4.4. Capítulo 4: Ampliaciones del sistema

Ya se ha indicado que el objetivo principal del trabajo no es el obtener una aplicación de tienda virtual 100% utilizable en el mundo real, por ello esta aplicación está limitada en cuanto a posibilidades, sin embargo, con algunas modificaciones si podríamos lograr tener un producto competitivo.

Por el proceso de diseño e implementación llevado a cabo, dichas modificaciones no son costosas de realizar y no afectarían al software ya realizado.

Por ejemplo, si se decide cambiar el menú, que actualmente es estático, por un menú dinámico y con otro diseño, se podría hacer modificando los siguientes elementos:

- Desarrollar la nueva jsp cabecera.jsp, que implementa la presentación del menú y el formulario de login/logout de usuarios
- Implementar la clase actión que de servicio a esta página.
- Implementar los métodos de acceso a la BD tanto en la clase service como los EJB's necesarios y los métodos en la fachada.

Puede parecer complejo, pero se puede observar que en ningún momento se ha modificado nada de lo que ya estaba desarrollado. Todo es nueva implementación con lo que se puede comprobar la independencia de cada uno de los elementos.

4.5. Capítulo 5: Conclusiones.

Son varias las conclusiones a las que he llegado después de este proyecto. Son conclusiones particulares pero que he ido contrastando en algunos casos con desarrolladores y otros profesionales habituados a realizar desarrollos de aplicaciones con esta y otras tecnologías.

Es evidente que Java está enfocado al desarrollo de aplicaciones principalmente destinadas a su utilización en Internet, sin embargo, considero que es complicado desarrollar aplicaciones de gestión complejas, donde sea necesario pantallas muy interactivas, con continuos accesos a la base de datos por tener campos dependientes unos de otros, aunque en este aspecto han surgido lenguajes que aceleran notablemente este tipo de presentación como

Respecto a los EJB considero que son excesivamente complicados y en algunos casos se complica su utilización, por lo que se tiende a un mal uso de algunas de sus propiedades como puede ser la utilización de las relaciones. En cuanto al lenguaje de consulta a BD, el EJBQL, no da toda la potencia que sería de desear para determinadas consultas.

Por lo que he podido leer y ver Hibernate se va abriendo camino a marchas forzadas precisamente porque viene a solucionar algunos de estos inconvenientes, aunque seguramente tendrá otros que no voy a entrar a comentar en este documento.

En general son muchos los elementos que hay que integrar en un proyecto J2EE, Jsp, Servlets, HTML, Hojas de estilo, JAVA, JavaScript, XML, EJB, Jboss y su configuración, base de datos, conectores para jboss-base de datos, etc. Todo esto son elementos que pueden ser fuente de problemas si no se tiene una buena organización, planificación y documentación de la aplicación. Es más necesario que nunca las fases de análisis y diseño y dejar, el menor número posible de decisiones, a la improvisación, para evitar pasos en falso que nos obliguen a dar marcha atrás a desarrollos que pueden afectar a otros y retrasar notablemente el proyecto.

Por último está la herramienta de desarrollo. Al ser muchos los elementos que intervienen en el desarrollo es complicado encontrar una herramienta que te permita controlarlos a todos, y en todo caso es necesario instalar una serie de IDEs que te permitan hacerlos.

5. Glosario

Contenedor EJB	Parte del servidor encargada de gestionar los EJB
DTO	Data Transfer Object (Objeto de transferencia de datos)
EJB	Enterprise Java Bean
Hibernate	herramienta para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones.
JSP	Java Server Page
MVC	Modelo Vista Controlador
Patrón de diseño	Una solución a un problema de diseño no trivial que es efectiva
Struts	Herramientas de soporte para el desarrollo de aplicaciones Web bajo el patrón MVC
Validarse	Identificarse en el sistema mediante un usuario y una password válidas.
VO	Value Object
XML	acrónimo en inglés de e**X**tensible **M**arkup **L**anguage («lenguaje de marcas extensible»), es un metalenguaje extensible de etiquetas
JBOSS	Contenedor de EJB's
TOMCAT	Contenedor Web

6. Bibliografía.

- The J2EE Architect's Handbook. How to be a successful thecnical architect for J2EE applications

Libro con el que repasar los principales conceptos a tener en cuenta en el proceso de ingeniería de aplicaciones J2EE.

- The J2EE 1.4 Tutorial for NetBeans IDE 4.1

Tutorial de uso de la herramienta NetBeans y en concreto del IDE para la desarrollo de sistemas J2EE con EJB's

- EJB Design Patterns, Advanced Patterns, Processes and Idioms

Patrones de diseño.

- www.solomanuales.org

Web que publica manuales escritos por desarrolladores para diferentes tecnologías.

- www.programacion.com

Web donde encontrar artículos y manuales de programación en diferentes lenguajes.

- www.lawebdelprogramador.com

Web donde encontrar artículos de programación

- www.forosdelweb.com

Foro dedicado a la resolución de problemas de programadores con secciones especializadas en Java

- http://dev.mysql.com/doc/mysql/en/SQL_Syntax.html

Web dedicada a Mysql con manuales de instalación, manuales de referencia y otros recursos.

Aunque destaco esta bibliografía y recursos Web, Internet es hoy en día una gran biblioteca donde existen una infinidad de recursos que nos puede ayudar, o en algunos casos confundir, en la realización cualquier trabajo de investigación. Por lo cual la principal referencia es:

- www.google.es