

# **Escuela Profesional de Nuevas Tecnologías**



## **Programación para Internet con Java EE**

### **Proyecto Web – BookStore Online**

#### **Resumen del proyecto**

**Alumno:**

**Carlos Alberto Díaz Barboza**

**Profesor: Antonio Santos Ramos**

**Abril 2012**

# Índice de contenido

## *C*

Clases Empleadas .....	12
------------------------	----

## *F*

Fichero Web XML .....	19
Funcionalidad IBookStore .....	12
Funcionalidad NBookStore .....	13
Funcionalidad WBookStore .....	15
Funcionamiento de la Base de Datos .....	17

## *M*

Mapa del Sitio Web .....	11
--------------------------	----

## *P*

Página Búsqueda Avanzada .....	5
Página Carrito .....	7
Página Categorías .....	6
Página de Resultados .....	10
Página Inicio .....	4
Página Registro Cliente .....	8
Panel Detalle de Libro .....	9
Problemática .....	3

## *S*

Servlets Empleados .....	16
--------------------------	----

## *T*

Tecnologías Empleadas .....	3
-----------------------------	---

## **Problemática**

La empresa BookStore requiere la creación de un portal web para la venta de libros a través de internet porque está perdiendo clientes debido a que la mayoría de las personas prefiere consultar la información de los libros a través de internet en lugar de ir directamente a la librería.

## **Descripción del Proyecto**

Para la elaboración del portal web se requiere cumplir con una serie de requerimientos por ejemplo una zona de usuarios y una de administración, dichos requerimientos se describen a continuación:

Zona de usuarios: en la zona de usuarios se permitirá el acceso al catalogo de libros y deberá tener las siguientes funcionalidades:

- Libros recomendados
- Libros de ofertas
- Libros de novedades
- Zona cliente
- Búsqueda de libros por criterio único
- Búsqueda de libros por criterio avanzado
- Ficha de cada libro
- Selección de libros por categoría
- Carrito de compra

Zona de administración: en esta zona el usuario podrá dar de alta libros, modificar información, precios, etc. Podrá realizar búsqueda del catalogo de libros, así también podrá realizar la eliminación de libros que ya no se venderán. Para ello existirán varios roles y dependiendo de ellos el usuario tendrá ciertos permisos sobre el catalogo de libros, se deberá pasar por una página de logon que validara al usuario. Este desarrollo de este modulo no se llevara a cabo debido a los plazos de desarrollo, se queda como pendiente para una fase posterior del desarrollo.

## **Tecnologías Empleadas**

Se desarrollo esta aplicación web con el patrón de diseño Modelo Vista Controlador (MVC) como base, implementado con el lenguaje de programación java versión 6 así como XHTML 1.1 y hojas de estilo CSS. Como herramienta de trabajo se utilizo el entorno de IDE eclipse empleando los frameworks de desarrollo web Java Server Faces 2.0, PrimeFaces 3.1, Spring Framework 3.1, Ajax y JQuery. La base de datos donde se almacena toda la información es MySQL porque es una base de datos open source, robusta y de fácil manejo.

# Prototipos del Sitio Web

A continuación se visualiza el prototipo final del sitio web BookStore Online, se detalla la funcionalidad de cada pantalla, además se describen algunos aspectos importantes en el uso de las pantallas.

## Página Inicio

La página de inicio muestra un listado de libros que son novedades así como las ofertas que se ofrecen al en el portal para la compra de libros, esta página principal que se muestra cuando se accede al sitio web.

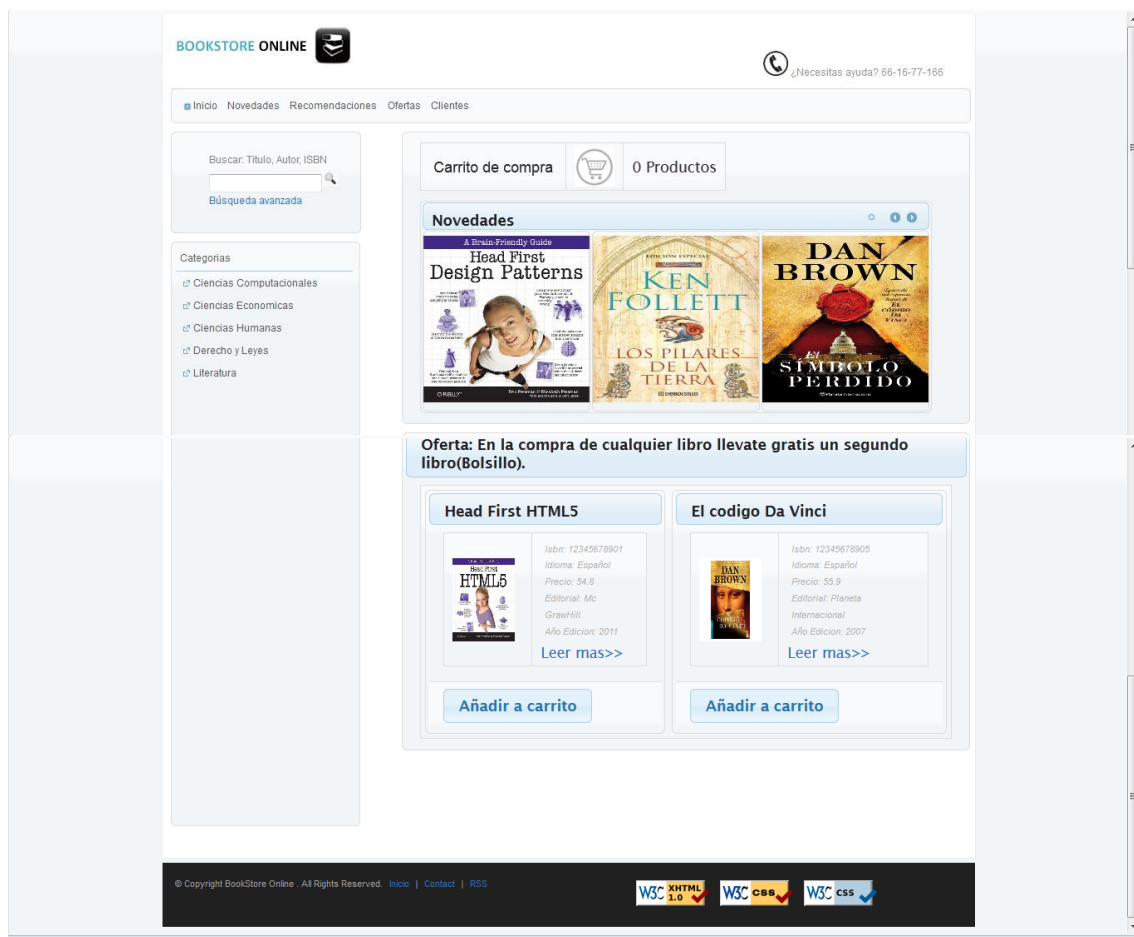


Figura 1. Pagina Inicio. Se muestra la página de inicio del portal web BookStore Online, se muestra un listado de libros de novedades y una selección de libros de ofertas que se tienen a disposición.

## Página Búsqueda Avanzada

La pagina de búsqueda avanzada tiene varios campos por los cuales se puede realizar una búsqueda, ninguno de ellos es obligatorio por lo cual el usuario puede realizar una búsqueda según sus propios criterios.

The screenshot displays the 'BookStore ONLINE' website's advanced search interface. At the top, the site's logo and a contact number are visible. A navigation bar includes links for 'Inicio', 'Novedades', 'Recomendaciones', 'Ofertas', and 'Clientes'. Below this, a search bar prompts the user to 'Buscar: Título, Autor, ISBN' with a magnifying glass icon and a link to 'Búsqueda avanzada'. To the right, a shopping cart icon shows '0 Productos'. The main section, titled 'Busqueda Avanzada', contains several input fields: 'Título:', 'Autor:', 'ISBN:', 'Editorial:', and 'Categoria:'. The 'Categoria:' field is a dropdown menu currently showing 'Selecciona categoria'. Below these fields are 'Limpiar' and 'Buscar' buttons. On the left side of the main section, a 'Categorías' sidebar lists various categories with expandable icons: 'Ciencias Computacionales', 'Ciencias Economicas', 'Ciencias Humanas', 'Derecho y Leyes', and 'Literatura'. The footer of the page includes copyright information, links to 'Inicio', 'Contact', and 'RSS', and three W3C validation logos for XHTML 1.0, CSS, and CSS.

Figura 2. Página Búsqueda Avanzada. Se muestran las opciones por las cuales se puede realizar una búsqueda avanzada.

## Página Categorías

La página de categorías muestra el listado de libros según la categoría seleccionada en el menú izquierdo de la pantalla, se muestra un listado de cuatro libros por paginación, el título del panel de resultados varía según la categoría seleccionada.

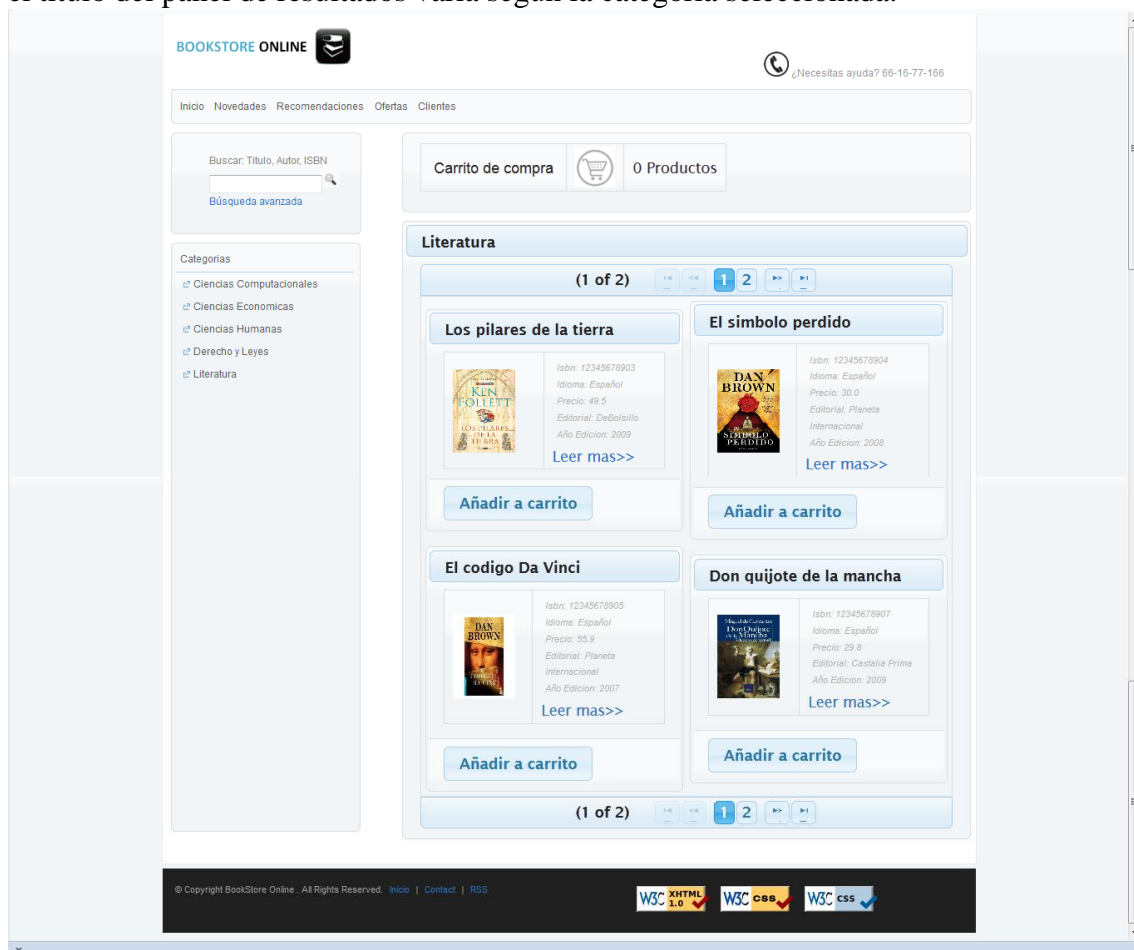


Figura 3. Página de Categorías. Se muestran los resultados de los libros por categorías, según se ha seleccionad del panel de categorías.

## Página Carrito

La página de carrito de compras muestra el listado de los libros añadidos al carrito de la compra, esta página es un paso posterior a la compra ya que se puede indicar el número de libros que se desean adquirir de cada título.

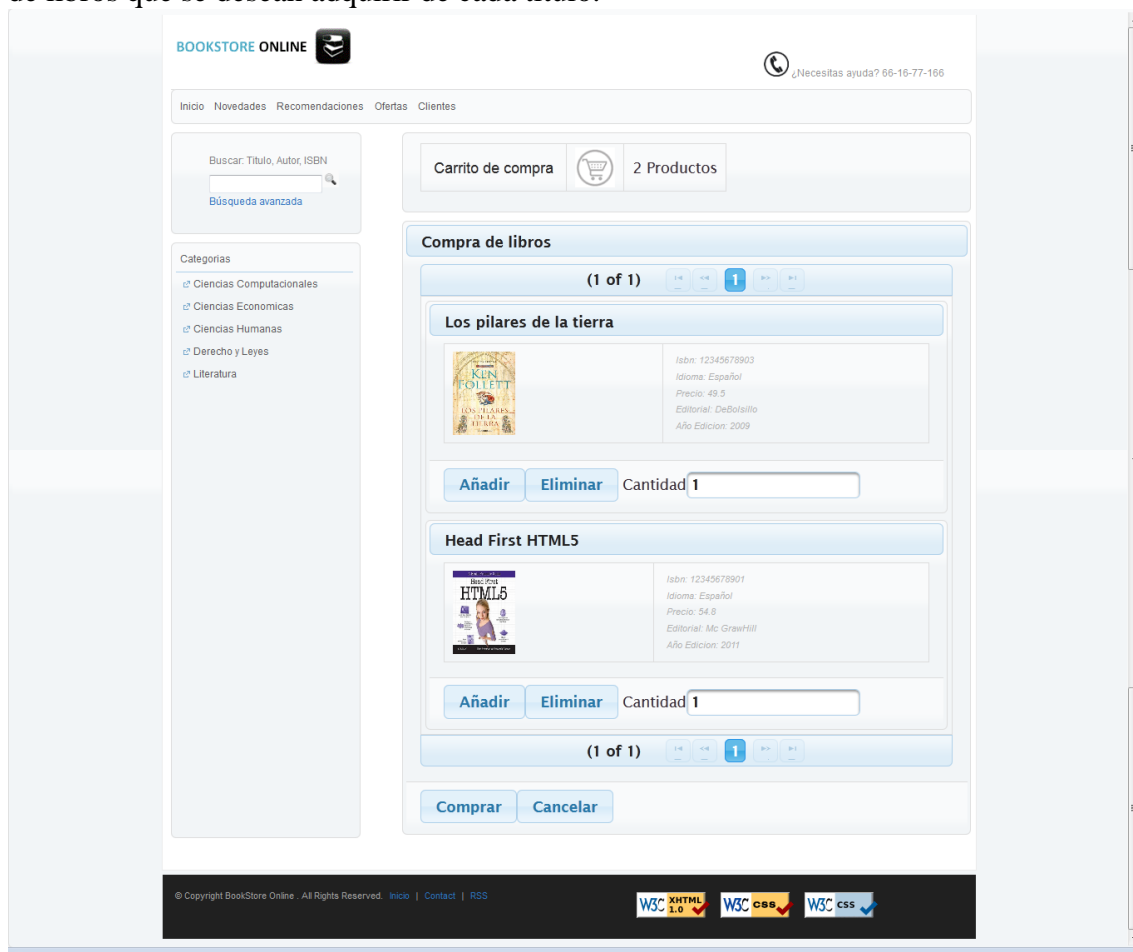


Figura 4. Pagina Carrito de compra. Se muestra el listado de los libros incluidos en el carrito de compras, previo a la realización de la compra.

## Página Registro Cliente

La pagina de registro de clientes es donde se inserta un cliente para lo cual se le piden diferentes datos que cumplimentar algunos de ellos son obligatorios y por lo cual se pide que se llenen en caso contrario se muestra un mensaje de error al tratar de registrarse.

BOOKSTORE ONLINE

Inicio Novedades Recomendaciones Ofertas Clientes

Buscar: Título, Autor, ISBN  
Búsqueda avanzada

Carrito de compra 2 Productos

### Registro

NIF:\*  
Nombre:\*  
A. Paterno:\*  
A. Materno:\*  
Dirección:\*  
Email:\*  
Población:\*  
C. Postal: 0  
Teléfono:  
Intereses Personales:

Limpiar Registrar

© Copyright BookStore Online. All Rights Reserved. Inicio | Contact | RSS

W3C XHTML 1.0 W3C CSS W3C CSS

Figura 5. Pagina Registro Clientes. Se muestran los datos que son requeridos para dar registrarse en BookStore Online.



## Panel Detalle de Libro

Este panel se muestra cuando el usuario presiona el enlace “Leer mas >>” mostrado en las opciones del libro, este panel se despliega como la ficha bibliográfica del libro posicionándose encima de la pantalla, este panel contiene los detalles de la información del libro.

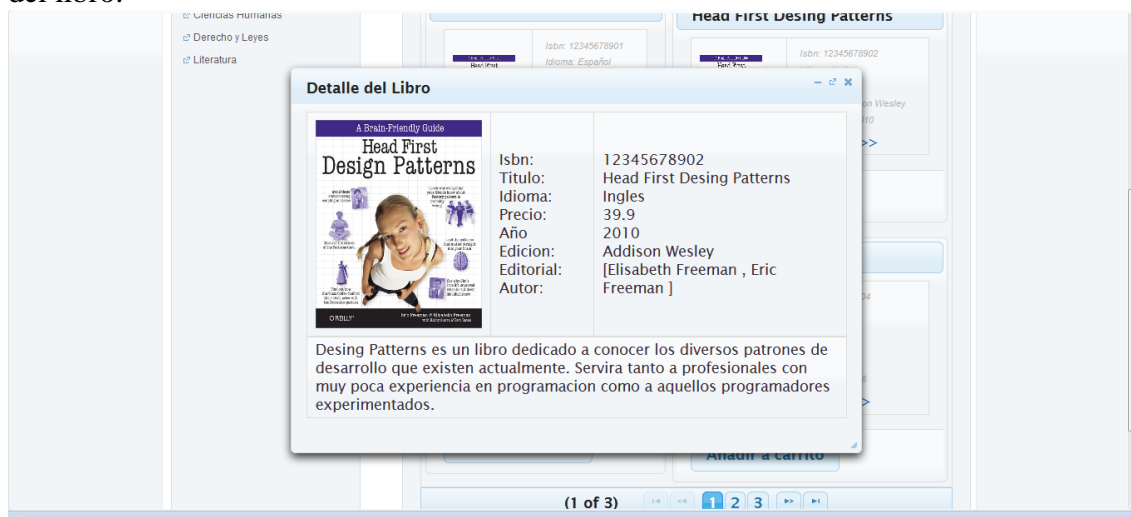


Figura 6. Detalle del Libro. Se muestran los datos del libro de forma detallada.

## Página de Resultados

La pagina de resultados contiene el listado de los libros obtenidos después de realizar una búsqueda, la búsqueda puede ser simple o avanzada, solo se mostraran como máximo cuatro libros por paginación dentro de la pantalla de resultados.

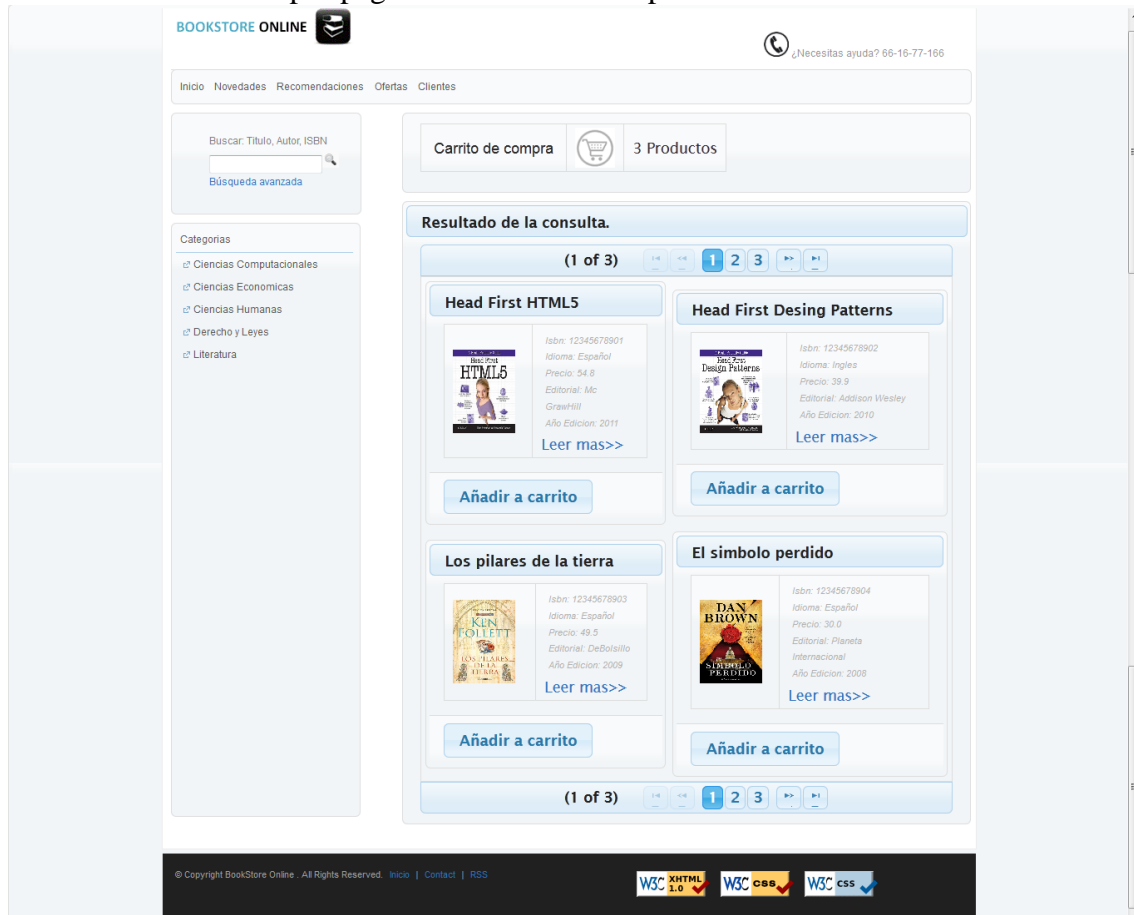


Figura 7. Pagina de Resultados. Se muestran los libros obtenidos de la consulta ya sea por búsqueda avanzada o por búsqueda normal. Los resultados de la consulta se muestran en un listado de 4 libros como máximo, por paginación.

## Mapa del Sitio Web

Se muestra el mapa del sitio web BookStore Online donde se observan las diferentes navegaciones que se realizan dentro de la aplicación, cuando se produce una excepción se navega directamente a la página de Error debido a la cantidad de páginas se ha omitido su flujo ya que esta navegación se configura en el archivo web.xml.

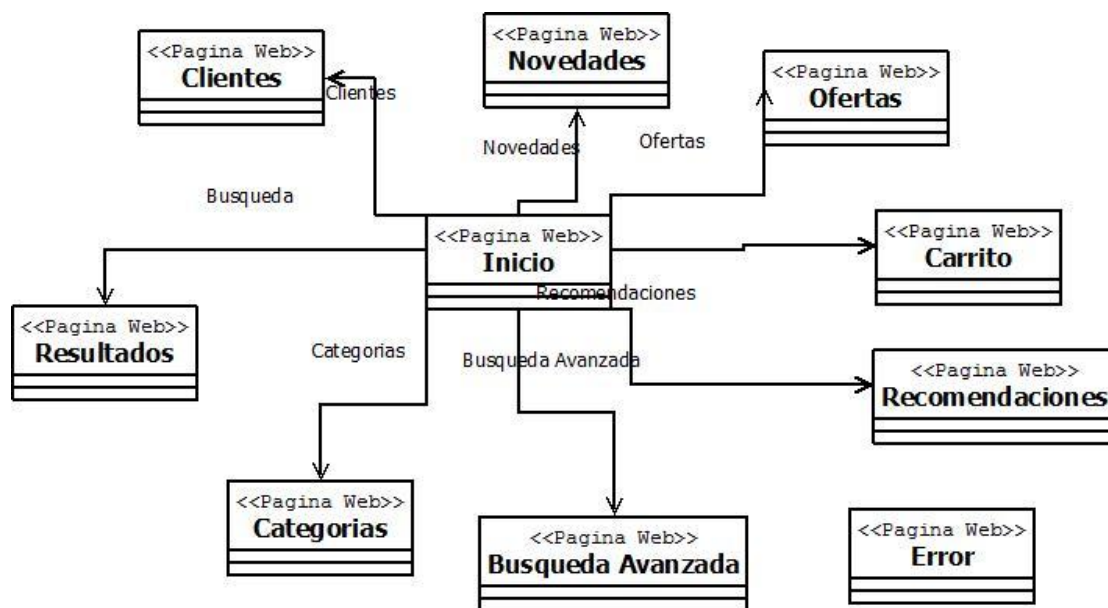


Figura 8. Navegaciones. Se muestra el mapa de navegación del sitio web BookStore Online.

## Clases Empleadas

A continuación se detallan las clases empleadas en el proyecto BookStore Online divididas por funcionales contenidas en su paquete funcional, así como una tabla de resumen donde se describe para que se utilizan esas clases según el paquete donde se encuentran.

### Funcionalidad IBookStore

Contiene la declaración de las interfaces empleadas en el sitio web así como todos los Java Beans que se utilizan para su funcionamiento, de esta forma se realiza una separación de cómo se implementa el negocio, cambiando esta implementación ya que toda la funcionalidad se realiza a través de inyección de beans por medio de las factory beans donde las interfaces son el pilar de la aplicación.

Paquete/Clases	Funcionalidad
<b>com.bookstore.adapter.interfaz</b> IFactoryAdapter.java INegocioAdapter.java	Interfaces de negocio y adaptadores para interacción de lógica del negocio
<b>com.bookstore.business.interfaz</b> INegocioAlmacen.java INegocioCatalogo.java INegocioCliente.java INegocioLibro.java INegocioOferta.java INegocioUsuario.java	Interfaces de los métodos de negocio implementados.
<b>com.bookstore.crud.interfaz</b> ICRUDAlmacen.java ICRUDAutor.java ICRUDCatalogo.java ICRUDCliente.java ICRUDLibro.java ICRUOferta.java ICRUDUsuario.java ICRUDVenta.java	Interfaces de los métodos de persistencia para la conexión con las tablas de base de datos de la aplicación.
<b>com.bookstore.jbeans</b>	Java beans utilizados para la conexión a

JB_Almacen.java	base de datos y utilización en los métodos de negocio de la aplicación.
JB_Autor.java	
JB_Catalogo.java	
JB_Cliente.java	
JB_Libro.java	
JB_Oferta.java	
JB_Usuario.java	
JB_Venta.java	

## Funcionalidad NBookStore

Contiene la implementación de las interfaces declaradas en la funcionalidad IBookStore así como la construcción de la parte de negocio de la aplicación, esta implementación se puede modificar y gracias a la separación de las clases de negocio y la interfaces se aplica el polimorfismo para cambiar funcionalidades de negocio sin necesidad de realizar ningún cambio en alguna otra capa de la aplicación.

Paquete/Clases	Funcionalidad
<b>cfg</b>	Contiene los beans de spring framework creados para la aplicación. Así como properties para la conexión a la base de datos, todos estos archives son cargados en el contenedor de inyecciones cuando se ejecuta la aplicación.
almacenBean.xml	
applicationContext.xml	
autorBean.xml	
catalogoBean.xml	
clienteBean.xml	
configuracion.properties	
daoBean.xml	
factoryBean.xml	
gestionBean.xml	
libroBean.xml	
ofertaBean.xml	
usuarioBean.xml	
ventaBean.xml	
<b>com.bookstore.adapter</b>	Contiene Factorias de beans de spring estas clases son el backbone de la
IFactoryAdaterImpl.java	

INegocioAdapterImpl.java	aplicación. Ya que permite la comunicación de los beans de negocio con la presentación.
<b>com.bookstore.business</b> INegocioAlmacenImpl.java INegocioCatalogoImpl.java INegocioClienteImpl.java INegocioLibroImpl.java INegocioOfertaImpl.java INegocioUsuarioImpl.java	Contiene la implementación de negocio de la aplicación. Este paquete es el core de la aplicación ya que allí es donde se realiza la funcionalidad del negocio.
<b>com.bookstore.config</b> ConfigBeanList.java Configuration.java	Este paquete es de configuraciones aquí se cargan las configuraciones empleadas en la aplicación.
<b>com.bookstore.enums</b> BookStoreEnum.java	Paquete de enumeración empleada en la aplicación.
<b>com.bookstore.factoria</b> FactoryAlmacenDao.java FactoryAutorDao.java FactoryCatalogoDao.java FactoryClienteDao.java FactoryLibroDao.java FactoryOfertaDao.java FactoryUsuarioDao.java FactoryVentaDao.java	Factorys de los beans de spring framework son utilizados para la obtención de los beans de persistencia del negocio.
<b>com.bookstore.gestion</b> GestionBeanAdapter.java GestionFactoryAdapter.java GestionNegocioAdapter.java	Paquete utilizado para la gestión de las factorías de beans.
<b>com.bookstore.mapper</b> AlmacenRowMapper.java AutorRowMapper.java CatalogoRowMapper.java	Paquete de Mapeo de los resulsets obtenidos de la persistencia de las tablas con los java beans de la aplicación.

ClienteRowMapper.java	
LibroRowMapper.java	
OfertaRowMapper.java	
UsuarioRowMapper.java	
VentaRowMapper.java	
<b>com.bookstore.persistencia</b>	Paquete de implementación de las interfaces de persistencia, depende del paquete de mappers ya que se mapean los resultados de las queries a los java beans.
AlmacenDaoImpl.java	
AutorDaoImpl.java	
CatalogoDaoImpl.java	
ClienteDaoImpl.java	
LibroDaoImpl.java	
OfertaDaoImpl.java	
UsuarioDaoImpl.java	
VentaDaoImpl.java	
<b>com.bookstore.test</b>	Paquete de test se emplea para pruebas única y exclusivamente del negocio, esto para pruebas en local sin despliegue de la aplicación en servidores.
TestClass.java	

## Funcionalidad WBookStore

Contiene la presentación de la aplicación web BookStore Online, contiene las páginas empleadas para la navegación y ejecución de la aplicación, así como sus propios Manage beans y baking beans para el control de la navegación.

Paquete/Clases	Funcionalidad
<b>com.bookstore.bbeans</b>	Paquete de Baking beans utilizados dentro de la aplicación, son similares a los java beans solo que estos se emplean para la presentación de la parte web.
BB_Busqueda.java	
BB_Carrito.java	
BB_Cliente.java	
BB_LibrosList.java	
<b>com.bookstore.mbeans</b>	Paquete de manage bean contiene cierta funcionalidad conectándose con la capa de
MB_Carrito.java	

MB_Categorias.java	negocio para la ejecución del mismo
MB_LOferasNovedades.java	sirven de control entre la presentación y el
MB_NavegacionCliente.java	negocio.
MB_Navegaciones.java	
MB_Resultados.java	

## Servlets Empleados

Al utilizar la tecnología java server faces en combinación del framework spring el desarrollo de la conexión entre el negocio y la presentación se simplifica, mas sin embargo se deben emplear los archivos de configuración de cada una de las tecnologías para la gestión de la navegación de la aplicación.

Archivo	Funcionalidad
<b>applicationContext.xml</b>	Archivo de configuración de spring framework, contiene los beans que serán inyectados dentro de las diferentes funcionalidades de la aplicación, escucha las peticiones web y posteriormente sede el control al archivo de configuración java server faces
<b>faces-config.xml</b>	Archivo de configuración de JSF contiene la declaración de los Manage beans y Baking beans empleados en la aplicación así como el tipo de ámbito de cada uno de ellos, gracias a spring framework los beans son inyectados en las clases de manage bean de JSF.



## Funcionamiento de la Base de Datos

Para la comunicación de la base de datos entre la parte web y el negocio se emplean las clases de gestión de factorías descritos en el apartado anterior de la funcionalidad NBookStore a través de spring framework. Esto se hace gracias a la inyección de dependencias por medio del contenedor de beans spring, desde los manage beans se invocan adaptadores de negocio de esta forma se mapean los parámetros de entrada del método de negocio y este ejecutara desde el negocio la funcionalidad deseada, todo esto es posible gracias a la arquitectura del framework spring.

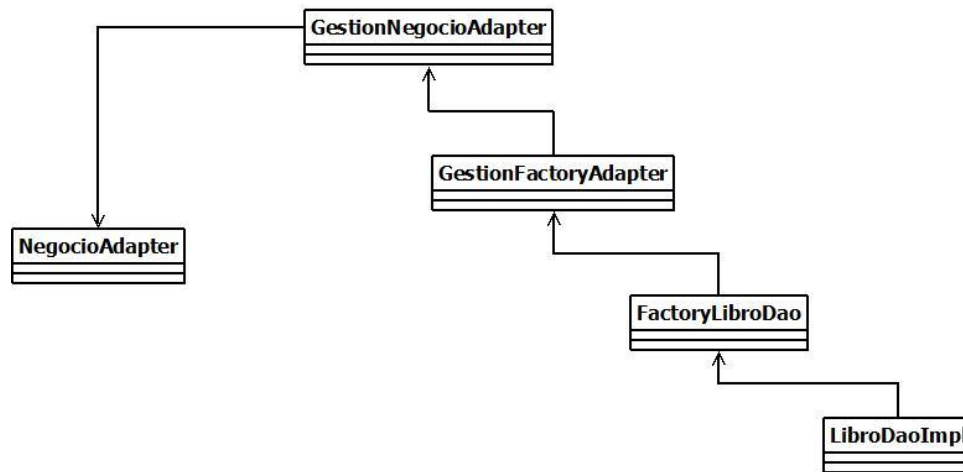


Figura 9. La figura muestra la inyección de dependencias que se realizan para la comunicación del negocio con cualquier tipo de presentación.

Se adjunta el diagrama de entidad relación de la base de datos bookstore, de esta forma se pueden observar las relaciones entre las diferentes tablas, antes de mostrar algunas de las consultas utilizadas para el funcionamiento del negocio.

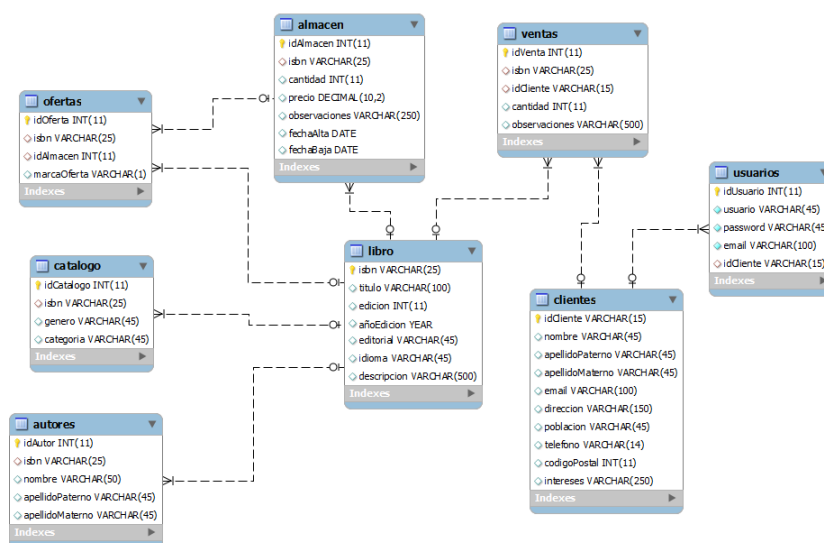


Figura 10. Diagrama entidad relación de la base de datos bookstore.

Se realiza una select de la tabla libro donde se busca por los parámetros de isbn y la relación que se tiene con las tablas de almacén y catalogo.

```
select libro.isbn, titulo, edicion, añoEdicion, editorial, idioma, descripcion, categoria, genero, precio from libro, almacen, catalogo where ((libro.isbn = ?) and ((libro.isbn = almacen.isbn) and (libro.isbn = catalogo.isbn)))
```

Se realiza una select de la tabla autores donde se busca por los parámetros de nombre, apellidoPaterno o apellidoMaterno.

```
select idAutor, isbn, nombre, apellidoPaterno, apellidoMaterno from autores where (((nombre like ?) or (apellidoPaterno like ?) or (apellidoMaterno like ?)))
```

Se realiza un insert en la tabla de clientes donde se pasan como parámetros todos los campos de la tabla.

```
insert into clientes (idCliente, nombre, apellidoPaterno, apellidoMaterno, direccion, poblacion, telefono, email, codigoPostal, intereses) values(?, ?, ?, ?, ?, ?, ?, ?, ?, ?).
```

## Fichero Web XML

Se adjunta la configuración empleada en el archivo Web.XML gracias a la sencilla configuración del framework PrimeFaces 3.1, no es necesario añadir referencias a dependencias externas en el archivo web.xml, se tiene la configuración del framework spring para tratar las peticiones de los clientes, se configura además como servlets la tecnología java servlet faces que es la encargada de resolver el flujo de las navegaciones. Se configuran además las páginas que se pueden tratar con esta tecnología, se tiene configurado el tema de PrimeFaces para la presentación, la pagina de error cuando se produzca una excepción y las páginas de bienvenida de la aplicación.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID"
version="3.0">
  <display-name>WBookStore</display-name>
  <!-- Soporte de Spring para el manejo de inyeccion de dependencias -->
  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/applicationContext.xml</param-value>
  </context-param>
  <listener>
    <listener-class>
      org.springframework.web.context.ContextLoaderListener
    </listener-class>
  </listener>
  <listener>
    <listener-class>
      org.springframework.web.context.request.RequestContextListener
    </listener-class>
  </listener>
  <!-- Cambiar a "Production" cuando este listo para desplegar -->
  <context-param>
    <param-name>javax.faces.PROJECT_STAGE</param-name>
    <param-value>Development</param-value>
  </context-param>
  <context-param>
    <param-name>primefaces.THEME</param-name>
    <param-value>cupertino</param-value>
  </context-param>
  <!-- JavaServerFace asignar como inicial -->
  <servlet>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
```

```

</servlet>
<!-- Mapeo de archivos con JSF -->
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>/faces/*</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>*.jsf</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>*.faces</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>*.xhtml</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>*.jspx</url-pattern>
</servlet-mapping>
<!-- Página de error -->
<error-page>
  <error-code>
    404
  </error-code>
  <location>
    /Html/Error.jsf
  </location>
</error-page>
<error-page>
  <error-code>
    500
  </error-code>
  <location>
    /Html/Error.jsf
  </location>
</error-page>
<!-- Páginas de Bienvenida -->
<welcome-file-list>
  <welcome-file>index.html</welcome-file>
  <welcome-file>index.htm</welcome-file>
  <welcome-file>index.jsp</welcome-file>
  <welcome-file>default.html</welcome-file>
  <welcome-file>default.htm</welcome-file>
  <welcome-file>default.jsp</welcome-file>
</welcome-file-list>
</web-app>

```