

PROYECTO: Minitienda con MVC, EL/JSTL y bases de datos

Objetivos del proyecto

1. *Diseñar la minitienda con el patrón MVC*
2. *Aplicar las tecnologías EL y JSTL para eliminar scriptlets de las Vistas*
3. *Añadir una base de datos*

El objetivo del proyecto es completar el desarrollo de la minitienda, incluyendo tanto la fase de diseño como la fase de implementación con todas las tecnologías web que hemos visto en el segundo bloque de la asignatura.

1. Requisitos de la aplicación web

1.1 Funcionalidades

A las funcionalidades F1-F5 de la práctica 5, hay que añadir:

- F6. Para confirmar la compra, preguntar al usuario por sus credenciales (correo electrónico y contraseña) para acceder a su cuenta. Si no la tiene, registrar al nuevo usuario pidiendo que introduzca sus credenciales (correo electrónico y contraseña) y su tarjeta de crédito (tipo y número de tarjeta. Crear una tabla en la BD para los nuevos usuarios.
- F7. Almacenar el pedido en una base de datos. Crear una tabla en la BD para almacenar los pedidos. Recuerda relacionar cada pedido con el usuario que realizó la compra Para el pedido basta con guardar el importe final.
- F8. Después de guardar los datos en la BD, mostrar al usuario un mensaje de confirmación de la compra que contenga los datos del pedido.

1.2 Diseño de la aplicación

- Aplicar el patrón MVC.

1.3 Implementación de la solución

La implementación de esta práctica debe cumplir los siguientes puntos:

- Vistas: Utilizar HTML, EL y JSTL.
- Controlador: Identificar las posibles acciones del usuario y los planes necesarios para satisfacer cada acción.
- Modelo: Implementar las clases identificadas en el diseño de la aplicación. La conexión a la BD debe realizarse con JDBC.

2. Presentación del proyecto

- **Ficheros UML describiendo el diseño de la aplicación.** Son los siguientes:

- Diagrama de clases del MODELO. Sólo del MODELO.
- Diagramas de secuencia para estas dos acciones: 1. Seleccionar producto del catálogo, 2. Guardar el pedido y el usuario en la BD.
- **Generar un fichero *proyecto_nombre_apellidos.war***, que contenga los siguientes ficheros en el árbol de directorios requerido para una aplicación web:
 - Fichero(s) .html
 - Imágenes utilizadas
 - Fichero(s) .jsp
 - Fichero(s).java
 - Fichero(s).class (Fichero(s).java compilados)
- El fichero .war debe estar preparado para su despliegue automático en TOMCAT. Esto significa que la aplicación debe estar disponible en [http://localhost:8080/ proyecto_nombre_apellidos/](http://localhost:8080/proyecto_nombre_apellidos/), una vez copiado el fichero .war en <TOMCAT>/webapps, y activado el contenedor web.
- Crear un ZIP con el fichero UML y el fichero *proyecto_nombre_apellidos.war* y subirlo en el Campus Virtual.
- Fecha tope: **Martes, 20 de Mayo – 23:59 horas.**

3. Evaluación del proyecto

- Funcionalidades del proyecto: 1 punto

F1. Seleccionar un CD de una lista, indicar la cantidad que se desea comprar, y almacenar la selección en un carrito de la compra. **Puntos: 0.1**

F2. Mostrar la lista de CDs del carrito de la compra y el importe total de la Selección. **Puntos: 0.15.**

F3. Presentar el importe final de la selección para realizar el pago de la misma. **Puntos: 0.1**

F4. Eliminar un CD del carrito y seguir con la compra o solicitar el importe final de la selección. **Puntos: 0.15**

F5. Inicializar el carrito de la compra cuando se realiza el pago y se vuelve a la página principal, y preparar la aplicación para realizar una nueva compra. **Puntos: 0.1**

F6. Para confirmar la compra, preguntar al usuario sus datos de contacto (nombre y correo electrónico) y su tarjeta de crédito (tipo y número de tarjeta). **Puntos: 0.1**

F7. Almacenar el pedido en una base de datos. Crear al menos dos tablas en la BD: una para guardar los nuevos usuarios y otra para guardar sus pedidos. Para el pedido basta con guardar el importe final. **Puntos: 0.15.**

F8. Después de guardar los datos en la BD, mostrar al usuario un mensaje de confirmación de la compra que contenga los datos del pedido. **Puntos: 0.15.**

- **Diseño del proyecto: 1 punto**
 - Diagrama de clases del MODELO. **Puntos: 0.5.**
 - Diagramas de secuencia para estas dos acciones:
 - Seleccionar producto del catálogo, **Puntos: 0.25**
 - Guardar el pedido y el usuario en la BD. **Puntos: 0.25.**
 - **Puntos extra:** Aplicación del patrón “Service to Worker”, lo que implica añadir “Asistentes/Helpers” que ayuden al “Controlador. **Puntos: 0.2**