

Escuela Multimedial Da Vinci

The wizard
Store {

<Por="Melina Sol Alonso"/>

}



Contenidos

03	Introducción
04	Especificaciones
05	Estructura
06	Flujo de usuario
07-08	Sistema de login y registro
09	Carrito de compras
10	Funcionalidades Carrito
11	Base de datos
12	Interfaz de usuario
13	Consideraciones de seguridad

Introducción {

Nombre del alumno: Melina Sol Alonso

Comision: DWM3AV

Nombre y Apellido Profesor: Cano Bruno

Nombre del proyecto: The Wizard Store



}

Especificaciones {

Funcionalidades principales del portal web.

- Portal de E-commerce: Permite a los usuarios registrarse, iniciar sesión, ver productos, añadir productos al carrito, y completar la compra.
- Gestión de Usuarios: Registro y autenticación de usuarios con roles diferenciados.
- Gestión de Productos: Visualización, búsqueda y filtrado de productos disponibles en la tienda.
- Carrito de Compras: Gestión completa del carrito con opciones para añadir, eliminar y actualizar productos.
- Proceso de Checkout: Finalización de compra con actualización del estado del carrito.

Tecnologías utilizadas

- Lenguajes de Programación: PHP
- Base de Datos: MySQL
- Estilos: CSS
- Interfaz de Usuario: HTML, CSS
- Framework: Bootstrap

}

Estructura del proyecto {

DWM3AV_ALONSO_FINAL/

— clases/

— Carrito.php

— connexion.php

— Producto.php

— Usuario.php

— css/

— style.css

— img/

— (Imágenes del proyecto)

— vistas/

— 404.php

— contacto.php

— datos_alumno.php

— detalle_producto.php

— home.php

— login.php

— logout.php

— perfil.php

— productos.php

— register.php

— vista-carrito.php

— index.php

Breve explicación del propósito de los archivos y directorios principales
clases/: Contiene las clases PHP que manejan la lógica del negocio.

Carrito.php: Lógica relacionada con la gestión del carrito de compras.

connexion.php: Gestión de la conexión a la base de datos.

Producto.php: Lógica para gestionar productos en el sistema.

Usuario.php: Lógica para la autenticación y registro de usuarios.

css/: Contiene los archivos de estilo CSS.

style.css: Archivo principal de estilos para el portal web.

img/: Almacena las imágenes usadas en el sitio.

vistas/: Contiene las vistas PHP que componen la interfaz del usuario.

404.php: Página de error para rutas no encontradas.

contacto.php: Formulario de contacto.

datos_alumno.php: Muestra información del alumno que realizó el proyecto.

detalle_producto.php: Vista detallada del producto que selecciona el usuario.

home.php: Página de inicio del portal.

login.php: Formulario de inicio de sesión.

logout.php: Manejador de cierre de sesión.

perfil.php: Página de perfil del usuario.

productos.php: Listado de productos.

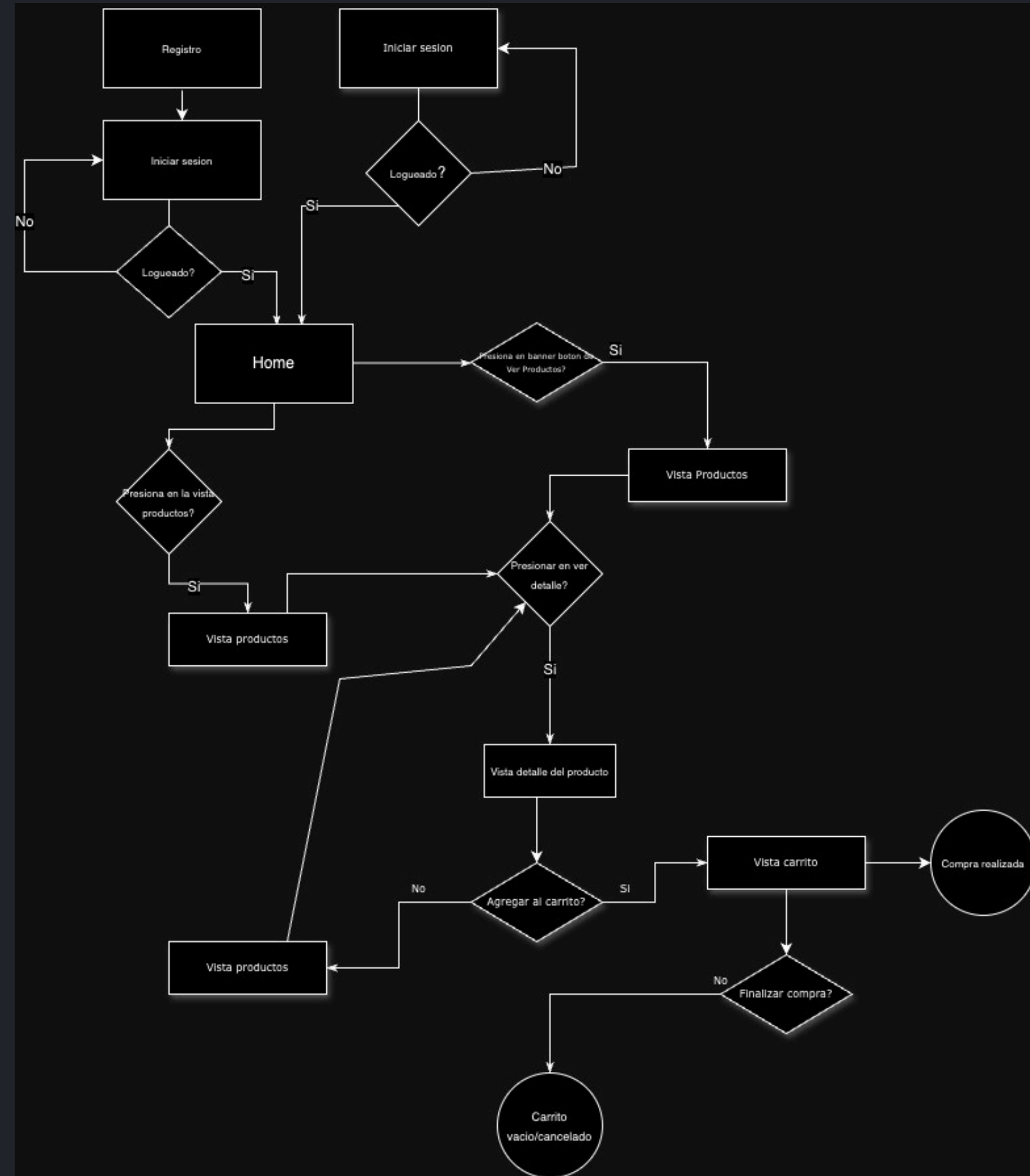
register.php: Formulario de registro de usuario.

vista-carrito.php: Visualización del carrito de compras.

index.php: Archivo principal que actúa como punto de entrada al portal.

}

Flujo de usuario {



}

Sistema de Login {

Autenticación

Consulta de Usuario:

- El método login de la clase Usuario busca en la base de datos un registro que coincida con el correo electrónico proporcionado por el usuario.
- Se prepara y ejecuta una consulta SQL para seleccionar todos los campos de la tabla usuarios donde el email coincide con el proporcionado.

Verificación de Contraseña:

- Si la consulta devuelve un resultado, el método password_verify se utiliza para comparar la contraseña proporcionada por el usuario con la contraseña hash almacenada en la base de datos.
- Si las contraseñas coinciden, se inicia una sesión (session_start()) y se almacenan en la variable de sesión detalles importantes del usuario, como su id, email y nombre.
- El método devuelve true si el inicio de sesión es exitoso; de lo contrario, devuelve false.

Autorización

Verificación de Sesión:

En las páginas que requieren acceso autorizado, se puede verificar si las variables de sesión \$_SESSION['user_id'], \$_SESSION['email'], o \$_SESSION['nombre'] están establecidas. Si las variables de sesión no están establecidas, significa que el usuario no está autenticado y puede ser redirigido al formulario de inicio de sesión.

}

Sistema de Registro {

Autenticación

Hash de Contraseña:

- Antes de almacenar la contraseña en la base de datos, se utiliza `password_hash` para convertirla en un hash seguro. Esto es crucial para proteger las contraseñas contra accesos no autorizados.

Inserción en Base de Datos:

- Se prepara una consulta `INSERT` para añadir un nuevo registro en la tabla `usuarios` con el nombre, email, y la contraseña hasheada.
- Los parámetros de la consulta se vinculan usando `bindParam` para prevenir inyecciones SQL y garantizar la seguridad.
- Si la ejecución de la consulta es exitosa, el método devuelve `true`, indicando que el registro se completó correctamente; de lo contrario, devuelve `false`.

}

Gestión de Productos en el Carrito de Compras {



1. AGREGAR UN PRODUCTO AL CARRITO

Cuando un usuario agrega un producto al carrito:
Verificación de Sesión:
Se verifica que el usuario esté autenticado comprobando la existencia de `$_SESSION['user_id']`.
Carrito Activo:
Se ejecuta una consulta SQL para verificar si existe un carrito activo (`estado_compra = 'Activo'`) para el usuario actual.
Si no existe un carrito activo, se crea uno nuevo y se guarda en la base de datos con el estado Activo.
Actualizar o Insertar Producto:
Si el producto ya está en el carrito, se incrementa su cantidad.
Si el producto no está presente, se inserta en la tabla de detalles del carrito (`carrito_detalle`) con una cantidad inicial de 1.

2. ELIMINAR O REDUCIR UN PRODUCTO DEL CARRITO

Cuando un usuario decide eliminar un producto:

- Verificación de Carrito Activo:
- Comprueba si hay un carrito activo para el usuario.
- Verificar Existencia del Producto:
 - Si el producto está en el carrito, su cantidad es verificada.
 - Si la cantidad es mayor que uno, se reduce la cantidad en uno.
 - Si la cantidad es uno, se elimina el producto del carrito.

3. VACIAR EL CARRITO

Cambiar Estado del Carrito:
Al vaciar el carrito, su estado se actualiza a Anulado y se eliminan todos los detalles asociados al carrito.

4. FINALIZAR COMPRA

Completar Compra:
El estado del carrito cambia a Finalizado, y todos los productos del carrito son eliminados. Se notifica al usuario que se pondrán en contacto para confirmar la compra.

5. CANCELAR COMPRA

Anular Carrito:
El carrito y sus productos son eliminados de manera similar a vaciarlo, pero se notifica al usuario que el carrito ha sido cancelado correctamente.

}

Funcionalidades disponibles en el carrito {

Proceso de Finalización de Compra (Checkout)

1. Iniciar el Proceso de Checkout:

- Los usuarios pueden proceder al checkout para completar su compra.

Proceso:

- Se verifica que todos los productos deseados están en el carrito.
- El total de la compra se calcula sumando los subtotales de cada producto y se muestra en el final de la vista.

2. Finalizar Compra:

- Los usuarios confirman la compra, completando el proceso de checkout.

Proceso:

- El estado del carrito se actualiza a Finalizado.
- Todos los productos en el carrito se eliminan de la tabla carrito_detalle.
- Se muestra un mensaje de confirmación al usuario indicando que se pondrán en contacto para confirmar la compra.

3. Opciones Adicionales en el Checkout:


• Vaciar Carrito:

- Cambia el estado del carrito a Anulado.
- Elimina todos los productos del carrito.

• Cancelar Compra:

- Similar al vaciado del carrito, pero con notificación específica de cancelación.

Mi Carrito




Buzo Gryffindor
Precio: 55,000.00 \$

Cantidad: 2

Subtotal: 110,000.00 \$

Eliminar producto



Toga Slytherin
Precio: 70,000.00 \$

Cantidad: 1

Subtotal: 70,000.00 \$

Eliminar producto

Total: 180,000.00 \$

Finalizar Compra

Vaciar Carrito

Cancelar Compra



Base de Datos {

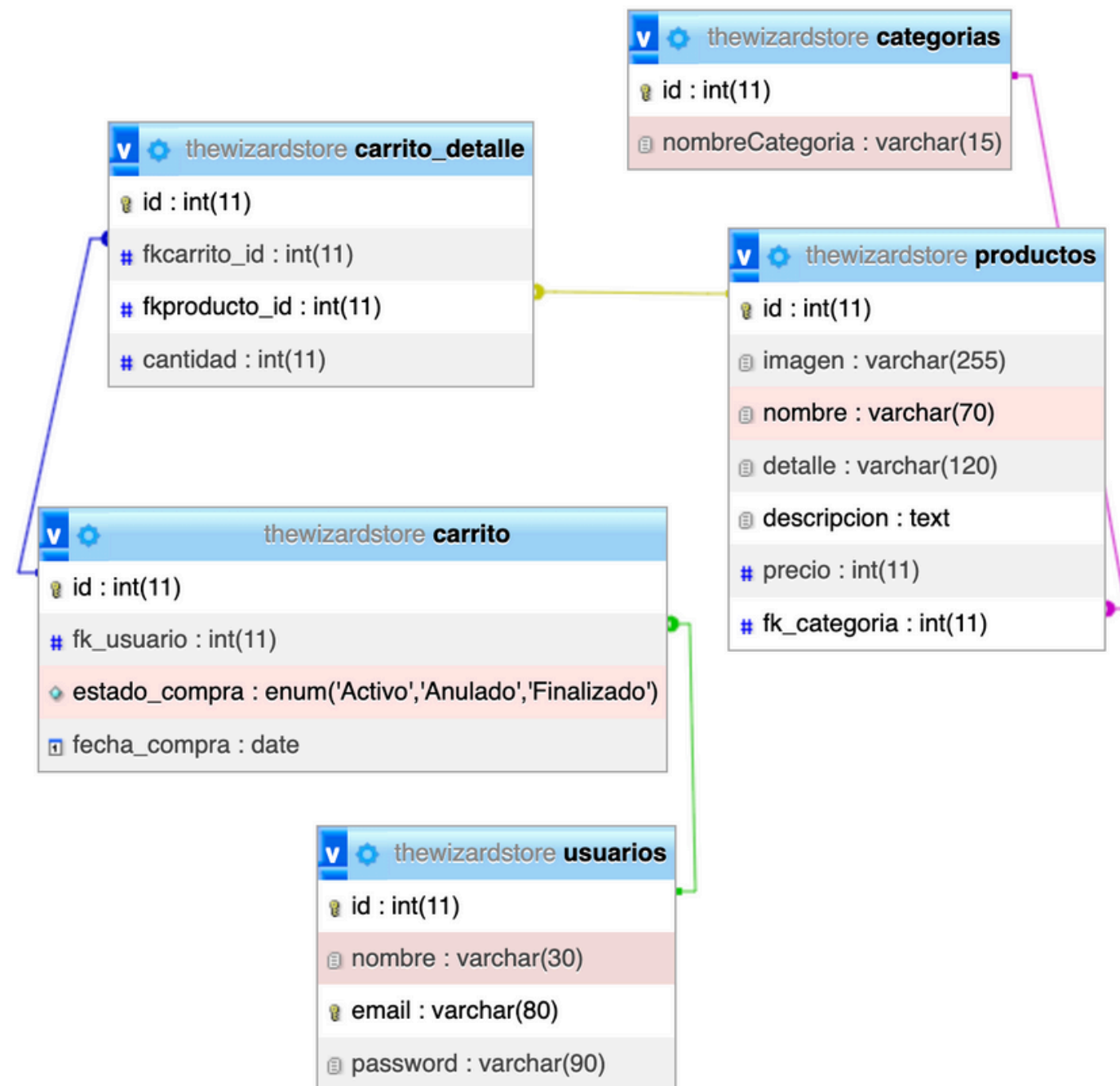


Diagrama entidad-relación
(DER) de la base de datos.

}

Interfaz de Usuario {

El diseño de la interfaz de usuario para "The Wizard Store" se centra en crear una experiencia mágica, fluida y agradable para los usuarios, asegurando que puedan navegar, encontrar y comprar productos con facilidad y placer.

1. Diseño Responsivo

- **Compatibilidad Multidispositivo:** La interfaz está diseñada para ser completamente responsiva, garantizando una experiencia de usuario óptima en dispositivos móviles, tabletas y ordenadores de escritorio.
- **Framework CSS:** Se utiliza un framework como Bootstrap para asegurar que el diseño se ajuste automáticamente a diferentes tamaños de pantalla y resoluciones.

2. Paleta de Colores

- **Estética Mágica:** Los elementos gráficos y los fondos están diseñados con una estética mágica que recuerda al mundo de Harry Potter, incluyendo texturas de pergamino, bordes decorativos y tipografía temática.

3. Interactividad y Feedback

- **Interacciones Visuales:** Efectos visuales sutiles para botones y enlaces (hover, focus), proporcionando feedback al usuario.
- **Mensajes de Confirmación:** Notificaciones para confirmar acciones del usuario, como añadir productos al carrito, completar la compra y registrarse.

4. Estructura de la Página

• Página de Tienda:

- **Filtros y Categorías:** Barra superior con opciones para filtrar productos por categoría y un buscador por nombre.
- **Listado de Productos:** Productos con imágenes, nombres, precios y botones para ver detalles.

• Página de Detalle de Producto:

- **Imagen del Producto:** Imagen grande y de alta calidad del producto.
- **Descripción del Producto:** Detalles del producto, incluyendo nombre, descripción completa, características y precio.
- **Botones de Acción:** Botones para añadir al carrito, comprar ahora y volver atrás.

• Página de Carrito:

- **Listado de Productos en el Carrito:** Tabla con los productos seleccionados, sus cantidades, precios unitarios y el total.
- **Opciones de Modificación:** Botones para actualizar cantidades, eliminar productos.
- **Resumen y Checkout:** Resumen del total a pagar y botón para proceder al pago.

}

Consideraciones de Seguridad {

1. ENCRIPCIÓN DE CONTRASEÑAS

Hashing de Contraseñas: Las contraseñas de los usuarios no se almacenan en texto plano. En su lugar, se utilizan algoritmos de hashing seguros (como bcrypt) para encriptar las contraseñas antes de almacenarlas en la base de datos.

2. AUTENTICACIÓN

Utilización de consultas preparadas y sentencias parametrizadas con PDO (prepare y bindParam) para evitar inyecciones SQL. Este enfoque asegura que las entradas del usuario se traten como parámetros en lugar de ser directamente interpoladas en la consulta SQL, lo cual previene que un atacante pueda manipular las consultas SQL para ejecutar código malicioso.

3. VERIFICACIÓN CONTRASEÑA

La verificación de contraseñas se realiza mediante password_verify, asegurando que las contraseñas en texto plano se comparen de manera segura con los hashes almacenados

}

```
<!--Escuela multimedial Da Vinci-->
```

Gracias {

```
<Por="Melina Alonso"/>
```

}