INFORME DSW

Tienda llaveros

INDICE

6. **Sistema de Autenticación y Sesiones**
   6.1. Inicio de sesión
   6.2. Registro de usuarios
   6.3. Gestión de roles (admin / usuario)
   6.4. Cierre de sesión
7. **Panel de Administración**
   7.1. Control de acceso
   7.2. Gestión de productos (CRUD)
   7.3. Gestión de pedidos
8. **Carrito de Compras y Pedidos**
   8.1. Gestión del carrito
   8.2. Persistencia en base de datos
   8.3. Confirmación de compra
   8.4. Historial de pedidos
9. **Personalización del Usuario (Cookies)**
   9.1. Selección de idioma
   9.2. Selección de estilo visual
   9.3. Aplicación global de preferencias
10. **Seguridad y Buenas Prácticas**
    10.1. Protección frente a SQL Injection
    10.2. Gestión segura de contraseñas
    10.3. Control de sesiones
    10.4. Optimización y reutilización de código

Introducción

Este proyecto consiste en el desarrollo de una **tienda online funcional** utilizando **PHP orientado a objetos (POO)**, **MySQL** como sistema gestor de base de datos, **sesiones**, **cookies** y **HTML/CSS** para la presentación.

La aplicación permite:

- Autenticación y registro de usuarios.
- Diferenciación de roles (usuario / administrador).
- Gestión de productos (CRUD) exclusiva para el administrador.
- Carrito de compras persistente en base de datos.
- Sistema de pedidos.
- Preferencias de idioma y tema visual mediante cookies.

- Uso de clases y herencia conforme a los principios de POO.

Arquitectura del Proyecto

La aplicación sigue una **estructura modular**, separando responsabilidades en carpetas específicas:

Proyecto UT5 DSW/

```
|
├── admin/          → Panel de administración
├── auth/           → Login, logout y registro
├── carrito/        → Gestión del carrito y pedidos
├── clases/         → Clases POO del dominio
├── config/         → Configuración global
├── css/            → Estilos y temas
├── idiomas/        → Internacionalización
├── uploads/        → Imágenes de productos
├── DB-SCRIPT/      → Script SQL de la base de datos
└── index.php       → Entrada principal
```

Esta organización **mejora la mantenibilidad, escalabilidad y legibilidad** del código, cumpliendo buenas prácticas profesionales.

Programación Orientada a Objetos (POO)

Clase abstracta Producto :

abstract class Producto {

protected string $nombre;

protected float $precio;

}

Es una **clase abstracta**, no instanciable.

Define atributos comunes a todos los productos.

Obliga a usar herencia.

Aplica el principio de **abstracción**.

Clase Llavero (herencia) :

class Llavero extends Producto {

private string $tipo;

}

Hereda de Producto.

Añade el atributo tipo (llaves, bolso, mochila).

Cumple el requisito de **especialización** del producto.

Clase abstracta Paquete:

- Representa un conjunto de productos.
- Contiene:
  - Producto (objeto Llavero)
  - Cantidad
  - Importe total
- Define el método abstracto calcularImporte().

Esto garantiza que **cada tipo de paquete implemente su propia lógica**, aplicando **polimorfismo**.

Paquetes concretos :

Clase	Comportamiento
PaqueteLlaves	Precio × cantidad
PaqueteBolso	Precio individual
PaqueteMochila	Precio × cantidad con descuento

Cada clase:

- Hereda de Paquete
- Implementa su propia lógica de cálculo
- Respeta el principio **Open/Closed**

Base de Datos MySQL

DB-SCRIPT/creacion-DB.sql

Tablas principales

- **usuarios**
- Gestión de autenticación
- Rol (admin / user)
- Contraseñas cifradas (password_hash())

Autenticación y Sesiones

auth/

login.php

- Verifica credenciales contra la BD.
- Inicia sesión con $_SESSION.
- Detecta el rol del usuario.

registro.php

- Crea nuevos usuarios.
- Evita duplicados.
- Asigna rol user por defecto.
- El rol admin está restringido.

logout.php

- Destruye sesión completamente.
- Redirige al login.

Seguridad:

- Contraseñas cifradas.
- Sesiones bien gestionadas.
- Control de accesos por rol.

Panel de Administración

admin/

$_SESSION['usuario_rol'] === 'admin'

Funcionalidades:

- Crear productos
- Editar productos
- Eliminar productos
- Ver pedidos

Control de acceso correcto
Separación clara entre usuario y administrador

Carrito y Pedidos

carrito/

- El carrito se guarda en **base de datos**, no cookies.
- Cada usuario tiene su propio carrito.
- Se permite:
  - Añadir productos
  - Modificar cantidades
  - Vaciar carrito
  - Confirmar pedido

confirmacion_pago.php
mis_pedidos.php

Esto simula el flujo real de una tienda online.

Preferencias de Usuario

preferencias.php

- Idioma: Español / Inglés
- Tema: Claro / Oscuro
- Guardadas mediante **cookies**
- Aplicadas globalmente con:
  - config/i18n.php
  - CSS dinámico

Uso correcto de cookies
Persistencia entre sesiones

Seguridad y Buenas Prácticas

require_once centralizado
Autoload de clases
Consultas preparadas (SQL Injection evitado)
Separación lógica (MVC simplificado)

Código reutilizable y optimizado

(CUALQUIER DUDA CON EL CODIGO SE PUEDE OBSERVAR EN EL PROYECTO, RECOMENDACION EJECUTAR TODA LA BASE DE DATOS DEL SCRIPT MAS EL AÑADIDO DEL USUARIO ADMIN, LOS PRODUCTOS Y LOS INDICES DE LAS ULTIMAS TABLAS INDICADAS EN EL PROPIO SCRIPT)