

TIENDA DE SEÑALES DE TRÁFICO

Proyecto Final DWES

2º DAW (2024/2025)



Alonso Hernández Robles

Índice

- 0. Introducción
- 1. Archivos Raíz
 - 1.1. `index.php`
 - 1.2. `config.php`
 - 1.3. `autoload.php`
 - 1.4. `.htaccess`
 - 1.5. `.env`
 - 1.6. `.gitignore`
 - 1.7. `README.md`
 - 1.8. `README.pdf`
- 2. Activos (`assets/`)
 - 2.1. Estilo (`css/style.css`)
 - 2.2. Fuente (`fonts/Rubik-Regular.ttf`)
 - 2.3. Imágenes (`images/`)
 - 2.3.1. Recursos
 - 2.3.2. Subidas (`uploads/`)
 - 2.3.2.1. Productos (`productos/`)
 - 2.3.2.2. Usuarios (`usuarios/`)
- 3. Base de Datos (`database/database.sql`)
- 4. Clase `BaseDatos` (`lib/BaseDatos.php`)
- 5. Clase `Utils` (`helpers/Utils.php`)
- 6. Modelos (`models/`)
 - 6.1. `Categoria.php`
 - 6.2. `Producto.php`
 - 6.3. `Usuario.php`
 - 6.4. `Pedido.php`
- 7. Controladores (`controllers/`)
 - 7.0. `ErrorController.php`
 - 7.1. `CategoriaController.php`
 - 7.2. `ProductoController.php`
 - 7.3. `UsuarioController.php`
 - 7.4. `CarritoController.php`
 - 7.5. `PedidoController.php`
- 8. Vistas (`views/`)
 - 8.1. Diseño (`layout/`)
 - 8.1.1. `header.php`
 - 8.1.2. `footer.php`
 - 8.2. `categoria/`
 - 8.2.1. `admin.php`
 - 8.2.2. `crear.php`
 - 8.2.3. `editar.php`
 - 8.3. `producto/`
 - 8.3.1. `admin.php`

- 8.3.2. crear.php
- 8.3.3. editar.php
- 8.3.4. ver.php
- 8.3.5. recomendados.php
- 8.4. usuario/
 - 8.4.1. registrarse.php
 - 8.4.2. login.php
 - 8.4.3. gestion.php
 - 8.4.4. admin.php
 - 8.4.5. crear.php
 - 8.4.6. editar.php
- 8.5. carrito/
 - 8.5.1. gestion.php
- 8.6. pedido/
 - 8.6.1. admin.php
 - 8.6.2. crear.php
 - 8.6.3. listo.php
 - 8.6.4. ver.php

0. Introducción

Este proyecto consiste en una **Tienda de Señales de Tráfico**, donde los usuarios pueden registrarse, iniciar sesión y ver productos. También existen usuarios con un rol **Administrador**, que tienen ciertos privilegios.

A continuación se detallará la estructura de archivos y carpetas del proyecto.

```
C:.\
├── .env
├── .gitignore
├── .htaccess
├── autoload.php
├── config.php
├── index.php
├── README.md
├── README.pdf
├── assets
│   ├── css
│   │   └── style.css
│   ├── fonts
│   │   └── Rubik-Regular.ttf
│   └── images
│       ├── carrito.svg
│       ├── categoria.svg
│       ├── edit.svg
│       ├── left.svg
│       ├── login.svg
│       ├── logo.svg
│       ├── logout.svg
│       ├── pedido.svg
│       ├── producto.svg
│       ├── registrarse.svg
│       ├── right.svg
│       ├── usuario.svg
│       └── usuarios.svg
│       └── uploads
│           ├── productos
│           │   └── {...}
│           └── usuarios
│               └── {...}
└── controllers
    ├── CarritoController.php
    ├── CategoriaController.php
    └── ErrorController.php
```

```
PedidoController.php
ProductoController.php
UsuarioController.php

—database
  database.sql

—helpers
  Utils.php

—lib
  BaseDatos.php

—models
  Categoria.php
  Pedido.php
  Producto.php
  Usuario.php

—views
  —carrito
    gestion.php

  —categoria
    admin.php
    crear.php
    editar.php

  —layout
    footer.php
    header.php

  —pedido
    admin.php
    crear.php
    listo.php
    ver.php

  —producto
    admin.php
    crear.php
    editar.php
    recomendados.php
    ver.php

  —usuario
    admin.php
    crear.php
    editar.php
    gestion.php
    login.php
    registrarse.php
```

1. Archivos Raíz

1.1. `index.php`

Archivo principal de la aplicación. Se encarga de iniciar la sesión, inicializar algunas variables y cargar el controlador y la acción correspondiente.

Esto se consigue construyendo la página a partir de las vistas de diseño (`views/layout/header.php` y `views/layout/footer.php`) intercalando la vista correspondiente al controlador y acción.

Si no se encuentra el controlador o la acción, se redirige al controlador de errores.

1.2. `config.php`

Archivo de configuración de la aplicación. Contiene las constantes de conexión a la base de datos, el controlador y acción por defecto, la ruta base de la aplicación y los objetos por página cuando se haga uso de la paginación.

1.3. `autoload.php`

Archivo que se encarga de cargar automáticamente las clases que se vayan a utilizar en la aplicación.

1.4. `.htaccess`

Archivo de configuración de Apache. Se encarga de redirigir todas las peticiones a `index.php`.

1.5. `.env`

Archivo de configuración de variables de entorno. Contiene las variables de conexión a la base de datos.

En este caso, es necesario usar **Composer** para cargar las variables de entorno. Sin embargo, no se ha conseguido instalar **Composer** en el proyecto.

De todas formas, cabe aclarar que las variables de entorno se encuentran en el archivo `.env` y deberían reemplazarse en el archivo `config.php` para evitar exponer las credenciales de la base de datos.

1.6. `.gitignore`

Archivo que contiene los archivos y carpetas que se deben ignorar al subir el proyecto a un repositorio de **Git**. En este caso, solamente se ignora `.env`.

1.7. `README.md`

Este archivo. Contiene la documentación del proyecto.

1.8. `README.pdf`

Archivo PDF generado a partir de este archivo.

2. Activos (`assets/`)

2.1. Estilo (`css/style.css`)

Archivo CSS que contiene los estilos de la aplicación.

2.2. Fuente (`fonts/Rubik-Regular.ttf`)

Fuente **Rubik** utilizada en la aplicación.

2.3. Imágenes (`images/`)

2.3.1. Recursos

Esta carpeta contiene las imágenes estáticas inmutables que se utilizan en la aplicación.

2.3.2. Subidas (`uploads/`)

2.3.2.1. Productos (`productos/`)

Esta carpeta contiene las imágenes de los productos subidas por los usuarios administradores en la creación y edición de productos.

2.3.2.2. Usuarios (`usuarios/`)

Esta carpeta contiene las imágenes de perfil de los usuarios subidas por los usuarios tanto convencionales como administradores en la creación y edición de usuarios, además de en la gestión de usuarios convencionales.

3. Base de Datos (`database/database.sql`)

Archivo SQL que contiene la estructura de la base de datos. Se debe importar en **phpMyAdmin** o en la terminal de **MySQL**.

4. Clase `BaseDatos` (`lib/BaseDatos.php`)

Clase que contiene los métodos necesarios para la conexión a la base de datos.

5. Clase `Utils` (`helpers/Utils.php`)

Clase que contiene métodos útiles para la aplicación, como validaciones de rol, identidad y eliminación de campos en la variable superglobal `$_SESSION`, cookies, etc.

6. Modelos (`models/`)

6.1. `Categoria.php`

Modelo de la categoría. Contiene los métodos necesarios para la gestión de las categorías.

6.2. `Producto.php`

Modelo del producto. Contiene los métodos necesarios para la gestión de los productos.

6.3. `Usuario.php`

Modelo del usuario. Contiene los métodos necesarios para la gestión de los usuarios.

6.4. `Pedido.php`

Modelo del pedido. Contiene los métodos necesarios para la gestión de los pedidos.

7. Controladores (`controllers/`)

7.0. `ErrorController.php`

Controlador de errores. Es invocado cuando no se encuentra el controlador o la acción correspondiente.

7.0.1. `ErrorController::index()`

Método que muestra un mensaje de error.

7.1. `CategoriaController.php`

Controlador de categorías. Contiene los métodos necesarios para la gestión de las categorías.

7.1.1. `CategoriaController::admin()`

Método que muestra toda la administración de las categorías con paginación.

7.1.2. `CategoriaController::crear()`

Método que muestra el formulario de creación de categorías.

7.1.3. `CategoriaController::guardar()`

Método que guarda la categoría en la base de datos y redirige a la administración de categorías.

7.1.4. `CategoriaController::editar()`

Método que actualiza la categoría en la base de datos y redirige a la administración de categorías.

7.1.5. `CategoriaController::gestion()`

Método que muestra el formulario de edición de categorías.

7.1.6. `CategoriaController::eliminar()`

Método que elimina la categoría de la base de datos y redirige a la administración de categorías.

7.2. `ProductoController.php`

Controlador de productos. Contiene los métodos necesarios para la gestión de los productos.

7.2.1. `ProductoController::recomendados()`

Método que muestra la vista de los productos recomendados con paginación.

7.2.2. `ProductoController::admin()`

Método que muestra toda la administración de los productos con paginación.

7.2.3. `ProductoController::crear()`

Método que muestra el formulario de creación de productos.

7.2.4. `ProductoController::guardar()`

Método que guarda el producto en la base de datos y redirige a la administración de productos.

7.2.5. `ProductoController::editar()`

Método que actualiza el producto en la base de datos y redirige a la administración de productos.

7.2.6. `ProductoController::gestion()`

Método que muestra el formulario de edición de productos.

7.2.7. `ProductoController::eliminar()`

Método que elimina el producto de la base de datos y redirige a la administración de productos.

7.2.8. `ProductoController::ver()`

Método que muestra la vista de un producto.

7.3. `UsuarioController.php`

Controlador de usuarios. Contiene los métodos necesarios para la gestión de los usuarios.

7.3.1. `UsuarioController::registrarse()`

Método que muestra el formulario de registro de usuarios.

7.3.2. `UserController::guardar()`

Método que guarda el usuario en la base de datos y redirige a la página de inicio.

7.3.3. `UserController::login()`

Método que muestra el formulario de inicio de sesión.

7.3.4. `UserController::entrar()`

Método que inicia la sesión del usuario y redirige a la página de inicio.

7.3.5. `UserController::salir()`

Método que cierra la sesión del usuario y redirige a la página de inicio.

7.3.6. `UserController::gestion()`

Método que muestra la página de gestión del usuario.

7.3.7. `UserController::admin()`

Método que muestra toda la administración de los usuarios con paginación.

7.3.8. `UserController::crear()`

Método que muestra el formulario de creación de usuarios.

7.3.9. `UserController::editar()`

Método que actualiza el usuario en la base de datos, tanto para usuarios convencionales como para administradores.

7.3.10. `UserController::eliminar()`

Método que elimina el usuario de la base de datos.

7.4. `CarritoController.php`

Controlador del carrito. Contiene los métodos necesarios para la gestión del carrito.

7.4.1. `CarritoController::gestion()`

Método que muestra la página de gestión del carrito.

7.4.2. `CarritoController::add()`

Método que añade un producto al carrito.

7.4.3. `CarritoController::delete()`

Método que elimina un producto del carrito.

7.4.4. `CarritoController::clear()`

Método que vacía el carrito.

7.4.5. `CarritoController::up()`

Método que incrementa la cantidad de un producto en el carrito.

7.4.6. `CarritoController::down()`

Método que decrementa la cantidad de un producto en el carrito.

7.5. `PedidoController.php`

Controlador de pedidos. Contiene los métodos necesarios para la gestión de los pedidos.

7.5.1. `PedidoController::admin()`

Método que muestra toda la administración de los pedidos con paginación.

No se ha implementado.

7.5.2. `PedidoController::crear()`

Método que muestra el formulario de datos de envío.

7.5.3. `PedidoController::hacer()`

Método que guarda el pedido en la base de datos y redirige a la página de pedido listo.

No se ha implementado del todo.

7.5.4. `PedidoController::listo()`

Método que muestra la página de pedido listo.

7.5.5. `PedidoController::ver()`

Método que muestra la vista de un pedido para administradores.

No se ha implementado.

7.5.6. `PedidoController::confirmar()`

Método que confirma o actualiza en general el estado de un pedido para administradores.

No se ha implementado.

7.5.7. `PedidoController::eliminar()`

Método que elimina un pedido de la base de datos para administradores.

No se ha implementado.

8. Vistas (**views/**)

8.1. Diseño (**layout/**)

8.1.1. **header.php**

Vista de cabecera. La vista para usuarios convencionales y administradores es diferente.

Usuarios convencionales:

- **Título de la página.** Redirige a la página de productos recomendados.
- **Registrarse.** Redirige a la página de registro.
- **Iniciar sesión.** Redirige a la página de inicio de sesión.
- **Carrito.** Muestra el número de productos en el carrito y redirige a la página de gestión del carrito.
- **Usuario.** Muestra el nombre de usuario y redirige a la página de gestión del usuario.
- **Cerrar sesión.** Cierra la sesión del usuario.

Administradores (añade lo siguiente):

- **Categorías.** Redirige a la página de administración de categorías.
- **Productos.** Redirige a la página de administración de productos.
- **Pedidos.** No se ha implementado.
- **Usuarios.** Redirige a la página de administración de usuarios.

8.1.2. **footer.php**

Vista de pie de página. Contiene el año actual y el nombre del autor.

8.2. **categoria/**

8.2.1. **admin.php**

Vista de administración de categorías. Muestra todas las categorías con paginación. Contiene botones para crear, editar y eliminar categorías.

8.2.2. **crear.php**

Vista de creación de categorías. Contiene un formulario con el campo necesario para crear una categoría.

8.2.3. **editar.php**

Vista de edición de categorías. Contiene un formulario con el campo necesario para editar una categoría.

8.3. **producto/**

8.3.1. **admin.php**

Vista de administración de productos. Muestra todos los productos con paginación. Contiene botones para crear, editar y eliminar productos.

8.3.2. `crear.php`

Vista de creación de productos. Contiene un formulario con los campos necesarios para crear un producto.

8.3.3. `editar.php`

Vista de edición de productos. Contiene un formulario con los campos necesarios para editar un producto.

8.3.4. `ver.php`

Vista de un producto. Muestra la información de un producto. Contiene un botón para añadir el producto al carrito en caso de que haya stock. Si no, muestra un mensaje de que el producto está agotado.

8.3.5. `recomendados.php`

Vista de productos recomendados. Muestra 6 productos recomendados.

El usuario puede filtrar por categorías. En tal caso y si hubiera más de 6 productos en la categoría a visualizar (caso bastante probable), se mostrarán los 6 primeros productos de la categoría ordenadamente mediante paginación. Esto es configurable en el archivo `config.php`.

8.4. `usuario/`

8.4.1. `registrarse.php`

Vista de registro de usuarios. Contiene un formulario con los campos necesarios para registrarse.

8.4.2. `login.php`

Vista de inicio de sesión. Contiene un formulario con los campos necesarios para iniciar sesión y una casilla para recordar la sesión una semana (7 días) mediante una cookie.

8.4.3. `gestion.php`

Vista de gestión de usuarios. Muestra la información del usuario y contiene botones para guardar cambios y eliminar el usuario.

8.4.4. `admin.php`

Vista de administración de usuarios. Muestra todos los usuarios con paginación. Contiene botones para crear, editar y eliminar usuarios.

8.4.5. `crear.php`

Vista de creación de usuarios. Contiene un formulario con los campos necesarios para crear un usuario, incluyendo la posibilidad de establecer el rol del usuario.

8.4.6. `editar.php`

Vista de edición de usuarios. Contiene un formulario con los campos necesarios para editar un usuario, incluyendo la posibilidad de establecer el rol del usuario.

8.5. carrito/

8.5.1. gestion.php

Vista de gestión del carrito. Muestra los productos del carrito con la cantidad y el precio total. Contiene botones para eliminar un producto, vaciar el carrito, incrementar y decrementar la cantidad de un producto.

8.6. pedido/

8.6.1. admin.php

Vista de administración de pedidos. Muestra todos los pedidos con paginación. Contiene botones para ver, confirmar y eliminar pedidos.

No se ha implementado.

8.6.2. crear.php

Vista de datos de envío. Contiene un formulario con los campos necesarios de ubicación para realizar un pedido.

8.6.3. listo.php

Vista de pedido listo. Muestra un mensaje de que el pedido se ha realizado correctamente junto a un botón para seguir comprando.

8.6.4. ver.php

Vista de un pedido. Muestra la información de un pedido. Contiene botones para confirmar y eliminar el pedido.

No se ha implementado.