

# Maldito Café

## Introducción

## Estructura del proyecto

- **/img:** Contiene todos los recursos estáticos del proyecto como imágenes, íconos, fuentes, etc.
- **/clases:** Aquí se encuentran las clases PHP que manejan la lógica de la aplicación, como la conexión a la base de datos, manejo de sesiones, y la funcionalidad de las operaciones.
- **/styles:** Todos los archivos CSS y estilos del proyecto.
- **/views:** Contiene las vistas principales del sistema:
- **/admin:** Vistas específicas para el panel de administración.

## Archivos Principales

admin.php:

Archivo base para las vistas del área de administración. Este archivo sirve como plantilla o contenedor para las vistas administrativas, cargando los elementos comunes como cabecera, menú, pie de página, etc., que se reutilizan en todas las vistas de administración.

index.php:

Vista predeterminada para los usuarios. Es la página principal del sitio donde los usuarios pueden ver el catálogo de café, navegar por categorías, y acceder a la tienda. Además, maneja la funcionalidad del carrito de compras.

### Funcionalidad

Se obtiene el valor de sec desde la URL para definir qué vista mostrar (home, productos, carrito, etc.).

Si no se especifica una sección válida, se muestra una página 404.

## Sección: Tienda de Café (Vista General)

### Login

La sección de login permite a los usuarios autenticarse en el sitio. Si no están logueados, se muestra un formulario donde deben ingresar su email y contraseña. Al enviar el formulario, el sistema verifica las credenciales usando el método login() de la clase Usuario.

Si las credenciales son correctas, se inician las variables de sesión (\$\_SESSION['email'], \$\_SESSION['username'], etc.) con los datos del usuario y se redirige a la misma página.

Si las credenciales son incorrectas, se muestra un mensaje de error indicando que el email o contraseña son incorrectos.

Si el usuario ya está logueado, se muestra un mensaje de bienvenida con el nombre de usuario y opciones para cerrar sesión o volver al sitio. También se ofrece un enlace para registrarse si no tienen cuenta.

### Registro

La sección de registro permite a los nuevos usuarios crear una cuenta en el sitio.

1. Se muestra un formulario donde el usuario debe ingresar su nombre, email y contraseña.
2. El formulario se envía utilizando el método POST.
3. Cuando se envía el formulario, se recogen los datos y se crea un nuevo objeto de la clase Usuario.
4. El método Register() de la clase Usuario es llamado con los datos del formulario (username, email, password) y un rol predeterminado (en este caso, rol = 2).
5. A. Si el registro es exitoso, se muestra un mensaje confirmando que el usuario ha sido registrado con éxito y se le da la opción de ir al login con un enlace.  
B. Si el usuario ya tiene una cuenta, se le ofrece un enlace para acceder a la página de login.

**Este sistema permite la creación de nuevos usuarios en el sitio, asegurando que solo aquellos registrados puedan acceder a funcionalidades específicas.**

## Vista de productos

Esta sección muestra el catálogo completo de productos disponibles en el sitio, con la opción de filtrar por categorías.

Filtro de Categorías:

Se incluyen enlaces para filtrar productos según diferentes categorías como Café, Pastelería, Cápsulas, Vegano, Sin TACC.

La categoría activa se resalta mediante la clase `active`, que se ajusta en función del valor de `cat` en la URL.

Listado de Productos:

Se crean productos dinámicamente desde la base de datos utilizando el método `readAll()` de la clase `Producto`, que recupera todos los productos.

Al hacer clic en el botón "Ver" se redirige a la página de detalles del producto (`sec=detalle`).

Al hacer clic en el botón "Agregar al carrito", el producto se añade al carrito utilizando el parámetro `accion=agregar` en la URL.

## Filtrado de productos

La sección de filtrado permite a los usuarios ver productos específicos según la categoría seleccionada.

En la parte superior de la sección, se muestra una lista de categorías (Café, Pastelería, Cápsulas, Vegano, Sin TACC) que los usuarios pueden seleccionar.

Al hacer clic en una categoría, se agrega el parámetro `cat` en la URL, lo que indica qué categoría debe mostrarse.

La categoría activa se resalta mediante la clase **`active`**.

El nombre de la categoría seleccionada se muestra en el encabezado de la sección. Dependiendo del valor de `cat` en la URL, se cambia el título dinámicamente a algo como "Café", "Pastelería", "Cápsulas", etc.

Se llama al método `readCat()` de la clase `Producto`, que recupera los productos de la base de datos según la categoría seleccionada.

Cada producto tiene un botón que, al hacer clic, agrega el producto al carrito utilizando el parámetro `accion=agregar` en la URL.

Esta funcionalidad facilita a los usuarios encontrar y explorar productos según sus intereses específicos, mejorando la experiencia de compra al ofrecer filtros por categorías.

## Compra

### Carrito

1. El código comienza verificando si el usuario está logueado (\$\_SESSION['email']). Si no es así, lo redirige a la página de login para que se autentique antes de acceder al carrito.
2. Se obtiene el userID de la sesión del usuario logueado. Luego, se crea una instancia del objeto Carrito y se obtiene el carritoID utilizando el método iniciarCarrito, que se asegura de que el usuario tenga un carrito existente. Si no existe, se crea uno nuevo.
3. Si el carrito tiene productos, se listan todos los productos dentro del carrito, mostrando la imagen, nombre, precio, cantidad y total por producto.
4. También se calcula el total general del carrito sumando el total de cada producto.
5. Se proporciona un enlace para eliminar productos del carrito, redirigiendo a index.php?sec=eliminarProdCarrito.
6. Al final, se muestra el total del carrito y un botón para ir a la sección de pago (index.php?sec=pago).
7. Si el usuario hace clic en el enlace de **agregar** un producto al carrito (con la URL index.php?sec=carrito&accion=agregar&id=<producto\_id>), el producto se añade al carrito usando el método añadirProducto de la clase Carrito. Después, redirige al carrito nuevamente.
8. Si el usuario quiere **eliminar** un producto, se ejecuta la acción de eliminar usando la URL index.php?sec=carrito&accion=eliminar&id=<producto\_id>. Aquí se elimina el producto correspondiente mediante la llamada al método eliminarProducto de la clase Productos.
9. Si el carrito está vacío, se muestra un mensaje y un enlace para volver a la página de productos.

### Pago

Este código maneja el proceso de compra, creando una nueva orden para el usuario, vaciando los productos del carrito y transfiriéndolos a la tabla de órdenes. Cuando el usuario completa la compra, los productos de la tabla carrito\_items se eliminan (según el carritoID), pero los mismos productos se agregan a la tabla ordenes\_items vinculados a la nueva orden.

Se muestran los detalles del pago con instrucciones para hacer una transferencia, y también hay un enlace para que el usuario suba el comprobante de pago. Además, si la orden es pagada, se redirige al usuario a su lista de órdenes.

Luego, el estado de la orden se actualiza a "pagada" una vez que el usuario realiza el pago, y se muestran los detalles para completar la transferencia. El usuario puede cargar el comprobante de pago a través de un enlace, y si ya se ha realizado el pago, se redirige al usuario a la página de sus órdenes.

## Orden

La página de órdenes muestra al usuario todas sus compras pasadas. Al ingresar, se obtienen todas las órdenes relacionadas con el userID y se muestran los productos de cada orden. También se indica el estado de la orden (pendiente, pagada, cancelada) y el precio total.

Si la orden aún no ha sido pagada ni cancelada, se ofrecen botones para realizar el pago o cancelar la orden. Si el usuario decide cancelar, se realiza una actualización del estado de la orden para marcarla como cancelada. En el caso de que la orden esté lista para pagarse, el sistema dirige al usuario al proceso de pago, donde podrá realizar la transacción.

Este flujo también mantiene la integridad de los datos de la orden al mostrar los productos relacionados y asegurar que el estado se mantenga actualizado conforme avance el proceso de compra.

## Aprobación

# Sección de Adminsitrador

## Productos

### 1. Crear Producto

El administrador accede a un formulario para agregar un nuevo producto, donde puede ingresar los datos como el nombre, categoría, descripción, imagen, precio y stock.

POST: Al enviar el formulario, los datos son procesados y se intenta crear el nuevo producto en la base de datos.

Las categorías asociadas al producto se muestran como casillas de verificación.

Al enviar el formulario, se guarda el nuevo producto en la base de datos.

Si el proceso es exitoso, se muestra un mensaje de confirmación y el administrador es redirigido a la lista de productos.

## 2. Eliminar Producto

En la lista de productos, el administrador tiene la opción de eliminar un producto. Al hacer clic en "Eliminar", se elimina el producto seleccionado de la base de datos. Después de la eliminación, se muestra un mensaje de éxito y se redirige a la página principal de productos.

## 3. Editar Producto

El administrador puede modificar los datos de un producto existente seleccionando "Modificar" junto al producto en la lista.

Si la solicitud es POST, los datos modificados se actualizan en la base de datos.

Si es GET, se cargan los datos actuales para que el administrador pueda editarlos.

Las categorías asociadas al producto se muestran como casillas de verificación.

Se carga un formulario con la información actual del producto, que puede ser editada.

Una vez que se envían los cambios, el producto se actualiza en la base de datos.

Tras la actualización, se redirige a la lista de productos con los cambios reflejados.

## Usuarios

La gestión de usuarios permite al administrador visualizar, editar y modificar el rol de los usuarios registrados:

1. **Listado de usuarios:** Muestra una tabla con los detalles básicos de los usuarios, como nombre, correo electrónico y rol. Además, permite al administrador realizar acciones como editar través de un botón.
2. **Actualizar datos de usuario:** Cuando se selecciona un usuario para editar, el sistema permite modificar su nombre, correo electrónico y rol (Administrador o Cliente). El rol se selecciona a través de un dropdown.
3. **Flujo de actualización:** Los cambios se procesan mediante un formulario POST y se actualizan en la base de datos. Al completarse la actualización, el sistema redirige al administrador de nuevo al listado de usuarios.

Este sistema facilita la administración de usuarios y sus roles dentro de la plataforma.

## Ordenes

La gestión de órdenes permite al administrador visualizar y actualizar el estado de cada pedido realizado en la plataforma:

1. **Listado de órdenes:** Muestra una tabla con el número de orden, el correo del usuario, el precio total y el estado actual de la orden. Cada fila tiene un enlace para editar el estado de la orden.
2. **Editar estado de la orden:** Al hacer clic en "Editar estado", se redirige al administrador a un formulario donde puede seleccionar un nuevo estado para la orden. Los estados disponibles se obtienen desde la base de datos y se presentan en un dropdown. Los estados son:
  - **Pendiente de pago:** La orden está en espera de confirmación de pago.
  - **Pago en revisión:** El pago está siendo verificado por el sistema.
  - **Pago aprobado:** El pago ha sido confirmado y la orden está siendo procesada.
  - **Pago rechazado:** El pago no fue aceptado por alguna razón.
3. **Flujo de actualización:** Cuando el administrador selecciona un estado y envía el formulario, se actualiza el estado de la orden en la base de datos y se redirige de vuelta a la lista de órdenes.

Esto permite al administrador tener control sobre el proceso de los pedidos y asegurarse de que los estados de las órdenes se mantengan actualizados.

## Seguridad

Se ha implementado seguridad para asegurar que solo los usuarios autorizados accedan a la administración y puedan realizar compras.

- **Autenticación y roles:** Solo los usuarios logueados y con rol de Administrador pueden acceder a la administración. Si un usuario común intenta acceder, se le redirige a una página de acceso denegado.
- **Acceso a secciones válidas:** Se controlan las secciones que se pueden acceder para evitar rutas no autorizadas. Si un usuario intenta acceder a una sección incorrecta, se muestra un error 404.

- **Compras solo para usuarios logueados:** Solo los usuarios autenticados pueden realizar compras. Si no están logueados, se les redirige a la página de inicio de sesión.

Estas medidas protegen el sistema y garantizan que solo los usuarios autorizados puedan gestionar la tienda o realizar compras.