|  |
| --- |
| DAW |
| Golden Cotton |
| Manual Administrador |

|  |
| --- |
| Miguel Sánchez Porras |

Contenido

# Base de datos

## BD.sql

### Creación de la base de datos

Lo primero que haremos será borrar la base de datos si existe y la volveremos a crear con las características que queremos.

DROP DATABASE IF EXISTS oro;

CREATE DATABASE IF NOT EXISTS oro CHARACTER SET utf8 COLLATE utf8\_general\_ci;

USE oro;

### Creación de las tablas

Eliminamos las tablas si existen y vamos creando las tablas según las características que queremos

DROP TABLE IF EXISTS USUARIO;

DROP TABLE IF EXISTS PRODUCTO;

DROP TABLE IF EXISTS PEDIDOS;

DROP TABLE IF EXISTS VALORACIONES;

DROP TABLE IF EXISTS STOCK;

En este caso creamos la tabla usuario si no existe con las siguientes columnas siendo 6 varchar de 255 1 de tipo date y otro de tipo boolean de los cuales la columna DNI es la llave primaria.

CREATE TABLE IF NOT EXISTS USUARIO(

    DNI VARCHAR(255) PRIMARY KEY,

    Email VARCHAR(255) NOT NULL,

    Nombre VARCHAR(255) NOT NULL,

    Apellido VARCHAR(255) NOT NULL,

    Clave VARCHAR(255) NOT NULL,

    Fecha\_naci DATE NOT NULL,

    Es\_Admin BOOLEAN DEFAULT 0,

    Direccion VARCHAR(255) NOT NULL

);

Para esta tabla llamada categoría con una única columna siendo también llave primaria de tipo varchar.

CREATE TABLE IF NOT EXISTS CATEGORIAS(

    categorias VARCHAR(255) PRIMARY KEY

);

Esta tabla producto tiene 2 columnas de tipo int siendo la primera la llave primaria tiene 6 columnas de tipo varchar de las cuales características es una clave foránea una de tipo decimal y otro de tipo boolean.

CREATE TABLE IF NOT EXISTS PRODUCTO(

    ID\_Producto INT AUTO\_INCREMENT PRIMARY KEY,

    Nombre VARCHAR(255) NOT NULL,

    Categorias VARCHAR(255) NOT NULL,

    Precio DECIMAL(10,2) NOT NULL,

    IVA INT NOT NULL,

    Es\_Promocionado BOOLEAN DEFAULT 0,

    Imagen1 VARCHAR(255) NOT NULL,

    Imagen2 VARCHAR(255),

    Imagen3 VARCHAR(255),

    Descripcion VARCHAR(255),

    FOREIGN KEY(Categorias) REFERENCES CATEGORIAS(categorias)

);

Esta tabla almacena los productos favoritos de los usuarios teniendo una columna int y otra varchar siendo las 2 claves foráneas

CREATE TABLE IF NOT EXISTS PRODUCTOS\_FAVORITOS(

    ID\_Producto INT,

    DNI VARCHAR(255),

    FOREIGN KEY(ID\_Producto) REFERENCES PRODUCTO(ID\_Producto),

    FOREIGN KEY(DNI) REFERENCES USUARIO(DNI)

);

Esta tabla pedidos almacena los datos en una columna de valor int siendo esta la clave primaria, 4 varchar de las cuales 1 de ellas es una clave foránea también tiene una columna enum y una columna.

CREATE TABLE IF NOT EXISTS PEDIDOS(

    ID INT AUTO\_INCREMENT PRIMARY KEY,

    Nombre VARCHAR(255) NOT NULL,

    Estado ENUM('Almacen','Paqueteria', 'Entregado') NOT NULL,

    Fecha DATE NOT NULL,

    Descripcion VARCHAR(255),

    DNI VARCHAR(255) NOT NULL,

    Direccion VARCHAR(255) NOT NULL,

    FOREIGN KEY(DNI) REFERENCES USUARIO(DNI)

);

Esta tabla almacena las líneas de pedido los cuales tiene 6 columnas tipo int todas ellas sin poder ser nulas y un decimal que tampoco puede ser nula.

CREATE TABLE IF NOT EXISTS LINEAS\_PEDIDO(

    numPedido INT NOT NULL,

    numLinea INT NOT NULL,

    ID\_Producto INT NOT NULL,

    unidades int NOT NULL,

    IVA INT NOT NULL,

    Precio DECIMAL(10,2) NOT NULL,

    descuento INT NOT NULL,

    PRIMARY KEY (numPedido, numLinea),

    FOREIGN KEY (numPedido) REFERENCES PEDIDOS(ID),

    FOREIGN KEY (ID\_Producto) REFERENCES PRODUCTO(ID\_Producto)

);

En esta tabla se almacenan las valoraciones de los usuarios.

en stock se almacena la cantidad de unidades que tienen cada talla y color de un producto siendo todas ellas no nulas.

CREATE TABLE IF NOT EXISTS STOCK(

    Unidades INT NOT NULL,

    Color ENUM('Negro','Blanco','Azul','Rojo','Verde') NOT NULL,

    Talla ENUM('XXS','XS','S','M','L','XL','XXL','XXXL') NOT NULL,

    ID\_Producto INT NOT NULL,

    PRIMARY KEY (Color, Talla, ID\_Producto),

    FOREIGN KEY(ID\_Producto) REFERENCES PRODUCTO(ID\_Producto)

);

CREATE TABLE IF NOT EXISTS VALORACIONES(

    ID INT AUTO\_INCREMENT PRIMARY KEY,

    Estrellas INT NOT NULL,

    Fecha DATE NOT NULL,

    Comentario VARCHAR(255),

    DNI VARCHAR(255) NOT NULL,

    ID\_Producto INT NOT NULL,

    FOREIGN KEY(DNI) REFERENCES USUARIO(DNI),

    FOREIGN KEY(ID\_Producto) REFERENCES PRODUCTO(ID\_Producto)

);

### Insertar datos en las tablas

Ahora pasamos a insertar datos en las tablas para que no estén vacías y tener datos de referencia.

INSERT INTO USUARIO(DNI,Email, Nombre, Apellido, Clave, Fecha\_naci, Es\_Admin)

VALUES ('04514257R','correo@admin.com', 'Soy', 'Admin', 'admin123', '2000-6-4', 1);

INSERT INTO USUARIO(DNI,Email, Nombre, Apellido, Clave, Fecha\_naci, Es\_Admin)

VALUES ('14514258H','correo@cliente.com', 'timmy', 'timmyton', 'user123', '2001-7-19', 0);

INSERT INTO CATEGORIAS(categorias)

VALUES ('Camiseta');

INSERT INTO CATEGORIAS(categorias)

VALUES ('Sudadera');

INSERT INTO CATEGORIAS(categorias)

VALUES ('Accesorios');

INSERT INTO PEDIDOS(Nombre, Estado, Fecha, Descripcion, DNI)

VALUES ('Pedido prueba 1','Almacen', '2025-03-18', 'pedido de prueba 1', '14514258H');

INSERT INTO PEDIDOS(Nombre, Estado, Fecha, Descripcion, DNI)

VALUES ('Pedido prueba 2','Paqueteria', '2025-03-18', 'pedido de prueba 2', '14514258H');

INSERT INTO PEDIDOS(Nombre, Estado, Fecha, Descripcion, DNI)

VALUES ('Pedido prueba 3','Entregado', '2025-03-18', 'pedido de prueba 3', '14514258H');

INSERT INTO CATEGORIAS(categorias)

VALUES ('Camiseta');

INSERT INTO CATEGORIAS(categorias)

VALUES ('Sudadera');

INSERT INTO CATEGORIAS(categorias)

VALUES ('Accesorios');

INSERT INTO PRODUCTO(ID\_Producto, Nombre, Categorias, Precio, Imagen1, Imagen2, Descripcion)

VALUES (1,'CAMISETA MONSTER TRIO','Camiseta',22.95, '../imagenes/1.jpg', '../imagenes/2.jpg', 'Camiseta Monster Trio de la marca de anime Nakama Clothing. Se han reunido los 3 piratas más fuertes de la tripulación del futuro rey de los piratas, para hacer frente a cualquier calamidad.');

INSERT INTO PRODUCTO(ID\_Producto, Nombre, Categorias, Precio, Imagen1, Imagen2, Descripcion)

VALUES (2,'CAMISETA EVIL GHOST','Camiseta', 22.95,'../imagenes/3.jpg','../imagenes/4.jpg','Camiseta morado Evil Ghost de la marca de anime Nakama Clothing. Soy un canalla difícil de atrapar, incorpóreo, de ojos rojos y gran sonrisa. Cuando menos te lo esperes un buen lengüetazo te vas a llevar Muajajajaja');

INSERT INTO PRODUCTO(ID\_Producto, Nombre, Categorias, Precio, Imagen1, Imagen2, Descripcion)

VALUES (3,'SUDADERA BRAVE PIG','Sudadera',45.95,'../imagenes/5.jpg','../imagenes/6.jpg','Sudadera Brave Pig de la marca de anime Nakama Clothing. Soy el Cazador de demonios enmascarado ¡RESPIRACIÓN!');

INSERT INTO PRODUCTO(ID\_Producto, Nombre, Categorias, Precio, Imagen1, Imagen2, Descripcion)

VALUES (4,'SUDADERA BLOODY SAW','Sudadera',45.95,'../imagenes/7.jpg','../imagenes/8.jpg','Sudadera Bloody Saw de la marca de anime Nakama Clothing. Cortaré y desmembraré lo que haga falta con tal de conseguir un beso de una chica.');

INSERT INTO PRODUCTO(ID\_Producto, Nombre, Categorias, Precio, Imagen1, Imagen2, Descripcion)

VALUES (5,'Anime Sword Art Online Sao Denim Jacket Chaqueta Vaquera Adulto Cosplay Jeans Hoodie Outwear Abrigo','Sudadera', 45.95,'../imagenes/9.jpg','../imagenes/10.jpg','Anime Sword Art Online Sao Denim Jacket Chaqueta Vaquera Adulto Cosplay Jeans Hoodie Outwear Abrigo Azul 1 M');

INSERT INTO PRODUCTO(ID\_Producto, Nombre, Categorias, Precio, Imagen1, Imagen2, Descripcion)

VALUES (6,'GORRA FIVE CLOVER','Accesorios',19.95,'../imagenes/11.jpg','../imagenes/12.jpg','Gorra Five Clover de la marca de anime Nakama Clothing. A pesar de no tener una gota de magia, entrenaré duro para proteger este reino.');

INSERT INTO PRODUCTO(ID\_Producto, Nombre, Categorias, Precio, Imagen1, Imagen2, Descripcion)

VALUES (7,'GORRO DEMON SAND','Accesorios',19.95,'../imagenes/13.jpg','../imagenes/14.jpg','Gorro Demon Sand de la marca de anime Nakama Clothing. Sufrí mucho en mi pasado, más que el dolor que sientes cuando se te mete arena dentro del ojo. Ahora sufriréis mi venganza.');

INSERT INTO PRODUCTO(ID\_Producto, Nombre, Categorias, Precio, Imagen1, Imagen2, Descripcion)

VALUES (8,'MOCHILA THE GOD ELEVEN','Accesorios',24.95,'../imagenes/15.jpg','../imagenes/16.jpg','Mochila naranja The God Eleven de la marca de anime Nakama Clothing. Arriba, chuta, la victoria es tuya! Con este equipo la portería será impenetrable!');

INSERT INTO PRODUCTO(ID\_Producto, Nombre, Categorias, Precio, Imagen1, Imagen2, Descripcion)

VALUES (9,'MOCHILA WONDERFUL PIG','Accesorios',15.95,'../imagenes/17.jpg','../imagenes/18.jpg','Mochila Wonderful pig de la marca de anime Nakama Clothing. Vive las mejores aventuras, junto al cerdito más valiente que la imaginación de un niño de 5 años Japonés ha creado!');

INSERT INTO PRODUCTO(ID\_Producto, Nombre, Categorias, Precio, Imagen1, Descripcion)

VALUES (10,'LLAVERO DANCING SWEET','Accesorios',5.95,'../imagenes/19.jpg','Llavero Dancing Sweet de la marca de anime Nakama Clothing. No tener magia, no hará que no pueda bailar mientras tenga mis profiteroles.');

INSERT INTO PRODUCTO(ID\_Producto, Nombre, Categorias, Precio, Imagen1, Descripcion)

VALUES (11,'LLAVERO BLOODY SAW','Accesorios',5.95,'../imagenes/20.jpg','Llavero Bloody Saw de la marca de anime Nakama Clothing. Cortaré y desmembraré lo que haga falta con tal de conseguir un beso de una chica.');

INSERT INTO PRODUCTO(ID\_Producto, Nombre, Categorias, Precio, Imagen1, Imagen2, Descripcion)

VALUES (12,'POSAVASOS ALIEN FROG','Accesorios',15.95,'../imagenes/21.jpg','../imagenes/22.jpg','Posavasos Alien Frog de la marca de anime Nakama Clothing. A pesar de tener un aspecto animal adorable, somos unos alienígenas temibles. Pensamos conquistar el planeta tierra, jurado por las nuestras ancas!');

INSERT INTO PRODUCTO(ID\_Producto, Nombre, Categorias, Precio, Imagen1, Imagen2, Descripcion)

VALUES (13,'MUÑEQUERA NAKAMA','Accesorios',4.95,'../imagenes/23.jpg','../imagenes/24.jpg','Muñequera Nakama de la marca de anime Nakama Clothing. Recrea el símbolo de la amistad pirata.');

INSERT INTO STOCK(Unidades, Color, Talla, ID\_Producto)

VALUES (50, 'Negro', 'L', 1);

INSERT INTO STOCK(Unidades, Color, Talla, ID\_Producto)

VALUES (25, 'Blanco', 'XL', 1);

INSERT INTO STOCK(Unidades, Color, Talla, ID\_Producto)

VALUES (50, 'Azul', 'L', 2);

INSERT INTO STOCK(Unidades, Color, Talla, ID\_Producto)

VALUES (30, 'Negro', 'L', 2);

INSERT INTO STOCK(Unidades, Color, Talla, ID\_Producto)

VALUES (50, 'Negro', 'S', 5);

INSERT INTO STOCK(Unidades, Color, Talla, ID\_Producto)

VALUES (50, 'Negro', 'XXL', 5);

INSERT INTO STOCK(Unidades, ID\_Producto)

VALUES (10, 10);

INSERT INTO STOCK(Unidades, ID\_Producto)

VALUES (5, 12);

# PHP

## comprobarSesion.php

en este php comprobamos que la sesión este inicializada en caso contrario nos redirige a index.php

<?php

function comprobar\_sesion(){

    if(session\_status() == PHP\_SESSION\_NONE){

        session\_start();

    }

    if(!isset($\_SESSION['activa']) || empty($\_SESSION['activa'])){

        header("Location: /Tienda/index.php");

        exit;

    }}

## comprobarUsuario. Php

Se comprueba que el usuario haya insertado bien los datos

<?php

include 'conexionBD.php';

session\_start();

if (!empty($\_SESSION['activa'])) {

    header("Location: ./inicioSesion.php");

}else {

    if(isset($\_POST['Email']) && isset($\_POST['Clave'])){

        function validar($dato){

            $dato = trim($dato);

            $dato = stripcslashes($dato);

            $dato = htmlspecialchars($dato);

            return $dato;

  }

        $Email = validar($\_POST['Email']);

        $Clave = validar($\_POST['Clave']);

        if (empty($Email)) {

            header("Location: ./inicioSesion.php?error=El Correo Electronico es Requerido");

            exit();

        }elseif(empty($Clave)){

            header("Location: ./inicioSesion.php?error=La Contraseña es Requerida");

            exit();

        }else {

            $sql = mysqli\_query($conexion,"SELECT \* FROM USUARIO WHERE Email = '$Email' AND Clave = '$Clave'");

            $resultado = mysqli\_num\_rows($sql);

            if ($resultado === 1) {

                $datos = mysqli\_fetch\_array($sql);

                $\_SESSION['activa'] = true;

                $\_SESSION['DNI'] = $datos['DNI'];

                $\_SESSION['Email'] = $datos['Email'];

                $\_SESSION['Nombre'] = $datos['Nombre'];

                $\_SESSION['Apellido'] = $datos['Apellido'];

                $\_SESSION['Clave'] = $datos['Clave'];

                $\_SESSION['Fecha\_naci'] = $datos['Fecha\_naci'];

                $\_SESSION['Es\_Admin'] = $datos['Es\_Admin'];

                $\_SESSION['Direccion'] = $datos['Direccion'];

                header("Location: inicio.php");

            }else {

                header("Location: ./inicioSesion.php?error=El Correo electronico o La Contraseña son incorrectos 2"); session\_destroy();}

        }

    }else { header("Location: ./inicioSesion.php");

        exit(); }}?>

## ConexiónBD.php

esta consulta lo que hace es realizar una conexión con la base de datos.

<?php

$conexion = mysqli\_connect("localhost", "root", "", "oro") or die("No se ha conectado a la base de datos, comprueba conexionBD.php");

mysqli\_set\_charset($conexion, "utf8");

?>

## Nav1.php

Contiene un nav genérico que se reparte entre los demás php, en esta parte del código lo que se hace es que si la sesion esta iniciado en vez de Perfil ponga el nombre del usuario y cerrar sesion, también en el enlace a pedidos le agregamos el correo del usuario como id.

<li class="nav-item active">

                    <a class="nav-link" href="/Tienda/PHP/perfil.php"><?php echo empty($\_SESSION['Nombre']) ? 'Perfil' : $\_SESSION['Nombre']; ?></a>

                </li>

 <?php

                    if (session\_status() == PHP\_SESSION\_NONE) {

                        session\_start();

                    }

                    if (isset($\_SESSION['activa']) && $\_SESSION['activa'] == true) {

                ?>

                <li class="nav-item">

                    <a class="nav-link" href="/Tienda/PHP/pedidos.php?id=<?php echo $\_SESSION['Email']; ?>">Pedidos</a>

                </li>

## Nav2.php

Contiene un nav genérico que se reparte entre los demás php que solo se muestra para los administradores.

<?php

    if (session\_status() == PHP\_SESSION\_NONE) {

        session\_start();

    }

    if (isset($\_SESSION['Es\_Admin']) && $\_SESSION['Es\_Admin'] == 1) {

?>

## Header.php

Contiene la etiqueta head con todos los links a los estilos de css.

## Footer.php

Una etiquieta footer que se reparte entre todo el código php que muestra las redes sociales.

## carrito.php

se encarga de mostrar los datos datos del localstorage para realizar la compra de los productos

<script>

document.getElementById('form-pedido').addEventListener('submit', function(event){

    const carrito = JSON.parse(localStorage.getItem('carrito')) || {};

    if(Object.keys(carrito).length === 0){

        event.preventDefault();

        return false;

    }

    document.getElementById('carrito\_json').value = JSON.stringify(carrito);

});

</script>

## categoría.php

en este archivo mostramos todos los productos de manera paginada y también se pueden filtrar para facilitar encontrar el producto deseado además también de poder buscar el producto por el nombre.

<script>

document.getElementById('form-pedido').addEventListener('submit', function(event){

    const carrito = JSON.parse(localStorage.getItem('carrito')) || {};

    if(Object.keys(carrito).length === 0){

        event.preventDefault();

        return false;

    }

    document.getElementById('carrito\_json').value = JSON.stringify(carrito);

});

</script>

## cerrarSesion.php

destruye la sesion para que se tenga que volver a iniciar que una vez hecho redirige al index.

<?php

session\_start();

session\_unset();

session\_destroy();

header("Location: /Tienda/index.php");

?>

## crearProducto.php

es un formulario para crear nuevos productos el cual tiene un select para las categorías.

$categorias = [];

$sql = "SELECT categorias FROM CATEGORIAS";

$result = $conexion->query($sql);

if ($result && $result->num\_rows > 0) {

    while ($row = $result->fetch\_assoc()) {

        $categorias[] = $row['categorias'];}}

<select name="categoria">

                <?php foreach ($categorias as $categoria): ?>

                    <option value="<?= $categoria ?>"

                        <?= (isset($\_GET['categoria']) && $\_GET['categoria'] == $categoria) ? 'selected' : '' ?>>

                        <?= $categoria ?>

                    </option>

                <?php endforeach; ?>

            </select><br>

## editarPerfil.php

es un formulario que se carga los productos previamente para mostrarlo gracias el $\_GET.

$correo = $\_GET['id'];

$sql = mysqli\_query($conexion, "SELECT DNI, Email, Nombre, Apellido, Fecha\_naci, Direccion FROM USUARIO WHERE Email = '$correo'");

$resultado = mysqli\_num\_rows($sql);

## editarProducto.php

en este formulario se puede editar los productos también gracias el $\_GET.

$id = $\_GET['id'];

$sqlProducto = "SELECT \* FROM PRODUCTO WHERE ID\_Producto = $id";

$resultProducto = $conexion->query($sqlProducto);

$producto = $resultProducto->fetch\_assoc();

## eliminarFavorito.php

consulta que elimina el producto que se haya añadido a favorito.

<?php

include 'conexionBD.php';

include 'comprobarSesion.php';

comprobar\_sesion();

if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST['id\_producto'])) {

    $dni = $\_SESSION['DNI'];

    $idProducto = $\_POST['id\_producto'];

    $sql = mysqli\_query($conexion, "DELETE FROM PRODUCTOS\_FAVORITOS WHERE DNI = '$dni' AND ID\_Producto = $idProducto");

}

header("Location: favoritos.php");

exit();

?>

## favoritos.php

muestra todos los productos favoritos del usuario al cual la variable $dni sea igual al de la tabla.

$dni = $\_SESSION['DNI'];

$sql = mysqli\_query($conexion, "SELECT P.\* FROM PRODUCTO P INNER JOIN PRODUCTOS\_FAVORITOS PF ON P.ID\_Producto = PF.ID\_Producto WHERE PF.DNI = '$dni'");

$favoritos = [];

if ($sql) {

    while ($fila = mysqli\_fetch\_assoc($sql)) {

        $favoritos[] = $fila;

    }}

header("Location: favoritos.php");

exit();

?>

## guardarCategoria.php

consulta para guardar los cambios que se han hecho.

<?php

session\_start();

include 'conexionBD.php';

if (isset($\_POST['categoria']) && !empty($\_POST['categoria'])) {

    $categoria = $\_POST['categoria'];

    $sql = "INSERT INTO CATEGORIAS(categorias) VALUES('$categoria')";

    $query = mysqli\_query($conexion, $sql);

    if ($query) {

        header("Location: nuevaCategoria.php?success=Categoria guardada");

        exit();

    } else {

        header("Location: nuevaCategoria.php?error=Categoria Existente");

        exit();

    }

} else {

    header("Location: nuevaCategoria.php?error=Categoria vacia");

    exit();

}

?>

## guardarEditarProducto.php

consulta para guardar los cambios realizados al producto y elimina las fotos antiguas con unlink() en caso de que se cambien.

if ($imagen3 != '') unlink($imagen3);

## guardarEditarUsuario.php

guarda los cambios realizados del usuario.

## guardarFavorito.php

guarda en la tabla el producto seleccionado.

## guardarPedido.php

guarda los datos de la compra tanto en la tabla pedidos con la de líneas de pedidos.

$sqlPedido = "INSERT INTO PEDIDOS (Nombre, Estado, Fecha, Descripcion, DNI, Direccion)

              VALUES ('$nombreUsuario', '$estado', '$fechaPedido', '$descripcion', '$dniUsuario', '$direccionUsuario')";

$queryPedido = mysqli\_query($conexion, $sqlPedido);

$sqlLinea = "INSERT INTO LINEAS\_PEDIDO (numPedido, numLinea, ID\_Producto, unidades, IVA, Precio, descuento)

                 VALUES ($idPedido, $numLinea, $idProducto, $unidades, $iva, $precio, $descuento)";

    $queryLinea = mysqli\_query($conexion, $sqlLinea)

## guardarProducto.php

guarda los datos del producto creado en su correspondiente tabla.

## guardarRegistrarUsuario.php

inserta al usuario creado y muestra un mensaje si se ha creado correctamente o no.

## guardarStock.php

guarda el stock mediante un insert y devuelve al formulario mostrando los cambios.

## Inicio.php

Es un archivo muestra idéntico a index.php que su función es mostrar los productos promocionados cuando se inicie la sesión.

<?php

$sql = "SELECT \* FROM PRODUCTO WHERE Es\_Promocionado = 1";

$resultado = mysqli\_query($conexion, $sql);

?>

## inicioSesion.php

<?php if ($resultado && mysqli\_num\_rows($resultado) > 0): ?>

            <?php while ($producto = mysqli\_fetch\_assoc($resultado)): ?>

                <div class="col-6 col-md-4 col-lg-3 mb-4">

                    <div class="card h-100">

                        <img src="<?= $producto['Imagen1'] ?>" class="card-img-top" alt="<?= $producto['Nombre'] ?>">

                        <div class="card-body d-flex flex-column">

                            <h5 class="card-title"><?= $producto['Nombre'] ?></h5>

                            <p class="card-text text-truncate"><?= $producto['Descripcion'] ?></p>

                            <p class="fw-bold"><?= $producto['Precio'], 2 ?> €</p>

                            <a href="producto.php?id=<?= $producto['ID\_Producto'] ?>" class="btn btn-primary mt-auto">Ver detalle</a>

                        </div>

                    </div>

                </div>

            <?php endwhile; ?>

        <?php else: ?>

            <p>No hay productos promocionados en este momento.</p>

        <?php endif; ?>

    </div>

formulario para el inicio de la sesion que en caso de llegar destruye la sesion.

<?php

session\_start();

session\_unset();

session\_destroy();

?>

## listaProducto.php

muestra una lista de todos los productos los cuales puedes editar y comprobar el stock.

<?php

                if ($resultado > 0) {

                    while ($producto = mysqli\_fetch\_assoc($sql)) {

                        echo "<tr>";

                        echo "<td class='d-inline-block text-truncate' style='max-width: 250px;''>" . $producto['Nombre'] . "</td>";

                        echo "<td>" . $producto['Precio'] . "€" . "</td>";

                        echo "<td>" . ($producto['Es\_Promocionado'] == 1 ? 'Sí' : 'No') . "</td>";

                        echo "<td>" . $producto['Categorias'] . "</td>";

                        echo "<td><a href='editarProducto.php?id=" . $producto['ID\_Producto'] . "' class='btn btn-outline-primary'>Editar</a></td>";

                        echo "<td><a href='stock.php?id=" . $producto['ID\_Producto'] . "' class='btn btn-outline-primary'>Stock</a></td>";

                        echo "</tr>";

                    }

                }

                ?>

## nuevaCategoria.php

muestra un formulario con una tabla las cuales muestra las categorías que hay en la base de datos para que el administrador lo tenga de referencia

<h2>Categorías existentes:</h2>

        <ul style="list-style: none;">

        <?php

            foreach ($categorias as $cat) {

                echo "<li>$cat</li>";  }

        ?></ul>

## pedidos.php

muestra los pedidos del usuario que coincidan con el $dni.

<div class="profile-container"><div class="profile-card"><h3>Datos del usuario</h3>

            <p><strong>Correo Electrónico:</strong> <?php echo $\_SESSION['Email']; ?></p>

            <p><strong>Nombre:</strong>  <?php echo $\_SESSION['Nombre']; ?></p>

            <p><strong>Apellido:</strong>  <?php echo $\_SESSION['Apellido']; ?></p>

            <p><strong>Fecha de nacimiento:</strong> <?php echo $\_SESSION['Fecha\_naci']; ?></p><a href="/Tienda/PHP/editarPerfil.php?id=<?php echo $\_SESSION['Email']; ?>" class="btn btn-outline-primary btn-edit-profile">Editar Perfil</a>

        </div>

    </div>

<<tbody>

                <?php

                if ($resultado > 0) {

                    while ($pedido = mysqli\_fetch\_assoc($sql)) {

                        echo "<tr>";

                        echo "<td>" . $pedido['Nombre'] . "</td>";

                        echo "<td>" . $pedido['Estado'] . "</td>";

                        echo "<td>" . $pedido['Fecha'] . "</td>";

                        echo "<td>" . $pedido['Descripcion'] . "</td>";

                        echo "<td>" . $pedido['DNI'] . "</td>";

                        echo "<td>" . $pedido['Direccion'] . "</td>";

                        echo "<td>" . "<a href='pedidos.php' class='btn btn-outline-primary btn-edit-profile'>Detalles</a>" . "</td>";

                        echo "</tr>";

                    }

                }

                ?>

            </tbody>

## Perfil.php

Muestra los datos del usuario.

## Producto.php

Muestra los datos en concreto de un producto a través del $\_GET el cual puedes añadir a favoritos o a la cesta y las valoraciones que tiene.

$id = $\_GET['id'];

$sql = mysqli\_query($conexion, "SELECT \* FROM PRODUCTO WHERE ID\_Producto = '$id'");

$resultado = mysqli\_num\_rows($sql);

$producto = [];

if ($resultado == 0) {

    header('Location: categoria.php');

    exit();

} else {

    $producto = mysqli\_fetch\_assoc($sql);}

$sqlValoraciones = mysqli\_query($conexion, "SELECT V.\*, U.Nombre, U.Apellido FROM VALORACIONES V JOIN USUARIO U ON V.DNI = U.DNI WHERE V.ID\_Producto = '$id' ORDER BY V.Fecha DESC");

$valoraciones = [];

if ($sqlValoraciones) {

    while ($fila = mysqli\_fetch\_assoc($sqlValoraciones)) {

        $valoraciones[] = $fila;

    }

}

## registrarUsuario.php

formulario para registrar a los nuevos usuarios con la acción apuntando a guardarUsuario.php.

<form action="guardarRegistrarUsuario.php" method="POST">

## stock.php

formulario para crear el stock de un producto en especifico que sabremos gracias a $\_GET, también muestra el stock que tienen cada talla y color

$id\_producto = $\_GET['id'];

$sql = mysqli\_query($conexion, "SELECT \* FROM STOCK WHERE ID\_Producto = $id\_producto");

$resultado = mysqli\_num\_rows($sql);

while ($item = mysqli\_fetch\_assoc($sql)) {

                echo "<tr>";

                echo "<td>" . $item['Unidades'] . "</td>";

                echo "<td>" . $item['Color'] . "</td>";

                echo "<td>" . $item['Talla'] . "</td>";

                echo "</tr>";

            }

## index.php

página principal que muestra un carrusel y donde acaba el usuario al entrar a la pagina web o cuando cierra sesión.

<ol class="carousel-indicators">

    <li data-bs-target="#carouselExampleIndicators" data-bs-slide-to="0" class="active"></li>

    <li data-bs-target="#carouselExampleIndicators" data-bs-slide-to="1"></li>

    <li data-bs-target="#carouselExampleIndicators" data-bs-slide-to="2"></li>

  </ol>

<button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide="prev">

    <span class="carousel-control-prev-icon" aria-hidden="true"></span>

    <span class="visually-hidden">Previous</span>

  </button>

  <button class="carousel-control-next" type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide="next">

    <span class="carousel-control-next-icon" aria-hidden="true"></span>

    <span class="visually-hidden">Next</span>

# JS

## AnadirCarrito.js

Archivo que esta a la espera en la pagina producto para cuando el usuario le de a añadir al carrito cree una localstorage si esta vacío y añada los datos del producto o añada 1 si ya está en el localstorage

document.addEventListener("DOMContentLoaded", () => {

    const btn = document.getElementById("anadirCarrito");

    let carrito = JSON.parse(localStorage.getItem("carrito")) || {};

    btn.addEventListener("click", () => {

      const id = btn.dataset.id;

      const nombre = btn.dataset.nombre;

      const descripcion = btn.dataset.descripcion;

      const precio = btn.dataset.precio;

      const imagen1 = btn.dataset.imagen;

      if (carrito.hasOwnProperty(id)) {

        carrito[id].cantidad += 1;

      } else {

        carrito[id] = {

          nombre: nombre,

          descripcion: descripcion,

          precio: precio,

          imagen1: imagen1,

          cantidad: 1 };}

      localStorage.setItem("carrito", JSON.stringify(carrito));

      console.log("Carrito actualizado:", carrito);}); });

## cambiarImagen.js

función que cambia las imágenes cuando se le hace clic a la imagen deseada

## functions.js

thumbs.onclick = function(event) {

    let thumbnail = event.target.closest('a');

    if (!thumbnail) return;

    showThumbnail(thumbnail.href, thumbnail.title);

    event.preventDefault();

  }

  function showThumbnail(href, title) {

    let largeImg = document.getElementById('largeImg');

    largeImg.src = href;

    largeImg.alt = title;

  }

consiste en 3 funciones una de ellas para realizar el paginado que para mostrar unos pocos productos de una vez, para mostrar las páginas que son y otra función para realizar el filtrado de los productos.

function renderProductos() {

for (const prod of mostrar) {

        const div = document.createElement('div');

        div.className = 'col-6 col-md-4 col-lg-3 mb-4';

        div.innerHTML = `

            <div class="card text-decoration-none">

                <img src="${prod.Imagen1}" class="card-img-top" alt="${prod.Nombre}">

                <div class="card-body">

                    <h5 class="card-title text-dark text-truncate">${prod.Nombre}</h5>

                    <p class="card-text fw-bold">${prod.Precio}€</p>

                    <a href="./producto.php?id=${prod.ID\_Producto}" class="btn btn-outline-primary btn-sm w-100">Ver Más</a>

                </div> </div>`;productosContainer.appendChild(div);}

function renderPaginacion() {

    paginacion.innerHTML = '';

    const totalPaginas = Math.ceil(productosFiltrados.length / porPagina);

    if (totalPaginas <= 1) return;

    for (let i = 1; i <= totalPaginas; i++) {

        const li = document.createElement('li');

        li.className = 'page-item' + (i === paginaActual ? ' active' : '');

        const a = document.createElement('a');

        a.className = 'page-link';

        a.href = '#';

        a.textContent = i;

        a.addEventListener('click', e => {

            e.preventDefault();

            paginaActual = i;

function aplicarFiltros() {

productosFiltrados = productos.filter(p => {

        const prodCategoria = (p.Categorias || '').toLowerCase();

        const prodTallas = (p.Tallas || []).map(t => t.toLowerCase());

        const prodColores = (p.Colores || []).map(c => c.toLowerCase());

        const prodNombre = (p.Nombre || '').toLowerCase();

        if (categoria && prodCategoria !== categoria) return false;

        if (talla && !prodTallas.includes(talla)) return false;

        if (color && !prodColores.includes(color)) return false;

        if (buscar && !prodNombre.includes(buscar)) return false;

        return true;

    });

    paginaActual = 1;

    renderProductos();

## listarCarrito.js

lista en contenido en carrito.php que cuando el usuario realiza la compra borra el local storage.

# CSS

function crearProductoHTML(id, producto) {

        const descuento = producto.descuento ? parseFloat(producto.descuento) : 0;

        const precioUnitario = parseFloat(producto.precio);

        const cantidad = producto.cantidad;

        const totalProducto = (precioUnitario - descuento) \* cantidad;

        totalProductos += precioUnitario \* cantidad;

        totalDescuento += descuento \* cantidad;

        return `

             listaCarrito.innerHTML = "";

    for (const id in carrito) {

        const producto = carrito[id];

        listaCarrito.innerHTML += crearProductoHTML(id, producto);

    }

const formPedido = document.getElementById('form-pedido');

    if(formPedido){

        formPedido.addEventListener('submit', () => {

            localStorage.removeItem('carrito');

        });

    }

## style.css

este es el estilo que tiene la página, esta parte esta dedicada al carrusel y a los card.

.carousel {

    height: 400px;

    overflow: hidden;}

.carousel-inner {

    height: 100%;}

.carousel-item img {

    object-fit: cover;

    object-position: center;

    height: 100%;

    width: 100%;}

.card {

    border: none;

    border-radius: 5px;

    overflow: hidden;

    height: 100%;}

.card-img-top {

    width: 100%;

    height: 100%;

    object-fit: cover;}

.card-body {

    font-weight: bold;

    text-align: center;

    padding: 5%;}

esta parte es para los cuando devuelven error.

.error{

    background-color: red;

    color: white;

    padding: 10px;

    width: 95%;

    border-radius: 5px;

    margin: 20px auto;}

.success{

    background-color: greenyellow;

    color: black;

    padding: 10px;

    width: 95%;

    border-radius: 5px;

    margin: 20px auto;

}

.contenedor{

    width: 75%;

    padding: 1rem 1rem;

    margin: 45px auto;

    color: white;

    background-color: black;

    border-radius: 15px;

    text-align: center;

    box-shadow: 0 20px 35px black;

}

Esta parte del estilo se utiliza para los formularios

form input{

    width: 100%;

    outline: none;

    padding: .7rem;

    background-color: white;

    border: none;

    border-radius: 5px;

}

label{

    color: blanchedalmond;

    font-size: 18px;

    padding: 10px;

    font-weight: 300;

}

h1{

    display: block;

    border: 2px solid black;

    width: 85%;

    padding: 10px;

    margin: 10px;

    border-radius: 10px;

}

h5{

    color: blanchedalmond;

}

input{

    display: block;

    border: 2px solid black;

    width: 100%;

    border-radius: 10px;

}

.button{

    font-size: 20px;

    font-weight: bold;

    text-decoration: none;

    padding: 10px;

    border-radius: 15px;

    outline: none;

    border: none;

    background-color: white;

    color: black;

    cursor: pointer;

}

.button:hover{

    background-color: blanchedalmond;

    color: black;

}

.a{

    float: left;

    margin-left: 10px;

    padding: .4rem;

    text-decoration: none;

    color: cyan;

}

.a:hover{

    color: blanchedalmond;

}