



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ



FACULTAD DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

LICENCIATURA EN INGENIERÍA DE SOFTWARE

FASE 4 - PROYECTO SEMESTRAL

INTEGRANTES:

CRISTOFER JIMENEZ 20-70-7100
FERNANDO LEZCANO 8-1013-1857
ALEJANDRA GOMEZ 20-14-7146
JEREMIAS APARICIO 6-725-1106

INGENIERIA WEB
ELBA VALDERRAMA

JUEVES 4 DE DICIEMBRE DEL 2025

Contenido

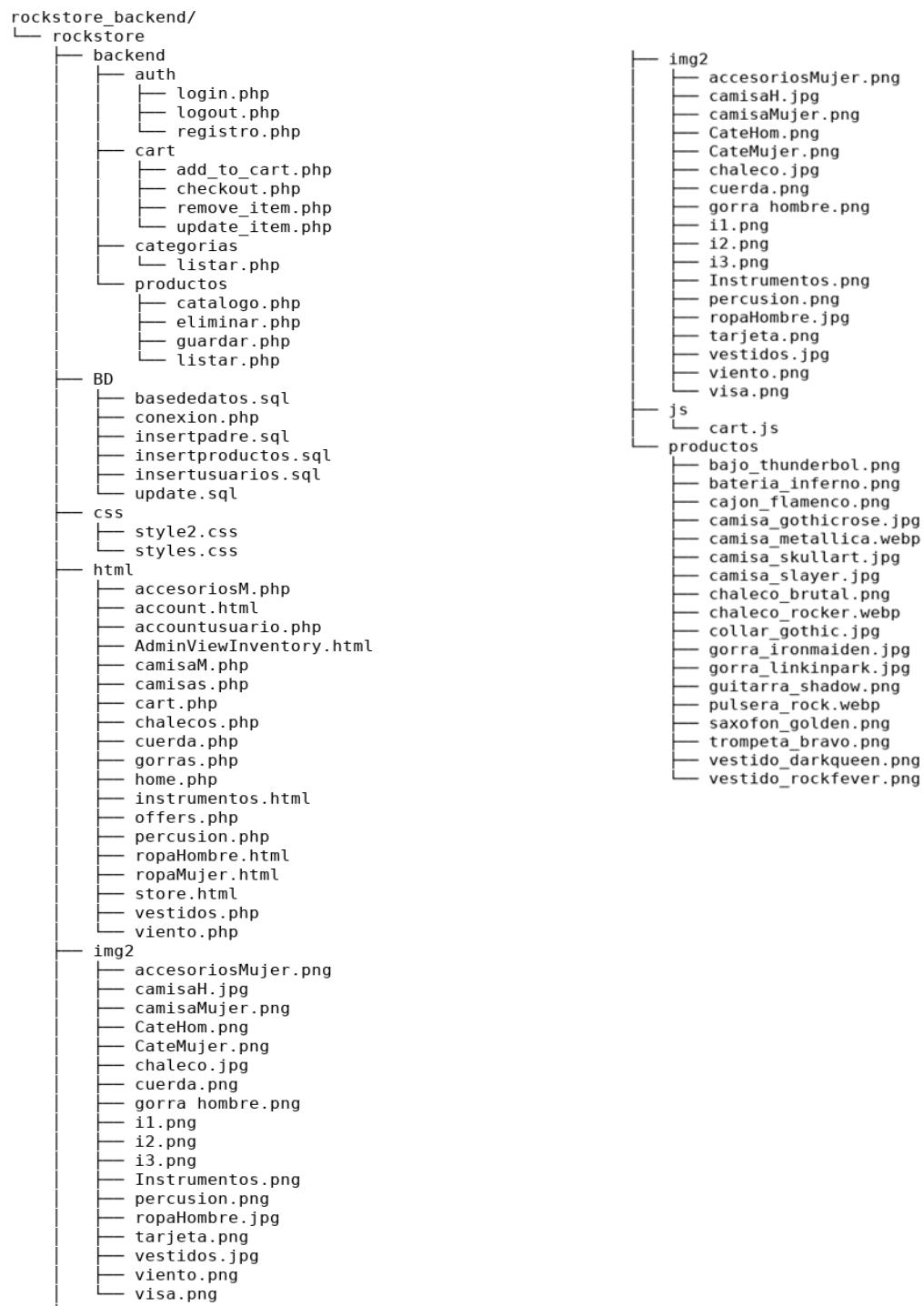
Documentación del proyecto Rockstore (Backend).....	4
1. Diagrama de carpetas del proyecto	4
2. Código fuente por carpetas	5
Carpeta: rockstore/BD.....	5
Archivo: de la BD	5
Archivo: conexion.php	14
Carpeta: rockstore/backend/auth.....	16
Archivo: login.php.....	16
Archivo: logout.php.....	18
Archivo: registro.php.....	19
Carpeta: rockstore/backend/cart.....	21
Archivo: add_to_cart.php.....	21
Archivo: checkout.php	23
Archivo: remove_item.php	26
Archivo: update_item.php.....	27
Carpeta: rockstore/backend/categorias.....	29
Archivo: listar.php	29
Carpeta: rockstore/backend/productos	31
Archivo: catalogo.php.....	31
Archivo: eliminar.php.....	33
Archivo: guardar.php.....	35
Archivo: listar.php	37
Carpeta: rockstore/html	39
Archivo: accesoriosM.php.....	39
Archivo: accountusuario.php	41
Archivo: camisaM.php	44
Archivo: camisas.php.....	46
Archivo: cart.php	48
Archivo: chalecos.php.....	51
Archivo: cuerda.php	53
Archivo: gorras.php.....	55
Archivo: home.php.....	57

Archivo: offers.php	60
Archivo: percusion.php.....	62
Archivo: vestidos.php.....	64
Archivo: viento.php	66
3.Documentacion en Github.....	68
4.Tabla colaborativa	69
5.Link del sitio web	70

Documentación del proyecto Rockstore (Backend)

1. Diagrama de carpetas del proyecto

La siguiente imagen muestra la estructura de carpetas y archivos del proyecto incluido en el archivo ZIP. Representa jerárquicamente cómo se organizan los módulos del backend, la base de datos, vistas y recursos.



2. Código fuente por carpetas

A continuación se listan todos los archivos PHP del proyecto, agrupados por su ruta dentro de la estructura de carpetas. Cada sección indica el archivo y su contenido.

Carpeta: rockstore/BD

Archivo: de la BD

-- Crear base de datos (opcional)

```
CREATE DATABASE IF NOT EXISTS rockstore
```

```
CHARACTER SET utf8mb4
```

```
COLLATE utf8mb4_general_ci;
```

```
USE rockstore;
```

-- 1) ROLES

```
CREATE TABLE roles (
```

```
    id      INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
```

```
    name    VARCHAR(50) NOT NULL UNIQUE,
```

```
    created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
```

```
    updated_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE  
    CURRENT_TIMESTAMP
```

```
) ENGINE=InnoDB
```

```
DEFAULT CHARSET=utf8mb4
```

```
COLLATE=utf8mb4_general_ci;
```

-- 2) USERS

```
CREATE TABLE users (
```

```
    id      INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
```

```
role_id      INT UNSIGNED NOT NULL,  
name        VARCHAR(150) NOT NULL,  
email       VARCHAR(150) NOT NULL UNIQUE,  
password_hash VARCHAR(255) NOT NULL,
```

-- CAMPOS PARA EL PERFIL

```
phone      VARCHAR(20) NULL,  
country    VARCHAR(100) NULL,  
address    VARCHAR(255) NULL,  
member_since DATE NULL,  
is_active   BOOLEAN DEFAULT TRUE,
```

```
created_at  DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
updated_at   DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP,
```

```
CONSTRAINT fk_users_roles  
FOREIGN KEY (role_id) REFERENCES roles(id)  
ON UPDATE CASCADE  
ON DELETE RESTRICT  
) ENGINE=InnoDB  
DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_general_ci;
```

```
CREATE INDEX idx_users_role_id ON users(role_id);
```

-- 3) CATEGORIES

```
CREATE TABLE categories (
    id      INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    name    VARCHAR(100) NOT NULL,
    slug    VARCHAR(100) NOT NULL UNIQUE,
    description VARCHAR(255),
    created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    updated_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
    CURRENT_TIMESTAMP
) ENGINE=InnoDB
DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;
```

-- 4) SUBCATEGORIES

```
CREATE TABLE subcategories (
    id      INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    category_id INT UNSIGNED NOT NULL,
    name    VARCHAR(100) NOT NULL,
    slug    VARCHAR(100) NOT NULL,
    description VARCHAR(255),
    created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    updated_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
    CURRENT_TIMESTAMP,
    CONSTRAINT fk_subcategories_categories
        FOREIGN KEY (category_id) REFERENCES categories(id)
        ON UPDATE CASCADE
        ON DELETE RESTRICT
```

```

) ENGINE=InnoDB
DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;

CREATE INDEX idx_subcategories_category_id ON subcategories(category_id);
CREATE INDEX idx_subcategories_slug ON subcategories(slug);

```

-- 5) PRODUCTS

```

CREATE TABLE products (
    id      INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    category_id  INT UNSIGNED NOT NULL,
    subcategory_id  INT UNSIGNED NOT NULL,
    code      VARCHAR(50) NOT NULL UNIQUE,
    name      VARCHAR(150) NOT NULL,
    short_description VARCHAR(255),
    long_description TEXT,
    price     DECIMAL(10,2) NOT NULL,
    stock     INT UNSIGNED NOT NULL DEFAULT 0,
    status     ENUM('en_stock','stock_bajo','agotado') NOT NULL DEFAULT 'en_stock',
    image_path  VARCHAR(255),
    has_offer   TINYINT(1) NOT NULL DEFAULT 0,
    offer_type   ENUM('porcentaje','precio_fijo') NULL,
    offer_value  DECIMAL(10,2) NULL,
    offer_start   DATETIME NULL,
    offer_end     DATETIME NULL,

```

```
    is_featured    TINYINT(1) NOT NULL DEFAULT 0,  
  
    created_at    DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    updated_at    DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE  
                  CURRENT_TIMESTAMP,
```

```
CONSTRAINT fk_products_categories  
FOREIGN KEY (category_id) REFERENCES categories(id)  
ON UPDATE CASCADE  
ON DELETE RESTRICT,  
CONSTRAINT fk_products_subcategories  
FOREIGN KEY (subcategory_id) REFERENCES subcategories(id)  
ON UPDATE CASCADE  
ON DELETE RESTRICT  
) ENGINE=InnoDB  
DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_general_ci;
```

```
CREATE INDEX idx_products_category_id ON products(category_id);  
CREATE INDEX idx_products_subcategory_id ON products(subcategory_id);  
CREATE INDEX idx_products_name ON products(name);  
CREATE INDEX idx_products_has_offer ON products(has_offer);
```

-- 6) USER_FAVORITES

```
CREATE TABLE userFavorites (
```

```

user_id  INT UNSIGNED NOT NULL,
product_id  INT UNSIGNED NOT NULL,
created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY (user_id, product_id),
CONSTRAINT fk_fav_user
    FOREIGN KEY (user_id) REFERENCES users(id)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
CONSTRAINT fk_fav_product
    FOREIGN KEY (product_id) REFERENCES products(id)
    ON UPDATE CASCADE
    ON DELETE CASCADE
) ENGINE=InnoDB
DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;

```

-- 7) CARTS

```

CREATE TABLE carts (
id      INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
user_id  INT UNSIGNED NOT NULL,
status   ENUM('active','converted','abandoned') NOT NULL DEFAULT 'active',
created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
updated_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
CONSTRAINT fk_carts_users
    FOREIGN KEY (user_id) REFERENCES users(id)

```

```
    ON UPDATE CASCADE  
    ON DELETE CASCADE  
) ENGINE=InnoDB  
DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_general_ci;
```

```
CREATE INDEX idx_carts_user_id ON carts(user_id);
```

```
-- 8) CART_ITEMS
```

```
CREATE TABLE cart_items (  
    id      INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    cart_id  INT UNSIGNED NOT NULL,  
    product_id  INT UNSIGNED NOT NULL,  
    quantity  INT UNSIGNED NOT NULL DEFAULT 1,  
    unit_price DECIMAL(10,2) NOT NULL,  
    created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    updated_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE  
    CURRENT_TIMESTAMP,  
    CONSTRAINT fk_cart_items_carts  
        FOREIGN KEY (cart_id) REFERENCES carts(id)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
    CONSTRAINT fk_cart_items_products  
        FOREIGN KEY (product_id) REFERENCES products(id)  
        ON UPDATE CASCADE  
        ON DELETE RESTRICT
```

```
) ENGINE=InnoDB  
DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_general_ci;  
  
CREATE INDEX idx_cart_items_cart_id ON cart_items(cart_id);  
CREATE INDEX idx_cart_items_product_id ON cart_items(product_id);
```

-- 9) ORDERS

```
CREATE TABLE orders (  
    id      INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    user_id  INT UNSIGNED NOT NULL,  
    cart_id   INT UNSIGNED NULL,  
    total_amount DECIMAL(10,2) NOT NULL,  
    shipping_amount DECIMAL(10,2) NOT NULL DEFAULT 0.00,  
    status     ENUM('pending','paid','shipped','cancelled') NOT NULL DEFAULT 'pending',  
    created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    updated_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE  
    CURRENT_TIMESTAMP,  
    CONSTRAINT fk_orders_users  
        FOREIGN KEY (user_id) REFERENCES users(id)  
        ON UPDATE CASCADE  
        ON DELETE RESTRICT,  
    CONSTRAINT fk_orders_carts  
        FOREIGN KEY (cart_id) REFERENCES carts(id)  
        ON UPDATE CASCADE  
        ON DELETE SET NULL
```

```

) ENGINE=InnoDB
DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;

CREATE INDEX idx_orders_user_id ON orders(user_id);

-- 10) ORDER_ITEMS
CREATE TABLE order_items (
    id      INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    order_id  INT UNSIGNED NOT NULL,
    product_id  INT UNSIGNED NOT NULL,
    quantity   INT UNSIGNED NOT NULL,
    unit_price  DECIMAL(10,2) NOT NULL,
    created_at  DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    updated_at  DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    CONSTRAINT fk_order_items_orders
        FOREIGN KEY (order_id) REFERENCES orders(id)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT fk_order_items_products
        FOREIGN KEY (product_id) REFERENCES products(id)
        ON UPDATE CASCADE
        ON DELETE RESTRICT
) ENGINE=InnoDB
DEFAULT CHARSET=utf8mb4

```

```
COLLATE=utf8mb4_general_ci;
```

```
CREATE INDEX idx_order_items_order_id ON order_items(order_id);
```

```
CREATE INDEX idx_order_items_product_id ON order_items(product_id);
```

Archivo: conexion.php

```
<?php  
// BD/conexion.php  
// Archivo de configuración y conexión a la base de datos MySQL usando PDO  
// Este archivo debe ser incluido en todos los scripts que necesiten acceso a la BD
```

// 1. CONFIGURACIÓN DE CREDENCIALES DE ACCESO

```
// Configuración de conexión a MySQL/MariaDB  
$db_host = 'localhost'; // Dirección del servidor de base de datos  
$db_name = 'rockstore'; // Nombre de la base de datos a la que conectarse  
$db_user = 'root'; // Usuario de la base de datos  
$db_pass = ""; // Contraseña del usuario (vacia en desarrollo local)
```

// 2. CONSTRUCCIÓN DEL DSN (Data Source Name)

```
// DSN: Cadena de conexión que PDO usa para identificar la base de datos  
// Formato: "mysql:host=HOST;dbname=NOMBRE_BD;charset=CODIFICACION"  
$dsn = "mysql:host=$db_host;dbname=$db_name;charset=utf8mb4";
```

```
// Explicación de charset=utf8mb4:  
// - utf8mb4 soporta caracteres Unicode completos (incluyendo emojis)  
// - Es superior a utf8 estándar que solo soporta BMP (Basic Multilingual Plane)  
// - Necesario para almacenar caracteres como emojis o ciertos ideogramas
```

// 3. OPCIONES DE CONFIGURACIÓN DE PDO

```
// Array de opciones para configurar el comportamiento de PDO  
$options = [  
    // ATTR_ERRMODE: Configura cómo PDO reporta errores  
    // PDO::ERRMODE_EXCEPTION: Lanza excepciones (PDOException) cuando ocurren
```

```
errores
// Esto permite usar try-catch para manejo elegante de errores
PDO::ATTR_ERRMODE      => PDO::ERRMODE_EXCEPTION,

// ATTR_DEFAULT_FETCH_MODE: Define el modo de recuperación por defecto
// PDO::FETCH_ASSOC: Devuelve arrays asociativos (nombre_columna => valor)
PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,

// ATTR_EMULATE_PREPARES: Controla si PDO emula prepared statements
// false: Usa prepared statements nativos del driver MySQL (más seguro)
// true: PDO emula los prepared statements (menos seguro, pero más compatible)
PDO::ATTR_EMULATE_PREPARES  => false,
];
```

// 4. ESTABLECIMIENTO DE LA CONEXIÓN

```
try {
    // Crea una nueva instancia de PDO (objeto de conexión a la base de datos)
    // Parámetros: DSN, usuario, contraseña, opciones de configuración
    $pdo = new PDO($dsn, $db_user, $db_pass, $options);

    // Si la conexión es exitosa, $pdo estará disponible para todas las consultas
    // Este objeto será reutilizado por todos los scripts que incluyan este archivo
} catch (PDOException $e) {
```

// 5. MANEJO DE ERRORES DE CONEXIÓN

```
// PDOException se lanza cuando hay problemas de conexión
// Posibles causas:
// 1. Servidor MySQL no está ejecutándose
// 2. Credenciales incorrectas (usuario/contraseña)
// 3. Base de datos no existe
// 4. Problemas de red o firewall

// IMPORTANTE: En producción, NO muestres el mensaje de error completo
// Esto podría exponer información sensible (credenciales, estructura de BD)
```

```
// Para desarrollo: Muestra el error completo para debugging  
die('Error de conexión')
```

Carpeta: rockstore/backend/auth

Archivo: login.php

```
<?php  
// backend/auth/login.php  
// Script para autenticación de usuarios mediante correo electrónico y contraseña  
  
header('Content-Type: application/json; charset=utf-8');  
  
// 1. INICIALIZACIÓN DE SESIÓN  
if (session_status() === PHP_SESSION_NONE) {  
    session_start();  
}  
  
require_once __DIR__ . '/../BD/conexion.php';  
  
// 3. FUNCIÓN AUXILIAR PARA RESPUESTAS  
function response_and_exit(bool $success, string $message, array $extra = []): void {  
    echo json_encode(array_merge([  
        'success' => $success,  
        'message' => $message  
    ], $extra), JSON_UNESCAPED_UNICODE);  
    exit;  
}  
  
// 2. VALIDACIÓN DEL MÉTODO HTTP  
if ($_SERVER['REQUEST_METHOD'] !== 'POST') {  
    http_response_code(405);  
    response_and_exit(false, 'Método no permitido. Usa POST.');
```

}

```
try {  
    // 4.1. RECEPCIÓN Y LIMPIEZA DE CREDENCIALES  
    $email = isset($_POST['email']) ? trim($_POST['email']) : '';  
    $password = isset($_POST['password']) ? $_POST['password'] : '';  
  
    // 4.2. VALIDACIÓN DE DATOS DE ENTRADA  
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {  
        response_and_exit(false, 'El correo electrónico no es válido.');
```

}

```

if ($password === '') {
    response_and_exit(false, 'La contraseña es obligatoria.');
}

// 4.3. BÚSQUEDA DEL USUARIO EN LA BASE DE DATOS
$sql = "
SELECT
    u.id,
    u.name,
    u.email,
    u.password_hash,
    r.name AS role_name
FROM users u
INNER JOIN roles r ON u.role_id = r.id
WHERE u.email = :email
LIMIT 1
";

$stmt = $pdo->prepare($sql);
$stmt->execute([':email' => $email]);
$user = $stmt->fetch(PDO::FETCH_ASSOC);

// 4.4. VERIFICACIÓN DE EXISTENCIA DEL USUARIO
if (!$user) {
    response_and_exit(false, 'Correo o contraseña incorrectos.');
}

// 4.5. VERIFICACIÓN DE CONTRASEÑA
if (!password_verify($password, $user['password_hash'])) {
    response_and_exit(false, 'Correo o contraseña incorrectos.');
}

// 4.6. CREACIÓN DE LA SESIÓN DE USUARIO
$_SESSION['user_id'] = (int)$user['id']; //
$_SESSION['user_name'] = $user['name'];
$_SESSION['user_email'] = $user['email'];
$_SESSION['user_role'] = $user['role_name'];

// 4.7. DETERMINACIÓN DE REDIRECCIÓN SEGÚN ROL
if ($user['role_name'] === 'admin') {
    $redirect = 'AdminViewInventory.html';
} else {
    $redirect = 'store.html';
}

```

```

    }

// 4.8. RESPUESTA EXITOSA
response_and_exit(true, 'Inicio de sesión exitoso.', [
    'role' => $user['role_name'],
    'redirect' => $redirect
]);

} catch (PDOException $e) {
    http_response_code(500);
    response_and_exit(false, 'Error en la base de datos: ' . $e->getMessage());
} catch (Exception $e) {
    http_response_code(500);
    response_and_exit(false, 'Error inesperado: ' . $e->getMessage());
}

```

Archivo: *logout.php*

```

<?php
// backend/auth/logout.php

if (session_status() === PHP_SESSION_NONE) {
    session_start();
}

// Eliminar todas las variables de sesión
$_SESSION = [];

// Destruir la sesión
if (ini_get('session.use_cookies')) {
    $params = session_get_cookie_params();
    setcookie(session_name(), '', time() - 42000,
        $params['path'], $params['domain'],
        $params['secure'], $params['httponly'])
};

}

session_destroy();

// Si viene por fetch, no pasa nada que devolvamos algo sencillo
header('Content-Type: application/json; charset=utf-8');
echo json_encode([
    'ok' => true,
    'msg' => 'Sesión cerrada correctamente.'
]
);

```

```
], JSON_UNESCAPED_UNICODE);
```

Archivo: registro.php

```
<?php
// backend/auth/registro.php
// Script para registrar nuevos usuarios en el sistema

// Establece el tipo de contenido como JSON con codificación UTF-8
header('Content-Type: application/json; charset=utf-8');

// Incluye el archivo de conexión a la base de datos
require_once __DIR__ . '/../BD/conexion.php';

// VALIDACIÓN DEL MÉTODO HTTP
if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
    http_response_code(405);
    echo json_encode([
        'success' => false,
        'message' => 'Método no permitido. Usa POST.'
    ], JSON_UNESCAPED_UNICODE);
    exit;
}

/**
 * Función que envía una respuesta JSON y termina la ejecución
 */
function response_and_exit(bool $success, string $message, array $extra = []): void {
    echo json_encode(array_merge([
        'success' => $success,
        'message' => $message
    ], $extra), JSON_UNESCAPED_UNICODE);
    exit;
}

// LÓGICA PRINCIPAL DEL REGISTRO
try {
    // RECEPCIÓN Y LIMPIEZA DE DATOS DEL FORMULARIO
    $name  = isset($_POST['name']) ? trim($_POST['name']) : '';
    $email = isset($_POST['email']) ? trim($_POST['email']) : '';
    $password = isset($_POST['password']) ? $_POST['password'] : '';

    // NUEVOS CAMPOS - con valores por defecto
    $phone = isset($_POST['phone']) ? trim($_POST['phone']) : '';
```

```

$country = isset($_POST['country']) ? trim($_POST['country']) : '';
$address = isset($_POST['address']) ? trim($_POST['address']) : '';

// VALIDACIÓN DE DATOS DE ENTRADA
$errors = [];

if ($name === '') {
    $errors[] = 'El nombre es obligatorio.';
}

if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    $errors[] = 'El correo electrónico no es válido.';
}

if (strlen($password) < 8) {
    $errors[] = 'La contraseña debe tener al menos 8 caracteres.';
}

// NUEVA VALIDACIÓN: País es obligatorio
if ($country === '') {
    $errors[] = 'El país es obligatorio.';
}

// Si hay errores de validación, se devuelven al frontend
if (!empty($errors)) {
    response_and_exit(false, 'Errores de validación.', ['errors' => $errors]);
}

// VERIFICACIÓN DE EMAIL EXISTENTE
$sqlCheck = "SELECT id FROM users WHERE email = :email LIMIT 1";
$stmtCheck = $pdo->prepare($sqlCheck);
$stmtCheck->execute([':email' => $email]);

if ($stmtCheck->fetch()) {
    response_and_exit(false, 'Ya existe una cuenta registrada con ese correo.');
}

// OBTENCIÓN DEL ROL "CLIENTE"
$sqlRole = "SELECT id FROM roles WHERE name = 'cliente' LIMIT 1";
$stmtRole = $pdo->prepare($sqlRole);
$stmtRole->execute();
$role = $stmtRole->fetch();

```

```

if (!$role) {
    response_and_exit(false, 'No se encontró el rol de cliente en la base de datos.');
}

roleId = (int)$role['id'];

// HASH DE CONTRASEÑA
$passwordHash = password_hash($password, PASSWORD_DEFAULT);

// INSERCIÓN DEL NUEVO USUARIO - CON CAMPOS ACTUALIZADOS
$sqlInsert = "
    INSERT INTO users (

```

Carpeta: rockstore/backend/cart

Archivo: add_to_cart.php

```

<?php
// backend/cart/add_to_cart.php
header('Content-Type: application/json; charset=utf-8');

if (session_status() === PHP_SESSION_NONE) {
    session_start();
}

require_once __DIR__ . '/../BD/conexion.php'; // aquí tienes $pdo

function json_response($ok, $msg, $extra = [])
{
    echo json_encode(array_merge([
        'ok' => $ok,
        'msg' => $msg,
    ], $extra), JSON_UNESCAPED_UNICODE);
    exit;
}

// 1) Validar método
if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
    http_response_code(405);
    json_response(false, 'Método no permitido');
}

// 2) Verificar login
if (!isset($_SESSION['user_id'])) {

```

```

    http_response_code(401);
    json_response(false, 'Debes iniciar sesión para agregar productos al carrito.');
}

$userId = (int)$_SESSION['user_id'];
$productId = isset($_POST['product_id']) ? (int)$_POST['product_id'] : 0;
$quantity = isset($_POST['quantity']) ? (int)$_POST['quantity'] : 1;

if ($productId <= 0 || $quantity <= 0) {
    http_response_code(400);
    json_response(false, 'Datos inválidos.');
}

try {
    // 3) Obtener producto y calcular precio final (con oferta si aplica)
    $sqlProd = "
        SELECT price, has_offer, offer_type, offer_value, offer_start, offer_end
        FROM products
        WHERE id = :id
        LIMIT 1
    ";
    $stmtProd = $pdo->prepare($sqlProd);
    $stmtProd->execute([':id' => $productId]);
    $prod = $stmtProd->fetch(PDO::FETCH_ASSOC);

    if (!$prod) {
        http_response_code(404);
        json_response(false, 'Producto no encontrado.');
    }

    $basePrice = (float)$prod['price'];
    $unitPrice = $basePrice;

    if ((int)$prod['has_offer'] === 1 && !empty($prod['offer_type']) && $prod['offer_value'] !== null) {
        $ahora = new DateTimeImmutable();
        $enRango = true;

        if (!empty($prod['offer_start'])) {
            $inicio = new DateTimeImmutable($prod['offer_start']);
            if ($ahora < $inicio) $enRango = false;
        }
        if (!empty($prod['offer_end'])) {
    
```

```

$fin = new DateTimeImmutable($prod['offer_end']);
if ($ahora > $fin) $enRango = false;
}

if ($enRango) {
    $offerValue = (float)$prod['offer_value'];

    if ($prod['offer_type'] === 'porcentaje') {
        $unitPrice = max(0, $basePrice - ($basePrice * ($offerValue / 100)));
    } elseif ($prod['offer_type'] === 'precio_fijo') {
        $unitPrice = max(0, $offerValue);
    }
}
}

// 4) Buscar o crear carrito activo
$sqlCart = "
    SELECT id FROM carts
    WHERE user_id = :uid AND status = 'active'
    LIMIT 1
";
$stmtCart = $pdo->prepare($sqlCart);
$stmtCart->execute([':uid' => $userId]);
$cart = $stmtCart->fetch(PDO::FETCH_ASSOC);

if ($cart) {
    $cartId = (int)$cart['id'];
} else {
    $sqlNewCart = "INSERT INTO carts (user_id, status) VALUES (:uid, 'active')";
    $stmtNew = $pdo->prepare($sqlNewCart);
    $stmtNew->e
}

```

Archivo: *checkout.php*

```

<?php
// backend/cart/checkout.php
header('Content-Type: application/json; charset=utf-8');

if (session_status() === PHP_SESSION_NONE) {
    session_start();
}

require_once __DIR__ . '/../BD/conexion.php';

function json_response($ok, $msg, $extra = []) {

```

```

echo json_encode(array_merge(['ok' => $ok, 'msg' => $msg], $extra),
JSON_UNESCAPED_UNICODE);
exit;
}

if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
    http_response_code(405);
    json_response(false, 'Método no permitido.');
}

if (!isset($_SESSION['user_id'])) {
    http_response_code(401);
    json_response(false, 'Debes iniciar sesión para comprar.');
}

$userId = (int)$_SESSION['user_id'];

// Campos de tarjeta (solo se valida que no estén vacíos)
$cardName = trim($_POST['card_name'] ?? '');
$cardNumber = trim($_POST['card_number'] ?? '');
$cardExp = trim($_POST['card_exp'] ?? '');
$cardCvv = trim($_POST['card_cvv'] ?? '');

if (!$cardName || !$cardNumber || !$cardExp || !$cardCvv) {
    http_response_code(400);
    json_response(false, 'Por favor completa todos los datos de la tarjeta.');
}

try {
    // 1) Obtener carrito activo
    $sqlCart = "
        SELECT id
        FROM carts
        WHERE user_id = :uid
        AND status = 'active'
        LIMIT 1
    ";
    $stmtCart = $pdo->prepare($sqlCart);
    $stmtCart->execute([':uid' => $userId]);
    $cart = $stmtCart->fetch(PDO::FETCH_ASSOC);

    if (!$cart) {
        json_response(false, 'No hay carrito activo para este usuario.');
    }
}

```

```

}

$cartId = (int)$cart['id'];

// 2) Obtener items del carrito
$sqlItems =
    SELECT ci.product_id, ci.quantity, ci.unit_price,
           p.stock
    FROM cart_items ci
   INNER JOIN products p ON ci.product_id = p.id
  WHERE ci.cart_id = :cid
";
$stmtItems = $pdo->prepare($sqlItems);
$stmtItems->execute([':cid' => $cartId]);
$items = $stmtItems->fetchAll(PDO::FETCH_ASSOC);

if (empty($items)) {
    json_response(false, 'Tu carrito está vacío.');
}

// 3) Verificar stock y calcular montos
$subtotal = 0;
foreach ($items as $it) {
    if ($it['quantity'] > $it['stock']) {
        json_response(false, 'No hay stock suficiente para uno de los productos.');
    }
    $subtotal += $it['quantity'] * $it['unit_price'];
}

$shipping = $subtotal > 0 ? 20 : 0;
$total   = $subtotal + $shipping;

// 4) Crear orden + order_items en una transacción
$pdo->beginTransaction();

$sqlOrder =
    INSERT INTO orders (user_id, cart_id, total_amount, shipping_amount, status)
      VALUES (:uid, :cid, :total, :shipping, 'paid')
";
$stmtOrder = $pdo->prepare($sqlOrder);
$stmtOrder->execute([
    ':uid'    => $userId,
    ':cid'    => $cartId,
])

```

```

':total' => $total,
':shipping' => $shipping
]);
$orderId = (int)$pdo->lastInsertId();

$sqlOrderItem =
    INSERT INTO order_items (order_id, product_id, quantity, unit_p

```

Archivo: remove_item.php

```

<?php
// backend/cart/remove_item.php
header('Content-Type: application/json; charset=utf-8');

if (session_status() === PHP_SESSION_NONE) {
    session_start();
}

require_once __DIR__ . '/../BD/conexion.php';

function json_response($ok, $msg, $extra = []) {
    echo json_encode(array_merge(['ok' => $ok, 'msg' => $msg], $extra),
JSON_UNESCAPED_UNICODE);
    exit;
}

if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
    http_response_code(405);
    json_response(false, 'Método no permitido.');
}

if (!isset($_SESSION['user_id'])) {
    http_response_code(401);
    json_response(false, 'Debes iniciar sesión.');
}

$userId = (int)$_SESSION['user_id'];
$itemId = isset($_POST['item_id']) ? (int)$_POST['item_id'] : 0;

if ($itemId <= 0) {
    http_response_code(400);
    json_response(false, 'Datos inválidos.');
}

```

```

try {
    // Verificar que el item pertenece al carrito activo del usuario
    $sql = "
        SELECT ci.id
        FROM cart_items ci
        INNER JOIN carts c ON ci.cart_id = c.id
        WHERE ci.id = :iid
        AND c.user_id = :uid
        AND c.status = 'active'
        LIMIT 1
    ";
    $stmt = $pdo->prepare($sql);
    $stmt->execute([
        ':iid' => $itemId,
        ':uid' => $userId
    ]);
    $item = $stmt->fetch(PDO::FETCH_ASSOC);

    if (!$item) {
        http_response_code(404);
        json_response(false, 'Item no encontrado en tu carrito.');
    }

    $sqlDel = "DELETE FROM cart_items WHERE id = :iid";
    $stmtDel = $pdo->prepare($sqlDel);
    $stmtDel->execute([':iid' => $itemId]);

    json_response(true, 'Producto eliminado del carrito.');

} catch (PDOException $e) {
    http_response_code(500);
    json_response(false, 'Error en la base de datos: ' . $e->getMessage());
}

```

Archivo: update_item.php

```

<?php
// backend/cart/update_item.php
header('Content-Type: application/json; charset=utf-8');

if (session_status() === PHP_SESSION_NONE) {
    session_start();
}

```

```

require_once __DIR__ . '/../BD/conexion.php';

function json_response($ok, $msg, $extra = []) {
    echo json_encode(array_merge(['ok' => $ok, 'msg' => $msg], $extra),
        JSON_UNESCAPED_UNICODE);
    exit;
}

if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
    http_response_code(405);
    json_response(false, 'Método no permitido.');
}

if (!isset($_SESSION['user_id'])) {
    http_response_code(401);
    json_response(false, 'Debes iniciar sesión.');
}

$userId = (int)$_SESSION['user_id'];
$itemId = isset($_POST['item_id']) ? (int)$_POST['item_id'] : 0;
$quantity = isset($_POST['quantity']) ? (int)$_POST['quantity'] : 0;

if ($itemId <= 0 || $quantity <= 0) {
    http_response_code(400);
    json_response(false, 'Datos inválidos.');
}

try {
    // Verificar que el ítem pertenece a un carrito activo del usuario
    $sql = "
        SELECT ci.id, ci.cart_id, ci.product_id, ci.quantity,
               p.stock
        FROM cart_items ci
        INNER JOIN carts c ON ci.cart_id = c.id
        INNER JOIN products p ON ci.product_id = p.id
        WHERE ci.id = :iid
        AND c.user_id = :uid
        AND c.status = 'active'
        LIMIT 1
    ";
    $stmt = $pdo->prepare($sql);
    $stmt->execute([
        ':iid' => $itemId,

```

```

':uid' => $userId
]);
$item = $stmt->fetch(PDO::FETCH_ASSOC);

if (!$item) {
    http_response_code(404);
    json_response(false, 'Item no encontrado en tu carrito.');
}

$stock = (int)$item['stock'];
if ($quantity > $stock) {
    json_response(false, 'No hay suficiente stock disponible.');
}

$sqlUpd = "
UPDATE cart_items
SET quantity = :qty, updated_at = NOW()
WHERE id = :iid
";
$stmtUpd = $pdo->prepare($sqlUpd);
$stmtUpd->execute([
    ':qty' => $quantity,
    ':iid' => $itemId
]);

json_response(true, 'Cantidad actualizada.');

} catch (PDOException $e) {
    http_response_code(500);
    json_response(false, 'Error en la base de datos: ' . $e->getMessage());
}

```

Carpeta: rockstore/backend/categorias

Archivo: *listar.php*

```

<?php
// backend/categorias/listar.php

// Establece el tipo de contenido como JSON con codificación UTF-8
// para asegurar que los caracteres especiales se muestren correctamente
header('Content-Type: application/json; charset=utf-8');

// Incluye el archivo de conexión a la base de datos

```

```

// __DIR__ garantiza que la ruta sea absoluta y relativa al directorio actual
require_once __DIR__ . '/../BD/conexion.php';

// Bloque try-catch para manejar posibles excepciones
try {
    // Consulta SQL para obtener subcategorías con información de sus categorías
    // JOIN para relacionar subcategorías con sus categorías correspondientes
    $sql = "
        SELECT
            sc.id, -- ID de la subcategoría
            sc.name AS subcategory_name, -- Nombre de la subcategoría
            sc.slug AS subcategory_slug, -- Slug/URL amigable de la subcategoría
            c.id AS category_id, -- ID de la categoría principal
            c.name AS category_name, -- Nombre de la categoría principal
            c.slug AS category_slug -- Slug/URL amigable de la categoría
        FROM subcategories sc -- Tabla de subcategorías (alias 'sc')
        INNER JOIN categories c ON sc.category_id = c.id -- Relación con categorías
        ORDER BY c.id, sc.id -- Ordena primero por categoría, luego por subcategoría
    ";
    // Ejecuta la consulta directamente (sin parámetros)
    $stmt = $pdo->query($sql);

    // Obtiene todos los resultados como array asociativo
    $rows = $stmt->fetchAll();

    // Convierte los resultados a JSON y los imprime
    // JSON_UNESCAPED_UNICODE: preserva caracteres Unicode sin escaparlos
    echo json_encode($rows, JSON_UNESCAPED_UNICODE);

} catch (PDOException $e) {
    // En caso de error en la base de datos:

    // Establece código de respuesta HTTP 500 (Error interno del servidor)
    http_response_code(500);

    // Devuelve un objeto JSON con información del error
    echo json_encode([
        'success' => false, // Indica que la operación falló
        'message' => 'Error al obtener subcategorías: ' . $e->getMessage() // Mensaje de error
    ], JSON_UNESCAPED_UNICODE);
}

```

Carpeta: rockstore/backend/productos

Archivo: catalogo.php

```
<?php
// backend/productos/catalogo.php
// Reutilizable para TODAS las páginas de catálogo

require_once __DIR__ . '/../..../BD/conexion.php'; // Ajusta si tu conexión está en otra ruta

/**
 * Obtiene productos desde la BD según filtros opcionales.
 *
 * Opciones soportadas:
 * - 'subcategory_slug' => string (ej: 'chalecos', 'viento', 'cuerda', etc.)
 * - 'only_offers'      => bool  (true = solo productos con oferta)
 *
 * @param array $options
 * @return array
 */
function obtenerProductos(array $options = []): array
{
    global $pdo; // conexión desde BD/conexion.php

    // Siempre empezamos con una condición verdadera
    $where = ["1 = 1"];
    $params = [];

    // 🔹 Filtrar por subcategoría (slug), ej: 'chalecos', 'viento', 'cuerda'
    if (!empty($options['subcategory_slug'])) {
        $where[] = "s.slug = :subcategory_slug";
        $params[':subcategory_slug'] = $options['subcategory_slug'];
    }

    // 🔹 Solo productos con oferta (NO filtra por fechas, solo por has_offer = 1)
    if (!empty($options['only_offers'])) {
        $where[] = "p.has_offer = 1";
    }

    // 🔹 Solo productos destacados (para "Lo más vendido" del home)
    if (!empty($options['only_featured'])) {
        $where[] = "p.is_featured = 1";
    }
}
```

```

// Consulta principal
$sql = "
SELECT
    p.id,
    p.code,
    p.name,
    p.short_description,
    p.price,          -- columna REAL de tu tabla
    p.stock,
    p.status,
    p.image_path,
    p.has_offer,
    p.offer_type,
    p.offer_value,
    p.offer_start,
    p.offer_end,
    p.is_featured,
    s.name AS subcategory_name,
    s.slug AS subcategory_slug,
    c.name AS category_name,
    c.slug AS category_slug
FROM products p
INNER JOIN subcategories s ON p.subcategory_id = s.id
INNER JOIN categories c  ON s.category_id  = c.id
WHERE " . implode(' AND ', $where) . "
ORDER BY p.name ASC
";

$stmt = $pdo->prepare($sql);
$stmt->execute($params);

return $stmt->fetchAll(PDO::FETCH_ASSOC);
}

/**
 * Renderiza el GRID de productos con el HTML de las cards.
 * Usa las mismas clases que ya tienes en tus .html:
 * - .products-grid
 * - .product-card
 * - .favorite-btn
 * - .add-to-cart-btn
 */
function renderProductosGrid(array $options = []): void

```

```

{
    $productos = obtenerProductos($options);

    if (empty($productos)) {
        echo '<p class="text-muted">No hay productos disponibles en esta sección.</p>';
        return;
    }

    foreach ($productos as $p) {
        $precioOriginal = (float) $p['price'];
        $precioFinal = $precioOriginal;
        $tieneOferta = (int)$p['has_offer'] === 1;

        // Calcular precio final según tipo de oferta
        if ($tieneOferta && !empty($p['offer_type']) && $p['of

```

Archivo: eliminar.php

```

<?php
// backend/productos/eliminar.php
// Script para eliminar productos de la base de datos

// Establece el tipo de contenido como JSON con codificación UTF-8
// Esto asegura que el cliente reciba una respuesta bien formateada
header('Content-Type: application/json; charset=utf-8');

// Incluye el archivo de conexión a la base de datos
// __DIR__ garantiza una ruta absoluta desde el directorio actual
require_once __DIR__ . '/../BD/conexion.php';

// VALIDACIÓN DEL MÉTODO HTTP

// Este script solo debe ser accedido mediante POST por seguridad
// GET podría permitir eliminaciones accidentales mediante enlaces
if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
    // Código 405: Método no permitido
    http_response_code(405);
    echo json_encode([
        'success' => false,
        'message' => 'Método no permitido. Usa POST.'
    ], JSON_UNESCAPED_UNICODE);
    exit; // Termina la ejecución del script
}

```

```
}
```

```
// FUNCIÓN AUXILIAR PARA RESPUESTAS
```

```
/**  
 * Función helper que envía una respuesta JSON y termina la ejecución  
 *  
 * @param bool $success Indica si la operación fue exitosa  
 * @param string $message Mensaje descriptivo para el cliente  
 * @param array $extra Datos adicionales a incluir en la respuesta  
 */  
function response_and_exit($success, $message, $extra = [])  
{  
    // Combina el array base con los datos adicionales  
    echo json_encode(array_merge([  
        'success' => $success,  
        'message' => $message  
    ], $extra), JSON_UNESCAPED_UNICODE);  
    exit; // Importante: termina la ejecución después de enviar la respuesta  
}
```

```
// LÓGICA PRINCIPAL
```

```
try {
```

```
    // VALIDACIÓN DEL ID DEL PRODUCTO
```

```
    // Obtiene el ID del producto desde los datos POST  
    // (int) convierte el valor a entero para seguridad  
    $id = isset($_POST['id']) ? (int)$_POST['id'] : 0;  
  
    // Verifica que el ID sea un número positivo  
    // Esto previene IDs inválidos como 0, -1, o no numéricos  
    if ($id <= 0) {  
        response_and_exit(false, 'ID de producto inválido.');
```

```

// VERIFICACIÓN DE EXISTENCIA (OPTIONAL CHECK)

// Consulta preparada para verificar que el producto existe
// Esto evita mensajes de error confusos si el producto ya fue eliminado
$sqlCheck = "SELECT id FROM products WHERE id = :id";
$stmtCheck = $pdo->prepare($sqlCheck); // Prepara la consulta
$stmtCheck->execute([':id' => $id]); // Ejecuta con parámetro seguro
$exists = $stmtCheck->fetch(); // Obtiene el resultado

// Si no existe, informa al cliente sin intentar eliminarlo
if (!$exists) {
    response_and_exit(false, 'El producto no existe o ya fue eliminado.');
}

```

// ELIMINACIÓN DEL PRODUCTO

```

// NOTA: Esta es una eliminación permanente (HARD DELETE)
$sql = "DELETE FROM products WHERE id = :id";
$stmt = $pdo->prepare($sql); // Prepara la consulta
$stmt->execute([':id' => $id]); // Ejecuta con parámetro seguro

// Si llegamos aquí, la eliminación fue exitosa

```

Archivo: guardar.php

```

<?php
// backend/productos/guardar.php
// Script para crear o actualizar productos en la base de datos

// Establece el tipo de contenido como JSON con codificación UTF-8
// Esto asegura que el cliente (frontend) reciba una respuesta bien formateada
header('Content-Type: application/json; charset=utf-8');

// Incluye el archivo de conexión a la base de datos
// __DIR__ garantiza una ruta absoluta desde el directorio actual del script
require_once __DIR__ . '/../BD/conexion.php';

```

// VALIDACIÓN DEL MÉTODO HTTP

```

// Este script solo debe ser accedido mediante POST
// Los métodos GET podrían exponer datos sensibles o causar acciones no deseadas
if ($_SERVER['REQUEST_METHOD'] !== 'POST') {

```

```

// Código 405: Método no permitido
http_response_code(405);
echo json_encode([
    'success' => false,
    'message' => 'Método no permitido. Usa POST.'
], JSON_UNESCAPED_UNICODE);
exit; // Termina la ejecución del script inmediatamente
}

// FUNCIÓN AUXILIAR PARA RESPUESTAS

/**
 * Función helper que envía una respuesta JSON y termina la ejecución
 *
 * @param bool $success Indica si la operación fue exitosa (true/false)
 * @param string $message Mensaje descriptivo para el cliente
 * @param array $extra Datos adicionales a incluir en la respuesta (opcional)
 */
function response_and_exit($success, $message, $extra = [])
{
    // Combina el array base con los datos adicionales usando array_merge
    // Esto permite extender la respuesta con datos específicos
    echo json_encode(array_merge([
        'success' => $success,
        'message' => $message
    ], $extra), JSON_UNESCAPED_UNICODE);
    exit; // Importante: termina la ejecución después de enviar la respuesta
}

// LÓGICA PRINCIPAL

try {
    // 1. RECEPCIÓN Y LIMPIEZA DE DATOS DEL FORMULARIO

    // ID del producto: si viene, es una actualización; si no, es una creación
    // (int) convierte el valor a entero para seguridad
    $idProducto = isset($_POST['id']) ? (int)$_POST['id'] : null;

    // trim() elimina espacios en blanco al inicio y final
    $codigo = isset($_POST['codigo']) ? trim($_POST['codigo']) : '';
    $nombre = isset($_POST['producto']) ? trim($_POST['producto']) : '';
}

```

```

// IMPORTANTE: Se recibe el slug de la subcategoría, no su ID
// Esto es más amigable y evita problemas con valores numéricos
$slugSubcat = isset($_POST['subcategoria']) ? trim($_POST['subcategoria']) : "";

// Conversión a tipos numéricos apropiados
$precio    = isset($_POST['precio']) ? (float)$_POST['precio'] : 0;
$stock     = isset($_POST['stock']) ? (int)$_POST['stock'] : 0;
$estadoLabel = isset($_POST['estado']) ? trim($_POST['estado']) : "";

// DATOS DE OFERTA (CAMPOS OPCIONALES)

// Verifica si el producto tiene oferta activa
$tieneOferta = isset($_POST['tieneOferta']) && $_POST['tieneOferta'] === '1';

// Tipo de oferta: 'porcentaje' o 'precio_fijo'
$tipoOferta = isset($_POST[

```

Archivo: listar.php

```

<?php
// backend/productos/listar.php
// Script para obtener y listar todos los productos de la base de datos

// Establece el tipo de contenido como JSON con codificación UTF-8
// Esto asegura que el cliente reciba una respuesta en formato JSON con caracteres
especiales correctamente codificados
header('Content-Type: application/json; charset=utf-8');

// Incluye el archivo de conexión a la base de datos
// __DIR__ proporciona la ruta absoluta del directorio actual del script
require_once __DIR__ . '/../BD/conexion.php';

```

// BLOQUE PRINCIPAL DE EJECUCIÓN

```

try {
    // 1. CONSULTA SQL PARA OBTENER PRODUCTOS

```

```

    // Consulta que obtiene productos junto con información de su subcategoría
    // Usa alias (AS) para nombres más descriptivos en los resultados
    $sql = "

```

```

SELECT
    p.id,          -- ID único del producto
    p.code AS codigo, -- Código del producto (alias para claridad)
    p.name AS producto, -- Nombre del producto
    sc.name AS subcategoria_nombre, -- Nombre de la subcategoría
    sc.slug AS subcategoria_slug, -- Slug/URL amigable de la subcategoría
    p.price AS precio, -- Precio del producto
    p.stock,        -- Cantidad en inventario
    p.status,       -- Estado: 'en_stock', 'stock_bajo', 'agotado'
    p.image_path,   -- Ruta de la imagen del producto
    p.has_offer,    -- Indica si tiene oferta (1/0)
    p.offer_type,   -- Tipo de oferta: 'porcentaje' o 'precio_fijo'
    p.offer_value   -- Valor de la oferta
FROM products p -- Tabla principal de productos (alias 'p')
INNER JOIN subcategories sc ON p.subcategory_id = sc.id -- Une con subcategorías
ORDER BY p.id ASC -- Ordena por ID ascendente (del más antiguo al más nuevo)
";

```

// 2. EJECUCIÓN DE LA CONSULTA

```

// Ejecuta la consulta SQL directamente (sin parámetros variables)
$stmt = $pdo->query($sql);

// Obtiene todos los resultados como array asociativo
// Cada fila es un array con los nombres de columna como claves
$rows = $stmt->fetchAll();

```

// 3. PROCESAMIENTO Y TRANSFORMACIÓN DE DATOS

```

$products = []; // Array que contendrá los productos procesados

// Itera sobre cada fila de resultados para transformar los datos
foreach ($rows as $row) {
    // Construye un array con la estructura esperada por el frontend
    $products[] = [
        'id'      => (int)$row['id'],      // Convierte a entero
        'codigo'  => $row['codigo'],      // Código del producto
        'producto' => $row['producto'],    // Nombre del producto
        // NOTA IMPORTANTE: En el frontend (JS) se llama "categoria",

```

```
// pero en realidad es el nombre de la SUBCATEGORÍA  
// Esto podría causar confusión - considera renombr
```

Carpeta: rockstore/html

Archivo: accesoriosM.php

```
<!DOCTYPE html>  
<html lang="es">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>RockStore - Página</title>  
  
    <!-- Bootstrap CSS -->  
    <link  
        href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"  
        rel="stylesheet">  
  
    <!-- Font Awesome CDN -->  
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@fortawesome/fontawesome-free@6.4.0/css/all.min.css">  
  
    <!-- Tu CSS -->  
    <link rel="stylesheet" href="../css/styles.css">  
</head>  
  
<body>  
  
    <!-- ● METANAVEGACIÓN -->  
    <div class="top-meta">  
        <a href="account.html">Iniciar sesión</a>  
        <a href="account.html">Registrarse</a>  
        <input type="text" placeholder="Buscar...">  
    </div>  
  
    <!-- ● NAVBAR PRINCIPAL -->  
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark px-3">  
        <a class="navbar-brand" href="../html/home.php">🎸 RockStore</a>  
  
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#menu">  
            <span class="navbar-toggler-icon"></span>  
        </button>
```

```

<div class="collapse navbar-collapse" id="menu">
  <ul class="navbar-nav ms-3">

    <li class="nav-item"><a class="nav-link" href="../html/home.php">Home</a></li>

    <!-- ☐ Dropdown con hover -->
    <li class="nav-item dropdown hover-dropdown">
      <a class="nav-link dropdown-toggle" href="../html/store.html" id="storeMenu" role="button">
        Tienda
      </a>

      <ul class="dropdown-menu" aria-labelledby="storeMenu">
        <li><a class="dropdown-item" href="../html/instrumentos.html">Instrumentos</a></li>
        <li><a class="dropdown-item" href="../html/ropaMujer.html">Ropa Mujer</a></li>
        <li><a class="dropdown-item" href="../html/ropaHombre.html">Ropa Hombre</a></li>
      </ul>
    </li>

    <li class="nav-item"><a class="nav-link" href="offers.php">Ofertas</a></li>
  </ul>

  <!-- Enlaces Cart y Account alineados a la derecha -->
  <ul class="navbar-nav ms-auto">
    <li class="nav-item">
      <a class="nav-link" href="cart.html">
        <i class="fas fa-shopping-cart"></i> Carrito
      </a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="account.html">
        <i class="fas fa-user"></i> Cuenta
      </a>
    </li>
  </ul>
</div>
</nav>

```

```

<body>

    <!-- CATÁLOGO DE PRODUCTOS -->
    <div class="catalog-section">
        <h6 class="section-title">Lo más popular hasta ahora</h6>

        <div class="products-grid">
            <!-- Pulsera Rock Rebel -->

            <?php
                // Incluir el catálogo reutilizable
                require_once __DIR__ . '/../backend/productos/catalogo.php';

                // Cargar productos de

```

Archivo: accountusuario.php

```

<?php
if (session_status() === PHP_SESSION_NONE) {
    session_start();
}

require_once __DIR__ . '/../BD/conexion.php';

// Si no hay usuario logueado, lo mandamos al login
if (!isset($_SESSION['user_id'])) {
    header('Location: account.html');
    exit;
}

$userId = (int)$_SESSION['user_id'];

// 1) Datos del usuario
$sqlUser = "
    SELECT
        name,
        email,
        phone,
        country,
        address,
        member_since
    FROM users
    WHERE id = :id
    LIMIT 1
";

```

```

$stmtUser = $pdo->prepare($sqlUser);
$stmtUser->execute([':id' => $userId]);
$user = $stmtUser->fetch(PDO::FETCH_ASSOC);

if (!$user) {
    // Si por alguna razón no existe el usuario, forzamos logout
    header('Location: ../backend/auth/logout.php');
    exit;
}

// Formatear fecha "miembro desde"
$memberSinceText = 'N/D';
if (!empty($user['member_since'])) {
    $fecha = new DateTime($user['member_since']);
    // Ej: Enero 2023
    $meses = [
        1 => 'Enero', 2 => 'Febrero', 3 => 'Marzo', 4 => 'Abril',
        5 => 'Mayo', 6 => 'Junio', 7 => 'Julio', 8 => 'Agosto',
        9 => 'Septiembre', 10 => 'Octubre', 11 => 'Noviembre', 12 => 'Diciembre'
    ];
    $mes = $meses[(int)$fecha->format('n')] ?? $fecha->format('m');
    $memberSinceText = $mes . ' ' . $fecha->format('Y');
}

// 2) Pedidos del usuario
$sqlOrders = "
SELECT
    id,
    total_amount,
    shipping_amount,
    status,
    created_at
FROM orders
WHERE user_id = :uid
ORDER BY created_at DESC
";
$stmtOrders = $pdo->prepare($sqlOrders);
$stmtOrders->execute([':uid' => $userId]);
$allOrders = $stmtOrders->fetchAll(PDO::FETCH_ASSOC);

// Últimos 3 pedidos para la sección "recientes"
$recentOrders = array_slice($allOrders, 0, 3);

```

```

// Helper para mostrar estado en texto y badge
function getOrderStatusLabel(string $status): array {
    // status en BD: 'pending','paid','shipped','cancelled'
    switch ($status) {
        case 'shipped':
            return ['Enviado', 'badge-shipping'];
        case 'paid':
            return ['Pagado', 'badge-delivered'];
        case 'pending':
            return ['Pendiente', 'badge-processing'];
        case 'cancelled':
        default:
            return ['Cancelado', 'badge-cancelled'];
    }
}

function formatOrderDate(string $datetime): string {
    $dt = new DateTime($datetime);
    return $dt->format('d M Y'); // Ej: 15 Nov 2025
}
?>
<!DOCTYPE html>
<html lang="es">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Mi Cuenta - RockStore</title>

    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">

    <!-- Font Awesome CDN -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@fortawesome/fontawesome-free@6.4.0/css/all.min.css">

    <!-- Tu CSS -->
    <link rel="stylesheet" href="../css/style2.css">
</head>

<body>
```

```

<header>
    <!-- METANAVEGACIÓN -->
    <di

Archivo: camisaM.php
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>RockStore - Página</title>

    <!-- Bootstrap CSS -->
    <link
        href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
        rel="stylesheet">

    <!-- Font Awesome CDN -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">

    <!-- Tu CSS -->
    <link rel="stylesheet" href="../css/styles.css">
</head>

<body>

    <!-- ● METANAVEGACIÓN -->
    <div class="top-meta">
        <a href="account.html">Iniciar sesión</a>
        <a href="account.html">Registrarse</a>
        <input type="text" placeholder="Buscar... ">
    </div>

    <!-- ● NAVBAR PRINCIPAL -->
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark px-3">
        <a class="navbar-brand" href="../html/home.php"></a> RockStore</a>

        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#menu">
            <span class="navbar-toggler-icon"></span>
        </button>

```

```

<div class="collapse navbar-collapse" id="menu">
  <ul class="navbar-nav ms-3">

    <li class="nav-item"><a class="nav-link" href="../html/home.php">Home</a></li>

    <!-- ☐ Dropdown con hover -->
    <li class="nav-item dropdown hover-dropdown">
      <a class="nav-link dropdown-toggle" href="../html/store.html" id="storeMenu" role="button">
        Tienda
      </a>

      <ul class="dropdown-menu" aria-labelledby="storeMenu">
        <li><a class="dropdown-item" href="../html/instrumentos.html">Instrumentos</a></li>
        <li><a class="dropdown-item" href="../html/ropaMujer.html">Ropa Mujer</a></li>
        <li><a class="dropdown-item" href="../html/ropaHombre.html">Ropa Hombre</a></li>
      </ul>
    </li>

    <li class="nav-item"><a class="nav-link" href="offers.php">Ofertas</a></li>
  </ul>
  <!-- Enlaces Cart y Account alineados a la derecha -->
  <ul class="navbar-nav ms-auto">
    <li class="nav-item">
      <a class="nav-link" href="cart.php">
        <i class="fas fa-shopping-cart"></i> Carrito
      </a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="accountusuario.php">
        <i class="fas fa-user"></i> Cuenta
      </a>
    </li>
  </ul>
</div>
</nav>

<body>
```

```

<!-- CATÁLOGO DE PRODUCTOS -->
<div class="catalog-section">
    <h6 class="section-title">Lo más popular hasta ahora</h6>

    <div class="products-grid">
        <!-- Camisa Rock Queen -->

        <?php
        // Incluir el catálogo reutilizable
        require_once __DIR__ . '/../backend/productos/catalogo.php';

        // Cargar productos de

```

Archivo: camisas.php

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>RockStore - Página</title>

    <!-- Bootstrap CSS -->
    <link
        href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
        rel="stylesheet">

    <!-- Font Awesome CDN -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">

    <!-- Tu CSS -->
    <link rel="stylesheet" href="../css/styles.css">
</head>

<body>

    <!-- ● METANAVEGACIÓN -->
    <div class="top-meta">
        <a href="account.html">Iniciar sesión</a>
        <a href="account.html">Registrarse</a>
        <input type="text" placeholder="Buscar... ">
    </div>

```

```

<!-- ● NAVBAR PRINCIPAL -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark px-3">
  <a class="navbar-brand" href=".//html/home.php">✖ RockStore</a>

  <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#menu">
    <span class="navbar-toggler-icon"></span>
  </button>

  <div class="collapse navbar-collapse" id="menu">
    <ul class="navbar-nav ms-3">

      <li class="nav-item"><a class="nav-link" href=".//html/home.php">Home</a></li>

      <!-- □ Dropdown con hover -->
      <li class="nav-item dropdown hover-dropdown">
        <a class="nav-link dropdown-toggle" href=".//html/store.html" id="storeMenu" role="button">
          Tienda
        </a>

        <ul class="dropdown-menu" aria-labelledby="storeMenu">
          <li><a class="dropdown-item" href=".//html/instrumentos.html">Instrumentos</a></li>
          <li><a class="dropdown-item" href=".//html/ropaMujer.html">Ropa Mujer</a></li>
          <li><a class="dropdown-item" href=".//html/ropaHombre.html">Ropa Hombre</a></li>
        </ul>
      </li>

      <li class="nav-item"><a class="nav-link" href="offers.php">Ofertas</a></li>
    </ul>
    <!-- Enlaces Cart y Account alineados a la derecha -->
    <ul class="navbar-nav ms-auto">
      <li class="nav-item">
        <a class="nav-link" href="cart.php">
          <i class="fas fa-shopping-cart"></i> Carrito
        </a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="accountusuario.php">

```

```

        <i class="fas fa-user"></i> Cuenta
    </a>
</li>
</ul>
</div>
</nav>

<body>


<div class="catalog-section">
    <h6 class="section-title">Lo más popular hasta ahora</h6>

    <div class="products-grid">
        <!-- Camisa Rock Rebel -->

        <?php
        // Incluir el catálogo reutilizable
        require_once __DIR__ . '/../backend/productos/catalogo.php';

        // Cargar productos de

```

Archivo: cart.php

```

<?php
// html/cart.php
if (session_status() === PHP_SESSION_NONE) {
    session_start();
}

require_once __DIR__ . '/../BD/conexion.php';

// Verificar que el usuario esté logueado
if (!isset($_SESSION['user_id'])) {
    // Redirigir a login o mostrar mensaje
    header('Location: account.html');
    exit;
}

$userId = (int)$_SESSION['user_id'];

// 1) Obtener carrito activo
$sqlCart =

```

```

SELECT id
FROM carts
WHERE user_id = :uid AND status = 'active'
LIMIT 1
";
$stmtCart = $pdo->prepare($sqlCart);
$stmtCart->execute([':uid' => $userId]);
$cart = $stmtCart->fetch(PDO::FETCH_ASSOC);

$items = [];
$subtotal = 0;

if ($cart) {
    $cartId = (int)$cart['id'];

    // 2) Obtener items del carrito + datos del producto
    $sqlItems = "
        SELECT
            ci.id      AS cart_item_id,
            ci.product_id AS product_id,
            ci.quantity  AS quantity,
            ci.unit_price AS unit_price,
            p.name      AS product_name,
            p.short_description,
            p.image_path,
            p.stock     AS stock
        FROM cart_items ci
        INNER JOIN products p ON ci.product_id = p.id
        WHERE ci.cart_id = :cid
    ";
    $stmtItems = $pdo->prepare($sqlItems);
    $stmtItems->execute([':cid' => $cartId]);
    $items = $stmtItems->fetchAll(PDO::FETCH_ASSOC);

    foreach ($items as $it) {
        $subtotal += $it['unit_price'] * $it['quantity'];
    }
} else {
    $cartId = null;
}

$shipping = $subtotal > 0 ? 20 : 0;
$total   = $subtotal + $shipping;

```

```

$totalItems = 0;
foreach ($items as $it) {
    $totalItems += (int)$it['quantity'];
}
?>
<!DOCTYPE html>
<html lang="es">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>RockStore - Carrito</title>

    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">

    <!-- Font Awesome -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.0/css/all.min.css">

    <!-- Tu CSS -->
    <link rel="stylesheet" href="../css/styles.css">
</head>

<body>

    <!-- ● METANAVEGACIÓN -->
    <div class="top-meta">
        <a href="account.html">Iniciar sesión</a>
        <a href="account.html">Registrarse</a>
        <input type="text" placeholder="Buscar...">
    </div>

    <!-- ● NAVBAR -->
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark px-3">
        <a class="navbar-brand" href="home.php">🔗 RockStore</a>

        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#menu">
            <span class="navbar-toggler-icon"></span>
        </button>

```

```
<div class="collapse navbar-collapse" id="menu">
    <ul class="navbar-nav ms-3">
        <li class="nav-item"><a class="nav-link" href="home.php">Home</a></li>
```

```
        <li class="nav-item dropdown hover-dropdown">
            <a class="nav-link"
```

Archivo: chalecos.php

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>RockStore - Página</title>

    <!-- Bootstrap CSS -->
    <link
        href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
        rel="stylesheet">

    <!-- Font Awesome CDN -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">

    <!-- Tu CSS -->
    <link rel="stylesheet" href="../css/styles.css">
</head>

<body>

    <!-- ● METANAVEGACIÓN -->
    <div class="top-meta">
        <a href="account.html">Iniciar sesión</a>
        <a href="account.html">Registrarse</a>
        <input type="text" placeholder="Buscar...">
    </div>

    <!-- ● NAVBAR PRINCIPAL -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark px-3">
    <a class="navbar-brand" href="../html/home.php">RockStore</a>

    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#menu">
```

```

<span class="navbar-toggler-icon"></span>
</button>

<div class="collapse navbar-collapse" id="menu">
  <ul class="navbar-nav ms-3">

    <li class="nav-item"><a class="nav-link" href="../html/home.php">Home</a></li>

    <!-- ▼ Dropdown con hover -->
    <li class="nav-item dropdown hover-dropdown">
      <a class="nav-link dropdown-toggle" href="../html/store.html" id="storeMenu" role="button">
        Tienda
      </a>

      <ul class="dropdown-menu" aria-labelledby="storeMenu">
        <li><a class="dropdown-item" href="../html/instrumentos.html">Instrumentos</a></li>
        <li><a class="dropdown-item" href="../html/ropaMujer.html">Ropa Mujer</a></li>
        <li><a class="dropdown-item" href="../html/ropaHombre.html">Ropa Hombre</a></li>
      </ul>
    </li>

    <li class="nav-item"><a class="nav-link" href="offers.php">Ofertas</a></li>
  </ul>
  <!-- Enlaces Cart y Account alineados a la derecha -->
  <ul class="navbar-nav ms-auto">
    <li class="nav-item">
      <a class="nav-link" href="cart.php">
        <i class="fas fa-shopping-cart"></i> Carrito
      </a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="accountusuario.php">
        <i class="fas fa-user"></i> Cuenta
      </a>
    </li>
  </ul>
</div>
</nav>

```

```

<body>

    <!-- CATÁLOGO DE PRODUCTOS -->
    <div class="catalog-section">
        <h6 class="section-title">Lo más popular hasta ahora</h6>

        <div class="products-grid">
            <!-- Chaleco Rocker Classic -->

            <?php
                // Incluir el catálogo reutilizable
                require_once __DIR__ . '/../backend/productos/catalogo.php';

            // Car

```

Archivo: cuerda.php

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>RockStore - Página</title>

    <!-- Bootstrap CSS -->
    <link
        href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
        rel="stylesheet">

    <!-- Font Awesome CDN -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">

    <!-- Tu CSS -->
    <link rel="stylesheet" href="../css/styles.css">
</head>

<body>

    <!-- ● METANAVEGACIÓN -->
    <div class="top-meta">
        <a href="account.html">Iniciar sesión</a>

```

```

<a href="account.html">Registrarse</a>
<input type="text" placeholder="Buscar...">
</div>

<!-- ● NAVBAR PRINCIPAL -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark px-3">
  <a class="navbar-brand" href="#">rif RockStore</a>

  <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#menu">
    <span class="navbar-toggler-icon"></span>
  </button>

  <div class="collapse navbar-collapse" id="menu">
    <ul class="navbar-nav ms-3">

      <li class="nav-item"><a class="nav-link" href="#">Home</a></li>

      <!-- □ Dropdown con hover -->
      <li class="nav-item dropdown hover-dropdown">
        <a class="nav-link dropdown-toggle" href="#" id="storeMenu" role="button">
          Tienda
        </a>

        <ul class="dropdown-menu" aria-labelledby="storeMenu">
          <li><a class="dropdown-item" href="#">Instrumentos</a></li>
          <li><a class="dropdown-item" href="#">Ropa Mujer</a></li>
          <li><a class="dropdown-item" href="#">Ropa Hombre</a></li>
        </ul>
      </li>

      <li class="nav-item"><a class="nav-link" href="#">Offers</a></li>
    </ul>
    <!-- Enlaces Cart y Account alineados a la derecha -->
    <ul class="navbar-nav ms-auto">
      <li class="nav-item">
        <a class="nav-link" href="cart.php">
          <i class="fas fa-shopping-cart"></i> Carrito
        </a>
      </li>
    </ul>
  </div>
</nav>

```

```

</li>
<li class="nav-item">
    <a class="nav-link" href="accountusuario.php">
        <i class="fas fa-user"></i> Cuenta
    </a>
</li>
</ul>
</div>
</nav>

```

```

<body>


<div class="catalog-section">
    <h6 class="section-title">Lo más popular hasta ahora</h6>

    <div class="products-grid">
        <!-- Guitarra Eléctrica Rockmaster -->

        <?php
        // Incluir el catálogo reutilizable
        require_once __DIR__ . '/../backend/productos/catalogo.php';

```

Archivo: gorras.php

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>RockStore - Página</title>

    <!-- Bootstrap CSS -->
    <link
        href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
        rel="stylesheet">

    <!-- Font Awesome CDN -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">

```

```

<!-- Tu CSS -->
<link rel="stylesheet" href="../css/styles.css">
</head>

<body>

<!-- ● METANAVEGACIÓN -->
<div class="top-meta">
    <a href="account.html">Iniciar sesión</a>
    <a href="account.html">Registrarse</a>
    <input type="text" placeholder="Buscar...">
</div>

<!-- ● NAVBAR PRINCIPAL -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark px-3">
    <a class="navbar-brand" href="../html/home.php"></a> RockStore</a>

    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#menu">
        <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="menu">
        <ul class="navbar-nav ms-3">

            <li class="nav-item"><a class="nav-link" href="../html/home.php">Home</a></li>

            <!-- ▼ Dropdown con hover -->
            <li class="nav-item dropdown hover-dropdown">
                <a class="nav-link dropdown-toggle" href="../html/store.html" id="storeMenu" role="button">
                    Tienda
                </a>

                <ul class="dropdown-menu" aria-labelledby="storeMenu">
                    <li><a class="dropdown-item" href="../html/instrumentos.html">Instrumentos</a></li>
                    <li><a class="dropdown-item" href="../html/ropaMujer.html">Ropa Mujer</a></li>
                    <li><a class="dropdown-item" href="../html/ropaHombre.html">Ropa Hombre</a></li>
                </ul>
            </li>
        </ul>
    </div>
</nav>

```

```

<li class="nav-item"><a class="nav-link" href="offers.php">Ofertas</a></li>
</ul>
<!-- Enlaces Cart y Account alineados a la derecha -->
<ul class="navbar-nav ms-auto">
    <li class="nav-item">
        <a class="nav-link" href="cart.php">
            <i class="fas fa-shopping-cart"></i> Carrito
        </a>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="accountusuario.php">
            <i class="fas fa-user"></i> Cuenta
        </a>
    </li>
</ul>
</div>
</nav>

```

```

<body>

<!-- CATÁLOGO DE PRODUCTOS -->
<div class="catalog-section">
    <h6 class="section-title">Lo más popular hasta ahora</h6>

    <div class="products-grid">
        <!-- Gorra Rock Legend -->

        <?php
        // Incluir el catálogo reutilizable
        require_once __DIR__ . '/../backend/productos/catalogo.php';

        // Cargar p

```

Archivo: home.php

```

<!DOCTYPE html>
<html lang="es">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>RockStore - Página</title>

```

```

<!-- Bootstrap CSS -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">

<!-- Font Awesome CDN -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.0/css/all.min.css">

<!-- CSS -->
<link rel="stylesheet" href="../css/styles.css">
</head>

<body>

<header>
<!-- METANAVEGACIÓN -->
<div class="top-meta">
    <a href="accountusuario.php">Iniciar sesión</a>
    <a href="accountusuario.php">Registrarse</a>
    <input type="text" placeholder="Buscar...">
</div>

<!-- NAVBAR PRINCIPAL -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark px-3">
    <a class="navbar-brand" href="home.php"></a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#menu">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="menu">
        <ul class="navbar-nav ms-3">
            <li class="nav-item">
                <a class="nav-link active" href="home.php">Home</a>
            </li>
            <!-- Dropdown Tienda -->
            <li class="nav-item dropdown hover-dropdown">
                <a class="nav-link dropdown-toggle" href="#" id="storeMenu" role="button">

```

```

data-bs-toggle="dropdown">
    Tienda
</a>

<ul class="dropdown-menu" aria-labelledby="storeMenu">
    <li><a class="dropdown-item"
        href="instrumentos.html">Instrumentos</a></li>
        <li><a class="dropdown-item" href="ropaMujer.html">Ropa Mujer</a></li>
        <li><a class="dropdown-item" href="ropaHombre.html">Ropa
        Hombre</a></li>
        <!-- Si luego creas una tienda general, se añade aquí -->
    </ul>
</li>

<li class="nav-item">
    <a class="nav-link" href="offers.php">Ofertas</a>
</li>
</ul>

<!-- Menú derecho -->
<ul class="navbar-nav ms-auto">
    <li class="nav-item">
        <a class="nav-link" href="cart.php">
            <i class="fas fa-shopping-cart"></i> Carrito
        </a>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="accountusuario.php">
            <i class="fas fa-user"></i> Cuenta
        </a>
    </li>
</ul>
</div>
</nav>

```

```

<div class="page-grid">
    <main class="content">
        <!-- Contenido según la página -->

```

```

<div id="carouselExampleIndicators

Archivo: offers.php
<?php
// html/offers.php
// Muestra TODOS los productos que tengan oferta activa en la BD

// No es obligatorio usar sesión aquí, a menos que quieras mostrar el nombre del usuario.
// Si luego quieres condicionar algo por login, puedes activar esto:
// if (session_status() === PHP_SESSION_NONE) { session_start(); }

?>
<!DOCTYPE html>
<html lang="es">

<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>RockStore - Ofertas</title>

<!-- Bootstrap CSS -->
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js">
<link
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
    rel="stylesheet">

<!-- Font Awesome CDN -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">

<!-- Tu CSS -->
<link rel="stylesheet" href="../css/styles.css">
</head>

<body>

<!-- METANAVEGACIÓN -->
<div class="top-meta">
    <a href="login.html">Iniciar sesión</a>
    <a href="register.html">Registrarse</a>
    <input type="text" placeholder="Buscar...">

```

```

</div>

<!-- ● NAVBAR PRINCIPAL -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark px-3">
    <a class="navbar-brand" href="home.php"></a>

    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#menu">
        <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="menu">
        <ul class="navbar-nav ms-3">
            <li class="nav-item"><a class="nav-link" href="home.php">Home</a></li>

            <!-- ▾ Dropdown -->
            <li class="nav-item dropdown hover-dropdown">
                <a class="nav-link dropdown-toggle" id="storeMenu" role="button">
                    Tienda
                </a>
                <ul class="dropdown-menu" aria-labelledby="storeMenu">
                    <li><a class="dropdown-item" href="instrumentos.html">Instrumentos</a></li>
                    <li><a class="dropdown-item" href="ropaMujer.html">Ropa Mujer</a></li>
                    <li><a class="dropdown-item" href="ropaHombre.html">Ropa Hombre</a></li>
                </ul>
            </li>

            <li class="nav-item"><a class="nav-link active" href="offers.php">Ofertas</a></li>
        </ul>
    </div>

    <!-- A la derecha -->
    <ul class="navbar-nav ms-auto">
        <li class="nav-item">
            <a class="nav-link" href="cart.php">
                <i class="fas fa-shopping-cart"></i> Carrito
            </a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="accountusuario.php">
                <i class="fas fa-user"></i> Cuenta
            </a>
        </li>
    </ul>
</nav>

```

```
</li>
```

Archivo: percusion.php

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>RockStore - Página</title>

    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">

    <!-- Font Awesome CDN -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@fortawesome/fontawesome-free@6.4.0/css/all.min.css" />

    <!-- Tu CSS -->
    <link rel="stylesheet" href="../css/styles.css" />
</head>

<body>

    <!-- METANAVEGACIÓN -->
    <div class="top-meta">
        <a href="account.html">Iniciar sesión</a>
        <a href="account.html">Registrarse</a>
        <input type="text" placeholder="Buscar..." />
    </div>

    <!-- NAVBAR PRINCIPAL -->
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark px-3">
        <a class="navbar-brand" href="..//html/home.php">RockStore</a>

        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#menu">
            <span class="navbar-toggler-icon"></span>
        </button>

        <div class="collapse navbar-collapse" id="menu">
```

```

<ul class="navbar-nav ms-3">

    <li class="nav-item"><a class="nav-link" href="../html/home.php">Home</a></li>

    <!-- ☐ Dropdown con hover -->
    <li class="nav-item dropdown hover-dropdown">
        <a class="nav-link dropdown-toggle" href="../html/store.html" id="storeMenu" role="button">
            Tienda
        </a>

        <ul class="dropdown-menu" aria-labelledby="storeMenu">
            <li><a class="dropdown-item" href="../html/instrumentos.html">Instrumentos</a></li>
            <li><a class="dropdown-item" href="../html/ropaMujer.html">Ropa Mujer</a></li>
            <li><a class="dropdown-item" href="../html/ropaHombre.html">Ropa Hombre</a></li>
        </ul>
    </li>

    <li class="nav-item"><a class="nav-link" href="offers.php">Ofertas</a></li>
</ul>
<!-- Enlaces Cart y Account alineados a la derecha -->
<ul class="navbar-nav ms-auto">
    <li class="nav-item">
        <a class="nav-link" href="cart.php">
            <i class="fas fa-shopping-cart"></i> Carrito
        </a>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="accountusuario.php">
            <i class="fas fa-user"></i> Cuenta
        </a>
    </li>
</ul>
</div>
</nav>
</head>

<body>

```

```

<!-- CATÁLOGO DE PRODUCTOS -->
<div class="catalog-section">
    <h6 class="section-title">Lo más popular hasta ahora</h6>

    <div class="products-grid">
        <!-- Batería Rock Thunder -->

        <?php
        // Incluir el catálogo reutilizable
        require_once __DIR__ . '/../backend/productos/catalogo.php';
        rend

```

Archivo: vestidos.php

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>RockStore - Página</title>

    <!-- Bootstrap CSS -->
    <link
        href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
        rel="stylesheet">

    <!-- Font Awesome CDN -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">

    <!-- Tu CSS -->
    <link rel="stylesheet" href="../css/styles.css">
</head>

<body>

    <!-- ● METANAVEGACIÓN -->
    <div class="top-meta">
        <a href="account.html">Iniciar sesión</a>
        <a href="account.html">Registrarse</a>
        <input type="text" placeholder="Buscar...">
    </div>

```

```

<!-- ● NAVBAR PRINCIPAL -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark px-3">
  <a class="navbar-brand" href="#">RockStore</a>

  <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#menu">
    <span class="navbar-toggler-icon"></span>
  </button>

  <div class="collapse navbar-collapse" id="menu">
    <ul class="navbar-nav ms-3">

      <li class="nav-item"><a class="nav-link" href="#">Home</a></li>

      <!-- ☑ Dropdown con hover -->
      <li class="nav-item dropdown hover-dropdown">
        <a class="nav-link dropdown-toggle" href="#" id="storeMenu"
role="button">
          Tienda
        </a>

        <ul class="dropdown-menu" aria-labelledby="storeMenu">
          <li><a class="dropdown-item"
href="#">Instrumentos</a></li>
          <li><a class="dropdown-item" href="#">Ropa
Mujer</a></li>
          <li><a class="dropdown-item" href="#">Ropa
Hombre</a></li>
        </ul>
      </li>

      <li class="nav-item"><a class="nav-link" href="#">Ofertas</a></li>
    </ul>
    <!-- Enlaces Cart y Account alineados a la derecha -->
    <ul class="navbar-nav ms-auto">
      <li class="nav-item">
        <a class="nav-link" href="cart.php">
          <i class="fas fa-shopping-cart"></i> Carrito
        </a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="accountusuario.php">
          <i class="fas fa-user"></i> Cuenta
        </a>
      </li>
    </ul>
  </div>
</nav>

```

```

        </a>
    </li>
</ul>
</div>
</nav>

<body>


<div class="catalog-section">
    <h6 class="section-title">Lo más popular hasta ahora</h6>

    <div class="products-grid">
        <!-- Vestido Funk Groove -->

        <?php
        // Incluir el catálogo reutilizable
        require_once __DIR__ . '/../backend/productos/catalogo.php';

        // Cargar

```

Archivo: viento.php

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>RockStore - Página</title>

    <!-- Bootstrap CSS -->
    <link
        href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
        rel="stylesheet">

    <!-- Font Awesome CDN -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
    <!-- Tu CSS -->
    <link rel="stylesheet" href="../css/styles.css">
</head>

<body>

```

```

<!-- ● METANAVEGACIÓN -->
<div class="top-meta">
    <a href="account.html">Iniciar sesión</a>
    <a href="account.html">Registrarse</a>
    <input type="text" placeholder="Buscar...">
</div>

<!-- ● NAVBAR PRINCIPAL -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark px-3">
    <a class="navbar-brand" href=".//html/home.php">✖ RockStore</a>

    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#menu">
        <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="menu">
        <ul class="navbar-nav ms-3">

            <li class="nav-item"><a class="nav-link" href=".//html/home.php">Home</a></li>

            <!-- ▾ Dropdown con hover -->
            <li class="nav-item dropdown hover-dropdown">
                <a class="nav-link dropdown-toggle" href=".//html/store.html" id="storeMenu" role="button">
                    Tienda
                </a>

                <ul class="dropdown-menu" aria-labelledby="storeMenu">
                    <li><a class="dropdown-item" href=".//html/instrumentos.html">Instrumentos</a></li>
                    <li><a class="dropdown-item" href=".//html/ropaMujer.html">Ropa Mujer</a></li>
                    <li><a class="dropdown-item" href=".//html/ropaHombre.html">Ropa Hombre</a></li>
                </ul>
            </li>

            <li class="nav-item"><a class="nav-link" href="offers.php">Ofertas</a></li>
        </ul>
    </div>
    <!-- Enlaces Cart y Account alineados a la derecha -->
    <ul class="navbar-nav ms-auto">

```

```

<li class="nav-item">
    <a class="nav-link" href="cart.php">
        <i class="fas fa-shopping-cart"></i> Carrito
    </a>
</li>
<li class="nav-item">
    <a class="nav-link" href="accountusuario.php">
        <i class="fas fa-user"></i> Cuenta
    </a>
</li>
</ul>
</div>
</nav>
>

<body>

<!-- CATÁLOGO DE PRODUCTOS -->
<div class="catalog-section">
    <h6 class="section-title">Lo más popular hasta ahora</h6>

    <div class="products-grid">
        <!-- Saxofón Rock Legend -->

        <?php
        // Incluir el catálogo reutilizable
        require_once __DIR__ . '/../backend/productos/catalogo.php';
        renderProductosGrid([

```

3. Documentacion en Github

<https://github.com/Fernando-lezcano/trabajo-fianla-ing-web>

cliente: correo: cliente1@rockstore.com m contraseña: 12345678

usuaios normales se registran sin necesidad de contraseña

4. Tabla colaborativa

Aporte	Integrante	Fecha
base de datos	Jeremias Aparicio Fernando Lezcano	01/12/2025
login.php	Jeremias Aparicio Fernando Lezcano Cristofer Jimenez	01/12/2025
logout.php	Jeremias Aparicio Fernando Lezcano Alejandra Gomez	01/12/2025
registro.php	Jeremias Aparicio Fernando Lezcano Cristofer Jimenez	01/12/2025
add_to_cart.php	Jeremias Aparicio Fernando Lezcano Alejandra Gomez	01/12/2025
checkout.php	Jeremias Aparicio Fernando Lezcano Alejandra Gomez	01/12/2025
remove_item.php	Jeremias Aparicio Fernando Lezcano Cristofer Jimenez	01/12/2025
update_item.php	Jeremias Aparicio Fernando Lezcano Alejandra Gomez	01/12/2025
listar.php	Jeremias Aparicio Fernando Lezcano Cristofer Jimenez	02/12/2025
catalogo.php	Jeremias Aparicio Fernando Lezcano Alejandra Gomez	02/12/2025
eliminar.php	Jeremias Aparicio Fernando Lezcano Cristofer Jimenez	02/12/2025
guardar.php	Jeremias Aparicio Fernando Lezcano Alejandra Gomez	02/12/2025
listar.php	Jeremias Aparicio Fernando Lezcano Cristofer Jimenez	02/12/2025
conexion.php	Jeremias Aparicio Fernando Lezcano Alejandra Gomez	02/12/2025

accesoriosM.php	Jeremias Aparicio Fernando Lezcano Cristofer Jimenez	02/12/2025
accountusuario.php	Jeremias Aparicio Fernando Lezcano Alejandra Gomez	03/12/2025
camisaM.php	Jeremias Aparicio Fernando Lezcano Cristofer Jimenez	03/12/2025
camisas.php	Jeremias Aparicio Fernando Lezcano Alejandra Gomez	03/12/2025
cart.php	Jeremias Aparicio Fernando Lezcano Cristofer Jimenez	03/12/2025
chalecos.php	Jeremias Aparicio Fernando Lezcano Alejandra Gomez	03/12/2025
cuerda.php	Jeremias Aparicio Fernando Lezcano Cristofer Jimenez	03/12/2025
gorras.php	Jeremias Aparicio Fernando Lezcano Alejandra Gomez	03/12/2025
home.php	Jeremias Aparicio Fernando Lezcano Cristofer Jimenez	03/12/2025
offers.php	Jeremias Aparicio Fernando Lezcano Alejandra Gomez	03/12/2025
percusion.php	Jeremias Aparicio Fernando Lezcano Cristofer Jimenez	03/12/2025
vestidos.php	Jeremias Aparicio Fernando Lezcano Alejandra Gomez	03/12/2025
viento.php	Jeremias Aparicio Fernando Lezcano Cristofer Jimenez	03/12/2025

5. Link del sitio web

<https://rockstore.infinityfreeapp.com/>