Vintage Store

Documentación de Proyecto

**Integrantes:** Josefa Cardenal

**Fecha:** 14/10/2024

# DESCRIPCIÓN DEL SISTEMA

## Introducción

Este proyecto es una tienda en línea de estilo vintage que permite a los usuarios navegar por productos, filtrarlos por categorías y rangos de precio, y realizar compras. El sistema incluye una página de inicio para la navegación y un sistema de autenticación de usuarios.

***1.2 Proceso de Desarrollo***

El desarrollo del proyecto se llevó a cabo en varias fases:

1. **Conexión a la Base de Datos**: Se estableció una conexión segura con una base de datos MySQL para almacenar y recuperar información de los productos y usuarios.
2. **Filtrado y Visualización de Productos**: Se desarrollaron funciones para filtrar productos según categoría, precio y término de búsqueda.
3. **Sistema de Autenticación**: Se creó una página de inicio de sesión para verificar los datos de los usuarios y proporcionar acceso a la tienda.
4. **Estructura de las Páginas HTML y CSS**: Se diseñó la interfaz con HTML y CSS para ofrecer una experiencia de usuario coherente y amigable.
5. **Tecnologías Utilizadas**

* **PHP**: Para la lógica del servidor, procesamiento de formularios y conexión con la base de datos.
* **MySQL**: Para almacenar datos de productos y usuarios.
* **HTML/CSS**: Para la estructura y estilo de la interfaz de usuario.

***1.3 Estructura de la Base de Datos***

El proyecto incluye al menos dos tablas principales:

1. **Tabla productos**:
   * id\_producto (INT): Identificador único del producto.
   * nombre\_producto (VARCHAR): Nombre del producto.
   * descripcion (TEXT): Descripción detallada del producto.
   * precio (DECIMAL): Precio del producto.
   * imagen (VARCHAR): Ruta de la imagen del producto.
   * categoria (VARCHAR): Categoría del producto.
2. **Tabla usuarios**:
   * id\_usuario (INT): Identificador único del usuario.
   * nombre\_cliente (VARCHAR): Nombre de usuario.
   * contraseña (VARCHAR): Contraseña del usuario (en texto plano en el código actual, debería ser mejorada con hashing).

**Funcionamiento y Estructura del Sistema**

|  |
| --- |
| **Archivo:** db.php  Este archivo se encarga de la conexión a la base de datos. Utiliza los siguientes parámetros:   * **Host**: localhost * **Usuario**: root * **Contraseña**: (vacía en este caso) * **Nombre de la base de datos**: tienda vintage   $conn = new mysqli($servername, $username, $password, $dbname) |

|  |
| --- |
| **Archivo:** index.html (Login) (Vista)  El archivo HTML contiene los siguientes elementos:   1. **Encabezado (login-header)**: Incluye un título y un mensaje de bienvenida. 2. **Formulario de Login (login-form)**: Permite a los usuarios ingresar sus credenciales.    * **Campo Usuario**: Un campo de texto para ingresar el nombre de usuario.    * **Campo Contraseña**: Un campo de tipo password para ingresar la contraseña.    * **Botón de Inicio de Sesión**: Envía el formulario a login.php mediante el método POST. 3. **Enlace de Registro (signup-link)**: Un enlace para redirigir a los usuarios al registro (actualmente sin funcionalidad, solo texto). |

|  |
| --- |
| **Archivo:** login.php (Login) (funcionalidad)  Este archivo maneja la validación de las credenciales de inicio de sesión ingresadas en index.php. Se conecta a la base de datos, verifica si el usuario existe y valida que la contraseña sea correcta. Si las credenciales son válidas, redirige al usuario a la página principal; si no, muestra un mensaje de error.   1. **Lógica del Código** 2. **Conexión a la Base de Datos**: Incluye el archivo db.php para establecer la conexión a la base de datos. 3. **Captura de Datos**: Utiliza $\_POST para obtener el nombre de usuario y contraseña enviados desde el formulario. 4. **Consulta a la Base de Datos**: Prepara una consulta SQL para verificar si el usuario existe en la base de datos.    * Utiliza prepare y bind\_param para evitar inyecciones SQL. 5. **Verificación de Usuario y Contraseña**:    * Si el usuario existe, compara la contraseña ingresada con la almacenada en la base de datos (actualmente en texto plano, sin hashing).    * Si la contraseña es correcta, redirige a la página principal (../).    * Si la contraseña o el nombre de usuario son incorrectos, muestra un mensaje de error. 6. **Cierre de Conexión**: Cierra la conexión a la base de datos. |

|  |
| --- |
| **Archivo:** index.php (Home) (Tienda)  El archivo index.php es la página principal de la tienda en línea. Esta página muestra una lista de productos que se pueden filtrar por categoría, rango de precio o mediante una búsqueda. Los resultados se muestran dinámicamente en tarjetas de producto, con la opción de realizar una compra mediante un formulario.  La página incluye el archivo **db.php** para establecer una conexión a la base de datos, permitiendo que los productos se obtengan de la tabla correspondiente.    La página verifica si se han pasado filtros de categoría, precio o término de búsqueda a través de la URL (GET). Si existen, estos filtros se capturan en variables.    Se construye una **consulta** **SQL** para seleccionar productos desde la base de datos en función de los filtros proporcionados. Para cada filtro (categoría, precio, búsqueda), se añade una condición a la consulta si se ha especificado.    Se prepara y ejecuta la consulta con los filtros aplicados. Si hay condiciones de categoría o búsqueda, los valores se vinculan a la consulta preparada para evitar inyecciones SQL.    **ESTRUCTURA HTML**  **Encabezado (Header)**  El encabezado contiene el logo y una barra de búsqueda. La búsqueda envía una solicitud GET para actualizar la lista de productos en función del término ingresado.    **Barra Lateral (Sidebar)**  La barra lateral contiene enlaces para filtrar productos por categoría y precio. Al hacer clic en un filtro, la página se recarga con los productos correspondientes.    **Sección de Productos (Products)**  Esta sección muestra una lista de productos en forma de tarjetas. Si hay productos disponibles según los filtros aplicados, cada producto se muestra con su imagen, descripción, precio y un botón de compra. Si no hay productos, se muestra un mensaje informativo. |
| **Archivo:** index.php (Confirmación de Compras)  Este archivo muestra la página de confirmación de compra, que se carga después de que un usuario completa una compra en la tienda. La página muestra un mensaje de confirmación junto con los detalles del producto comprado, tales como la imagen, el nombre y el precio.  **Conexión a la base de datos**  Incluye el archivo db.php para conectarse a la base de datos.  **Verificación de Datos POST**  La página verifica que se haya enviado un formulario POST con los datos del producto. Si la solicitud no es POST, se redirige al usuario a la página principal para evitar accesos no autorizados.    **Estructura HTML**  **Encabezado (Header)**  El encabezado incluye el logo de la tienda y una barra de búsqueda, permitiendo al usuario realizar nuevas búsquedas o navegar desde esta página.    **Sección de Confirmación (Confirmation Section)**  La sección principal de la página contiene un mensaje de confirmación y una tarjeta que muestra los detalles del producto comprado, incluyendo:   * **Imagen del Producto**: Muestra la imagen del producto usando la ruta especificada en el formulario POST. * **Nombre del Producto**: Se muestra el nombre del producto. * **Precio del Producto**: Se muestra el precio del producto, formateado a dos decimales. |

|  |
| --- |
| **Estructura de la BD**  está compuesta por tres tablas principales: productos, usuarios y ventas. A continuación se describe cada tabla, junto con los campos que contiene y su propósito.  **Tabla productos**  Esta tabla almacena la información de los productos disponibles en la tienda.   * **Campos**:   + **id\_producto** (INT, PRIMARY KEY): Identificador único de cada producto.   + **categoria** (VARCHAR): Categoría a la que pertenece el producto (por ejemplo, "Accesorios", "Calzado").   + **descripcion** (TEXT): Descripción detallada del producto.   + **imagen** (VARCHAR): Nombre o ruta de la imagen del producto.   + **nombre\_producto** (VARCHAR): Nombre del producto.   + **precio** (INT): Precio del producto en la moneda local.   + **stock** (INT): Cantidad disponible del producto en el inventario.   *Esta tabla permite gestionar y mostrar los productos a los usuarios en la página principal. Los campos categoria, precio y stock son clave para los filtros y la disponibilidad de los productos.*  **Tabla usuarios**  Esta tabla contiene los datos de los clientes registrados en la tienda.   * **Campos**:   + **id\_cliente** (INT, PRIMARY KEY): Identificador único de cada cliente.   + **nombre\_cliente** (VARCHAR): Nombre del cliente o nombre de usuario.   + **contraseña** (VARCHAR): Contraseña del cliente, idealmente en formato hash para mayor seguridad.   + **correo** (VARCHAR): Dirección de correo electrónico del cliente.   + **dirección** (VARCHAR): Dirección física del cliente para fines de envío.   *Esta tabla es esencial para la autenticación y el perfil del usuario. Los campos nombre\_cliente y contraseña se utilizan para iniciar sesión en el sistema, mientras que correo y dirección se pueden emplear en el proceso de compra y envío.*  ***Tabla ventas***  La tabla ventas registra cada transacción realizada en la tienda, almacenando la relación entre clientes y productos comprados.   * **Campos**:   + **id\_venta** (INT, PRIMARY KEY): Identificador único de cada transacción de venta.   + **fecha\_venta** (DATE): Fecha en la que se realizó la venta.   + **id\_cliente** (INT, FOREIGN KEY): Identificador del cliente que realizó la compra, asociado a usuarios.id\_cliente.   + **id\_producto** (INT, FOREIGN KEY): Identificador del producto vendido, asociado a productos.id\_producto.   + **monto\_total** (DECIMAL): Monto total de la transacción.   *La tabla ventas permite realizar un seguimiento de las compras de cada cliente. Almacena tanto el cliente como el producto, junto con la fecha y el monto total de la venta, facilitando el análisis de ventas y el historial de transacciones.*  **Relaciones entre las Tablas**   * **ventas.id\_cliente → usuarios.id\_cliente**: Relación que asocia cada venta a un cliente específico. * **ventas.id\_producto → productos.id\_producto**: Relación que asocia cada venta a un producto específico. |

|  |
| --- |
| **Interfaz Tienda Vintage**  Página de Ingreso (Login.html)    Página de inicio/tienda (index.php)    Página de compra confirmada (/compras/index.php) |