Logotipo

Descripción generada automáticamente

**SISTEMA DE INFORMACIÓN PARA EL MANEJO DE INVENTARIO Y VENTAS ONLINE DE LA LIBRERIA SIGLO DEL HOMBRE EN EL BARRIO PERDOMO DE LA CIUDAD DE BOGOTÁ**

**INTEGRANTES:**

**BRET MILTON SIERRA ROYERO**

**LUIS GABRIEL MENDEZ ÑUSTES**

**CARLOS ANDRÉS POLANÍA QUIJANO**

**TÉCNICO EN PROGRAMACIÓN DE SOFTWARE**

**CENTRO DE ELECTRICIDAD, ELECTRONICA Y TELECOMUNICACIONES**

**SENA**

**Control De Versiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autores** |
| 10/11/2024 | 1.0 | Versión Inicial del manual | Bret Milton Sierra Royero  Carlos Andrés Polania Quijano  Luis Gabriel Mendez Ñustes |

**Tabla de Contenido de Figuras y Diagramas**

1. Figura 1 17
2. Figura 2 21
3. Diagrama 1 29
4. Diagrama 2 30
5. Diagrama 3 32
6. Diagrama 4 34

### **Tabla De Contenido**

1. Introducción 4
2. Objetivo del manual 5
3. Objetivo General 6
4. Objetivos Específicos 6
5. Requerimientos técnicos del sistema 7
6. Tecnologías usadas 9
7. Instalación y configuración 10
8. Arquitectura del sistema 15
9. Requerimientos Del Proyecto 17
10. Modelo entidad relación 19
11. Diccionario de datos 21
12. Diagramas de casos de uso 28
13. Subcasos Principales 29
14. Metodología de desarrollo 35
15. Manual de usuario 37
16. Mantenimiento del sistema 40

### **Introducción**

El presente manual técnico detalla la implementación, configuración, uso y mantenimiento de la página web de ventas en línea de la librería \*El Hombre del Siglo\*. Este sistema está diseñado para ofrecer una experiencia de compra rápida y amigable a los usuarios interesados en adquirir libros desde cualquier ubicación.

Además, el sistema también incluye una funcionalidad integral de gestión de inventario, permitiendo a la librería mantener un control preciso y actualizado de sus existencias. Esto facilita la toma de decisiones relacionadas con reposiciones, ventas y proyecciones de stock.

La aplicación fue desarrollada utilizando tecnologías modernas como PHP, Bootstrap, CSS, HTML y JavaScript, y se ejecuta localmente con XAMPP como servidor.

**Objetivo del manual**

Este manual está dirigido a los desarrolladores, administradores de sistema y cualquier otra persona encargada de operar o mantener la aplicación, con el fin de que puedan operar mantener y mejorar el sistema para una mejor interacción con los usuarios clientes y administradores.

También está pensado para los responsables de la gestión del inventario, quienes podrán usar las herramientas del sistema para supervisar y controlar los productos de la librería de manera eficiente, tener control del stock de inventario, conocimiento de las existencias en las diferentes ubicaciones, tener registro de cada una de las ventas, y usuarios activos en el sistema.

**OBJETIVO GENERAL**

Desarrollar e implementar un sistema de información para el manejo de inventario y ventas online de la librería Siglo del hombre, ubicada en la ciudad de Bogotá en el barrio Perdomo ciudad de Bogotá

**OBJETIVOS ESPECÍFICOS**

1. Desarrollar un módulo específico y eficiente destinado a la gestión de ventas online, integrado de manera efectiva con el sistema general de información.
2. Implementar un módulo especializado para el manejo, control y operación de inventarios que permita un seguimiento preciso de los niveles de stock y facilite la reposición oportuna de productos.
3. Implementar técnicas actuales de desarrollo de software para diseñar eficazmente la base de datos y las interfaces del sistema, centrándose en la creación de un sistema sencillo y funcional.
4. Realizar pruebas de software para garantizar la funcionalidad, precisión y eficiencia del software antes de su implementación.

**Requerimientos técnicos del sistema**

Para garantizar el correcto funcionamiento de la página web de ventas en línea y la gestión de inventario, se establecen los siguientes requisitos mínimos y recomendados:

**Hardware:**

**Procesador:**

Mínimo: Intel Core i3 de cuarta generación o equivalente.

Recomendado: Intel Core i5 o superior para un rendimiento óptimo.

**Memoria RAM:**

Mínimo: 4 GB para ejecutar el sistema y el entorno del servidor.

Recomendado: 8 GB o más para un mejor desempeño, especialmente en entornos multitarea.

**Espacio en disco:**

Mínimo: 500 MB de almacenamiento libre para la instalación del servidor, la base de datos y los archivos de la aplicación.

Recomendado: 1 GB para permitir expansiones futuras y almacenamiento de datos adicionales.

**Software:**

**Sistema operativo:**

Mínimo: Windows 10 o distribuciones de Linux compatibles, como Ubuntu 20.04 LTS.

Recomendado: Windows 11 o distribuciones Linux más recientes para garantizar soporte y actualizaciones de seguridad.

**Navegador web:**

Compatible con los navegadores modernos como:

Google Chrome (versión 90 o superior).

Mozilla Firefox (versión 85 o superior).

Microsoft Edge (versión 90 o superior).

**Servidor local:**

XAMPP, con una versión mínima de 8.0, que incluye:

Apache como servidor web.

MySQL para la gestión de la base de datos.

PHP en su versión 8.0 o superior, para garantizar compatibilidad con el código del sistema.

**Requerimientos adicionales:**

Acceso a Internet: Aunque el sistema funciona en un entorno local, se recomienda una conexión a internet para realizar actualizaciones de software o cargar la página en servidores remotos en caso de migración.

**Tecnologías utilizadas**

1. **PHP:**
   1. Es el lenguaje de scripting del lado del servidor que gestiona la lógica de negocio, incluyendo la validación de datos, la interacción con la base de datos MySQL y el manejo de sesiones.
   2. Permite la integración dinámica de contenido en las páginas, lo que hace posible mostrar productos actualizados y resultados personalizados para cada usuario.
2. **Software Libre:**

El uso de software libre es una práctica común en el desarrollo de aplicaciones web, ya que ofrece ventajas como la reducción de costos y la posibilidad de modificar el código fuente según las necesidades del proyecto. En este caso, el proyecto de la librería hace uso de herramientas de software libre como PHP y MySQL, lo que permite un alto grado de personalización y control sobre el sistema.

1. **Bootstrap:**
   1. Framework CSS ampliamente usado para desarrollar sitios web con diseño responsive.
   2. Facilita la creación de interfaces atractivas, consistentes y adaptables a distintos dispositivos como teléfonos, tablets y computadoras de escritorio.
   3. Incluye componentes predefinidos como botones, menús desplegables y barras de navegación, optimizando el tiempo de desarrollo.
2. **CSS y HTML:**
   1. HTML se utiliza para la estructura de las páginas web, definiendo los elementos principales como encabezados, formularios y tablas.
   2. CSS se encarga de la personalización visual, incluyendo colores, tipografía, espaciado y otros aspectos estilísticos que mejoran la experiencia visual.
   3. Juntos, proporcionan un contenido organizado y una presentación clara para el usuario.
3. **JavaScript:**
   1. Mejora la interactividad de la página con funciones como validación en tiempo real de formularios, menús desplegables dinámicos y ventanas emergentes.
   2. Permite implementar animaciones y efectos visuales, haciendo la experiencia más fluida y atractiva.
   3. Además, se utiliza para la comunicación asincrónica con el servidor mediante AJAX, lo que reduce la necesidad de recargar la página.
4. **XAMPP:**
   1. Paquete de software que incluye Apache como servidor web, MySQL como sistema de gestión de bases de datos, y soporte para PHP y Perl.
   2. Es esencial para probar y ejecutar la aplicación en un entorno local antes de implementarla en un servidor en producción.
   3. Facilita la instalación y configuración del entorno de desarrollo, permitiendo que los desarrolladores trabajen en un sistema unificado y eficiente.
5. **Visual studio code**
6. Visual Studio Code es un editor de código fuente gratuito y de código abierto, desarrollado por Microsoft. Es altamente personalizable mediante extensiones, lo que permite adaptarlo para manejar múltiples lenguajes de programación como PHP, conectarse a bases de datos, y proporcionar herramientas avanzadas para la depuración e identificación de errores, optimizando la experiencia del desarrollador.

**Instalación y Configuración**

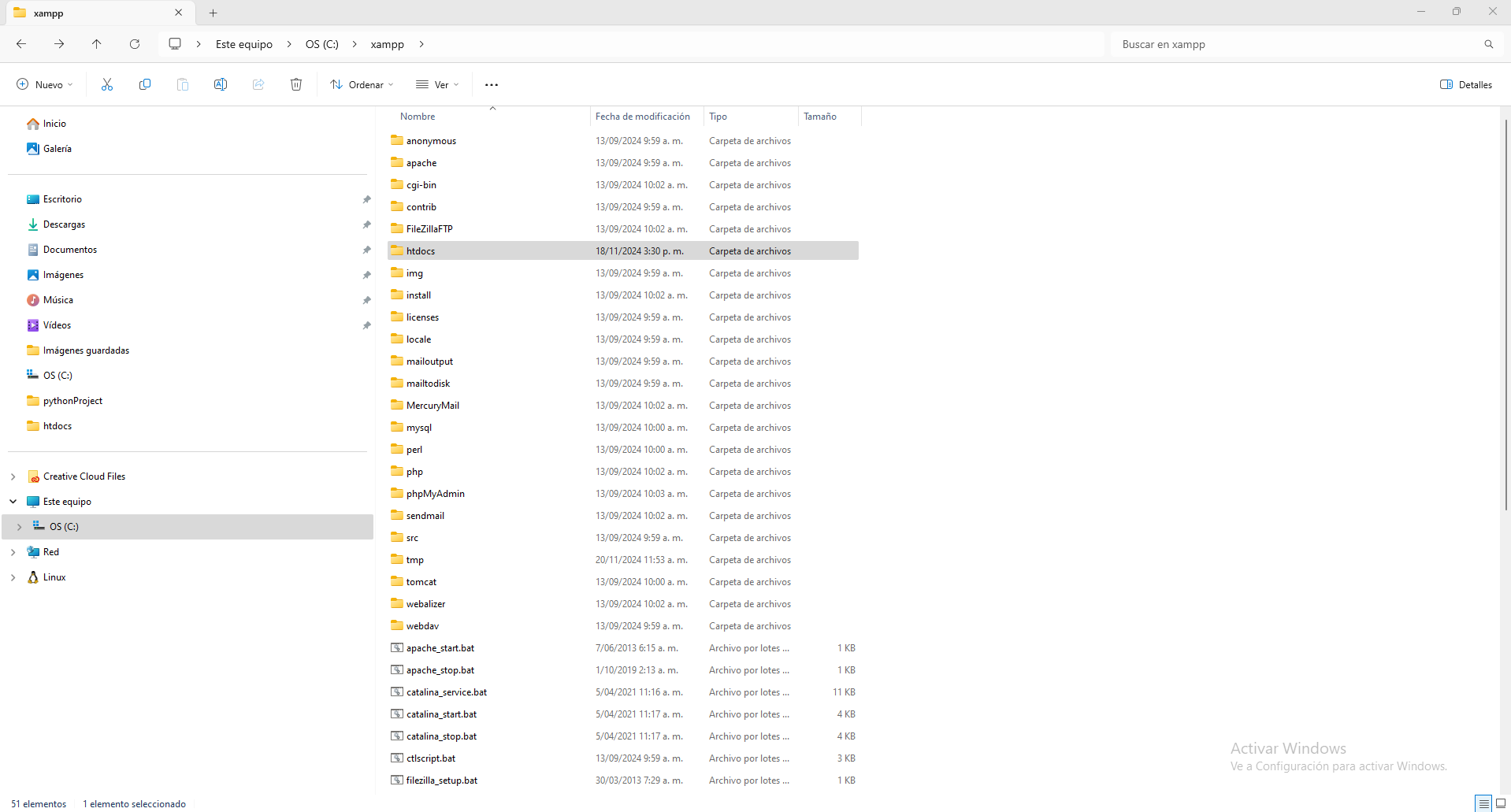
1. **Instalación de XAMPP:**
   1. Descargue la última versión de XAMPP compatible (mínimo 8.0) desde su página oficial: <https://www.apachefriends.org>.
   2. Complete la instalación siguiendo las instrucciones del asistente.



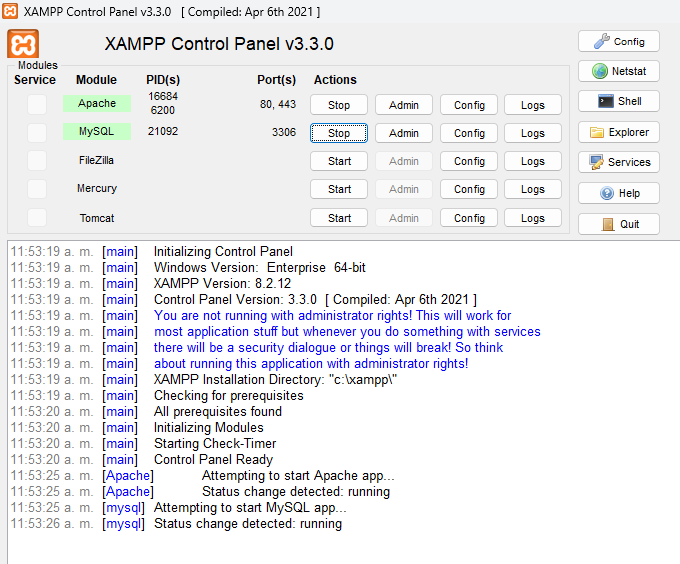
1. **Obtención del proyecto:**
   1. Descargue el proyecto de la librería desde el repositorio proporcionado (por ejemplo, GitHub) o mediante una carpeta compartida.
   2. Asegúrese de incluir todos los archivos necesarios, incluidos los scripts de la base de datos (*libreria.sql*).

**Pasos de instalación:**

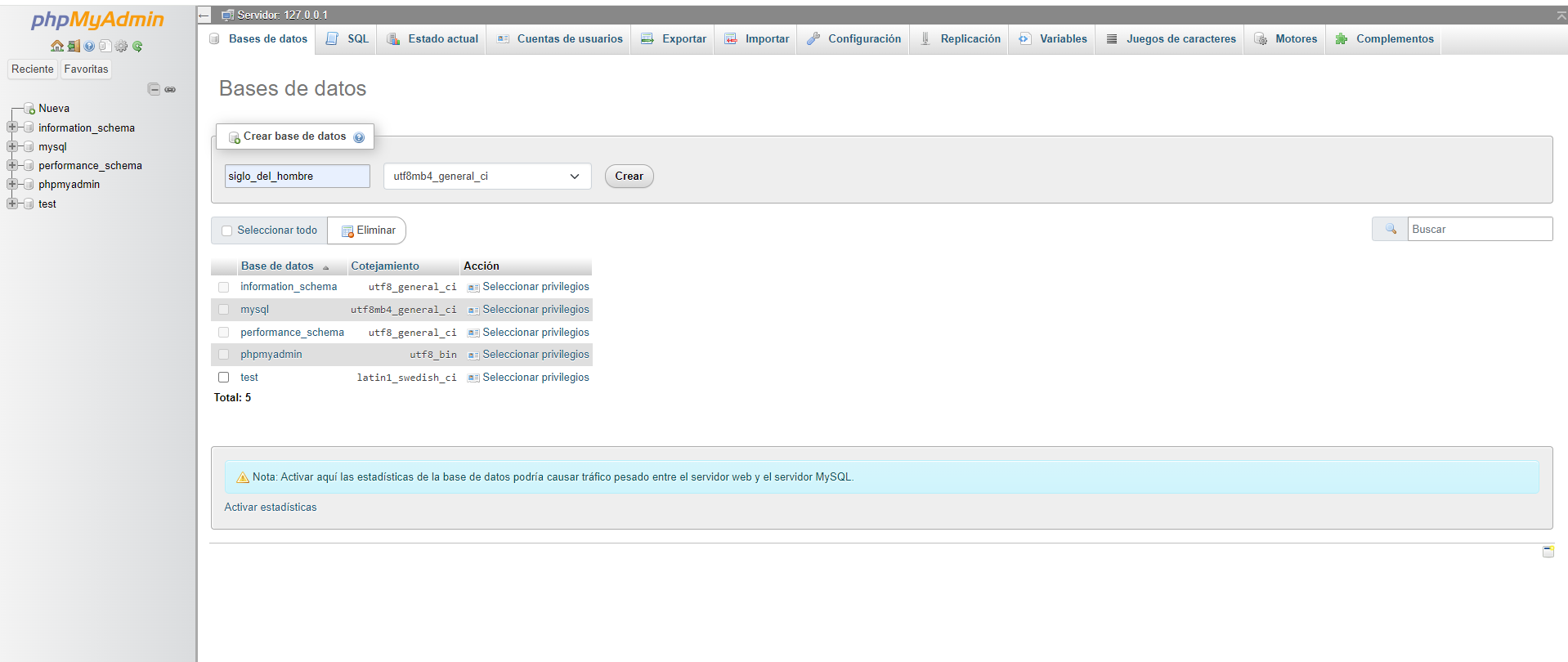
1. **Instalar XAMPP:**
   1. Ejecute el instalador de XAMPP y siga los pasos para completar la instalación.
   2. Durante la configuración, asegúrese de habilitar los módulos **Apache** y **MySQL**.
2. **Preparar los archivos del proyecto:**
   1. Copie la carpeta del proyecto (por ejemplo, *libreria*) en el directorio htdocs ubicado dentro de la carpeta de instalación de XAMPP (por defecto, C:\xampp\htdocs en Windows o /opt/lampp/htdocs en Linux)



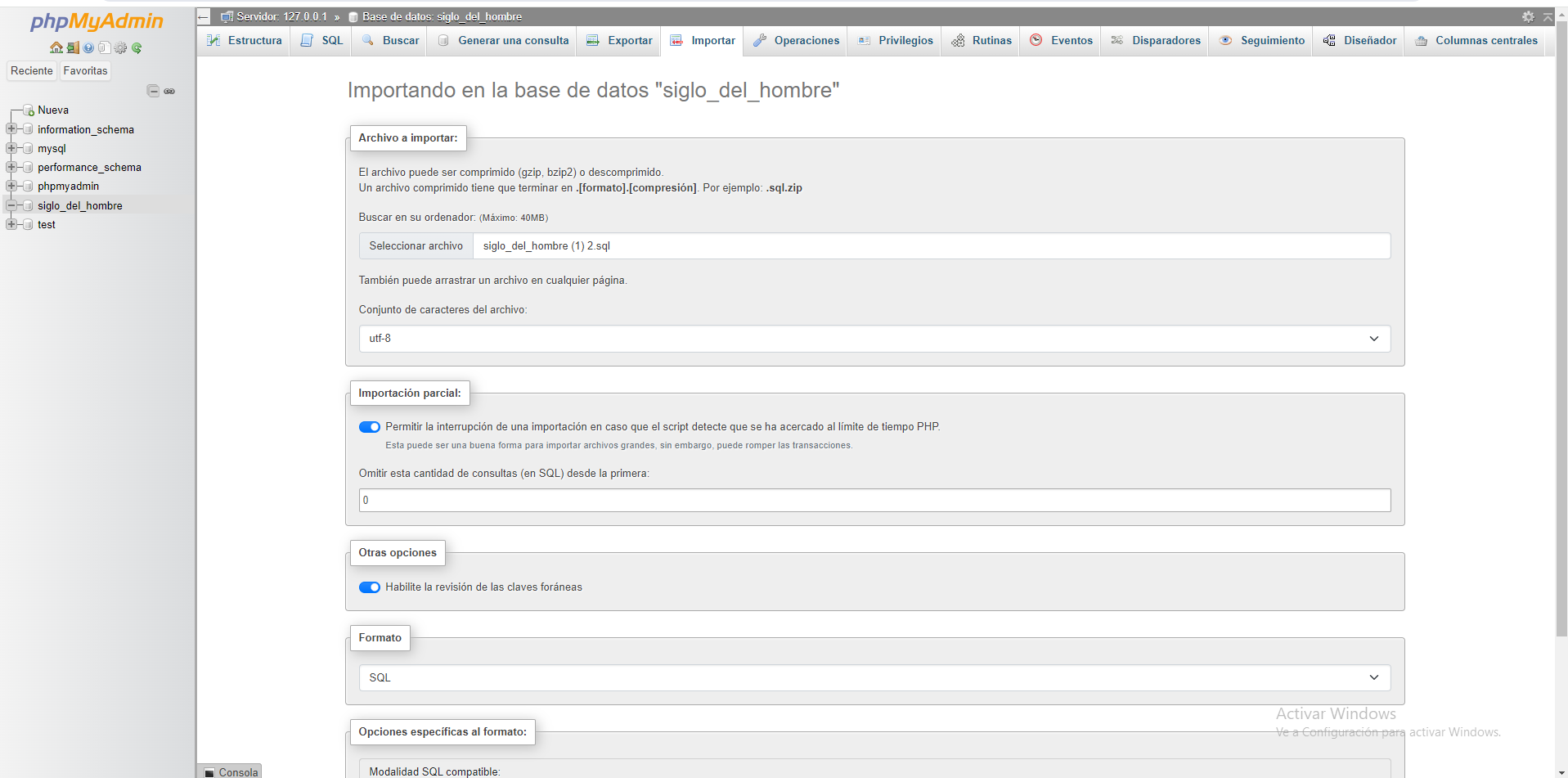
1. **Iniciar el entorno de servidor:**
   1. Abra el **Panel de Control de XAMPP**.
   2. Inicie los módulos **Apache** (servidor web) y **MySQL** (gestor de base de datos). Asegúrese de que ambos servicios estén en estado "Running".



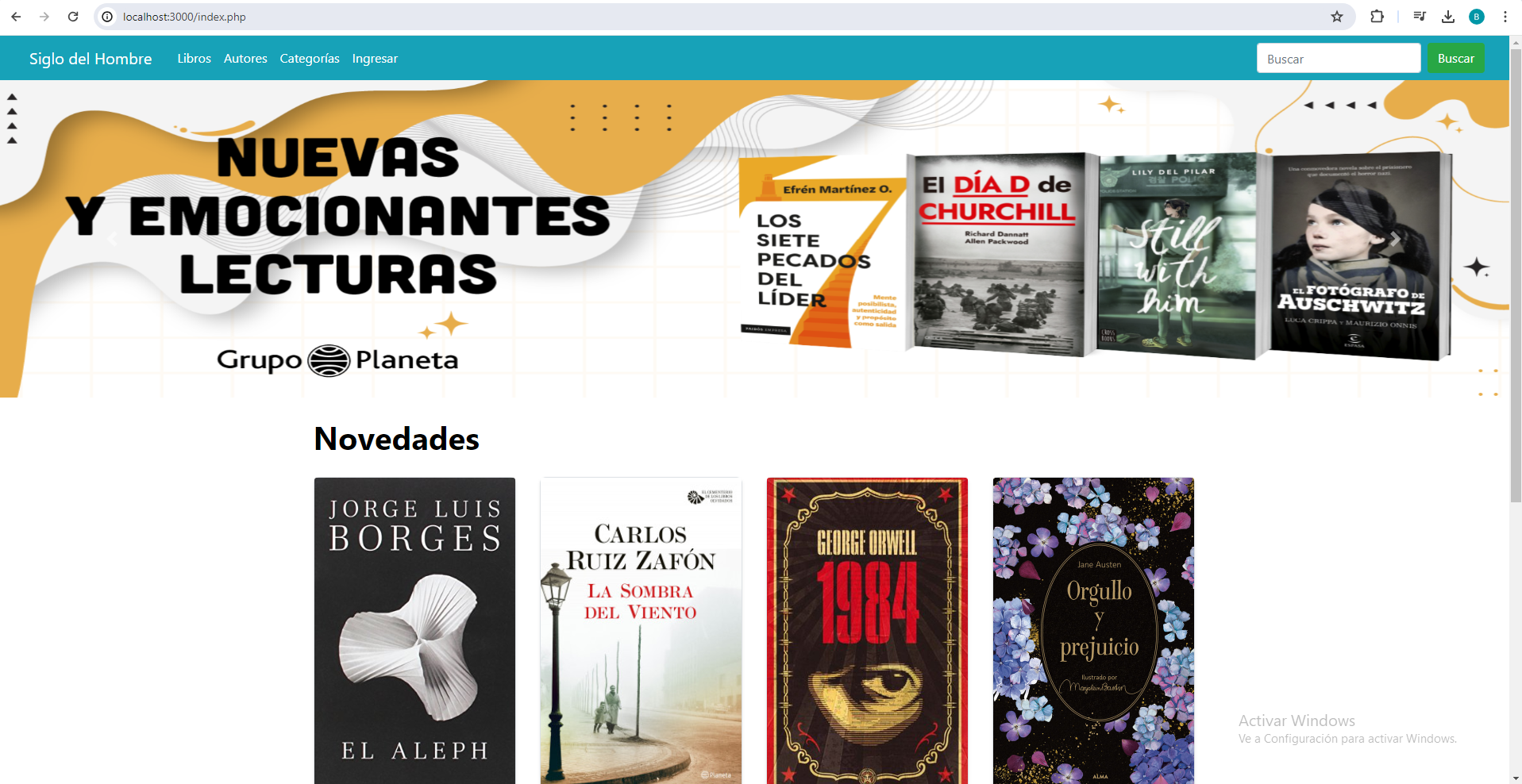
1. **Configurar la base de datos:**
   1. Acceda a **phpMyAdmin** abriendo su navegador y navegando a <http://localhost/phpmyadmin>.
   2. Cree una nueva base de datos llamada libreria (o el nombre especificado en el archivo de configuración del proyecto).



* 1. Importe el archivo SQL proporcionado (*libreria.sql*) para cargar la estructura y datos iniciales necesarios para el funcionamiento del sistema.



1. **Acceso a la aplicación:**
   1. Abra su navegador web y acceda al sistema ingresando la URL: <http://localhost/libreria>.
   2. Verifique que la aplicación se carga correctamente y pruebe las funcionalidades básicas (inicio de sesión, gestión de inventario, etc.).



**Nota:**

* En caso de experimentar errores, verifique los permisos de la carpeta htdocs, la conexión de la base de datos (archivo de configuración de PHP), y que los módulos de XAMPP estén ejecutándose correctamente.

Si el proyecto requiere configuraciones personalizadas, como un archivo .env o ajustes en el archivo config.php, edítelos según las instrucciones proporcionadas en la documentación del proyecto.

**Arquitectura Del Sistema**

El sistema está diseñado bajo un modelo **cliente-servidor**, lo que garantiza una interacción eficiente y modular entre el usuario final y el backend. A continuación, se detallan sus componentes principales y su interacción:

### **Cliente (Frontend):**

* **Navegador web:**

Ejecuta tecnologías como **HTML**, **CSS** y **JavaScript** para presentar la interfaz gráfica al usuario.

Compatible con navegadores modernos como Google Chrome, Mozilla Firefox y Microsoft Edge.

* **Interfaz de usuario:**

Desarrollada con **Bootstrap** para asegurar un diseño adaptable (*responsive*) y funcional en dispositivos de escritorio, tablets y móviles.

Incluye elementos interactivos como botones, formularios y tablas, que mejoran la experiencia de usuario.

* **JavaScript:**

Implementado para mejorar la interactividad, gestionar eventos en el navegador y realizar solicitudes asincrónicas al servidor mediante AJAX, reduciendo la necesidad de recargar páginas.

### **Servidor (Backend):**

* **PHP:**

Procesa la lógica del negocio, como la validación de formularios, la gestión de sesiones y las operaciones relacionadas con usuarios, pedidos e inventarios.

Genera páginas HTML dinámicas según las solicitudes realizadas por los clientes.

* **MySQL:**

Sistema de gestión de bases de datos utilizado para almacenar información estructurada, como:

* + - Datos de usuarios (clientes y administradores).
    - Catálogo de libros y stock del inventario.
    - Historial de pedidos y transacciones.

Diseñado con tablas relacionales para mantener la integridad de los datos y optimizar las consultas.

### **Interacción entre componentes:**

1. El cliente (usuario) utiliza el navegador para interactuar con la aplicación mediante una interfaz gráfica.
2. Cuando el usuario realiza una acción (por ejemplo, buscar un libro o procesar un pedido), el navegador envía una **solicitud HTTP** al servidor.
3. El servidor:
   1. Recibe la solicitud, la interpreta con PHP y realiza las operaciones necesarias en la base de datos (si corresponde).
   2. Prepara una **respuesta dinámica** (generalmente en HTML o JSON) y la envía al cliente.
4. El cliente muestra los resultados al usuario en tiempo real, actualizando la interfaz según la información recibida.

### **Diagrama conceptual:**

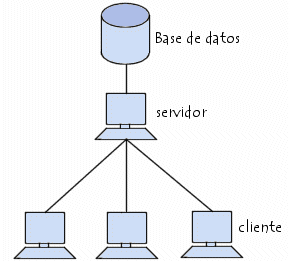
* **Cliente:**

Navegador web -> Interfaz de usuario -> Solicitudes HTTP

* **Servidor:**

Solicitudes HTTP -> PHP (lógica de negocio) -> MySQL (base de datos) -> Respuesta dinámica

Esta arquitectura asegura una clara separación de responsabilidades entre el cliente y el servidor, permitiendo escalabilidad, mantenibilidad y una experiencia fluida para los usuarios finales.

Figura 1

**Imagen 1**. infraestructura sistema de información (tomado de https://www.monografias.com/trabajos77/base-datos-sistema-informacion/base-datos-sistema-informacion)

**Modelos entidad relación**

El modelo entidad-relación representa la estructura de la base de datos utilizada en el sistema de información de la librería Siglo del Hombre. Este diseño asegura una gestión eficiente de los datos relacionados con el inventario, los pedidos, los clientes y otros elementos del sistema.

**Descripción general:**

Usuarios: Los usuarios tienen atributos como su rol (cliente o administrador), correo electrónico y datos de contacto. Cada usuario puede generar múltiples pedidos.

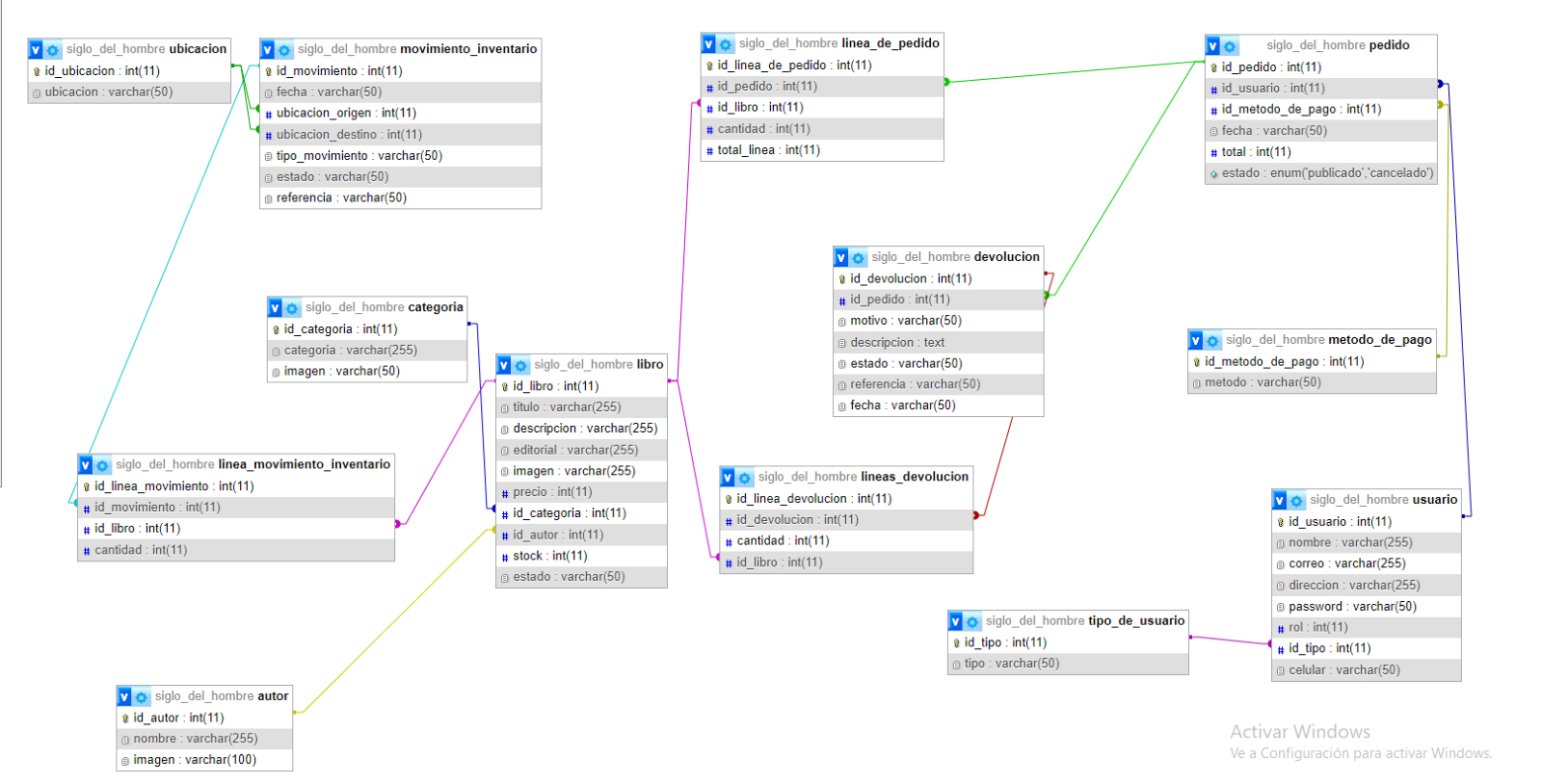
Pedidos: Los pedidos incluyen información como fecha, total, estado y el método de pago seleccionado. Están vinculados a las líneas de pedido que especifican los libros adquiridos.

Libros: Cada libro está clasificado por categorías y está asociado a un autor. Además, incluye detalles como título, descripción, precio y stock.

Inventario: Se gestionan los movimientos del inventario mediante registros de transferencias entre ubicaciones, permitiendo rastrear entradas y salidas de libros.

Devoluciones: Los pedidos pueden tener asociadas devoluciones, las cuales incluyen motivos, descripción y líneas de devolución que detallan los libros involucrados.

Propósito: El modelo entidad-relación permite organizar y conectar todas las entidades principales del sistema, garantizando la integridad de los datos y facilitando el desarrollo de consultas complejas para satisfacer las necesidades del sistema.

  
Figura 2

**Diccionario de datos**

#### **Tabla:** autor

**Descripción**: Almacena información sobre los autores disponibles en el sistema.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre del Campo** | **Tipo de Dato** | **Longitud** | **Nulo** | **Descripción** |
| id\_autor | int | 11 | No | Identificador único del autor. |
| nombre | varchar | 255 | No | Nombre completo del autor. |
| imagen | varchar | 100 | No | Ruta de la imagen representativa del autor. |

#### **Tabla:** categoría.

**Descripción**: Almacena información de las categorías de los libros.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre del Campo** | **Tipo de Dato** | **Longitud** | **Nulo** | **Descripción** |
| id\_categoria | int | 11 | No | Identificador único de la categoría. |
| categoria | varchar | 255 | No | Nombre de la categoría. |
| imagen | varchar | 50 | No | Ruta de la imagen representativa de la categoría. |

#### **Tabla:** devolución

**Descripción**: Registra las devoluciones realizadas por los usuarios.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre del Campo** | **Tipo de Dato** | **Longitud** | **Nulo** | **Descripción** |
| id\_devolucion | int | 11 | No | Identificador único de la devolución. |
| id\_pedido | int | 11 | No | Relación con el pedido devuelto. |
| motivo | varchar | 50 | No | Razón de la devolución. |
| descripcion | text | N/A | No | Detalle de la devolución. |
| estado | varchar | 50 | No | Estado actual de la devolución. |
| referencia | varchar | 50 | No | Referencia para identificar la devolución. |
| fecha | varchar | 50 | No | Fecha en que se realizó la devolución. |

**Tabla:** libro

**Descripción**: Contiene la información de los libros disponibles en el sistema.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre del Campo** | **Tipo de Dato** | **Longitud** | **Nulo** | **Descripción** |
| id\_libro | int | 11 | No | Identificador único del libro. |
| titulo | varchar | 225 | No | Titulo del libro. |
| descripcion | varchar | 225 | No | Resumen o sinopsis del libro. |
| editorial | varchar | 225 | No | Nombre de la editorial. |
| imagen | varchar | 225 | No | |  | | --- | |  |  |  | | --- | | Ruta de la imagen representativa del libro. | |
| precio | int | 11 | No | Precio del libro. |
| id\_categoria | int | 11 | No | Relación con la categoría del libro. |
| id\_autor | int | 11 |  | Relación con el autor del libro. |
| stock | int | 11 |  | Cantidad disponible en inventario. |
| estado | varchar | 50 |  | Estado del libro (e.g., activo o inactivo) |

#### **Tabla:** lineas\_devolucion

**Descripción**: Contiene los detalles específicos de las líneas de las devoluciones realizadas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre del Campo** | **Tipo de Dato** | **Longitud** | **Nulo** | **Descripción** |
| id\_linea\_devolucion | int | 11 | No | Identificador unico de la linea de devolucion. |
| id\_devolucion | int | 11 | No | Relación con la devolución a la que pertenece. |
| cantidad | int | 11 | No | Cantidad de productos devueltos en esta línea. |
| id\_libro | int | 11 | No | Relación con el libro devuelto. |

#### **Tabla:** linea\_de\_pedido

**Descripción**: Almacena los detalles de los productos incluidos en un pedido.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre del Campo** | **Tipo de Dato** | **Longitud** | **Nulo** | **Descripción** |
| id\_linea\_de\_pedido | int | 11 | No | Identificador único de la línea del pedido. |
| id\_pedido | int | 11 | No | Relación con el pedido al que pertenece |
| id\_libro | int | 11 | No | Relación con el libro incluido en el pedido. |
| cantidad | int | 11 | No | Cantidad de libros en esta línea del pedido. |
| total\_linea | int | 11 | No | Total monetario de la línea del pedido. |

#### **Tabla:** linea\_movimiento\_inventario

**Descripción**: Registra los detalles de los movimientos de inventario.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre del Campo** | **Tipo de Dato** | **Longitud** | **Nulo** | **Descripción** |
| id\_linea\_de\_movimiento | int | 11 | No | Identificador único de la línea de movimiento. |
| id\_movimiento | int | 11 | No | Relación con el movimiento de inventario. |
| Id\_libro | int | 11 | No | Relación con el libro involucrado en el movimiento. |
| cantidad | int | 11 | No | Cantidad de libros movidos en esta operación. |

#### **Tabla:** metodo\_de\_pago

**Descripción**: Define los métodos de pago disponibles para los usuarios.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre del Campo** | **Tipo de Dato** | **Longitud** | **Nulo** | **Descripción** |
| id\_metodo\_de\_pago | int | 11 | no | Identificador único del método de pago. |
| metodo | varchar | 50 | no | Nombre del método de pago. |

#### **Tabla:** movimiento\_inventario

**Descripción**: Registra los movimientos de inventario, incluyendo transferencias y ajustes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre del Campo** | **Tipo de Dato** | **Longitud** | **Nulo** | **Descripción** |
| id\_movimiento | int | 11 | No | Identificador único del movimiento. |
| fecha | varchar | 50 | No | Fecha del movimiento de inventario. |
| ubicacion\_origen | int | 11 | No | Relación con la ubicación de origen. |
| ubicacion\_destino | int | 11 | No | Relación con la ubicación de destino. |
| tipo\_movimiento | varchar | 50 | No | Tipo de movimiento (e.g., ingreso, salida). |
| estado | varchar | 50 | No | Estado actual del movimiento. |
| referencia | varchar | 50 | No | |  | | --- | |  |  |  | | --- | | Referencia asociada al movimiento. | |

#### **Tabla:** pedido

**Descripción**: Contiene los pedidos realizados por los usuarios.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre del Campo** | **Tipo de Dato** | **Longitud** | **Nulo** | **Descripción** |
| id\_pedido | int | 11 | No | |  | | --- | | Identificador único del pedido. | |
| id\_usuario | int | 11 | No | Relación con el usuario que realizó el pedido. |
| id\_metodo\_de\_pago | int | 11 | No | |  | | --- | | Relación con el metodo de pago utilizado. | |
| fecha | varchar | 50 | No | Fecha en que se realizó el pedido. |
| total | int | 11 | No | Total del pedido en términos monetarios. |
| estado | enum | N/A | No | Estado del pedido (e.g., publicado, cancelado). |

#### **Tabla:** tipo\_de\_usuario

**Descripción**: Define los tipos de usuarios que existen en el sistema.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre del Campo** | **Tipo de Dato** | **Longitud** | **Nulo** | **Descripción** |
| id\_tipo | int | 11 | No | Identificador único del tipo de usuario. |
| tipo | varchar | 50 | Si | Nombre del tipo de usuario (e.g., admin, cliente). |

#### **Tabla:** ubicación

**Descripción**: Almacena las ubicaciones utilizadas en el sistema.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre del Campo** | **Tipo de Dato** | **Longitud** | **Nulo** | **Descripción** |
| id\_ubicacion | int | 11 | No | Identificador único de la ubicación. |
| ubicacion | varchar | 50 | No | Nombre o descripción de la ubicación. |

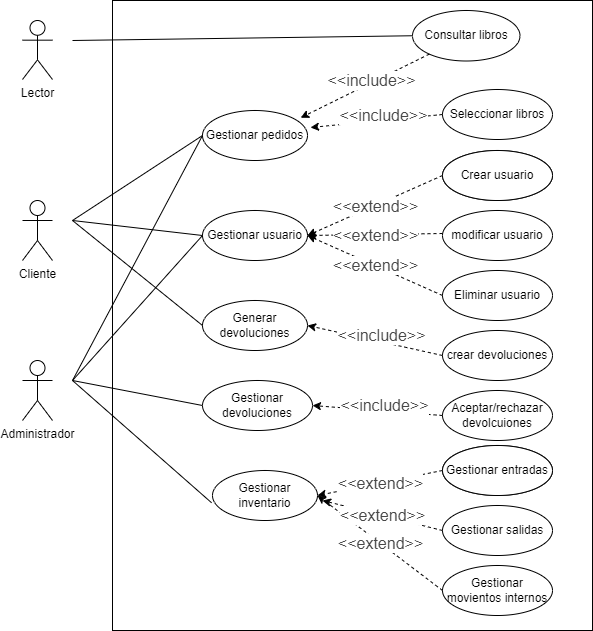
#### **Tabla:** usuario

**Descripción**: Contiene información sobre los usuarios registrados en el sistema.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre del Campo** | **Tipo de Dato** | **Longitud** | **Nulo** | **Descripción** |
| id\_usuario | int | 11 | No | Identificador único del usuario. |
| nombre | varchar | 225 | No | Nombre del usuario. |
| correo | varchar | 225 | No | Dirección de correo electrónico del usuario. |
| direccion | varchar | 225 | Si | Dirección física del usuario. |
| password | varchar | 50 | Si | Contraseña del usuario. |
| rol | int | 11 | No | Relación con el rol asignado al usuario. |
| Id\_tipo | int | 11 | No | Relación con el tipo de usuario. |
| celular | varchar | 50 | No | Número de contacto del usuario. |

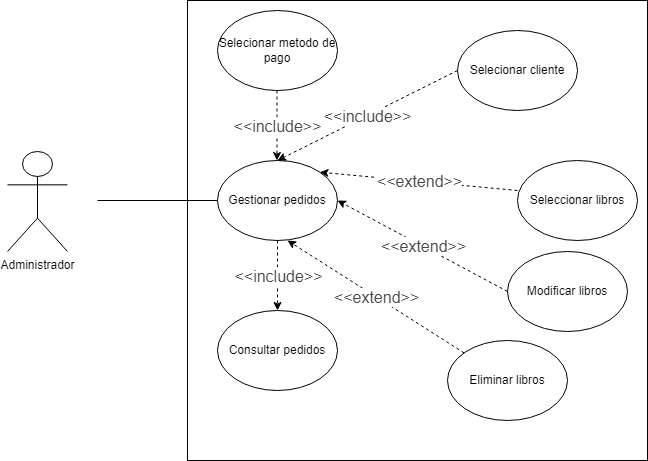
**Diagramas de casos de uso**

**Diagrama de casos de uso General:**

Diagrama 1

**Subcasos Principales**

**Subcaso 1. Gestionar pedidos – cliente:**

Diagrama 2

**Resumen:**  
Un cliente genera un pedido nuevo seleccionando los libros que desea adquirir y la cantidad de cada uno. Luego elige un método de pago, completa la transacción y recibe un ID único de pedido.

**Actores:**

Cliente (actor principal).

Sistema (actor secundario).

Personal involucrado e intereses:

**Librería:** Desea generar ventas y satisfacer las necesidades de sus clientes.

**Precondición:**

El cliente debe estar logueado en el sistema.

**Postcondiciones:**

El pedido se registra en la base de datos.

Se calcula el valor total de la compra.

El inventario se actualiza automáticamente.

**Extensiones (Flujos alternativos)**

El cliente puede agregar libros al carrito y no proceder al pago, guardando la selección para un uso posterior.

**Requisitos de interfaz de usuario**

La interfaz debe ser fácil de navegar, con un diseño intuitivo que guíe al usuario durante el proceso de compra.

Debe incluir un acceso visible al carrito de compras desde cualquier página.

Facilitar la búsqueda y selección de libros mediante opciones de filtrado y categorización.

Permitir al usuario ajustar las cantidades de libros directamente en el carrito.

Mostrar claramente los métodos de pago aceptados con instrucciones claras para completar la transacción.

**Requisitos no funcionales**

**Seguridad de datos**: El sistema debe proteger la información del cliente y garantizar transacciones seguras.

**Escalabilidad**: Diseñar el sistema para manejar un aumento en el número de pedidos y usuarios concurrentes.

**Usabilidad**: Asegurar que la interfaz sea accesible y fácil de usar para usuarios con diferentes niveles de experiencia tecnológica.

**Subcaso 2. Gestionar inventario-Administrador:**

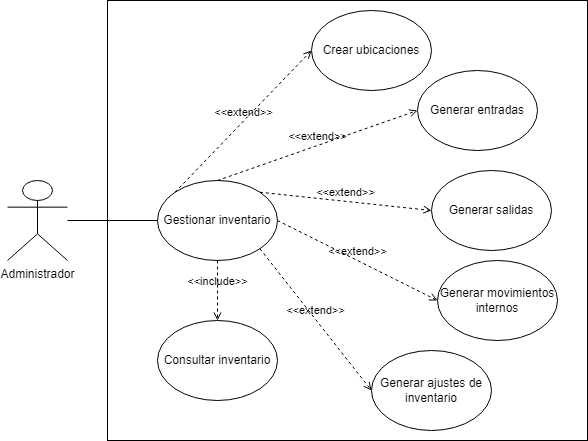


Diagrama 3

**Resumen:**  
El administrador gestiona el inventario de la librería, añadiendo nuevos productos, actualizando sus detalles (como cantidad, precio y descripción), y eliminando productos que ya no están disponibles. También puede recibir notificaciones sobre productos con bajo stock para facilitar la reposición.

**Actores:**

Administrador (actor principal).

Sistema (actor secundario).

Personal involucrado e intereses:

Librería: Desea mantener un inventario preciso y actualizado para satisfacer la demanda de los clientes y optimizar la gestión de stock.

**Precondición:**

El administrador debe iniciar sesión en el sistema.

**Postcondiciones:**

Los cambios realizados en el inventario (añadir, actualizar, eliminar) se reflejan en tiempo real para los clientes y en la base de datos.

Los productos con bajo stock generan alertas para reposición.

**Extensiones (Flujos alternativos)**

El administrador puede buscar productos específicos en el inventario utilizando filtros por nombre, categoría o autor.

El sistema puede alertar automáticamente sobre productos con stock bajo antes de su agotamiento.

**Requisitos de interfaz de usuario**

La interfaz debe incluir una tabla con el listado de productos del inventario, mostrando información como nombre, cantidad, precio y categoría.

El administrador debe tener opciones claras para agregar, editar o eliminar productos desde la interfaz.

Debe permitir búsquedas rápidas y filtrado de productos para facilitar la gestión.

Alertas visuales o notificaciones para productos con bajo stock.

**Requisitos no funcionales**

Seguridad de datos: Garantizar que solo los administradores autorizados puedan acceder y modificar el inventario.

Escalabilidad: El sistema debe ser capaz de manejar un inventario grande sin perder rendimiento.

Usabilidad: Diseñar una interfaz intuitiva que facilite la gestión del inventario incluso a administradores con poca experiencia técnica.

**Subcaso 2. Gestionar pedidos-Administrador:**

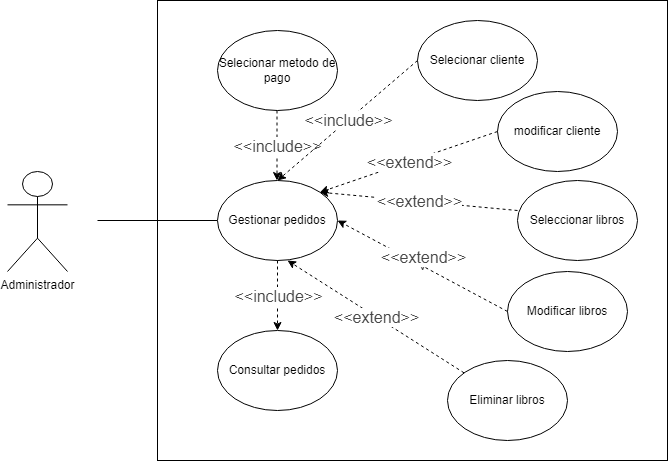


Diagrama 4

**Resumen:**

El administrador visualiza los pedidos generados por los clientes, verifica sus detalles, actualiza el estado (pendiente, procesado, enviado, entregado) y, si es necesario, se comunica con el cliente.

**Actores:**

Administrador (actor principal).

Sistema (actor secundario)

**Precondición:**

El administrador debe estar logueado en el sistema.

**Postcondiciones:**

El estado del pedido se actualiza y el cliente es notificado.

**Extensiones:**

Cancelar un pedido y notificar al cliente.

Modificar detalles del pedido antes de procesarlo.

**Requisitos de interfaz de usuario:**

Acceso rápido a la lista de pedidos desde el panel principal.

Opción para actualizar el estado de los pedidos con un solo clic.

Visualización clara de detalles como productos solicitados, estado actual y datos del cliente.

**Metodología de desarrollo**

* Para el desarrollo del proyecto, se adoptó una metodología de trabajo en etapas organizada de acuerdo con los tiempos y objetivos establecidos para un proyecto de grado del SENA. El proceso está diseñado para cumplir con las fechas de entrega estipuladas, asegurando la calidad del producto final.

**Fases del proceso de desarrollo**

1. **Planificación:**
   1. Se llevó a cabo una reunión inicial para identificar las necesidades de la librería, enfocándose en las funcionalidades de ventas en línea y gestión de inventario.
   2. Se elaboró un cronograma detallado que incluyó actividades específicas y fechas clave, con el objetivo de completar el proyecto para diciembre.
2. **Diseño:**
   1. Creación de prototipos de baja fidelidad para definir la estructura visual y funcional de la aplicación.
   2. Desarrollo de diagramas como el de casos de uso, de base de datos y de flujo, para documentar los procesos del sistema.
   3. Validación del diseño por parte de los tutores y stakeholders antes de proceder al desarrollo.
3. **Desarrollo:**
   1. Implementación de las funcionalidades del sistema por módulos, comenzando con el catálogo de libros, el carrito de compras y la funcionalidad de inventario.
   2. Integración progresiva de las tecnologías mencionadas (PHP, Bootstrap, JavaScript, etc.) para construir un sistema funcional y amigable para el usuario.
   3. Trabajo en un entorno de pruebas local (XAMPP) para garantizar la estabilidad del sistema.
4. **Pruebas:**
   1. Se ejecutaron pruebas funcionales para garantizar que las características principales, como las compras en línea y el control de inventario, funcionen correctamente.
   2. Realización de pruebas de compatibilidad en diferentes navegadores y dispositivos para verificar la experiencia del usuario en entornos variados.
   3. Validación del sistema frente a los requerimientos establecidos, asegurando que cumple con los estándares de calidad requeridos por el SENA.
5. **Documentación y entrega:**
   1. Elaboración del manual técnico con la descripción detallada de la configuración, el uso y el mantenimiento del sistema.
   2. Preparación de la presentación final para el jurado del proyecto de grado, incluyendo las demostraciones funcionales y la documentación del proceso de desarrollo.
   3. Entrega del proyecto completo en diciembre, cumpliendo con los requisitos académicos y las expectativas del cliente.

Esta metodología permitió mantener un enfoque estructurado y eficiente, asegurando que se cumplieran los plazos y objetivos del proyecto sin comprometer la calidad del sistema desarrollado.

**Manual De Usuario**

Este manual proporciona una guía detallada para acceder y utilizar las funcionalidades principales del sistema de ventas en línea y gestión de inventario de la librería *El Siglo del Hombre*.

### **Acceso al sistema:**

1. **Abrir el navegador web:**
   1. Utilice un navegador compatible como Google Chrome, Mozilla Firefox o Microsoft Edge.
   2. Ingrese la URL en la barra de direcciones: <http://localhost/libreria>.
2. **Navegación por las secciones principales:**
   1. **Inicio:**
      1. Página principal con recomendaciones de libros destacados, promociones y novedades.
   2. **Buscar libros:**
      1. Barra de búsqueda para encontrar libros por título, autor o categoría.
   3. **Carrito de compras:**
      1. Muestra los productos seleccionados para revisar antes de realizar el pedido.
   4. **Perfil de usuario:**
      1. Sección donde puede consultar y actualizar sus datos personales, además de revisar el historial de pedidos.

### **Funciones principales:**

1. **Registro de usuario:**
   1. Haga clic en "Registrar" en la parte superior derecha de la pantalla.
   2. Complete el formulario con sus datos (nombre, correo electrónico y contraseña).
   3. Inicie sesión una vez registrado para acceder a las funcionalidades del sistema.
2. **Compra de libros:**
   1. **Seleccionar libros:** Navegue por las categorías o use la barra de búsqueda para encontrar los títulos deseados.
   2. **Añadir al carrito:** Haga clic en el botón correspondiente en la página del libro.
   3. **Revisar carrito:** Ajuste cantidades, elimine productos o proceda al pago desde esta sección.
   4. **Realizar pedido:** Complete los datos de entrega y confirme su compra.

### **Gestión de inventario: *(para administradores)***

Los administradores tienen acceso a funcionalidades avanzadas para gestionar inventario, usuarios y pedidos.

1. **Inventario de libros:**
   1. **Agregar libros:** Complete el formulario con los detalles del libro (título, autor, precio, cantidad en stock, etc.).
   2. **Editar libros:** Actualice datos como precio, cantidad o descripción.
   3. **Eliminar libros:** Retire libros del inventario cuando ya no estén disponibles para la venta.
   4. **Entradas y salidas:**
      1. **Entradas:** Registre nuevos ingresos al inventario por compras a proveedores.
      2. **Salidas:** Controle las reducciones del stock por ventas o transferencias internas.
2. **Gestión de devoluciones:**
   1. Registre las devoluciones de los clientes, incluyendo el motivo y los detalles del libro devuelto.
   2. Actualice el inventario para reflejar las devoluciones.
3. **Gestión de usuarios:**
   1. **Ver y editar usuarios:** Acceda a un listado de todos los usuarios registrados.
   2. Edite información como nombre, correo electrónico, rol de acceso (usuario o administrador) y estado (activo o inactivo).
   3. **Eliminar usuarios:** Retire usuarios de la base de datos si ya no necesitan acceso al sistema.
   4. **Crear usuarios:**
      1. Los administradores pueden crear nuevos usuarios manualmente desde el panel de control.
      2. Complete los campos necesarios (nombre, correo electrónico, contraseña y rol) para generar un nuevo usuario en el sistema.
4. **Gestión de pedidos:**
   1. **Ver y editar pedidos:** Consulte todos los pedidos realizados, incluyendo detalles como libros comprados, cantidades y estado del pedido.
   2. Actualice el estado de los pedidos (pendiente, procesado, enviado, entregado, cancelado).
   3. **Crear pedidos manualmente:** Registre pedidos desde el panel de administración para clientes que hayan realizado compras por otros medios (por ejemplo, en tienda física o por teléfono).

**Consejos adicionales:**

* Mantenga los datos de inventario y usuarios actualizados para evitar errores en la operación diaria.
* Verifique regularmente los pedidos pendientes para asegurar un procesamiento rápido y eficiente.
* Asegúrese de documentar correctamente las devoluciones, entradas y salidas para mantener un control preciso del inventario.

Este manual busca facilitar tanto el uso del sistema para los clientes como las tareas de gestión para los administradores.

**Mantenimiento Del Sistema**

El mantenimiento adecuado del sistema es crucial para asegurar su funcionamiento óptimo y la seguridad de los datos. A continuación, se detallan las recomendaciones de mantenimiento que deben seguirse de manera regular.

**Recomendaciones de Mantenimiento:**

1. **Base de datos:**
   1. **Respaldo regular:** Realice un respaldo completo de la base de datos al menos una vez por semana a través de **phpMyAdmin**. Esto garantiza la integridad de la información y la capacidad de restaurar los datos en caso de falla del sistema.
   2. **Verificación de integridad:** Realice una revisión periódica de la integridad de las tablas y registros, utilizando las herramientas de diagnóstico de **MySQL** para evitar errores o corrupciones en la base de datos.
2. **Archivos del sistema:**
   1. **Copia de seguridad de archivos:** Mantenga una copia de seguridad actualizada de los archivos ubicados en la carpeta htdocs (donde reside la aplicación). Realice copias de seguridad automáticas o manuales, según lo permita el entorno de desarrollo, para evitar pérdidas de datos o archivos importantes del sistema.
   2. **Verificación de dependencias:** Asegúrese de que todas las dependencias del proyecto (como bibliotecas y frameworks) estén correctamente almacenadas y respaldadas, para evitar inconvenientes con actualizaciones o cambios imprevistos.
3. **Actualización del software:**
   1. **Actualizaciones de XAMPP:** Verifique regularmente las actualizaciones de XAMPP (Apache, MySQL, PHP) para asegurarse de que el entorno de desarrollo y producción esté actualizado con las últimas mejoras y correcciones de seguridad.
   2. **PHP y Bootstrap:** Revise las actualizaciones de **PHP** y **Bootstrap** para asegurarse de que el sistema esté utilizando las versiones más recientes, compatibles con la funcionalidad y rendimiento óptimo de la aplicación.
4. **Seguridad:**
   1. **Revisión de permisos de archivos y carpetas:** Revise periódicamente las configuraciones de **permisos** en las carpetas sensibles del sistema, especialmente aquellas que contienen archivos de configuración y datos de usuarios. Limite el acceso a los archivos críticos para evitar modificaciones no autorizadas.
   2. **Actualización de contraseñas y accesos:** Cambie las contraseñas de acceso a sistemas administrativos (como phpMyAdmin) de manera periódica y utilice métodos de autenticación robustos.
   3. **Monitoreo de actividad:** Implemente un sistema de monitoreo para detectar actividades sospechosas o intentos de acceso no autorizado al sistema, y realice auditorías de seguridad regulares.

**Recomendaciones adicionales:**

* **Documentación continua:** Mantenga la documentación del sistema actualizada, especialmente en lo que respecta a las configuraciones del servidor, base de datos y archivos de configuración.
* **Pruebas de recuperación:** Realice pruebas periódicas de recuperación de datos y restauración del sistema a partir de los respaldos para garantizar que pueda recuperarse correctamente en caso de fallos.