

# MazacuataStore

03/07/20204

# Equipo:

- Barros Martínez Luis Enrique
- Bautista Rios Alfredo
- García Martínez Angel Yahel
- Hernández Vázquez Johan Cristopher
- Ruiz Martínez Erick
- Fernández Caudillo Luis Mario

Repositorio: <a href="https://github.com/LArkadia/-MazacuataStore-">https://github.com/LArkadia/-MazacuataStore-</a>

Visión general	2
Partes interesadas	2
Alcance del sistema	3
Objetivos	3
Exclusiones	4
Suposiciones	5
Especificaciones	6
Requerimientos	6
Arquitectura del sistema	7
Flujo de navegación:	7
Diseño de la base de datos	11
Relaciones entre Tablas	13
Prototipos de vistas	14
1. Bienvenida	14
2. Inicio de sesión	15
3. Registro de usuario	15
4. Carrito	16
5. Consulta de información de un libro	17
Consulta de comprobantes de compra	17
7. Consulta de libros	18
Agregar productos al carrito	19
Resultados finales	20
Frameworks y Páginas	20
Frameworks	20
Componentes React	20
Páginas Detalladas	21
Implementaciones	22
Vistas	28
1. Inicio de sesión	28
2. Pantalla de Bienvenida	29
3. Descripción de libro	30
4. Carrito de compras y "Buscar y agregar"	30
5. Buscador de un libro/Terminales de usuario en tienda	31
6. Ventana de Pago	31
7. Confirmación del pedido	32
8. Comprobante de compra	32

# Visión general

Desarrollar un sistema para la facilitar y optimizar la gestión del inventario de la librería "LibroNetStore" a través de un portal web, así como, la venta de libros en sucursal física a través de puntos de ventas que permitan el chequeo de precios, así como, permitir al cliente verificar las existencias en tienda, por otra parte, el sistema facilitará a los empleados la gestión del inventario y las ventas.

## Partes interesadas

#### Dueño:

El dueño tiene control total sobre el sistema y es responsable de la supervisión del desempeño de todos los empleados de la tienda. Además, gestiona el alta y baja de usuarios en el sistema.

#### • Vendedores o "Caja":

Los vendedores, que operan en turnos, manejan las cajas registradoras del portal de ventas. Están a cargo de procesar todas las ventas, manejar las transacciones de efectivo y otros pagos, y mantener un registro preciso del dinero en caja. Se prevé la utilización de dos usuarios para esta función.

#### • Usuarios Finales o Clientes:

Los clientes pueden interactuar con el sistema a través de una interfaz de usuario online. Ellos tienen cuentas personales que les permiten realizar pedidos, revisar el historial de compras, recibir recomendaciones personalizadas y gestionar múltiples direcciones de envío. Además, pueden acceder a sus facturas y recibos electrónicos, incluyendo detalles de cada transacción realizada.

# Alcance del sistema

# Objetivos

- 1. Automatización del proceso de ventas y gestión del Inventario.
  - a. Se realizará la automatización de la generación del comprobante de venta.
  - b. Se realizará la gestión de los siguientes datos de los libros:
    - i. ISBN
    - ii. Título.
    - iii. Autor
    - iv. Editorial.
    - v. Edición.
    - vi. Precio.
    - vii. Portada (URL de la imagen).
    - viii. Stock disponible.
    - ix. Ubicación física en librería.
    - x. Género.
  - c. Se realizará la gestión de los siguientes datos de las transacciones realizadas:
    - i. Ventas por vendedor
      - 1. ID de venta
      - 2. ID de usuario
      - 3. ISBN
      - 4. Fecha de venta.
      - 5. Cantidad de libros.
      - 6. Cantidad monetaria.
    - ii. Compras por cliente.
      - 1. ID de compra.
      - 2. ID de usuario.
      - 3. Fecha de compra.
      - 4. ISBN.
      - 5. Cantidad de libros.
      - 6. Cantidad monetaria.
      - 7. Estado de la compra.
- 2. Accesibilidad multiplataforma.

- a. El sistema está contemplado para poder ser accedido de manera remota a través de la web para los usuarios finales o "clientes" a través de diferentes plataformas. Las plataformas contempladas son:
  - i. Ordenadores de sobremesa.
  - ii. Ordenadores portátiles.
- 3. Gestión de los datos del usuario.
  - a. El sistema almacenará y gestionará los siguientes datos del usuario:
    - i. Identidad.
      - 1. Nombre de pila.
      - 2. Nombre familiar (Apellidos).
      - 3. RFC.
    - ii. Dirección.

## **Exclusiones**

- 1. Venta de productos no relacionados con libros físicos.
  - a. El cliente no contempla la venta de productos ajenos a los libros físicos.
- 2. Devoluciones de productos.
  - a. El cliente no contempla permitir las devoluciones.
- 3. Aplicación móvil.
  - a. El cliente no requiere de una aplicación móvil dedicada.
- Gestión de datos.
  - a. El sistema no realizará el almacenamiento de los datos de las tarjetas utilizadas para realizar las compras en línea o en tienda física
  - b. El sistema no almacenará más que una sola dirección por cada usuario
  - c. El sistema no realizará la gestión ni generación de datos estadísticos tales como patrones de compra, frecuencia de compra o de cualquier otro tipo.
- 5. Integración con comercios externos.
  - a. El cliente no requiere la integración de proveedores externos en el portal de ventas.

## 6. Expansión.

a. El sistema no está pensado para expandirse a más sucursales.

## Suposiciones

## 1. Presupuesto:

a. A pesar de contar con un presupuesto inicial de \$10,000.00 MXN, se contempla la posibilidad de aumentar este límite.

## 2. Recursos:

a. Se presupone la disponibilidad de adquisición de los medios necesarios para la implementación del sistema.

#### 3. Permisos:

a. El administrador será el encargado de realizar las altas manualmente de su plantilla de empleados

# Especificaciones

## Requerimientos

## 1. Requerimientos Funcionales

### a. Registro de Ventas:

El sistema debe permitir el registro detallado de todas las ventas realizadas, incluyendo información sobre los productos vendidos, fecha, hora y vendedor asociado.

#### b. Gestión del Inventario:

Se requiere la capacidad de gestionar el inventario de libros físicos.

#### c. Automatización de Procesos:

El sistema debe automatizar procesos específicos de la librería, como la búsqueda de libros para los empleados o los clientes.

#### d. Administración de datos:

El sistema deberá ser capaz de almacenar de forma segura las contraseñas de acceso de los usuarios, así como sus datos personales.

#### e. Gestión de Usuarios:

El sistema deberá ser capaz de almacenar de forma segura las contraseñas de acceso de los usuarios, así como sus datos personales.

#### f. Administración de datos:

**g.** El sistema deberá ser capaz de almacenar de forma segura las contraseñas de acceso de los usuarios, así como sus datos personales.

## h. Gestión de Productos

Se requiere una api para poder consultar los tipos de productos que estamos vendiendo en este caso

#### i. Gestión de Secciones

Se requiere gestionar los accesos para cada tipo de usuario y que este no se meta a una parte en la cual no tenga privilegios, con esto evitar la pérdida de datos y la protección de estos mismos.

## j. Gestión de Compra

Se requiere una sección en la cual podamos controlar que se vende, si esta compra está activa, o cancelada o si ya se completó de esta forma saber cuanto nos tendrá que pagar un cliente, para de esta forma aplicar un método de pago.

#### k. Gestión de tokens

Gestionar los tokens, para la autentificación de los usuarios, para ello se re quiere guardarlos en algún lugar

#### I. Versiones de la base de datos

Se requiere gestionar las versiones de la base de datos y hacer un backup periódicamente para evitar la pérdida de los datos, en las actualizaciones o en las implementaciones de algún bloque nuevo dentro del sistema

## 2. Requerimientos No funcionales

#### a. Diseño de Interfaz:

Se desea una interfaz de usuario sencilla, moderna y minimalista tanto para el punto de venta como para el portal web.

#### b. Seguridad y Permisos:

Se requiere una gestión de usuarios segura, con diferentes roles y permisos para garantizar la privacidad y seguridad de la información.

#### c. Compatibilidad Plataforma:

El sistema debe ser compatible con diversas plataformas, incluyendo Windows, Linux y web, para adaptarse a las necesidades del cliente.

#### d. Gestor de Versiones:

Gestión de las diferentes versiones del software, esto dará seguridad a la hora de hacer cambios, ya que se puede regresar fácilmente a una versión anterior.

## e. Código Reutilizable:

El código debe de estar diseñado de tal forma que se pueda reutilizar la mayor parte de este, para evitar la pérdida del tiempo en los desarrollos de esto, con lo cual facilitar la tarea para los nuevos desarrolladores que se integren dentro del proyecto o por su parte para los que ya llevan más rato, para no hacer código redundante.

#### f. Ideales de Código Limpio:

Se requiere ideales de código limpio, de tal forma que cuando se observa alguna funcionalidad esta pueda ser modificada o en su defecto utilizarla, pero también eliminada si esto es necesario.

# Arquitectura del sistema

Flujo de navegación:

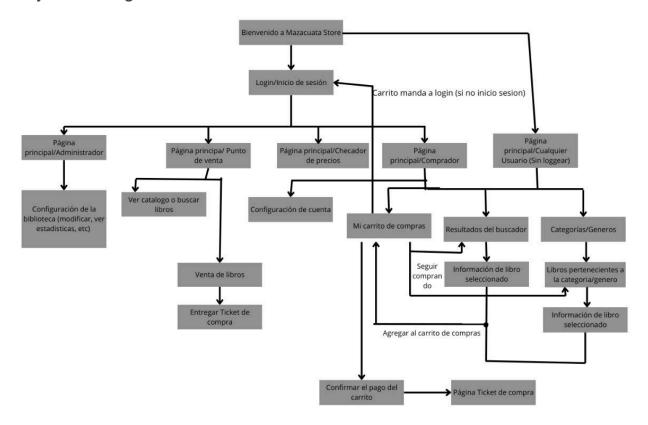


Figura 1. Diagrama de flujo del sitio Web.

El diagrama de flujo de navegación representa la estructura y las relaciones entre las diferentes páginas y funciones del sistema, tanto para usuarios registrados como no registrados.

## 1. Página de Bienvenida

 Descripción: La página de inicio que da la bienvenida a los usuarios de LibroNetStore.

#### Conexiones:

■ Dirige a la página de login/inicio de sesión.

## 2. Login/Inicio de Sesión

o **Descripción:** Página donde los usuarios pueden iniciar sesión o registrarse.

#### Conexiones:

- Después del login, los usuarios pueden acceder a:
  - Página principal/Administrador
  - Página principal/Punto de venta
  - Página principal/Checador de precios
  - Página principal/Comprador

## 3. Página Principal

 Descripción: Página principal de venta, es la primera vista que el usuario tiene sobre el inventario y los libros en venta; también se muestran los libros destacados, ofertas ,etc....

#### Conexiones:

- Ver catálogo o buscar libros
- Venta de libros

#### 4. Checador de Precios

 Descripción: Página utilizada por las terminales para revisar el precio, disponibilidad y ubicación de los libros.

#### 5. Página Datos del Comprador

 Descripción: Página con los datos del usuario, aquí puede revisar sus comprobantes de compras y libros adquiridos.

#### Conexiones:

- Ver catálogo o buscar libros
- Mi carrito de compras
- Resultados del buscador
- Información de libro seleccionado
- Categorías/Géneros

## 6. Página Principal/Cualquier Usuario (Sin loguear)

• **Descripción:** Página inicial accesible para cualquier usuario no registrado.

## Conexiones:

- Resultados del buscador de libros
- Categorías/Géneros
- Carrito de compras

## 7. Configuración de la Biblioteca

 Descripción: Permite a los administradores modificar la biblioteca, ver estadísticas y realizar otras configuraciones, (CRUD).

## 8. Catálogo y Búsqueda de Libros

- Descripción: Página donde los usuarios pueden ver el catálogo de libros o realizar búsquedas dentro de este y ver los resultados arrojados por la búsqueda.
- o Conexiones: Resultados del buscador.

## 9. Mi Carrito de Compras

 Descripción: Página que muestra los productos agregados al carrito, (libros y chucherias).

#### Conexiones:

- Confirmar el pago del carrito
- Continuar comprando

#### 10. Resultados del Buscador

Descripción: Página que muestra los resultados de las búsquedas de libros.

#### Conexiones:

- Información de libro seleccionado
- Agregar al carrito de compras

#### 11. Géneros

Descripción: Página que muestra los géneros de libros disponibles.

#### Conexiones:

■ Libros pertenecientes al género.

#### 12. Información de Libro Seleccionado

Descripción: Página que muestra la información detallada de un libro seleccionado tales como su sinopsis, precio y portada.

## Conexiones:

- Agregar al carrito de compras
- Seguir comprando

## 13. Confirmar el Pago del Carrito

 Descripción: Página para confirmar y realizar el pago de los libros en el carrito, es en esta página donde se pueden ingresar los datos bancarios y fiscales (RFC) para realizar la compra.

#### Conexiones:

■ Página de Ticket de Compra

## 14. Página de comprobante de Compra

 Descripción: Se muestra el comprobante de compra generado después de la transacción.

## Diseño de la base de datos

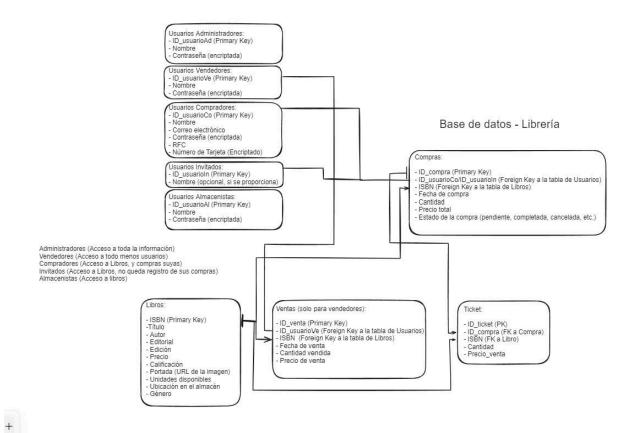


Figura 2. Diagrama de diseño de la Base de datos.

El diseño de la base de datos para "LibroNetStore" está estructurado para manejar eficientemente la información relacionada con los diferentes tipos de usuarios, los libros disponibles, las compras realizadas, las ventas procesadas por los vendedores y la generación de tickets. La base de datos está organizada en varias tablas que representan entidades clave del sistema, con atributos específicos y relaciones bien definidas entre ellas para asegurar la integridad y consistencia de los datos. A continuación, se detallan cada una de las tablas y sus respectivas relaciones:

#### 1. Usuarios Administradores

- o Atributos:
  - ID\_usuarioAd (Primary Key)
  - Nombre
  - Contraseña (encriptada)
- Acceso: A toda la información del sistema.

#### 2. Usuarios Vendedores

- Atributos:
  - ID\_usuarioVe (Primary Key)
  - Nombre
  - Contraseña (encriptada)
- Acceso: A toda la información menos la de los usuarios.

#### 3. Usuarios Compradores

- Atributos:
  - ID\_usuarioCo (Primary Key)
  - Nombre
  - Correo electrónico
  - Contraseña (encriptada)
  - RFC
  - Número de Tarjeta (encriptado)
- Acceso: A los libros y a sus propias compras.

#### 4. Usuarios Invitados

- Atributos:
  - ID\_usuarioIn (Primary Key)
  - Nombre (opcional, si se proporciona)
- Acceso: A los libros, sin registro de sus compras.

## 5. Usuarios Almacenistas

- Atributos:
  - ID\_usuarioAl (Primary Key)
  - Nombre
  - Contraseña (encriptada)
- Acceso: A los libros.

### 6. Libros

- Atributos:
  - ISBN (Primary Key)
  - Título
  - Autor
  - Editorial
  - Edición
  - Precio
  - Cantidad
  - Portada (URL de la imagen)
  - Unidades disponibles
  - Ubicación en el almacén
  - Género

### 7. Compras

- Atributos:
  - ID\_compra (Primary Key)
  - ID\_usuarioCo/ID\_usuarioIn (Foreign Key a la tabla de Usuarios)
  - ISBN (Foreign Key a la tabla de Libros)
  - Fecha de compra
  - Cantidad
  - Precio total
  - Estado de la compra (pendiente, completada, cancelada, etc.)

#### 8. Ventas

- Atributos:
  - ID\_venta (Primary Key)
  - ID\_usuarioVe (Foreign Key a la tabla de Usuarios)
  - ISBN (Foreign Key a la tabla de Libros)
  - Fecha de venta
  - Cantidad vendida
  - Precio de venta
- Acceso: Solo para vendedores.

#### 9. Tickets

- Atributos:
  - ID\_ticket (Primary Key)
  - ID\_compra (Foreign Key a Compras)
  - ISBN (Foreign Key a Libros)
  - Cantidad
  - Precio\_venta

## Relaciones entre Tablas

- Usuarios Administradores, Vendedores, Compradores, Invitados y Almacenistas están relacionados mediante sus ID únicos.
- La tabla **Libros** contiene información detallada sobre cada libro, y está relacionada con las tablas **Compras**, **Ventas** y **Tickets** a través del ISBN.
- La tabla **Compras** conecta a usuarios compradores e invitados con los libros que adquieren, y se vincula con la tabla **Tickets** para generar registros de transacciones.
- La tabla **Ventas** específica para los vendedores, registra las ventas realizadas, incluyendo información del usuario vendedor y el libro vendido.

## Prototipos de vistas

En esta sección se presentan los diseños de cada página web de "Mazacuatastore". Estos diseños están destinados a proporcionar una interfaz de usuario intuitiva y eficiente, que facilite la navegación y la realización de tareas específicas dentro del sistema. Cada vista está diseñada con el usuario final en mente, asegurando que tanto administradores, vendedores, compradores, invitados y almacenistas puedan interactuar con el sistema de manera efectiva y sin complicaciones. A continuación, se detallan los diferentes diseños de las vistas, especificando los elementos visuales y funcionales que cada una incluye, así como su propósito dentro del flujo general de la aplicación.

## 1. Bienvenida

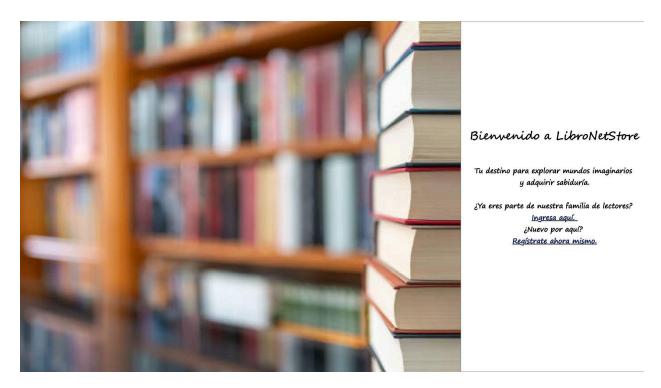


Figura 3. Vista de bienvenida.

## 2. Inicio de sesión



Figura 4.

# 3. Registro de usuario

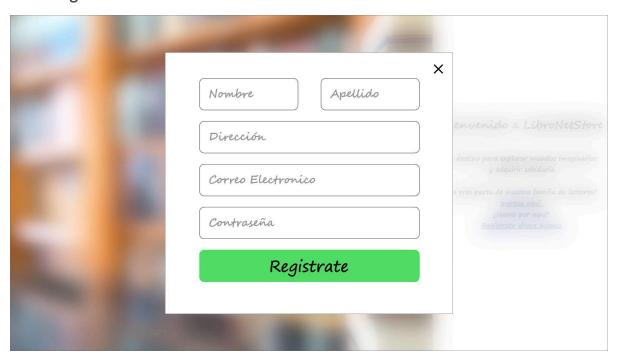


Figura 5.

## 4. Carrito

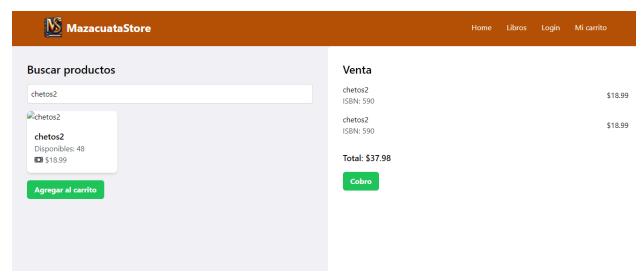


Figura 6.

5. Consulta de información de un libro

```
GET ▼ localhost:4000/api/books/single/chetos

Send ▼ 200 OK 136 ms 279 B 15 Days Ago ▼

Preview ▼ Headers 10 Cookies Timeline Mo

JSON ▼

1 ...

1 ...

1 ...

1 ...

2 "error": false,
3 "status": 200,
4 "body": [
5 * {
6 "ishn": "590",
7 "titulo": "Chetos",
8 "autor": "F1 Bananero",
9 "edición": "2nd",
11 "precio": 18.99,
12 "calificacion": 4.67,
13 "portada": "https://example.com/portada2.jpg",
14 "unidades_disponibles": 48,
15 "unicacion": null,
16 "genero": null,
16 "genero": null,
18 ]
19 }
```

Figura 7.

# 6. Consulta de comprobantes de compra

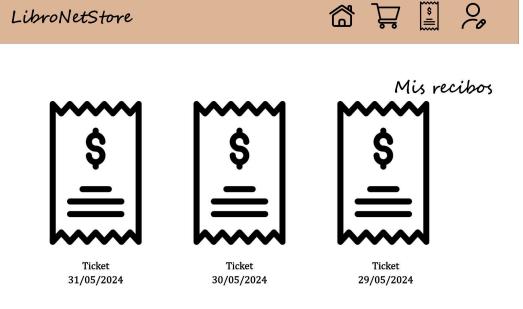


Figura 8.

## 7. Consulta de libros



Figura 9.

# 8. Agregar productos al carrito

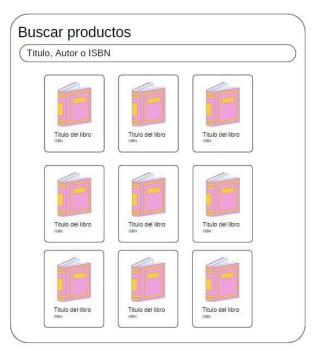




Figura 10.

## Resultados finales

En esta sección se mostrarán y explicarán los diferentes componentes que conforman el estado final del sistema. Se ha hecho uso de frameworks y herramientas modernas para el desarrollo, destacando Vite para la optimización del entorno, React para la creación de una interfaz de usuario interactiva y Tailwind CSS para el diseño responsive. Además, se detallarán los componentes React como BlockBooks, Button y Card, diseñados para maximizar la reusabilidad y la coherencia estética en toda la aplicación. Este documento también incluye descripciones detalladas de las páginas implementadas que facilitan diversas funcionalidades del sistema, desde la administración hasta la interacción directa con el usuario final.

## Frameworks y Páginas

#### Frameworks

- Vite: Utilizado para la configuración y optimización del entorno de desarrollo.
- React: El framework principal empleado para la construcción de la interfaz de usuario, permitiendo la creación de componentes reutilizables.
- Tailwind CSS: Utilizado para la estilización de la interfaz, ofreciendo una metodología de diseño basada en clases utilitarias que facilita la creación de diseños modernos y responsivos.

## Componentes React

- **BlockBooks**: Muestra una lista de libros. Si está en la página de inicio, muestra libros destacados; de lo contrario, permite buscar libros por título y muestra los resultados.
- **Button**: Un componente de botón estilizado que se puede reutilizar en todo el proyecto, mostrando el contenido que se le pase como hijos (children).
- Card: Un contenedor con estilo de tarjeta que puede envolver cualquier contenido, dándole un fondo y un borde redondeados.
- **Footer**: Un componente de pie de página que muestra derechos de autor estáticos para la tienda, usando un fondo oscuro y texto en color blanco.
- **Header**: La barra de navegación principal del sitio, que incluye enlaces a las principales secciones del sitio y muestra el logo de la tienda.
- **Hero**: Un carrusel de imágenes que rota automáticamente, mostrando diferentes promociones o características de la tienda.

- **Input**: Un campo de entrada estilizado para formularios que puede ser utilizado en diversas partes del sitio para recoger información del usuario.
- **Label**: Define un componente de etiqueta para formularios, proporcionando texto descriptivo para un campo de entrada y mejorando la accesibilidad.
- ListeningBooks: Muestra una cuadrícula de libros utilizando el componente ListingBook para cada libro, permitiendo una visualización organizada de múltiples libros.
- **ListingBook**: Presenta la información de un libro individual en forma de tarjeta, que incluye una imagen, título, y otros detalles relevantes del libro.
- **NotFoundComponent**: Muestra una página de error 404 cuando no se encuentra un contenido solicitado, ofreciendo un enlace para regresar a la página principal.
- **SingleBook**: Detalla la información de un libro individual, mostrando su imagen, detalles como autor, edición, y precio, y permite añadir el libro al carrito de compras.

## Páginas Detalladas

- CRUD: Página dedicada a la gestión CRUD para el administrador. Está organizada por tablas de la base de datos, con la excepción de las tablas de compras, tickets y ventas que son de solo lectura. Antes de guardar cualquier modificación, se realiza una validación de datos para asegurar que solo se realicen cambios con tipos de datos válidos y que todos los campos estén correctamente completados.
- BookFormPage: Permite a los usuarios modificar detalles de un libro específico utilizando parámetros de búsqueda desde la URL. Utiliza el componente SingleBook para mostrar o editar la información del libro.
- BookPage: Muestra un bloque de libros, potencialmente en la página de inicio dependiendo del prop que recibe. Utiliza el componente BlockBooks.
- HomePage: Página principal que muestra un componente hero y un bloque de libros destacados. Combina los componentes Hero y BookPage.
- LoginPage: Gestiona el inicio de sesión de los usuarios, incluye formularios para autenticación y redirección condicional basada en el estado del usuario (admin o no admin).
- **NotFound**: Muestra una página de error cuando no se encuentra una ruta especificada usando el componente NotFoundComponent.

- **PointOfSale**: Funciona como una terminal de punto de venta, permitiendo buscar productos y añadirlos al carrito de compras.
- **ProfilePage**: Permite a los usuarios ver y editar su información personal como nombre, email, dirección y método de pago.
- **RegisterPage**: Gestiona el registro de nuevos usuarios, recoge datos personales y los envía al servidor para crear un nuevo perfil.
- **Scanner**: Proporciona una función de búsqueda de precios por ISBN, ideal para terminales de autoservicio en tiendas.
- **ShopCartPage**: Administra el carrito de compras, permitiendo a los usuarios añadir, editar cantidades o proceder al pago de los productos seleccionados.
- **Statistics**: Muestra estadísticas de ventas como libros más vendidos y categorías populares, usando visualizaciones como gráficos de barras.
- TicketPage: Genera un recibo detallado post-compra, con opciones para enviar por correo o guardar como PDF.
- Payment: Maneja la información de pago del usuario, permitiendo ingresar y validar detalles de tarjetas de crédito o débito.

## **Implementaciones**

Para el desarrollo de la tienda virtual, se han configurado meticulosamente diversas tablas en la base de datos MySQL para asegurar una gestión eficiente y detallada de la información. Estas tablas son cruciales para el funcionamiento del sistema, facilitando procesos clave como la gestión de inventarios, la administración de usuarios y las transacciones de venta. Cada tabla ha sido diseñada con campos específicos que capturan datos vitales, permitiendo operaciones suaves y eficaces que mejoran la experiencia de los usuarios y proporcionan a los administradores las herramientas necesarias para el mantenimiento y análisis del negocio. En los siguientes apartados, se explicará en detalle la estructura y el propósito de cada tabla implementada.

#### Tabla cliente

La tabla cliente está dedicada a almacenar la información de los clientes registrados en el sistema. Cada cliente tiene un identificador único id, el cual es un entero auto-incremental. La tabla incluye campos para el nombre (nombre) y apellidos (apellidos) del cliente, ambos

definidos como cadenas de texto con una longitud máxima de 100 caracteres y sin permitir valores nulos. El tipo de usuario se define mediante un campo tipo\_usuario, que puede tomar los valores 'activo' o 'inactivo'. Adicionalmente, se almacenan la dirección (direccion) y el RFC (rfc) del cliente, este último con una longitud máxima de 13 caracteres. Las columnas created\_at y updated\_at registran las marcas de tiempo de creación y última actualización del registro, respectivamente, utilizando la marca de tiempo actual por defecto. La combinación de estas columnas proporciona un seguimiento completo del estado y actividad de los clientes a lo largo del tiempo. La configuración del conjunto de caracteres y collation asegura el soporte adecuado para caracteres especiales y la correcta ordenación de los datos.

Figura 11.

#### Tabla compra

La tabla compra registra todas las compras realizadas por los clientes en el sistema. Cada registro de compra es único y se identifica mediante la columna id\_compra, la cual es un entero auto-incremental. La tabla incluye referencias al usuario que realizó la compra (id\_usuario) y al libro adquirido (isbn). La fecha de la compra se almacena en fecha\_compra, mientras que la cantidad comprada se especifica en cantidad. El precio total de la compra se guarda en precio\_total, un campo decimal con dos decimales de precisión. Además, se incluye un campo estado\_compra que puede tomar los valores 'pendiente', 'completada' o 'cancelada', proporcionando un seguimiento del estado de cada compra. Las claves foráneas aseguran la integridad referencial con las tablas usuario y libro, y se establecen restricciones para manejar adecuadamente las actualizaciones y eliminaciones de registros relacionados. Esta tabla es crucial para llevar un control detallado y preciso de todas las transacciones de compra dentro del sistema.

```
DROP TABLE IF EXISTS `compra`;

/*!40101 SET @saved_cs_client = @@character_set_client */;

/*!50503 SET character_set_client = utf8mb4 */;

CREATE TABLE `compra` (
    `ID_compra` int NOT NULL AUTO_INCREMENT,
    `id_usuario` int DEFAULT NULL,
    `isbn` varchar(100) DEFAULT NULL,
    `cantidad` int NOT NULL,
    `cantidad` int NOT NULL,
    `estado_compra` date NOT NULL,
    `estado_compra` enum('pendiente', 'completada', 'cancelada') NOT NULL,
    PRIMARY KEY ('ID_compra`),
    KEY `isbn` (`isbn`),
    KEY `isbn` (`isbn`),
    CONSTRAINT `compra_ibfk_1` FOREIGN KEY (`id_usuario`) REFERENCES `usuario` (`id`),
    CONSTRAINT `compra_ibfk_2` FOREIGN KEY (`id_usuario`) REFERENCES `usuario` (`id`),
    CONSTRAINT `compra_ibfk_2` FOREIGN KEY (`id_usuario`) REFERENCES `usuario` (`id`) ON DELETE RESTRICT ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=utf8mb4_0900_ai_ci;

/*!40101 SET character_set_client = @saved_cs_client */;
```

Figura 12.

#### Tabla libros

La tabla libro almacena la información detallada de los libros disponibles en el sistema. Cada libro se identifica de manera única mediante el campo isbn, una cadena de texto con una longitud máxima de 100 caracteres. Los campos adicionales incluyen el título del libro (titulo), el autor (autor), la editorial (editorial), y la edición (edicion), todos definidos como cadenas de texto con una longitud máxima de 255 caracteres. La descripción del libro se almacena en descripcion, mientras que el precio se registra en un campo de tipo flotante (precio). También se incluyen campos para la calificación del libro (calificacion), la imagen de portada (portada), y la cantidad de unidades disponibles (unidades\_disponibles). La ubicación física del libro se especifica en el campo ubicacion y el género en genero, ambos definidos como cadenas de texto con una longitud máxima de 255 caracteres. Esta tabla proporciona una estructura completa y detallada para la gestión de los libros en el sistema, asegurando que toda la información relevante esté disponible y organizada de manera eficiente.

```
DROP TABLE IF EXISTS `libro`;
                                  = @@character set client */;
/*!50503 SET character set client = utf8mb4 */;
CREATE TABLE "libro" (
  isbn varchar(100) NOT NULL,
  `titulo` varchar(255) NOT NULL,
  `autor` varchar(255) NOT NULL,
  'editorial' varchar(255) NOT NULL,
  'edición' varchar(255) DEFAULT NULL,
  'descripcion' varchar(255) DEFAULT NULL,
  `precio` float NOT NULL,
  'calificacion' float DEFAULT NULL,
  `portada` varchar(255) DEFAULT NULL,
  'unidades_disponibles' int NOT NULL,
  `ubicacion` varchar(255) DEFAULT NULL,
  `genero` varchar(255) DEFAULT NULL,
 PRIMARY KEY ('isbn')
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4 0900 ai ci;
/*!40101 SET character_set_client = @saved_cs_client */;
```

Figura 13.

#### Tabla Tickets

La tabla ticket está diseñada para registrar los tickets generados por cada compra en el sistema. Cada ticket se identifica de manera única mediante el campo id\_ticket, el cual es un entero auto-incremental. La tabla incluye referencias a la compra asociada (id\_compra) y al libro relacionado (isbn). La cantidad de libros comprados se registra en el campo cantidad, y el precio total del ticket se almacena en precio\_venta, un campo decimal con dos decimales de precisión. Para garantizar la integridad referencial, se establecen claves foráneas que vinculan id\_compra con la tabla compra y isbn con la tabla libro. Esta estructura permite un seguimiento detallado de cada ticket generado, asegurando que toda la información relevante esté correctamente relacionada y gestionada dentro del sistema. La configuración del conjunto de caracteres y collation asegura el soporte adecuado para caracteres especiales y la correcta ordenación de los datos.

```
DROP TABLE IF EXISTS `ticket`;
/*!40101 SET @saved cs client = @@character set client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE "ticket" (
  'id ticket' int NOT NULL AUTO INCREMENT,
  `id_compra` int DEFAULT NULL,
  `isbn` varchar(100) DEFAULT NULL,
 `cantidad` int NOT NULL,
  'precio_venta' decimal(10,2) NOT NULL,
 PRIMARY KEY ('id_ticket'),
 KEY 'id_compra' ('id_compra'),
 KEY 'isbn' ('isbn'),
 CONSTRAINT 'ticket_ibfk_1' FOREIGN KEY ('id_compra') REFERENCES 'compra'),
 CONSTRAINT 'ticket_ibfk_2' FOREIGN KEY ('isbn') REFERENCES 'libro' ('isbn')
) ENGINE=InnoDB AUTO INCREMENT=13 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4 0900 ai ci;
/*!40101 SET character set client = @saved cs client */;
```

Figura 14.

#### Tabla Usuarios

La tabla usuario almacena la información de los usuarios registrados en el sistema. Cada usuario tiene un identificador único id, el cual es un entero auto-incremental. La tabla incluye campos para el nombre (nombre) y apellidos (apellidos) del usuario, ambos definidos como cadenas de texto con una longitud máxima de 100 caracteres y sin permitir valores nulos. El tipo de usuario se define mediante un campo tipo\_usuario, que puede tomar los valores 'admin' o 'vendedor'. Adicionalmente, se almacenan la dirección (dirección) y el RFC (rfc) del usuario, este último con una longitud máxima de 13 caracteres. Las columnas created\_at y updated\_at registran las marcas de tiempo de creación y última actualización del registro, respectivamente, utilizando la marca de tiempo actual por defecto. La combinación de estas columnas proporciona un seguimiento completo del estado y actividad de los usuarios a lo largo del tiempo. La configuración del conjunto de caracteres y collation asegura el soporte adecuado para caracteres especiales y la correcta ordenación de los datos.

```
DROP TABLE IF EXISTS `usuario`;

/*!40101 SET @saved_cs_client = @@character_set_client */;

/*!50503 SET character_set_client = utf8mb4 */;

CREATE TABLE `usuario` (
    id` int NOT NULL AUTO_INCREMENT,
    nombre` varchar(100) NOT NULL,
    `apellidos` varchar(100) NOT NULL,
    `tipo_usuario` enum('admin','vendedor') NOT NULL,
    `direccion` varchar(255) DEFAULT NULL,
    `rfc` varchar(13) DEFAULT NULL,
    `created_at` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
    `updated_at` timestamp NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    PRIMARY KEY ('id`)

) ENGINE=InnoDB AUTO_INCREMENT=77 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

/*!40101 SET character_set_client = @saved_cs_client */;
```

Figura 15.

#### Tabla venta

La tabla venta está diseñada para registrar las ventas realizadas en el sistema por empleados en tienda física. Cada registro en esta tabla representa una venta individual y se identifica de manera única mediante la columna id\_venta, la cual es un campo de tipo entero y se incrementa automáticamente con cada nueva inserción. Además, la tabla incluye una referencia al usuario que realizó la venta a través del campo id\_usuario, y una referencia al libro vendido mediante el campo isbn. La fecha de la venta se almacena en la columna fecha\_venta, y la cantidad de unidades vendidas se registra en cantidad\_vendida. El precio total de la venta se almacena en precio\_venta, un campo de tipo decimal con precisión de hasta dos decimales. Para garantizar la integridad referencial, se establecen claves foráneas que vinculan id\_usuario con la tabla usuario y isbn con la tabla libro. Además, se definen restricciones para manejar la actualización y eliminación de registros relacionados, asegurando que los cambios en los usuarios y libros se reflejen adecuadamente en las ventas asociadas. Esta estructura permite una gestión robusta y coherente de la información de ventas dentro del sistema.

Figura 16.

## **Vistas**

1. Inicio de sesión



Figura 17.

## 2. Pantalla de Bienvenida

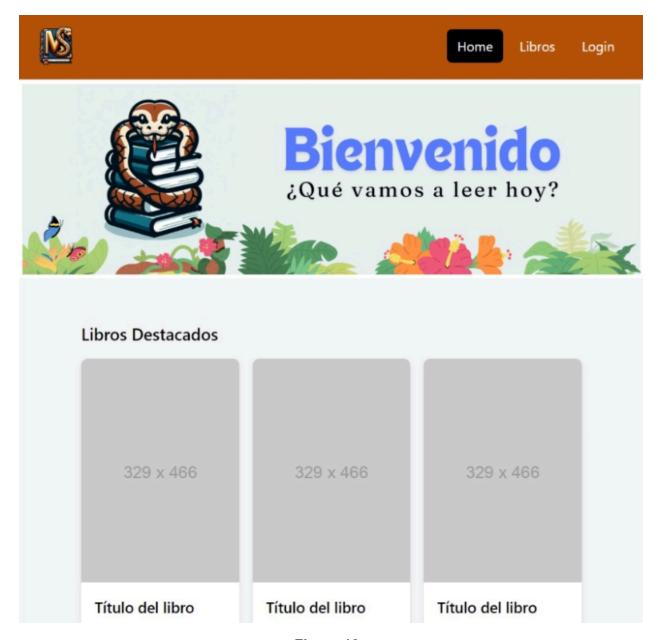


Figura 18.

## 3. Descripción de libro



Figura 19.

4. Carrito de compras y "Buscar y agregar"

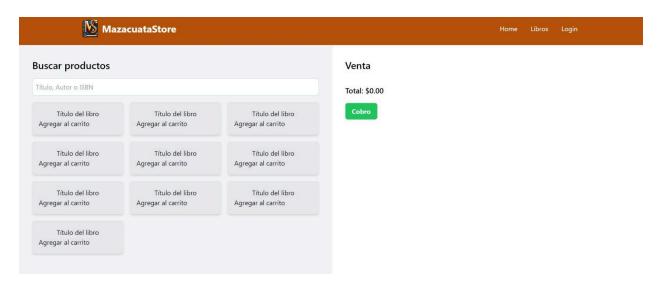


Figura 20.

5. Buscador de un libro/Terminales de usuario en tienda





Figura 21.

6. Ventana de Pago

<u>w</u>			Home	Libros	Login
	Forma de pago Pago con Tarjeta d Número de la tarjet 1234 1234 8975 4	ébito / crédito a			
	MM/AA	CVV/CVC			
	Nombre como apare	ece en la tarjeta			
	Ag	regar forma de pago			

Figura 22.

## 7. Confirmación del pedido



Figura 23.

## 8. Comprobante de compra



Figura 24.