



MR. BOOK

Manual Técnico

Versión 1.1

10/02/2025

Autores:

-Chavez Mayerli

-Chuga Juan

-Fuentes Carlos

-Gamboa Anthony

Índice

1.	Introducción	3
2.	Objetivo.....	3
3.	Requisitos del sistema	3
4.	Arquitectura del Software	4
4.1	Diagrama de la arquitectura	4
4.1.1	Capas de la Aplicación	4
4.1.2	Comunicación entre Capas.....	6
4.1.3	Beneficios de la Arquitectura en Capas.....	6
4.2	Diagrama de caso de uso	7
4.2.1	Actores del Sistema.....	7
4.2.2	Casos de Uso y Relaciones	7
4.2.3	Relaciones entre Casos de Uso	11
4.3	Diagrama de clases	12
4.3.1	Clases principales y descripción	12
4.3.2	Relaciones entre clases	14
4.4	Modelo Entidad-relación.....	15
5.	Base de datos	17
5.1	DAC (Data Access Component).....	¡Error! Marcador no definido.

Índice de figuras

Figura 1.	Diagrama de la arquitectura del software.....	4
Figura 2.	Diagrama de caso de uso.	7
Figura 3.	Diagrama de clases.	12
Figura 4.	Diagrama de MER.....	15

1. Introducción

Se propone el desarrollo de un sistema integral para la gestión de alquileres y devoluciones en la librería Mr. Book. Este sistema se apoyará en un registro detallado de usuarios y una documentación precisa del inventario de la librería.

Para agilizar los procesos, el sistema incorporará un lector de códigos de barras, asignando a cada libro un código único compartido por todas sus copias.

2. Objetivo

El propósito de este manual es documentar en detalle el funcionamiento y la estructura interna del Sistema de gestión bibliotecaria MR. BOOK diseñado para gestionar el alquiler y la devolución de libros mediante códigos ISBN y códigos de barras.

Este documento está dirigido a desarrolladores y personal técnico, proporcionando una descripción clara de la arquitectura del sistema, la estructura de la base de datos, los módulos principales y el flujo de operaciones. Además, incluye información sobre la integración con otros sistemas.

El manual servirá como referencia para la comprensión, optimización y futura evolución del sistema, asegurando su eficiencia y correcto desempeño en la gestión bibliotecaria.

3. Requisitos del sistema

Este sistema es compatible con cualquier versión de sistema operativo Windows.

Procesador: core i3 o superior

RAM: 4 GB mínimo

Para su uso, es necesario descargar el código de sistema realizado en lenguaje Java y solamente correrlo. El código lo podrá encontrar dentro del link de Github, el cual se encuentra citado en Referencias.

También, para un uso mas eficiente es recomendable tener un lector de código de barras. De esta manera, será posible encontrar cualquier libro escaneando todo su código.

Librerías:

-**Java Swing:** es una biblioteca gráfica de **Java** que proporciona un conjunto de componentes y herramientas para construir interfaces gráficas de usuario (GUI).

-**AWT:** es una librería grafica de Java que aparte de permitir crear interfaces graficas, permite crear evento para botones. (clicks, llamados, etc.).

4. Arquitectura del Software

4.1 Diagrama de la arquitectura

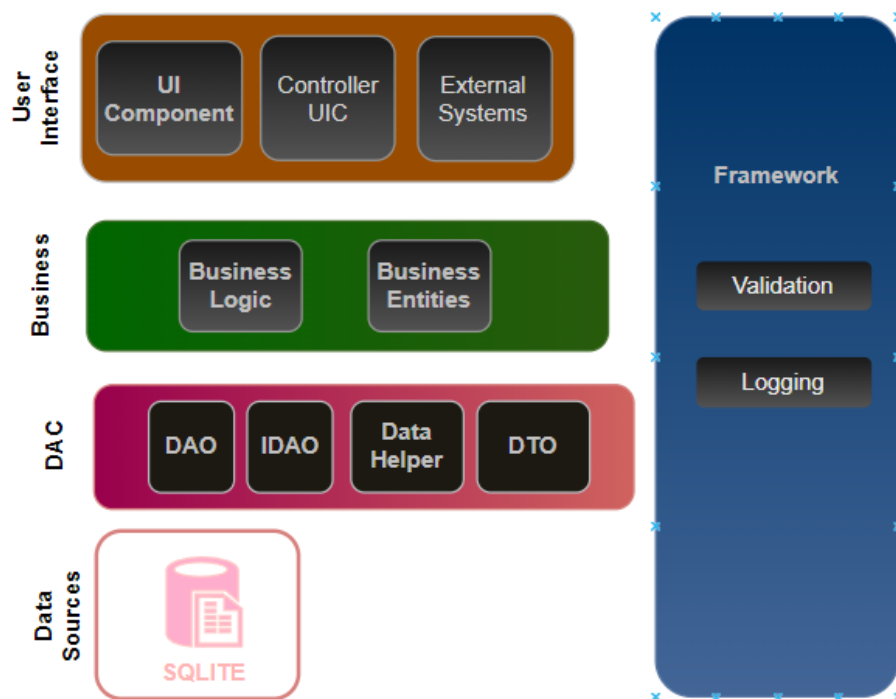


Figura 1. Diagrama de la arquitectura del software

La aplicación sigue una arquitectura en capas (N-Layer) con el objetivo de separar responsabilidades, mejorar la mantenibilidad y facilitar la escalabilidad del sistema. Esta estructura permite una organización clara del código y una mejor gestión de cambios o mejoras en el software.

4.1.1 Capas de la Aplicación

La arquitectura de la aplicación está dividida en varias capas, cada una con una función específica para garantizar la separación de responsabilidades y la modularidad del sistema.

La arquitectura de la aplicación está dividida en las siguientes capas:

a. *User Interface (UI):*

- Esta capa es la interfaz gráfica con la que interactúa el usuario. Se encarga de presentar la información y recibir entradas del usuario.
- Contiene los formularios, botones y elementos visuales.
- Se comunica con la Business Logic Layer para solicitar información o enviar datos.

b. *Business Logic Layer:*

- Es la capa intermedia encargada de la lógica de la aplicación y las reglas de negocio. Procesa la información antes de enviarla a la base de datos o de mostrarla en la UI.
- Contiene la lógica de la aplicación y las reglas de negocio.
- Procesa los datos que provienen de la User Interface antes de enviarlos a la base de datos.
- Se comunica con la Data Access Layer para obtener y almacenar información.

c. *Data Access Layer:*

- Actúa como intermediario entre la capa de negocio y la base de datos, garantizando la correcta gestión de los datos.
- Es responsable de la comunicación con la base de datos.
- Ejecuta consultas SQL, procedimientos almacenados y gestiona la persistencia de datos.
- Se asegura de que los datos sean recuperados y almacenados correctamente.

d. *Database Layer:*

- Es la capa donde se almacenan y gestionan los datos de la aplicación.
- Contiene las tablas, relaciones, índices y procedimientos almacenados.
- Se encarga de la gestión y almacenamiento de la información de la aplicación.

4.1.2 Comunicación entre Capas

Las capas interactúan entre sí mediante un flujo de información estructurado que garantiza la separación de responsabilidades y un funcionamiento eficiente del sistema.

- La **User Interface** envía peticiones a la **Business Logic Layer**, que las procesa y las reenvía a la **Data Access Layer** si es necesario.
- La **Data Access Layer** se comunica con la **Database Layer** para recuperar o almacenar información.
- Los datos retornan a la **Business Logic Layer**, donde pueden ser procesados antes de ser enviados nuevamente a la **User Interface** para mostrarse al usuario.

4.1.3 Beneficios de la Arquitectura en Capas

La arquitectura en capas ofrece diversas ventajas que facilitan el desarrollo, mantenimiento y escalabilidad de la aplicación.

- **Mantenibilidad:** Permite modificar una capa sin afectar directamente a las demás.
- **Escalabilidad:** Facilita la ampliación del sistema agregando nuevas funcionalidades sin afectar la estructura principal.
- **Reutilización de código:** Cada capa puede ser reutilizada en otros proyectos con una estructura similar.
- **Seguridad:** Se puede aplicar control de acceso en cada capa para proteger la información.

Este enfoque modular permite un desarrollo más organizado y eficiente, asegurando una separación clara de responsabilidades y facilitando futuras mejoras o integraciones.

4.2 Diagrama de caso de uso

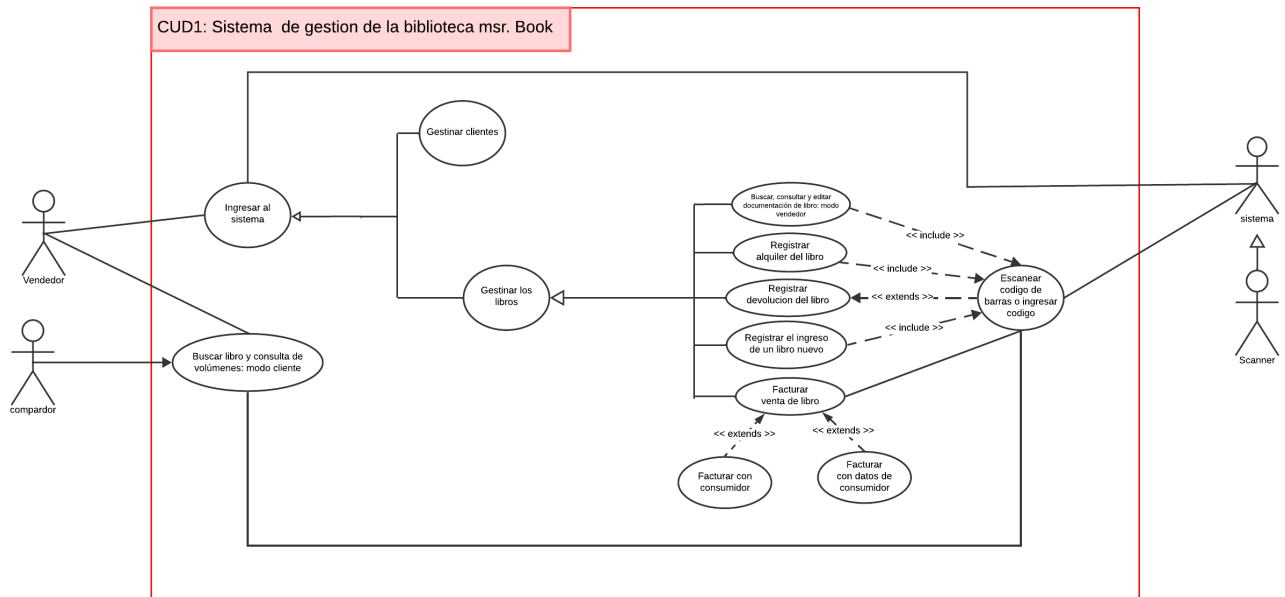


Figura 2. Diagrama de caso de uso.

funcionalidades que proporciona la aplicación de gestión bibliotecaria. Su propósito es identificar los actores principales y describir los procesos que pueden ejecutar dentro del sistema.

4.2.1 Actores del Sistema

En el sistema de gestión bibliotecaria se identifican los siguientes actores:

- **Vendedor:** Usuario encargado de la administración de libros, registro de transacciones y gestión de clientes.
- **Comprador:** Usuario que busca libros y realiza compras o alquileres.
- **Sistema:** Representa la plataforma que gestiona la información y operaciones del sistema.
- **Scanner:** Dispositivo o módulo utilizado para la identificación de libros mediante códigos de barras.

4.2.2 Casos de Uso y Relaciones

A continuación, se describen los principales casos de uso del sistema:

a. Casos de Uso del Vendedor

- ***CU01 – Ingresar al sistema***

El vendedor debe autenticarse en la aplicación ingresando su usuario y contraseña.

Flujo de eventos:

El vendedor ingresa sus credenciales en la interfaz de inicio de sesión.

El sistema valida los datos con la base de datos.

Si los datos son correctos, el sistema concede acceso.

Si los datos son incorrectos, muestra un mensaje de error.

- ***CU02 – Gestionar los libros***

Permite al vendedor agregar, actualizar o eliminar libros en el sistema.

Flujo de eventos:

El vendedor selecciona la opción "Gestión de libros".

Elige una de las siguientes acciones:

Registrar un nuevo libro.

Modificar la información de un libro existente.

Eliminar un libro del catálogo.

El sistema procesa la solicitud y actualiza la base de datos.

- ***CU03 – Registrar devolución del libro***

Permite registrar la devolución de un libro alquilado.

Flujo de eventos:

El vendedor accede a la opción "Registrar devolución".

Introduce el código del libro o lo escanea.

El sistema actualiza el estado del libro a "Disponible".

Se genera un comprobante de devolución.

- ***CU04 – Registrar alquiler del libro***

Proceso mediante el cual un cliente alquila un libro.

Flujo de eventos:

El vendedor selecciona la opción "Registrar alquiler".

Introduce el código del libro o lo escanea.

Registra los datos del cliente.

El sistema cambia el estado del libro a "Prestado".

Se genera un recibo de alquiler.

- ***CU05 – Facturar venta de libro***

Genera una factura cuando un cliente compra un libro.

Flujo de eventos:

El vendedor selecciona la opción "Facturar venta".

Escanea el código del libro o ingresa el código manualmente.

Introduce la cantidad de ejemplares vendidos.

El sistema calcula el total y muestra los detalles.

El vendedor selecciona el tipo de facturación:

Facturar con consumidor (*incluye*)

Facturar con datos de consumidor (*extiende*)

Se genera la factura y se registra la transacción.

- ***CU06 – Registrar ingreso de un libro nuevo***

Opción para agregar nuevos libros al catálogo.

Flujo de eventos:

El vendedor accede a la opción "Registrar nuevo libro".

Ingresa los datos del libro (título, autor, editorial, categoría, cantidad, etc.).

Guarda la información en la base de datos.

- ***CU07 – Buscar, consultar y editar documentación de libros (modo vendedor)***

Permite al vendedor buscar libros y modificar su información.

Flujo de eventos:

El vendedor accede al módulo de búsqueda.

Introduce el título o código del libro.

El sistema muestra los resultados.

El vendedor puede seleccionar un libro y editar su información.

- ***CU08 – Gestionar clientes***

Permite al vendedor registrar, modificar o eliminar clientes en el sistema.

Flujo de eventos:

El vendedor accede al módulo de gestión de clientes.

Puede realizar las siguientes acciones:

Registrar un nuevo cliente.

Modificar la información de un cliente.

Eliminar un cliente de la base de datos.

El sistema guarda los cambios en la base de datos.

b. Casos de Uso del Comprador

- **CU09 – Buscar libro y consultar volúmenes (modo cliente)**

Permite a los compradores acceder al catálogo de libros y consultar su información.

Flujo de eventos:

El comprador accede al módulo de búsqueda.

Ingresa el título o código del libro.

El sistema muestra los resultados con la disponibilidad de cada ejemplar.

c. *Casos de Uso del Sistema y Scanner*

- **CU10 – Escanear código de barras o ingresar código**

Proceso utilizado para identificar libros de forma rápida mediante escaneo.

Flujo de eventos:

El usuario escanea el código de barras del libro.

El sistema busca la información en la base de datos.

Si el libro está registrado, se muestra la información en pantalla.

4.2.3 Relaciones entre Casos de Uso

En el diagrama de casos de uso, se establecen relaciones entre funcionalidades como:

- **Relación "include":** Un caso de uso siempre llama a otro como parte de su proceso.

Escanear código de barras → (Incluye) → Buscar libro

Facturar con consumidor → (Incluye) → Facturar venta de libro

- **Relación "extend":** Un caso de uso opcional que amplía otro proceso.

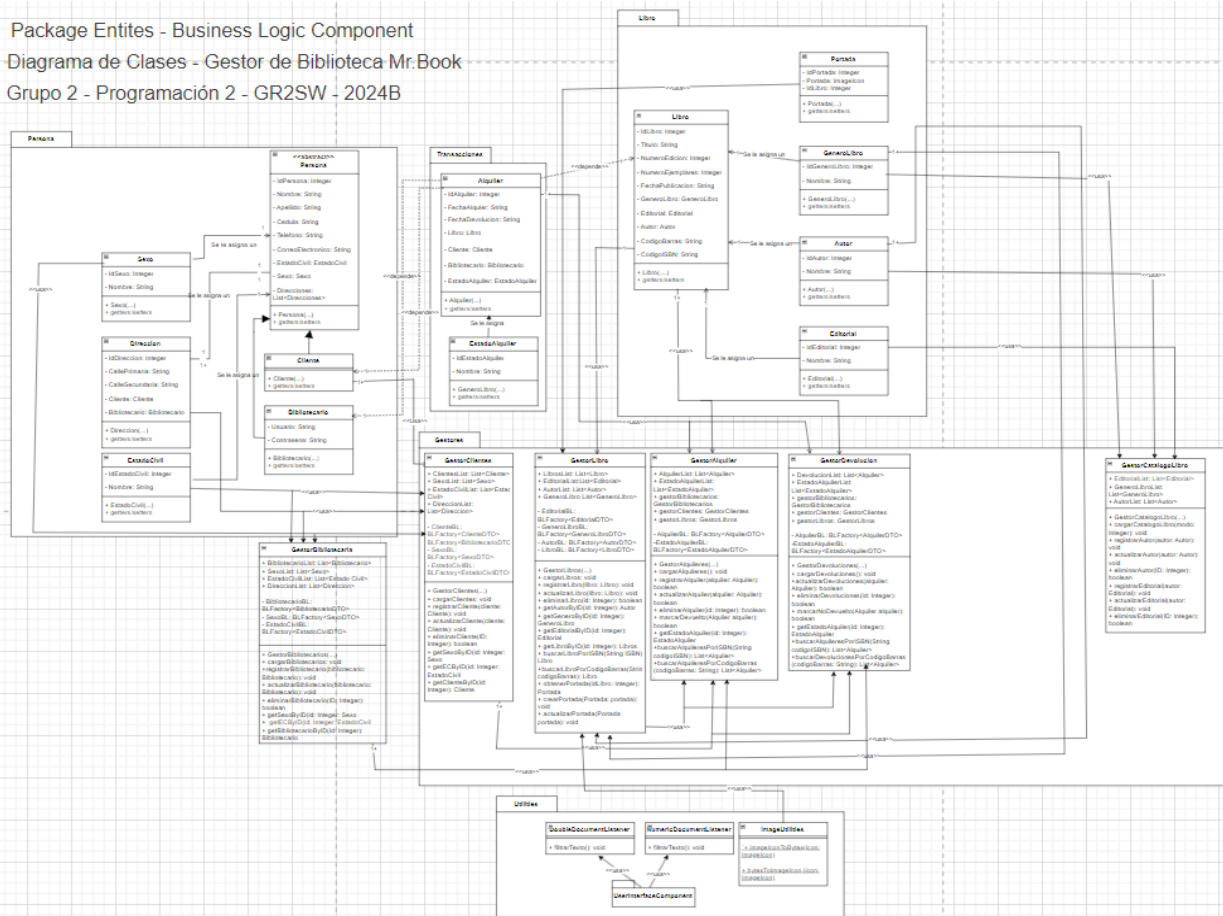
Facturar venta de libro (Extiende) → Facturar con datos de consumidor.

4.3 Diagrama de clases



Diagrama de Clases - Gestor de Biblioteca Mr.Book

Grupo 2 - Programación 2 - GR2SW - 2024B



A 3.1 Clases principales y descripción

Figura 3. Diagrama de clases.

a. Persona

Tipo: Clase Abstracta

Representa una entidad base para personas, que puede ser especializada en subclases. Es utilizada para abstraer características comunes a todos los tipos de personas en el sistema, facilitando la extensión del modelo.

Atributos:

IdPersona (Integer): Identificador único de la persona.

Nombre (String): Nombre de la persona.

Apellido (String): Apellido de la persona.

Cedula (String): Número de identificación oficial.

b. Libro

Tipo: Clase

Representa un libro dentro del sistema, permitiendo la gestión de información relacionada, como título, autor y disponibilidad.

Atributos:

Detalles no especificados en el análisis inicial.

c. Transacciones

Tipo: Clase

Maneja las operaciones realizadas en el sistema, como préstamos o devoluciones de libros. Es crucial para registrar y rastrear el historial de actividades.

Atributos:

Detalles no especificados en el análisis inicial.

d. Gestores

Tipo: Clase

Administra la lógica de las operaciones de las clases principales, actuando como intermediario entre las entidades y los servicios del sistema.

Atributos:

Detalles no especificados en el análisis inicial.

e. Utilities

Tipo: Clase

Proporciona herramientas o funcionalidades adicionales para soportar la lógica del sistema, como validaciones, conversión de datos o utilidades comunes.

Atributos:

Detalles no especificados en el análisis inicial.

4.3.2 Relaciones entre clases

Herencia: La clase Persona actúa como clase base abstracta. Subclases específicas podrían derivarse de esta para representar diferentes tipos de personas en el sistema.

Composición/Agrupación:

Se observan relaciones implícitas que podrían representar composición entre las entidades Transacciones y Libro

Los Gestores podrían actuar como controladores para las operaciones de entidades como Libro y Transacciones.

Asociaciones: Relación directa entre las entidades, aunque los detalles específicos no se han identificado en esta versión inicial del diagrama.

- Atributos: IDAlquiler, Estado, FechaAlquiler, FechaDevolucion, FechaModificacion, IDLibro, IDCliente, IDBibliotecario, IDEstadoAlquiler.

4. **Libro**

- Atributos: Titulo, NumeroEdicion, NumeroEjemplares, FechaPublicacion, Estado, FechaCreacion, FechaModificacion, IDEstado, IDGenero, IDEditorial, IDAutor, ConfigBarras,CodigoISBN.

5. **Direccion**

- Atributos: IDDireccion, CallePrincipal, CalleSecundaria, Estado, FechaCreacion, FechaModificacion.

6. **Sexo**

- Atributos: IDSexo, Nombre, Estado, FechaCreacion, FechaModificacion.

7. **EstadoCivil**

- Atributos: IDEstadoCivil, Nombre, Estado, FechaCreacion, FechaModificacion.

8. **EstadoAlquiler**

- Atributos: IDEstadoAlquiler, Nombre, Estado, FechaCreacion, FechaModificacion.

9. **Portada**

- Atributos: IDPortada, IDLibro, Estado, FechaCreacion, FechaModificacion.

10. **GeneroLibro**

- Atributos: IDGeneroLibro, Nombre, Estado, FechaCreacion, FechaModificacion.

11. **Editorial**

- Atributos: IDEditorial, Nombre, Estado, FechaCreacion, FechaModificacion.

12. **Autor**

- Atributos: IDAutor, Nombre, Estado, FechaCreacion, FechaModificacion.

4.4.1.3 *Relaciones Entre Entidades:*

- Un **Bibliotecario** puede gestionar múltiples **Alquileres** y atender a múltiples **Clientes**.

- Un **Cliente** puede realizar múltiples **Alquileres** de **Libros**.
- Un **Libro** puede tener una **Portada**, pertenecer a un **GeneroLibro**, y estar asociado con una **Editorial** y uno o más **Autores**.
- Las entidades **Direccion**, **Sexo**, **EstadoCivil**, y **EstadoAlquiler** se utilizan para almacenar atributos relacionados y se relacionan con otras entidades principales como **Bibliotecario** y **Cliente**.

5. Base de datos

• 1. Introducción

La base de datos **biblioteca.db** está diseñada para gestionar y almacenar información relacionada con una biblioteca, incluyendo libros, clientes, bibliotecarios, alquileres y otros elementos asociados a la gestión de una biblioteca. A través de las tablas y relaciones definidas, se facilita la administración de libros, su alquiler, y el seguimiento de la información de los usuarios y personal bibliotecario.

Este manual técnico proporciona una descripción detallada de las tablas, sus campos, las relaciones entre ellas y las reglas de negocio implementadas.

• 2. Definición de Tablas y Relaciones

• Tablas

• 2.1 GeneroLibro

- **Propósito:** Almacenar los géneros literarios disponibles para los libros de la biblioteca.

• Campos:

- IdGeneroLibro: Identificador único del género (clave primaria, autoincremental).
- NombreGeneroLibro: Nombre del género literario (e.g., Ficción, Ciencia, etc.).
- Estado: Estado del género, 'A' por activo (valor predeterminado).
- FechaCreacion: Fecha de creación del registro.
- FechaModificacion: Fecha de la última modificación del registro.

- **2.2 Editorial**

- **Propósito:** Contiene información sobre las editoriales que publican los libros en la biblioteca.

- **Campos:**

- IdEditorial: Identificador único de la editorial (clave primaria, autoincremental).
- NombreEditorial: Nombre de la editorial.
- Estado: Estado de la editorial, 'A' por activo.
- FechaCreacion: Fecha en que se creó el registro de la editorial.
- FechaModificacion: Fecha de la última modificación del registro.

- **2.3 Autor**

- **Propósito:** Información sobre los autores de los libros en la biblioteca.

- **Campos:**

- IdAutor: Identificador único del autor (clave primaria, autoincremental).
- NombreAutor: Nombre completo del autor.
- Estado: Estado del autor, 'A' por activo.
- FechaCreacion: Fecha de creación del registro.
- FechaModificacion: Fecha de la última modificación.

- **2.4 Sexo**

- **Propósito:** Gestiona los géneros de las personas (clientes y bibliotecarios).

- **Campos:**

- IdSexo: Identificador único (clave primaria, autoincremental).
- Nombre: Género (por ejemplo, 'Masculino', 'Femenino').
- Estado: Estado del género, 'A' por activo.
- FechaCreacion: Fecha de creación del registro.
- FechaModificacion: Fecha de la última modificación.

- **2.5 EstadoCivil**

- **Propósito:** Almacena los estados civiles disponibles (e.g., soltero, casado) para los clientes y bibliotecarios.

- **Campos:**

- IdEstadoCivil: Identificador único del estado civil (clave primaria, autoincremental).
- Nombre: Nombre del estado civil (e.g., Soltero, Casado).
- Estado: Estado, 'A' por activo.
- FechaCreacion: Fecha de creación.
- FechaModificacion: Fecha de la última modificación.
- **2.6 EstadoAlquiler**
- **Propósito:** Almacena los posibles estados de los alquileres de libros (e.g., activo, devuelto).
- **Campos:**
 - IdEstadoAlquiler: Identificador único del estado de alquiler (clave primaria, autoincremental).
 - Nombre: Nombre del estado (e.g., 'Activo', 'Devuelto').
 - Estado: Estado, 'A' por activo.
 - FechaCreacion: Fecha de creación del estado.
 - FechaModificacion: Fecha de la última modificación.
- **2.7 Libro**
- **Propósito:** Información sobre los libros disponibles en la biblioteca.
- **Campos:**
 - IdLibro: Identificador único del libro (clave primaria, autoincremental).
 - Titulo: Título del libro.
 - NumeroEdicion: Número de edición del libro.
 - NumeroEjemplares: Número total de ejemplares del libro.
 - FechaPublicacion: Año de publicación del libro.
 - Precio: Precio del libro.
 - Estado: Estado del libro, 'A' por activo.
 - FechaCreacion: Fecha de creación del registro.
 - FechaModificacion: Fecha de la última modificación.
 - IdGeneroLibro: Relación con el género del libro (referencia a la tabla **GeneroLibro**).

- IdEditorial: Relación con la editorial del libro (referencia a la tabla **Editorial**).
- IdAutor: Relación con el autor del libro (referencia a la tabla **Autor**).
- CodigoBarras: Código de barras único del libro.
- CodigoISBN: ISBN único del libro.
- **2.8 Portada**
- **Propósito:** Almacena la portada de cada libro en formato binario (BLOB).
- **Campos:**
 - IdPortada: Identificador único de la portada (clave primaria, autoincremental).
 - Portada: Imagen de la portada en formato BLOB.
 - IdLibro: Relación con el libro (referencia a la tabla **Libro**).
 - Estado: Estado de la portada, 'A' por activo.
 - FechaCreacion: Fecha de creación.
 - FechaModificacion: Fecha de la última modificación.
- **2.9 Cliente**
- **Propósito:** Información de los clientes que alquilan libros en la biblioteca.
- **Campos:**
 - IdCliente: Identificador único del cliente (clave primaria, autoincremental).
 - Nombre: Nombre del cliente.
 - Apellido: Apellido del cliente.
 - Cedula: Cédula de identidad del cliente (única).
 - Telefono: Teléfono de contacto del cliente.
 - CorreoElectronico: Correo electrónico del cliente (único).
 - Estado: Estado del cliente, 'A' por activo.
 - FechaCreacion: Fecha de creación del registro.
 - FechaModificacion: Fecha de la última modificación.
 - IdEstadoCivil: Relación con el estado civil del cliente (referencia a **EstadoCivil**).

- IdSexo: Relación con el género del cliente (referencia a **Sexo**).

- **2.10 Bibliotecario**

- **Propósito:** Información de los bibliotecarios encargados de la gestión de la biblioteca.

- **Campos:**

- IdBibliotecario: Identificador único del bibliotecario (clave primaria, autoincremental).
- Nombre: Nombre del bibliotecario.
- Apellido: Apellido del bibliotecario.
- Cedula: Cédula de identidad del bibliotecario (única).
- Telefono: Teléfono de contacto del bibliotecario.
- CorreoElectronico: Correo electrónico del bibliotecario (único).
- Usuario: Nombre de usuario para el acceso al sistema.
- Contraseña: Contraseña para el acceso al sistema.
- Estado: Estado del bibliotecario, 'A' por activo.
- FechaCreacion: Fecha de creación del registro.
- FechaModificacion: Fecha de la última modificación.
- IdEstadoCivil: Relación con el estado civil del bibliotecario (referencia a **EstadoCivil**).
- IdSexo: Relación con el género del bibliotecario (referencia a **Sexo**).

- **2.11 Alquiler**

- **Propósito:** Almacena los registros de los alquileres de libros realizados por los clientes.

- **Campos:**

- IdAlquiler: Identificador único del alquiler (clave primaria, autoincremental).
- Estado: Estado del alquiler, 'A' por activo.
- FechaAlquiler: Fecha en que se realizó el alquiler.
- FechaDevolucion: Fecha de devolución del libro.
- FechaModificacion: Fecha de la última modificación.
- IdLibro: Relación con el libro alquilado (referencia a **Libro**).

- IdCliente: Relación con el cliente que alquila el libro (referencia a **Cliente**).
 - IdBibliotecario: Relación con el bibliotecario que procesa el alquiler (referencia a **Bibliotecario**).
 - IdEstadoAlquiler: Relación con el estado del alquiler (referencia a **EstadoAlquiler**).
- **2.12 Direccion**
 - **Propósito:** Almacena las direcciones de los clientes y bibliotecarios.
 - **Campos:**
 - IdDireccion: Identificador único de la dirección (clave primaria, autoincremental).
 - CallePrincipal: Nombre de la calle principal.
 - CalleSecundaria: Nombre de la calle secundaria.
 - Estado: Estado de la dirección, 'A' por activo.
 - FechaCreacion: Fecha de creación del registro.
 - FechaModificacion: Fecha de la última modificación.
 - IdCliente: Relación con el cliente asociado (referencia a **Cliente**).
 - IdBibliotecario: Relación con el bibliotecario asociado (referencia a **Bibliotecario**).

Inserción de Datos (DML)

En esta parte, el código inserta datos en las distintas tablas de la base de datos relacionadas con una biblioteca. Los datos insertados corresponden a géneros de libros, editoriales, autores, clientes, bibliotecarios, direcciones, alquileres y otros detalles. A continuación, se explican las inserciones realizadas:

- **Género de Libro (GeneroLibro)**

Se insertan los géneros de libros disponibles en la biblioteca, tales como "Acción", "Terror", "Biografía", etc.

- **Editorial (Editorial)**

Se insertan los nombres de las editoriales disponibles, como "Planeta", "Santillana", "Anaya", entre otros.

- **Autor (Autor)**

Se insertan los nombres de autores como "Gabriel García Márquez", "J.R.R. Tolkien", "Mario Vargas Llosa", entre otros.

- **Sexo (Sexo)**

Se insertan los valores posibles para el sexo de los bibliotecarios y clientes: "Masculino", "Femenino", y "Otros".

- **Estado Civil (EstadoCivil)**

Se insertan los estados civiles posibles de los bibliotecarios y clientes: "Soltero", "Casado", "Unión Libre", "Divorciado", "Viudo".

- **Estado de Alquiler (EstadoAlquiler)**

Se insertan los posibles estados de un alquiler de libros: "Alquilado no devuelto", "Devuelto", "Inválido".

- **Libro (Libro)**

Se insertan registros de libros en la biblioteca con información como título, número de edición, número de ejemplares, fecha de publicación, precio, y otros detalles relacionados con el género, editorial y autor.

- **Bibliotecario (Bibliotecario)**

Se insertan datos de los bibliotecarios como nombre, apellido, cédula, teléfono, correo electrónico, usuario, contraseña, estado civil y sexo.

- **Dirección (Dirección)**

Se insertan las direcciones de clientes y bibliotecarios, especificando las calles principales y secundarias relacionadas.

- **Cliente (Cliente)**

Se insertan los datos de los clientes, incluyendo nombre, apellido, cédula, teléfono, correo electrónico, estado civil y sexo.

- **Alquiler (Alquiler)**

Se insertan registros de alquiler de libros, incluyendo las fechas de alquiler y devolución, junto con los detalles del libro alquilado, cliente, bibliotecario, y estado del alquiler.