

Aplicación para gestionar los datos de una biblioteca

Libni Jahzeel Torres Castro

Universidad Popular del Cesar
Facultad de ingenierías y tecnologías
Ingeniería de sistemas
Valledupar
2024



CO-SC-CER518726



www.unicesar.edu.co
Balneario Hurtado Vía a Patillal. PBX (57) (5) 5841000 EXT. 1040
Línea de atención al ciudadano 01 8000 400380
Valledupar Cesar Colombia

CONTENIDO

	Página.
INTRODUCCIÓN	4
Objetivo General	4
Objetivos Específicos	4
Tecnologías utilizadas	5
Arquitectura del sistema	6
1. Frontend (Interfaz Gráfica de Usuario)	6
2. Backend (Lógica de Negocio y Persistencia de Datos)	6
Diagramas UML	7
Descripción de clases y funciones	9
Clases y funciones relevantes agrupadas por módulo:	9
Módulo modelo	9
1. Clase usuario (extiende persona)	9
2. Clase persona	10
3. Clase item (abstracta)	10
4. Clase libro (extiende item)	11
Módulo vista	13
1. Clase devoluciones (extends javax.swing.JPanel)	13
2. Clase usuarios (extends javax.swing.JPanel)	13
3. Clase dashboard (extends javax.swing.JFrame)	14
4. Clase Prestamos (extends javax.swing.JPanel)	14
5. Clase libros_actuales (extends javax.swing.JPanel)	14
6. Clase AGREGAR_USUARIO (extends javax.swing.JPanel)	15
7. Clase principal_vista (extends javax.swing.JPanel)	15
8. Clase reportes (extends javax.swing.JPanel)	15
9. Clase AGREGAR_LIBROS (extends javax.swing.JPanel)	15
10. Clase Login (extends javax.swing.JFrame)	16
11. Clase AGREGAR_LIBROS (extends javax.swing.JPanel)	16
12. Clase AGREGAR_USUARIO (extends javax.swing.JPanel)	16
13. Clase reportes (extends javax.swing.JPanel)	17
14. Clase reportes_principal (extends javax.swing.JPanel)	17
15. Clase REPORTES_ENTREGADOS (extends javax.swing.JPanel)	17



Flujo de la aplicación	18
1. Página de Inicio de Sesión (Login).....	19
Campo de Usuario:.....	19
Campo de Contraseña:	20
Botón "Ingresar":	21
2. Panel Principal (Dashboard).....	23
3. Gestión de Libros (AGREGAR_LIBROS y libros_actuales)	24
4. Gestión de Usuarios (AGREGAR_USUARIO y usuarios)	27
5. Préstamos y Devoluciones (Prestamos y devoluciones)	30
6. Reportes (reportes y REPORTES_ENTREGADOS)	33
7. Vista Principal (principal_vista).....	35
Notas generales:	35
Pruebas realizadas	36
Consideraciones finales.....	44
Limitaciones:.....	44
Interfaz gráfica:	44
Escalabilidad:	44
Seguridad:	44
Posibles mejoras:.....	44
Notificaciones:	44
Mejora en la interfaz gráfica:	44
Optimización de la base de datos:	44
Seguridad:	45
Conclusión	46

INTRODUCCIÓN

Esta documentación describe el desarrollo de un sistema de gestión de bibliotecas diseñado para automatizar las principales tareas administrativas, como la gestión de libros, usuarios y préstamos. La aplicación permite realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar), gestionar de forma eficiente los préstamos y devoluciones, y generar reportes detallados. Su objetivo principal es mejorar la eficiencia y precisión en la administración de bibliotecas, reduciendo errores y optimizando el tiempo necesario para buscar y registrar información.

Esta documentación está orientada a detallar las características de la aplicación, así como los procesos y decisiones de diseño que permitieron alcanzar los siguientes objetivos:

Objetivo General

- Desarrollar un sistema automatizado de gestión de bibliotecas que permita administrar de manera eficiente los libros, los usuarios y los préstamos, optimizando el tiempo de búsqueda y registro de información.

Objetivos Específicos

- Digitalizar la gestión de libros y usuarios para facilitar su administración.
- Automatizar el registro de préstamos y devoluciones para minimizar errores.
- Optimizar la búsqueda de libros en el sistema.
- Garantizar la seguridad y persistencia de los datos almacenados.
- Facilitar la generación de reportes sobre préstamos y disponibilidad.
- Diseñar una interfaz gráfica amigable e intuitiva.
- Implementar buenas prácticas de programación y principios de Programación Orientada a Objetos (POO).

Esta documentación tiene como propósito proporcionar una guía clara y estructurada del sistema, abarcando su arquitectura, funcionalidades y las herramientas utilizadas para su desarrollo.



CO-SC-CER518726



www.unicesar.edu.co
Balneario Hurtado Vía a Patillal. PBX (57) (5) 5841000 EXT. 1040
Línea de atención al ciudadano 01 8000 400380
Valledupar Cesar Colombia

Tecnologías utilizadas

- Lenguaje de Programación: Java
- Framework de GUI: Swing
- Persistencia de Datos: Archivos de texto (TXT)
- IDE: NetBeans



Arquitectura del sistema

El proyecto se divide en dos componentes principales, diseñados para trabajar en conjunto y garantizar un sistema funcional, eficiente y escalable:

1. Frontend (Interfaz Gráfica de Usuario)

El frontend está implementado utilizando **Swing**, una biblioteca gráfica de Java. Este componente es responsable de proporcionar una interfaz gráfica amigable e intuitiva que facilite la interacción del usuario con el sistema. A través de esta interfaz, los usuarios pueden realizar operaciones como gestionar libros, registrar préstamos y devoluciones, y consultar reportes de forma visual y accesible.

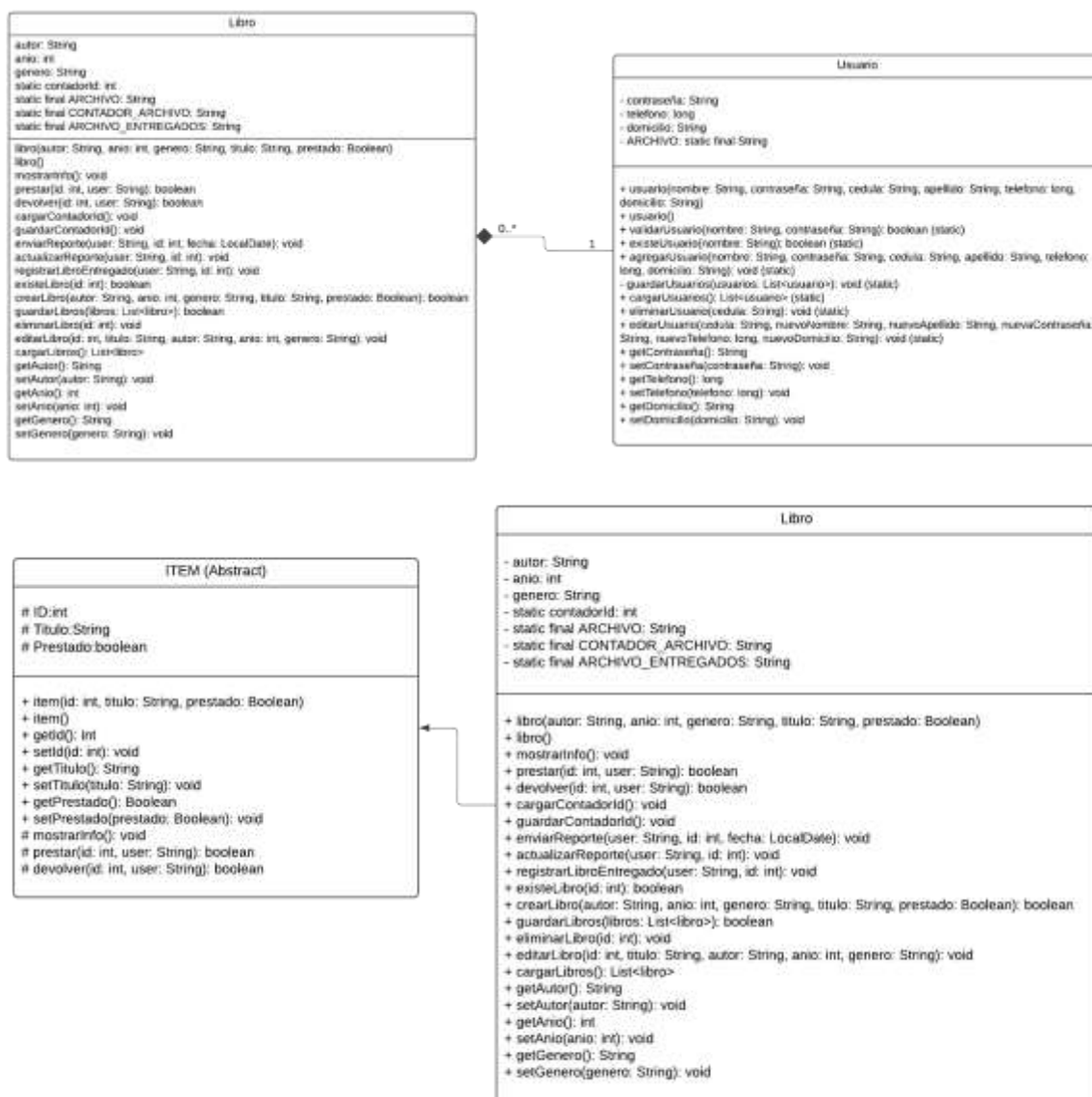
2. Backend (Lógica de Negocio y Persistencia de Datos)

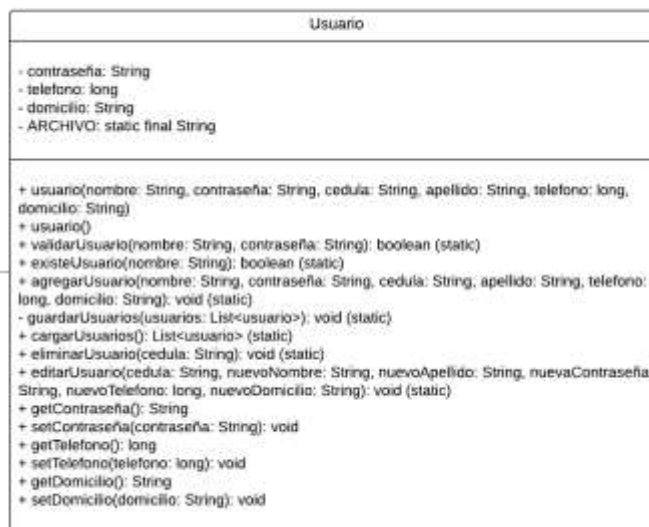
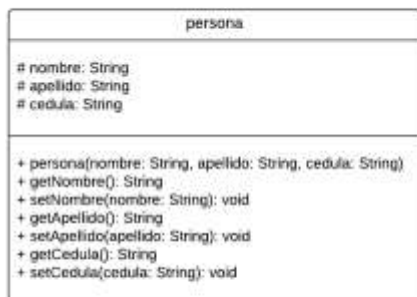
El backend se encarga de manejar la lógica de negocio del sistema, asegurando que las operaciones realizadas sean consistentes y efectivas. También gestiona la persistencia de datos utilizando archivos de texto, lo que permite almacenar y recuperar información de manera segura. Este componente incluye funciones clave como las operaciones CRUD, el registro automatizado de préstamos y devoluciones, y la generación de reportes detallados.

Esta división permite una clara separación de responsabilidades, haciendo que el sistema sea más mantenible y flexible para futuras ampliaciones o mejoras.



Diagramas UML





Descripción de clases y funciones

Clases y funciones relevantes agrupadas por módulo:

Módulo modelo

Documentación del Sistema de Gestión de Biblioteca

1. Clase usuario (extiende persona)

Descripción:

La clase usuario representa a un usuario del sistema de biblioteca. Hereda atributos básicos de la clase persona y añade información específica del usuario.

Atributos:

- serialVersionUID: long (constante)
Identificador de versión para serialización.
- ARCHIVO: String (constante)
Ruta del archivo donde se almacenan los datos de los usuarios.
- contraseña: String
Contraseña del usuario.
- telefono: long
Número de teléfono del usuario.
- domicilio: String
Dirección del domicilio del usuario.

Constructores:

- usuario(String nombre, String contraseña, String cedula, String apellido, long telefono, String domicilio)
Constructor completo que inicializa todos los atributos.
- usuario()
Constructor vacío que llama al constructor de la clase padre con valores nulos.

Métodos principales:

- validarUsuario(String nombre, String contraseña): boolean
Verifica si existe un usuario con el nombre y contraseña proporcionados.
- existeUsuario(String nombre): boolean
Comprueba si existe un usuario con el nombre dado.
- agregarUsuario(String nombre, String contraseña, String cedula, String apellido, long telefono, String domicilio): void

Añade un nuevo usuario al sistema.



CO-SC-CER518726



www.unicesar.edu.co
Balneario Hurtado Vía a Patillal. PBX (57) (5) 5841000 EXT. 1040
Línea de atención al ciudadano 01 8000 400380
Valledupar Cesar Colombia

- eliminarUsuario(String cedula): void
Elimina un usuario del sistema basándose en su cédula.
- editarUsuario(String cedula, String nuevoNombre, String nuevoApellido, String nuevaContraseña, long nuevoTelefono, String nuevoDomicilio): void
Modifica la información de un usuario existente.
- cargarUsuarios(): List<usuario>
Carga la lista de usuarios desde el archivo.
- guardarUsuarios(List<usuario> usuarios): void (privado)
Guarda la lista de usuarios en el archivo.

Getters y setters:

- getContraseña(), setContraseña(String contraseña)
- getTelefono(), setTelefono(long telefono)
- getDomicilio(), setDomicilio(String domicilio)

2. Clase persona

Descripción:

La clase persona es una clase base que representa a una persona con atributos básicos.

Atributos:

- nombre: String
Nombre de la persona.
- apellido: String
Apellido de la persona.
- cedula: String

Número de cédula o identificación de la persona.

Constructor:

- persona(String nombre, String apellido, String cedula)
Inicializa todos los atributos de la persona.

Métodos:

- getNombre(), setNombre(String nombre)
- getApellido(), setApellido(String apellido)
- getCedula(), setCedula(String cedula)

3. Clase item (abstracta)

Descripción:

La clase item es una clase abstracta que sirve como base para los elementos que pueden ser prestados en el sistema de biblioteca.



CO-SC-CER518726



www.unicesar.edu.co
Balneario Hurtado Vía a Patillal. PBX (57) (5) 5841000 EXT. 1040
Línea de atención al ciudadano 01 8000 400380
Valledupar Cesar Colombia

Atributos:

- id: int
Identificador único del item.
- titulo: String
Título del item.
- prestado: Boolean
Indica si el item está prestado o no.

Constructores:

- item(int id, String titulo, Boolean prestado)
Constructor que inicializa todos los atributos.
- item()
Constructor vacío.

Métodos:

- getId(), setId(int id)
- getTitulo(), setTitulo(String titulo)
- getPrestado(), setPrestado(Boolean prestado)
- mostrarInfo(): void (abstracto)
Método para mostrar la información del item.
- prestar(int id, String user): boolean (abstracto)
Método para prestar el item.
- devolver(int id, String user): boolean (abstracto)
Método para devolver el item.

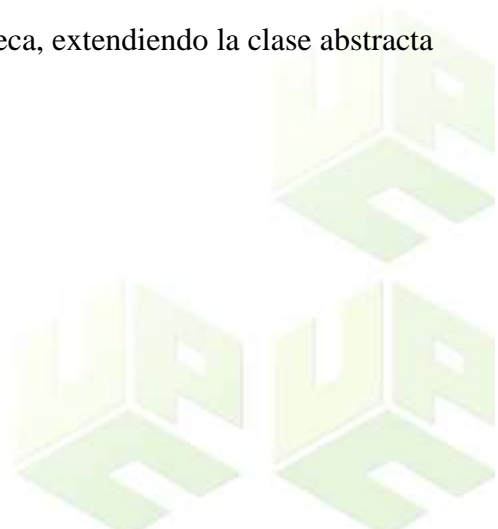
4. Clase libro (extiende item)

Descripción:

La clase libro representa un libro en el sistema de biblioteca, extendiendo la clase abstracta item.

Atributos:

- contadorId: int (estático)
Contador para generar IDs únicos.
- autor: String
Autor del libro.
- anio: int
Año de publicación del libro.
- genero: String
Género del libro.
- ARCHIVO: String (constante)



- Ruta del archivo para almacenar libros.
- CONTADOR_ARCHIVO: String (constante)
Ruta del archivo para el contador de ID.
- ARCHIVO_ENTREGADOS: String (constante)
Ruta del archivo para libros entregados.

Constructor:

- libro(String autor, int anio, String genero, String titulo, Boolean prestado)
Inicializa un nuevo libro con los atributos proporcionados.

Métodos principales:

- mostrarInfo(): void
Muestra la información del libro.
- prestar(int idLibro, String nombreUsuario): boolean
Realiza el préstamo de un libro.
- devolver(int idLibro, String nombreUsuario): boolean
Registra la devolución de un libro.
- crearLibro(String autor, int anio, String genero, String titulo, boolean prestado): boolean
Crea un nuevo libro y lo guarda en el sistema.
- eliminarLibro(int id): void
Elimina un libro del sistema por su ID.
- editarLibro(int id, String nuevoTitulo, String nuevoAutor, int nuevoAnio, String nuevoGenero): void
Edita la información de un libro existente.
- cargarLibros(): List<libro>
Carga la lista de libros desde el archivo.
- guardarLibros(List<libro> libros): boolean
Guarda la lista de libros en el archivo.

Métodos auxiliares:

- enviarReporte(String nombreUsuario, int idLibro, LocalDate fechaSalida): void
Registra un reporte de préstamo.
- registrarLibroEntregado(String nombreUsuario, int idLibro): void
Registra la entrega de un libro.
- actualizarReporte(String nombreUsuario, int idLibro): void
Actualiza el reporte de préstamo al devolver un libro.
- existeLibro(int idLibro): boolean
Verifica si existe un libro con el ID dado.
- cargarContadorId(): void
Carga el último ID utilizado.
- guardarContadorId(): void

Guarda el último ID utilizado.



CO-SC-CER518726



www.unicesar.edu.co
Balneario Hurtado Vía a Patillal. PBX (57) (5) 5841000 EXT. 1040
Línea de atención al ciudadano 01 8000 400380
Valledupar Cesar Colombia

Getters y setters:

- getAutor(), setAutor(String autor)
- getAnio(), setAnio(int anio)
- getGenero(), setGenero(String genero)

Módulo vista

Documentación del Sistema de Gestión de Biblioteca - Componentes de la Interfaz Gráfica de Usuario (GUI)

1. Clase devoluciones (extends javax.swing.JPanel)

Descripción:

Esta clase representa el panel de la interfaz gráfica para gestionar las devoluciones de libros.

Componentes principales:

- TxtIDLIBRO (JTextField): Campo para ingresar el ID del libro a devolver.
- TxtIDLIBRO1 (JTextField): Campo para ingresar el ID del usuario que devuelve el libro.
- BtnDevolver (JButton): Botón para confirmar la devolución del libro.

Métodos principales:

- devolverLibro(): Maneja la lógica de devolución del libro, interactuando con la clase libro.

2. Clase usuarios (extends javax.swing.JPanel)

Descripción:

Esta clase representa el panel de la interfaz gráfica para gestionar los usuarios del sistema.

Componentes principales:

- TBLUSUARIOS (JTable): Tabla que muestra la lista de usuarios.
- btnEditUser (JButton): Botón para editar un usuario seleccionado.
- btnNuevoUser1 (JButton): Botón para agregar un nuevo usuario.
- btnEliminarUser (JButton): Botón para eliminar un usuario seleccionado.

Métodos principales:

- cargarTabla(): Carga los datos de los usuarios en la tabla.
- btnEliminarUserMouseClicked(): Maneja la eliminación de un usuario.
- btnEditUserMouseClicked(): Maneja la edición de un usuario.

3. Clase dashboard (extends javax.swing.JFrame)

Descripción:

Esta clase representa el panel de control principal de la aplicación.

Componentes principales:

- PnlContent (JPanel): Panel principal donde se cargan los diferentes componentes de la aplicación.
- BtnPrincipal, BtnDevoluciones, BtnUsers, BtnLibros, BtnReportes, BtnPrestamos (JButton): Botones de navegación.

Métodos principales:

- cambiarPanel(JPanel p): Método estático para cambiar el contenido del panel principal.
- initContent(): Inicializa el contenido del dashboard.

4. Clase Prestamos (extends javax.swing.JPanel)

Descripción:

Esta clase representa el panel de la interfaz gráfica para gestionar los préstamos de libros.

Componentes principales:

- TxtIdLibro (JTextField): Campo para ingresar el ID del libro a prestar.
- TxtUsuario (JTextField): Campo para ingresar el nombre del usuario que solicita el préstamo.
- btnPedir (JButton): Botón para confirmar el préstamo del libro.

Métodos principales:

- prestar_libro(): Maneja la lógica de préstamo del libro, interactuando con la clase libro.

5. Clase libros_actuales (extends javax.swing.JPanel)

Descripción:

Esta clase representa el panel de la interfaz gráfica para gestionar el inventario actual de libros.

Componentes principales:

- tblcontenido (JTable): Tabla que muestra la lista de libros.
- BTNAGREGAR, BTNEDITAR, BRNELIMINAR (JButton): Botones para agregar, editar y eliminar libros.
- TxtBuscarLibro (JTextField): Campo para buscar libros por ID.

Métodos principales:

- cargarEnTabla(): Carga los datos de los libros en la tabla.
- cambiar_tabla_por_id(): Filtra la tabla por ID de libro.

- editar_libros(): Maneja la edición de un libro seleccionado.
- eliminar(): Maneja la eliminación de un libro seleccionado.

6. Clase AGREGAR_USUARIO (extends javax.swing.JPanel)

Descripción:

Esta clase representa el panel de la interfaz gráfica para agregar nuevos usuarios al sistema.

Componentes principales:

- TxtNombre, TxtApellido, TxtCedula, TxtContraseña, TxtTelefono, TxtDomicilio (JTextField): Campos para ingresar los datos del nuevo usuario.
- btnAgregarUser (JButton): Botón para confirmar la adición del nuevo usuario.

Métodos principales:

- limpiar(): Limpia todos los campos de texto.
- btnAgregarUserMouseClicked(): Maneja la lógica de agregar un nuevo usuario.

7. Clase principal_vista (extends javax.swing.JPanel)

Descripción:

Esta clase representa la vista principal de la aplicación, mostrando un mensaje de bienvenida.

Componentes principales:

- LblWelcome, LblWelcome1, LblWelcome2, LblWelcome3, LblWelcome4, LblWelcome5 (JLabel): Etiquetas que muestran el mensaje de bienvenida y la descripción del sistema.

8. Clase reportes (extends javax.swing.JPanel)

Descripción:

Esta clase representa el panel de la interfaz gráfica para generar y mostrar reportes.

Componentes principales:

- TBLREPORTES (JTable): Tabla que muestra los reportes de préstamos y devoluciones.

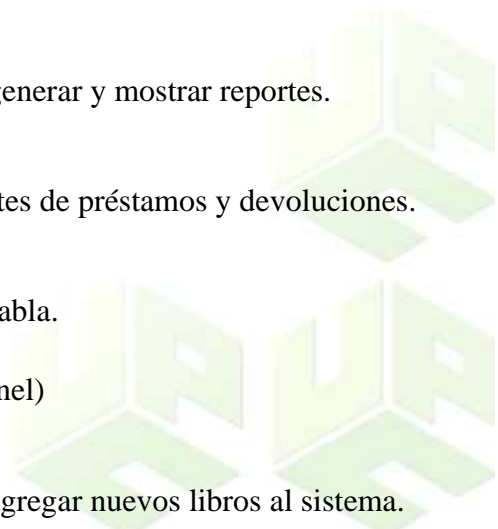
Métodos principales:

- cargarReportes(): Carga los datos de los reportes en la tabla.

9. Clase AGREGAR_LIBROS (extends javax.swing.JPanel)

Descripción:

Esta clase representa el panel de la interfaz gráfica para agregar nuevos libros al sistema.



Componentes principales:

- TxtAutor, TxtGenero, TxtAnio, txtTitulo (JTextField): Campos para ingresar los datos del nuevo libro.
- JComboBox1 (JComboBox): Selector para indicar si el libro está prestado o no.
- btnAgregarUser (JButton): Botón para confirmar la adición del nuevo libro.

Métodos principales:

- limpiar(): Limpia todos los campos de texto.
- btnAgregarUserMouseClicked(): Maneja la lógica de agregar un nuevo libro.

10. Clase Login (extends javax.swing.JFrame)

Descripción:

Esta clase representa la pantalla de inicio de sesión de la aplicación.

Componentes principales:

- TxtUsuario (JTextField): Campo para ingresar el nombre de usuario.
- Password_1 (JPasswordField): Campo para ingresar la contraseña.
- BtnIngresar (JButton): Botón para iniciar sesión.

Métodos principales:

- BtnIngresarActionPerformed(): Maneja la lógica de validación del usuario y contraseña.

11. Clase AGREGAR_LIBROS (extends javax.swing.JPanel)

Descripción:

Esta clase representa el panel de la interfaz gráfica para agregar nuevos libros al sistema.

Componentes principales:

- TxtAutor, TxtGenero, TxtAnio, txtTitulo (JTextField): Campos para ingresar los datos del nuevo libro.
- JComboBox1 (JComboBox): Selector para indicar si el libro está prestado o no.
- btnAgregarUser (JButton): Botón para confirmar la adición del nuevo libro.
- BtnLimpiar (JButton): Botón para limpiar los campos de entrada.

Métodos principales:

- limpiar(): Limpia todos los campos de texto.
- btnAgregarUserMouseClicked(): Maneja la lógica de agregar un nuevo libro.
- BtnLimpiarMouseClicked(): Maneja la acción de limpiar los campos.

12. Clase AGREGAR_USUARIO (extends javax.swing.JPanel)

Descripción:



CO-SC-CER518726



www.unicesar.edu.co
Balneario Hurtado Vía a Patillal. PBX (57) (5) 5841000 EXT. 1040
Línea de atención al ciudadano 01 8000 400380
Valledupar Cesar Colombia

Esta clase representa el panel de la interfaz gráfica para agregar nuevos usuarios al sistema.

Componentes principales:

- TxtNombre, TxtApellido, TxtCedula, TxtContraseña, TxtTelefono, TxtDomicilio (JTextField): Campos para ingresar los datos del nuevo usuario.
- btnAgregarUser (JButton): Botón para confirmar la adición del nuevo usuario.
- BtnLimpiar (JButton): Botón para limpiar los campos de entrada.

Métodos principales:

- limpiar(): Limpia todos los campos de texto.
- btnAgregarUserMouseClicked(): Maneja la lógica de agregar un nuevo usuario.
- BtnLimpiarMouseClicked(): Maneja la acción de limpiar los campos.

13. Clase reportes (extends javax.swing.JPanel)

Descripción:

Esta clase representa el panel de la interfaz gráfica para mostrar reportes de préstamos de libros.

Componentes principales:

- TBLREPORTES (JTable): Tabla que muestra los reportes de préstamos.

Métodos principales:

- cargarReportes(): Carga los datos de los reportes en la tabla desde un archivo.

14. Clase reportes_principal (extends javax.swing.JPanel)

Descripción:

Esta clase representa el panel principal para la sección de reportes, permitiendo cambiar entre diferentes tipos de reportes.

Componentes principales:

- btnEntregados (JButton): Botón para mostrar los libros entregados.
- btnPrestados (JButton): Botón para mostrar los libros prestados.
- panel_contenido (JPanel): Panel donde se muestran los diferentes reportes.

Métodos principales:

- cambiarPanel(JPanel p): Cambia el contenido del panel_contenido.
- btnEntregadosMouseClicked(): Muestra el reporte de libros entregados.
- btnPrestadosMouseClicked(): Muestra el reporte de libros prestados.

15. Clase REPORTES_ENTREGADOS (extends javax.swing.JPanel)

Descripción:



CO-SC-CER518726



www.unicesar.edu.co
Balneario Hurtado Vía a Patillal. PBX (57) (5) 5841000 EXT. 1040
Línea de atención al ciudadano 01 8000 400380
Valledupar Cesar Colombia

Esta clase representa el panel de la interfaz gráfica para mostrar reportes de libros entregados.

Componentes principales:

- TBLREPORTES (JTable): Tabla que muestra los reportes de libros entregados.

Métodos principales:

- cargarReportes(): Carga los datos de los libros entregados en la tabla desde un archivo.

Notas generales:

Notas generales:

- Todas estas clases utilizan componentes de Swing para crear la interfaz gráfica.
- La mayoría de las clases extienden javax.swing.JPanel, lo que quiere decir que son paneles que se pueden integrar en otras ventanas o marcos.
- Hay una fuerte interacción entre estos componentes de la GUI y las clases del modelo (como libro y usuario).
- El sistema sigue un patrón de diseño MVC (Modelo-Vista-Controlador).
- La navegación entre diferentes paneles se maneja principalmente a través de la clase dashboard.

Flujo de la aplicación



CO-SC-CER518726



www.unicesar.edu.co
Balneario Hurtado Vía a Patillal. PBX (57) (5) 5841000 EXT. 1040
Línea de atención al ciudadano 01 8000 400380
Valledupar Cesar Colombia

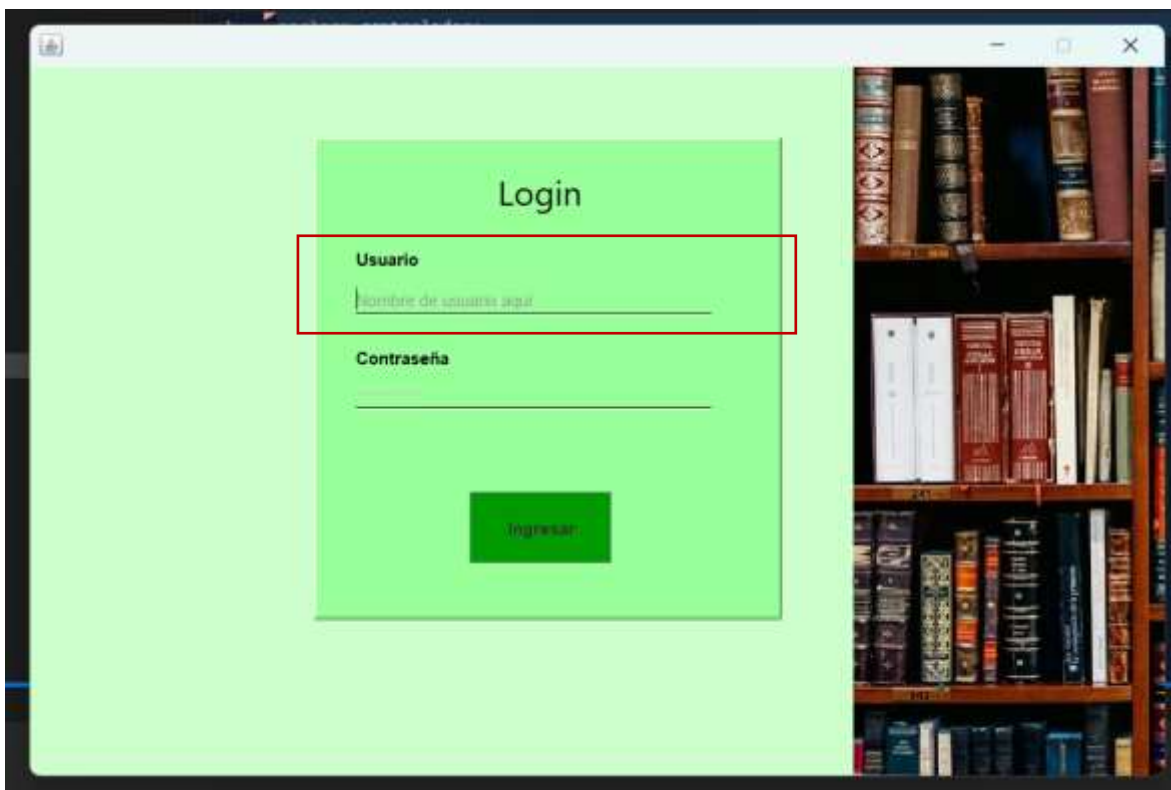
1. Página de Inicio de Sesión (Login)

Descripción:

Esta es la primera pantalla que el usuario ve al iniciar la aplicación. Permite a los usuarios autenticarse para acceder al sistema.

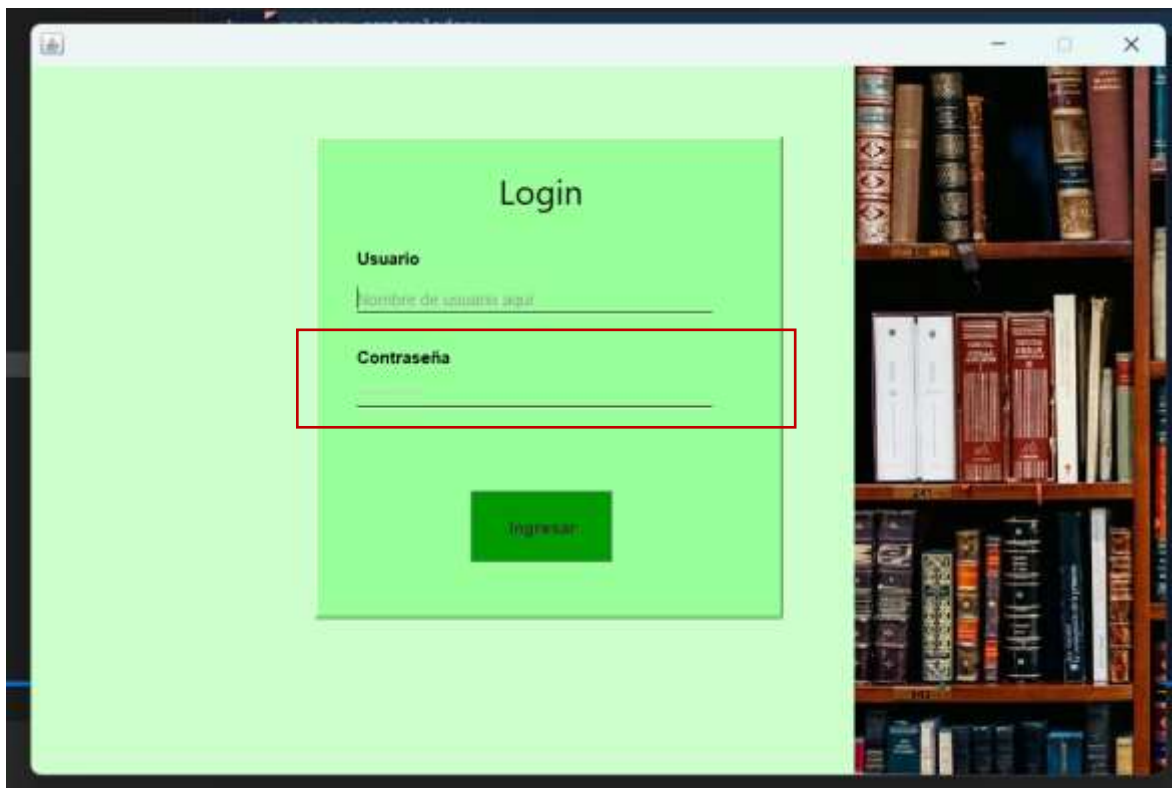
Campo de Usuario:

- Inicialmente muestra "Nombre de usuario aquí" en gris claro.
- Al hacer clic, este texto desaparece para que el usuario pueda escribir su nombre de usuario.



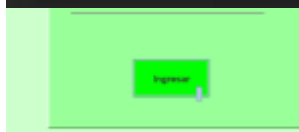
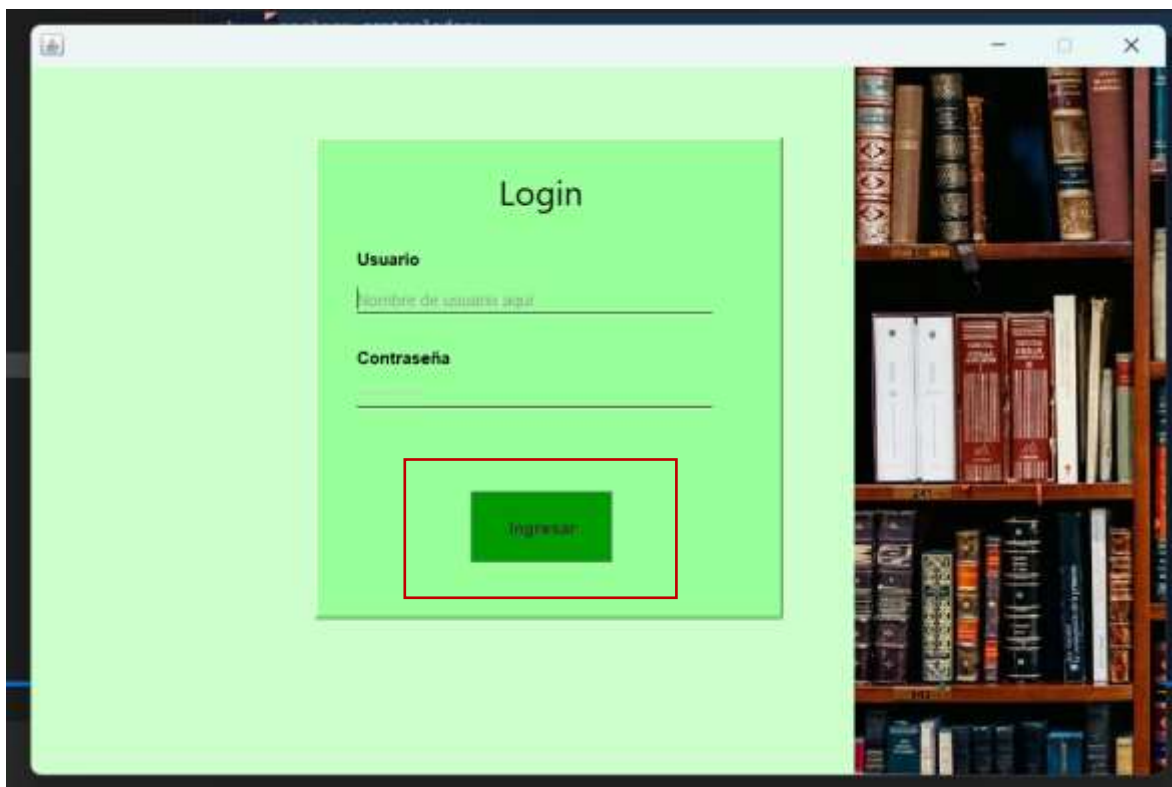
Campo de Contraseña:

- Inicialmente muestra "clavejaja" en gris claro como texto de ejemplo.
- Al hacer clic, este texto desaparece para que el usuario pueda escribir su contraseña de forma segura (los caracteres aparecerán como puntos por seguridad).



Botón "Ingresar":

- Es el botón principal para iniciar sesión.
- Cambia a un color verde más brillante cuando el usuario pasa el mouse por encima, indicando que es interactivo.
- Al hacer clic, intenta iniciar sesión con las credenciales proporcionadas.



Respuesta del sistema:

- Si las credenciales son correctas, el usuario verá un mensaje de bienvenida y será dirigido al panel principal de la aplicación.



- Si las credenciales son incorrectas, se mostrará un mensaje de error indicando que el usuario o la contraseña son incorrectos.



Elementos visuales:

- La página tiene un fondo verde claro que da una sensación de tranquilidad.
- Hay una imagen de libros en el lado derecho, que aunque no es interactiva, proporciona contexto visual sobre el propósito de la aplicación.

2. Panel Principal (Dashboard)

Descripción:

Esta es la pantalla principal de la aplicación, que proporciona acceso a todas las funcionalidades del sistema.

Interacciones del usuario:

- Menú lateral con botones para acceder a diferentes secciones:

- * Principal
- * Préstamos
- * Devoluciones
- * Usuarios
- * Libros
- * Reportes

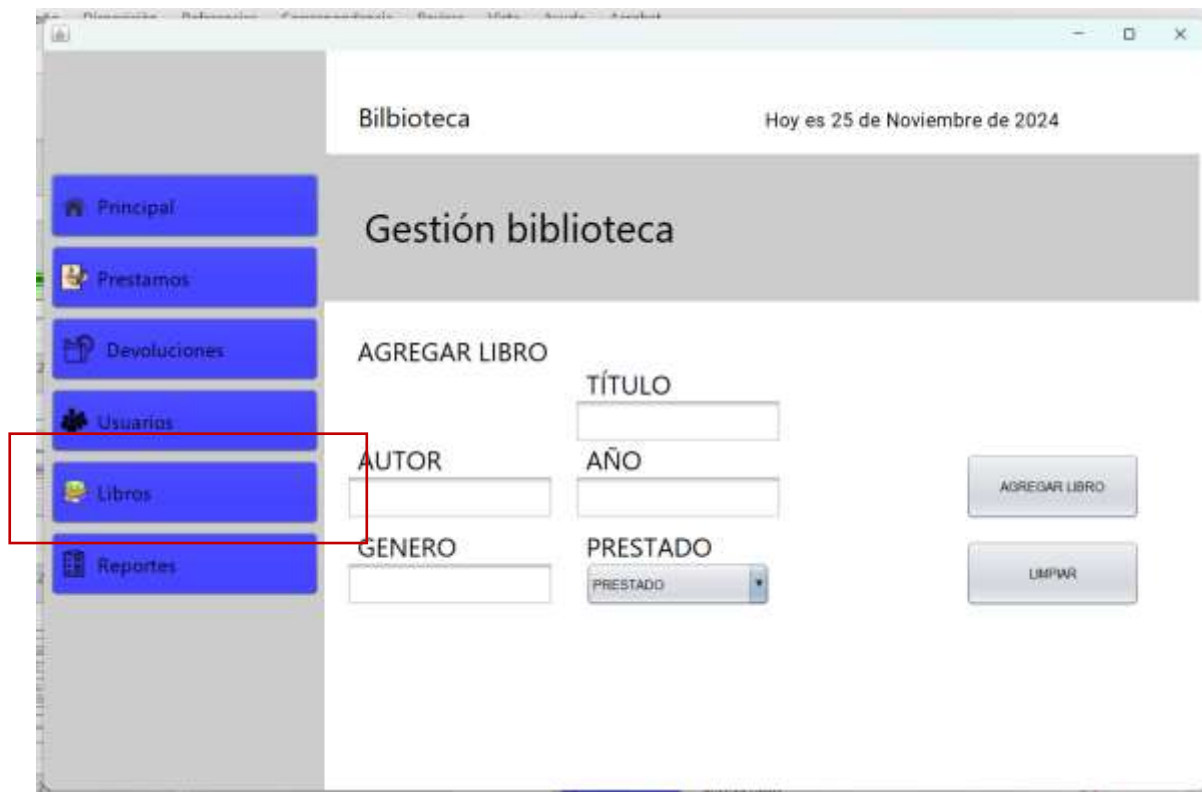
- Área principal de contenido que cambia según la sección seleccionada.



3. Gestión de Libros (AGREGAR_LIBROS y libros_actuales)

Descripción:

Estas pantallas permiten a los usuarios agregar nuevos libros al sistema y gestionar el inventario actual.



Interacciones del usuario en AGREGAR_LIBROS:

- Campos para ingresar detalles del libro (título, autor, género, año, etc.).
- Botón "AGREGAR LIBRO" para guardar el nuevo libro en el sistema.
- Botón "LIMPIAR" para borrar los campos de entrada.



Biblioteca Hoy es 25 de Noviembre de 2024

Principal Prestamos Devoluciones Usuarios Libros Reportes

Gestión biblioteca

AGREGAR LIBRO

TÍTULO

AUTOR AÑO

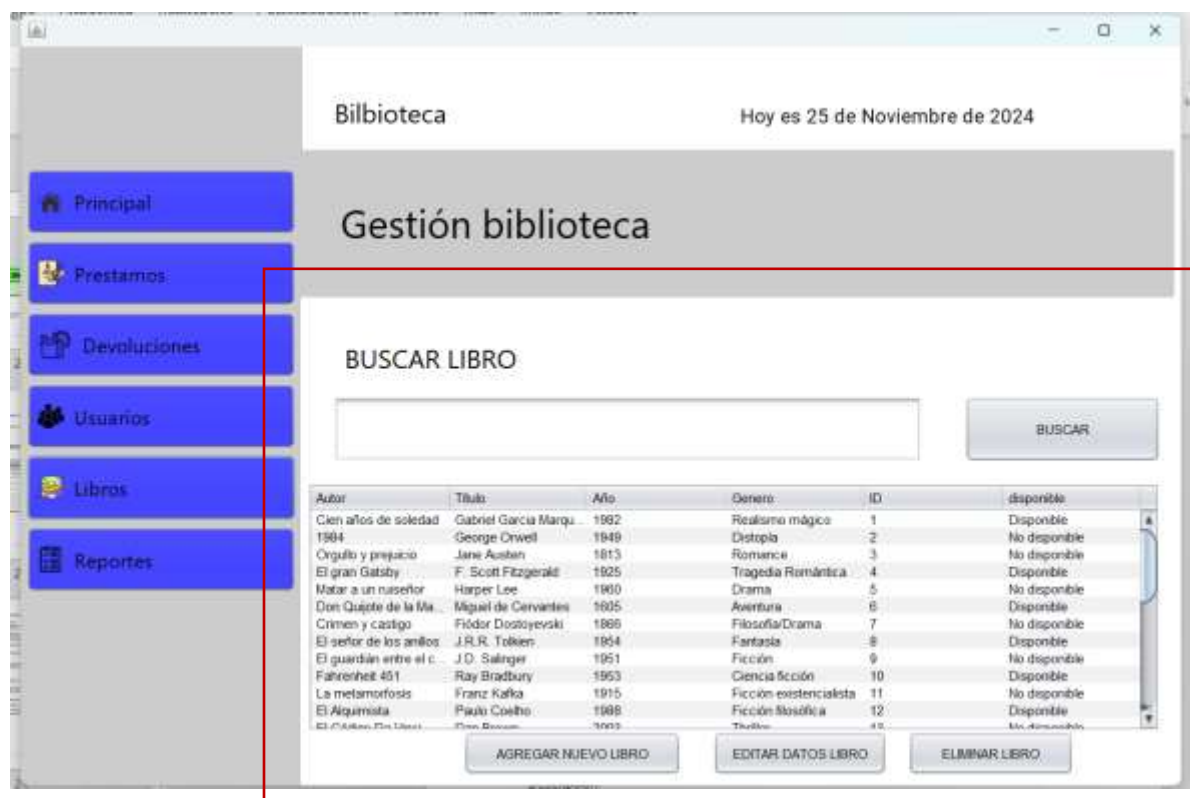
GENERO PRESTADO

AGREGAR LIBRO

LIMPIAR

Interacciones del usuario en libros_actuales:

- Tabla que muestra el inventario actual de libros.
- Campo de búsqueda para filtrar libros por ID.
- Botones para agregar, editar y eliminar libros.



4. Gestión de Usuarios (AGREGAR_USUARIO y usuarios)

Descripción:

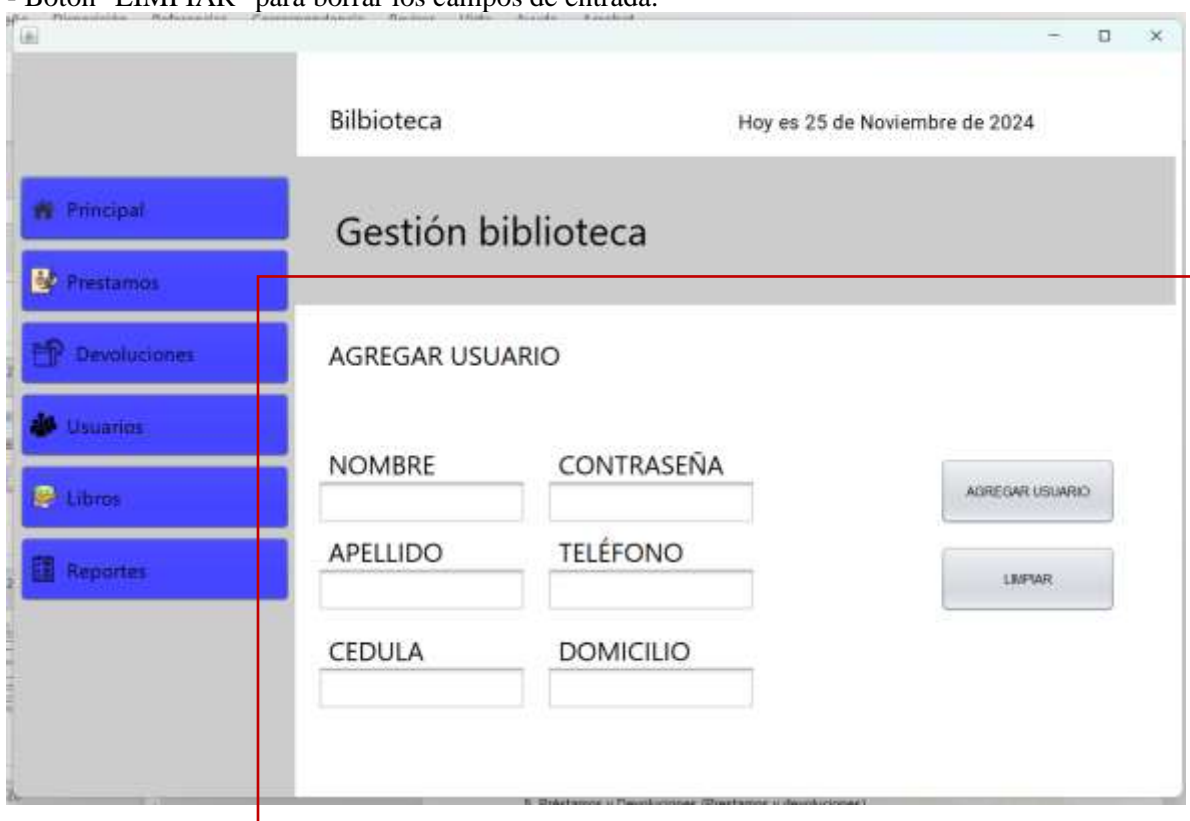
Estas pantallas permiten agregar nuevos usuarios al sistema y gestionar los usuarios existentes.



Nombre	Apellido	Cédula	Telefono	Domicilio
admin	user_admin	4321	1234	casa admin
Libri	Tones	1121530505	3234291012	mz 45 casa 1 belavista
HECTOR	CARRILLO	31321321	321321312	vive debajo de un puente
Carlos	Martinez	1122334455	3115678901	Carrera 12 #34-56
Ana	López	1123456789	3102345678	Av Siempreviva #742
Maria	Gonzalez	1128678543	3201234567	Calle 10 #20-30
Juan	Ramirez	1121123344	3136789012	Barrio La Esperanza
Luisa	Mejia	1122345678	3225678901	Barrio Las Flores
Anaí	Pérez	1125567788	3123456789	Calle 50 #60-70
Diana	Gómez	1120677889	3156789012	Conjunto Residencial Sol
Pedro	Fernández	1127789900	3191234567	Carrera 15 #10-17
Sofia	Rojas	1128999001	3103456789	Edificio El Mirador
Andrés	Ortiz	1129001122	3167890123	Calle 25 #20-27
Laura	Morales	1130012233	3145678901	Casa Verde
Miguel	Chavez	1131123344	3178901234	Torre Central Apto 101
Valentina	Rivera	1132234455	3189012345	Carrera 40 #50-51
Cristian	Castro	1133345566	3212345678	Barrio El Porvenir
Camila	Mendoza	1134456677	3234567890	Calle Real #100

Interacciones del usuario en AGREGAR_USUARIO:

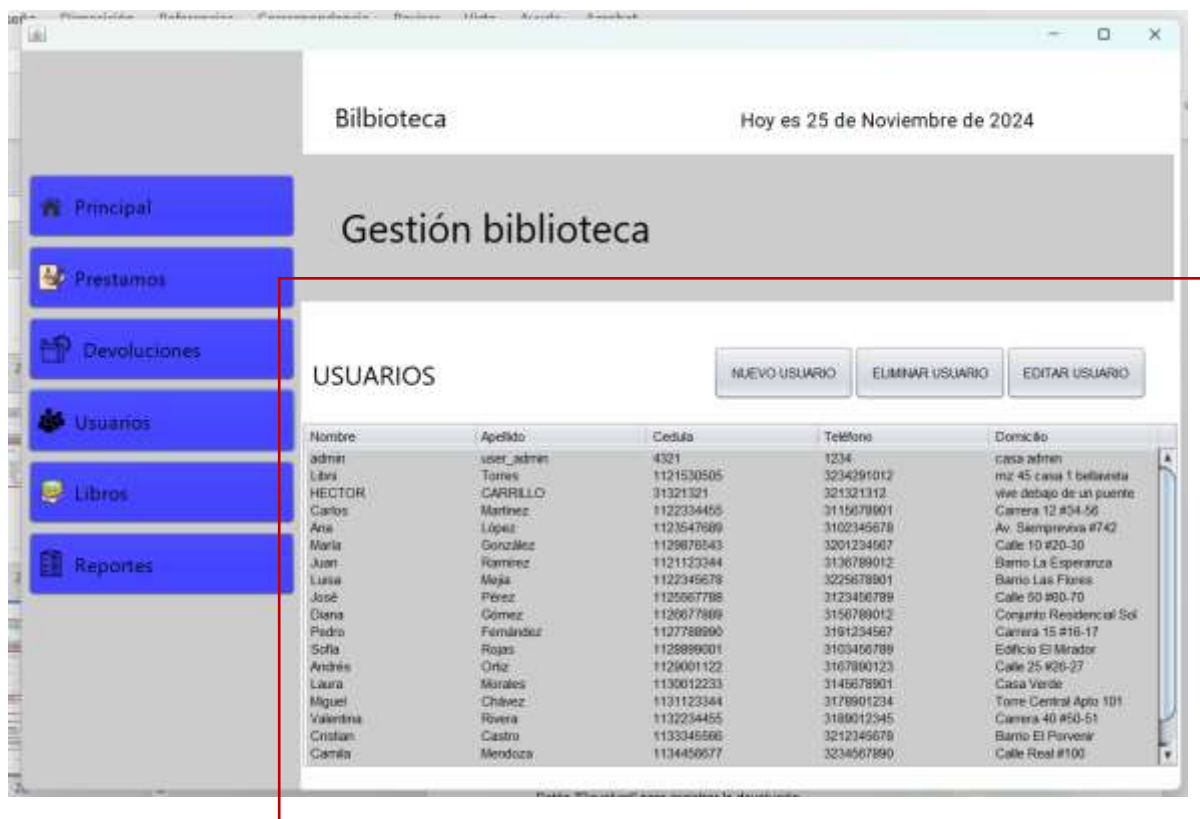
- Campos para ingresar detalles del usuario (nombre, apellido, cédula, etc.).
- Botón "AGREGAR USUARIO" para guardar el nuevo usuario en el sistema.
- Botón "LIMPIAR" para borrar los campos de entrada.



The screenshot shows a web application window titled 'Biblioteca' with a date 'Hoy es 25 de Noviembre de 2024'. On the left is a sidebar menu with options: Principal, Prestamos, Devoluciones, Usuarios, Libros, and Reportes. The main content area is titled 'Gestión biblioteca' and contains a form titled 'AGREGAR USUARIO'. The form has six input fields arranged in two columns: NOMBRE, CONTRASEÑA, APELLIDO, TELÉFONO, CEDULA, and DOMICILIO. To the right of the fields are two buttons: 'AGREGAR USUARIO' and 'LIMPIAR'. A red rectangular box highlights the entire form area.

Interacciones del usuario en usuarios:

- Tabla que muestra la lista de usuarios registrados.
- Botones para editar y eliminar usuarios.



The screenshot shows a web application interface for library management. On the left is a sidebar with navigation links: Principal, Prestamos, Devoluciones, Usuarios, Libros, and Reportes. The main content area is titled 'Gestión biblioteca' and includes a date 'Hoy es 25 de Noviembre de 2024'. Below the title is a section for 'USUARIOS' with three buttons: 'NUEVO USUARIO', 'ELIMINAR USUARIO', and 'EDITAR USUARIO'. A table displays the list of registered users with columns for Nombre, Apellido, Cedula, Telefono, and Domicilio.

Nombre	Apellido	Cedula	Telefono	Domicilio
admin	user_admin	4321	1234	casa admin
Lina	Torres	1121530505	3234291012	mz 45 casa 1 bellavista
HECTOR	CARRILLO	31321321	321321312	vive debajo de un puente
Carlos	Martinez	1122334455	3115078901	Camera 12 #34-56
Ana	López	1123547689	3102345678	Av. Sempreviva #742
Maria	González	1129876543	3201234567	Calle 10 #20-30
Juan	Romero	1121123344	3136789012	Barrio La Esperanza
Luisa	Mejía	1122345678	3225678901	Barrio Las Flores
José	Pérez	1125607788	3123456789	Calle 50 #50-70
Diana	Gómez	1126077889	3150789012	Conjunto Residencial Sol
Pedro	Fernández	1127788990	3191234567	Camera 15 #16-17
Sofía	Rojas	1128899001	3103400789	Edificio El Mirador
Andrés	Ortiz	1129001122	3167890123	Calle 25 #26-27
Laura	Murales	1130012233	3145678901	Casa Verde
Miguel	Chavez	1131123344	3178901234	Torre Central Apto 101
Valentina	Rivera	1132234455	3189012345	Camera 40 #50-51
Cristian	Castro	1133345566	3212345678	Barrio El Porvenir
Camila	Mendoza	1134456677	3234567890	Calle Real #100

5. Préstamos y Devoluciones (Prestamos y devoluciones)

Descripción:

Estas pantallas gestionan el proceso de préstamo y devolución de libros.



Interacciones del usuario en Prestamos:

- Campos para ingresar el ID del libro y el nombre del usuario.
- Botón "Pedir" para registrar el préstamo.



Interacciones del usuario en devoluciones:

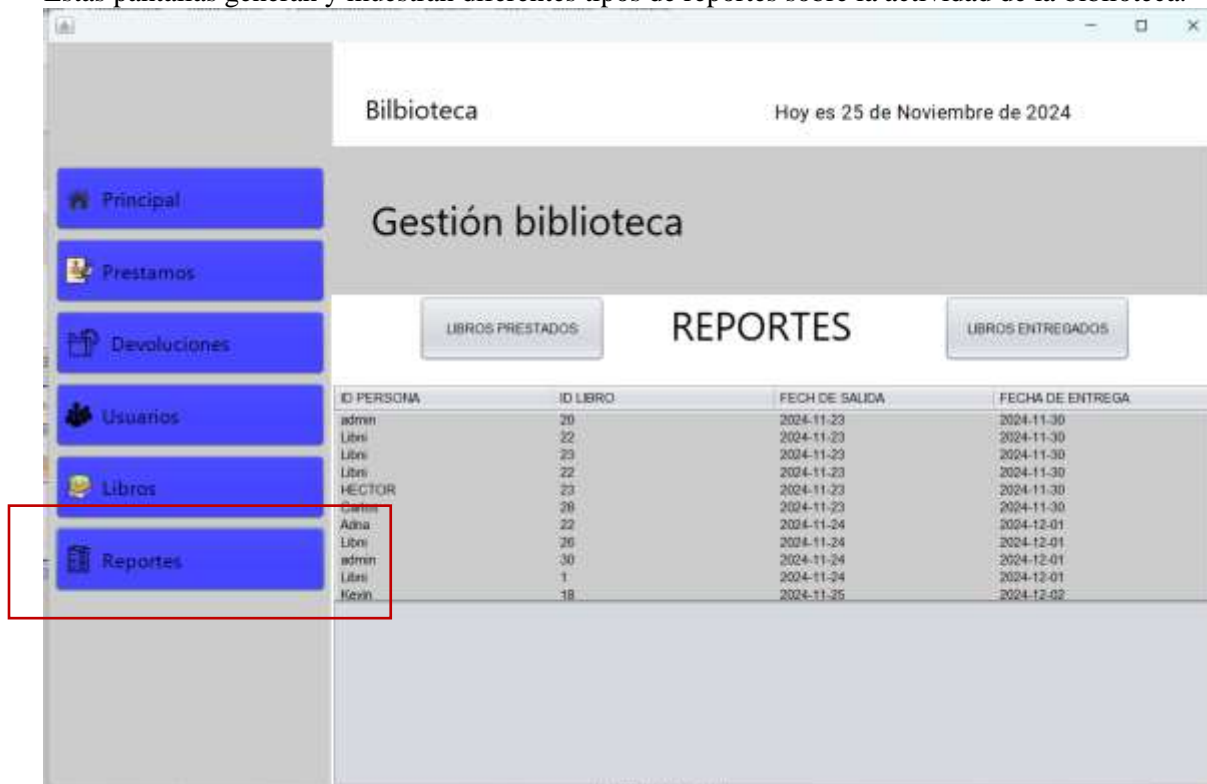
- Campos para ingresar el ID del libro y el ID del usuario.
- Botón "Devolver" para registrar la devolución.



6. Reportes (reportes y REPORTES_ENTREGADOS)

Descripción:

Estas pantallas generan y muestran diferentes tipos de reportes sobre la actividad de la biblioteca.



ID PERSONA	ID LIBRO	FECH DE SALIDA	FECHA DE ENTREGA
admin	20	2024-11-23	2024-11-30
Libri	22	2024-11-23	2024-11-30
Libri	23	2024-11-23	2024-11-30
Libri	22	2024-11-23	2024-11-30
HECTOR	23	2024-11-23	2024-11-30
Carlin	26	2024-11-23	2024-11-30
Adina	22	2024-11-24	2024-12-01
Libri	26	2024-11-24	2024-12-01
admin	30	2024-11-24	2024-12-01
Libri	1	2024-11-24	2024-12-01
Kevin	18	2024-11-25	2024-12-02

Interacciones del usuario:

- Tablas que muestran información sobre préstamos, devoluciones y libros entregados.
- Posibilidad de ver detalles específicos de cada reporte.

REPORTES			
LIBROS PRESTADOS		LIBROS ENTREGADOS	
ID PERSONA	ID LIBRO	FECH DE SALIDA	FECHA DE ENTREGA
admin	20	2024-11-23	2024-11-30
Libri	22	2024-11-23	2024-11-30
Libri	23	2024-11-23	2024-11-30
Libri	22	2024-11-23	2024-11-30
HECTOR	23	2024-11-23	2024-11-30
Carlos	28	2024-11-23	2024-11-30
Adna	22	2024-11-24	2024-12-01
Libri	26	2024-11-24	2024-12-01
admin	30	2024-11-24	2024-12-01
Libri	1	2024-11-24	2024-12-01
Kevin	18	2024-11-25	2024-12-02

REPORTES		
LIBROS PRESTADOS		LIBROS ENTREGADOS
ID PERSONA	ID LIBRO	DÍA ENTREGADO
Libri	19	2024-11-23
HECTOR	23	2024-11-23
admin	1	2024-11-23
Carlos	28	2024-11-23
Adna	22	2024-11-24
Libri	26	2024-11-24
admin	30	2024-11-24
Libri	1	2024-11-24
Kevin	18	2024-11-25

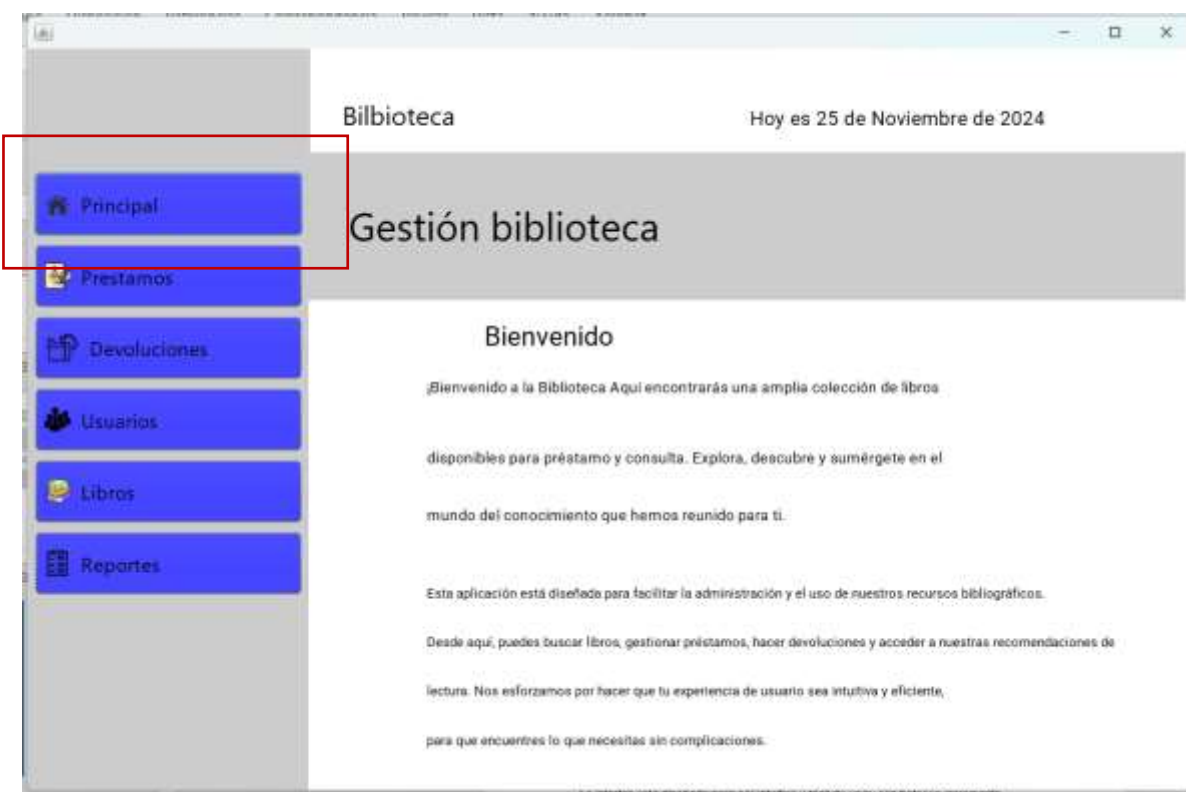
7. Vista Principal (principal_vista)

Descripción:

Esta es la pantalla de bienvenida que se muestra cuando el usuario accede al sistema.

Interacciones del usuario:

- Muestra información general sobre el sistema de biblioteca.
- No tiene interacciones directas, pero proporciona una visión general del sistema.



Notas generales:

- Todas las pantallas incluyen validaciones para asegurar que los datos ingresados sean correctos y completos.
- El sistema proporciona retroalimentación al usuario a través de mensajes de éxito o error después de cada acción.
- La interfaz está diseñada para ser intuitiva y fácil de usar, con botones claramente etiquetados y campos de entrada bien definidos.

Pruebas realizadas

Informe de Pruebas - Sistema de Gestión de Biblioteca

Fecha de ejecución: [23/11/2024]

Resumen:

- Total de pruebas ejecutadas: 4
- Pruebas exitosas: 4
- Pruebas fallidas: 0

Resultados detallados:

Pruebas de Usuario:

1. Crear usuario: [OK]
2. Leer usuario: [OK]
3. Actualizar usuario: [OK]
4. Eliminar usuario: [OK]

Pruebas de Libro:

1. Crear libro: [OK]
2. Leer libro: [OK]
3. Actualizar libro: [OK]
4. Eliminar libro: [OK/]
5. Prestar libro: [OK]
6. Devolver libro: [OK]

Observaciones adicionales:

- El código funciona bien, aunque hay funcionalidades nuevas que se le podrían añadir

Conclusiones:

Sistema estable: Todos los casos de prueba ejecutados para las funcionalidades clave del sistema (usuario y libro) fueron exitosos, lo que indica que las operaciones fundamentales, como la creación, lectura, actualización y eliminación de usuarios y libros, están funcionando correctamente.

Comportamiento esperado: El sistema está respondiendo de acuerdo con lo esperado para los procesos de préstamo y devolución de libros, lo que muestra que la funcionalidad principal está operativa y sin errores.

Interfaz sencilla y eficiente: La interfaz gráfica de usuario parece estar funcionando sin inconvenientes, y no se reportaron fallos relacionados con la interacción del usuario en el sistema.

Recomendaciones:

Añadir pruebas de seguridad: Aunque no se realizaron pruebas de seguridad (como la validación de contraseñas, protección contra inyecciones SQL, o el manejo de contraseñas encriptadas), es fundamental integrar estas pruebas para garantizar la protección de los datos sensibles del usuario.

Pruebas de rendimiento: Realizar pruebas de carga y rendimiento, especialmente para manejar múltiples usuarios simultáneamente, puede ayudar a identificar cuellos de botella o posibles fallos bajo alta demanda.



Validación de datos: Implementar validaciones adicionales en el lado del cliente y servidor, como verificar que los datos introducidos por el usuario sean válidos antes de guardarlos (por ejemplo, evitar caracteres especiales o campos vacíos) para prevenir errores o datos corruptos.

Pruebas de funcionalidades adicionales: Evaluar funcionalidades avanzadas, como la posibilidad de realizar búsquedas avanzadas de libros o usuarios, filtros por género o fecha de publicación, y la gestión de reservas de libros.

Automatización de pruebas: Considerar el uso de herramientas de automatización de pruebas, como JUnit o Selenium, para realizar pruebas repetitivas o en diferentes entornos sin intervención manual, lo que aumentaría la eficiencia y cobertura de las pruebas.

Pruebas de compatibilidad: Asegurarse de que el sistema sea compatible con diferentes navegadores, versiones del sistema operativo y dispositivos, para garantizar una experiencia de usuario coherente.

Mejorar la interfaz gráfica: Si se considera necesario, mejorar la interfaz de usuario, agregando elementos como mensajes de error más claros, confirmaciones visuales de acciones (como mensajes de éxito o error después de agregar o eliminar libros/usuarios), y mejoras en la accesibilidad.



Código utilizado para las pruebas

```
// LibroTest.java
package prueba;

import modelo.libro;
import java.util.List;

public class LibroTest {

    public void ejecutarPruebas() {

        System.out.println("\nPruebas de Libro:");

        testCrearLibro();

        testLeerLibro();

        testActualizarLibro();

        testEliminarLibro();

        testPrestarLibro();

        testDevolverLibro();

    }

    private void testCrearLibro() {

        String titulo = "Libro de Prueba";

        boolean resultado = libro.crearLibro("Autor Prueba", 2023, "Género Prueba", titulo, false);

        System.out.println("- Crear libro: " + (resultado ? "OK" : "Fallo"));

        // Limpieza: Eliminar el libro de prueba

        if (resultado) {

            List<libro> libros = libro.cargarLibros();

            libros.stream()

                .filter(l -> l.getTitulo().equals(titulo))

                .findFirst()

                .ifPresent(l -> libro.eliminarLibro(l.getId()));

        }

    }

    private void testLeerLibro() {

        List<libro> libros = libro.cargarLibros();

        boolean resultado = !libros.isEmpty();

        System.out.println("- Leer libros: " + (resultado ? "OK" : "Fallo"));

    }

    private void testActualizarLibro() {

        // Crear un libro temporal para la prueba

        String tituloOriginal = "Libro Temporal";
```



CO-SC-CER518726



www.unicesar.edu.co
Balneario Hurtado Vía a Patillal. PBX (57) (5) 5841000 EXT. 1040
Línea de atención al ciudadano 01 8000 400380
Valledupar Cesar Colombia

```
libro.crearLibro("Autor Temporal", 2023, "Género Temporal", tituloOriginal, false);

List<libro> libros = libro.cargarLibros();
libro libroTemp = libros.stream()
    .filter(l -> l.getTitulo().equals(tituloOriginal))
    .findFirst()
    .orElse(null);

if (libroTemp != null) {
    String nuevoTitulo = "Libro Actualizado";
    libro.editarLibro(libroTemp.getId(), nuevoTitulo, "Autor Actualizado", 2024, "Género Actualizado");

    libros = libro.cargarLibros();
    boolean resultado = libros.stream()
        .anyMatch(l -> l.getTitulo().equals(nuevoTitulo));
    System.out.println("- Actualizar libro: " + (resultado ? "OK" : "Fallo"));

    // Limpieza: Eliminar el libro temporal
    libro.eliminarLibro(libroTemp.getId());
} else {
    System.out.println("- Actualizar libro: Fallo (No se pudo crear libro temporal)");
}

private void testEliminarLibro() {
    // Crear un libro temporal para la prueba
    String tituloEliminar = "Libro para Eliminar";
    libro.crearLibro("Autor Eliminar", 2023, "Género Eliminar", tituloEliminar, false);

    List<libro> libros = libro.cargarLibros();
    libro libroEliminar = libros.stream()
        .filter(l -> l.getTitulo().equals(tituloEliminar))
        .findFirst()
        .orElse(null);

    if (libroEliminar != null) {
        int idEliminar = libroEliminar.getId();
        libro.eliminarLibro(idEliminar);

        libros = libro.cargarLibros();
        boolean resultado = libros.stream()
            .noneMatch(l -> l.getId() == idEliminar);
        System.out.println("- Eliminar libro: " + (resultado ? "OK" : "Fallo"));
```



CO-SC-CER518726



www.unicesar.edu.co
Balneario Hurtado Vía a Patillal. PBX (57) (5) 5841000 EXT. 1040
Línea de atención al ciudadano 01 8000 400380
Valledupar Cesar Colombia

```
} else {  
    System.out.println("- Eliminar libro: Fallo (No se pudo crear libro temporal)");  
}  
}  
  
private void testPrestarLibro() {  
    // Usar un libro existente que no esté prestado  
    List<libro> libros = libro.cargarLibros();  
    libro libroPrestamo = libros.stream()  
        .filter(l -> !l.getPrestado())  
        .findFirst()  
        .orElse(null);  
  
    if (libroPrestamo != null) {  
        libro libroObj = new libro();  
        boolean resultado = libroObj.prestar(libroPrestamo.getId(), "UsuarioPrueba");  
        System.out.println("- Prestar libro: " + (resultado ? "OK" : "Fallo"));  
  
        // Devolver el libro para no afectar el estado  
        if (resultado) {  
            libroObj.devolver(libroPrestamo.getId(), "UsuarioPrueba");  
        }  
    } else {  
        System.out.println("- Prestar libro: No se pudo realizar (No hay libros disponibles para prestar)");  
    }  
}  
  
private void testDevolverLibro() {  
    // Primero prestar un libro para luego devolverlo  
    List<libro> libros = libro.cargarLibros();  
    libro libroDevolucion = libros.stream()  
        .filter(l -> !l.getPrestado())  
        .findFirst()  
        .orElse(null);  
  
    if (libroDevolucion != null) {  
        libro libroObj = new libro();  
        libroObj.prestar(libroDevolucion.getId(), "UsuarioPrueba");  
        boolean resultado = libroObj.devolver(libroDevolucion.getId(), "UsuarioPrueba");  
        System.out.println("- Devolver libro: " + (resultado ? "OK" : "Fallo"));  
    } else {  
        System.out.println("- Devolver libro: No se pudo realizar (No hay libros disponibles para la prueba)");  
    }  
}
```



CO-SC-CER518726



www.unicesar.edu.co
Balneario Hurtado Vía a Patillal. PBX (57) (5) 5841000 EXT. 1040
Línea de atención al ciudadano 01 8000 400380
Valledupar Cesar Colombia

```
}  
}
```

```
// UsuarioTest.java  
  
package prueba;  
  
import modelo.usuario;  
import java.util.List;  
  
public class UsuarioTest {  
  
    public void ejecutarPruebas() {  
  
        System.out.println("Pruebas de usuario:");  
  
        testCrearUsuario();  
  
        testLeerUsuario();  
  
        testActualizarUsuario();  
  
        testEliminarUsuario();  
  
    }  
  
    private void testCrearUsuario() {  
  
        String nombre = "UsuarioPrueba";  
  
        String contraseña = "clavePrueba";  
  
        String cedula = "9999999999";  
  
        String apellido = "ApellidoPrueba";  
  
        long telefono = 9999999999L;  
  
        String domicilio = "Dirección de prueba";  
  
  
        usuario.agregarUsuario(nombre, contraseña, cedula, apellido, telefono, domicilio);  
  
        boolean resultado = usuario.existeUsuario(nombre);  
  
        System.out.println("- Crear usuario: " + (resultado ? "OK" : "Fallo"));  
  
  
        // Limpieza: Eliminar el usuario de prueba  
  
        if (resultado) {  
  
            usuario.eliminarUsuario(cedula);  
  
        }  
  
    }  
  
    private void testLeerUsuario() {  
  
        List<usuario> usuarios = usuario.cargarUsuarios();  
  
        boolean resultado = !usuarios.isEmpty();  
  
        System.out.println("- Leer usuarios: " + (resultado ? "OK" : "Fallo"));  
  
    }  
}
```



CO-SC-CER518726



www.unicesar.edu.co
Balneario Hurtado Vía a Patillal. PBX (57) (5) 5841000 EXT. 1040
Línea de atención al ciudadano 01 8000 400380
Valledupar Cesar Colombia

```
private void testActualizarUsuario() {
    // Crear un usuario temporal para la prueba
    String cedula = "8888888888";
    String nombre = "UsuarioTemp";
    usuario.agregarUsuario(nombre, "claveTemp", cedula, "ApellidoTemp", 8888888888L, "DirecciónTemp");

    String nuevoNombre = "UsuarioTempActualizado";
    String nuevoApellido = "ApellidoTempActualizado";
    String nuevaContraseña = "nuevaClaveTemp";
    long nuevoTelefono = 7777777777L;
    String nuevoDomicilio = "NuevaDirecciónTemp";

    usuario.editarUsuario(cedula, nuevoNombre, nuevoApellido, nuevaContraseña, nuevoTelefono, nuevoDomicilio);
    List<usuario> usuarios = usuario.cargarUsuarios();
    boolean resultado = usuarios.stream()
        .anyMatch(u -> u.getNombre().equals(nuevoNombre) && u.getApellido().equals(nuevoApellido));
    System.out.println("- Actualizar usuario: " + (resultado ? "OK" : "Fallo"));

    // Limpieza: Eliminar el usuario temporal
    usuario.eliminarUsuario(cedula);
}

private void testEliminarUsuario() {
    // Crear un usuario temporal para la prueba
    String cedula = "7777777777";
    String nombre = "UsuarioEliminar";
    usuario.agregarUsuario(nombre, "claveEliminar", cedula, "ApellidoEliminar", 7777777777L, "DirecciónEliminar");

    usuario.eliminarUsuario(cedula);
    List<usuario> usuarios = usuario.cargarUsuarios();
    boolean resultado = usuarios.stream()
        .noneMatch(u -> u.getCedula().equals(cedula));
    System.out.println("- Eliminar usuario: " + (resultado ? "OK" : "Fallo"));
}
}
```

```
package prueba;
```



CO-SC-CER518726



www.unicesar.edu.co
Balneario Hurtado Vía a Patillal. PBX (57) (5) 5841000 EXT. 1040
Línea de atención al ciudadano 01 8000 400380
Valledupar Cesar Colombia


```
public class MainTest {  
    public static void main(String[] args) {  
        System.out.println("Iniciando pruebas de la aplicación de gestión de biblioteca");  
        System.out.println("-----");  
  
        UsuarioTest usuarioTest = new UsuarioTest();  
        usuarioTest.ejecutarPruebas();  
  
        LibroTest libroTest = new LibroTest();  
        libroTest.ejecutarPruebas();  
  
        System.out.println("\n-----");  
        System.out.println("Finalizando pruebas de la aplicación de gestión de biblioteca");  
    }  
}
```



Consideraciones finales

Limitaciones:

Funcionalidades no implementadas:

Aunque el sistema cumple con los requisitos básicos de gestión de libros, usuarios y préstamos, existen algunas funcionalidades que no se han implementado, tales como la opción de búsqueda avanzada de libros por categorías, autor, o palabras clave. Además, la integración con servicios externos (como un sistema de reservas de libros en línea) es una característica aún pendiente.

Interfaz gráfica:

Aunque la interfaz gráfica es funcional y fácil de usar, la estética podría mejorar con el uso de un diseño más moderno, como el uso de temas visuales dinámicos o elementos interactivos más elaborados.

Escalabilidad:

Actualmente, la aplicación no está diseñada para manejar un gran volumen de usuarios o libros. En futuros desarrollos, sería necesario optimizar la base de datos y la arquitectura del sistema para mejorar el rendimiento bajo carga alta.

Seguridad:

El sistema carece de funcionalidades avanzadas de seguridad, como el cifrado de contraseñas o autenticación de múltiples factores. Esto podría ser una limitación si se desea utilizar la aplicación en un entorno con mayores riesgos de seguridad.

Posibles mejoras:

Implementación de búsqueda avanzada:

Se podría agregar una barra de búsqueda con filtros para que los usuarios puedan buscar libros por autor, categoría, año, etc. Esto facilitaría la gestión de grandes bibliotecas.

Notificaciones:

Implementar un sistema de notificaciones para recordar a los usuarios sobre los préstamos vencidos o los libros que deben devolver. Además, se podrían enviar alertas cuando un libro reservado esté disponible.

Mejora en la interfaz gráfica:

La aplicación podría beneficiarse de un diseño más atractivo y moderno, utilizando bibliotecas como JavaFX para crear interfaces más interactivas, con mejor experiencia de usuario.

Optimización de la base de datos:

La base de datos actual podría optimizarse para manejar mejor el aumento del número de usuarios y libros, implementando índices o migrando a un sistema más robusto si es necesario.



CO-SC-CER518726



www.unicesar.edu.co
Balneario Hurtado Vía a Patillal. PBX (57) (5) 5841000 EXT. 1040
Línea de atención al ciudadano 01 8000 400380
Valledupar Cesar Colombia

Seguridad:

La implementación de un sistema de cifrado para las contraseñas de los usuarios, así como la autenticación de dos factores, podría mejorar considerablemente la seguridad del sistema.



Conclusión

Este proyecto ha sido una oportunidad para desarrollar una solución completa de gestión de biblioteca, abordando las necesidades de control de libros, usuarios y préstamos en un sistema intuitivo y fácil de usar. Durante su desarrollo, se logró cumplir con los objetivos planteados inicialmente, que incluían la creación de un sistema capaz de gestionar las operaciones básicas de préstamo y devolución de libros, la administración de usuarios y la generación de reportes.

Los objetivos específicos de la aplicación se cumplieron satisfactoriamente, destacando la implementación de una interfaz gráfica accesible y la capacidad de gestionar la información de manera eficiente. Además, se logró crear una base de datos estructurada que soporta las operaciones necesarias para el funcionamiento de la biblioteca, mejorando la organización y accesibilidad de los datos.

Sin embargo, se identificaron varias áreas de mejora, tanto en términos de funcionalidades adicionales como de optimización de la seguridad y el rendimiento del sistema. Las lecciones aprendidas incluyen la importancia de priorizar la usabilidad en la interfaz, la necesidad de un sistema más seguro para proteger los datos de los usuarios, y la necesidad de pensar en la escalabilidad a medida que el sistema crece.

Este proyecto ha sido valioso no solo como una solución práctica para la gestión de bibliotecas, sino también como una base para futuras mejoras y ampliaciones que pueden convertirla en una herramienta aún más poderosa y robusta.

