

**UNIVERSIDAD PERUANA LOS ANDES**  
**FACULTAD DE INGENIERIA**  
**ESCUELA PROFECIONAL DE INGENIERIA DE**  
**SISTEMAS Y COMPUTACIÓN**



**TEMA**

---

**Diseño e Implementación de un sistema de login para  
una Biblioteca**

---

CURSO: Desarrollo de Aplicaciones

DOCENTE: Ing. Fernandez Bejarano Raul Enrique

INTEGRANTES:

- Ramón Puente Eduardo Fredy
- Untiveros Moreno Diego Alonso

CICLO: IV

SECCION: A1

**HUANCAYO - PERÚ**

# 2025

## Índice

<b>Índice .....</b>	2
<b>Introducción.....</b>	3
<b>Objetivos específicos .....</b>	3
<b>Justificación .....</b>	3
<b>Descripción del proyecto .....</b>	3
<b>Funcionalidades principales .....</b>	4
<b>Requisitos funcionales .....</b>	4
<b>Beneficios del proyecto .....</b>	4
<b>Metodología .....</b>	4
<b>Resultados Esperados .....</b>	5
<b>Conclusiones .....</b>	5
<b>Evidencias del Proyecto .....</b>	5
Script del Login.....	9
Script de la Base de Datos.....	19
<b>Estructura de la Base de Datos .....</b>	20
Codigo Base de Datos .....	20
<b>Diagrama del Proyecto .....</b>	22
 <b>Estructura del Proyecto iLib (Diagrama Textual).....</b>	22
<b>Dependencias (Librerías externas) .....</b>	23

## Introducción

El sistema LibConnect, una aplicación informática creada con el propósito de optimizar la gestión de acceso y administración de usuarios en una biblioteca pública. Este sistema permite que tanto lectores como administradores puedan ingresar de manera segura ofreciendo a su vez herramientas para el registro, control de préstamos, devoluciones, administración de libros y generación de reportes.

## Objetivos específicos

- Implementar un sistema de inicio de sesión y registro para usuarios de la biblioteca.
- Facilitar el control de préstamos, devoluciones, usuarios y libros desde una misma plataforma.
- Brindar una interfaz intuitiva para el personal encargado de la administración.
- Permitir la creación, edición y eliminación de registros dentro del sistema.
- Generar reportes de las actividades realizadas en la biblioteca.
- Mantener la seguridad de los datos con contraseñas

## Justificación

Las bibliotecas públicas y privadas manejan diariamente gran cantidad de información sobre usuarios y libros. Muchos de estos procesos aún se realizan de manera manual, lo que genera errores, pérdida de tiempo y falta de control sobre los préstamos y devoluciones.

El sistema LibConnect surge como una alternativa tecnológica para modernizar la gestión bibliotecaria, ofreciendo una herramienta segura, práctica y adaptable. A través de su sistema de login, se garantiza que solo personal autorizado tenga acceso a las funciones internas, reduciendo riesgos de manipulación indebida de datos. Además, centraliza toda la información en una sola aplicación, mejorando la eficiencia del trabajo administrativo y ofreciendo una experiencia más fluida a los usuarios.

## Descripción del proyecto

LibConnect es un sistema de gestión y autenticación diseñado especialmente para bibliotecas. Su propósito principal es administrar el flujo de préstamos y devoluciones de libros, el registro y control de usuarios, la edición de información y la generación de reportes de acciones realizadas dentro del sistema.

La interfaz del sistema cuenta con un panel lateral con los siguientes módulos:

- **Principal:** Muestra el mensaje de bienvenida, la fecha actual y una descripción general del sistema.
- **Préstamos:** Permite registrar nuevos préstamos de libros a los usuarios.
- **Devoluciones:** Gestiona la devolución de libros y actualiza su disponibilidad.
- **Usuarios:** Permite registrarse y iniciar sesión.
- **Libros:** Administra el catálogo de libros (registro, edición y eliminación).

- **Reportes:** Genera reportes de acciones, préstamos y devoluciones para control interno.

El sistema fue diseñado para que sea fácil de usar, adaptable a diferentes tipos de bibliotecas y que permita mantener un control detallado de todos los movimientos relacionados con los libros.

## **Funcionalidades principales**

- Registro de nuevos usuarios mediante un formulario conectado a la base de datos.
- Inicio de sesión con verificación de credenciales seguras.
- Panel administrativo con módulos funcionales para la gestión bibliotecaria.
- Control de préstamos y devoluciones de libros en tiempo real.
- Edición y eliminación de registros existentes.
- Generación de reportes y consultas rápidas sobre la información almacenada.
- Visualización del día y fecha actual en la pantalla principal.
- Seguridad en la conexión con la base de datos y manejo cifrado de contraseñas.

## **Requisitos funcionales**

1. Permite iniciar sesión solo a usuarios registrados.
2. Valida que los datos de acceso sean correctos antes de ingresar.
3. Permitir registrar, editar y eliminar usuarios y libros.
4. Registra los préstamos y devoluciones de manera precisa.
5. Generar reportes según los movimientos realizados.
6. Muestra mensajes informativos ante errores o acciones completadas.

## **Beneficios del proyecto**

- Optimiza la administración de usuarios y libros dentro de la biblioteca.
- Facilita el control de préstamos y devoluciones mediante una interfaz clara.
- Reduce errores manuales y mejora la eficiencia del personal.
- Incrementa la seguridad y confidencialidad de la información.
- Permite llevar un registro histórico y automatizado de todas las acciones realizadas.
- Brinda una base sólida para futuras ampliaciones, como reservas de libros o consultas en línea.

## **Metodología**

El proyecto se desarrolló aplicando la metodología incremental, la cual permite ir construyendo el sistema por módulos e ir probando cada parte antes de integrar nuevas funciones.

Las etapas seguidas fueron:

1. **Análisis de requerimientos:** Se definieron las necesidades del sistema y las funciones que debía cubrir.
2. **Diseño:** Se planificó la interfaz visual, la estructura de la base de datos y los diagramas de funcionamiento.
3. **Implementación:** Se desarrolló el código fuente en Java (NetBeans), conectando la aplicación con la base de datos MySQL.
4. **Pruebas:** Se verificó el correcto funcionamiento de los módulos (login, registro, préstamos, etc.).
5. **Documentación:** Se elaboraron los informes técnicos y manuales de uso del sistema.

## **Resultados Esperados**

- Funcionamiento correcto del login y del registro de usuarios.
- Conexión estable con la base de datos MySQL.
- Control eficiente de préstamos y devoluciones.
- Interfaz moderna y fácil de usar.
- Cumplimiento de los objetivos específicos establecidos.

## **Conclusiones**

El sistema LibConnect logró cumplir con el propósito de ofrecer una herramienta práctica para la gestión de bibliotecas, facilitando el control de préstamos, devoluciones, usuarios y libros.

Su diseño centrado en la simplicidad y seguridad permite que el personal de la biblioteca pueda realizar sus tareas con mayor rapidez y precisión.

Además, la implementación de un sistema de login contribuye a la protección de los datos y al acceso restringido solo a personal autorizado.

En conjunto, LibConnect representa un avance hacia la digitalización de procesos bibliotecarios, abriendo la posibilidad de futuras mejoras, como la incorporación de reservas en línea, notificaciones por correo o integración con catálogos virtuales.

## **Evidencias del Proyecto**

Donde cada libro encuentra a su lector ❤

## LibConnect

Principal Préstamos Devoluciones Usuarios Libros Reportes

### Administración/Control/Biblioteca

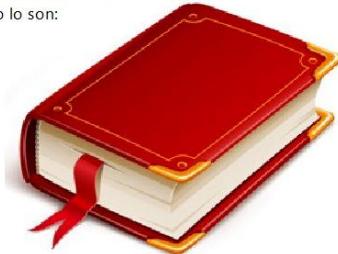
Hoy es viernes 17 de octubre de 2025

## Bienvenido

Sistema de Gestión para Biblioteca Pública. Controle y administre de forma óptima y fácil el flujo de préstamos y devoluciones de Libros.

Esta herramienta le permitirá llevar un control completo y detallado de su Biblioteca, tendrá acceso a herramientas especiales para tareas específicas, como lo son:

- Préstamos
- Devoluciones
- Registro de Usuarios y Libros Nuevos
- Edición de Usuarios y Libros existentes
- Eliminar todo tipo de Registros
- Sección de Reportes de acciones en el sistema



Donde cada libro encuentra a su lector ❤

## LibConnect

Principal Préstamos Devoluciones Usuarios Libros Reportes

### Administración/Control/Biblioteca

Hoy es viernes 17 de octubre de 2025



Folio Usuario  
Ingrese el folio del usuario.

Libro ID  
Ingrese el ID del Libro a prestar.

Prestar

Donde cada libro encuentra a su lector ❤

## LibConnect

Principal Préstamos Devoluciones Usuarios Libros Reportes

### Administración/Control/Biblioteca

Hoy es viernes 17 de octubre de 2025

#### Devolución de Libro

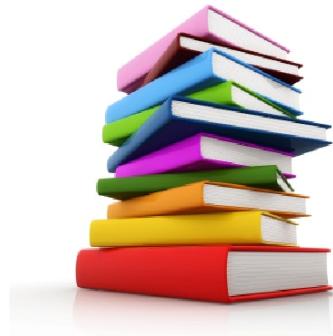
Folio Usuario

Ingrese el folio del usuario.

Libro ID

Ingrese el ID del Libro a devolver.

**Devolver**



Donde cada libro encuentra a su lector ❤

## LibConnect

Principal Préstamos Devoluciones Usuarios Libros Reportes

### Administración/Control/Biblioteca

Hoy es viernes 17 de octubre de 2025

#### Usuarios

Ingrese el nombre de usuario a buscar.

**Buscar**

ID	Nombre	Apellido P.	Apellido M.	Domicilio	Teléfono	Correo
1	diego	untiveros	moreno	aguirre	963852741	diego@gmail.com
2	juan	rodriguez	quispe	tupac	963258147	djuab@gmail.com
3	carlos	yupanqui	taipe	tambo	933567845	carlitos@gmail.com
4	eduardo	ramon	puente	tambo	968745602	ramoncin@gmail.c...

**Nuevo** **Editar** **Borrar**

Donde cada libro encuentra a su lector ❤

## LibConnect

**Administración/Control/Biblioteca**

Hoy es viernes 17 de octubre de 2025

### Registrar nuevo Usuario

Nombre <input type="text" value="Ingrese el nombre del usuario."/>	Domicilio <input type="text" value="Ingrese el domicilio del usuario."/>
Apellido Paterno <input type="text" value="Ingrese el apellido paterno del usuario."/>	Teléfono <input type="text" value="Ingrese el teléfono del usuario."/>
Apellido Materno <input type="text" value="Ingrese el apellido materno del usuario."/>	correo <input type="text" value="Ingrese el correo del usuario."/>

**Registrar**

Donde cada libro encuentra a su lector ❤

## LibConnect

**Administración/Control/Biblioteca**

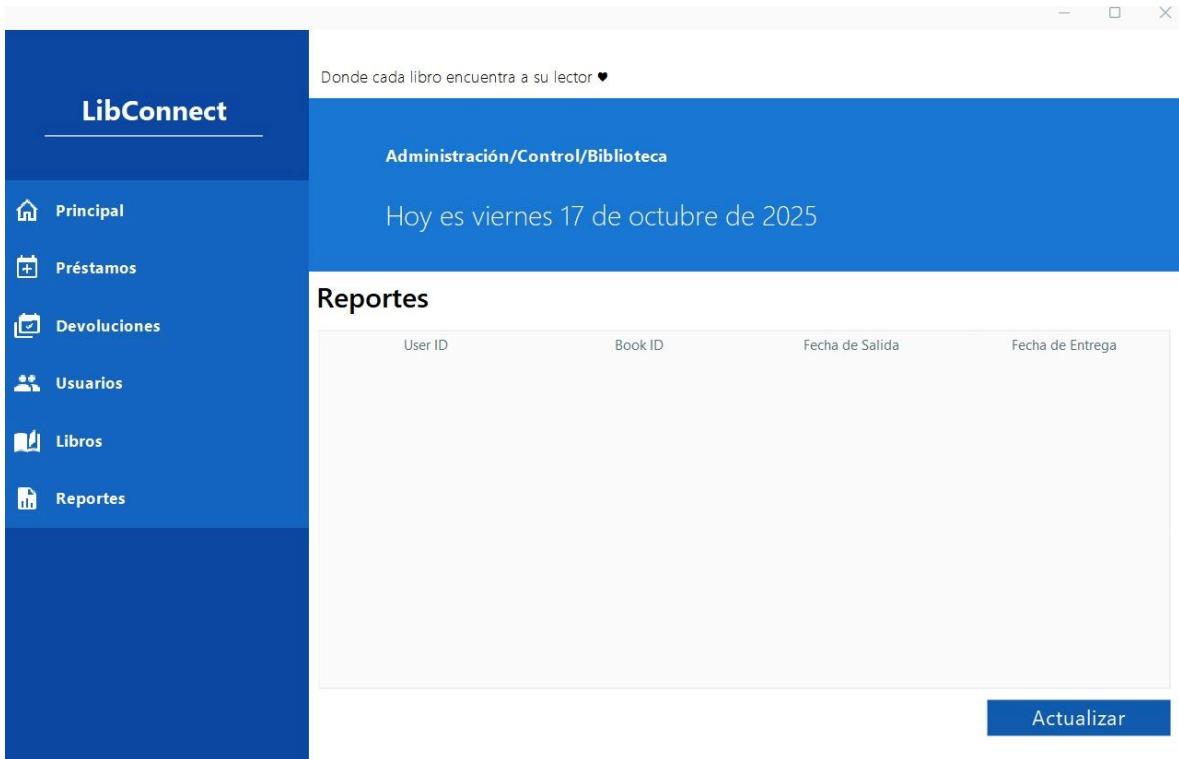
Hoy es viernes 17 de octubre de 2025

### Libros

<input style="width: 80%;" type="text" value="Ingrese el título del libro a buscar."/>	<b>Buscar</b>
--	---------------

ID	Título	Fecha de ...	Autor	Categoría	Edición	Idioma	Páginas	Descripción	Ejemplares...	Stock	Disponibl...
1	teoría g...	1976	Bertalan...	Ciencias ...	Fondo d...	Español	311 pg	La obra ...	10000	100	50
2	Cien año...	1967	Gabriel ...	Literatur...	Editorial ...	Español	471	Novela ...	40	40	33
3	El origen...	1859	Charles ...	Ciencias ...	John Mu...	Inglés	502	Obra fu...	15	15	12
4	Don Quij...	1605	Miguel d...	Literatur...	Real Aca...	Español	863	La gran ...	30	30	27
5	Harry P...	1997	J.K. Rowli...	Fantasi...	Bloomsb...	Inglés	223	Primer li...	60	60	52

**Nuevo** **Editar** **Borrar**



## Script del Login

```
package com.mycompany.views;

import com.mycompany.ilib.DAOUsersImpl; import
com.mycompany.interfaces.DAOUsers; import java.awt.Color;

public class UpUsers extends javax.swing.JPanel {

    boolean isEdition = false;
    com.mycompany.models.Users userEdition;
    public UpUsers() {
        initComponents();
        InitStyles();
    }

    public UpUsers(com.mycompany.models.Users user) {
        initComponents();
        isEdition = true;
        userEdition = user;
        InitStyles();
    }

    private void InitStyles() {
        title.putClientProperty("FlatLaf.styleClass", "h1");
    }
}
```

```

        title.setForeground(Color.black);
        nameTxt.putClientProperty("JTextField.placeholderText", "Ingrese
el nombre del usuario.");
        apPTxt.putClientProperty("JTextField.placeholderText", "Ingrese
el apellido paterno del usuario.");
        apMTxt.putClientProperty("JTextField.placeholderText", "Ingrese
el apellido materno del usuario.");
        domTxt.putClientProperty("JTextField.placeholderText", "Ingrese
el domicilio del usuario.");
        phoneTxt.putClientProperty("JTextField.placeholderText",
"Ingrese el teléfono del usuario.");
        emailTxt.putClientProperty("JTextField.placeholderText",
"Ingrese el correo del usuario.");

        if (isEdition) {
            title.setText("Editar Usuario");
            button.setText("Guardar");

            if (userEdition != null) {
                nameTxt.setText(userEdition.getName());
                apPTxt.setText(userEdition.getLast_name_p());
                apMTxt.setText(userEdition.getLast_name_m());
                domTxt.setText(userEdition.getDomicilio());
                phoneTxt.setText(userEdition.getTel());
                emailTxt.setText(userEdition.getEmail());
            }
        }
    }

    /**
     * This method is called from within the constructor to initialize
     * the form.
     *
     * WARNING: Do NOT modify this code. The content of this method is
     * always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    bg = new javax.swing.JPanel();

```

```
title = new javax.swing.JLabel();
nameLbl = new javax.swing.JLabel();
nameTxt = new javax.swing.JTextField();
apPLbl = new javax.swing.JLabel();
apPTxt = new javax.swing.JTextField();
apMLbl = new javax.swing.JLabel();
apMTxt = new javax.swing.JTextField();
jSeparator1 = new javax.swing.JSeparator();
domLbl = new javax.swing.JLabel();
domTxt = new javax.swing.JTextField();
button = new javax.swing.JButton();
phoneLbl = new javax.swing.JLabel();
phoneTxt = new javax.swing.JTextField();
emailTxt = new javax.swing.JTextField();
emailLbl = new javax.swing.JLabel();

setBackground(new java.awt.Color(255, 255, 255));

bg.setBackground(new java.awt.Color(255, 255, 255));

title.setText("Registrar nuevo Usuario");

nameLbl.setText("Nombre");

apPLbl.setText("Apellido Paterno");

apMLbl.setText("Apellido Materno");

jSeparator1.setForeground(new java.awt.Color(204, 204, 204));
jSeparator1.setOrientation(javax.swing.SwingConstants.VERTICAL);
jSeparator1.setPreferredSize(new java.awt.Dimension(200, 10));

domLbl.setText("Domicilio");

domTxt.setToolTipText("");
domTxt.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
{
    domTxtActionPerformed(evt);
}
});

});
```

```
button.setBackground(new java.awt.Color(18, 90, 173));
button.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
button.setForeground(new java.awt.Color(255, 255, 255));
button.setText("Registrar");
button.setBorderPainted(false);
button.setCursor(new
java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
button.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
{
    buttonActionPerformed(evt);
}
});
};

phoneLbl.setText("Teléfono");

phoneTxt.setToolTipText("");
phoneTxt.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
{
    phoneTxtActionPerformed(evt);
}
});
};

emailTxt.setToolTipText("");
emailTxt.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
{
    emailTxtActionPerformed(evt);
}
});
};

emailLbl.setText("correo");

javax.swing.GroupLayout bgLayout = new
javax.swing.GroupLayout(bg);
bg.setLayout(bgLayout);
bgLayout.setHorizontalGroup(
bgLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(bgLayout.createSequentialGroup()

```

```
        .addContainerGap()

    .addGroup(bgLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(bgLayout.createSequentialGroup()

    .addGroup(bgLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(bgLayout.createSequentialGroup()
            .addComponent(nameLbl,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGap(223, 223, 223))
            .addComponent(nameTxt)
        .addGroup(bgLayout.createSequentialGroup()
            .addComponent(apPLbl,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGap(182, 182, 182))
            .addComponent(apPTxt)
        .addGroup(bgLayout.createSequentialGroup()
            .addComponent(apMLbl,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGap(180, 180, 180))
            .addComponent(apMTxt))
            .addGap(68, 68, 68)
            .addComponent(jSeparator1,
javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(70, 70, 70)

    .addGroup(bgLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(domTxt)
        .addGroup(bgLayout.createSequentialGroup()
            .addComponent(domLbl,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGap(220, 220, 220))
            .addComponent(phoneTxt)
        .addGroup(bgLayout.createSequentialGroup()
```

```
        .addComponent(phoneLbl,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGap(218, 218, 218))
        .addComponent(button,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(emailTxt)
        .addComponent(emailLbl,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addGap(72, 72, 72))
.addGroup(bgLayout.createSequentialGroup()
        .addComponent(title,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGap(553, 553)))))

);
bgLayout.setVerticalGroup(
bgLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(bgLayout.createSequentialGroup()
            .addContainerGap()
            .addComponent(title,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(bgLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(bgLayout.createSequentialGroup()
                .addComponent(nameLbl,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(nameTxt,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addGap(18, 18, 18)
        .addComponent(apPLbl,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(apPTxt,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(apMLbl,
javax.swing.GroupLayout.DEFAULT_SIZE, 17, Short.MAX_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(apMTxt,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(170, 170, 170))
        .addGroup(bgLayout.createSequentialGroup())
        .addComponent(domLbl,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(domTxt,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(phoneLbl,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(phoneTxt,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(emailLbl,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGap(2, 2, 2)
        .addComponent(emailTxt,
```

```

javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(50, 50, 50)
        .addComponent(button,
javax.swing.GroupLayout.PREFERRED_SIZE, 50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(70, 70, 70))
        .addGroup(bgLayout.createSequentialGroup()
        .addComponent(jSeparator1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGap(26, 26, 26)))
);

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(this);
this.setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(bg, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);
layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(bg, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);
}// </editor-fold>

private void buttonActionPerformed(java.awt.event.ActionEvent evt) {
    String nombre = nameTxt.getText();
    String apP = apPTxt.getText();
    String apM = apMTxt.getText();
    String dom = domTxt.getText();
    String email = emailTxt.getText();
    String tel = phoneTxt.getText();

    // Validaciones para los campos

```

```

        if (nombre.isEmpty() || apP.isEmpty() || apM.isEmpty() || 
dom.isEmpty() || email.isEmpty() || tel.isEmpty()) {
javax.swing.JOptionPane.showMessageDialog(this, "Debe llenar todos
los campos. \n", "AVISO", javax.swing.JOptionPane.ERROR_MESSAGE);
nameTxt.requestFocus();
return;
}

String emailRegex = "^[A-Za-z0-9+_.-]+@(.+)$"; if (!email.matches(emailRegex)) {
javax.swing.JOptionPane.showMessageDialog(this, "Ingrese un correo válido (ejemplo:
usuario@gmail.com). \n", "AVISO", javax.swing.JOptionPane.ERROR_MESSAGE);
emailTxt.requestFocus(); return; }

com.mycompany.models.Users user = isEdition ? userEdition : new
com.mycompany.models.Users();
user.setName(nombre);
user.setLast_name_p(apP);
user.setLast_name_m(apM);
user.setDomicilio(dom);
user.setEmail(email);
user.setTel(tel);

try {
    DAOUsers dao = new DAOUsersImpl();

    if (!isEdition) {
        dao.registrar(user);
    } else {
        dao.modificar(user);
    }
}

String successMsg = isEdition ? "modificado" : "registrado";

javax.swing.JOptionPane.showMessageDialog(this, "Usuario " +
successMsg + " exitosamente.\n", "AVISO",
javax.swing.JOptionPane.INFORMATION_MESSAGE);

if (!isEdition) {
    nameTxt.setText("");
    apPTxt.setText("");
    apMTxt.setText("");
}

```

```

        domTxt.setText("");
        emailTxt.setText("");
        phoneTxt.setText("");
    }
} catch (Exception e) {
    String errorMsg = isEdition ? "modificar" : "registrar";
    javax.swing.JOptionPane.showMessageDialog(this, "Ocurrió un
error al " + errorMsg + " el usuario. \n", "AVISO",
javax.swing.JOptionPane.ERROR_MESSAGE);
    System.out.println(e.getMessage());
}
}

private void emailTxtActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
}

private void phoneTxtActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
}

private void domTxtActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

// Variables declaration - do not modify
private javax.swing.JLabel apMLbl;
private javax.swing.JTextField apMTxt;
private javax.swing.JLabel apPLbl;
private javax.swing.JTextField apPTxt;
private javax.swing.JPanel bg;
private javax.swing.JButton button;
private javax.swing.JLabel domLbl;
private javax.swing.JTextField domTxt;
private javax.swing.JLabel emailLbl;
private javax.swing.JTextField emailTxt;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JLabel nameLbl;
private javax.swing.JTextField nameTxt;

```

```
private javax.swing.JLabel phoneLbl;
private javax.swing.JTextField phoneTxt;
private javax.swing.JLabel title;
// End of variables declaration

}
```

## Script de la Base de Datos

-- Crear base de datos

```
CREATE DATABASE iLib;
```

```
USE iLib;
```

-- Tabla de usuarios

```
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL,
    password VARCHAR(100) NOT NULL,
    fullname VARCHAR(100),
    email VARCHAR(100),
    role VARCHAR(20) DEFAULT 'user'
);
```

-- Tabla de libros

```
CREATE TABLE books (
    id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(150) NOT NULL,
    author VARCHAR(100),
    category VARCHAR(50),
    year INT,
    available BOOLEAN DEFAULT TRUE
);
```

-- Tabla de préstamos

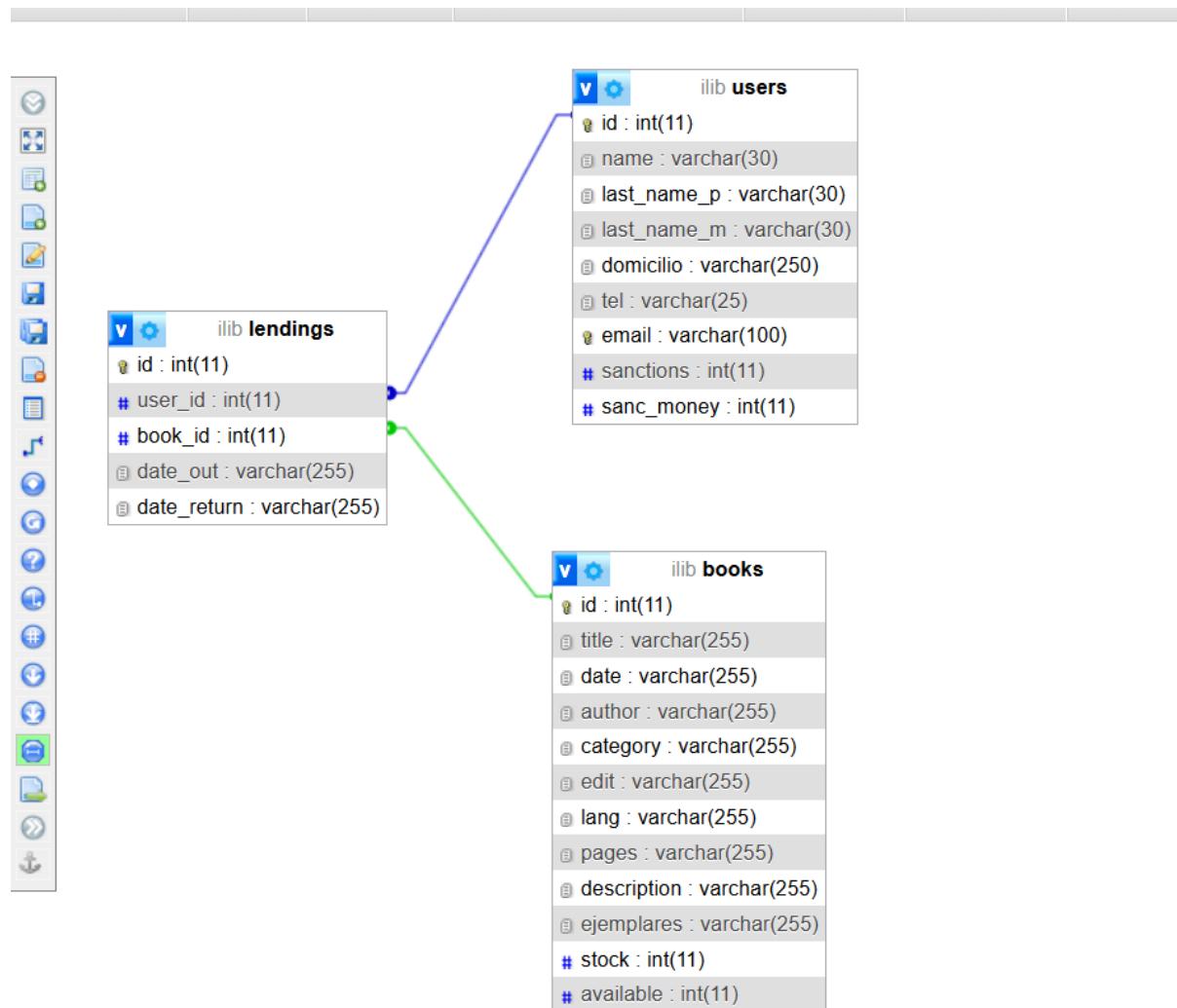
```
CREATE TABLE lendings ( id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    book_id INT NOT NULL,
    lending_date DATE NOT NULL,
    return_date DATE,
    status VARCHAR(20) DEFAULT 'Prestado',
```

```

FOREIGN KEY (user_id) REFERENCES users(id),
FOREIGN KEY (book_id) REFERENCES books(id)
);

```

## Estructura de la Base de Datos



## Código Base de Datos

```

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO"; SET AUTOCOMMIT = 0; START
TRANSACTION; SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
```

```
CREATE TABLE books ( id int(11) NOT NULL, title varchar(255) NOT NULL, date  
varchar(255) NOT NULL, author varchar(255) NOT NULL, category varchar(255) NOT  
NULL, edit varchar(255) NOT NULL, lang varchar(255) NOT NULL, pages varchar(255)  
NOT NULL, description varchar(255) NOT NULL, ejemplares varchar(255) NOT  
NULL, stock int(11) NOT NULL, available int(11) NOT NULL ) ENGINE=InnoDB  
DEFAULT CHARSET=utf8;
```

```
CREATE TABLE lendings ( id int(11) NOT NULL, user_id int(11) NOT NULL, book_id  
int(11) NOT NULL, date_out varchar(255) NOT NULL, date_return varchar(255)  
DEFAULT NULL ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE users ( id int(11) NOT NULL, name varchar(30) NOT NULL,  
last_name_p varchar(30) NOT NULL, last_name_m varchar(30) NOT NULL,  
domicilio varchar(250) DEFAULT NULL, tel varchar(25) DEFAULT NULL, email  
varchar(100) NOT NULL UNIQUE, sanctions int(11) DEFAULT '0', sanc_money int(11)  
NOT NULL DEFAULT '0' ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
ALTER TABLE books ADD PRIMARY KEY (id); ALTER TABLE lendings ADD PRIMARY  
KEY (id); ALTER TABLE users ADD PRIMARY KEY (id);
```

```
ALTER TABLE books MODIFY id int(11) NOT NULL AUTO_INCREMENT; ALTER TABLE  
lendings MODIFY id int(11) NOT NULL AUTO_INCREMENT; ALTER TABLE users  
MODIFY id int(11) NOT NULL AUTO_INCREMENT;
```

```
COMMIT;
```

```
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT ; !40101  
SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS ; !40101 SET  
@OLD_COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

```
ALTER TABLE lendings ADD CONSTRAINT fk_lendings_user FOREIGN KEY  
(user_id) REFERENCES users (id) ON DELETE CASCADE ON UPDATE CASCADE, ADD  
CONSTRAINT fk_lendings_book FOREIGN KEY (book_id) REFERENCES books (id)  
ON DELETE CASCADE ON UPDATE CASCADE;
```

## Diagrama del Proyecto

### Estructura del Proyecto iLib (Diagrama Textual)

```
iLib/
|
|   └── com.mycompany.db
|       └── Database.java
|
|   └── com.mycompany.ilib
|       ├── DAOBooksImpl.java
|       ├── DAOLendingsImpl.java
|       ├── DAOUsersImpl.java
|       └── Dashboard.java
|
|   └── com.mycompany.interfaces
|       ├── DAOBooks.java
|       ├── DAOLendings.java
|       └── DAOUsers.java
|
|   └── com.mycompany.models
|       ├── Books.java
|       ├── Lendings.java
|       └── Users.java
|
|   └── com.mycompany.utils
|       └── Utils.java
|
└── com.mycompany.views
    └── Books.java
```

```
|── Lendings.java  
|── Principal.java  
|── Reports.java  
|── Returns.java  
|── UpBooks.java  
|── UpUsers.java  
└── Users.java
```

## Dependencias (Librerías externas)

libs/

```
|── AbsoluteLayout-SNAPSHOT.jar  
|── flatlaf-2.4.jar  
|── flatlaf-intellij-themes-2.4.jar  
|── mysql-connector-j-8.0.33.jar  
└── protobuf-java-3.21.9.jar
```

