EXAMEN FINAL  
PROYECTO BIBLIOTECA

Nicolas Torres Jacome 192409  
Josef Geovanny Criado Duran 192417  
Jorge Alberto Alonso Sumalave 192427  
José Julián Pedraza Bustamante 192428

Universidad Francisco De Paula Santander Ocaña.

Facultad de Ingenieria

Programación I

Ing. Jesús Eduardo

06 de Diciembre de 2024

## ****Introducción****

### El proyecto de gestión de biblioteca, desarrollado en Java y MySQL, optimiza los procesos bibliotecarios mediante una interfaz intuitiva. El sistema permite hacer un seguimiento preciso de los préstamos y devoluciones, asegurando la disponibilidad de los libros. La base de datos MySQL almacena de manera segura toda la información sobre usuarios, libros e historial de transacciones, facilitando la gestión del inventario y mejorando la experiencia del usuario.

### ****Objetivos del proyecto****

* Implementar un sistema que automatice la gestión de bibliotecas mediante una interfaz gráfica intuitiva.
* Mejorar la eficiencia administrativa y el control de préstamos.
* Proporcionar a los usuarios una experiencia moderna, con funcionalidades accesibles desde cualquier dispositivo.

### ****Alcance del proyecto****

* Registro y autenticación de usuarios.
* Listado de libros con información detallada, como título, autor, categoría y disponibilidad.
* Gestión de préstamos y devoluciones de libros.
* Visualización del historial de préstamos de cada usuario.
* Interfaz gráfica amigable, desarrollada en Swing.

## ****Documentación Técnica****

### ****Estructura del sistema****

#### ****1. Conexión a la Base de Datos****

El sistema utiliza JDBC para establecer una conexión segura con una base de datos MySQL, donde se almacena toda la información de los usuarios, libros y transacciones.

#### ****2. Gestión de Usuarios****

Este módulo incluye:

* **Registro:** Añade nuevos usuarios a la base de datos.
* **Autenticación:** Valida las credenciales del usuario para iniciar sesión.
* **Visualización de información:** Muestra datos relevantes de cada usuario, como su historial de préstamos.

#### ****3. Gestión de Libros****

Este módulo abarca:

* **Listado:** Visualización del catálogo de libros disponibles.
* **Préstamos:** Permite asignar un libro a un usuario y actualizar su disponibilidad.
* **Devoluciones:** Marca un libro como disponible nuevamente y elimina su asociación con el usuario en el registro de préstamos.

#### ****4. Interfaz Gráfica (GUI)****

Desarrollada con **Swing**, la GUI organiza las funcionalidades en paneles:

* Inicio de sesión y registro de usuarios.
* Listado gráfico de libros, incluyendo imágenes.
* Botones de acción para prestar y devolver libros.
* Visualización del historial de libros prestados.

### *****5.* Tecnologías empleadas****

* **Java:** Lenguaje de programación principal por su versatilidad.
* **Swing:** Librería gráfica de Java.
* **JDBC:** Protocolo para conectar Java con bases de datos relacionales.
* **MySQL:** Almacena información estructurada del sistema.

## ****Detalles de la Interfaz Gráfica****

### ****Inicio de Sesión****

* Permite a los usuarios autenticarse ingresando su documento.
* Incluye la opción de registrar nuevos usuarios, lo que genera automáticamente un ID único en la base de datos.

### ****Pantalla Principal****

* **Listado de libros:**
  + Muestra los libros disponibles con detalles como título, autor, categoría y una imagen miniatura.
  + Cada libro se presenta en un panel individual con un diseño visual atractivo.
* **Préstamos:**
  + El usuario puede seleccionar un libro por su ID para solicitar un préstamo.
  + La base de datos actualiza el estado del libro a "No disponible".
* **Devoluciones:**
  + Permite al usuario devolver libros prestados ingresando su ID.
  + El libro regresa al estado "Disponible".
* **Historial de préstamos:**
  + Muestra una lista de libros prestados por el usuario actual, con detalles como fecha de préstamo y autor.

## ****Fragmentos de Código Destacados****

### ****1. Conexión a la Base de Datos****

public Connection conectar() {

Connection conexion = null;

try {

Class.forName("com.mysql.cj.jdbc.Driver");

conexion = DriverManager.getConnection(

"jdbc:mysql://localhost:3306/bibliotecabase",

"root", "Chimaru2005"

);

} catch (ClassNotFoundException | SQLException e) {

JOptionPane.showMessageDialog(null,

"Error al conectar con la base de datos: " + e.getMessage(),

"Error", JOptionPane.ERROR\_MESSAGE);

}

return conexion;

}

### ****2. Gestión de Usuarios****

private static Usuario buscarUsuarioEnBaseDeDatos(Main app, String documento) {

String sql = "SELECT id, nombre FROM usuarios WHERE documento = ?";

try (Connection conexion = app.conectar();

PreparedStatement pstmt = conexion.prepareStatement(sql)) {

pstmt.setString(1, documento);

ResultSet rs = pstmt.executeQuery();

if (rs.next()) {

return new Usuario(rs.getString("nombre"), documento, 0, rs.getInt("id"));

}

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, "Error al buscar usuario: " + e.getMessage());

}

return null;

}

### ****3. Visualización de Libros****

private static void listarLibros(Main app, JPanel panel) {

String sql = "SELECT id, nombre, autor, categoria, disponible, imagen FROM libros";

try (Connection conexion = app.conectar();

Statement stmt = conexion.createStatement();

ResultSet rs = stmt.executeQuery(sql)) {

panel.removeAll();

while (rs.next()) {

JPanel libroPanel = new JPanel(new BorderLayout());

libroPanel.add(new JLabel(new ImageIcon(rs.getString("imagen"))), BorderLayout.CENTER);

panel.add(libroPanel);

}

panel.revalidate();

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, "Error al listar libros: " + e.getMessage());

}

}