

Informe del Proyecto Java - Biblioteca

Materia: Programación II

Alumnos: Tomas Mastroprieto, Juan Segura, Gian Guzman

Profesor: Cristina Domizio

Año: 2025

Trabajo: Parcial 2

Introducción

Este trabajo consiste en una aplicación de biblioteca desarrollada en Java. Incluye una interfaz gráfica construida con JFrame, permitiendo que cualquier persona pueda usarla fácilmente. La aplicación permite registrar libros y personas, realizar préstamos, extenderlos, devolver libros y consultar información general del sistema. Aplicamos conceptos trabajados en clase como POO, JDBC, Swing y Gradle.

Objetivos

- Aplicar Programación Orientada a Objetos (POO).
- Conectar a una base de datos MySQL mediante JDBC.
- Crear una aplicación funcional con interfaz gráfica.
- Validar entradas de usuario y aplicar seguridad básica.
- Mantener una estructura de código clara y mantenible.

Herramientas utilizadas

- Lenguaje: Java
- Gestor del proyecto: Gradle
- Base de datos: MySQL
- IDE: IntelliJ IDEA / NetBeans
- Interfaz gráfica: Swing (JFrame)

Estructura y clases principales

Libro (Libro.java) Guarda los datos del libro: numero (que funciona como ID), titulo y clasificacion (categoría). Los campos son públicos para acceso directo. No implementa getters, setters explícitos ni

un método toString() personalizado. La disponibilidad de un libro se infiere por si está o no en un préstamo activo, no por un campo booleano en esta clase o en la tabla libros.

Persona (Persona.java) Representa a los usuarios. Tiene dni (que funciona como ID) y nombre. Los campos son públicos. Es una clase de datos (POJO) y no contiene métodos para mostrar sus datos; esa lógica reside en AppBiblioteca. Se relaciona con los préstamos a través de la clase Prestamo y las consultas a la base de datos.

Prestamo (Prestamo.java) Relaciona una Persona con un Libro. Guarda un numero (ID del préstamo), dia_prestamo (fecha de inicio), devolucion (fecha de fin), el Libro prestado y la Persona (socio) que realiza el préstamo. La condición de "devuelto" se maneja por la eliminación del préstamo de la lista activa y de la base de datos. La lógica para controlar si un préstamo puede extenderse (máximo una vez) se maneja en la clase AppBiblioteca mediante un Set llamado prestamosExtendidos, no con un método dentro de la clase Prestamo.

Ventana Principal (AppBiblioteca.java) Esta clase, que hereda de JFrame, gestiona toda la interfaz gráfica y la lógica de la aplicación. Es la ventana principal desde donde se accede a las funcionalidades, comenzando por un formulario de ingreso de usuario. Componentes importantes:

- JFrame (ventanaActual): Ventana principal de las diferentes vistas.
- JPanel: Se utilizan paneles para organizar los componentes en cada ventana.

Funcionalidades y sus métodos

1. Ver libros disponibles

- Método: ObtenerLibrosDisponibles()
- Llama a: LibroDAO.obtenerLibrosDisponibles()

2. Realizar préstamo

- Método: IngresarPrestamo()
- Llama a: PrestamoDAO.IngresarPrestamo(...)

3. Extender préstamo

- Método: extenderPrestamo()
- Llama a: AppBiblioteca.extenderPrestamo()

4. Devolver libro

- Método: devolverLibro()
- Llama a: AppBiblioteca.devolverLibro()

5. Ver préstamos por persona

0 Método: verMisPrestamos()

- Llama a: `AppBiblioteca.verMisPrestamos()`
- 6. Ver usuarios registrados (con contraseña)
 - Método: `mostrarUsuariosRegistrados()`
 - Llama a: `AppBiblioteca.mostrarUsuariosRegistrados()`
- 7. Salir del sistema
 - Cierra la aplicación con `System.exit(0)`

Clases DAO utilizadas

LibroDAO

- `obtenerLibrosDisponibles(Connection conn)`
- `insertarLibros(Connection conn)`
- `obtenerTituloLibro(Connection conn, int idLibro)`

PersonaDAO

- `IngresarUsuario(Connection conn, int dni, String nombre)`

PrestamoDAO

- `IngresarPrestamo(Connection conn, int dni, int idLibro, String tituloLibro, Date fechaSalida, Date fechaDevolucion)`
- `obtenerPrestamosUsuario(Connection conn, int dniUsuario)`
- `eliminarPrestamo(Connection conn, int idPrestamo)`

Estas clases encapsulan el acceso a la base de datos, separando la lógica de negocio del manejo directo de SQL, lo que facilita el mantenimiento y mejora la organización del código.

Flujo general del sistema

1. `Main.java` crea una instancia de `AppBiblioteca`.
2. El constructor de `AppBiblioteca` arma la interfaz gráfica.

3. Cada botón tiene un ActionListener que llama a una funcionalidad.
4. La funcionalidad accede a datos a través de las clases DAO.
5. Los resultados se muestran mediante JOptionPane.

Validaciones y seguridad

- No se pueden prestar libros que ya están prestados.
- Cada préstamo puede extenderse sólo una vez.
- Al devolver, el libro vuelve a estar disponible.
- El acceso a los usuarios está protegido con una contraseña (1234). Si se falla tres veces, la aplicación se cierra.
- Manejo de errores para entradas inválidas o IDs inexistentes.

Lecciones aprendidas

- Separación de responsabilidades entre clases.
- Conexión y consultas a MySQL desde Java usando JDBC.
- Manejo de interfaces gráficas con JFrame y eventos.
- Validación de datos en tiempo real y control de errores.
- Uso del patrón DAO para organización del código.
- Utilización de Gradle como herramienta de construcción.