

Sistema Biblioteca - Prog. Avanzada

10/03/2025

Julián Alfredo Preiti

Escuela Da Vinci

ACN3AV

Gabriel Ernesto Cavasso

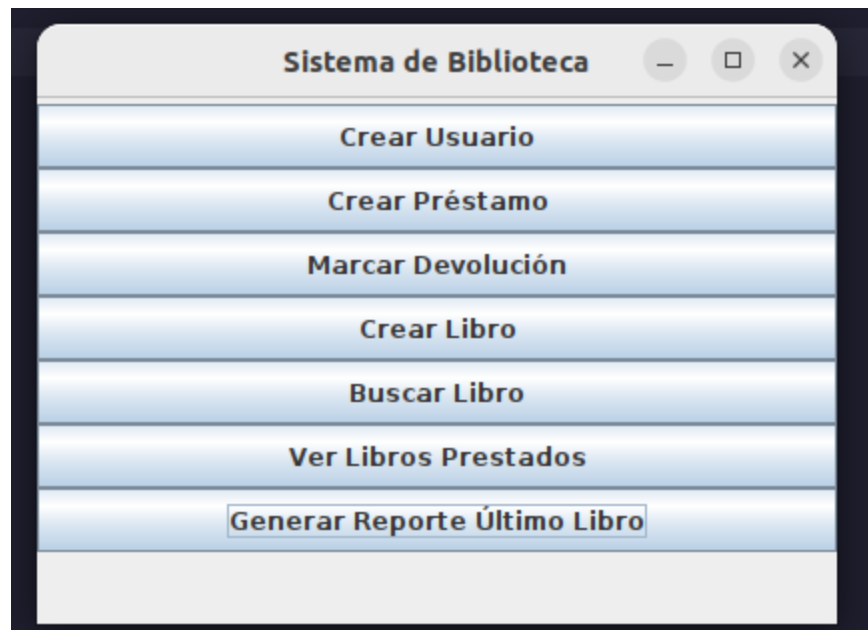
Resumen general

Vamos a trabajar en un sistema de gestión para una biblioteca digital en Java, organizando el código en cuatro paquetes principales:

- **Datos:** Contiene la base de datos junto a un script de creación de prueba y maneja la conexión con el patrón Singleton.
- **Modelo:** Agrupa las clases principales del sistema, como libros y usuarios.
- **Dialog:** Se encarga de las vistas e interacciones con el usuario.
- **Reportes:** Realiza consultas específicas y aplica el patrón Composite para combinar múltiples reportes en uno solo.

El sistema permite gestionar préstamos, devoluciones y disponibilidad de libros, generando reportes personalizados. Se usó JDBC para la base de datos.

¿Cómo se ve el sistema?

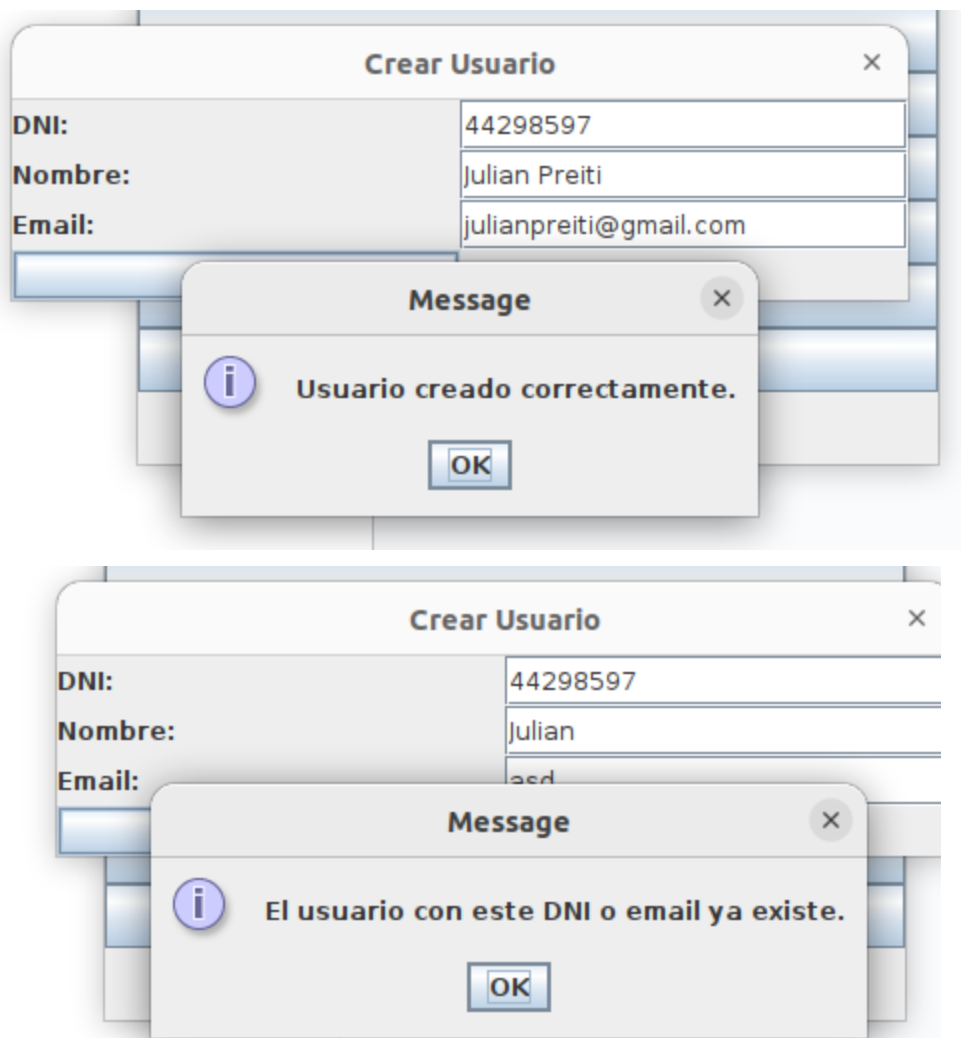


I. Flujo de Creación de Usuario y Libro

El sistema cuenta con un menú principal con botones para gestionar la biblioteca de manera simple. Desde el cual, se pueden crear usuarios y libros, registrar préstamos y devoluciones, buscar libros por nombre, ver el historial de préstamos de un usuario y generar un reporte unificado con información del último libro creado.

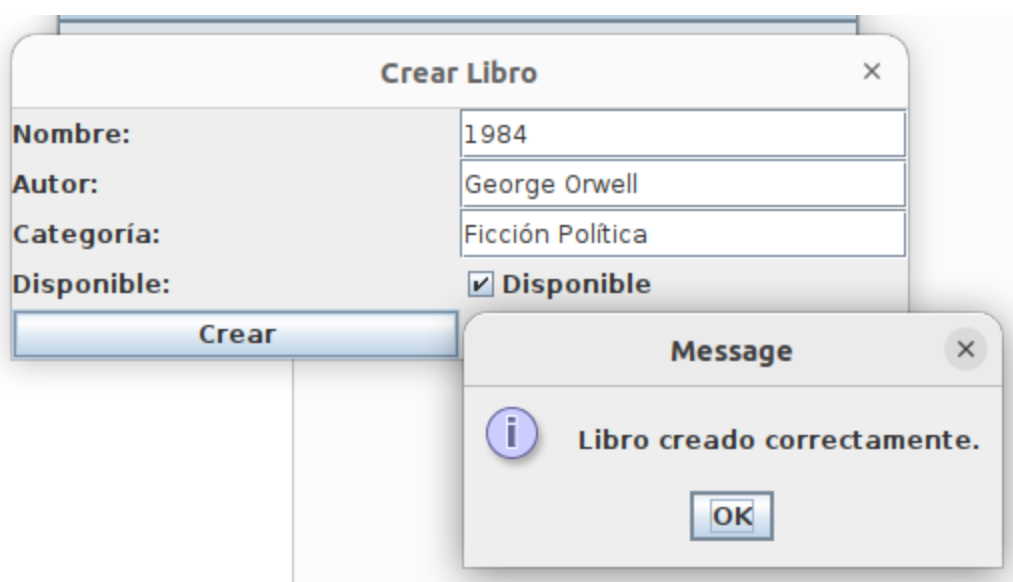
II. Crear Usuario

La ventana de creación de usuario se ve así, ésta valida que no exista un usuario con el mismo DNI o mismo Mail.



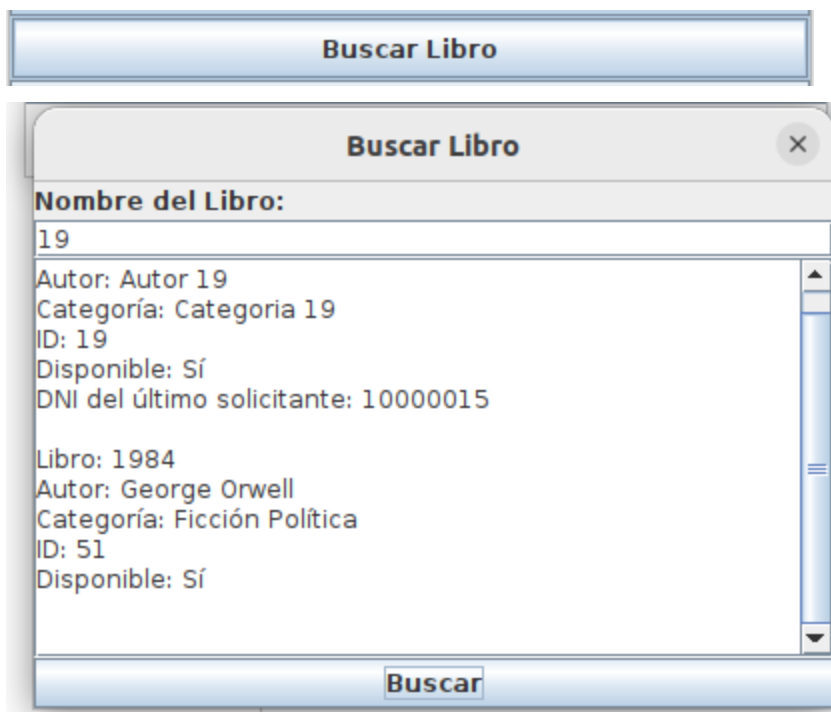
III. Crear Libro

Al crear un libro, la ventana es similar a la de crear un usuario, con la diferencia que podemos tildar un checkbox que determina si el libro está o no disponible.



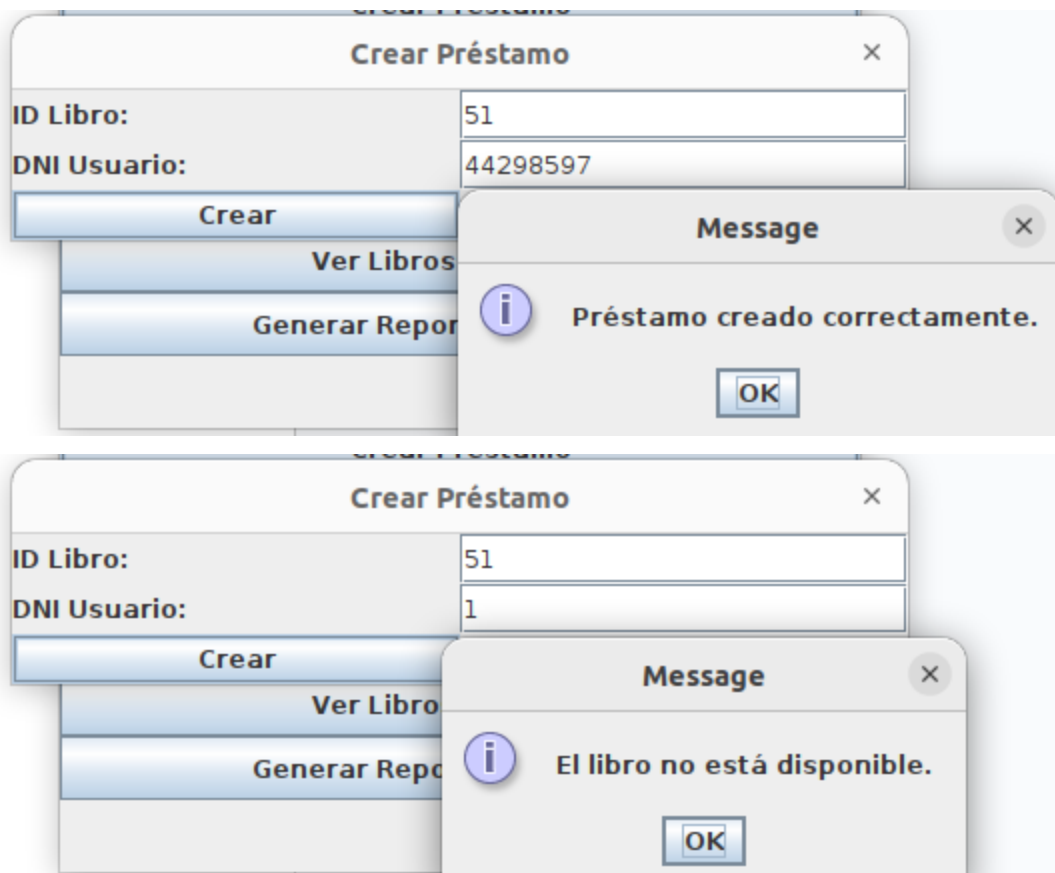
IV. Buscar Libro

Una vez creado el libro, en el buscador podemos indicar una parte del nombre de un libro y podremos acceder a un buscador que coincide con las palabras que contienen el nombre del libro. Indicando su ID, esta será importante ya que cada libro funciona de manera independiente, sabiendo si está disponible y su ID podremos prestarlo.



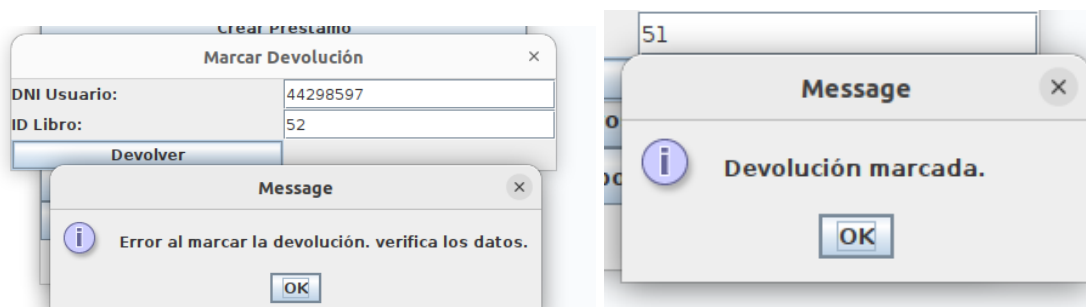
V. Crear Préstamo

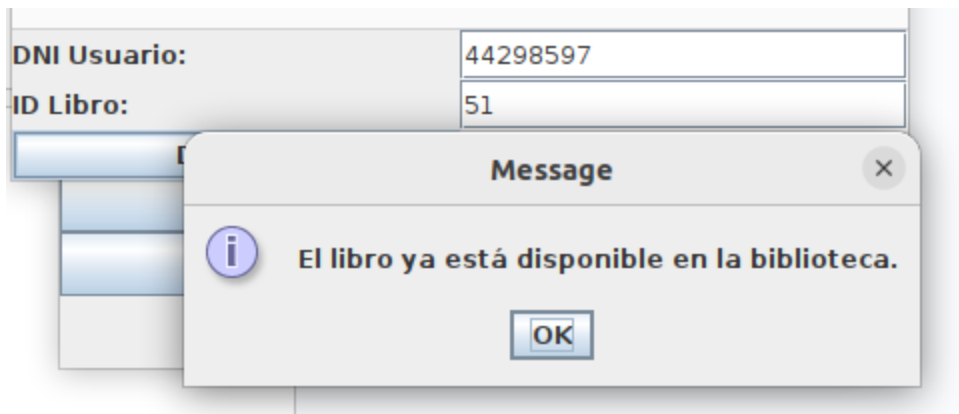
Para crear el préstamo necesitamos indicar el ID de libro, una vez creado el préstamo, nos notifica y si se intenta crear nuevamente, nos informa que ya no está disponible.



VI. Validación de Datos de Devolución

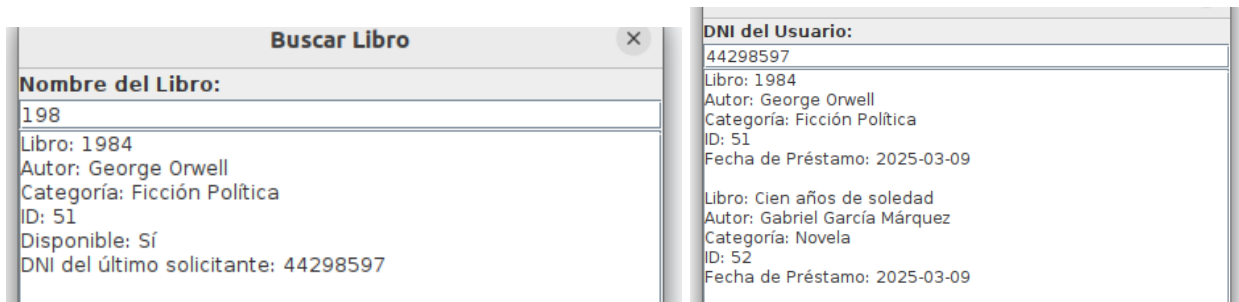
Una vez creado, podemos devolverlo indicando el DNI del usuario y el ID del libro. El cual retoma a su estado de disponible y almacena el último cliente que solicitó éste libro. Una vez devuelto si se intenta devolver nuevamente, informa que está en la biblioteca (En el primer ejemplo se pone 52 en el id para que devuelva error).





VII. ¿Y si lo buscamos de nuevo?

Si buscamos nuevamente el libro podemos ver que se encuentra nuevamente disponible y nos guarda como últimos solicitantes, además en nuestro historial de usuario ya nos figura en sistema junto el resto de libros prestados.



VIII. Patron Composite

Si buscamos nuevamente el libro podemos ver que se encuentra nuevamente disponible y nos guarda como últimos solicitantes, además en nuestro historial de usuario ya nos figura en sistema junto el resto de libros prestados.

Clase ReporteCompuesto

Implementa la interfaz Reporte y permite agregar múltiples reportes para generar un reporte compuesto.

Clase LibrosPrestados

Implementa la interfaz Reporte y genera un reporte de los libros prestados a un usuario específico. (Lo usamos al buscar el historial)

Clase LibroEstado

Implementa la interfaz Reporte y genera un reporte del estado de un libro específico. (Este lo usamos al buscar un libro para ver en qué estado se encuentra)

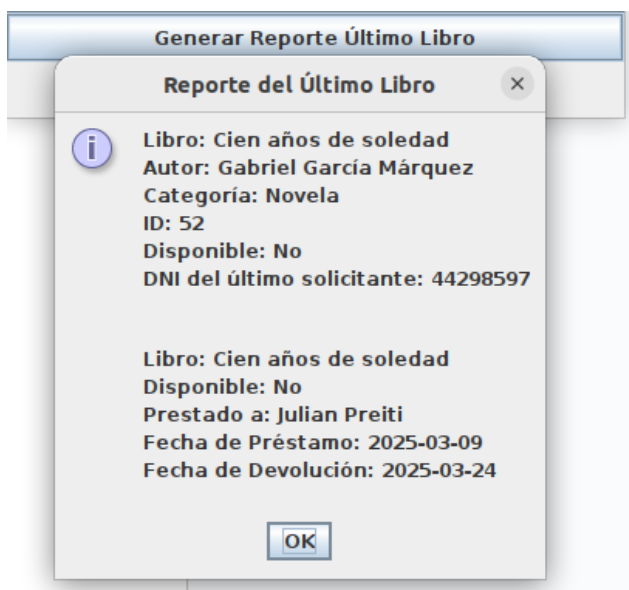
Clase BusquedaLibro

Descripción: Implementa la interfaz Reporte y genera un reporte de los libros que coinciden con un nombre específico.

Interfaz Reporte

Define la interfaz en la cual deben ser implementadas todos los reportes

El resultado de nuestro reporte compuesto entre LibroEstado y BusquedaLibro se aplica al último libro generado en el sistema.



¿Por qué lo armamos así?

El proyecto se estructuró utilizando el patrón Composite para permitir una mayor flexibilidad y escalabilidad en la generación de reportes. Al implementar la interfaz Reporte en las clases y permitir la composición de reportes a través de la clase Reporte Compuesto, se facilita la creación de informes complejos a partir de componentes más simples. Esto no solo mejora la modularidad y reutilización del código, sino que nos permite que con los datos que tenemos, podamos agregar infinitas formas de reportes que sean simples y unificarlos en reportes más complejos. Esto para hacer análisis y mantener el código a futuro es una buena decisión de diseño.

¡Gracias!