

*ESCUELAS SALESIANAS MARÍA
AUXILIADORA*

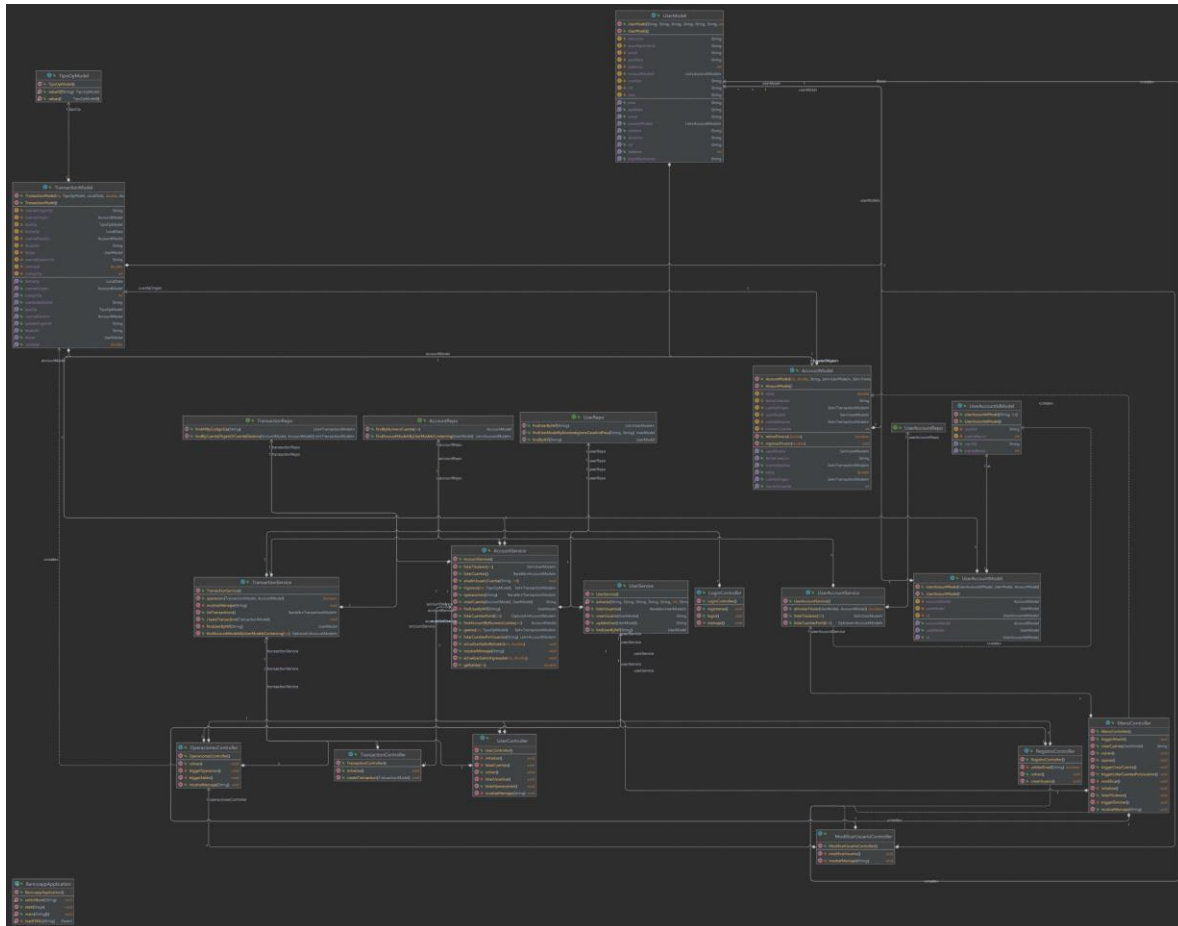
CICLO FORMATIVO DE GRADO
SUPERIOR DESARROLLO DE
APLICACIONES MULTIPLATAFORMA

Banco US-Caja

Liz Laura Campuzano González
21 de mayo de 2023

1. ESTUDIO DEL PROBLEMA Y ANÁLISIS DEL SISTEMA.

a. Se pretendía desarrollar una aplicación para gestionar los usuarios y cuentas de la entidad bancaria asociada a la universidad, donde cada miembro de la universidad es considerado como un usuario potencial.



b. Cuenta con las siguientes funcionalidades:

- Crear usuario
- Modificar los datos del usuario
- Crear cuentas bancarias
- Eliminar cuentas bancarias
- Añadir titulares a las cuentas bancarias
- Eliminar titulares de las cuentas bancarias
- Listar las operaciones realizadas en las cuentas
- Listar los titulares de cada cuenta
- Listar las cuentas de cada usuario

c. El objetivo de la aplicación es tener un control sobre los datos referentes a las transacciones bancarias de los estudiantes universitarios, con ella pueden gestionar sus movimientos y realizar operaciones.

d. Modelado de la solución

i. Recursos humanos (personal/usuarios necesarios)

- Programador java
- Diseñador de JavaFx
- Administrador de Base de Datos (SQL)

ii. Recursos hardware

Se recomienda un ordenador con las siguientes características:

- Los requisitos para esta aplicación son mínimos puesto que no supone un gran consumo de los recursos del ordenador, cualquier ordenador actual es capaz de ejecutarla.

iii. Recursos software

Se recomienda un ordenador con las siguientes características:

- Sistema Operativo Windows
- JDK version 19
- JavaFX version 19
- MySQL

2. EJECUCIÓN DE LA PRÁCTICA

a. Elaboración de la documentación técnica

Lo primero que he hecho es extraer los datos para la creación de un prototipo de clases con sus atributos, así como las relaciones que se establecían entre ellas.

He estructurado el proyecto de tal forma que tengo cuatro paquetes principales:

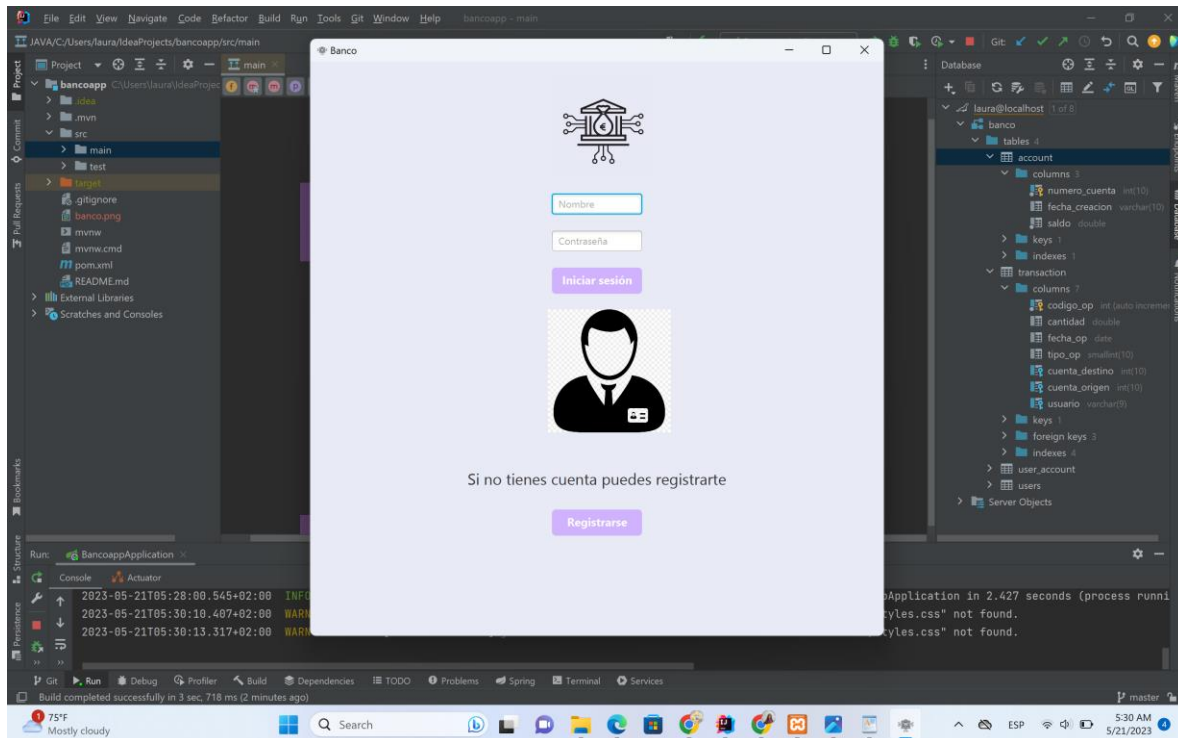
- Models: clases para crear las tablas y sus atributos, así como las relaciones
- Repositories: donde se encuentran los métodos CRUD, y alguno creado por mí para cumplir con las necesidades del proyecto
- Services: donde se encuentra toda la lógica de negocio
- Controllers: donde se enlaza la parte de código con lo que quiero mostrar por las vistas.

He utilizado Spring Boot con Hibernate/JPA para la creación de mi base de datos.

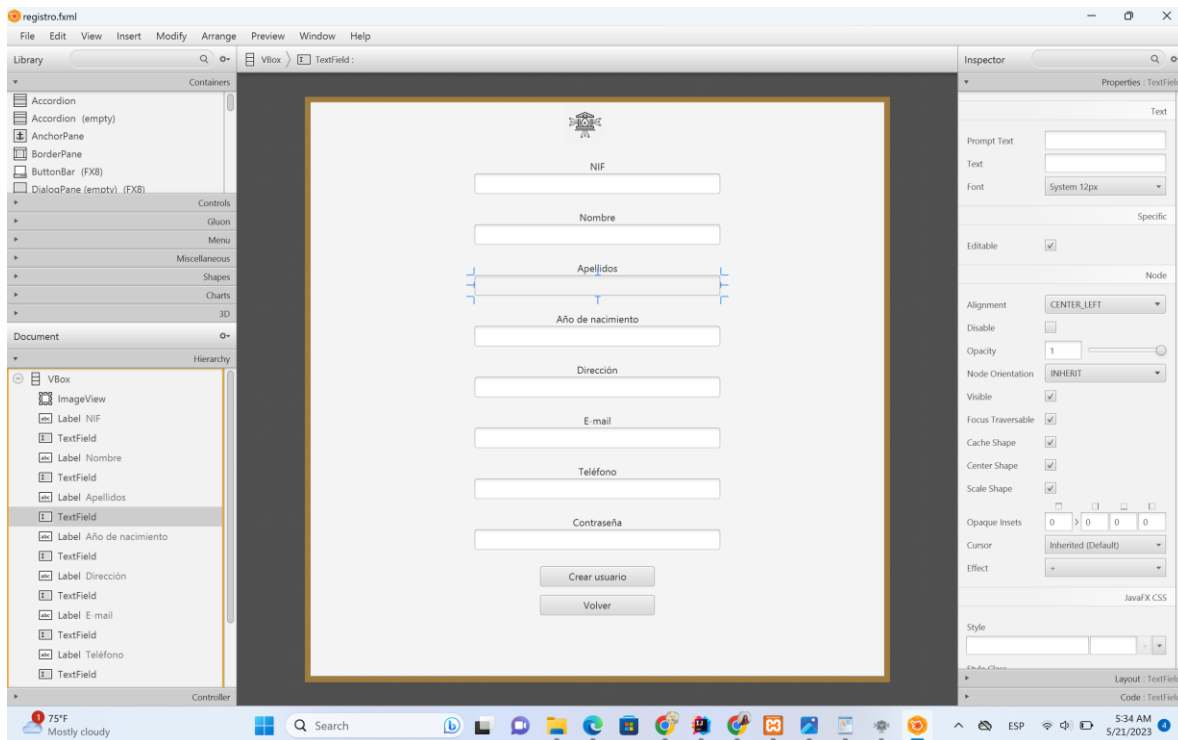
Con Lombok he creado los constructores, métodos Equals, Getter y Setter.

Luego de crear las clases necesarias así como los métodos para satisfacer las necesidades del proyecto, he dedicado gran parte a la visualización, utilizando la herramienta Scene Builder para diseñar mis ventanas de los fxml.

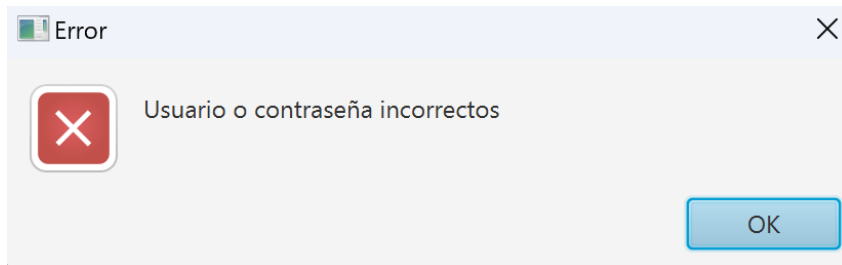
Desarrollo de Aplicaciones Multiplataforma. US-Caja



He puesto detalles de diseño en las vistas auxiliéndome de CSS e imágenes formato png, así como la incorporación de un icono para la aplicación, visible en todas sus escenas.



Cuenta con un sistema de mensajes que aparecen cuando el programa captura alguna excepción, como por ejemplo si el usuario o la contraseña están incorrectos.

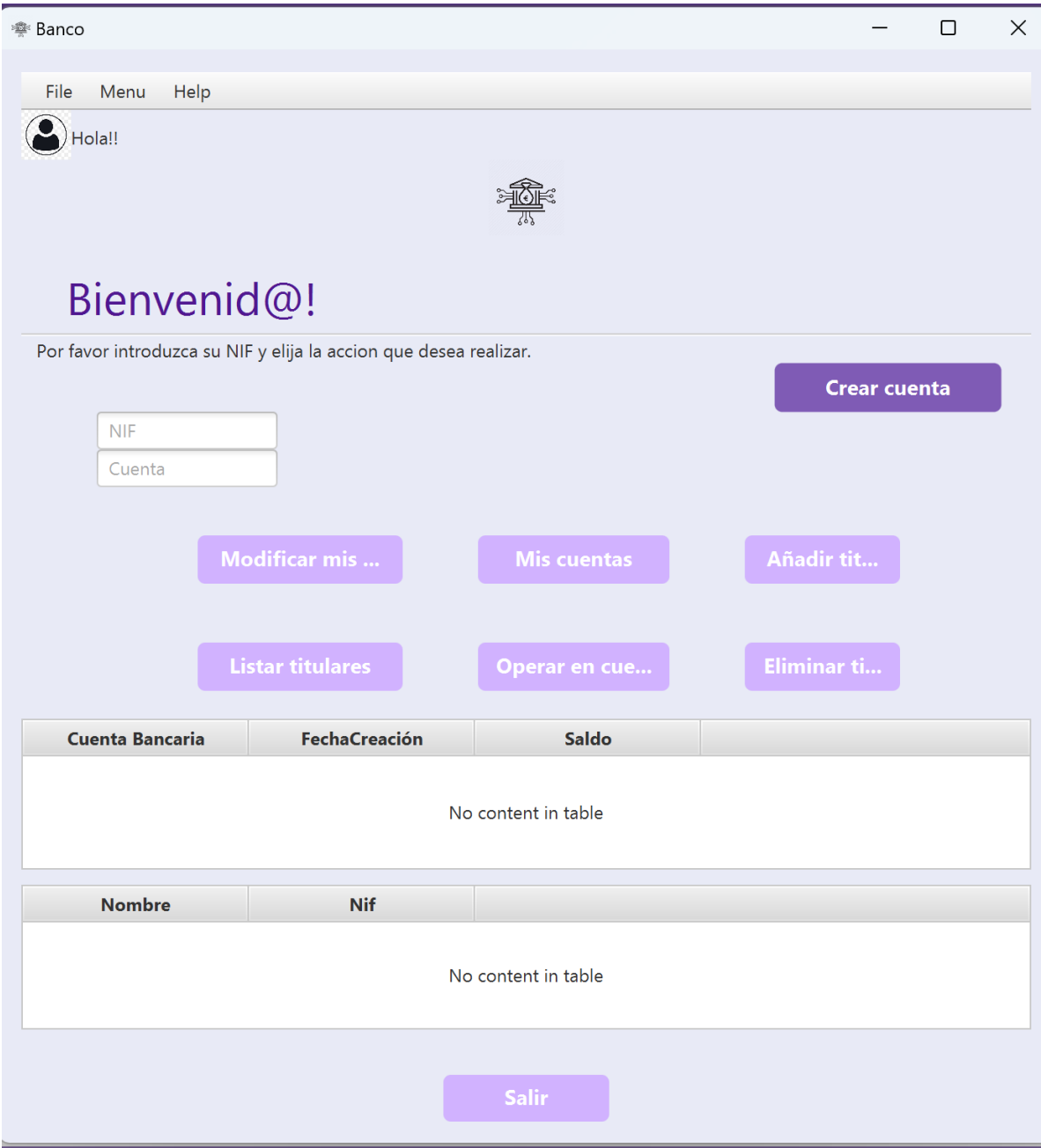


He implementado un diseño que respeta la privacidad del usuario, por ejemplo: la contraseña de cada uno no se encuentra visible en la ventana, pero sí se registra correctamente en la base de datos cuando el usuario se da de alta.

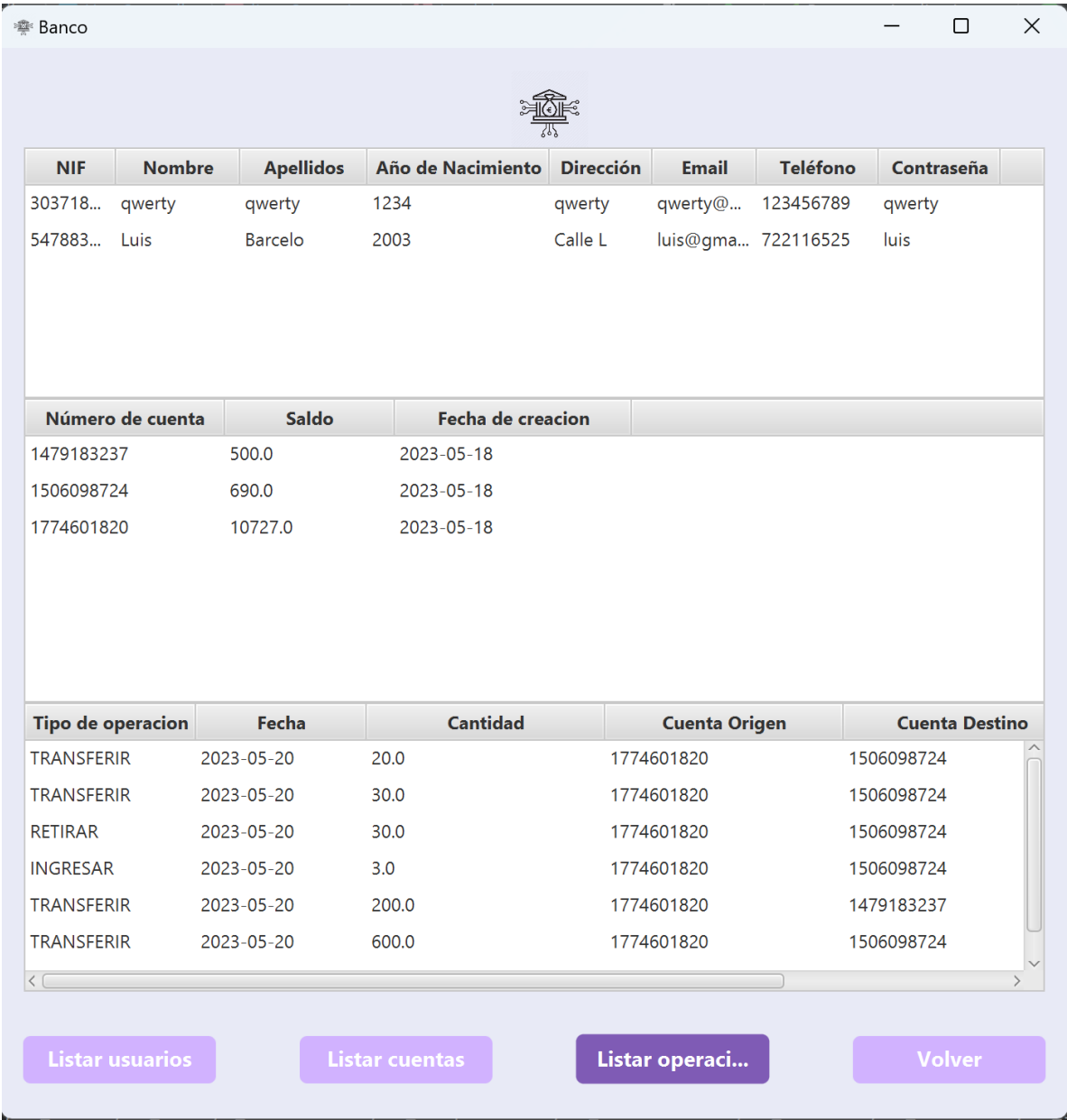
A login form with a light purple background. It features a text input field with the username 'laura'. Below it is a password input field with five black dots for masking. At the bottom is a purple button with the text 'Iniciar sesión' in white.

Tengo roles para la aplicación, está el usuario común y el administrador. Cuando se hace el login, en dependencia de qué tipo de usuario seas te redirige a una ventana o a otra.

Ventana del usuario común



Ventana de administrador



b. Implementación de la aplicación, incluida la base de datos usando el sistema gestor de base de datos elegido.

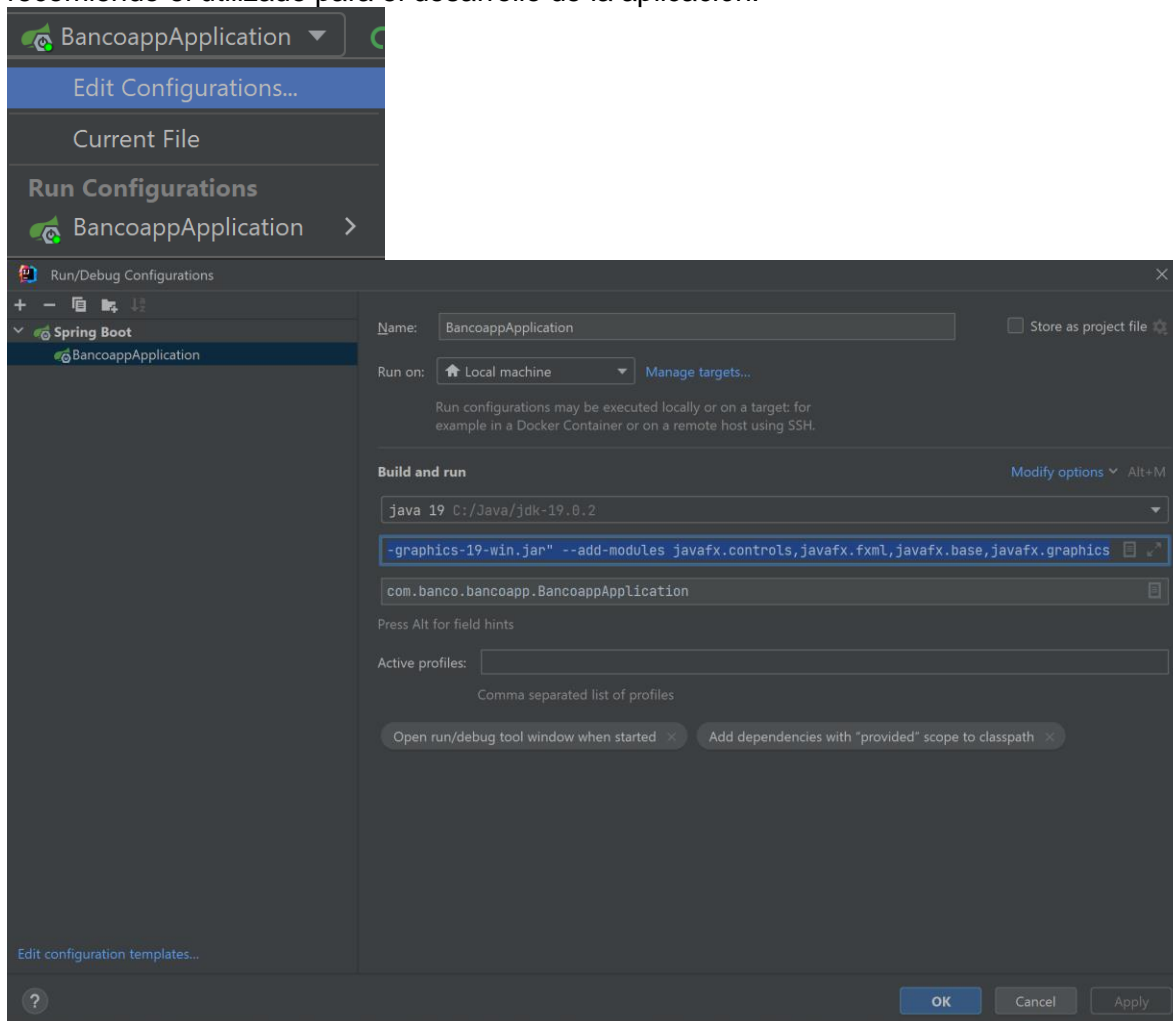
El sistema de gestor de base de datos utilizado fue MySQL Server, la base de datos se crea automáticamente al iniciar la aplicación gracias a la utilización de Spring Boot que tiene el proyecto. La conexión a la base de datos la he establecido con Xampp.

3. DOCUMENTACIÓN DEL SISTEMA

a. Manual de instalación y configuración de la aplicación

El proyecto se encuentra actualmente en mi repositorio de GitHub, el cual es público y accesible para ser clonado por cualquier usuario, o también puede descargarse el archivo .zip y descomprimirlo.

Está hecho en el IDE IntelliJ, pero no debe presentar ningún problema para extrapolarse a otros IDEs como NetBeans o Eclipse, aunque personalmente no lo he probado, por eso recomiendo el utilizado para el desarrollo de la aplicación.



Agregar los siguientes módulos:

```
--module-path  
"C:/Users/laura/.m2/repository/org/openjfx/javafx-controls/19/javafx-controls-19-win.jar";  
"C:/Users/laura/.m2/repository/org/openjfx/javafx-fxml/19/javafx-fxml-19-win.jar";  
"C:/Users/laura/.m2/repository/org/openjfx/javafx-base/19/javafx-base-19-win.jar";  
"C:/Users/laura/.m2/repository/org/openjfx/javafx-graphics/19/javafx-graphics-19-win.jar"  
--add-modules javafx.controls,javafx.fxml,javafx.base,javafx.graphics
```

Adaptándolos a la ruta path según su ordenador.

Las dependencias de Lombok, SpringBoot y Maven están incluidas en el pom del proyecto.

b. Manual de usuario

- La aplicación es bastante intuitiva, en cada una de sus ventanas está detallado como se debe proceder.
- Si es un usuario nuevo debe presionar en el botón "Registrarse" para darse de alta con una serie de datos que debe proporcionar, si ya se encontraba loggeado, solo debe introducir su nombre y contraseña.

- En su ventana de usuario puede ver sus cuentas, modificar sus datos, operar en una de sus cuentas(ingresar, retirar o transferir dinero), proporcionando el número de la misma.
- Puede ver con qué otros usuarios comparte cuenta, añadir más titulares o eliminarlos, siempre que proporcione el NIF.

4. CONCLUSIONES FINALES

a. Grado de cumplimiento de los objetivos fijados

Los objetivos fijados se cumplen en su totalidad, permitiendo que sea una aplicación funcional y bastante útil.

b. Propuesta de modificaciones o ampliaciones futuras del sistema implementado

Posibles modificaciones:

- Mejor diseño de su visualización
- Mayor desarrollo en cuanto a la captura de posibles errores o fallos
- Incorporar más funcionalidades
- Hacerla accesible para cualquier usuario, no solo los universitarios.

5. BIBLIOGRAFÍA

- www.google.com
- youtube
- github
- stackoverflow
- chat.openai

Proyecto 3.

Contexto del problema



Se pretende desarrollar una aplicación que gestione los usuarios y las cuentas de la entidad bancaria asociada al US, US-Caja. Cada miembro de la comunidad universitaria será considerado un potencial usuario de dicha caja, por tanto sus datos se almacenarán en los sistemas de información de la entidad US-Caja. Los datos almacenados para cada usuario potencial son: NIF, apellidos, nombre, año de nacimiento, dirección, e-mail y

teléfono de contacto. Cada usuario potencial del US-Caja podrá solicitar la creación de una o más cuentas

bancarias de las que será titular; dichas cuentas tendrán saldo 0 Euros inicialmente. Por otro lado, una misma cuenta bancaria podrá pertenecer a varios titulares, teniendo siempre al menos uno. La información almacenada para cada cuenta bancaria es la siguiente: número de cuenta, fecha de creación y saldo que contiene. Finalmente, los usuarios del US-Caja podrán realizar las siguientes operaciones bancarias en las cuentas en las que son titulares: ingresar dinero, retirar dinero y realizar una transferencia entre dos cuentas del US-Caja (siendo titular de la cuenta de la que se retira dinero). Para cada una de las operaciones bancarias se almacena la siguiente información: código de operación, tipo de operación, fecha de realización, cantidad involucrada en la operación, números de cuenta involucrados en la operación y usuario que realiza la operación.

Funcionalidades requeridas

Se debe diseñar una aplicación Java con una interfaz gráfica de usuario que permita:

- Crear nuevas cuentas bancarias para un determinado usuario.
- Modificar los datos de un determinado usuario.
- Dada una determinada cuenta bancaria, añadir/eliminar un titular de dicha cuenta.
- Listar todos los usuarios potenciales del US-Caja.
- Listar todas las cuentas bancarias de un determinado usuario potencial. Dicho usuario puede no ser titular de ninguna cuenta.
- Listar todas las operaciones bancarias realizadas sobre una determinada cuenta.
- Dado un determinado usuario y cuenta bancaria de la que es titular retirar o ingresar una determinada cantidad de dinero.
- Dado un determinado usuario y cuenta bancaria de la que es titular hacer una transferencia desde dicha cuenta a otra cuenta cualquiera de la entidad. La cuenta a la que se realiza la transferencia debe ser distinta a la cuenta de la que se retira el dinero. La realización de una transferencia implica la creación de dos operaciones: una operación de retirada de dinero y un ingreso en la cuenta destino.

En cada caso, se considerará la forma adecuada de interaccionar con los usuarios; por ejemplo, evitando que sea necesario escribir el nombre exacto de un usuario para poder visualizar la información relativa a este, evitando la existencia de tablas editables cuya actualización no se refleje en la base de datos, etc.