

# **Consigna del Trabajo Práctico -** **"Spring Banking"**

## **Descripción del Proyecto:**

El proyecto consiste en el desarrollo de un simulador de acciones que replicará un sistema de Homebanking denominado "Spring Banking". Los alumnos formarán equipos de 6 desarrolladores para crear los servicios necesarios que permitan el funcionamiento del sistema. Se trabajarán sobre tres entidades principales: Usuarios, Cuentas (ya sea cuenta corriente o caja de ahorro) y Transferencias.

## **Requisitos Técnicos:**

Utilizar Java 11 o 17 con Maven.

Conexión a la Base de Datos mediante JDBC.

Implementar Spring JPA y Hibernate para el manejo de la persistencia.

Configurar en el application.properties la conexión y las configuraciones (incluyendo ddl.auto=update).

### Arquitectura del Proyecto:

Se trabajará sobre un Sistema Monolítico con las siguientes capas:

Controllers

Services

Mappers

DTO (Data Transfer Objects)

Entities

Repositories

### Entidades y Relaciones:

Usuario:

Atributos: id, nombreUsuario, email, contraseña, dni, fecha\_nacimiento, domicilio (dirección), lista de cuentas (List<Cuenta>).

Cuenta:

Atributos: id, nombre, cbu, alias, monto, dueño (usuario dueño de la cuenta).

Transferencias:

id, monto, cuentaOrigen, cuentaDestino, fecha

### Tareas a Realizar:

Desarrollar un CRUD para cada entidad (Usuarios, Cuentas, y Transferencias).

Establecer las relaciones entre entidades, garantizando la integridad de los datos.

### Metodología de Trabajo:

División en Equipos: Se formarán equipos de 6 integrantes.

### Distribución de Tareas:

Cada equipo se encargará de una entidad (Usuarios, Cuentas, Transferencias). Utilizar Github y su metodología Github Flow o su derivado GitFlow.

En cada equipo, designar roles para cubrir el desarrollo de las capas definidas en la arquitectura del proyecto.

Coordinar reuniones regulares para revisar avances y resolver posibles obstáculos.

### Entregables:

Repositorio Git con el código del proyecto (Se entregará 1 repositorio por equipo, pero las ramas de cada integrante del grupo deben permanecer activas para el momento de la corrección).

### *Consideraciones Finales:*

Se evaluará la calidad del código, el manejo de las capas definidas, la correcta implementación de las relaciones entre entidades, y la funcionalidad del CRUD para cada entidad.