

## **Billetera Virtual (Nombre que desee el equipo)**

Para este tp vamos a desarrollar una mini-billetera virtual e implementar una solución Robusta y a la altura de Esp. Backend I.

Para eso vamos a necesitar un microservicio de clientes. (**api-customer**)

Su entidad principal va tener los siguientes atributos:

- Tipo de documento
- Número de documento
- Nombre
- Apellido
- Género
- Fecha de Nacimiento

Las operaciones básicas que va tener que implementar este microservicio son:

- POST (crear)
- PUT (modificar todos los campos excepto el tipo y número de documento)
- DELETE (solo baja lógica)
- GET (por tipo y número de documento)

**Además** otro microservicio que va estar en nuestra solución es **api-wallet**

Su entidad principal va ser Wallet y va tener los siguientes atributos:

- Tipo de documento
- Número de documento
- Código de Moneda (va a existir otra tabla **Moneda**(codigo, descripcion))
- Balance

Las operaciones básicas que va tener que implementar este microservicio son:

- POST (crear un registro en la billetera con esos datos)

- PUT (modificar solamente el balance)
- GET (obtener por tipo, número de documento y código de moneda el balance)
- GET (obtener por tipo y número de documento, todos los balances de todas las monedas)

### **Requerimientos Técnicos:**

- Ambos microservicios en JDK 17, con spring boot 2.7.5
- Base de datos relacional MySql
- Pueden programar en español o inglés a preferencia del equipo
- Todos los campos son obligatorios en ambas entidades
- Deben usar spring-jpa para hacer el mapeo clase-entidad
- Ambos proyectos deben estar al mismo nivel en el repo de github con los nombres correspondientes (api-customer) (api-wallet)
- Nomenclatura de paquetes com.dh.g(numero)
- Las entidades comparten base de datos, pero no se relacionan entre sí por FK. Solamente se puede relacionar Wallet con Moneda.