El sistema a desarrollar está planteado en el contexto del negocio bancario con enfoque en microservicios y base de datos no relacionales. Por lo tanto, en este enunciado solo reforzará conceptos manejados en el proyecto anterior.

## Bases a Desarrollar

Los desarrollos deben continuar con la base de conocimiento requerida y las bases del proyecto anterior, más las que se listan a continuación:

* Spring boot con Mongo DB
* Uso de Lombok para reducir código.
* Desarrollo de las nuevas funcionalidades con programación reactiva (Spring Webflux)
* Manejo de colecciones utilizando correctamente las APIs para Streams.

## Funcionalidades del sistema

**Gestión de Transacciones Bancarias**

1. **Descripción del Caso**:  
   El banco "XYZ" desea desarrollar un sistema de microservicios para registrar y gestionar el historial de transacciones bancarias de sus clientes. Las transacciones incluyen depósitos, retiros y transferencias entre cuentas. El historial de cada transacción debe ser registrado y estar disponible para su consulta.
2. **Detalles del Sistema**
3. **Gestión de Transacciones:**

* Cada transacción debe registrarse con la siguiente información:
  + - Tipo: Depósito, Retiro, Transferencia.
    - Monto: Cantidad monetaria.
    - Fecha: Fecha de la transacción.
    - Cuentas: Cuenta de origen y, en el caso de transferencias, cuenta de destino.
* Los endpoints para gestionar transacciones:
  + - POST /transacciones/deposito: Registrar un depósito,
    - POST /transacciones/retiro: Registrar un retiro
    - POST /transacciones/transferencia : Registrar una transferencia

1. **Reglas de Negocio**
2. **Validaciones de Transacciones:**
   * Los depósitos y retiros se aplican a cuentas
   * Las transferencias requieren que se indique la cuenta de destino y el monto.
   * No se pueden realizar retiros o transferencias si el saldo disponible es insuficiente.

## Requerimientos funcionales obligatorios del sistema

* Realizar diagramas de secuencia para mostrar el flujo de comunicación de los microservicios.
* Realizar diagramas de componentes para mostrar la arquitectura de los microservicios.
* Utilizar OpenAPI para documentar (usar contract first), Spring Boot para crear los microservicios.
* Desarrollar el microservicio para la gestión de transacciones.
* Utilizar MongoDB para almacenar el historial de transacciones de los usuarios.
* Cada documento representará una transacción con la información correspondiente
* Aplicar conceptos de POO, Programación funcional, programación Reactiva (spring webflux) para las funcionalidades de la solución.
* El sistema no tendrá implementado ninguna interfaz gráfica, la verificación de las funcionalidades se realizarán utilizando Postman.