

Prueba Técnica: Implementación de Servicio de Gestión de Clientes y Cuentas de Ahorros

Contexto

Una cooperativa requiere un servicio que gestione la información de sus clientes y sus cuentas de ahorros. Cada cliente tiene un número de identificación tributaria (NIT), nombre, y fecha de ingreso a la cooperativa. Las cuentas de ahorros están asociadas a un cliente y tienen un número de cuenta y un saldo disponible.

Requerimientos

Se solicita desarrollar un servicio en Spring o Spring Boot bien estructurado que implemente las siguientes funcionalidades (end points):

NOTA: Los datos de conexión a la base de datos se encuentran al final.

1. Crear Cliente

- **Descripción:** Crear un cliente en el sistema.
- **Entrada:** NIT, nombre, y fecha de ingreso del cliente.
- **Salida:** Un mensaje y un código que indiquen el resultado de la operación (éxito o error).

2. Consultar Cliente

- **Descripción:** Consultar la información de un cliente.
- **Entrada:** NIT del cliente.
- **Salida:** Toda la información asociada al cliente (NIT, nombre, fecha de ingreso).

3. Crear Cuenta de Ahorros

- **Descripción:** Crear una cuenta de ahorros asociada a un cliente.
- **Entrada:** NIT del cliente, número de cuenta de ahorros, y saldo inicial.
- **Validaciones:**
 - Verificar que el cliente exista.
 - Verificar que la cuenta de ahorros no exista previamente para ningún cliente.
- **Salida:** Un mensaje y un código indicando el resultado de la operación (éxito o error).

4. Generar Movimiento en la Cuenta de Ahorros

- **Descripción:** Realizar movimientos (créditos o débitos) en una cuenta de ahorros.
- **Entrada:** NIT del cliente, número de cuenta de ahorros, monto, y tipo de movimiento (DB para débito, CR para crédito).
- **Validaciones:**
 - Verificar que el cliente y la cuenta de ahorros existan.
 - Verificar que haya saldo suficiente en caso de un débito.
- **Salida:** Un mensaje y un código indicando el resultado de la operación.

Consideraciones Técnicas

- Desarrollo acorde a lo solicitado:
- Utilizar Spring o Spring Boot para el desarrollo del servicio.
- Manejo de Git con diferentes commits en el proceso (IMPORTANTE).
- Implementar validaciones y manejo de excepciones.
- Estructurar el código en capas: controladores, servicios, repositorios.
- Utilizar inyección de dependencias y gestión de transacciones cuando sea necesario.
- Se valorará la implementación de pruebas unitarias y/o de integración (opcional).
- Se valorará la documentación de los endpoints utilizando Swagger/OpenAPI (opcional).

Entregables

- Código fuente del proyecto en un repositorio (GitHub, GitLab).
- Archivo README.md con instrucciones para ejecutar el proyecto.

(Tiempo de prueba 4 horas)