



Instrucciones y consideraciones generales para el desarrollo de la prueba técnica

Prueba Técnica Java

Cargo: Analista programador backend Java

El objetivo de la prueba consiste en exponer servicios en Java basado en un modelo Oracle Financiero.

Una entidad financiera posee diversos clientes los cuales pueden ser personas naturales o jurídicas. Estos pueden tener varias cuentas asociadas donde a su vez están asociadas a movimientos los cuales pueden ser positivos y negativos.

Se debe de implementar las siguientes funcionalidades y condiciones:

Clientes

1. Si el cliente es una persona natural, los atributos requeridos serán los siguientes: nombre (máximo 80 caracteres), apellido (máximo 250 caracteres) tipo de documento, número de documento y RUT.
2. Si el titular es una persona jurídica, los atributos requeridos serán los siguientes: razón social (máximo 100 caracteres) año de fundación y RUT.
3. No pueden existir 2 titulares con el mismo RUT.

Cuentas

1. Las cuentas poseen un número de cuenta (valor requerido y único), una moneda (peso, dólar, euro) y un saldo (valor numérico de 2 decimales).
2. Las cuentas solo pueden eliminarse si no tienen movimientos asociados.

Movimientos

1. Una cuenta puede tener N movimientos.
2. Los movimientos no pueden eliminarse ni modificarse.
3. Cada vez que se incorpora un nuevo movimiento, y dependiendo de si es positivo y negativo, se debe actualizar el saldo de la cuenta asociada.
4. Si un movimiento genera un saldo en la cuenta mayor a 1'000.000 (para cuentas

en pesos), 300(para cuentas en dólares) o 150 (para cuentas en euros). se deberá rechazar ese movimiento.

En Oracle – PL SQL:

- Cree las tablas del modelo descrito.
- Realice la construcción de una función y de un procedimiento en un paquete aplicando alguna de las condiciones indicadas anteriormente.
- Cree las secuencias para los campos id auto-incrementales que se utilicen en el modelo elegido (Para mapear la secuencia en la Entidad Hibernate).

En Java:

- Mapear con Hibernate las tablas construidas en Oracle, debidamente relacionadas y utilizando las secuencias creadas anteriormente para los campos Id.
- Exponer los servicios Rest con Spring Boot realizando el CRUD de cada una de las funcionalidades especificadas.
- Exponer los servicios con Spring Boot invocando la función y el procedimiento creados anteriormente en Oracle.
- Los ids principales se deben manejar de tipo uuid;

Bonus:

- Aplicar prácticas de clean code.
- Implementar la solución en arquitecturas hexagonales.
- Verificar el coverage de la aplicación mediante la construcción de pruebas unitarias con JUnit.