



Prueba Técnica Senior Engineer

Ejercicio Práctico

1. Contexto del Problema

Tu equipo trabaja en un ecosistema de microservicios para una billetera digital. en el cual se realizan movimientos de carga y resta de saldo "procesamiento de transacciones".

2. Requerimientos

Crear un microservicio que procese transacciones financieras.

Input: Mensaje transacción.

Cada transacción tiene los siguientes campos:

- transaction_id
- user_id
- amount
- type (deposit o withdraw)
- timestamp

Output: Registro de la transacción en una base de datos PostgreSQL o similar (BD Relacional).

Incluir:

- Validación de datos (e.g., que el amount sea positivo).
- Simulación de operación de negocio (e.g., actualización del balance de un usuario en base a la transacción).
- Registro de logs para seguimiento de errores o comportamientos sospechosos.

Implementar:

- API REST para guardar una transacción, consultar el historial de transacciones por `user_id`, y consultar el saldo actual por `user_id`.
- Persistencia en PostgreSQL o similar usando tablas normalizadas.
- Configuración de despliegue en Docker preferiblemente para Kubernetes (e.g., un Deployment y un Service).

3. Extras (**Opcional**)

- Un mecanismo básico de alerta para detectar intentos de fraude (e.g., transacciones consecutivas de alto monto en un corto tiempo).
- Pruebas unitarias y de integración.

Preguntas Conceptuales

1. ¿Cómo manejarías picos altos de transacciones para garantizar escalabilidad?
2. ¿Qué estrategias usarías para prevenir fraudes en un sistema de billetera digital?
3. Si detectas lentitud en el procesamiento de transacciones por alta concurrencia, ¿cómo procederías para mejorar el rendimiento?

Enviar url del proyecto en GitHub y las respuestas a las preguntas conceptuales dejarlas en el `readme.md`