

2023.v1 - Prueba Técnica para el Rol SR DEVELOPER Nera

Parte 1: Backend (Necesaria, validación de skills principales del candidato)

Tecnologías Deseables: Spring Boot, MongoDB

Desarrollar con buenas prácticas de arquitectura de software los siguientes endpoints:

1. POST /accounts: Crea una nueva cuenta bancaria. Este endpoint debería aceptar detalles de la cuenta como el nombre y el número de cuenta, y devolver un ID de cuenta.
2. GET /accounts/<id>/balance: Obtiene el balance de la cuenta bancaria.
3. POST /transactions: Realiza una transacción bancaria. Este endpoint debería aceptar detalles de la transacción como el ID de la cuenta, el tipo de transacción (depósito o retiro) y el monto de la transacción.

Notas

- a) Se requiere el desarrollo con código limpio implementando los Patrones de Diseño que consideres para llegar al objetivo. El sistema debe utilizar Event Sourcing para el almacenamiento y recuperación de los cambios en las tareas. Cada acción realizada sobre una tarea (creación, actualización, eliminación, finalización) debe ser registrada como un evento y almacenada en un registro de eventos.
- b) Implementa un middleware que registre en la consola cada vez que se realiza un depósito de más de 10,000 US\$.
- c) Dejar docblocks en toda acción que incluya lógica de negocio.
- d) Se debe incluir el Docker Compose necesario para ejecutar la solución y las dependencias que consideres.
- e) Se debe incluir un archivo para acceder a los endpoints con POSTMAN
- f) Se debe crear un archivo README.md que incluya toda la información para ejecutar el código y sus pruebas.

Parte 2: Frontend (Opcional, validación de skills secundarios del candidato)

Tecnologías Deseables: React con o sin framework

Desarrolla una aplicación de interfaz de usuario que permita a los usuarios abrir una nueva cuenta bancaria y realizar transacciones bancarias como depósitos y retiros.

1. La aplicación debe permitir a los usuarios introducir los detalles de su cuenta, como el nombre, el número de cuenta y el saldo inicial.
2. Los usuarios deben poder realizar depósitos y retiros, introduciendo el monto de la transacción y seleccionando el tipo de transacción.
3. La aplicación debe mostrar el balance actualizado de la cuenta después de cada transacción.
4. La aplicación debe trabajar con state management.

Para Ambos casos: Compartir un repositorio github privado con el usuario `enoyanera` e
Incluir un archivo README que explique cómo ejecutar el código y las pruebas pensando
que quien lo lee no tiene conocimiento del servicio.