

---

# Prueba Técnica para el rol de Staff Software Engineer

## Parte 1: Frontend

### Tecnologías: a criterio

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 **CSS in JS** y **state management**.
5. Implementa pruebas unitarias con Jest y asegúrate de que la cobertura de las pruebas sea mayor al 80%.
6. Incluye un archivo README que explique cómo ejecutar el código y las pruebas.

## Parte 2: Backend

### Tecnologías: a criterio

Desarrollar con buenas practicas 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.

Además, implementa un middleware que registre en la consola cada vez que se realiza un depósito de más de 10,000 US\$.

## Requerimientos de Arquitectura

El sistema debe seguir los principios y patrones de Domain-Driven Design (DDD) para el diseño de la arquitectura y la estructura del código.

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, marcado como completada) debe ser registrada como un evento y almacenada en un registro de eventos.

Implementa pruebas unitarias con Jest y asegúrate de que la cobertura de las pruebas sea completa. Incluye un archivo README que explique cómo ejecutar el código y las pruebas.

Debes entregar un diagrama de alto nivel de como implementarías la solución en AWS.

## Requisitos Generales

1. Debes asegurarte de que el código sea de alta calidad, fácil de mantener y cumpla con los principios de diseño y buenas prácticas
2. El Sistema completo debe poder levantarse en local mediante el uso de `docker-compose`.

Por favor, ten en cuenta que el código que entregues puede estar incompleto en cuanto a funcionalidad, pero siempre debe ser capaz de compilar y ejecutar las pruebas.

## Instrucciones de Entrega

Una vez que hayas completado la prueba, por favor sigue estas instrucciones para entregarla:

1. Crea un nuevo repositorio en GitHub para tu proyecto.
2. Asegúrate de que el repositorio es privado.
3. Sube tu código al repositorio.
4. Agrega al usuario `lucas-luongo`, `HaroldGarzon`, `advargas`, `fedexepelin` y `mauriciosneiraxepelin` como colaborador del repositorio para que pueda revisar tu código.

Por favor, envía el enlace del repositorio en tu respuesta a esta prueba.

---

Espero que esta prueba técnica revisada sea útil. Si tienes alguna otra pregunta o necesitas más detalles, no dudes en preguntar.

