

The Asset Management Digital Challenge

Aspectos a tener en cuenta:

- La solución debe contener tests unitarios con una cobertura suficiente. Todos los tests deben ejecutarse y funcionar correctamente al hacer gradle build.
- Aun habiendo funcionado los tests unitarios, se debe probar la aplicación en local. Para probar los endpoints se puede usar una herramienta como por ejemplo Postman (<https://www.postman.com/>). Se probarán los siguientes casos:
 - Transferencia desde una cuenta inexistente: la app devolverá un mensaje de error específico indicando que la cuenta no existe.
 - Transferencia desde una cuenta con dinero insuficiente: la app devolverá un mensaje de error específico indicando que no hay suficiente balance en la cuenta.
 - Transferencia con una cantidad igual a cero: la app devolverá un mensaje de error específico indicando que la cantidad debe ser mayor que cero.
 - Transferencia con una cantidad negativa: la app devolverá un mensaje de error específico indicando que la cantidad debe ser mayor que cero.
 - Transferencia entre dos cuentas que son la misma: la app devolverá un mensaje de error específico indicando que la cuenta origen no puede ser la misma que la cuenta destino.
 - Transferencia correcta: se comprobará que los balances de las dos cuentas se han actualizado correctamente en el repositorio (invocando al método GET).
- La solución debe permitir realizar varias transferencias al mismo tiempo (threads simultáneos).
- La solución debe evitar deadlocks, es decir, que si ocurren dos transferencias simultáneas de la cuenta A a la cuenta B, y de la cuenta B a la cuenta A, el sistema no se queda colgado (más información sobre deadlocks y cómo testearlos aquí: <https://richardbarabe.wordpress.com/2014/02/21/java-deadlock-livelock-and-lock-starvation-examples/>, <https://www.javahelps.com/2015/06/thread-deadlock.html>, <https://www.baeldung.com/java-testing-multithreaded>). Incluir un test unitario que demuestre que no se produce deadlock en el código.
- La solución debe incluir un test unitario (utilizando un mock) para verificar que se invoca el servicio de notificación tras la transferencia.
- Incluir un README.md lo suficientemente amplio y explicativo, con todos los puntos y mejoras que harían falta para hacer de esta app un producto entregable (uso de bbdd reales, escalabilidad, integración continua, auditoría de operaciones, uso de contenedores para facilitar despliegue, uso de BDD... todo lo que sirva para demostrar que se está familiarizado con estos temas).