

Objetivo:

Realizar una aplicación web que simule el funcionamiento de un banco. Las entidades a manejar son 3: Cliente, Cuenta y Movimiento. Un Cliente puede tener varias cuentas y cada cuenta puede tener varios movimientos. Un cliente tiene: nombre, dirección y teléfono. Una cuenta tiene: número y saldo. Un movimiento tiene: tipo (si es débito o crédito; los débitos aumentan y los créditos reducen el saldo), fecha y valor.

El sistema debe permitir:

- La creación, modificación, consulta y eliminación de Clientes y Cuentas.
- Registrar movimientos en una cuenta. El sistema debe validar que no se permitan registrar movimientos que dejen un saldo negativo en la cuenta.
- Generar un reporte que, especificando un rango de fechas y un cliente, visualice las cuentas asociadas con sus respectivos saldos y el total de débitos y créditos realizados durante las fechas.
- Se deben mostrar en la parte superior de las pantallas los apuntes de Chuck Norris (La idea es simular un componente de publicidad):
 - o Los apuntes de Chuck Norris se encuentran en el siguiente api público, <https://api.chucknorris.io/>
 - o <https://api.chucknorris.io/jokes/random>

Las otras características o restricciones funcionales del aplicativo pueden ser establecidas de acuerdo al análisis que realice del problema.

Las condiciones para el desarrollo son las siguientes (no modificables):

- Debe ser una aplicación Web.
- Se debe utilizar alguno de los siguientes Frameworks Angular (8 – 9 superior) capa Front-End.
- Se debe utilizar alguno de los siguientes Frameworks SpringBoot capa Back-End usar JDK 1.8 o superior.
- Se debe mantener el estado de la aplicación en la memoria del Navegador.
- Debe tener cobertura de pruebas unitarias.
- Pruebas unitarias tanto back(JUnit) como front(Karma).

Los entregables de la prueba son:

1. Código fuente de la aplicación en GITHUB o zip.
2. README, con instrucciones claras para poder instalar la aplicación
3. Diagrama Relacional que describa las relaciones entre los modelos.

Tener en cuenta los correspondientes porcentajes dados como calificación de la prueba:

1. Código fuente de la aplicación en GitHub o zip → 10%
2. Descripción clara de pasos a seguir en archivo README → 10% (De no contar con las debidas aclaraciones a seguir y de no ser un reflejo de los pasos para el uso de la aplicación no será tomada en cuenta la revisión).
3. Pruebas Unitarias (Front, Back) → 30%
4. Pruebas E2E (Front) → 20%
5. Solución del Problema → 30%

LA SUMA DE TODOS PORCENTAJES ES DE 100% PARA PASAR LA PRUEBA SE REQUIERE TENER EL 80% COMO MINIMO.