Prueba Técnica

ArquitecturaMicroservicio

Indicaciones generales

- Considerar todos los puntos mencionados más adelante para realizar el ejercicio: tecnologías, herramientas, casos de uso.

Back End

- Aplique todas las buenas prácticas, patrones Repository, etc que considere necesario (se tomará en cuenta este punto para la calificación). El manejo de entidades se
- debe manejar con: JPA / Entity Framework Core Se debe manejar mensajes de
- excepciones. Se debe realizar como mínimo dos pruebas unitarias de los endpoints.
- La solución se debe desplegar en Docker. Posterior a la entrega de este ejercicio, se
- estará agendando una entrevista técnica donde el candidato deberá defender la
- solución planteada.

Front End

- Se debe implementar de buenas prácticas (clean code, SOLID)
- Se debe implementar pruebas unitarias.
- No se debe usar ningún framework de estilos o componentes pre-fabricados para maquetar el diseño propuesto, por ejemplo: Material, Bootstrap, PrimeNg, entre otros.

Herramientas y tecnologías utilizadas

BackEnd

- Java spring boot
- IDE de su preferencia
- Base de Datos Relacional
- Postman v9.13.2 (validador de API)

Front End

- Angular o React
- Testing Angular: Jest
- Testing React: Reacting Testing Library

Back End

Creación de Api Rest “Aplication Programming Interface”

Manejarlos verbos: Get, Post, Put, Push, Delete

Persona

- Implementar la clase persona con los siguientes datos: nombre, genero, edad, identificación, dirección, teléfono
- Debe manera su clave primaria (PK)

Cliente

- Cliente debe manejar una entidad, que herede de la clase persona.
- Un cliente tiene: clienteid, contraseña, estado.
- El cliente debe tener una clave única. (PK)

Cuenta.

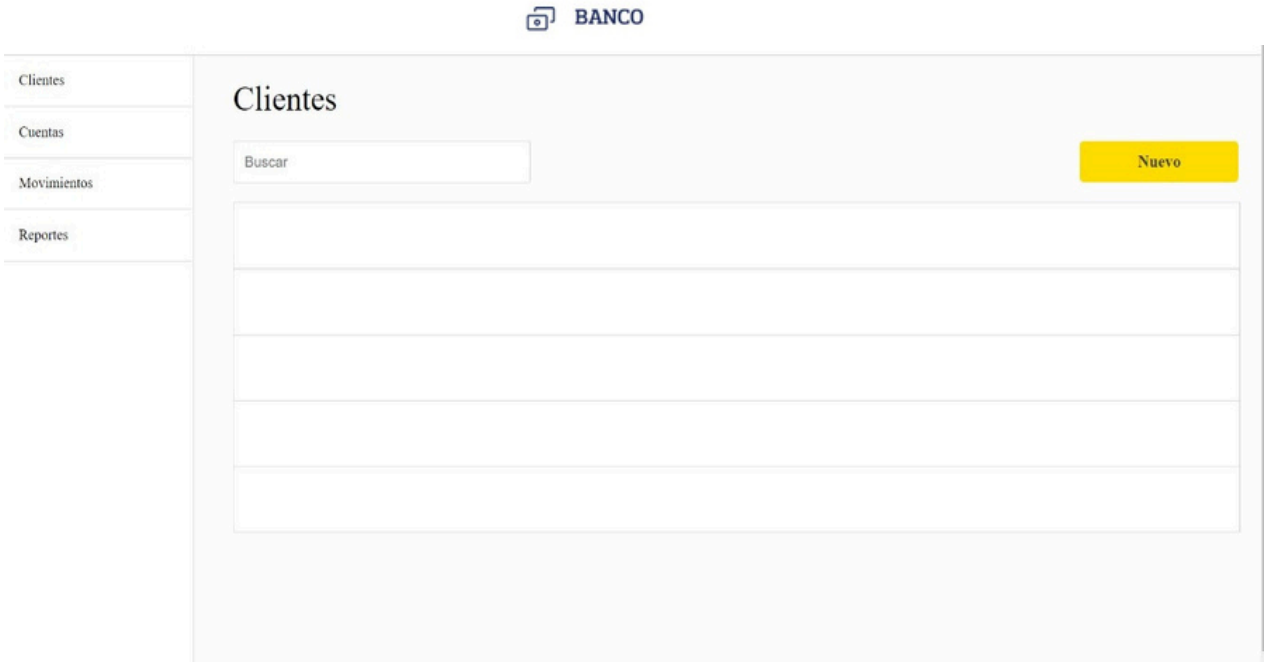
- Cuenta debe manejar una entidad
- Una cuenta tiene: número cuenta, tipo cuenta, saldo Inicial, estado.
- Debe manejar su Clave única

Movimientos

- Movimientos debe manejar una entidad
- Un movimiento tiene: Fecha, tipo movimiento, valor, saldo
- Debe manejar su Clave única

Funcionalidades

Front End



- Adaptar el layout propuesto para el diseño de las entidades (Cliente, cuentas, movimientos, reportes). Realizar la maquetación de los CRUD de las entidades
- propuestas (Cliente, cuentas, movimientos).
  Presentar de manera visual a nivel de reporte los movimientos realizados utilizando
- el endpoint correspondiente. También incluir un botón que permita descargar el reporte en formato PDF.
  Tener una acción que nos permita realizar las búsquedas rápidas de cualquier
- registro de las tablas. Los mensajes de validación se deben visualizar en pantalla.

-

Back End

Los API'sdebe tener las siguientes operaciones:

Podrá Crear, editar, actualizar y eliminar registros (Entidades: Cliente, Cuenta y Movimiento).

Los endpoints a crear son:

- /cuentas
- /clientes
- /movimientos

- Los valores cuando son crédito son positivos, y los débitos son negativos. Debe almacenarse el saldo disponible en cada transacción dependiendo del tipo de movimiento. (suma o resta).
- Si el saldo es cero, y va a realizar una transacción débito, debe desplegar mensaje "Saldo no disponible".
  Generar reporte (Estado de cuenta) 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 de ese cliente. Tomar en consideración que también se debe obtener los resultados del reporte en formato base64 (PDF) y Json. Por ejemplo:
  (/reportes?fecha=rango fechas)

Casos de Uso (Ejemplos)

1. Creación de Usuarios.

Nombres	**Dirección**	**Teléfono**	**Contraseña**	**estado**
Jose Lema	Otavalo sn y principal	098254785	1234	True
MarianelaMontalvo	Amazonas y NNUU	097548965	5678	True
Juan Osorio	13 junio y Equinoccial	098874587	1245	True

2. Creación de Cuentas de Usuario.

NumeroCuenta	**Tipo**	**Saldo Inicial**	**Estado**	**Cliente**
478758	Ahorro	2000	True	Jose Lema
225487	Corriente	100	True	Marianela Montalvo
495878	Ahorros	0	True	Juan Osorio
496825	Ahorros	540	True	Marianela Montalvo

3. Crear una nueva Cuenta Corriente para Jose Lema

Numero Cuenta	Tipo	Saldo Inicial	Estado	Cliente
585545	Corriente	1000	True	Jose Lema

4. Realizar los siguientes movimientos

NumeroCuenta	Tipo	Saldo Inicial	Estado	Movimiento
478758	Ahorro	2000	True	Retiro de 575
225487	Corriente	100	True	Depósito de 600
495878	Ahorros	0	True	Depósito de 150
496825	Ahorros	540	True	Retiro de 540

5. Listado de Movimiento, por fechas por usuario.

Fecha	Cliente	Numero Cuenta	Tipo	Saldo Inicial	Estado	Movimiento	Saldo Disponible
10/2/2022	Marianela Montalvo	225487	Corriente	100	True	600	700
8/2/2022	Marianela Montalvo	496825	Ahorros	540	True	-540	0

Ejemplo Json:

```
{
        "Fecha":"10/2/2022",
        "Cliente":"Marianela Montalvo",
        "Numero Cuenta":"225487"
        "Tipo":"Corriente",
        "Saldo Inicial":100,
        "Estado":true,
        "Movimiento":600,
        "Saldo Disponible":700
}
```

Instrucciones de despliegue

- Generar el script de base datos, entidades y esquema datos, con el nombre BaseDatos.sql.
- Desplegar en Docker la solución.
- Ejecutar Postman para poder realizar las verificaciones
  (http://{servidor}:{puerto}/api/{metodo}...{Parámetros})

Entregables

- **La Solución debe ser cargado a un repositorio git público, se debe enviar la ruta de este repositorio.**

- **Descarga archivo Json, de Aplicación Postman, para validación de los endpoints (agregarlo al reporitorio git).**
- **Se debe entregar antes de la fecha y hora indicada por correo.**

¡¡¡Muchos éxitos!!!