**Prueba Técnica**

**Arquitectura Microservicio (Nivel Intermedio)**

# Indicaciones generales

* 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 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.

# Herramientas y tecnologías utilizadas

* NetCore 5 o superior
* IDE de su preferencia
* Base de Datos Relacional
* Postman v9.13.2 (validador de API)

Creación de Api Rest “Aplication Programming Interface”

Manejar los 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

Los API’s debe 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. (resultado en 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 |
| Marianela Montalvo | Amazonas y NNUU | 097548965 | 5678 | True |
| Juan Osorio | 13 junio y Equinoccial | 098874587 | 1245 | True |

1. Creación de Cuentas de Usuario.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numero Cuenta** | **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 |

1. Crear una nueva Cuenta Corriente para Jose Lema

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

1. Realizar los siguientes movimientos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numero Cuenta** | **Tipo** | **Saldo**  **Inicial** | **Estado** | **Movimiento** |
| 478758 | Ahorro | 2000 | True | Retiro de 575 |
| 225487 | Corriente | 100 | True | Deposito de 600 |
| 495878 | Ahorros | 0 | True | Deposito de 150 |
| 496825 | Ahorros | 540 | True | Retiro de 540 |

1. Listado de Movimiento, por fechas x 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.
* Se debe entregar antes de la fecha y hora indicada por correo.

Muchos Éxitos en su Prueba…!!!