Diseño de una **REST API** en **asp.net Core 6.**

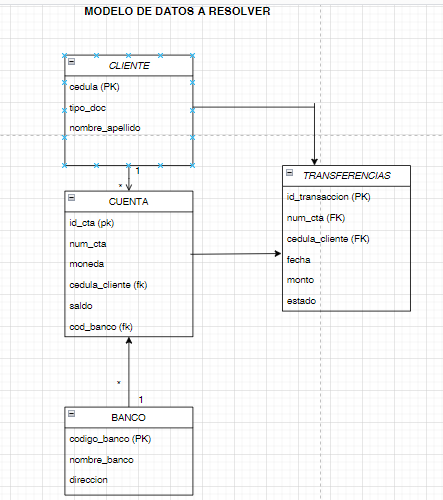
Diseñar una REST API para atender al siguiente caso de negocio:

Crear una API de **Transferencia** entre Cuentas Interbancarias. La misma debe recibir la como parámetros Cod\_banco\_origen, Nombre Banco de Origen, CTA Origen, Cod\_banco\_destino, Nombre Banco Destino, CTA Destino, fecha de operación, moneda, monto a transferir y estado.

Realizar apis con los sgtes. EndPoints:

1. ABM de Cliente: Crear, modificar, Obtener, Eliminar Clientes. El número de documento no se debe repetir.
2. ABM de Bancos: Crear, modificar, Obtener, Eliminar Bancos. Código del banco debe ser GUID en la API.
3. ABM de Cuentas: Crear, modificar, Obtener, Eliminar Cuenta de Bancos. Código de Cuenta debe ser GUID en la API.
4. Crear un método que permita realizar una Transferencia de un cliente de un banco a otro cliente de otro banco.
   1. Al realizar una transferencia, el saldo de la cuenta de origen debe disminuir, y el saldo de la cuenta de destino debe aumentar.
   2. Validar que la cuenta de origen posea saldo para realizar la operación, caso contrario debe manejar una excepción por “insuficiencia de fondos”.
   3. Validar que el Banco Ordenante y beneficiario NO sean los mismos, caso contrario emitir una excepción por “Bancos NO deben ser iguales”.
5. Crear un EndPoint que permita modificar el estado de una transferencia por ID.
6. Crear una API que permita consultar el estado de una transferencia, ya sea que haya sido aprobada, en proceso o rechazada.
7. Consultar el historial de transferencias Enviadas y Recibidas por una cuenta especifica o número de cedula de cliente, o todas en general.

**Modelo de BD**



# **Entregables:**

* Repositorio Github con el código fuente
* Instrucciones para ejecutar la aplicación, explicar detalladamente en un Word.
* Colección Postman para probar la API. Métodos a ejecutar.
* Diagrama de la arquitectura de la solución.
* Incluir capturas de pantalla de las pruebas realizadas en postman o en el navegador.

***Obs:*** *Todos los entregables pueden estar en el mismo repositorio de Github*

# **Consideraciones:**

* La API debe exponer una documentación en Swagger. Ver como implementar Swagger en la Api.
* Utilizar la Base de Datos **PostgreSQL,** para almacenar los datos en la BD.
* Definición de recursos disponibilizados por la API
* Uso de verbos HTTP apropiados
* Uso correcto de los estados HTTP
* Manejo de excepciones
* Uso correcto de logs de aplicación (logs descriptivos, bien categorizados según necesidad)
* Interfaces bien diseñadas
* Uso de estándares y arquitectura de software adecuada
* Código mantenible y extensible
* Código autodescriptivo

***Obs:*** *Luego de entregar el código fuente, se solicitará cambios mínimos mediante Github Issues.*