

Guía de uso - Sistema Banco

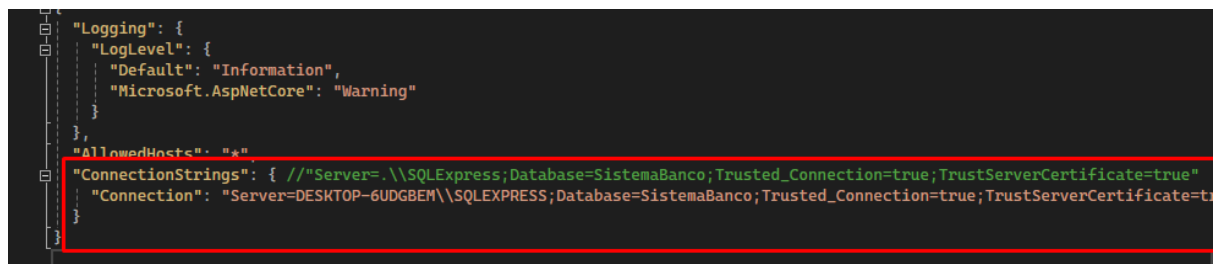
El siguiente documento es una explicación breve de los pasos para probar el funcionamiento del proyecto de "Sistema Banco".

Tendremos 2 proyectos, uno para el backend y otro para el frontend:

Proyecto Backend: SistemaBancoBack

Aquí debemos de actualizar la cadena de conexión a sql server, agregando nuestro name local, se encuentra en el archivo:

SistemaBancoBack/SistemaBancoBack/appsettings.json



```
"Logging": {
  "LogLevel": {
    "Default": "Information",
    "Microsoft.AspNetCore": "Warning"
  },
  "AllowedHosts": "*"
},
"ConnectionStrings": {
  "Default": "Server=.;Database=SistemaBanco;Trusted_Connection=true;TrustServerCertificate=true"
},
"Connection": "Server=DESKTOP-6UDGBEM\\SQLEXPRESS;Database=SistemaBanco;Trusted_Connection=true;TrustServerCertificate=true"
```

Ahora debemos abrir la consola de paquetes y crear las migraciones, se debe utilizar los siguientes comandos(eliminar primero la carpeta Migrations y luego seguir ejecutando los siguientes comandos):

Add-Migration InitialCreate

Update-Database

Realizará las migraciones correspondientes. Ahora podemos ejecutar el proyecto. Luego se debe dirigirse al script que se encuentra en el repositorio.

Script_SistemaBancoDB.sql

Y debe copiar los insert siguientes y ejecutar o si desea créalos con el endpoint en la colección de postman en la sección de "Tipo de transacción": (leer la nota).

```
INSERT INTO TipoTransaccion (Descripcion)
VALUES ('Compra'), ('Pago');
```

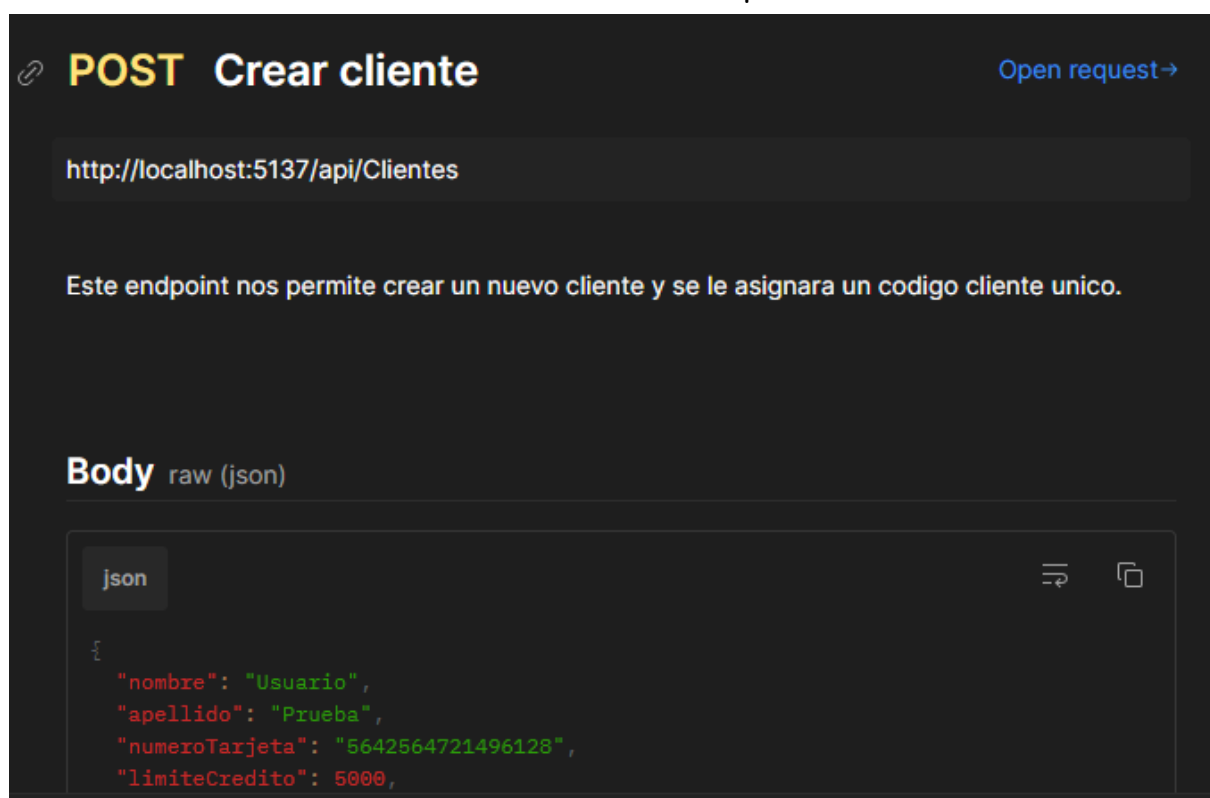
Nota: Este paso es sumamente importante, esto para tener el tipo de transacción compra con el código 1 y pago como número 2.

Ya teniendo creados el tipo de transacción con el orden de Compra 1 y pago 2. Debemos utilizar el endpoint para crear una configuración, que se encuentra en la sección de "Configuración" o debemos copiar y ejecutar el siguiente INSERT;

--configuraciones iniciales

```
INSERT INTO Configuracion (PorcentajeInteres, PorcentajeSaldoMinimo)
VALUES (25, 5);
```

A este punto ya podemos crear un nuevo usuario usando el endpoint de crear cliente en la sección de Cliente de la colección de postman:



Ahora ya podemos ejecutar el proyecto del frontend llamado "SistemaBancoFront".

Nota: Al momento de ejecutar el proyecto SistemaBancoBack, verá los endpoints que se utilizarían para el estado de cuenta y mostrar las compras del mes actual y anterior, solo que este por motivos de tiempo ya no pude seguir, de igual forma en el script

"Script_SistemaBancoDB.sql" se encuentra los procedimientos almacenados, los cuales hacen esa parte de la funcionalidad.

```
42
43 --##### PROCEDIMIENTO ALMACENADO - ESTADO DE CUENTA #####
44 CREATE PROCEDURE ObtenerEstadoGeneralCliente
45     @CodigoCliente INT
46 AS
47 BEGIN
48     DECLARE @SaldoTotal DECIMAL(10, 2);
49     DECLARE @PorcentajeInteres DECIMAL(5, 2);
50     DECLARE @InteresBonificable DECIMAL(10, 2);
51
52     SELECT --Obtener configuración
53         @PorcentajeInteres = PorcentajeInteres
54     FROM Configuracion;
55
56     --Calcular saldo total
57     SELECT @SaldoTotal =
58         SUM(CASE
59             WHEN tt.CodigoTipoTransaccion = 1 THEN t.Monto -- Compras(Cargos)
60             WHEN tt.CodigoTipoTransaccion = 2 THEN -t.Monto -- Pagos(Abonos)
61             ELSE 0
62         END)
63     FROM Transaccion t
64     INNER JOIN TipoTransaccion tt ON t.CodigoTipoTransaccion = tt.CodigoTipoTransaccion
65     WHERE t.CodigoCliente = @CodigoCliente;
66
67     --Si no hay transacciones, asignar un saldo total de 0
68     SET @SaldoTotal = ISNULL(@SaldoTotal, 0);
```

Y también se encuentra para llamar estos procedimientos almacenados:

```
--Ejecutar los procedimientos
EXEC ObtenerEstadoGeneralCliente @CodigoCliente = 1;
EXEC ObtenerTransaccionesMesActual @CodigoCliente = 1;
EXEC ObtenerTotalesCompras @CodigoCliente = 1;
```

Muchas gracias por tomarme en cuenta en este proceso.