# **Bank Application**

This is a simple Bank application built with .NET Core 6 and EntityFramework, with the database in PostgreSQL 15. The application provides APIs for Banks, Customers, Accounts and Transfers.

## **Prerequisites**

To run this application, you'll need to have the following software installed:

* .NET Core 6
* PostgreSQL 15

## **Getting Started**

1. Clone the repository: git clone https://github.com/franja1994/BankAPITest.git
2. Open the solution in Visual Studio 2022 (Sugestión) or any IDE of your preference.
3. Update the connection string in appsettings.json to point to your local PostgreSQL instance:

{ "ConnectionStrings": {

"<YourConectionName>": "Host=<localhost>; Database=<BankAPI>; Username=<postgres>; Password=<password>;"},

1. Replace the values inside the <> with yours.
2. Run the application.

## **Endpoints**

The following endpoints are available:

* GET /api/v1/Banks: get a list of all Banks
* GET /api/v1/Banks/{code}: get a specific Bank by BIC
* POST /api/v1/Banks: create a new Bank
* PUT /api/v1/Banks/{code}: update an existing Bank
* DELETE /api/v1/Banks/{code}: delete a Bank
* GET /api/v1/Customers: get a list of all Customers
* GET /api/v1/Customers/{documentNumber}: get a specific Customer by Document
* POST /api/v1/Customers: create a new Customer
* PUT /api/v1/Customers/{documentNumber}: update an existing Customer
* DELETE /api/v1/Customers/{documentNumber}: delete a Customer
* GET /api/v1/Accounts: get a list of all Accounts
* GET /api/v1/Accounts/{number}: get a specific Account by Number
* POST /api/v1/Accounts: create a new Account
* PUT /api/v1/Accounts/{number}: update an existing Account
* DELETE /api/v1/Accounts/{number}: delete an Account
* GET /api/v1/Transfers: get a list of all Transfers
* POST /api/v1/Transfers: create a new Transfer
* PUT /api/v1/Transfers/State/{id}: update an existing Transfer
* GET /api/v1/Transfers/State/{id}: get the state of a transfer
* GET /api/v1/Transfers/Account/{number}: get a list of all Transfers by Account
* GET /api/v1/Transfers/Customer/{number}: get a list of all Transfers by Customer