



**DOCUMENTACION DE PRUEBA TECNICA PARA LA POSICION DE
DESARROLLADOR DE SERVICIOS WEB**

ELABORADOR POR:

CHRISTIAN ARMANDO DIAZ LOPEZ

Contenido

API REST.....	3
ARQUITECTURA DE LA SOLUCION	4
Probar El Proyecto	5
Contacto.....	8

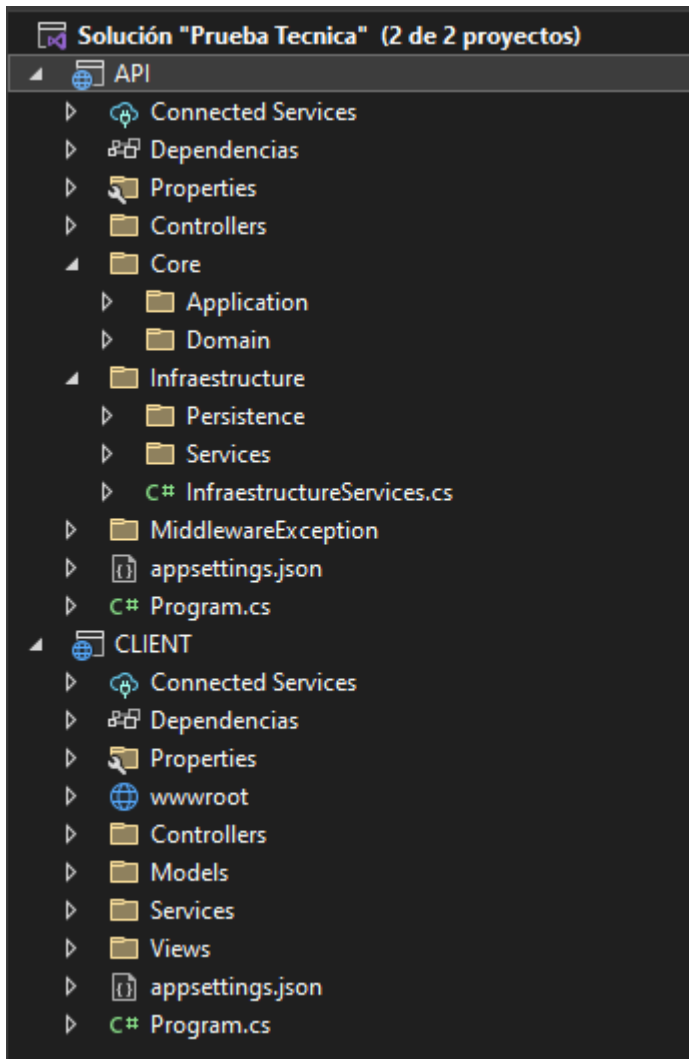
API REST

La API se ha desarrollado como parte de una aplicación web con .net 6 diseñada para proporcionar una interfaz que permite a los usuarios visualizar el estado de sus cuentas de tarjetas de crédito. Esta interfaz muestra detalles como el saldo total, el saldo disponible, la cuota mínima y el monto total a pagar. Además, los usuarios pueden ver todas las compras realizadas durante el mes, así como insertar nuevas compras y pagos, y revisar todas las transacciones mensuales realizadas por los clientes.

Para mejorar la experiencia del usuario, también se ha creado una interfaz gráfica con un diseño amigable que consume el API. Esto facilita la interacción del cliente con la aplicación.

Adicionalmente, se ha integrado Swagger para documentar y consultar fácilmente los endpoints del API, y se ha agregado una colección de Postman que permite probar y explorar todas las funcionalidades disponibles en el API de manera eficiente.

ARQUITECTURA DE LA SOLUCION



Interfaz grafica: Esta contiene lógica del frontend, el cual esta hecho con asp.net framework mvc (modelo-vista-controlador) implementando tecnologías como Razor y librerías como closedxml para manipular datos Excel) la cual ofrece una interfaz intuitiva para interactuar con la API

Proyecto API: Este proyecto se centra en la lógica del backend, desarrollado con ASP.NET Core Web API. Interactúa con una base de datos SQL Server utilizando procedimientos almacenados para realizar diversas operaciones. La arquitectura se basa en Clean Architecture (Arquitectura Limpia), implementando el patrón de diseño

CQRS (Command Query Responsibility Segregation) y siguiendo los principios SOLID. Esto permite una separación clara de las responsabilidades, facilita la prueba unitaria y la mantenibilidad del código. Además, la API utiliza CORS para gestionar las solicitudes desde diferentes dominios y está documentada de manera interactiva con Swagger, lo que permite probar fácilmente los endpoints de la API y entender su funcionalidad sin necesidad de conocer los detalles internos de implementación.

Probar El Proyecto

1. Clonar el repositorio desde GitHub:

Primero, clonamos el repositorio alojado en GitHub:

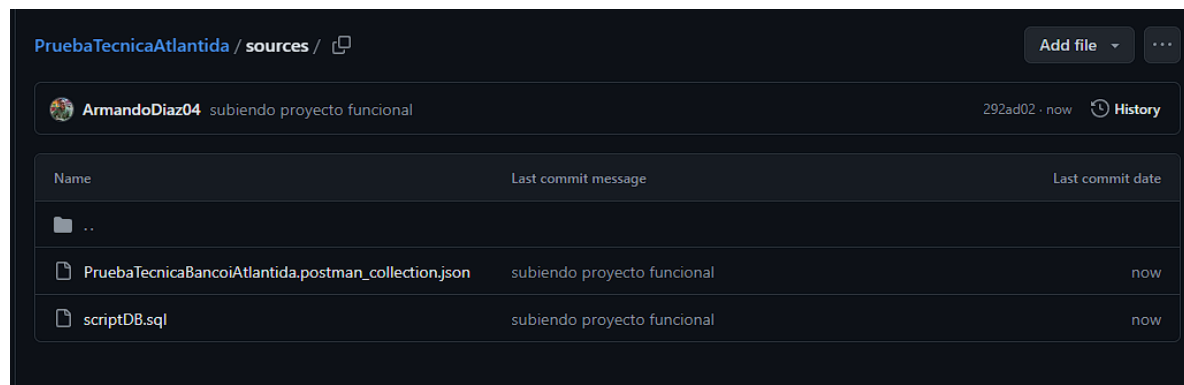
<https://github.com/ArmandoDiaz04/PruebaTecnicaAtlantida.git>

2. Abrir la solución en Visual Studio:

Después de clonar el repositorio, abrimos la solución utilizando Visual Studio IDE.

3. Crear la base de datos en SQL Server:

Dentro del proyecto, encontraremos un script SQL ubicado en la carpeta "sources". Utilizaremos este script para crear nuestra base de datos en SQL Server.



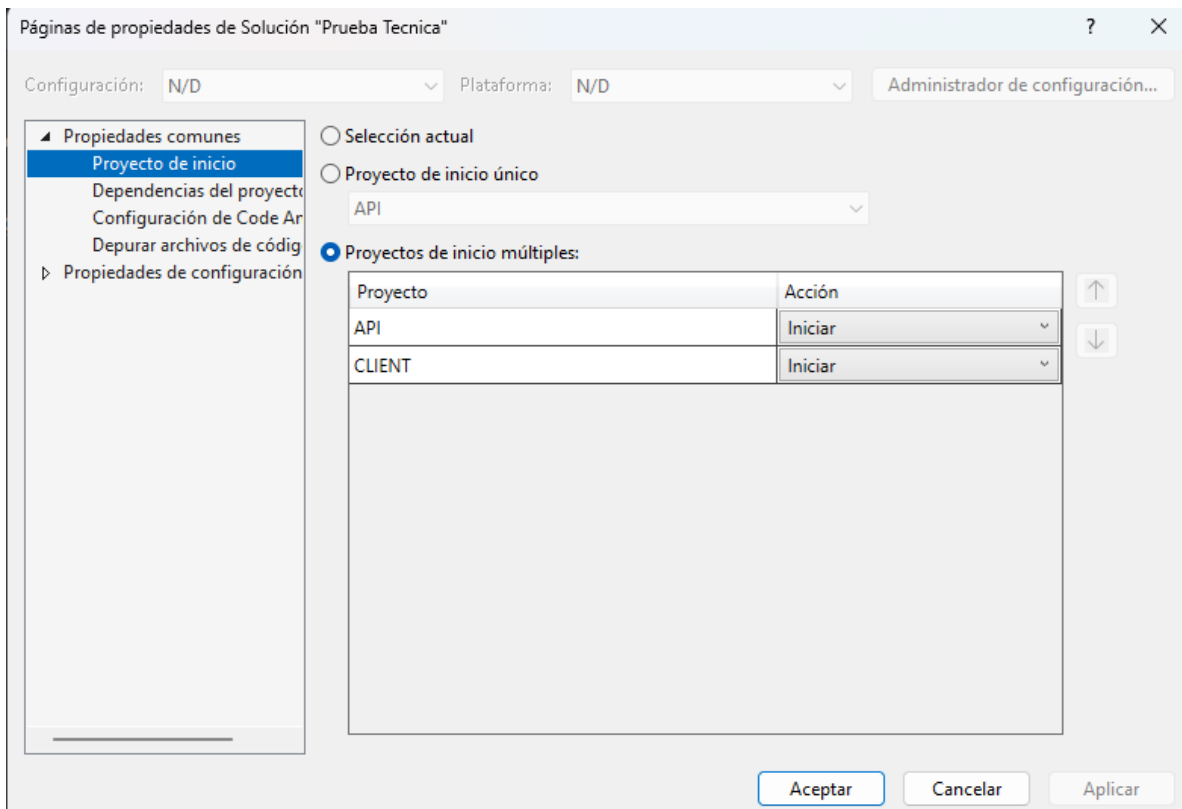
4. Editar la cadena de conexión de la base de datos:

Seguidamente, editaremos la cadena de conexión de la base de datos en el proyecto de la API en appsettings.json para asegurarnos de que se conecte correctamente a la base de datos recién creada.

```
appsettings.json
Esquema: <No se seleccionó ningún esquema>
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "ConnectionStrings": {
    "default": "Data Source=ARMANDO\\SQLEXPRESS01;Initial Catalog=PruebaTecnicaAtlantida;Integrated Security=True; Encrypt = False"
  },
  "AllowedHosts": "*"
}
```

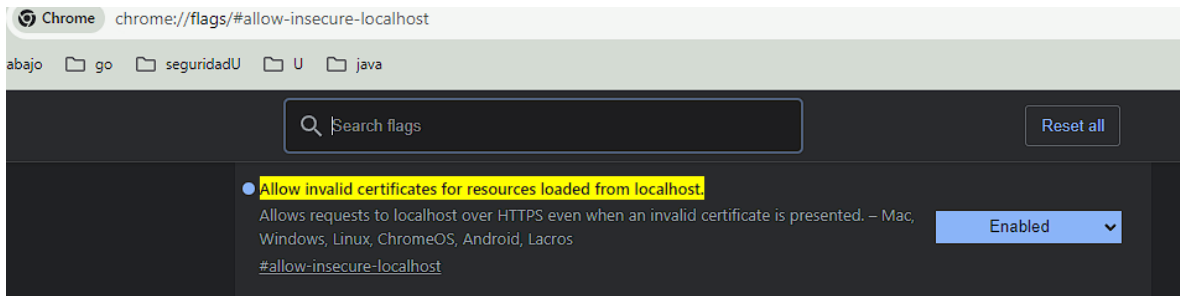
5. Configurar el inicio de la aplicación:

Configuraremos el inicio de la aplicación para que ambos proyectos (API y frontend) se inicien simultáneamente. Esto nos permitirá abrir automáticamente Swagger para probar los endpoints de la API y el frontend en el navegador.



6. Configurar Chrome para la aplicación:

Si Chrome es nuestro navegador predeterminado y necesitamos habilitar la configuración especial para localhost, abriremos Chrome y navegaremos a `chrome://flags/#allow-insecure-localhost`. Aquí habilitaremos la opción necesaria.



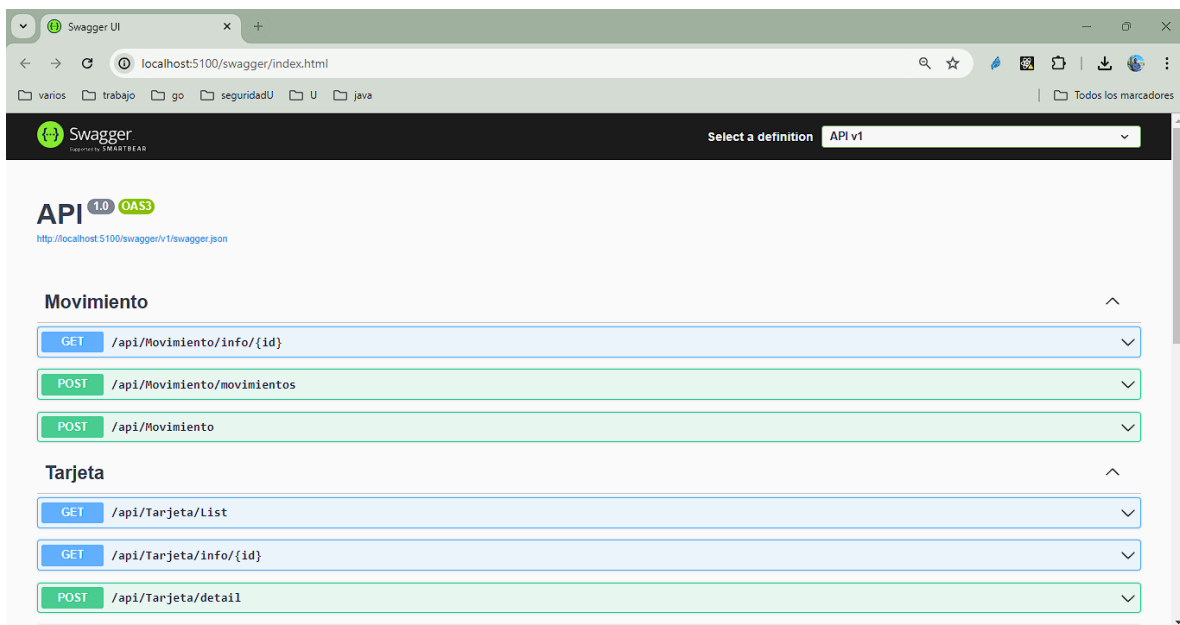
7. Reiniciar el proyecto:

Finalmente, reiniciaremos el proyecto para asegurarnos de que todas las configuraciones se apliquen correctamente y tanto Swagger como el frontend se abran automáticamente en el navegador Chrome para comenzar a probar la funcionalidad.

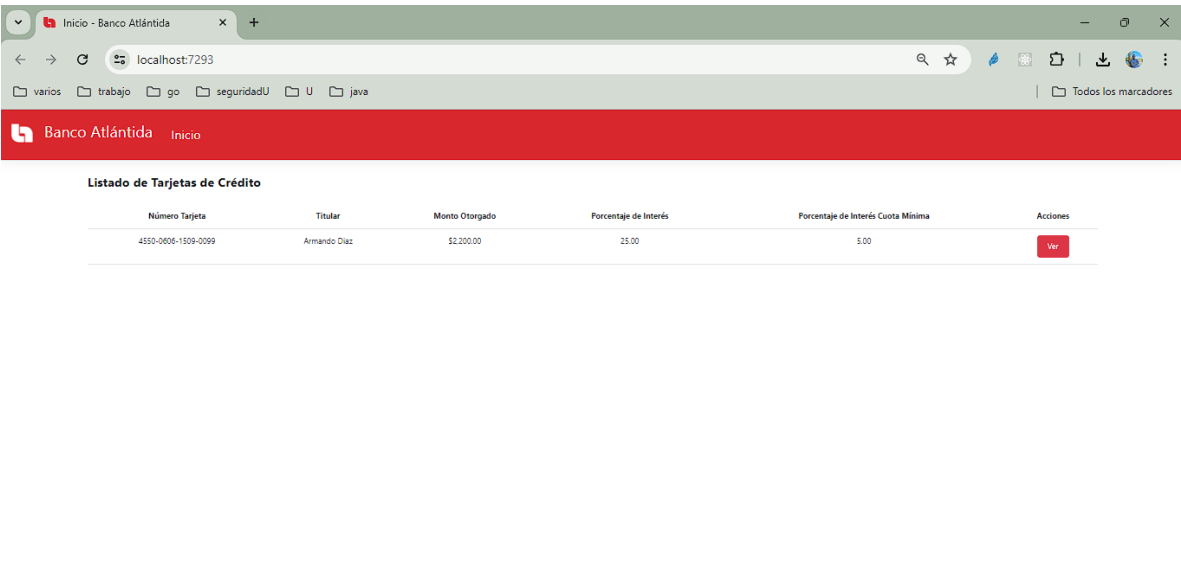
Este proceso asegura que todos los pasos necesarios están claros y bien definidos para configurar y ejecutar el proyecto desde cero de manera eficiente.

Cuando podamos levantar la aplicación se nos abrirán dos pestañas del navegador las cuales tendrán lo siguiente:

Swagger:



Aplicación web para probar funcionalidad:



Contacto

Cualquier duda sobre el funcionamiento de la aplicación contactar a:

Christian Armando Diaz Lopez.
mail: diazarmando783@gmail.com

Celular: +503 7783-9074