

Evidencias Bitácora No 10			
Ficha	Aprendiz	cedula	periodo
2675810	RODNEY ZAPATA PALACIO	72209311	11/02/2025 – 25/02/2025

- Se encarga de la lógica de negocio y la conexión con la base de datos.

2. Frontend

- Aplicación de escritorio
 - Desarrollada en Visual FoxPro (VFP)
 - Se comunica con el **backend** a través de la API REST
- Aplicación web
 - Desarrollada en **React** con HTML5
 - Accesible mediante un navegador web
 - También se comunica con el **backend** a través de la API REST

3. Base de Datos

- Motor: SQL Server
- Almacena la información de usuarios, productos, facturas, clientes, etc.

Explicación del Flujo

- La aplicación de escritorio en VFP se comunica con el **backend** mediante solicitudes a la API REST.
- La aplicación web en **React** accede al **backend** de la misma manera.
- El **backend** se encarga de procesar las solicitudes y recuperar la información de la base de datos.
- SQL Server almacena y gestiona los datos del sistema.

Detalles de cada componente:

- Cliente (Navegador web):** Los usuarios interactúan con la aplicación a través de una interfaz web, que está construida con HTML5 para la estructura, CSS para el estilo y **React** para la lógica en el lado del cliente. Las peticiones HTTP (GET, POST, PUT, DELETE) se envían a la API REST del servidor **backend**.
- Servidor de aplicaciones (Backend):** La aplicación **backend** está implementada en C# utilizando Net Core para la construcción de la API REST. El servidor de aplicaciones maneja las

facturas, clientes, técnicos, equipos, entre otros. C# utiliza Net Core para realizar operaciones CRUD en la base de datos.

Importación del proyecto en desarrollo:

Para desplegar en producción un proyecto de .NET Core con **Entity Framework** y SQL Server, necesitas instalar y configurar lo siguiente:

1. Servidor de Aplicaciones (Backend)

- .NET Core Runtime** (versión compatible con tu proyecto)
- ASP.NET Core Hosting Bundle** (si usas IIS)
- Servidor Web:** IIS, **Nginx** o Apache (según el sistema operativo)
- Configuración del Firewall** para permitir tráfico en los puertos de la API

2. Base de Datos (SQL Server)

- SQL Server** (Express, Standard o Enterprise)
- SQL Server Management Studio (SSMS)** (opcional, para administración)
- Conexión a la base de datos** configurada en **appsettings.json**
- Migraciones de Entity Framework** aplicadas (**dotnet ef database update**)

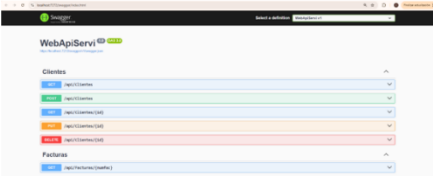
3. Cliente (Frontend)

- Aplicación Escritorio (VFP)**
 - ODBC Driver para SQL Server
 - Configuración de conexión a la API REST
- Aplicación Web (React)**
 - Servidor de hosting (puede ser IIS, **Nginx**, **Vercel**, **Netlify**, etc.)
 - Archivos compilados (**npm run build**) subidos al servidor

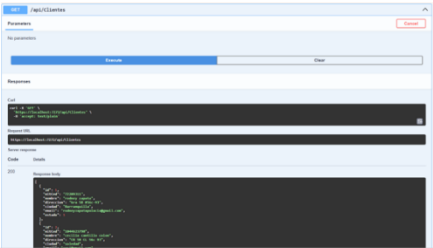
4. Seguridad y Monitoreo

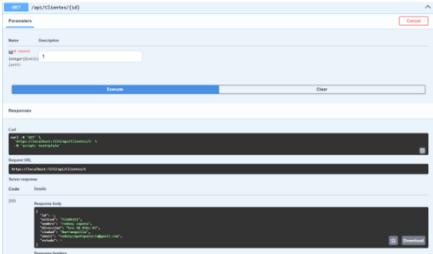
- Certificado SSL** para HTTPS
- Autenticación JWT** o **Identity Server** (si aplica)
- Herramienta de monitoreo/logs** (como **Serilog**, **Application Insights**)

3. 24/02/2025. Testeo usando la librería Swagger

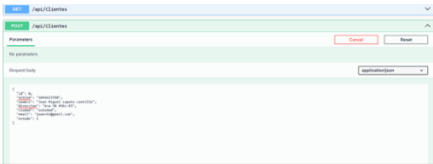


Consumo del servicio Clientes:
<http://localhost:7272/api/clientes>





1. 24/02/2025. Insertamos un cliente usando la librería y verificamos la respuesta satisfactoria de la inserción.



Evidencias Bitácora No 10			
Ficha	Aprendiz	cedula	periodo
2675810	RODNEY ZAPATA PALACIO	72209311	11/02/2025 – 25/02/2025