**Diagrama de Despliegue**

**Proyecto: WebApiServi**

**Consumo de web servi de facturación electrónica Serviparamo SAS**

**Presentado por :**

**Rodney Zapata Palacio**

**Presentado a:**

**Martha Isabel Ordoñez**

**James Calvo Dávila**

**Servicio Nacional de aprendizaje SENA**

**Centro de Comercio y Servicios (Regional Cauca)**

**Cauca - Popayán**

**Tecnólogo en Análisis y Desarrollo de Software**

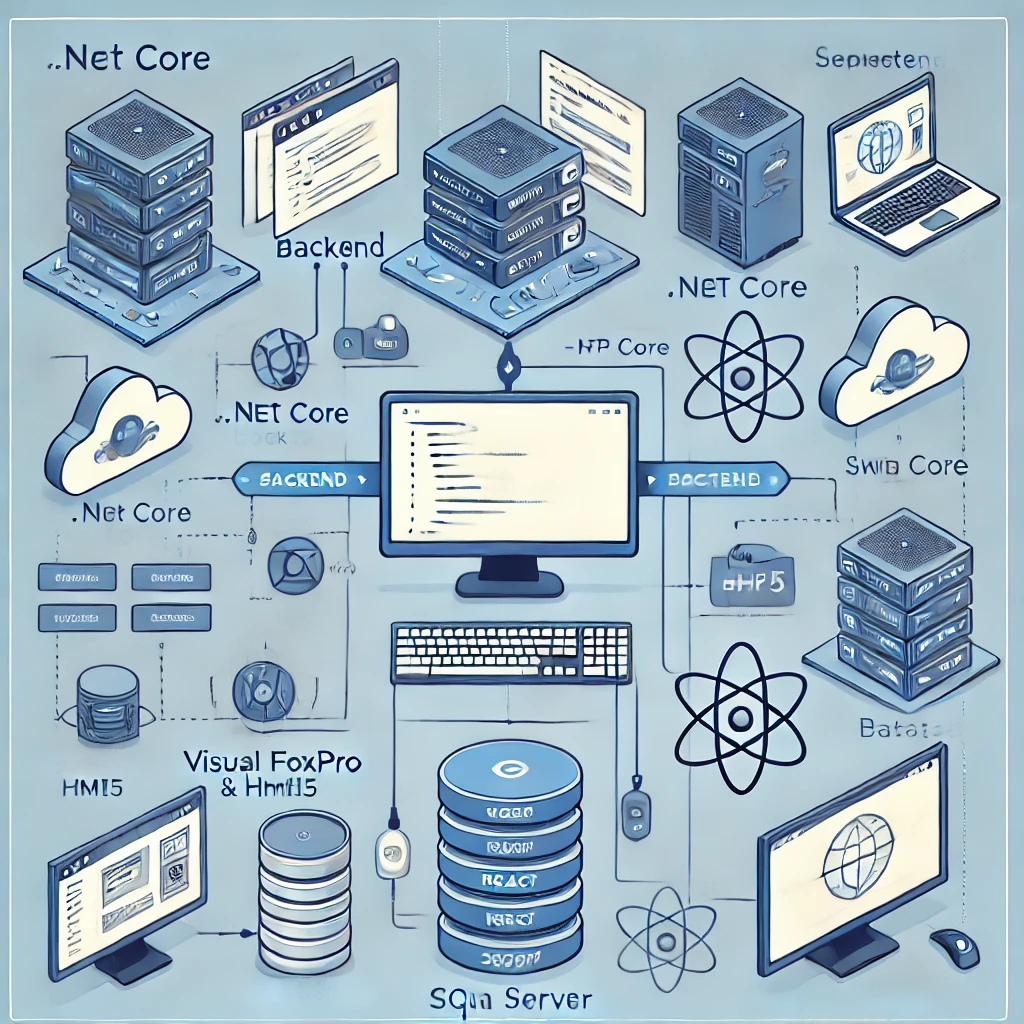
**Ficha: 2675810**

**11 de febrero del 2025**

**Propósito del desarrollo:** WebApiServi se compone de un front Pagina web y una aplicación de consola que consume un servicio web de una api rest realizada con Net Core Entity.

El cliente realiza las facturas y envia al proveedore tecnológico Facture el XML y el JSON que será validado por la DIAN.

**Arquitectura General:**

****

**Diagrama de Despliegue**

El sistema se compone de los siguientes elementos:

**1. Backend**

* Tecnología: .NET Core
* Servidor de aplicaciones que expone una API REST para la comunicación con los clientes (frontend).
* 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**

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

**Detalles de cada componente:**

1. **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.
2. **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 solicitudes del cliente, realiza la lógica del negocio (como el procesamiento de facturas) y responde con datos en formato JSON.
3. **Base de datos (SQL Server**): El servidor de base de datos SQL Server almacena toda la información de la aplicación, como 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)**

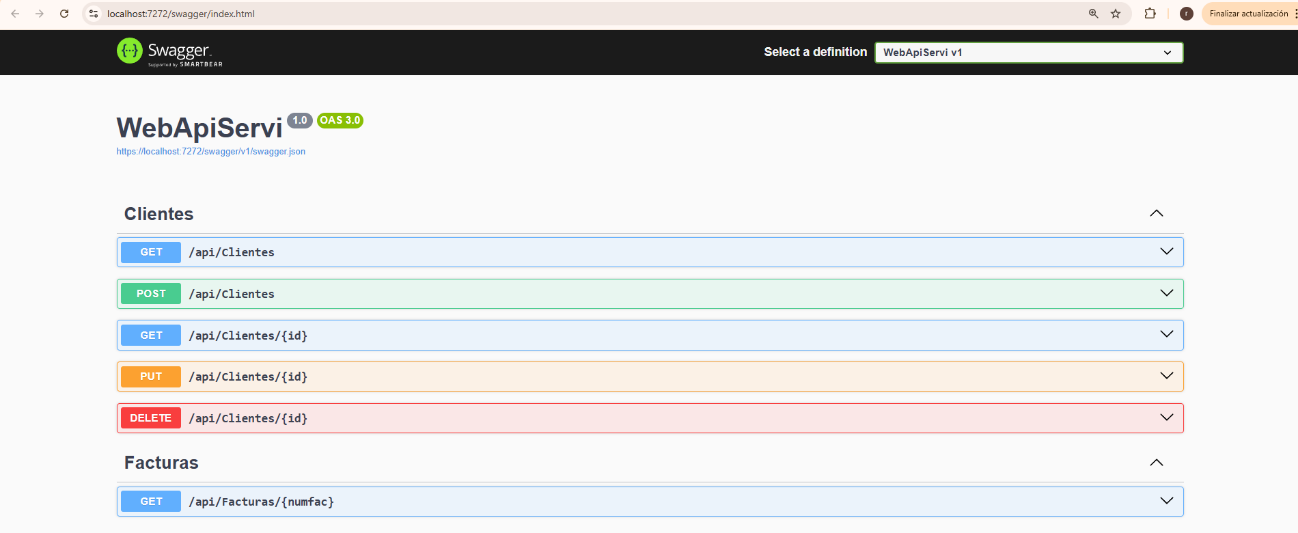
* **SQL Server** (Express, Standard o Enterprise)
* **SQL Server Management Studio (SSMS)** (opcional, para administración)
* **Conexión a la base de datos** configurada en appsettings.json
* **Migraciones de Entity Framework** aplicadas (dotnet ef database update)

**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

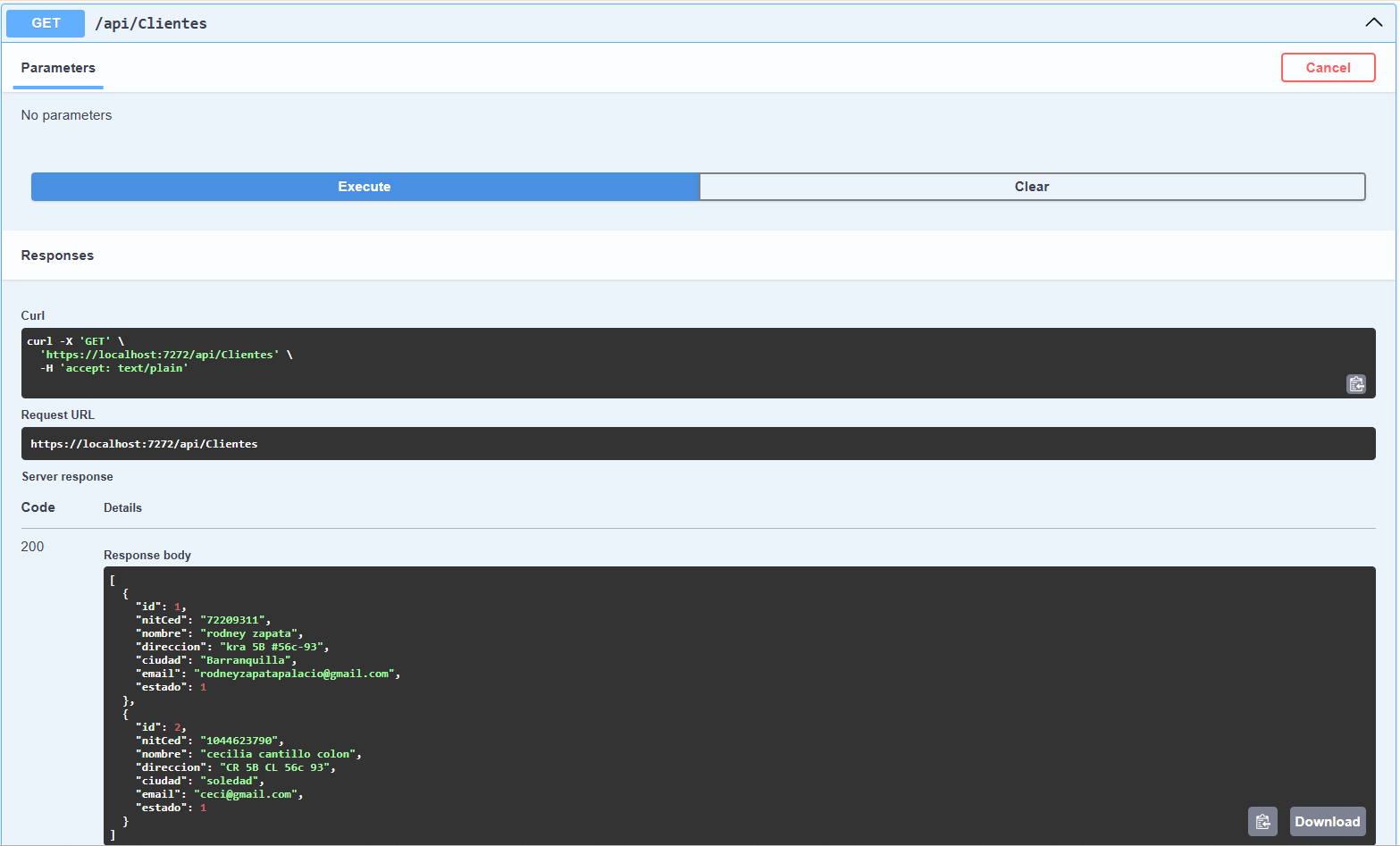
**4. Seguridad y Monitoreo**

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



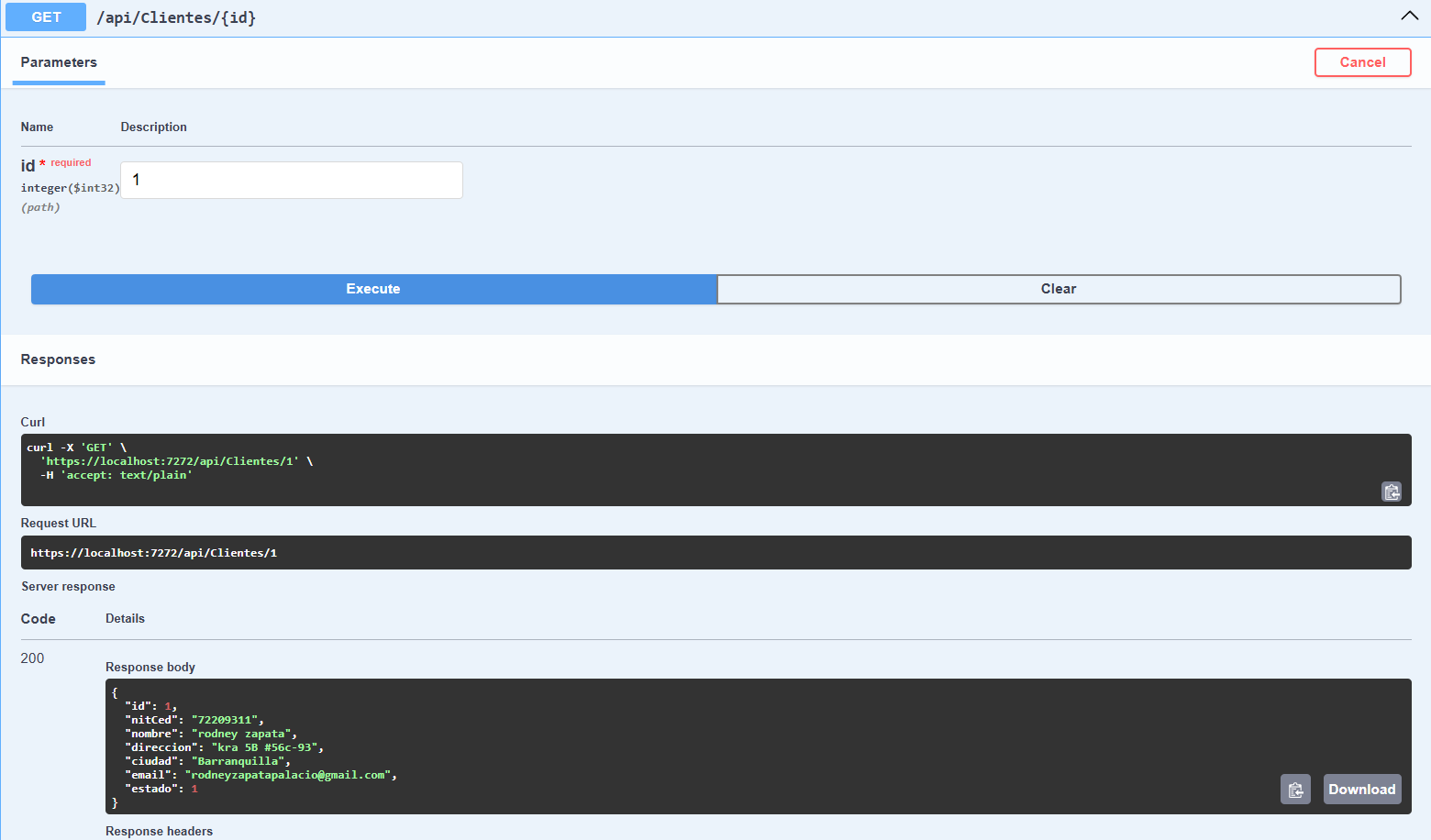
**Consumo del servicio Clientes:**

<http://localhost:7272/api/clientes>

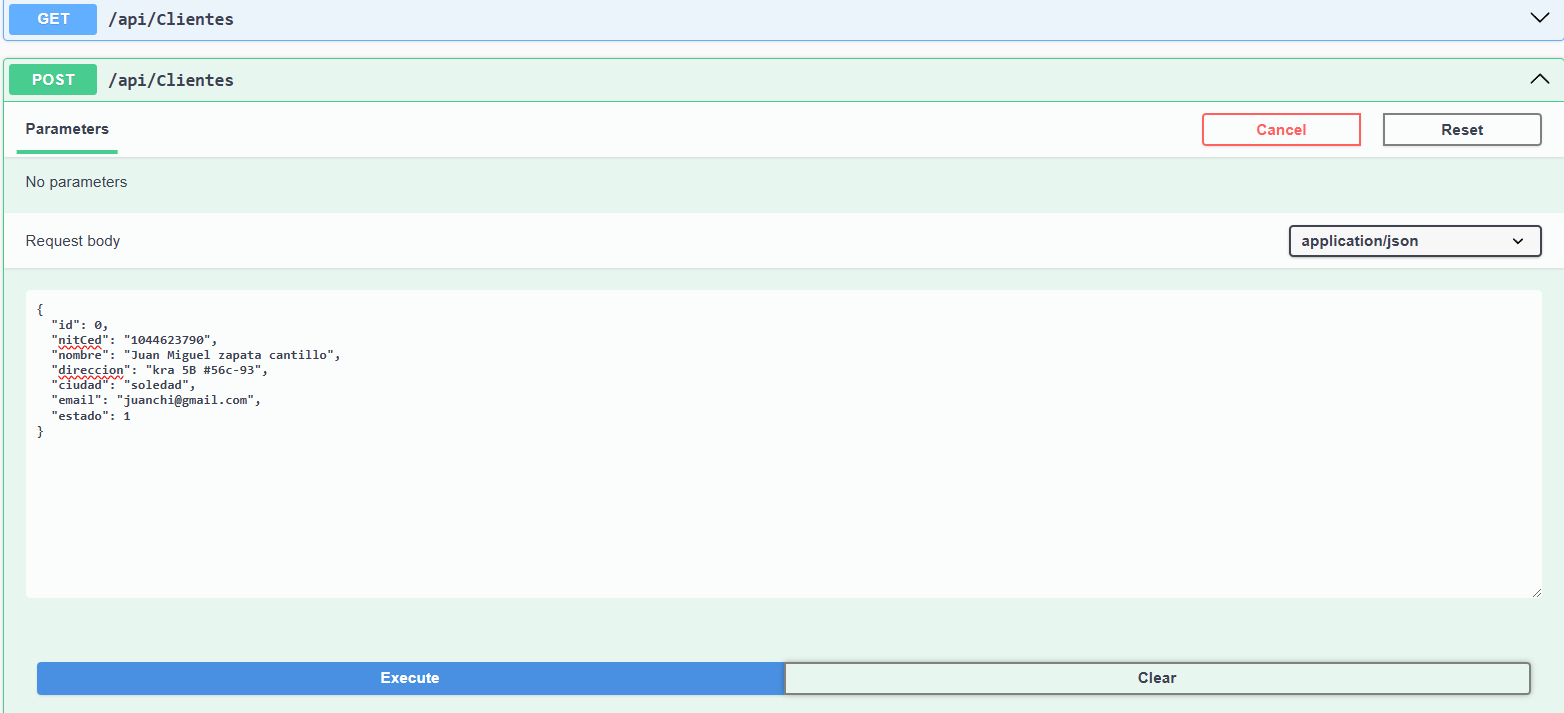


**Consumo del servicio Cliente/id:**

<http://localhost:7272/api/clientes>/1



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



Respuesta de la inserción:

