

**APLICACIÓN WEB FACTURACIÓN CON ANGULAR  
C#, .NET FRAMEWORK API Y SQL SERVER**

**PRESENTADO POR  
GIOVANNY ALEJANDRO TAPIERO CATAÑO  
A  
VIVIANA ANDREA ACHURY**

**DIGITAL WARE 2021**

## Introducción

En este documento se lista una serie de pasos acompañados de imágenes que hacen parte del proceso de desarrollo de una Aplicación Web Api Rest conectada a SQL para insertar, listar, eliminar y modificar datos (CRUD). Utilizando Angular 12 en el Front end, C# y .NET Apis en el Back end y una base de datos SQL SERVER EXPRESS.

En el modelo de datos se utilizó una base de datos y cuatro tablas con los respectivos campos para almacenar en forma ordenada y automatizada la información de facturación de una empresa, y así poder tener acceso a los datos de forma segura y poder operar con ellos.

En el Back end se utilizó el lenguaje de programación C# además de la tecnología .NET Framework la cual nos permite conectar con la base de datos por medio de una cadena de conexión, además haciendo la función de control y accediendo a toda la información por medio del protocolo HTTP y los métodos GET, PUT, POST Y DELETE, suministrando al Front end la información necesaria para mostrar al usuario final.

El Front end cuenta con una serie de controles, barra de navegación, formularios y botones Bootstrap los cuales dan una mejor experiencia en las diferentes tareas al usuario final.

Herramientas Utilizadas.

- Visual studio Code.
- Visual Studio 2019.
- SQL Server 2019 Express edition.

Lenguajes de programación.

- C#
- TypeScript.

Frameworks.

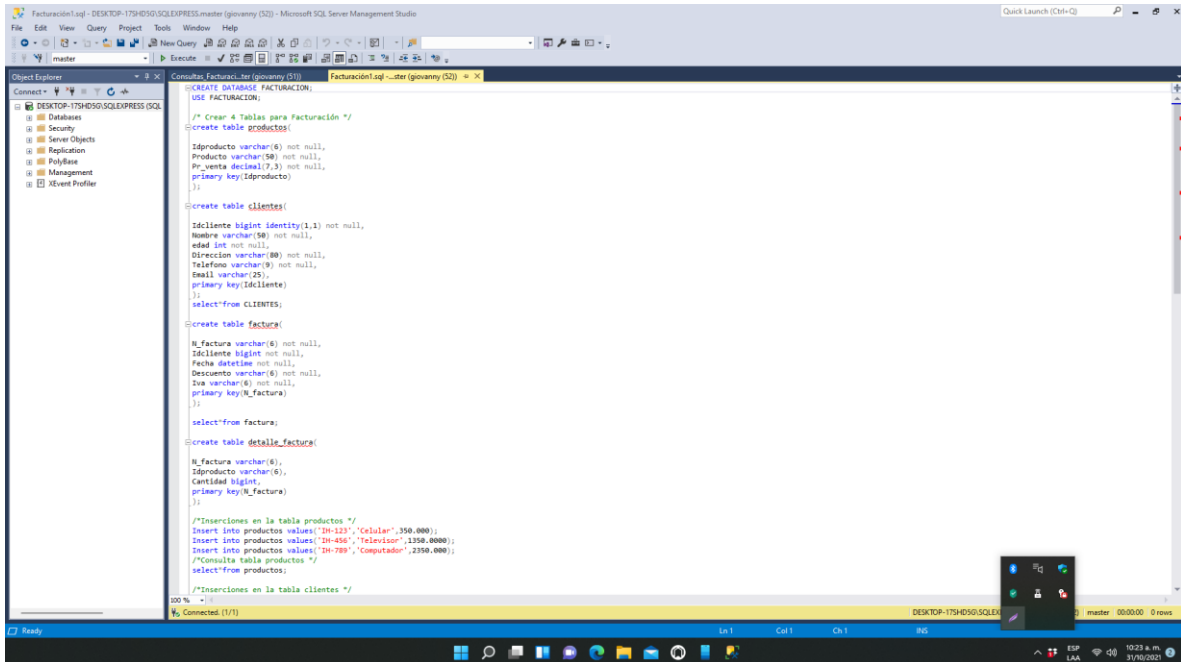
- Angular
- .Net Core versión del framework .Net para el desarrollo de aplicaciones y servicios multiplataforma.

## Marco teórico

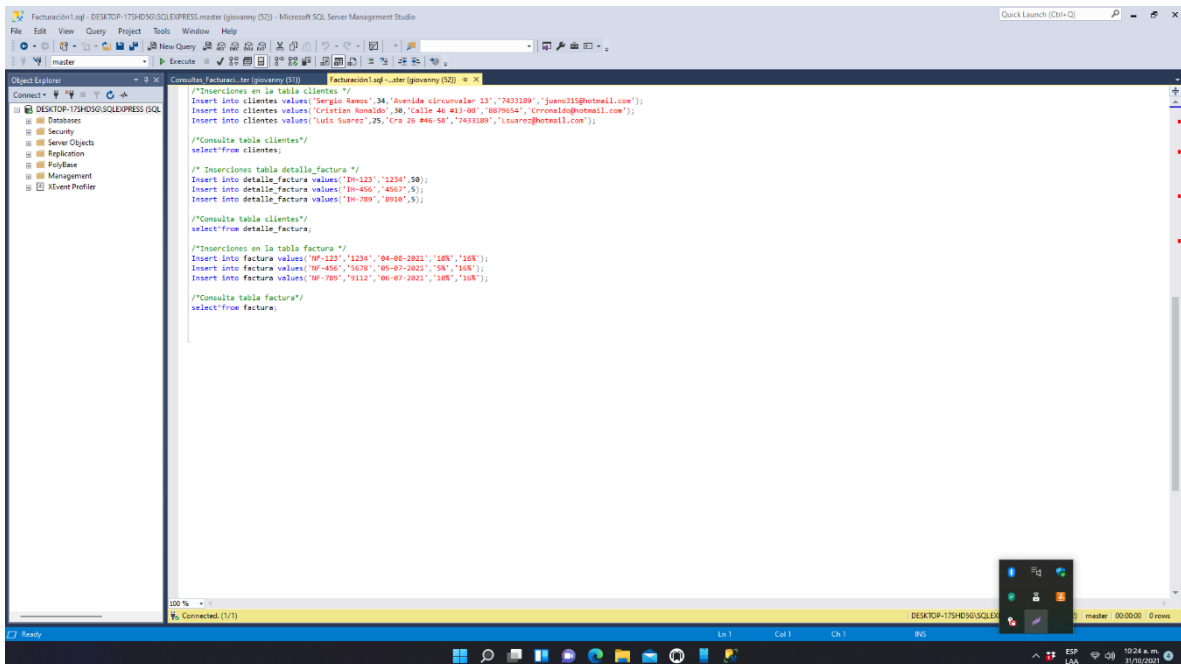
El modelo de datos anterior debe suplir las siguientes consultas las cuales deben ser entregadas en un script adicional.

Microsoft management studio.

Creación de base de datos y de tablas facturación, ingresando algunos datos.

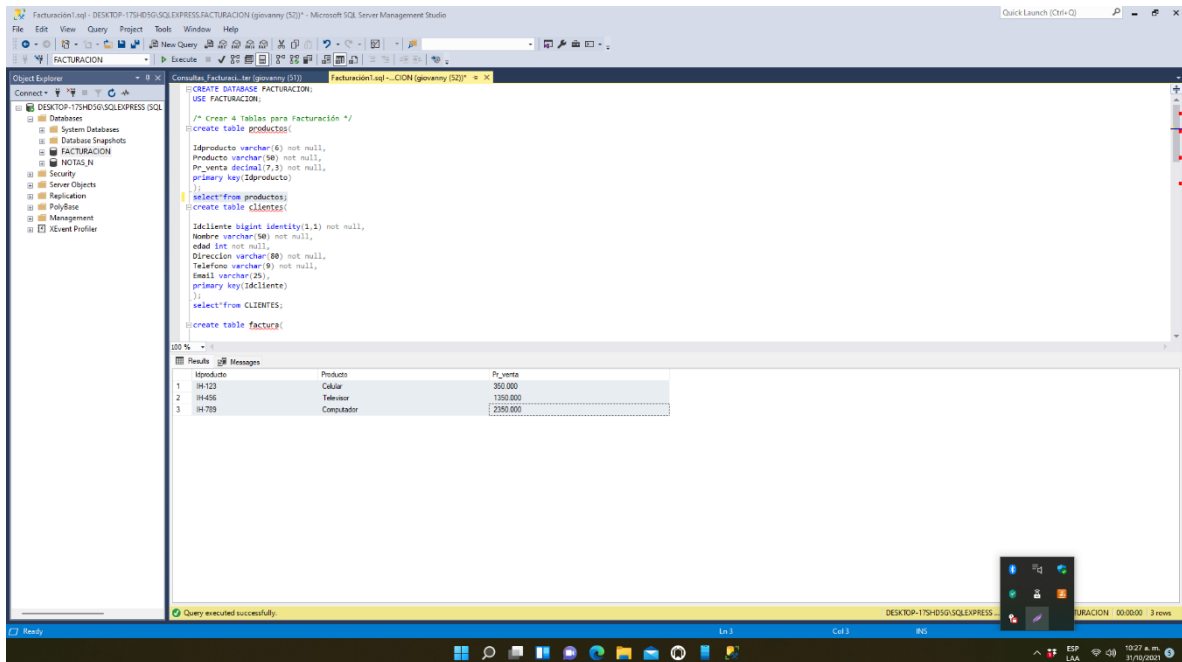


## Inserciones en las tablas.



## Consultas.

Obtener la lista de precios de todos los productos.



The screenshot shows the Microsoft SQL Server Enterprise interface. The query editor displays the following SQL code:

```
USE FACTURACION;

/* Crear 4 tablas para facturación */
create table productos(
    idproducto varchar(6) not null,
    producto varchar(50) not null,
    pr_venta decimal(7,3) not null,
    primary key(idproducto)
);
select * from productos;

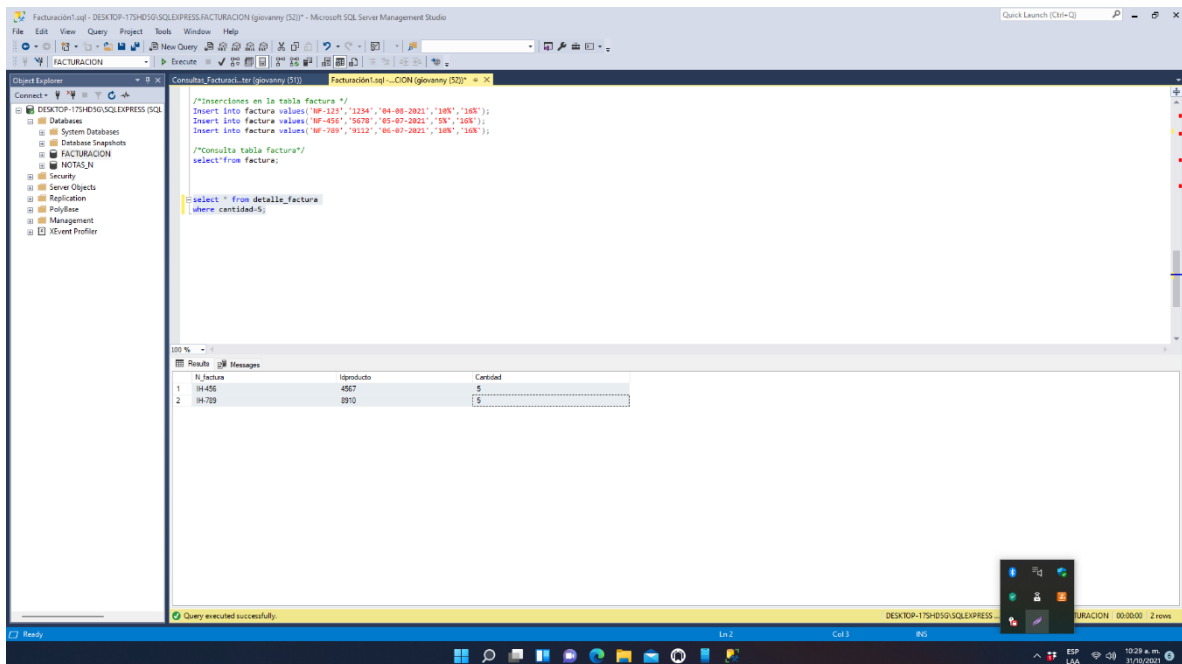
create table clientes(
    idcliente bigint identity(1,1) not null,
    nombre varchar(50) not null,
    edad int not null,
    direccion varchar(80) not null,
    telefono varchar(9) not null,
    email varchar(25),
    primary key(idcliente)
);
select * from clientes;

create table factura(
    n_factura int identity(1,1) not null,
    idproducto varchar(6) not null,
    cantidad int not null,
    fecha datetime not null,
    primary key(n_factura, idproducto),
    foreign key(idproducto) references productos(idproducto)
);
```

The Results pane shows the output of the query, displaying a table with 3 rows and 3 columns: idproducto, Producto, and Pr\_venta.

idproducto	Producto	Pr_venta
H-123	Celular	350.000
H-456	Tablet	1200.000
H-789	Computador	2350.000

Obtener la lista de productos cuya existencia en el inventario haya llegado al mínimo permitido (5 unidades).



The screenshot shows the Microsoft SQL Server Enterprise interface. The query editor displays the following SQL code:

```
/*Inserciones en la tabla factura*/
Insert into factura values('H-123','1234','04-06-2021','100','100');
Insert into factura values('H-456','5678','05-07-2021','50','50');
Insert into factura values('H-789','9112','06-07-2021','100','100');

/*Consulta tabla factura*/
select * from factura;

select * from detalle_factura
where cantidad < 5;
```

The Results pane shows the output of the query, displaying a table with 2 rows and 3 columns: n\_factura, idproducto, and Cantidad.

n_factura	idproducto	Cantidad
H-456	4567	5
H-789	8910	5

Obtener una lista de clientes no mayores de 35 años que hayan realizado compras entre el 1 de febrero de 2000 y el 25 de mayo de 2000.

Query executed successfully.

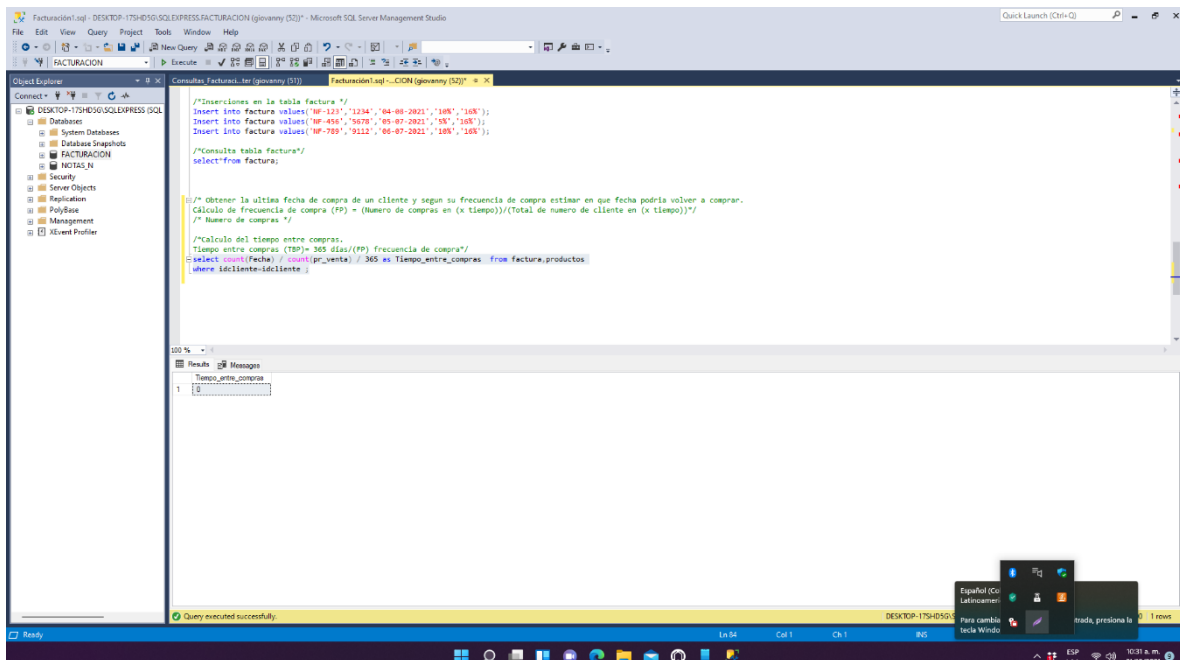
Idcliente	Nombre	edad	Direccion	Telefono	Email	N_factura	Idcliente	Fecha	Descuento	Iva
1	Sergio Ramos	34	Avenida circunvalar 13	7433189	juano315@hotmail.com	NF-123	1234	2021-04-08 00:00:00.000	10%	16%
2	Cristian Ronaldo	30	Calle 45 813-08	8879654	Cronaldo@hotmail.com	NF-123	1234	2021-04-08 00:00:00.000	10%	16%
3	Luis Suarez	25	Cra 26 846-50	7433189	Luauere@hotmail.com	NF-123	1234	2021-04-08 00:00:00.000	10%	16%
4	Sergio Ramos	34	Avenida circunvalar 13	7433189	juano315@hotmail.com	NF-456	5678	2021-05-07 00:00:00.000	5%	16%
5	Cristian Ronaldo	30	Calle 45 813-08	8879654	Cronaldo@hotmail.com	NF-456	5678	2021-05-07 00:00:00.000	5%	16%
6	Luis Suarez	25	Cra 26 846-50	7433189	Luauere@hotmail.com	NF-456	5678	2021-05-07 00:00:00.000	5%	16%
7	Sergio Ramos	34	Avenida circunvalar 13	7433189	juano315@hotmail.com	NF-789	9112	2021-06-07 00:00:00.000	10%	16%
8	Cristian Ronaldo	30	Calle 45 813-08	8879654	Cronaldo@hotmail.com	NF-789	9112	2021-06-07 00:00:00.000	10%	16%
9	Luis Suarez	25	Cra 26 846-50	7433189	Luauere@hotmail.com	NF-789	9112	2021-06-07 00:00:00.000	10%	16%

Obtener el valor total vendido por cada producto en el año 2000.

Query executed successfully.

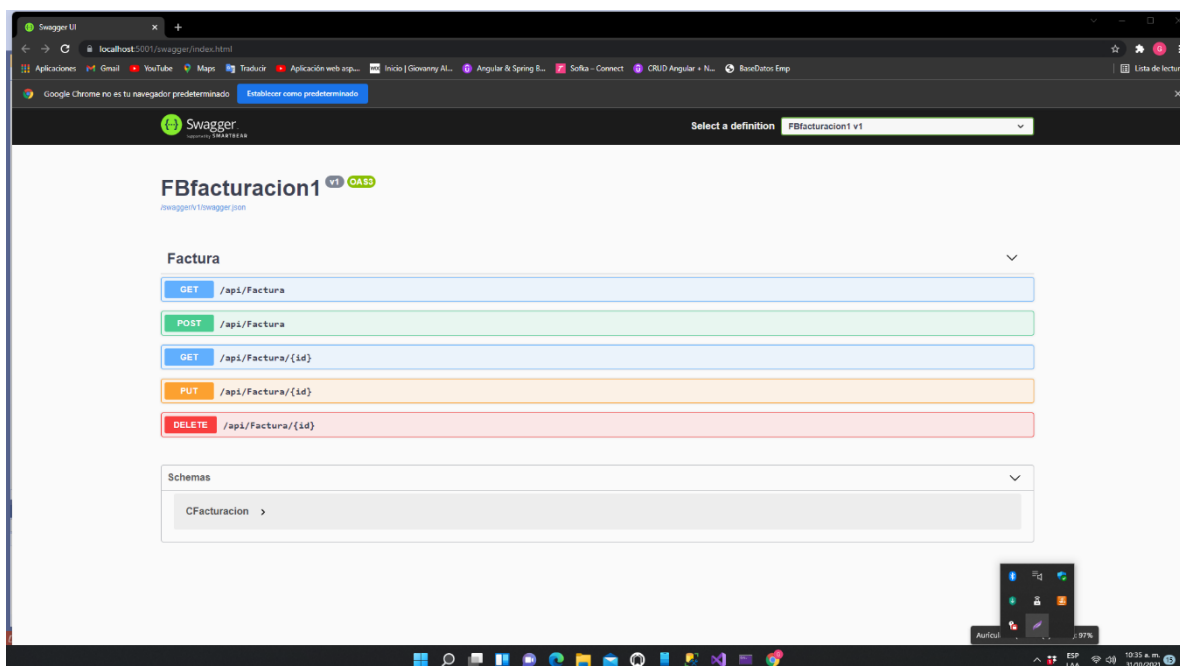
Total_venta_producto
12150.000

Obtener la última fecha de compra de un cliente y según su frecuencia de compra estimar en qué fecha podría volver a comprar.

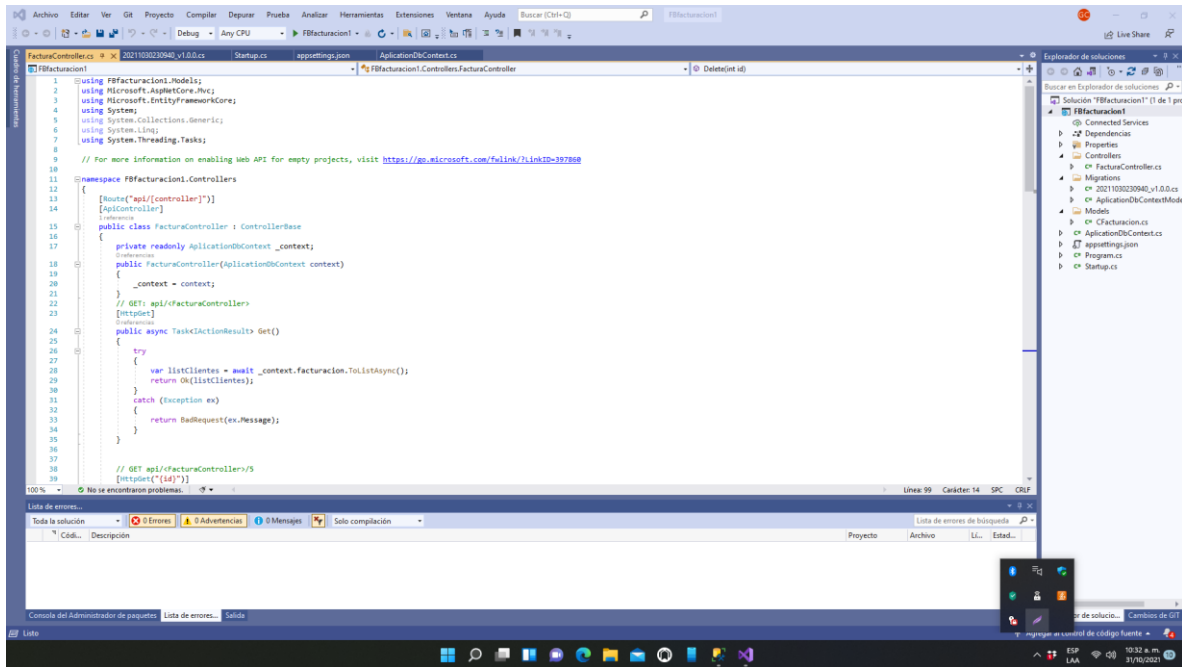


El Backend se debe construir haciendo uso del lenguaje c# con .NET exponiendo servicios REST con web api. Para la conexión a la base de datos se puede usar ADO.NET puro o un ORM como entity FrameWork y se apreciara mucho que se realice separación por capas.

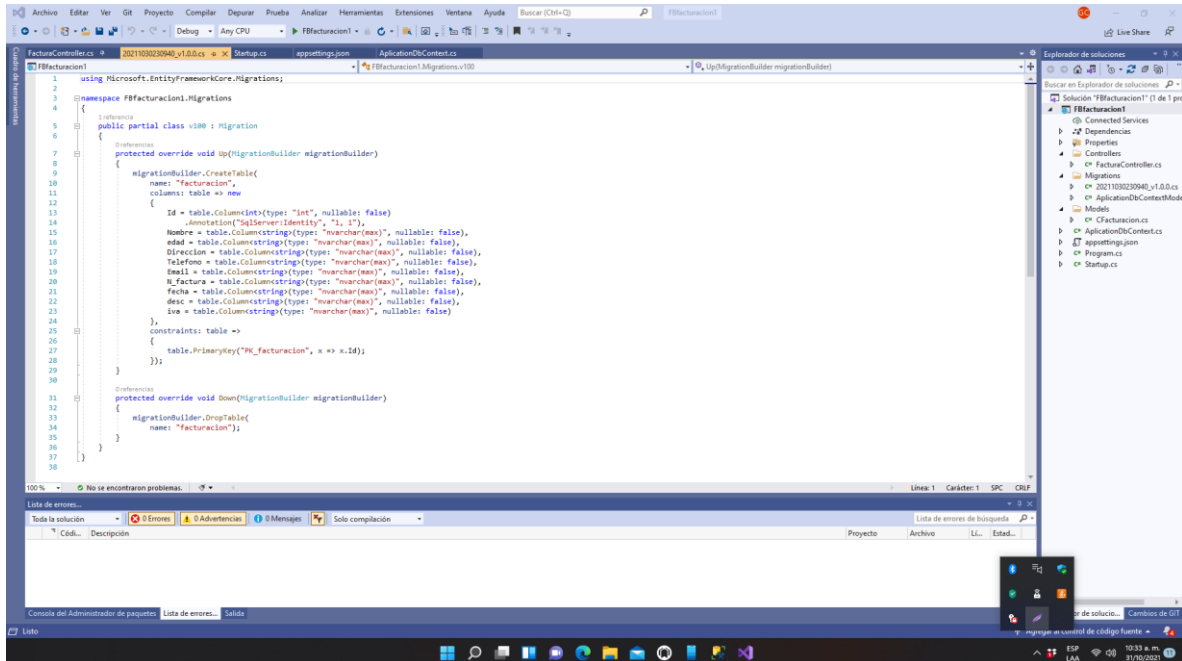
Arquitectura utilizada MVC (Modelo vista controlador).  
Api Rest. Visual Studio.



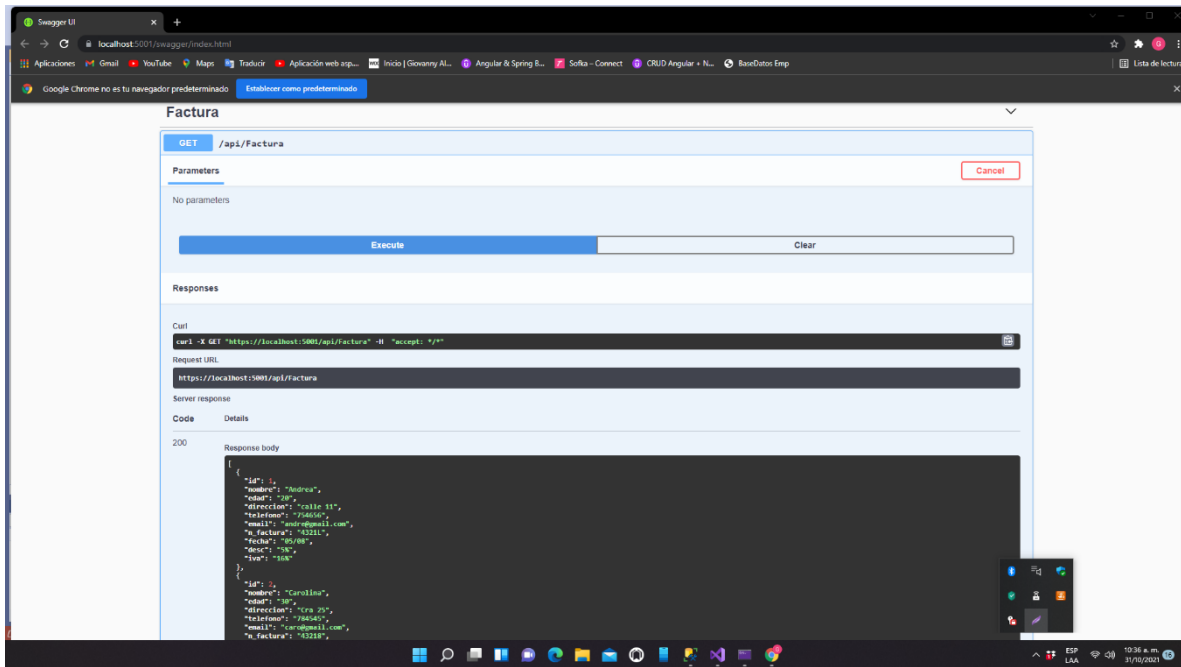
## Controlador .Net



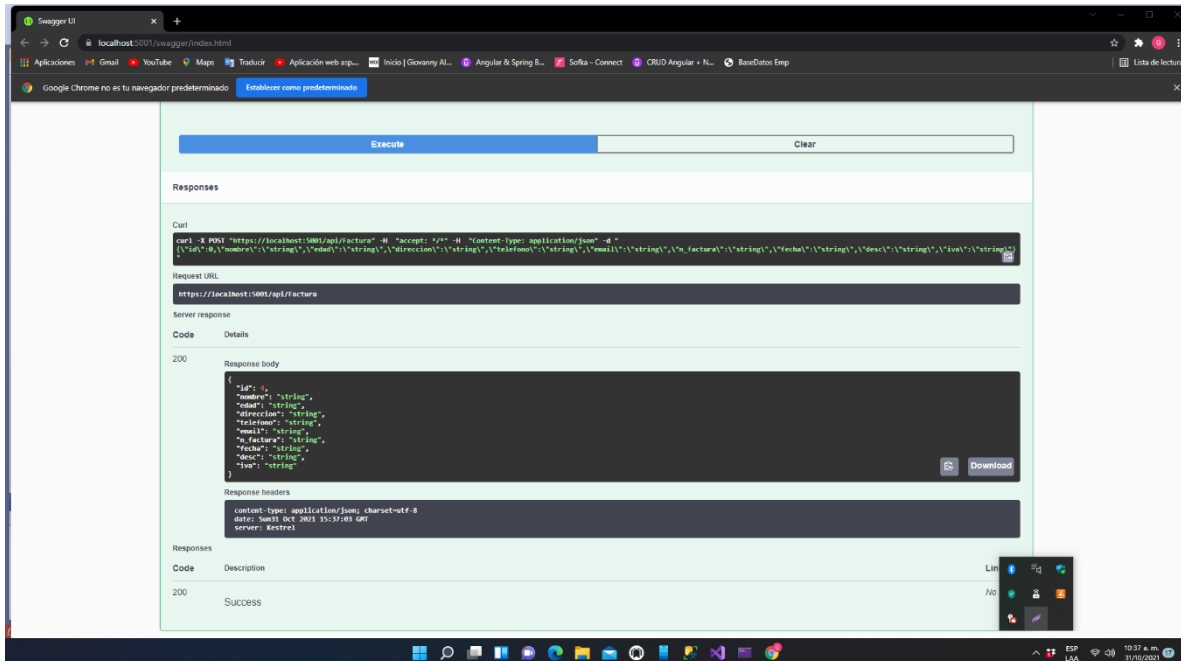
## Migracion a base de datos.



## Método Get.

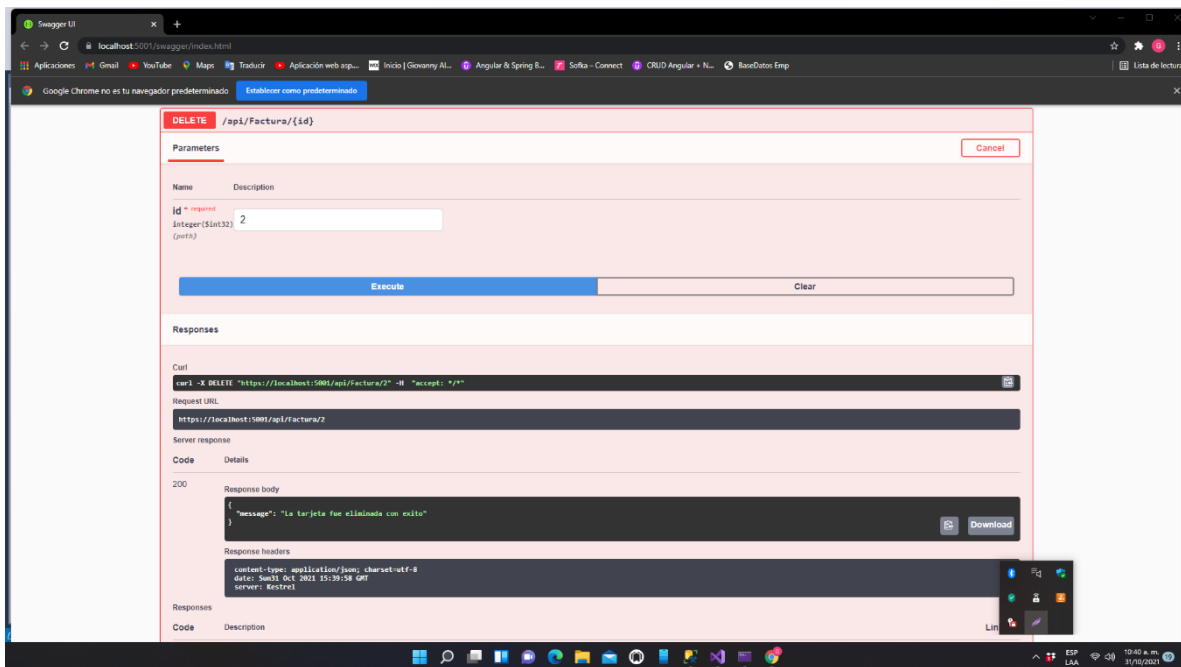


## Método Post.

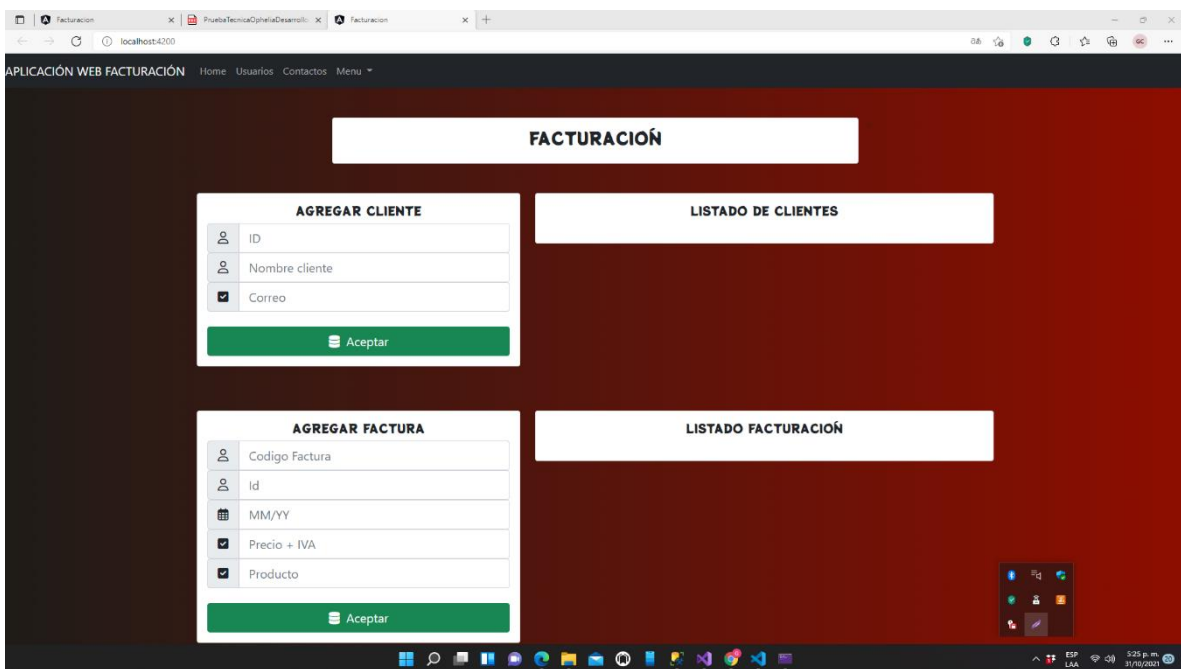




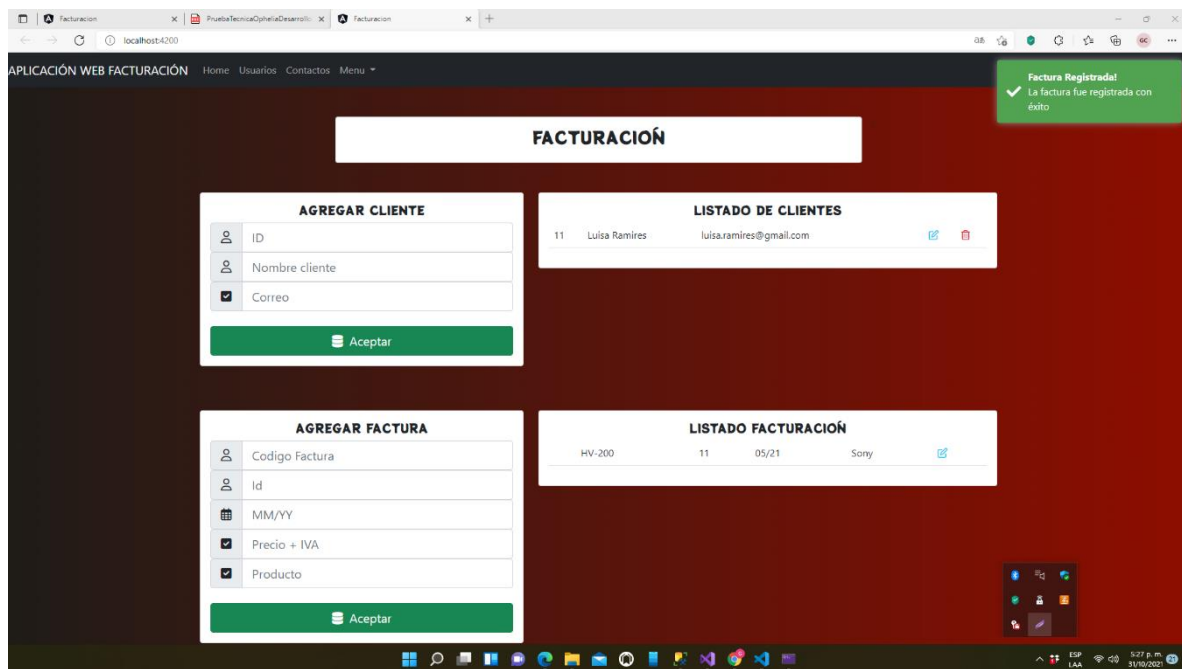
## Método Delete.



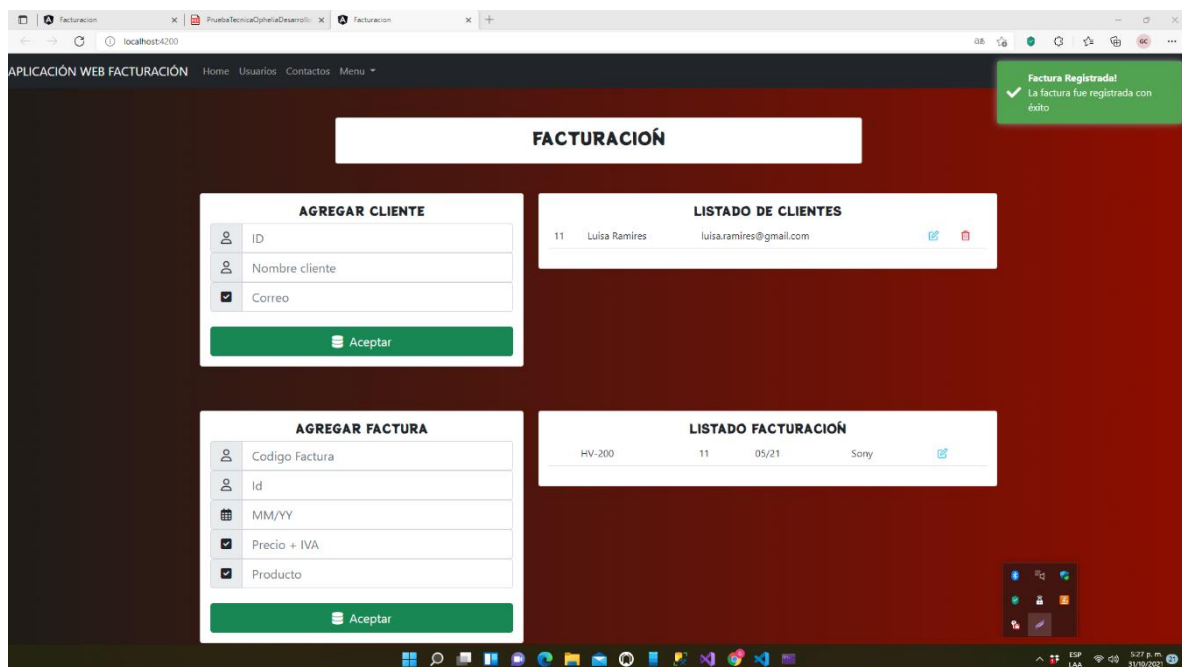
Se debe construir una aplicación web de facturación utilizando controles de la librería de devextreme (Grillas. Formularios, Botones) integrado con Angular mínimo versión 5. Donde se debe realizar el llamado de los servicios de la API para el CRUD de facturación.



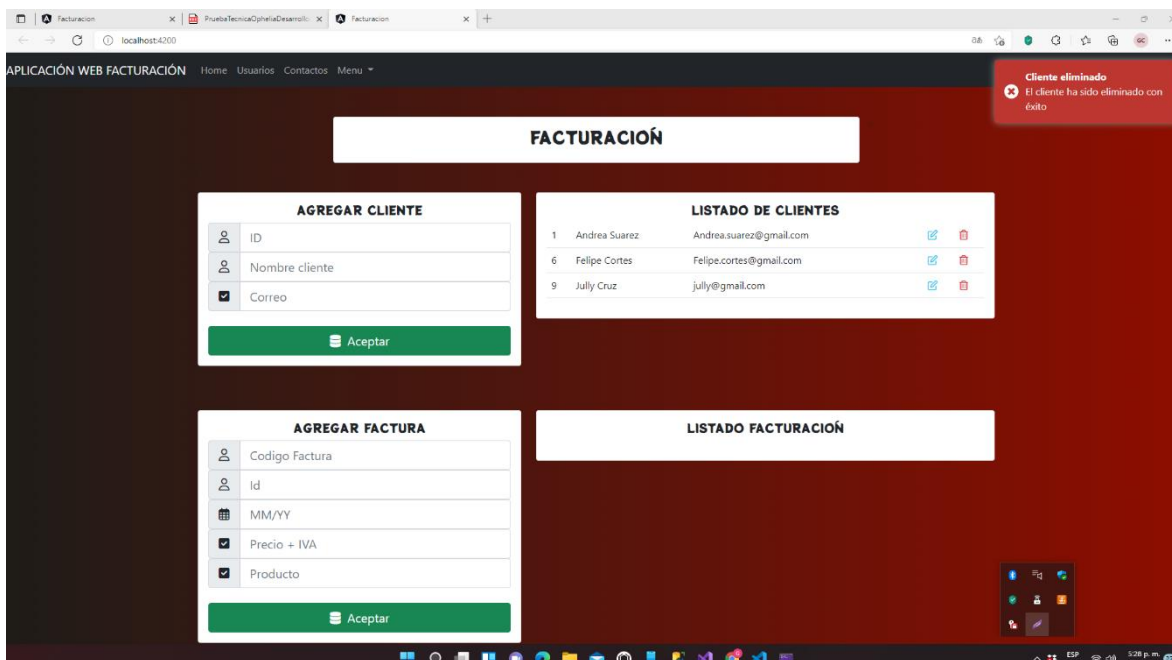
## Inserción Cliente.



## Inserción factura.

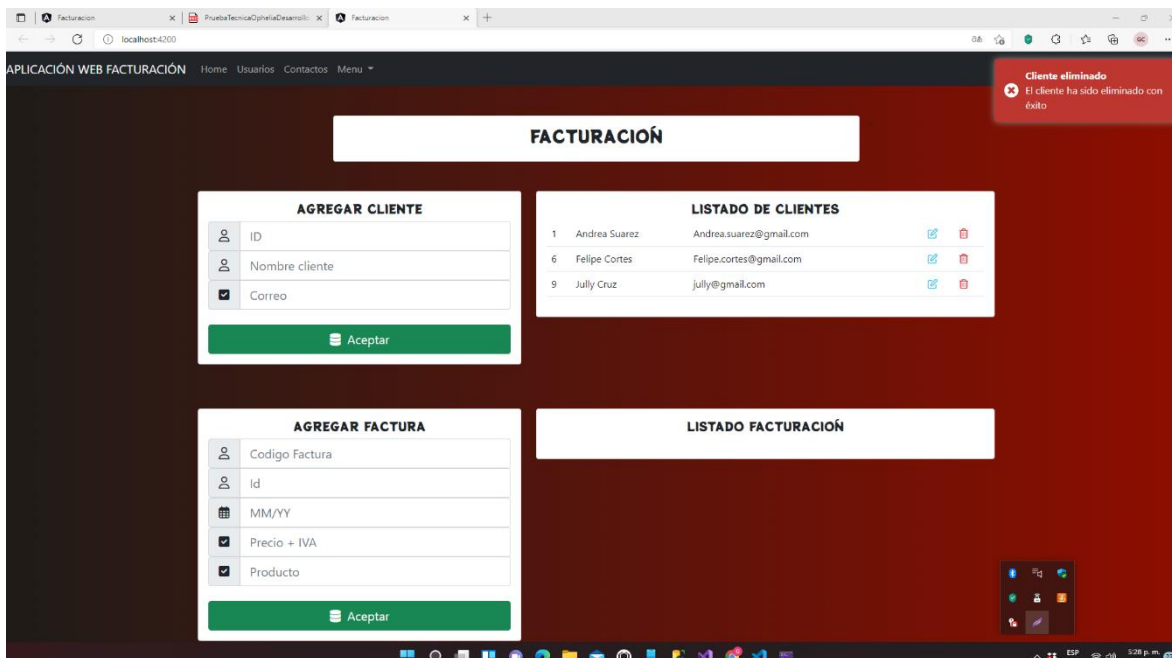


## Eliminación cliente.



## Consulta Datos de la base de datos.

Se hace la consulta en la tabla Listado Clientes y listado facturación.



## **Conclusiones**

La Aplicación web Facturación de acuerdo con los requerimientos del cliente cumple con las expectativas. Con un código limpio, bien estructurado y orientado a objetos el cual nos permite realizar los diferentes métodos del CRUD con conexión a base de datos SQL server, con un Back end perfectamente funcional y una aplicación en el Front dinámica y funcional.