

UNIVERSIDAD MARIANO GALVEZ DE GUATEMALA
INGENIERIA EN SISTEMAS
DESARROLLO WEB

RESUMEN EXAMEN FINAL

ROBERTO CARLOS PERNILLO DONIS

7690-18-9965

Enunciado:

Utilizando .NET Core 8 desarrolle una API que contenga los endpoints. Al finalizar deberá escribir un resumen de lo realizado para solventar cada ítem:

1. Creación de clientes con los siguientes campos:
 1. ID
 2. Nombres
 3. Apellidos
 4. Genero
 5. Fecha de nacimiento
 6. Estado
2. Información general de cliente con los siguientes campos:
 1. Tipo de información (Dirección, contactabilidad, información financiera)
 2. Fecha de creación
 3. Fecha de actualización
 4. Usuario creador
 5. Estado de la información.
3. Deberá mostrar información en endpoint llamado ListadoGeneral ordenado por fecha de creación y apellidos.
4. Debe realizar ER y migrarlo a su proyecto con **Scaffolding (Reverse Engineering)**. Base de datos a utilizar PostgreSQL.

Paso 1:

Correcta configuración de los sistemas:

La correcta configuración de los sistemas utilizados es la clave principal para poder hacer ejecutar el proyecto que realizamos, primero asegurarnos de tener la versión correcta de .net core 8 y la versión compatible de postgresql con esto, por lo que fue necesario hacer la actualización de drivers para asegurarnos de que tuviéramos los correctos, una vez completado esto realizamos la creación del proyecto y podíamos iniciar a trabajar.

```
C:\Users\roberto.pernillo>dotnet new webapi -n ClientesAPI
The template "ASP.NET Core Web API" was created successfully.

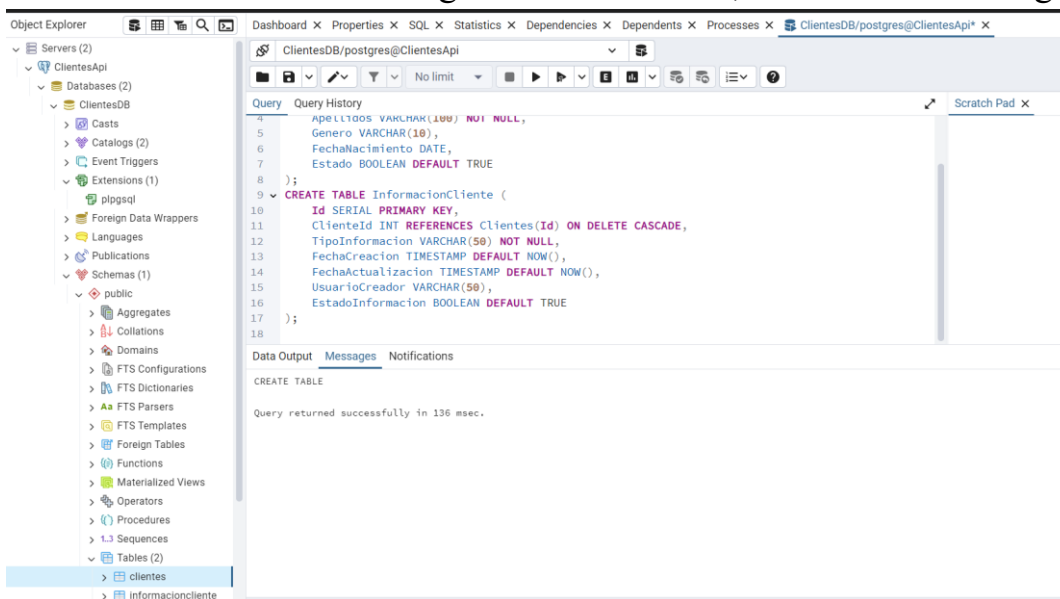
Processing post-creation actions...
Restoring C:\Users\roberto.pernillo\ClientesAPI\ClientesAPI.csproj:
  Determining projects to restore...
  Restored C:\Users\roberto.pernillo\ClientesAPI\ClientesAPI.csproj (in 2.06 sec).
Restore succeeded.
```

```
C:\Users\roberto.pernillo\ClientesAPI>dotnet add package Npgsql.EntityFrameworkCore.PostgreSQL --version 8.0.0
Determining projects to restore...
```

Paso 2:

Configuración de base de datos:

La correcta configuración de la base de datos me permitió luego hacer el proceso de reverse engineering sobre el proyecto principal así que se creo de la siguiente forma, desde PgAdmin4.



Paso 3:

Reverse engineering:

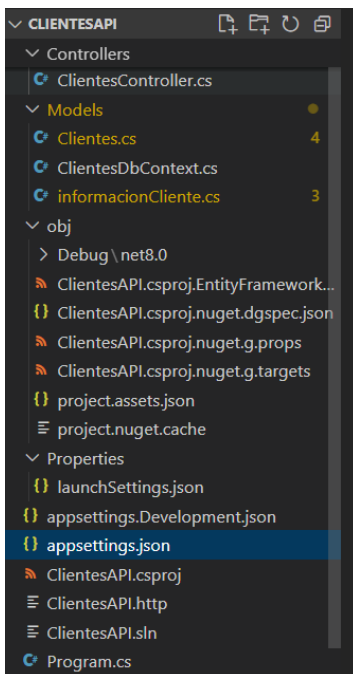
Para el reverse engineering según lo solicitado en el enunciado se utilizo scaffolding de la siguiente manera sobre la terminal del proyecto:

```
C:\Users\roberto.pernillo\ClientesAPI>dotnet ef dbcontext scaffold "Host=localhost;Database=ClientesDB;Username=postgres;Password=12345" Npgsql.EntityFrameworkCore.PostgreSQL -o Models -t Clientes -t InformacionCliente Build started... Build succeeded.
```

Paso 4:

Modificacion de entidades:

Luego de el scaffolding pude modificar las entidades creadas dentro del proyecto y así manipular las clases que me interesaban para hacer la lógica de el api con modelo controlador. (Para mas información y por tiempo el código completo esta en el proyecto y no en capturas de pantalla).



Paso 5:

Ejecución:

Finalmente podemos ejecutar el api y confirmar su funcionamiento.

```
C:\Users\roberto.fernando\ClientesAPI>dotnet run
Building...
C:\Users\roberto.fernando\ClientesAPI\Models\informacionCliente.cs(9,23): warning CS8618: Non-nullable property 'TipoInformacion' must contain a non-null value when exiting constructor. Consider declaring the property as nullable. [C:\Users\roberto.fernando\ClientesAPI\ClientesAPI.csproj]
C:\Users\roberto.fernando\ClientesAPI\Models\informacionCliente.cs(12,23): warning CS8618: Non-nullable property 'UsuarioCreador' must contain a non-null value when exiting constructor. Consider declaring the property as nullable. [C:\Users\roberto.fernando\ClientesAPI\ClientesAPI.csproj]
C:\Users\roberto.fernando\ClientesAPI\Models\informacionCliente.cs(15,24): warning CS8618: Non-nullable property 'Cliente' must contain a non-null value when exiting constructor. Consider declaring the property as nullable. [C:\Users\roberto.fernando\ClientesAPI\ClientesAPI.csproj]
C:\Users\roberto.fernando\ClientesAPI\Models\Clientes.cs(9,23): warning CS8618: Non-nullable property 'Nombres' must contain a non-null value when exiting constructor. Consider declaring the property as nullable. [C:\Users\roberto.fernando\ClientesAPI\ClientesAPI.csproj]
C:\Users\roberto.fernando\ClientesAPI\Models\Clientes.cs(10,23): warning CS8618: Non-nullable property 'Apellidos' must contain a non-null value when exiting constructor. Consider declaring the property as nullable. [C:\Users\roberto.fernando\ClientesAPI\ClientesAPI.csproj]
C:\Users\roberto.fernando\ClientesAPI\Models\Clientes.cs(11,23): warning CS8618: Non-nullable property 'Genero' must contain a non-null value when exiting constructor. Consider declaring the property as nullable. [C:\Users\roberto.fernando\ClientesAPI\ClientesAPI.csproj]
C:\Users\roberto.fernando\ClientesAPI\Models\Clientes.cs(15,48): warning CS8618: Non-nullable property 'InformacionClientes' must contain a non-null value when exiting constructor. Consider declaring the property as nullable. [C:\Users\roberto.fernando\ClientesAPI\ClientesAPI.csproj]
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5290
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\roberto.fernando\ClientesAPI
```

Finalmente ingresando al localhost podemos ver la estructura de nuestra api

