

Instituto Tecnológico de Costa Rica

Escuela de ingeniería en computadores

Profesor: Marco Rivera Meneses

Curso: Bases de datos

Grupo 1

Estudiantes:

Emanuel Marín Gutiérrez – 2019067500

Jose Andrés Rodríguez Rojas – 2019279722

Oscar Soto Varela – 2020092336

Sebastián Chen Cerdas – 2021571438

Proyecto I - TECAir

Manual de Instalación

II Semestre, 2023

Tabla de contenidos

Instalación de IIS	3
Vista administrador	5
Vista cliente	5
App móvil	6
API-Database.....	8
Método #1. Enfoque Código-Primero	8
Método #2. Enfoque Base de Datos Primero	9

Para instalar las distintas partes de TECAir primero es necesario tener instalado en el equipo los siguientes programas y ambientes de ejecución:

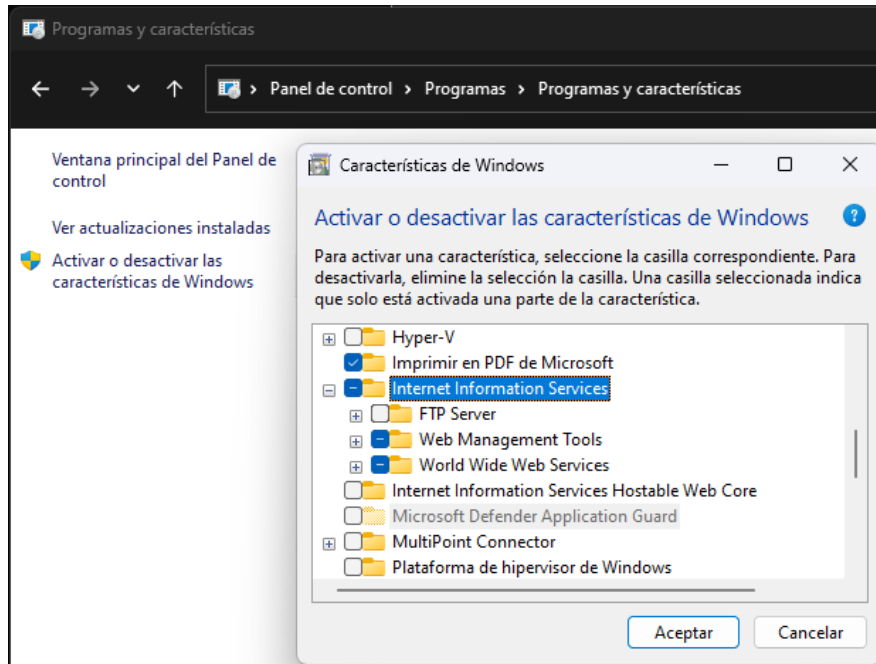
- .NET Runtime 6.0.22
- ASP.NET Core Runtime 6.0.22 (Hosting Bundle)
- PostgreSQL 16
- Node JS

Además de algunos requerimientos de hardware necesarios para el correcto funcionamiento de todos los programas:

- Procesador de arquitectura x64 de 4 núcleos o más
- Mínimo 4 GB de RAM
- Mínimo 30 GB de disco duro

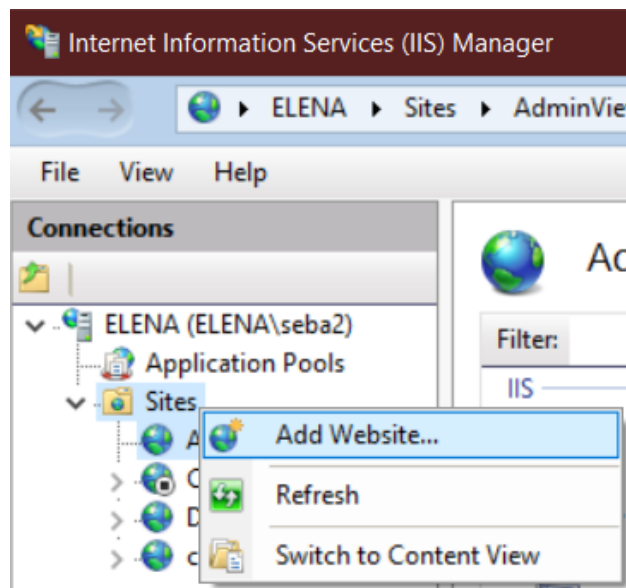
Instalación de IIS

Para instalar y desplegar un programa en IIS primero es necesario activar **Internet Information Services (IIS)** en las características de Windows. Para ello vamos a **panel de control > Programas y características > Activar o desactivar las características de Windows**.



Con esta característica de Windows activada ya se pueden desplegar los programas.

Para desplegar un programa en IIS primero necesitamos crear un sitio web, por lo que nos dirigimos a **Internet Information Services Manager > Sites > add Website**.



Al crear el sitio web se deben definir algunos valores importantes:

Con un sitio web creado para cada programa que se vaya a desplegar en IIS podemos continuar con la compilación del código fuente de cada uno de estos.

Para ejecutar la vista administrador basta con ejecutar los siguientes 2 comandos en consola en el directorio de TECARI-ADMIN:

```
npm install
```

```
ng serve -o
```

Con estos sencillos comandos será capaz de ejecutar la vista administrador.

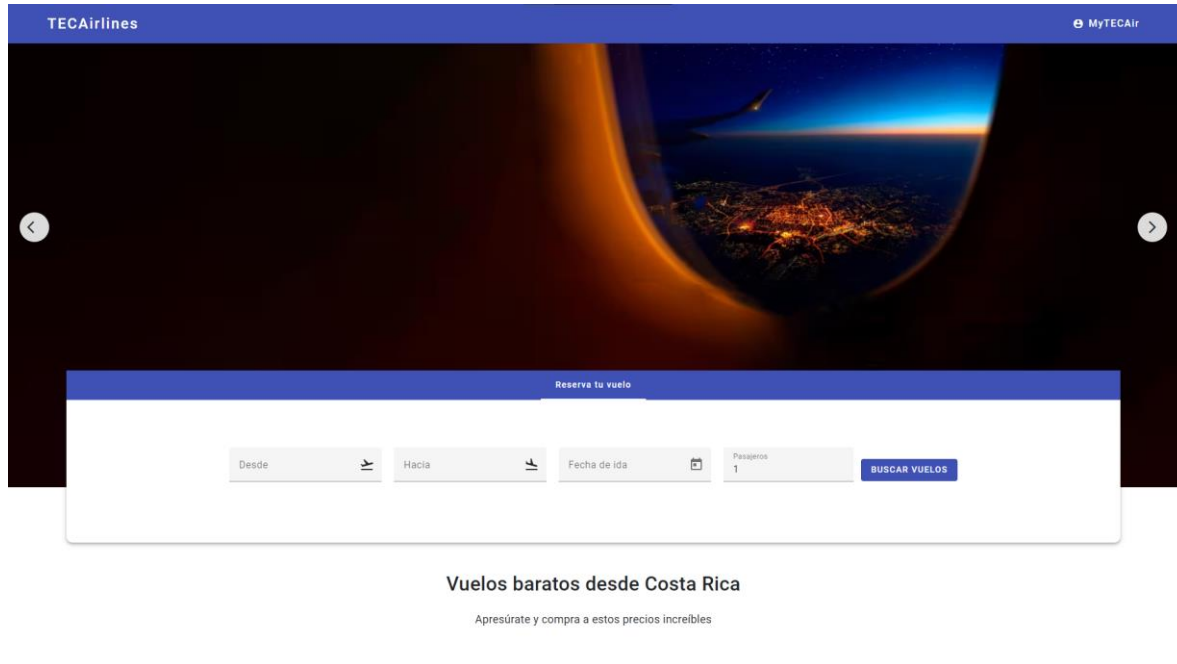
Vista cliente

De la misma forma que la vista administrador, deberá ejecutar los siguientes sencillos comandos en el respectivo directorio TECAIR-CLIENT:

Npm install

Npm i pdfmake

ng serve -o

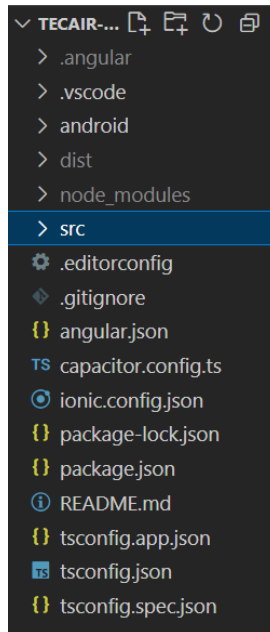


Tras ejecutar estos sencillos pasos podrá visualizar la pantalla principal de la vista cliente.

Para desplegar la vista de cliente se debe compilar el código de angular y copiar la carpeta en el directorio del sitio web de IIS

App móvil

Para generar al app móvil se debe abrir el folder de TECAIR-CLIENT en Visual Studio Code y ejecutar los siguientes comandos en consola:



Los primeros comandos son de instalación de las dependencias:

npm install para instalar las dependencias básicas de angular:

```
PS D:\Codes\TECAir\TECAir-Client> npm install
```

npm i pdfmake para importar las dependencias necesarias para gener un pdf:

```
PS D:\Codes\TECAir\TECAir-Client> Npm i pdfmake
```

npm install @capacitor/core

npm install @capacitor/cli --save-dev

npm install @capacitor/android

ng build --configuration production para actualizar la app movil con el contenido actual del repositorio:

```
PS D:\Codes\TECAir\TECAir-Client> ng build --configuration production
```

npx cap sync para sincronizar el build generado:

```
PS D:\Codes\TECAir\TECAir-Client> npx cap sync
```

npx cap open android para abrir la aplicación móvil en Android Studio:

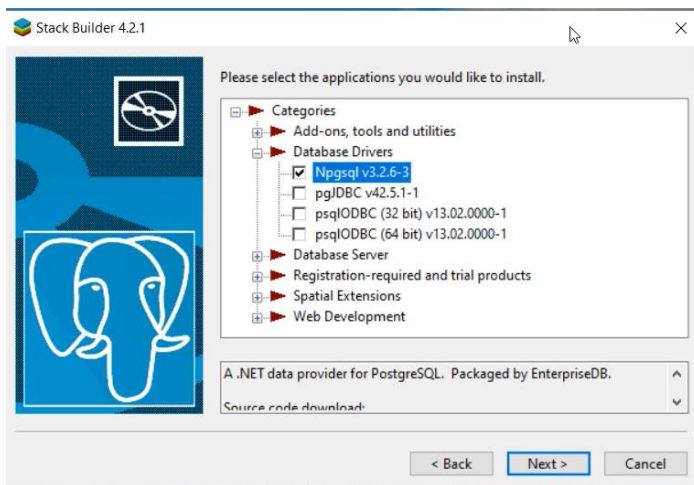
```
PS D:\Codes\TECAir\TECAir-Client> npx cap open android
```

API-Database

Existen 2 maneras diferentes de implementar una API para una Base de Datos a través del modelo entidad-relación: la primer forma es a través del enfoque de código primero y la segunda forma es mediante el enfoque de base de datos primero.

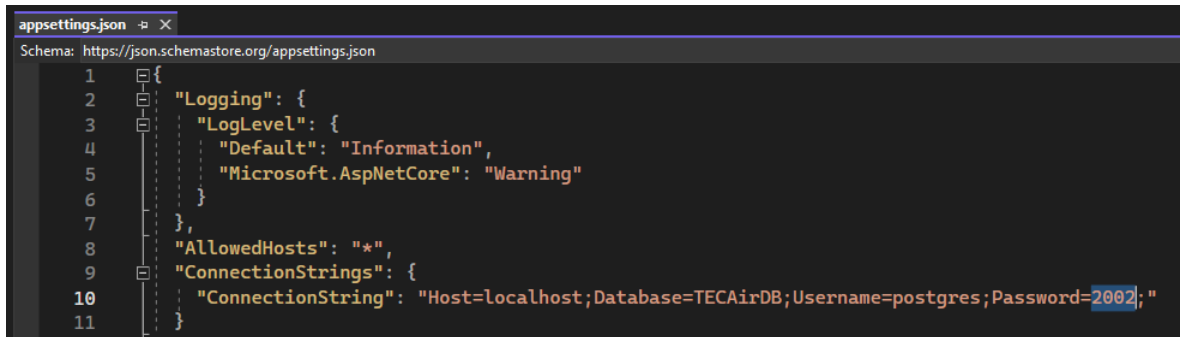
Método #1. Enfoque Código-Primero

A continuación, se detalla cómo ejecutar TECAir con ambos enfoques. El primero y más rápido es el enfoque código primero. Para esto lo primero que se debe de hacer es instalar PostgreSQL y asegurarse de instalar el siguiente driver:



Una vez instalado PostgreSQL, deberá ejecutar pgAdmin4 e ingresar a un servidor de su computador.

Luego deberá clonar el repositorio de github, abrir la solución del proyecto en Visual Studio y editar el archivo appsettings.json con su respectiva contraseña y usuario de PostgreSQL:

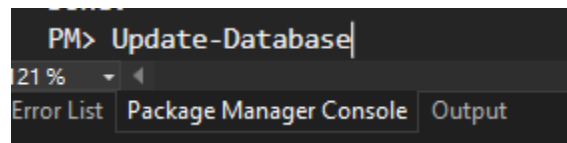


```

1  {
2    "Logging": {
3      "LogLevel": {
4        "Default": "Information",
5        "Microsoft.AspNetCore": "Warning"
6      }
7    },
8    "AllowedHosts": "*",
9    "ConnectionStrings": {
10     "ConnectionString": "Host=localhost;Database=TECAirDB;Username=postgres;Password=2002;"
11   }

```

ejecutar el comando “Update-Database” en la consola de comandos. Con lo cual se creará la base de datos. Ahora, solo debe modificar el archivo con la contraseña de su base de datos y ya podrá ejecutar el API.



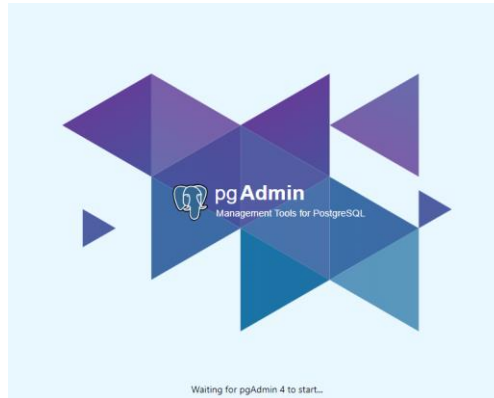
```

PM> Update-Database

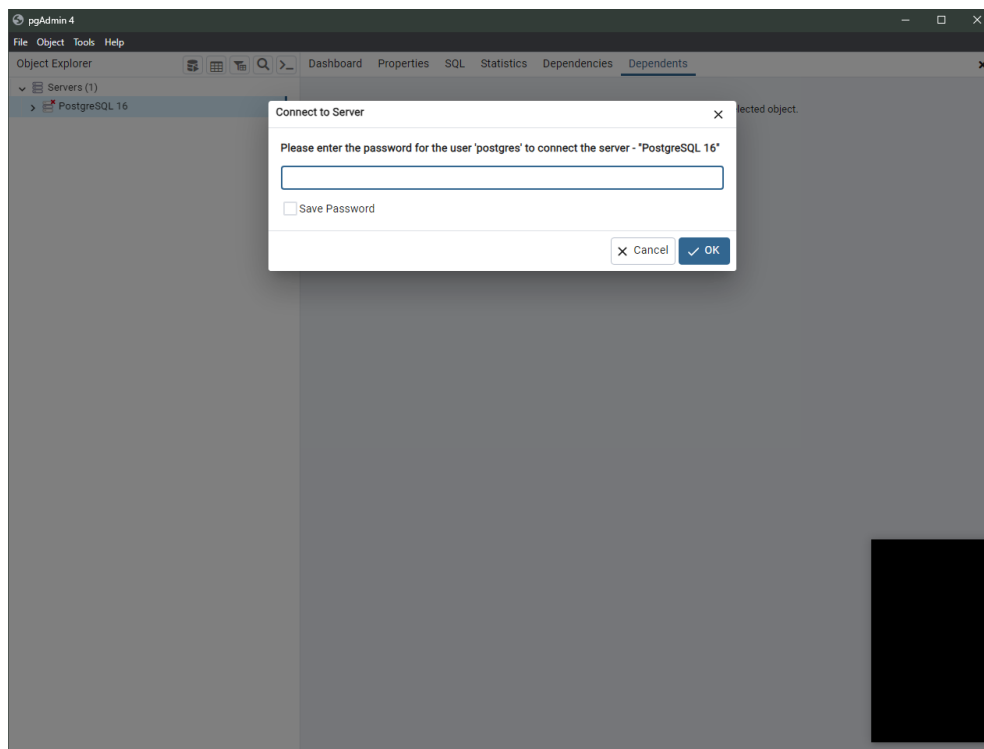
```

Método #2. Enfoque Base de Datos Primero

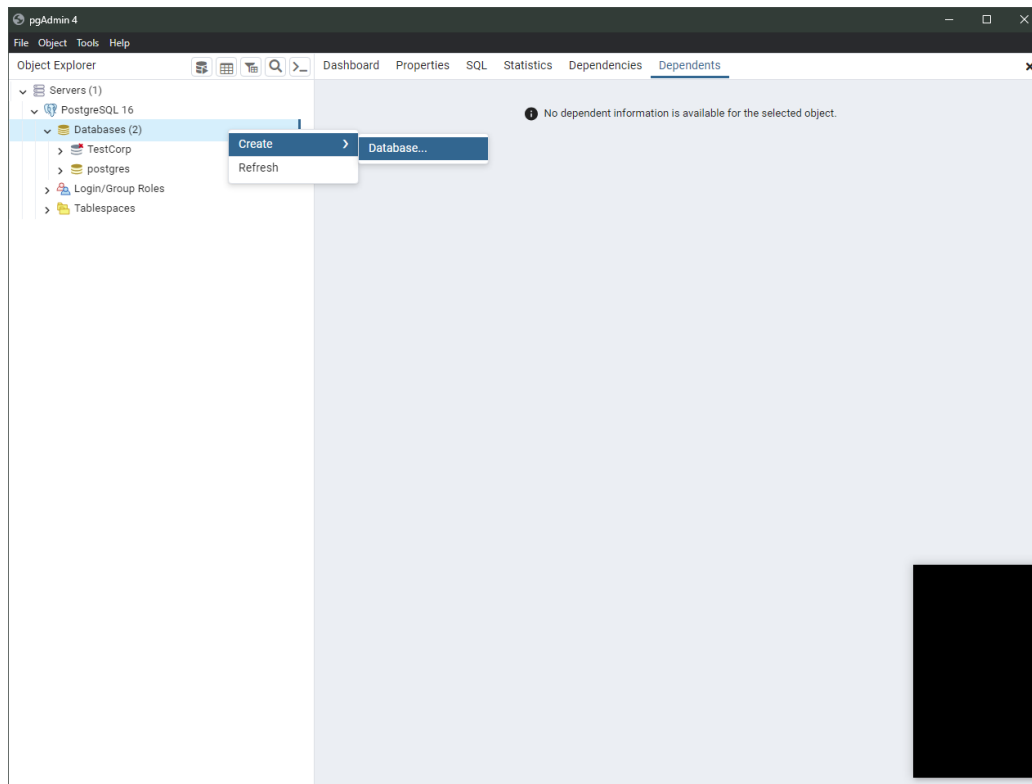
Al igual que en el método anterior, deberá instalar PostgreSQL junto con su driver para .NET y una vez hecho esto, deberá ejecutar el programa pgAdmin4 y abrir su servidor.



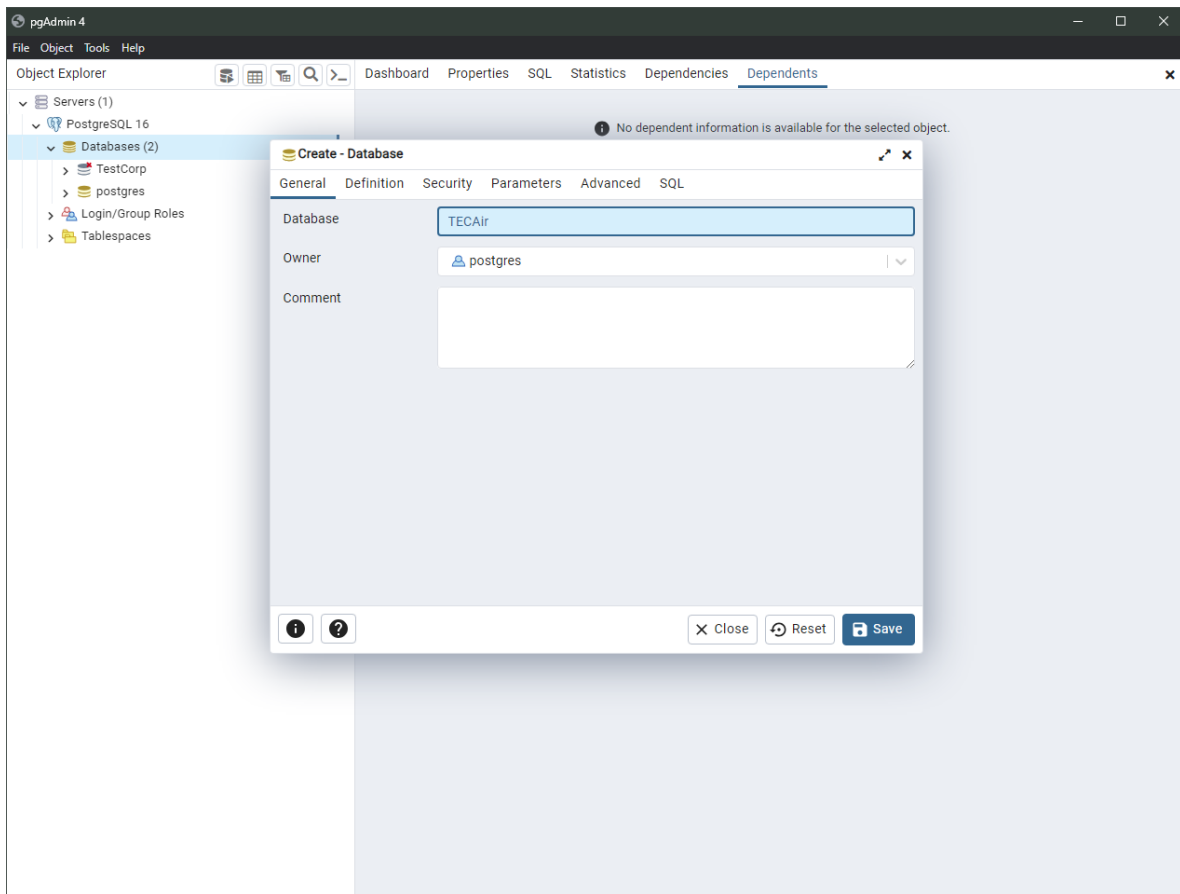
Se le solicitará su contraseña y deberá ingresarla:



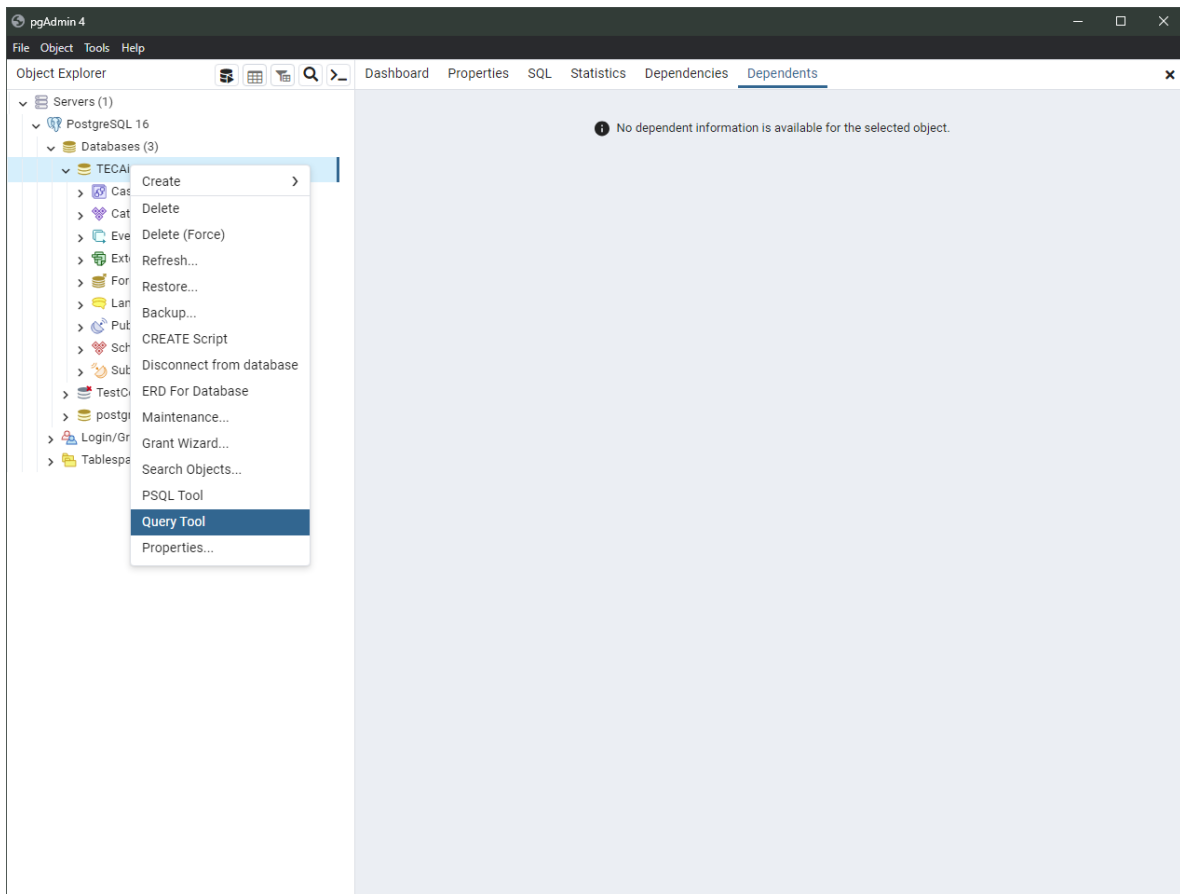
Presione Click-Derecho sobre la pestaña de “Databases”, seleccione Create -> Database...



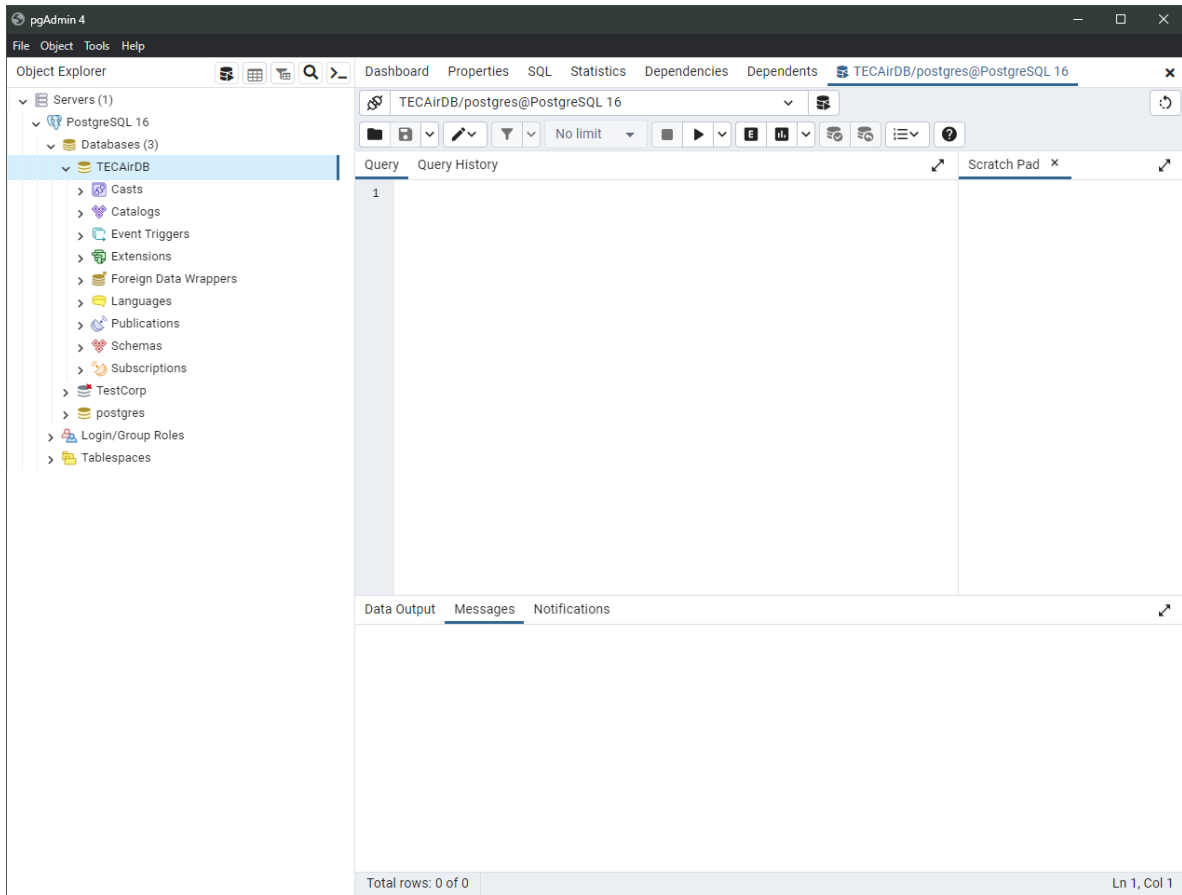
Ingresa como nombre de la base de datos “TECAirDB” y presiones “Save”.



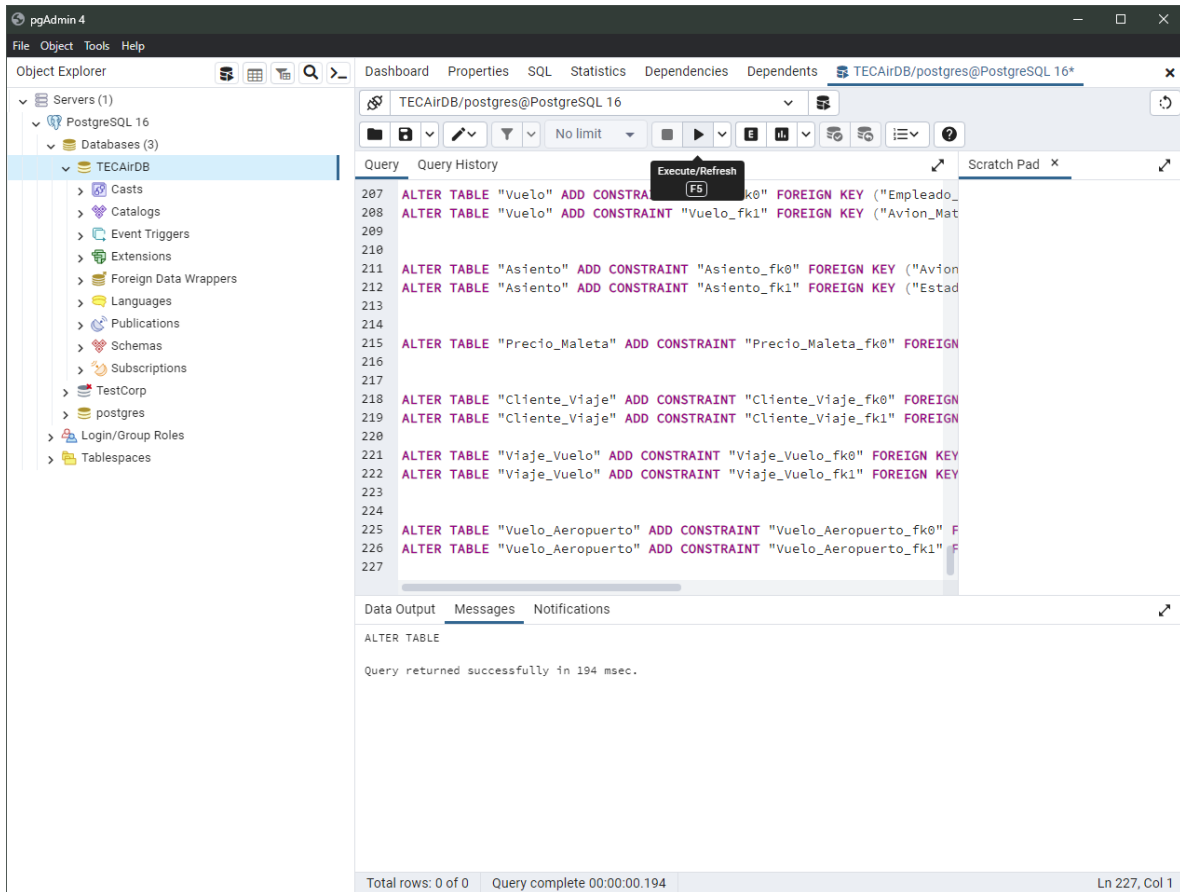
Ahora, presione Click-Derecho sobre “TECAirDB” y seleccione la opción “QueryTool”.



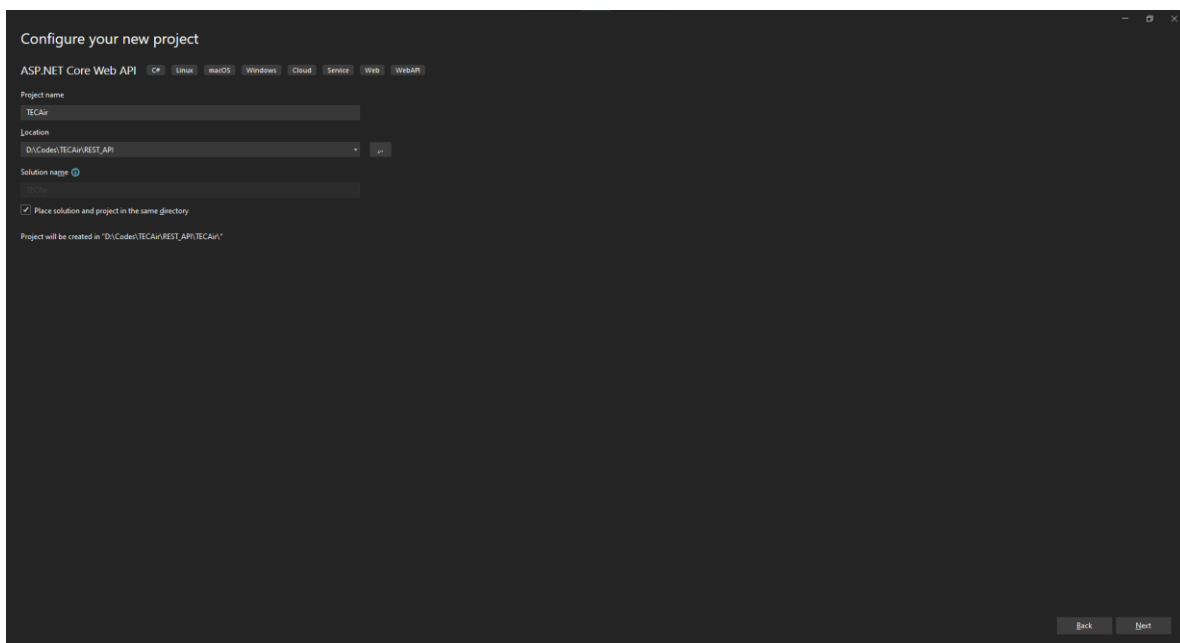
Se le abrirá la siguiente ventana:



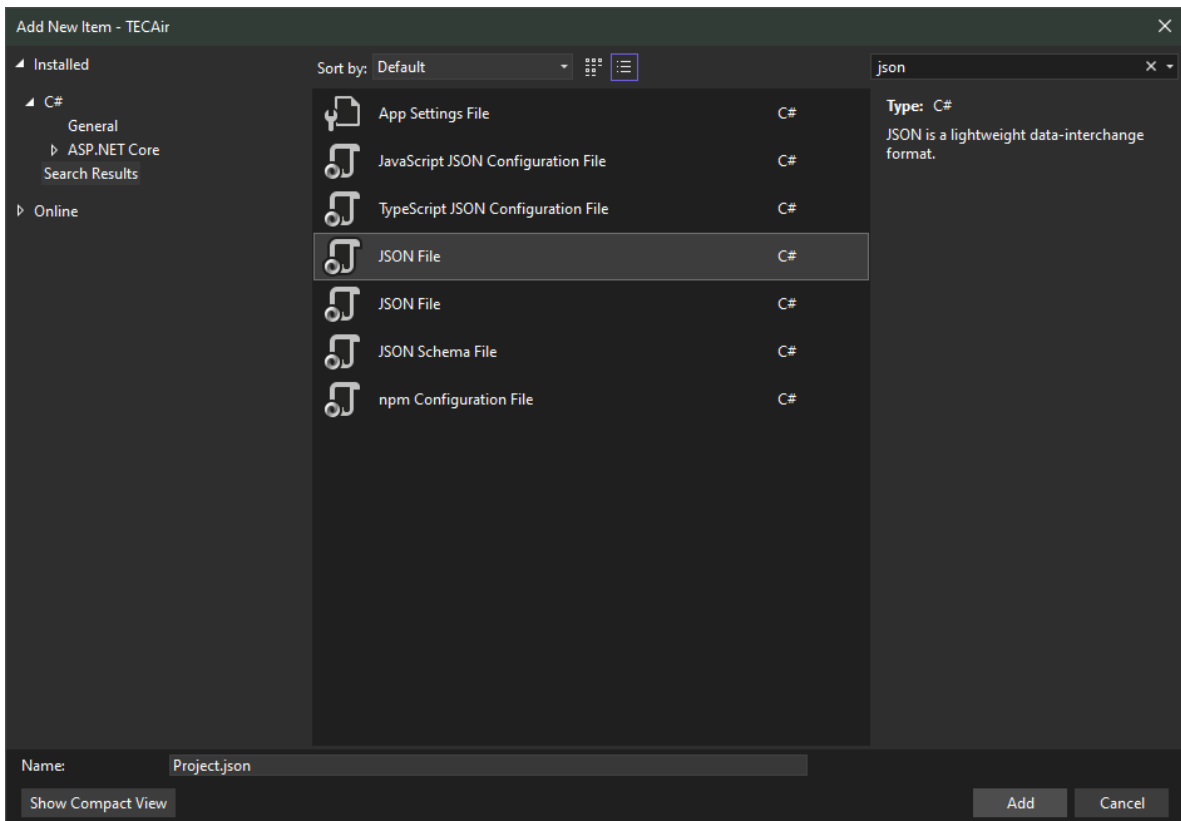
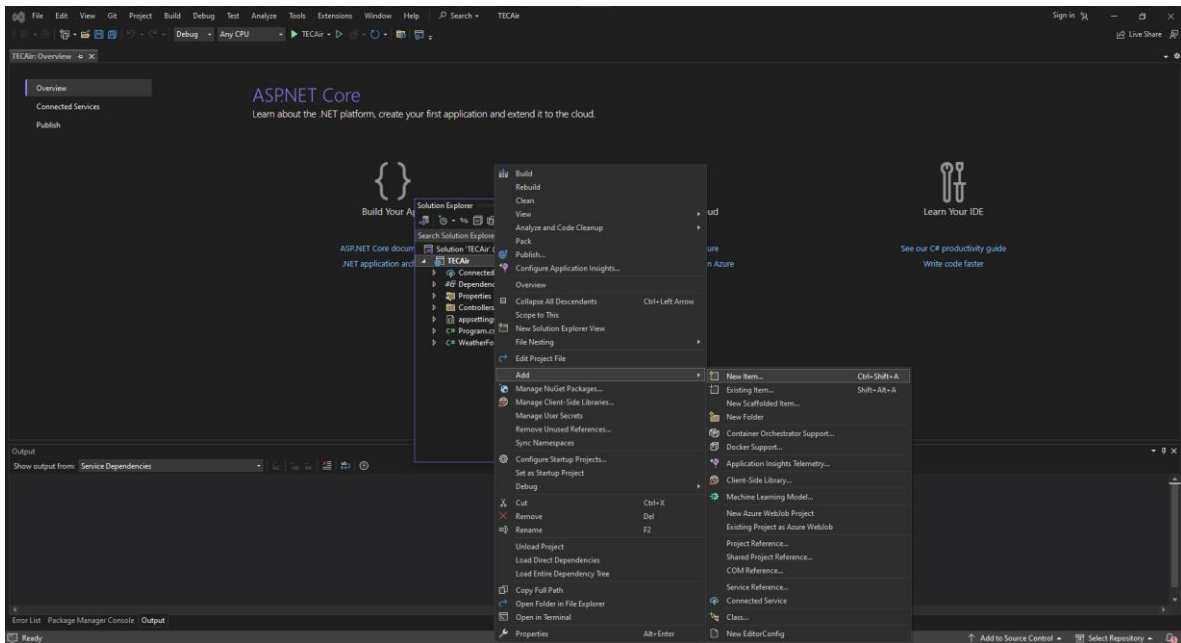
Allí deberá de pegar el código que encontrará en el archivo “CreationQuery.SQL” en la carpeta “DatabaseScripts” del proyecto y ejecutar el Query:

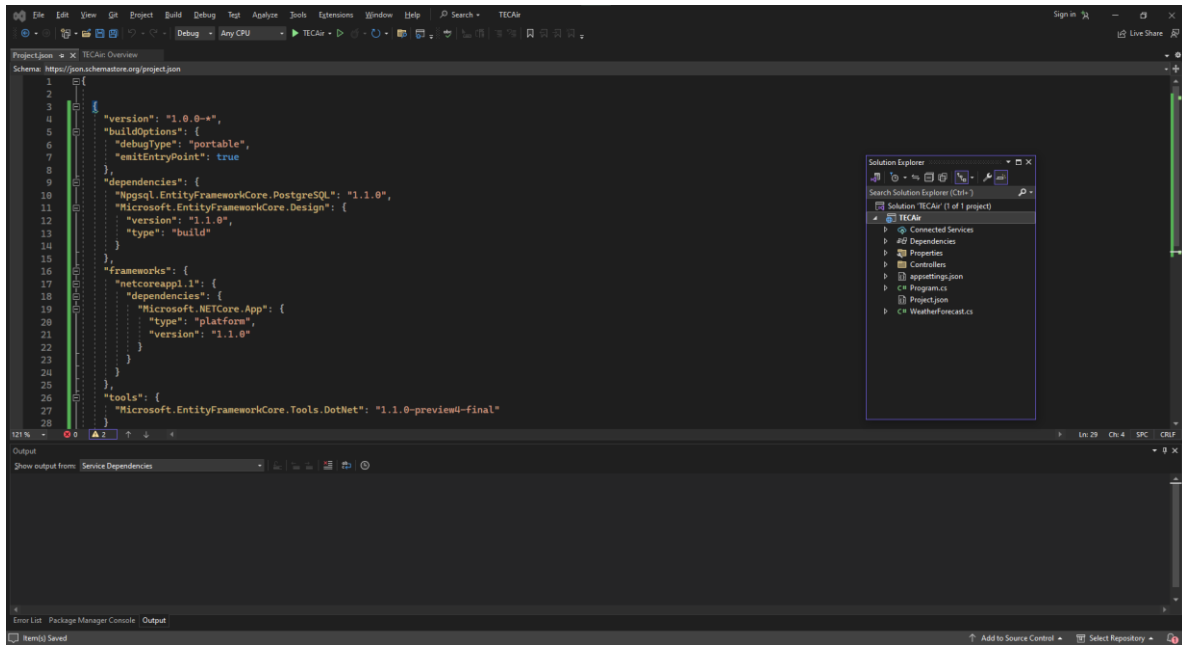


Ahora ya se ha creado la base de datos. Ahora hay que crear un proyecto Web API en Visual Studio:

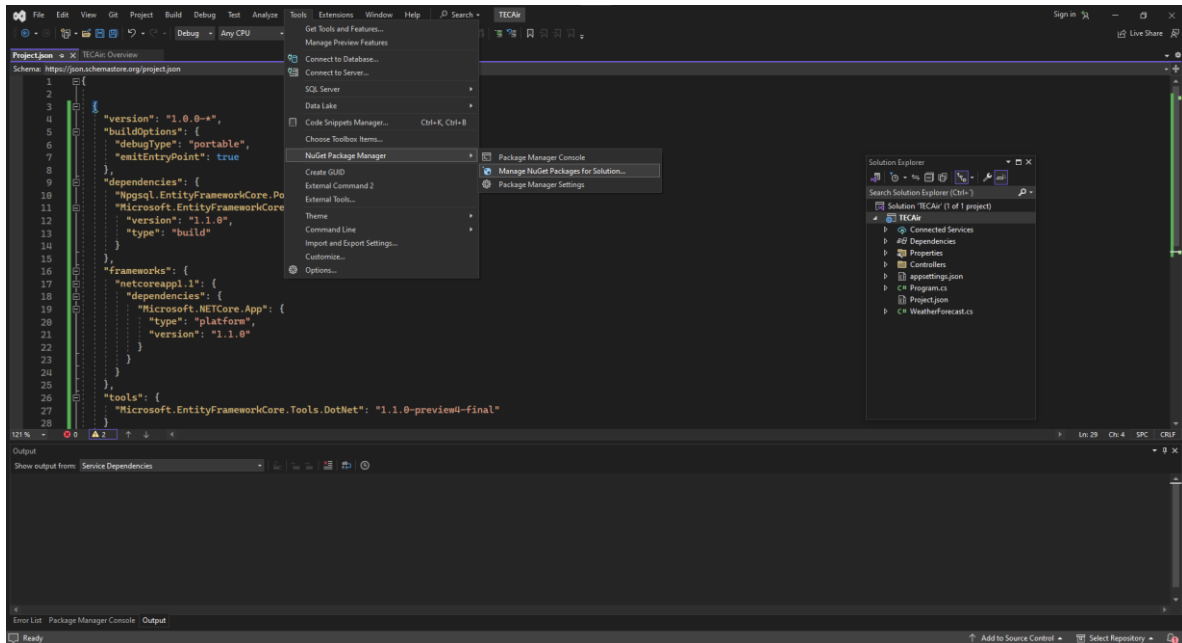


El siguiente paso es crear un nuevo ítem de tipo .json llamado “Project.json” y pegar el mismo código que encontrará en la carpeta del proyecto con el mismo nombre.



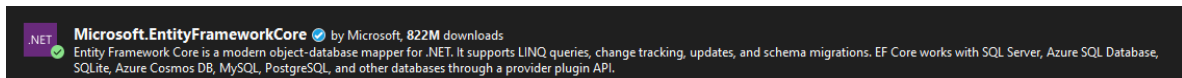


El siguiente paso será ir a Tools -> NuGet Package Manager -> Manage NuGet Packages for Solution:



Y buscar e instalar las siguientes extensiones:

Microsoft.EntityFrameworkCore



Microsoft.EntityFrameworkCore.Tools



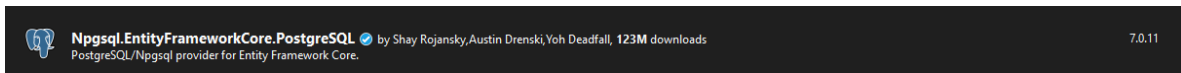
Microsoft.EntityFrameworkCore.Design



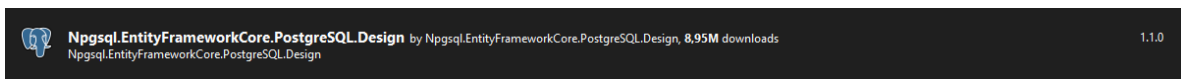
Microsoft.NETCore.App (2.1.30)



Npgsql.EntityFrameworkCore.PostgreSQL



Npgsql.EntityFrameworkCore.PostgreSQL.Design



El siguiente paso será ir a Tools -> NuGet Package Manager -> Package Manager Console y ejecutar “dotnet restore”:

```
Package Manager Console Host Version 6.7.0.127

Type 'get-help NuGet' to see all available NuGet commands.

PM> dotnet restore
Determining projects to restore...
Restored D:\Codes\TECAir\REST_API\TECAir\TECAir.csproj (in 327 ms).
PM> |
```

Luego de esto se debe ejecutar “Scaffold-DbContext

"Host=localhost;Database=mydatabase;Username=myuser;Password=mypassword"

Npgsql.EntityFrameworkCore.PostgreSQL -o Models” adaptado a su respectivo usuario y contraseña:

```
PM> Scaffold-DbContext "Host=localhost;Database=TECAirDB;Username=postgres;Password=2002" Npgsql.EntityFrameworkCore.PostgreSQL -o Models
Build started...
Build succeeded.
To protect potentially sensitive information in your connection string, you should move it out of source code. You can avoid scaffolding the connection string by using the Name= syntax to read it from configuration - see https://go.microsoft.com/fwlink/?linkid=2131148. For more guidance on storing connection strings, see http://go.microsoft.com/fwlink/?linkid=723263.
PM> |
```

Ahora deberá ir a Models -> TECAirDbContext.cs y comentar o borrar las líneas 47-51 correspondientes a la conexión con la base de datos:

```
46 public virtual DbSet<VueloAeropuerto> VueloAeropuertos { get; set; }
47
48 //protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
49 //warning To protect potentially sensitive information in your connection string, you should move it out of source code. You can avoid scaffolding the connection string by using the Name= syn
50 //=> optionsBuilder.UseNpgsql("Host=localhost;Database=TECAirDB;Username=postgres;Password=2002");
51
```

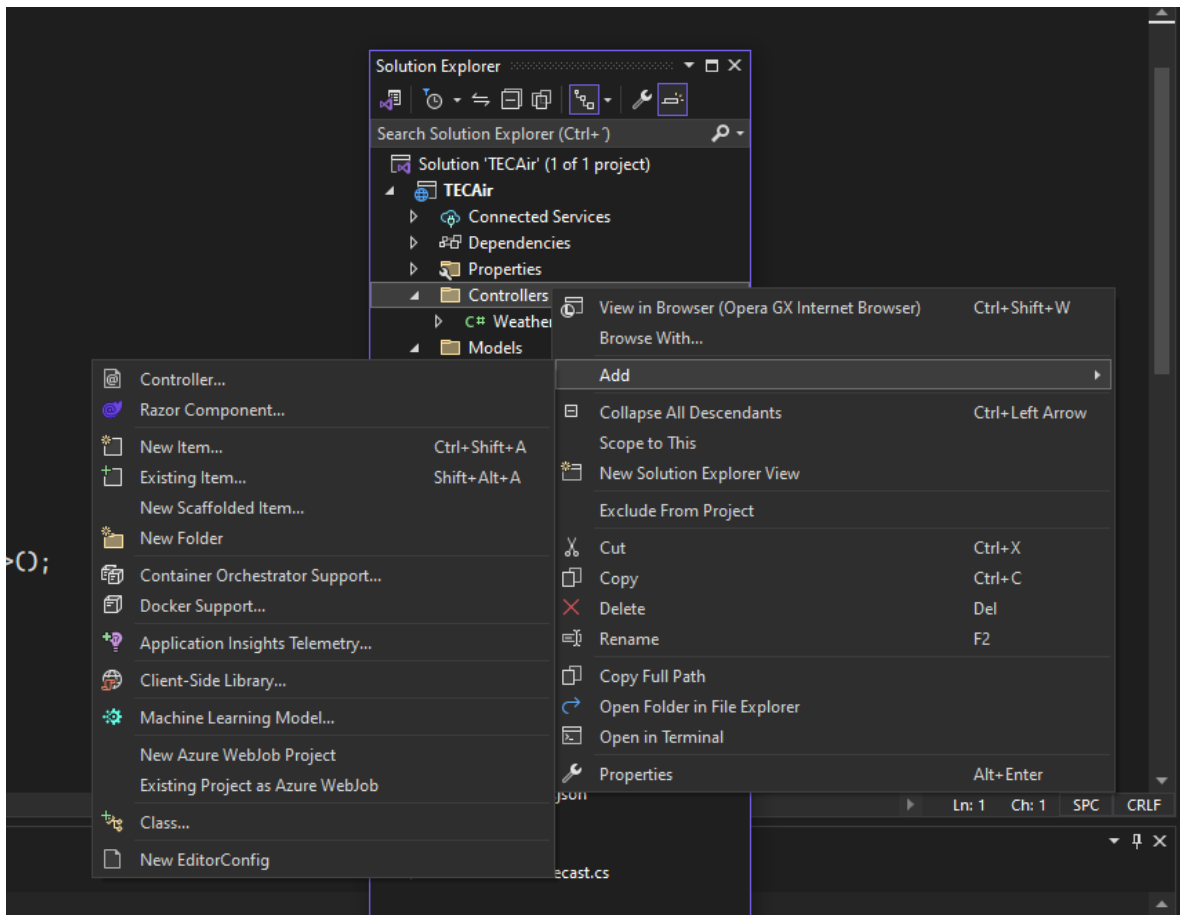
En su lugar ubicará los datos de la conexión con la base de datos en el archivo appsettings.json:

```
controller.cs | Program.cs | appsettings.json | TecairDbContext.cs
ps://json.schemastore.org/appsettings.json
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "ConnectionString": "Host=localhost;Database=TECAirDB;Username=postgres;Password=2002;"
  }
}
```

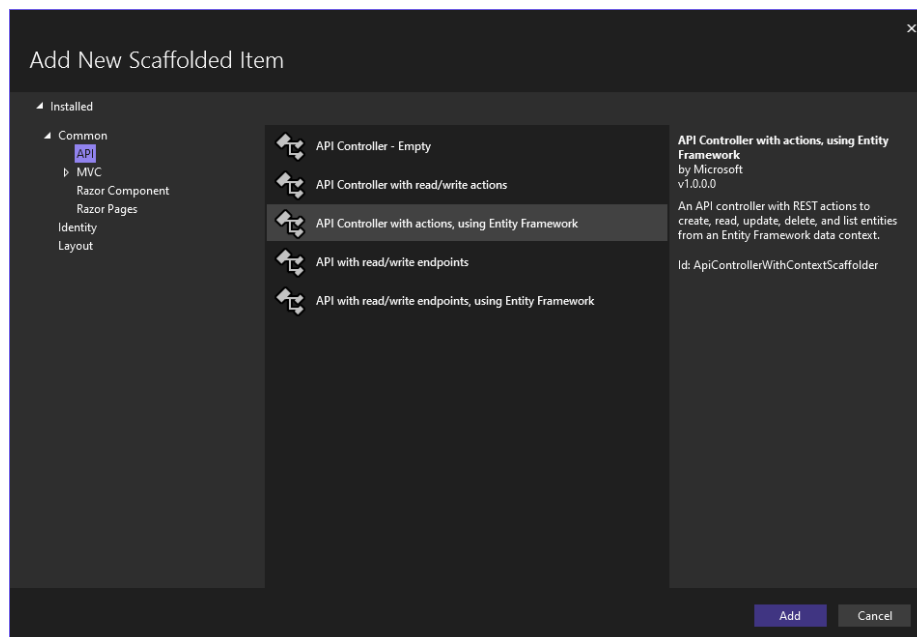
Y pegará el código a continuación en las líneas 13-14 de la siguiente imagen para que se ubiquen los datos de conexión con la base de datos.

```
Program.cs* | appsettings.json | TecairDbContext.cs*
TECAir
1 using Microsoft.EntityFrameworkCore;
2 using TECAir.Models;
3
4 var builder = WebApplication.CreateBuilder(args);
5
6 // Add services to the container.
7
8 builder.Services.AddControllers();
9 // Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
10 builder.Services.AddEndpointsApiExplorer();
11 builder.Services.AddSwaggerGen();
12
13 builder.Services.AddDbContext<TecairDbContext>(options =>
14     options.UseNpgsql(builder.Configuration.GetConnectionString("ConnectionString")));
15
16 var app = builder.Build();
```

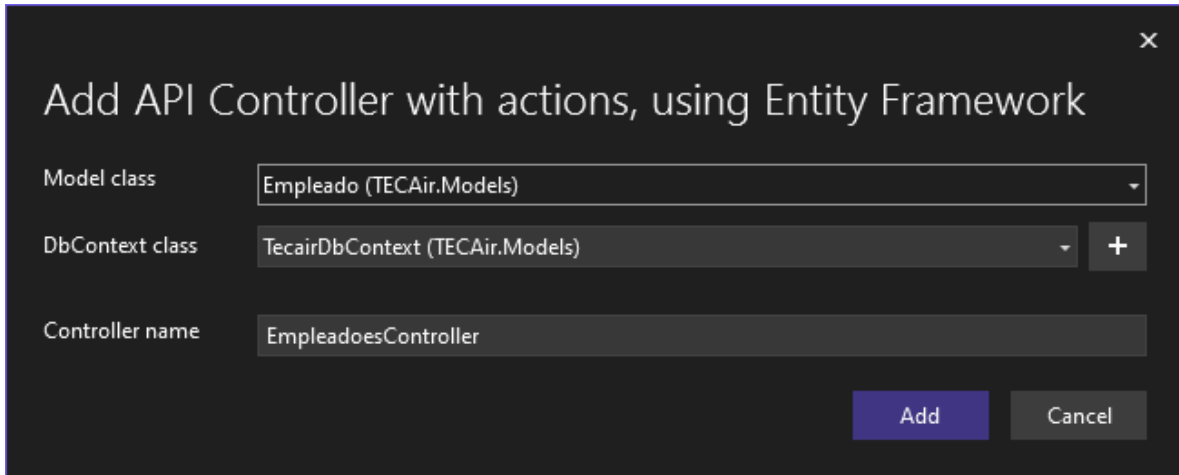
Ahora que la base de datos se encuentra configurada, podrá generar un controlador para probarla. Diríjase a Controllers, presione Click-Derecho y seleccione generar un nuevo item scaffold:



Elija la siguiente opción:



Seleccione el contexto correspondiente primero y luego el modelo al cuál le desea generar un controlador:



Close (X)

Add API Controller with actions, using Entity Framework

Model class:

DbContext class:

Controller name:

Una vez generado el controlador, ya debería poder ejecutar el programa. Si está configurando un nuevo proyecto, puede ejecutar <Add-Migration “Initial Migration”> para que la próxima persona que ejecute el API pueda generar la base de datos con solo ejecutar Update-Database.