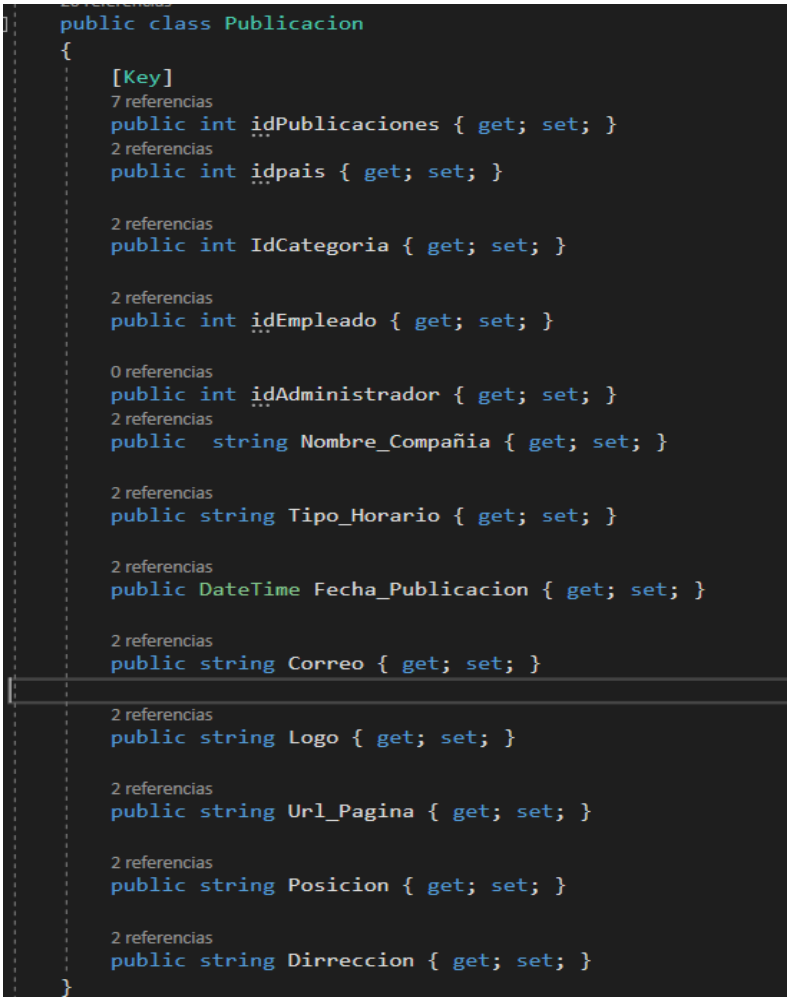
Manual Técnico con documentación del API

En esta primera parte vamos a crear un proyecto ASP.NET Core Web y crear una plantilla de Web Api sin autenticación. Luego de esto lo vamos a separar el proyecto por capas agregando clase librería.

MODEL

Representa nuestro modelo de la capa de dominio, es decir las tablas mapeadas a clases.

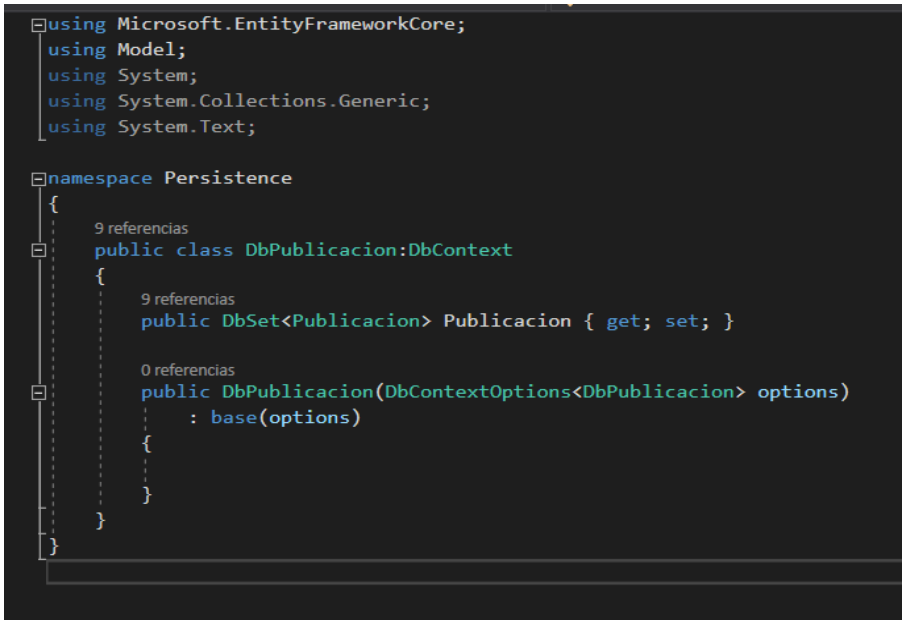


```
public class Publicacion
{
    [Key]
    7 referencias
    public int idPublicaciones { get; set; }
    2 referencias
    public int idpais { get; set; }

    2 referencias
    public int IdCategoria { get; set; }

    2 referencias
    public int idEmpleado { get; set; }

    0 referencias
    public int idAdministrador { get; set; }
    2 referencias
    public  string Nombre_Compañia { get; set; }

    2 referencias
    public string Tipo_Horario { get; set; }

    2 referencias
    public DateTime Fecha_Publicacion { get; set; }

    2 referencias
    public string Correo { get; set; }

    2 referencias
    public string Logo { get; set; }

    2 referencias
    public string Url_Pagina { get; set; }

    2 referencias
    public string Posicion { get; set; }

    2 referencias
    public string Dirreccion { get; set; }
}
```

PERSISTENCE

Nuestra clase que será encargada de manejar las migraciones y la conexión a la base de datos. Para hacer funcionar esta parte debemos agregar los siguientes package del NuGet y luego crear nuestro DbContext.

- EntityFrameworkCore.SqlServer
- EntityFrameworkCore.Tools

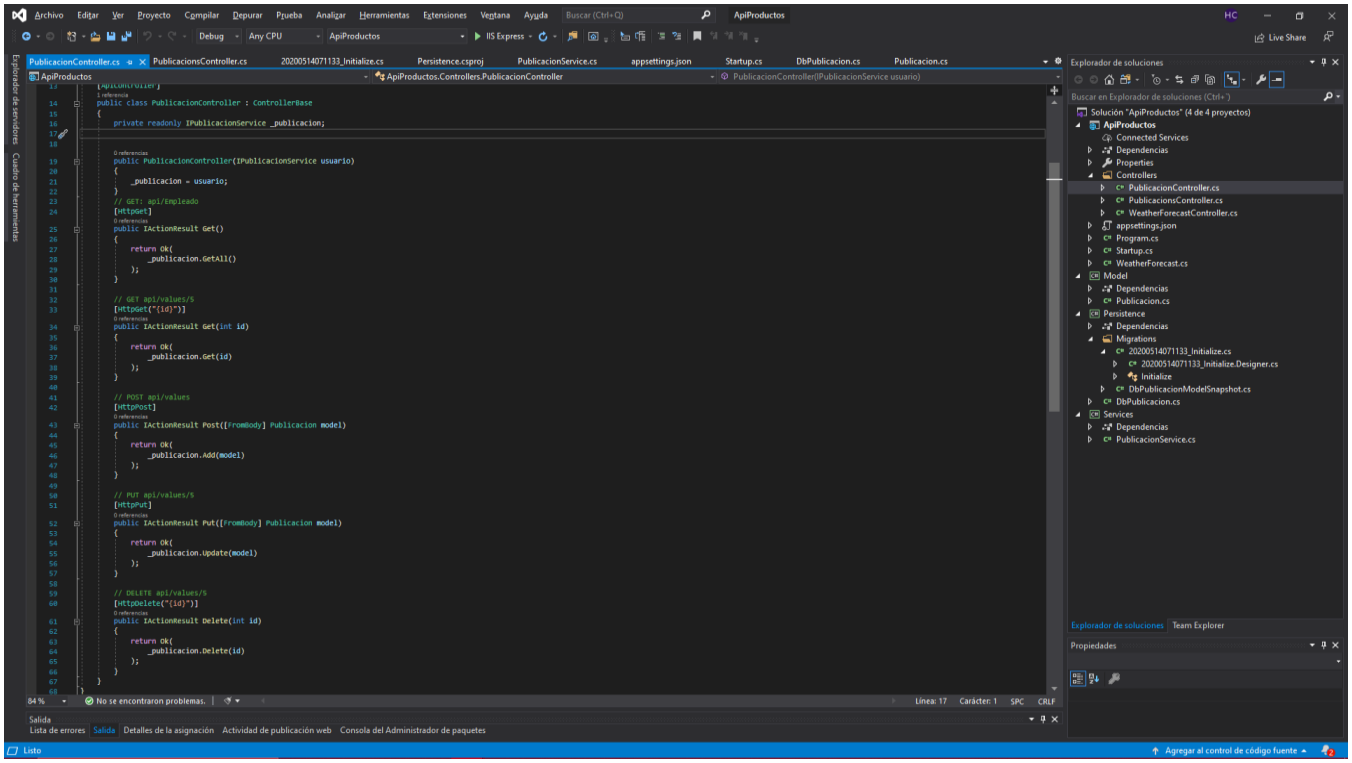


```
using Microsoft.EntityFrameworkCore;
using Model;
using System;
using System.Collections.Generic;
using System.Text;

namespace Persistence
{
    9 referencias
    public class DbPublicacion:DbContext
    {
        9 referencias
        public DbSet<Publicacion> Publicacion { get; set; }

        0 referencias
        public DbPublicacion(DbContextOptions<DbPublicacion> options)
            : base(options)
        {

        }
    }
}
```

Procedemos a registrar dependencia en nuestro StartUp para que esté disponible para todo el proyecto.

```
2 referencias
public class Startup
{
    0 referencias
    public Startup(IConfiguration configuration)
    {
        Configuration = configuration;
    }

    2 referencias
    public IConfiguration Configuration { get; }

    // This method gets called by the runtime. Use this method to add services to the container.
    0 referencias
    public void ConfigureServices(IServiceCollection services)
    {
        var connection = Configuration.GetConnectionString("Dev");
        services.AddControllers();
        services.AddDbContext<DbPublicacion>(options => options.UseSqlServer(connection));
        services.AddTransient<IPublicacionService, PublicacionService>();
        services.AddCors(options =>
        {
            options.AddPolicy("AllowSpecificOrigin", builder =>
                builder.AllowAnyHeader()
                       .AllowAnyMethod()
                       .AllowAnyOrigin()
            );
        });
        services.AddMvc();


    }
```

Nuestra capa REST que implementa los recursos a exponer como API.

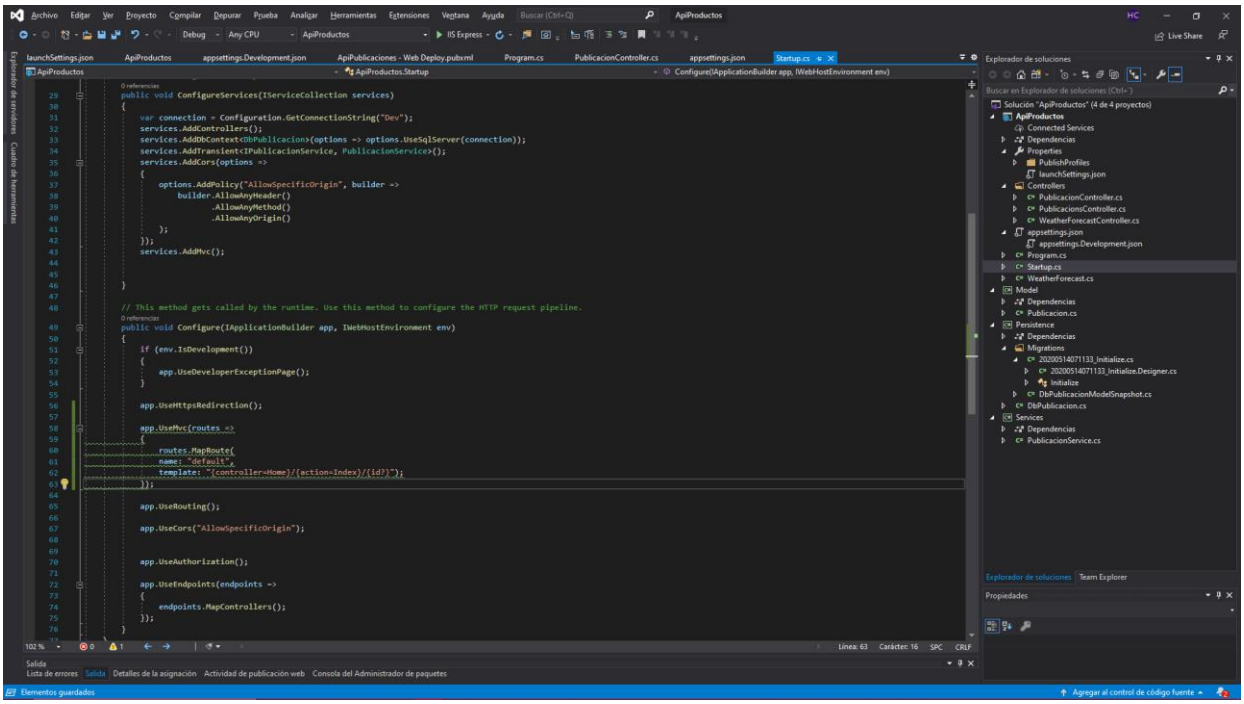


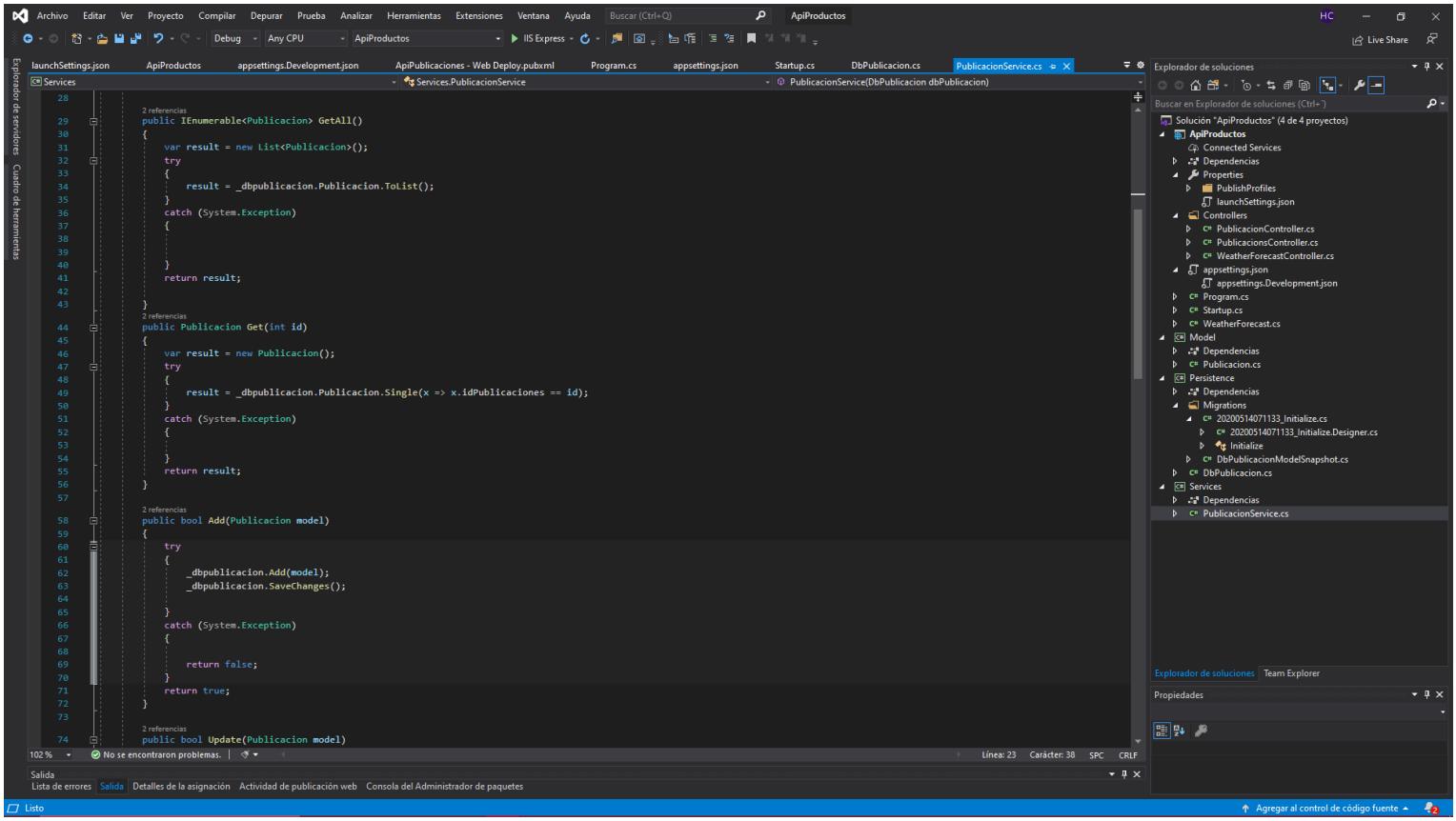
Con respecto a las rutas no hemos escrito ni unas reglas, así que estamos usando el ruteo REST. En un proyecto real preferimos modificar la escritura desde el StartUp tal como lo genera un proyecto MVC.

```
app.UseMvc(routes =>

{

    routes.MapRoute(

    name: "default",

    template: "{controller=Home}/{action=Index}/{id?}");

});
```

SERVICES

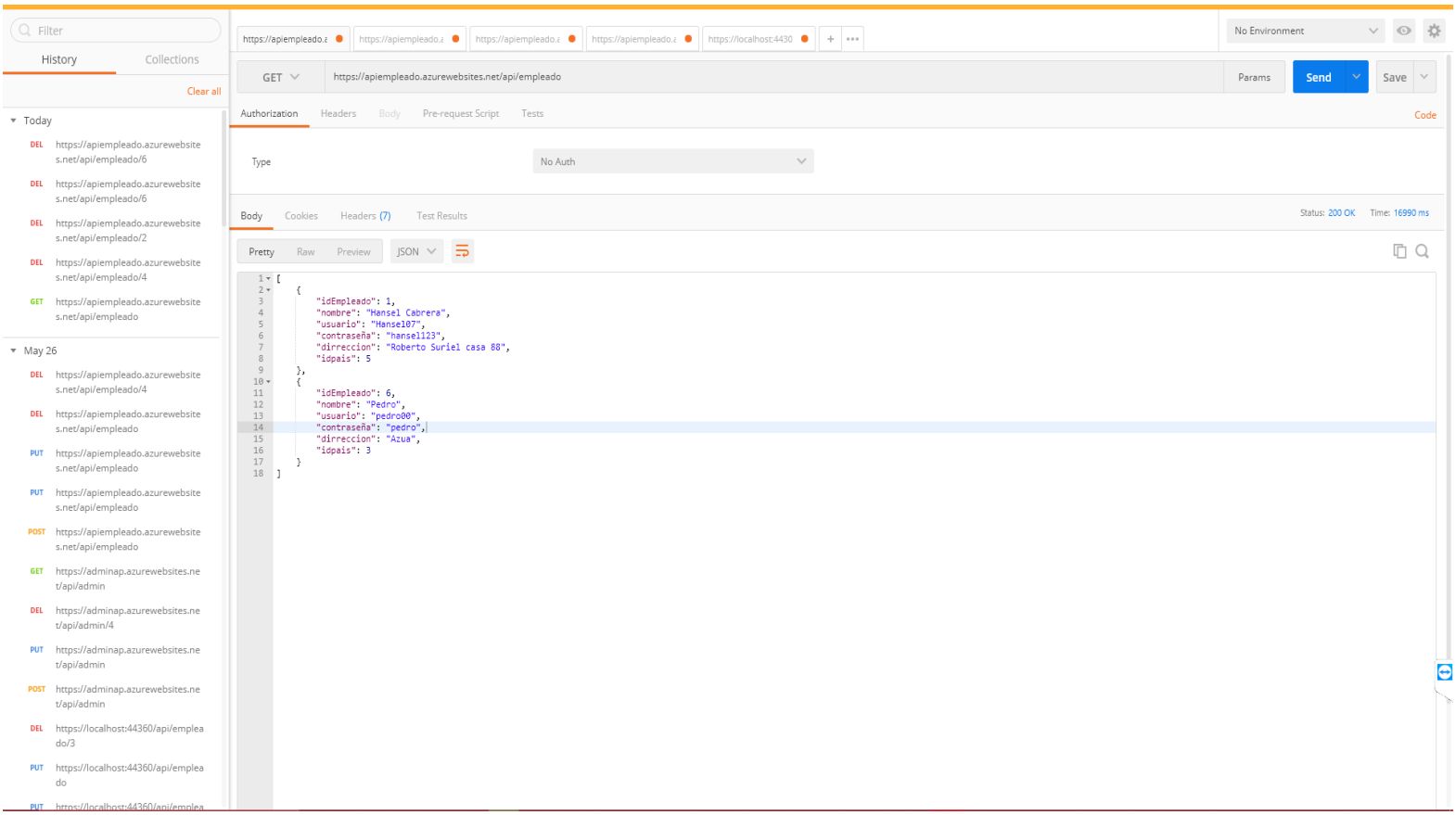
Nuestros servicios que vamos a declarar que se encargarán de realizar las consultas a la base de datos.



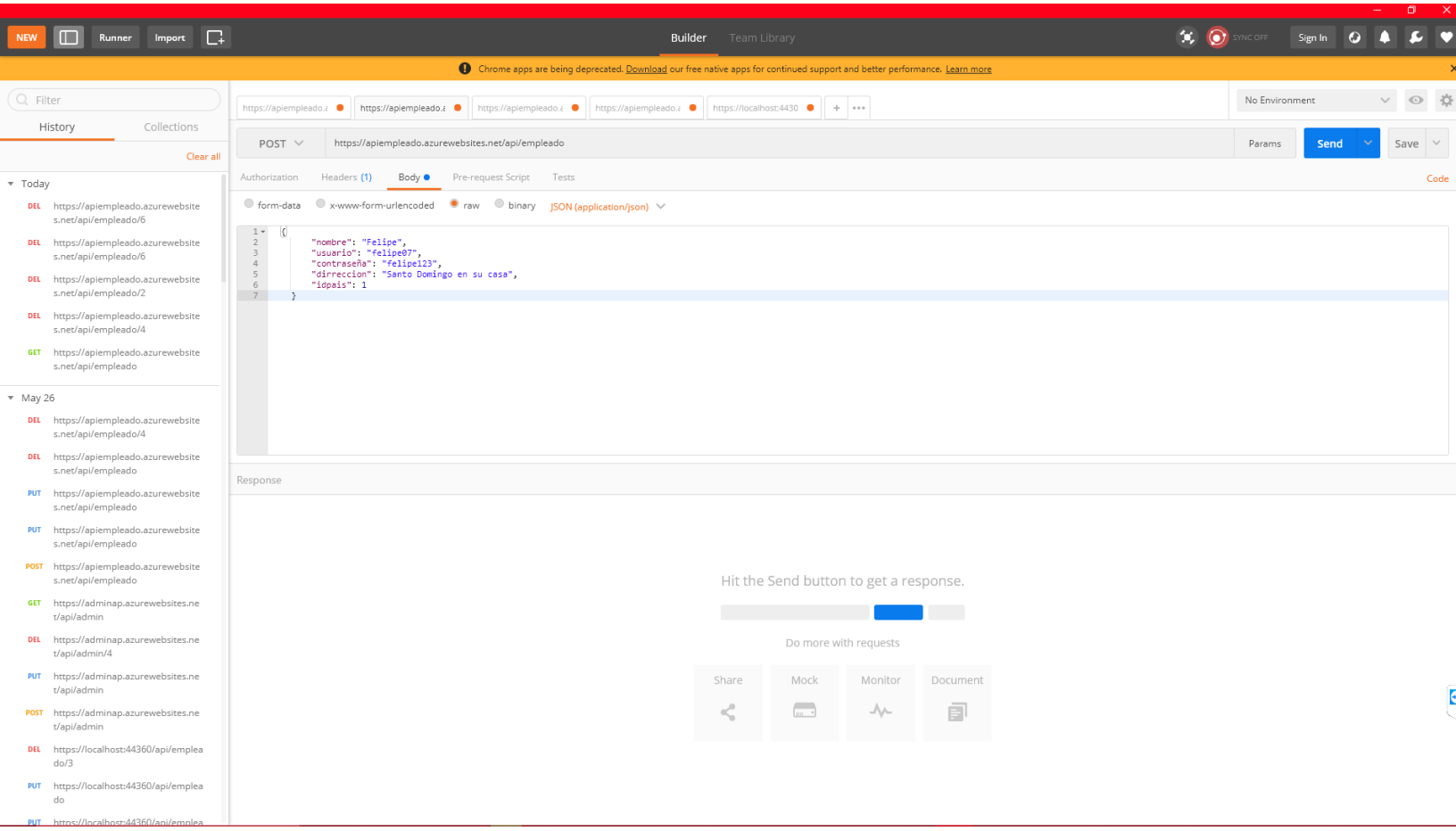

launchSettings.json
ApiProductos
appsettings.Development.json
ApiPublicaciones - Web Deploy.pubxml
Program.cs
appsettings.json
Startup.cs
DbPublicacion.cs
PublicacionService.cs
Services
Services.PublicacionService
PublicacionService(DbPublicacion dbPublicacion)
2 referencias
public IEnumerable<Publicacion> GetAll()
{
var result = new List<Publicacion>();
try
{
result = _dbpublicacion.Publicacion.ToList();
}
catch (System.Exception)
{
}
return result;
}
2 referencias
public Publicacion Get(int id)
{
var result = new Publicacion();
try
{
result = _dbpublicacion.Publicacion.Single(x => x.idPublicaciones == id);
}
catch (System.Exception)
{
}
return result;
}
2 referencias
public bool Add(Publicacion model)
{
try
{
_dbpublicacion.Add(model);
_dbpublicacion.SaveChanges();
}
catch (System.Exception)
{
return false;
}
return true;
}
2 referencias
public bool Update(Publicacion model)
No se encontraron problemas.
Línea: 23 Carácter: 38 SPC CRLF
Salida
Lista de errores Salida Detalles de la asignación Actividad de publicación web Consola del Administrador de paquetes
Explorador de soluciones
Solución "ApiProductos" (4 de 4 proyectos)
ApiProductos
Connected Services
Dependencias
Properties
PublishProfiles
launchSettings.json
Controllers
PublicacionController.cs
PublicacionsController.cs
WeatherForecastController.cs
appsettings.json
appsettings.Development.json
Program.cs
Startup.cs
WeatherForecast.cs
Model
Dependencias
Publicacion.cs
Persistence
Dependencias
Migrations
20200514071133_Initialize.cs
20200514071133_Initialize.Designer.cs
Initialize
DbPublicacionModelSnapshot.cs
DbPublicacion.cs
Services
Dependencias
PublicacionService.cs
Explorador de soluciones Team Explorer
Propiedades
Listo
Agregar al control de código fuente

Probando API en Postman Google

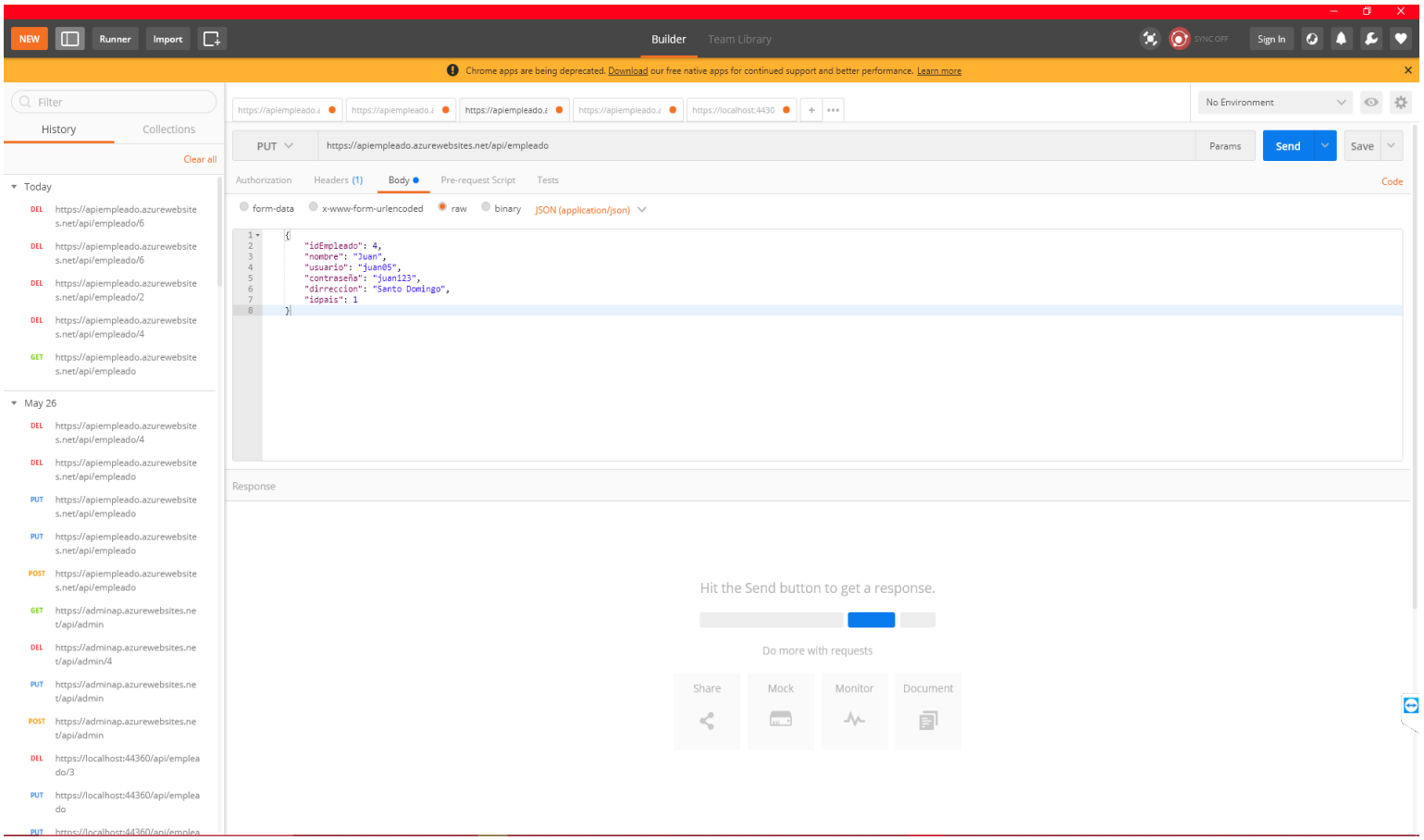
Get



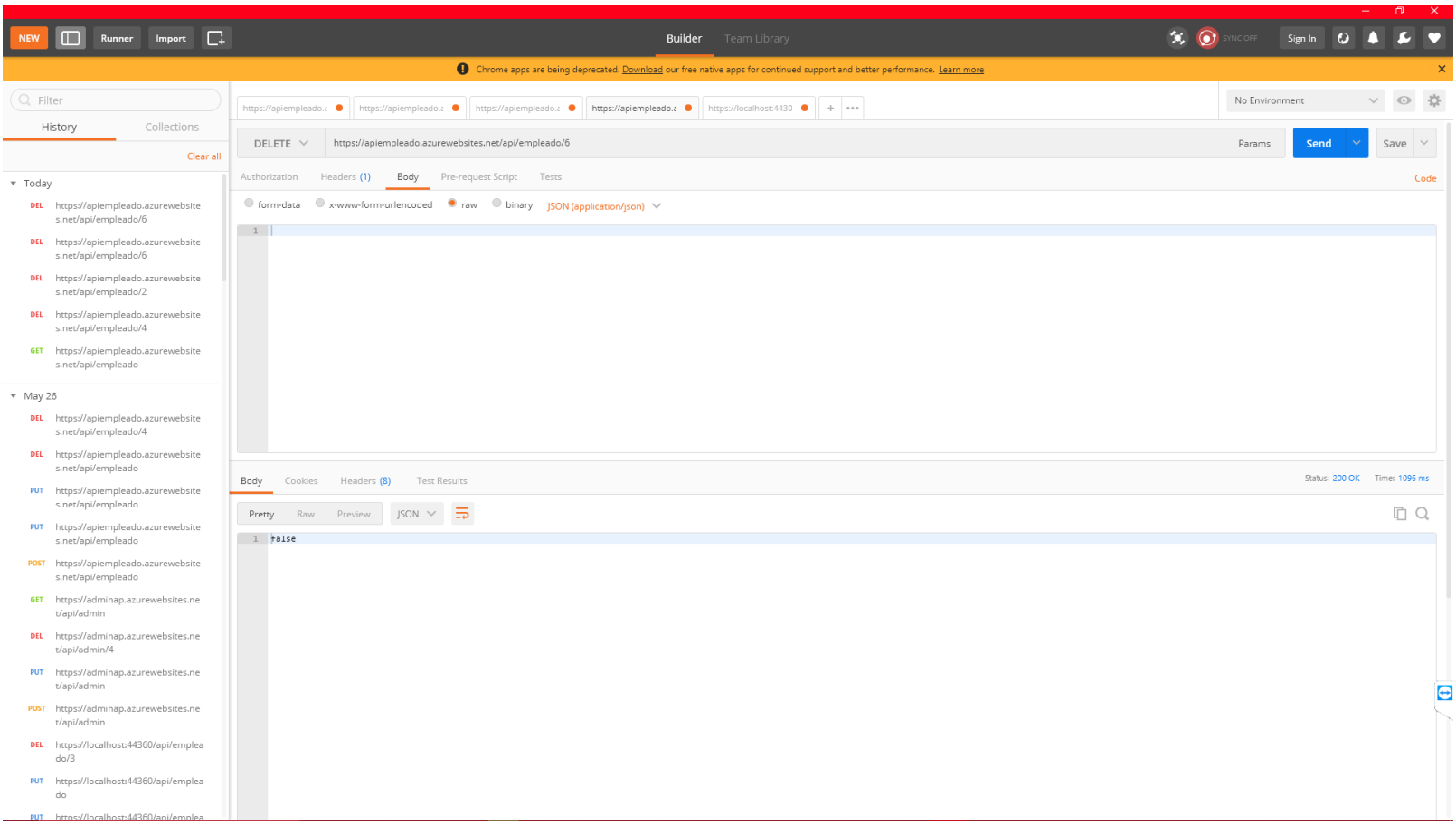
Post


NEW
Runner
Import
Builder
Team Library
Sign In
Chrome apps are being deprecated. Download our free native apps for continued support and better performance. Learn more
Filter
History
Collections
Clear all
Today
DEL https://apiempleado.azurewebsites.net/api/empleado/6
DEL https://apiempleado.azurewebsites.net/api/empleado/6
DEL https://apiempleado.azurewebsites.net/api/empleado/2
DEL https://apiempleado.azurewebsites.net/api/empleado/4
GET https://apiempleado.azurewebsites.net/api/empleado
May 26
DEL https://apiempleado.azurewebsites.net/api/empleado/4
DEL https://apiempleado.azurewebsites.net/api/empleado
PUT https://apiempleado.azurewebsites.net/api/empleado
PUT https://apiempleado.azurewebsites.net/api/empleado
POST https://apiempleado.azurewebsites.net/api/empleado
GET https://adminap.azurewebsites.net/api/admin
DEL https://adminap.azurewebsites.net/api/admin/4
PUT https://adminap.azurewebsites.net/api/admin
POST https://adminap.azurewebsites.net/api/admin
DEL https://localhost:44360/api/empleado/3
PUT https://localhost:44360/api/empleado
No Environment
POST
https://apiempleado.azurewebsites.net/api/empleado
Params
Send
Save
Authorization
Headers (1)
Body
Pre-request Script
Tests
Code
form-data
x-www-form-urlencoded
raw
binary
JSON (application/json)
{
"nombre": "Felipe",
"usuario": "felipe07",
"contraseña": "felipe123",
"dirreccion": "Santo Domingo en su casa",
"idpais": 1
}
Response
Hit the Send button to get a response.
Do more with requests
Share
Mock
Monitor
Document

Put


NEW
Runner
Import
Builder
Team Library
Sign In
Chrome apps are being deprecated. Download our free native apps for continued support and better performance. Learn more
Filter
History
Collections
Clear all
Today
DEL https://apiempleado.azurewebsites.net/api/empleado/6
DEL https://apiempleado.azurewebsites.net/api/empleado/6
DEL https://apiempleado.azurewebsites.net/api/empleado/2
DEL https://apiempleado.azurewebsites.net/api/empleado/4
GET https://apiempleado.azurewebsites.net/api/empleado
May 26
DEL https://apiempleado.azurewebsites.net/api/empleado/4
DEL https://apiempleado.azurewebsites.net/api/empleado
PUT https://apiempleado.azurewebsites.net/api/empleado
PUT https://apiempleado.azurewebsites.net/api/empleado
POST https://apiempleado.azurewebsites.net/api/empleado
GET https://adminap.azurewebsites.net/api/admin
DEL https://adminap.azurewebsites.net/api/admin/4
PUT https://adminap.azurewebsites.net/api/admin
POST https://adminap.azurewebsites.net/api/admin
DEL https://localhost:44360/api/empleado/3
PUT https://localhost:44360/api/empleado
No Environment
PUT
https://apiempleado.azurewebsites.net/api/empleado
Params
Send
Save
Authorization
Headers (1)
Body
Pre-request Script
Tests
Code
form-data
x-www-form-urlencoded
raw
binary
JSON (application/json)
{
"idEmpleado": 4,
"nombre": "Juan",
"usuario": "juan05",
"contraseña": "juan123",
"dirreccion": "Santo Domingo",
"idpais": 1
}
Response
Hit the Send button to get a response.
Do more with requests
Share
Mock
Monitor
Document

Delete


NEW
Runner
Import
Builder
Team Library
Sign In
Chrome apps are being deprecated. Download our free native apps for continued support and better performance. Learn more
Filter
History
Collections
Clear all
Today
DEL https://apiempleado.azurewebsites.net/api/empleado/6
DEL https://apiempleado.azurewebsites.net/api/empleado/6
DEL https://apiempleado.azurewebsites.net/api/empleado/2
DEL https://apiempleado.azurewebsites.net/api/empleado/4
GET https://apiempleado.azurewebsites.net/api/empleado
May 26
DEL https://apiempleado.azurewebsites.net/api/empleado/4
DEL https://apiempleado.azurewebsites.net/api/empleado
PUT https://apiempleado.azurewebsites.net/api/empleado
PUT https://apiempleado.azurewebsites.net/api/empleado
POST https://apiempleado.azurewebsites.net/api/empleado
GET https://adminap.azurewebsites.net/api/admin
DEL https://adminap.azurewebsites.net/api/admin/4
PUT https://adminap.azurewebsites.net/api/admin
POST https://adminap.azurewebsites.net/api/admin
DEL https://localhost:44360/api/empleado/3
PUT https://localhost:44360/api/empleado
No Environment
DELETE
https://apiempleado.azurewebsites.net/api/empleado/6
Params
Send
Save
Authorization
Headers (1)
Body
Pre-request Script
Tests
Code
form-data
x-www-form-urlencoded
raw
binary
JSON (application/json)
Body
Cookies
Headers (8)
Test Results
Status: 200 OK
Time: 1096 ms
Pretty
Raw
Preview
JSON
false