

NOMBRE DEL ALUMNO: ALEJANDRO MAGAÑA LOPEZ

GRADO: 4 CUATRIMESTRE

GRUPO: E

NOMBRE DE LA ASIGNATURA: APLICACIONES WEB
ORIENTADA A SERVICIOS

NOMBRE DEL PROFESOR: DIEGO IVAN MAY TUZ

ACTIVIDAD #2

“REVISION DE LA API MANGAS”

FECHA DE ENTREGA: 15/SEPTIEMBRE/2024

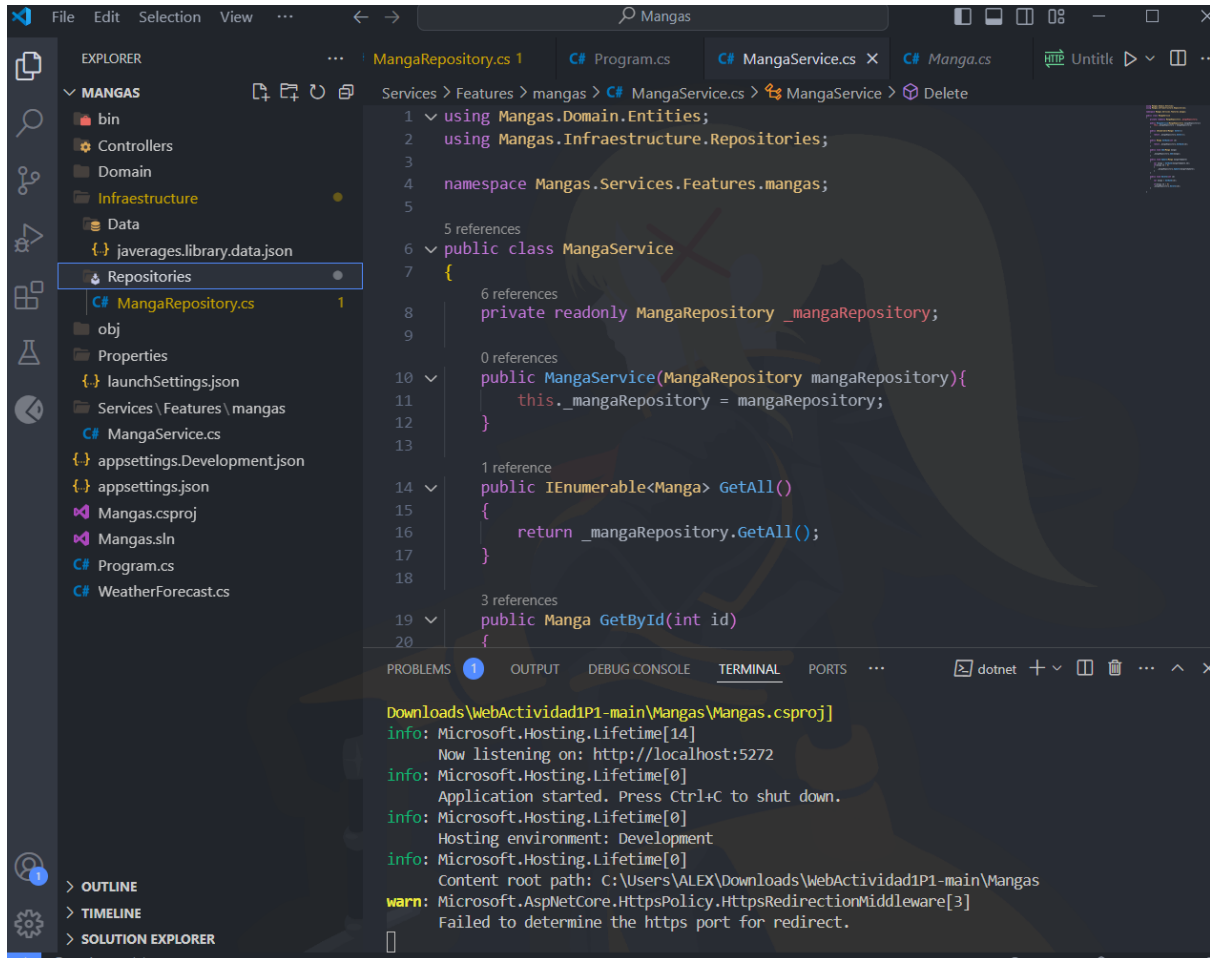
UNIVERSIDAD TEGNOLOGICA METROPOLITANA

Introducción

Como introducción en esta documentación presente demostrare que cambios note al momento de revisar y ver el código de la api que se nos dio por el profesor el cual se pueden apreciar varios cambios de la primera modificación a esta última que se realizó, desde agregar un listado completo un archivo json y la conexión de esta misma en la api realizado a continuación mostrare los cambios que note en la webapi.

Lo primero que se puede analizar o lo que se agrego fue 2 carpetas mas las cuales nos darían información de un formato json de una lista de libros completa la cual debíamos utilizar en la api y luego modificar en la clase MangaService para tener cambios con el listado de formato json

Ejemplo:



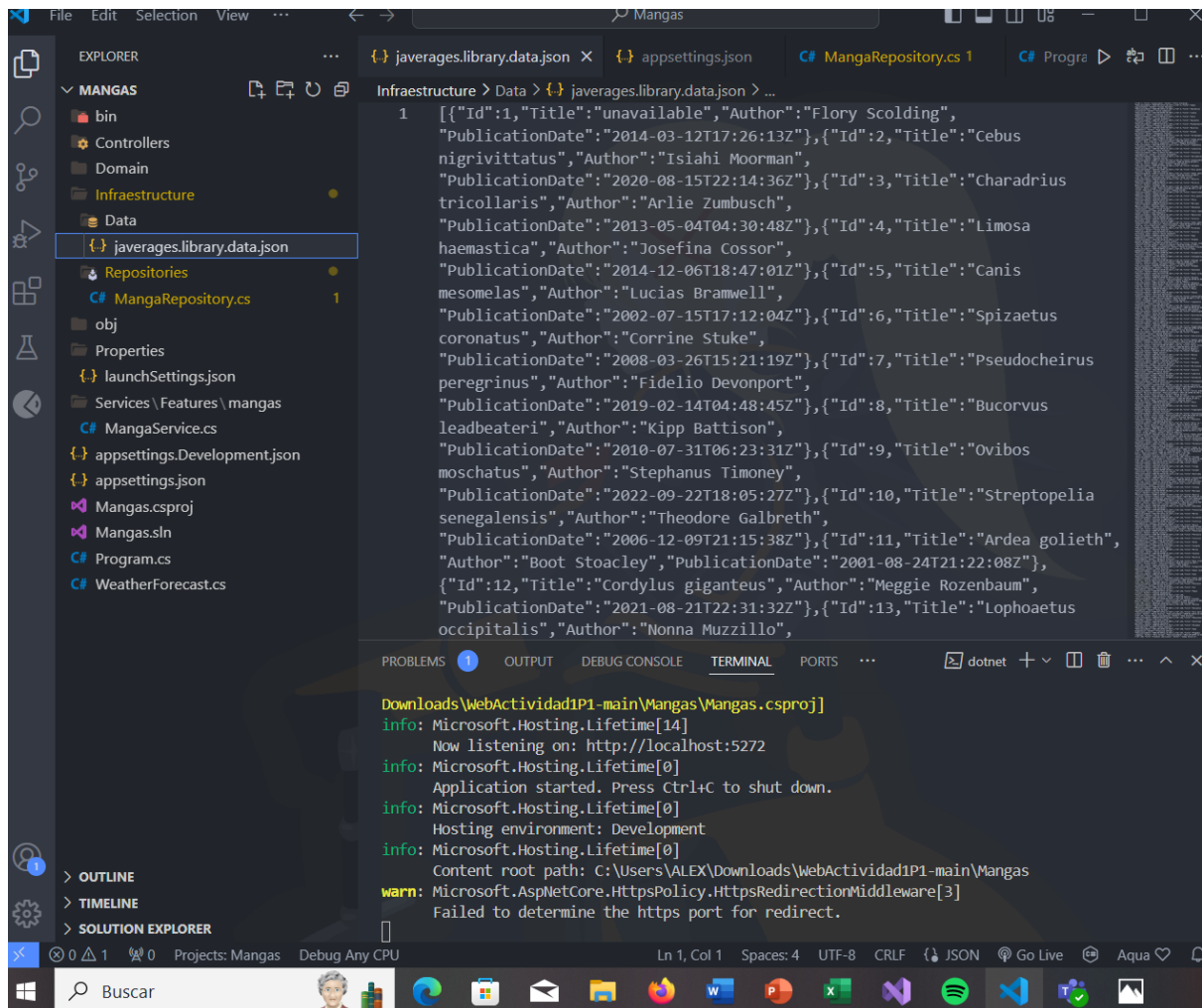
The screenshot shows the Visual Studio IDE with the following components:

- EXPLORER:** Displays the project structure. The 'Repositories' folder is selected, showing 'MangaRepository.cs'.
- Code Editor:** Displays the code for 'MangaService.cs'. The code is as follows:

```
1 using Mangas.Domain.Entities;
2 using Mangas.Infraestructure.Repositories;
3
4 namespace Mangas.Services.Features.mangas;
5
6 public class MangaService
7 {
8     private readonly MangaRepository _mangaRepository;
9
10    public MangaService(MangaRepository mangaRepository){
11        this._mangaRepository = mangaRepository;
12    }
13
14    public IEnumerable<Manga> GetAll()
15    {
16        return _mangaRepository.GetAll();
17    }
18
19    public Manga GetById(int id)
20    {
```
- TERMINAL:** Displays the output of the application. The output is as follows:

```
Downloads\WebActividad1P1-main\Mangas\Mangas.csproj
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5272
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\ALEX\Downloads\WebActividad1P1-main\Mangas
warn: Microsoft.AspNetCore.HttpPolicy.HttpsRedirectionMiddleware[3]
      Failed to determine the https port for redirect.
```

Ejemplo del formato Json:



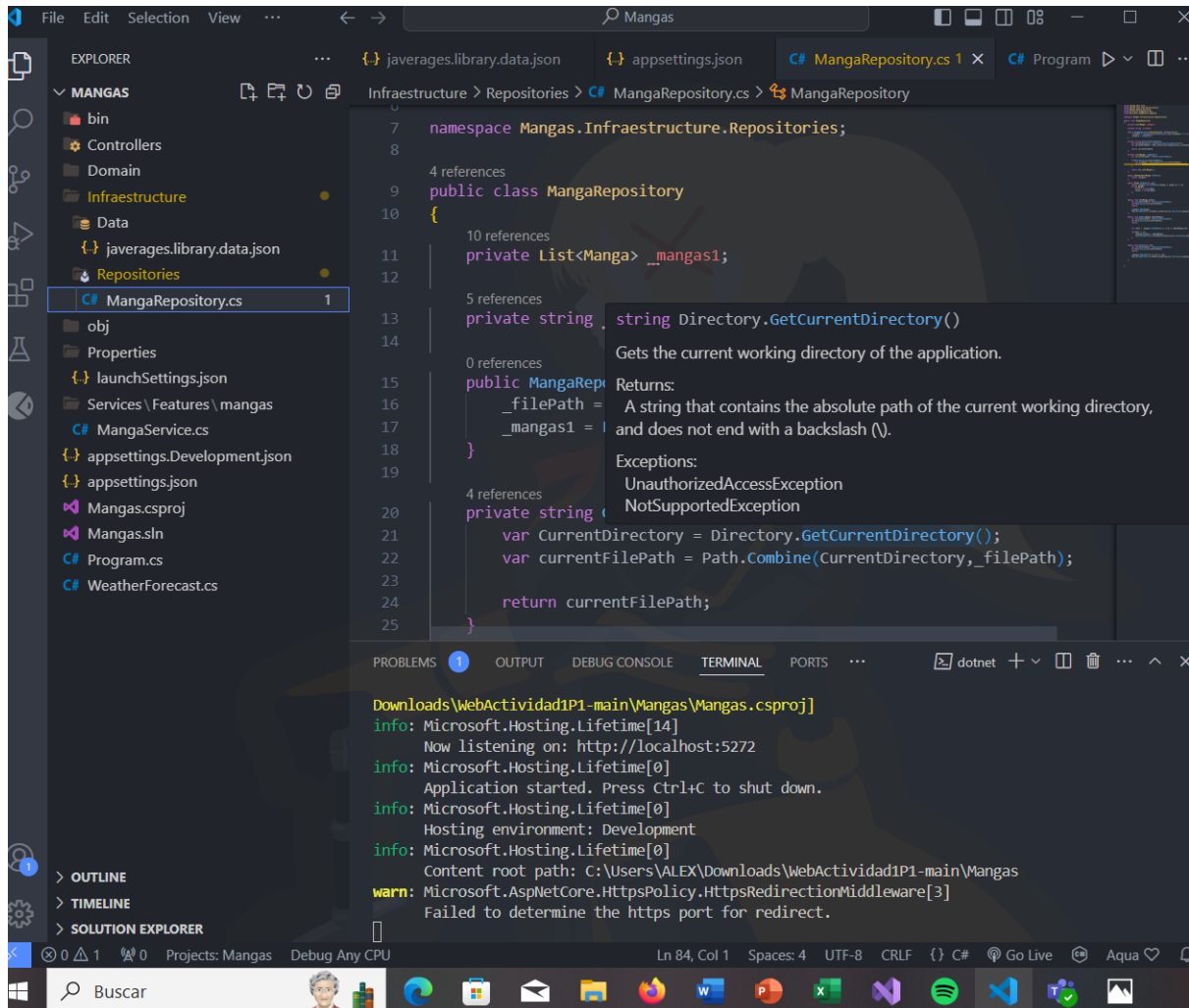
The screenshot shows the Visual Studio Code interface. The Explorer pane on the left displays the project structure for 'MANGAS', with 'javerages.library.data.json' selected under the 'Data' folder. The main editor displays the content of this JSON file, which is an array of 13 objects, each representing a book with fields for Id, Title, Author, and PublicationDate. The bottom panel shows the TERMINAL window with the output of the application, indicating it is running on http://localhost:5272.

```
1 [{"Id":1,"Title":"unavailable","Author":"Flory Scolding",
  "PublicationDate":"2014-03-12T17:26:13Z"}, {"Id":2,"Title":"Cebus
  nigrovittatus","Author":"Isiahi Moorman",
  "PublicationDate":"2020-08-15T22:14:36Z"}, {"Id":3,"Title":"Charadrius
  tricollaris","Author":"Arlie Zumbusch",
  "PublicationDate":"2013-05-04T04:30:48Z"}, {"Id":4,"Title":"Limosa
  haemastica","Author":"Josefina Cossor",
  "PublicationDate":"2014-12-06T18:47:01Z"}, {"Id":5,"Title":"Canis
  mesomelas","Author":"Lucias Bramwell",
  "PublicationDate":"2002-07-15T17:12:04Z"}, {"Id":6,"Title":"Spizaetus
  coronatus","Author":"Corrine Stuke",
  "PublicationDate":"2008-03-26T15:21:19Z"}, {"Id":7,"Title":"Pseudocheirus
  peregrinus","Author":"Fidelio Devonport",
  "PublicationDate":"2019-02-14T04:48:45Z"}, {"Id":8,"Title":"Bucorvus
  leadbeateri","Author":"Kipp Battison",
  "PublicationDate":"2010-07-31T06:23:31Z"}, {"Id":9,"Title":"Ovibos
  moschatus","Author":"Stephanus Timoney",
  "PublicationDate":"2022-09-22T18:05:27Z"}, {"Id":10,"Title":"Streptopelia
  senegalensis","Author":"Theodore Galbreth",
  "PublicationDate":"2006-12-09T21:15:38Z"}, {"Id":11,"Title":"Ardea golieth",
  "Author":"Boot Stoacley","PublicationDate":"2001-08-24T21:22:08Z"},
  {"Id":12,"Title":"Cordylus giganteus","Author":"Meggie Rozenbaum",
  "PublicationDate":"2021-08-21T22:31:32Z"}, {"Id":13,"Title":"Lophoaetus
  occipitalis","Author":"Nonna Muzzillo",
```

```
Downloads\WebActividad1P1-main\Mangas\Mangas.csproj
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5272
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\ALEX\Downloads\WebActividad1P1-main\Mangas
warn: Microsoft.AspNetCore.HttpsPolicy.HttpsRedirectionMiddleware[3]
      Failed to determine the https port for redirect.
```

Ejemplo de una clase llamada MangaRepositories , la cual como dice en los pasos del profesor

Este iba a servir para llamar a los datos de el archivo json que el profe nos dio, por medio de métodos que iban a funcionar luego en la clase MangaServivce, y para eso este es el ejemplo que se creo con los métodos correspondientes y la llamada del archivo json

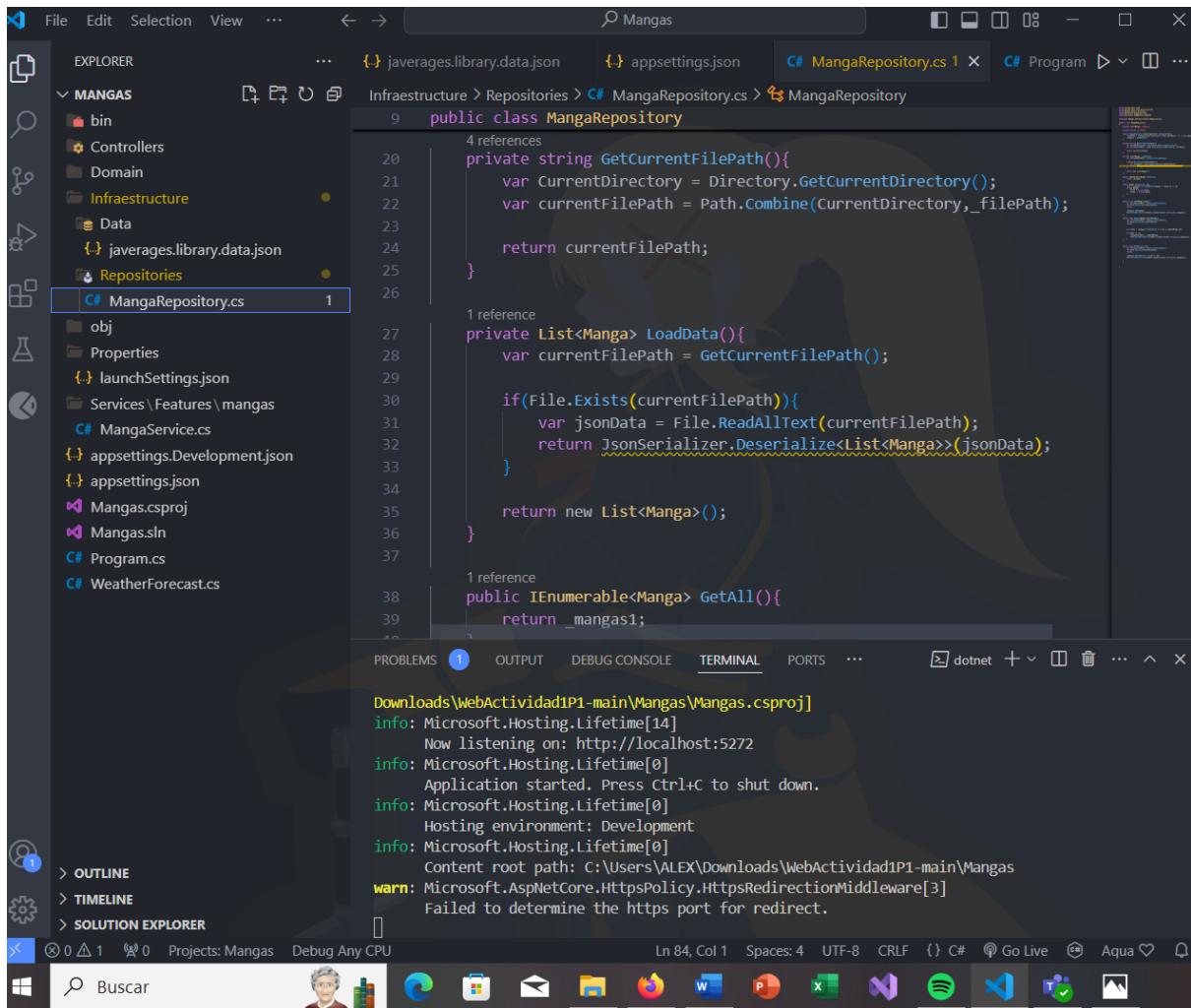


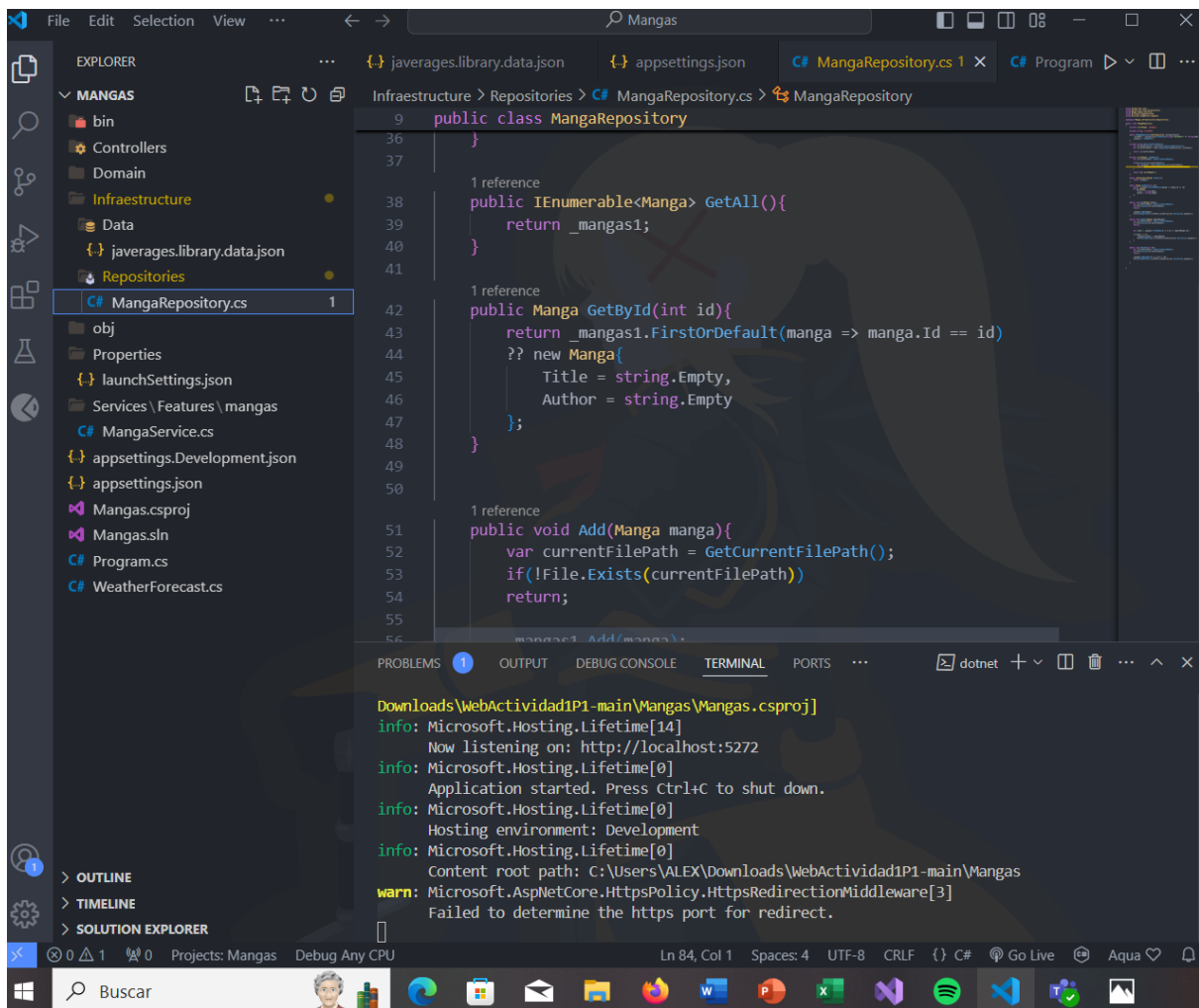
The screenshot shows the Visual Studio IDE with the `MangaRepository.cs` file open. The file is located in the `Infrastructure > Repositories` folder. The code defines a `MangaRepository` class with a `private List<Manga> _mangas1` and a `private string _filePath`. The `GetCurrentDirectory()` method is implemented to return the current directory path. A tooltip for `Directory.GetCurrentDirectory()` is visible, showing its return type and exceptions.

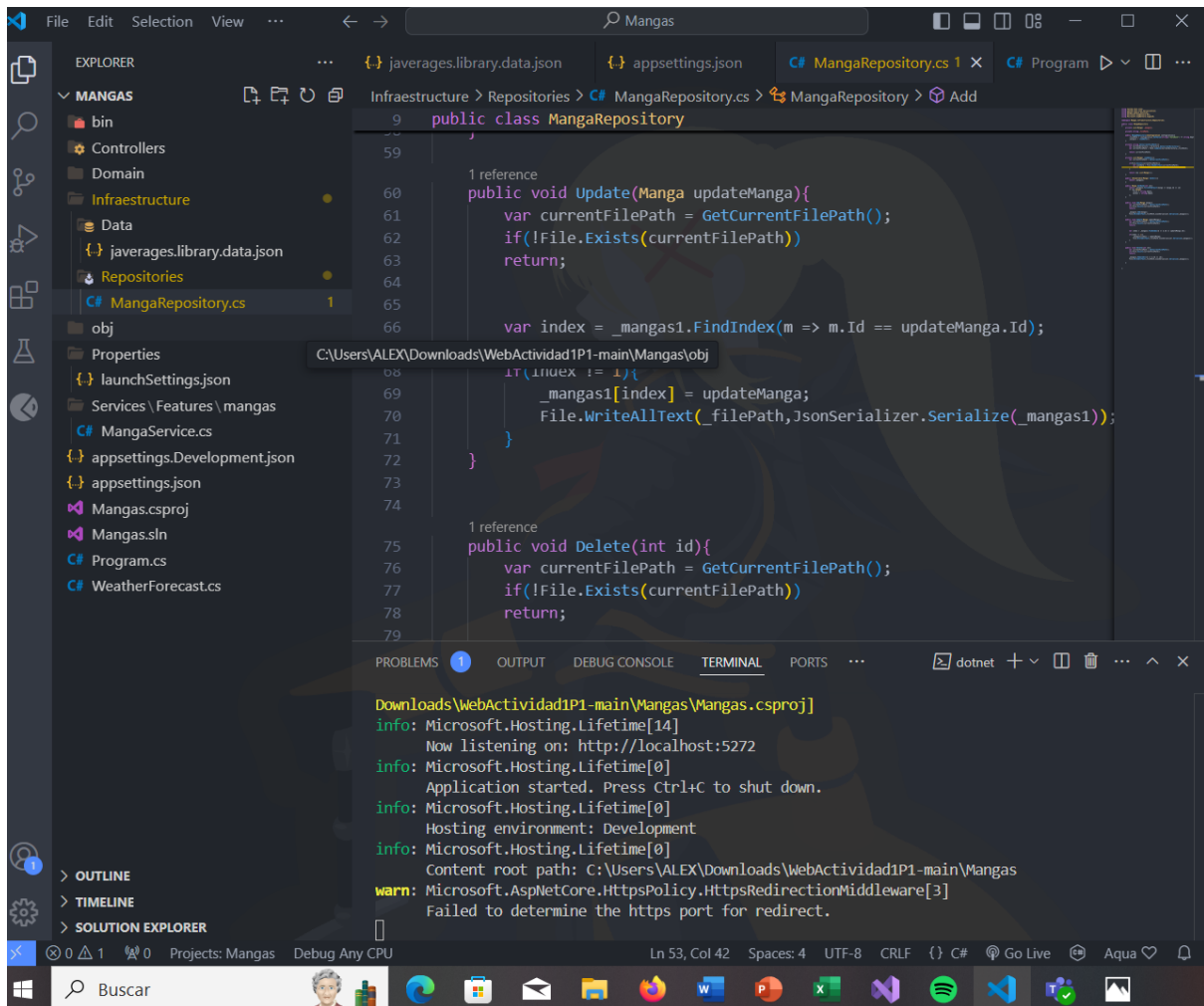
```
7 namespace Mangas.Infrastructure.Repositories;
8
9 public class MangaRepository
10 {
11     private List<Manga> _mangas1;
12
13     private string _filePath;
14
15     public MangaRepository()
16     {
17         _filePath = Path.Combine(Directory.GetCurrentDirectory(), "Data", "javerages.library.data.json");
18     }
19
20     private string GetCurrentDirectory()
21     {
22         var CurrentDirectory = Directory.GetCurrentDirectory();
23         var currentFilePath = Path.Combine(CurrentDirectory, _filePath);
24
25         return currentFilePath;
26     }
27 }
```

The terminal output shows the application starting and listening on `http://localhost:5272`. It also displays a warning from `Microsoft.AspNetCore.HttpPolicy.HttpsRedirectionMiddleware` about failing to determine the https port for redirect.

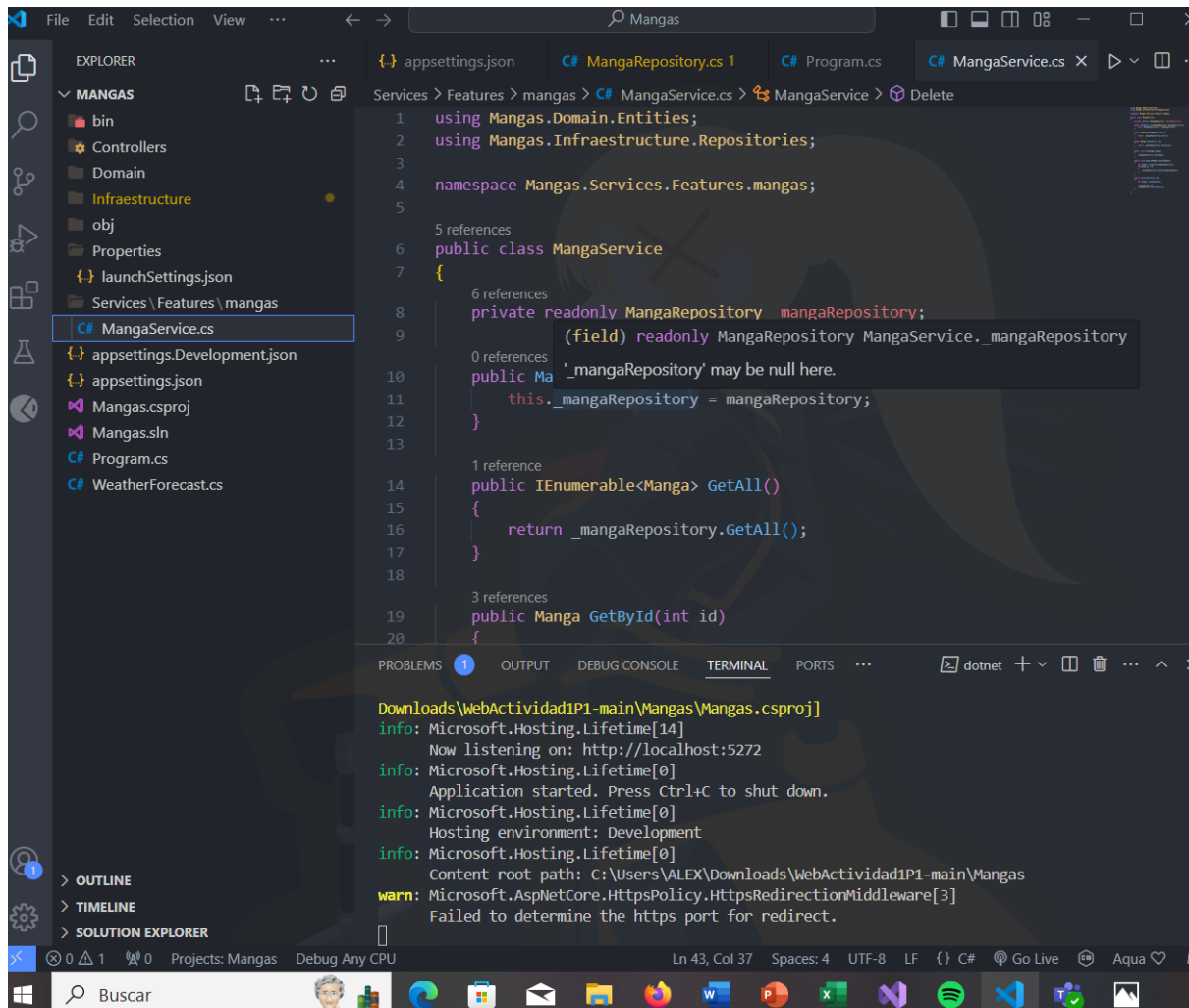
```
Downloads\WebActividad1P1-main\Mangas\Mangas.csproj]
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5272
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\ALEX\Downloads\WebActividad1P1-main\Mangas
warn: Microsoft.AspNetCore.HttpPolicy.HttpsRedirectionMiddleware[3]
      Failed to determine the https port for redirect.
```



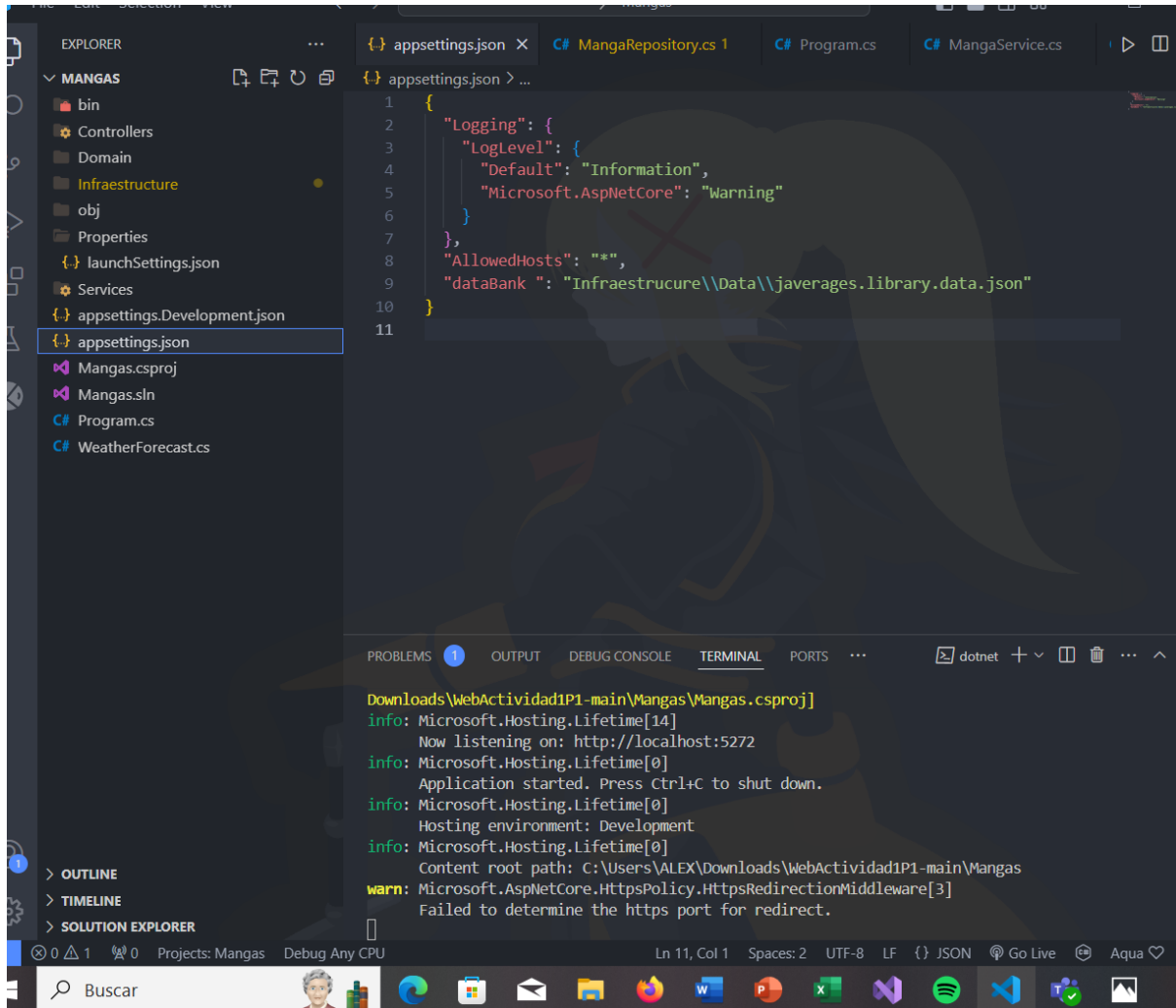




Luego de esto ya en clase MangaService ya no era necesario, la variable `_manga1` que había creado por que como tal ya iba a usar la clase `mangaRepositories`, a partir de la instanciación de esa clase y luego obtener los métodos, este debía ser funcional.



Este paso es casi similar a cuando conectas una base de datos en una api es casi el mismo procedimiento ya que en el appsettings.json se hace la conexión con una clase llamada dbcontext , pero en esta ocasión no estamos usando una base de datos como tal si no que estamos usando un archivo json y por ende me imagino que es para que se obtenga la información de todo ese archivo json por medio de la ruta que le pasamos



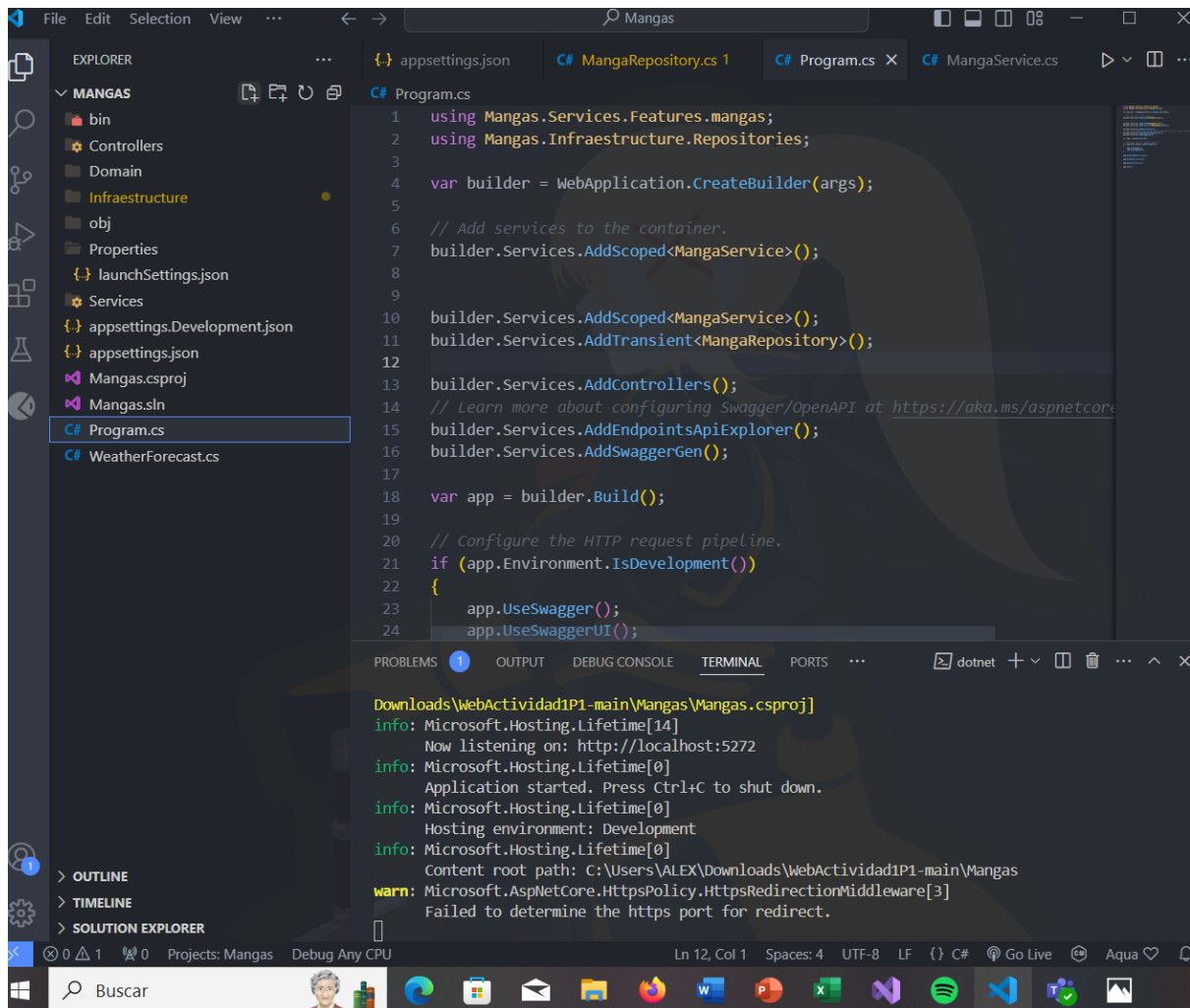
The screenshot shows the Visual Studio IDE with the following components:

- EXPLORER:** The file explorer on the left shows the project structure for 'MANGAS'. The file 'appsettings.json' is selected.
- Code Editor:** The main editor displays the content of 'appsettings.json':

```
1 {
2   "Logging": {
3     "LogLevel": {
4       "Default": "Information",
5       "Microsoft.AspNetCore": "Warning"
6     }
7   },
8   "AllowedHosts": "*",
9   "dataBank ": "Infraestructure\\Data\\javerages.library.data.json"
10 }
11
```
- TERMINAL:** The terminal at the bottom shows the output of the application startup:

```
Downloads\WebActividad1P1-main\Mangas\Mangas.csproj]
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5272
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\ALEX\Downloads\WebActividad1P1-main\Mangas
warn: Microsoft.AspNetCore.HttpPolicy.HttpsRedirectionMiddleware[3]
      Failed to determine the https port for redirect.
```
- STATUS BAR:** The bottom status bar indicates 'Ln 11, Col 1', 'Spaces: 2', 'UTF-8', 'LF', and the file type is 'JSON'.

Bueno y ya para finalizar ahora solo agrego el registro de la clase que se uso en este caso repositories



The screenshot shows the Visual Studio IDE with the 'Mangas' project open. The Explorer pane on the left shows the project structure, including folders like bin, Controllers, Domain, Infrastructure, obj, Properties, and Services. The Program.cs file is selected and open in the editor. The code in Program.cs registers the MangaRepository as a transient service and adds MangaService as a scoped service. The terminal at the bottom shows the application running successfully on http://localhost:5272.

```
1 using Mangas.Services.Features.mangas;
2 using Mangas.Infrastructure.Repositories;
3
4 var builder = WebApplication.CreateBuilder(args);
5
6 // Add services to the container.
7 builder.Services.AddScoped<MangaService>();
8
9
10 builder.Services.AddScoped<MangaService>();
11 builder.Services.AddTransient<MangaRepository>();
12
13 builder.Services.AddControllers();
14 // Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore
15 builder.Services.AddEndpointsApiExplorer();
16 builder.Services.AddSwaggerGen();
17
18 var app = builder.Build();
19
20 // Configure the HTTP request pipeline.
21 if (app.Environment.IsDevelopment())
22 {
23     app.UseSwagger();
24     app.UseSwaggerUI();
25 }
```

Terminal Output:

```
Downloads\WebActividad1P1-main\Mangas\Mangas.csproj]
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5272
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\ALEX\Downloads\WebActividad1P1-main\Mangas
warn: Microsoft.AspNetCore.HttpPolicy.HttpsRedirectionMiddleware[3]
      Failed to determine the https port for redirect.
```

CONCLUSION

Bueno ya para concluir con este documento, me siento muy satisfecho con la practica ya que en base a esto pude observar como funciona una api en todos los sentidos desde la primera actividad donde agregamos controladores para que la api tenga algo que que hacer, donde están los atributos de la clase Manga como libro, título, etc métodos y protocolos http que realizar en la api (get,post,put y delete) y como este se crea fue muy interesante y ya en este actividad unimos un archivo json lo cual fue mas interesante para mi por que ya tenemos una gran cantidad de registros de ese archivo json y como se llaman por medio de instancia, como se conectan como funciona en la misma api es algo impresionante asi que la actividad fue algo de ayuda para mi en fin esta seria la conclusión de la actividad .