

Universidad Tecnológica de la Región Norte de Guerrero

**“Ingeniería en Desarrollo y Gestión de
Software”**

Desarrollo para Dispositivos Inteligentes

Actividad:

Examen 1er Parcial

Maestro:

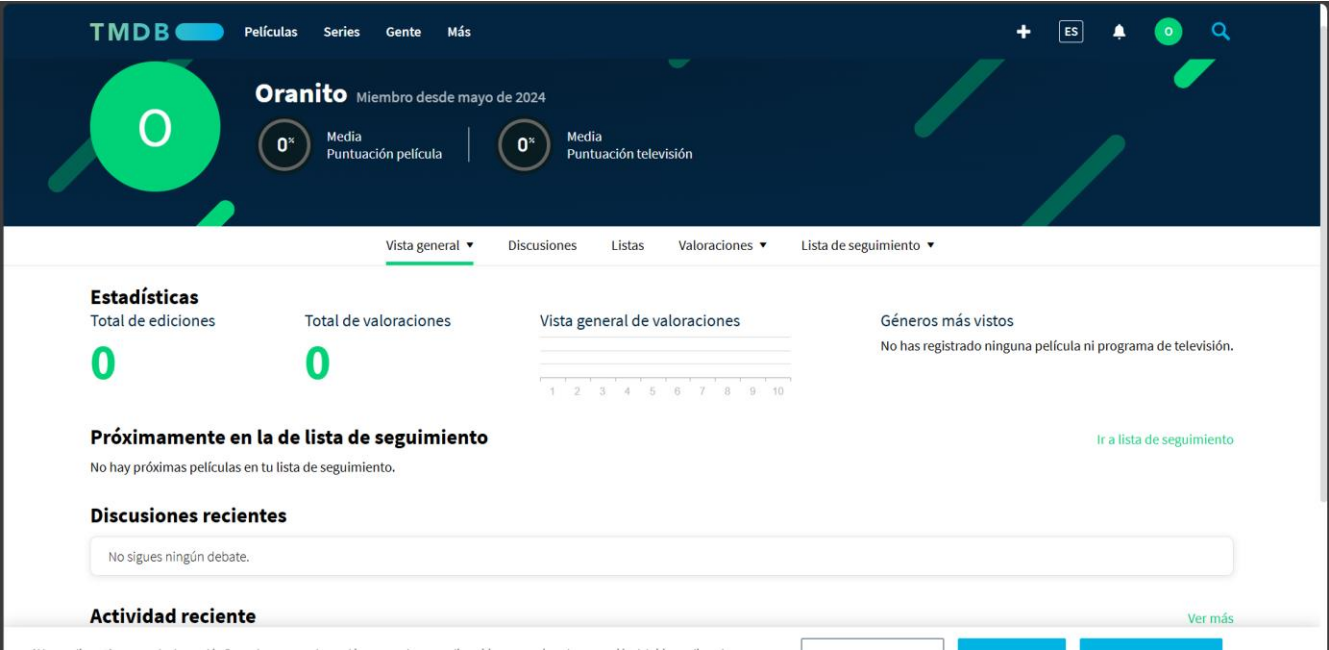
M.T.I. Roberto Estrada Benavides

Alumno:

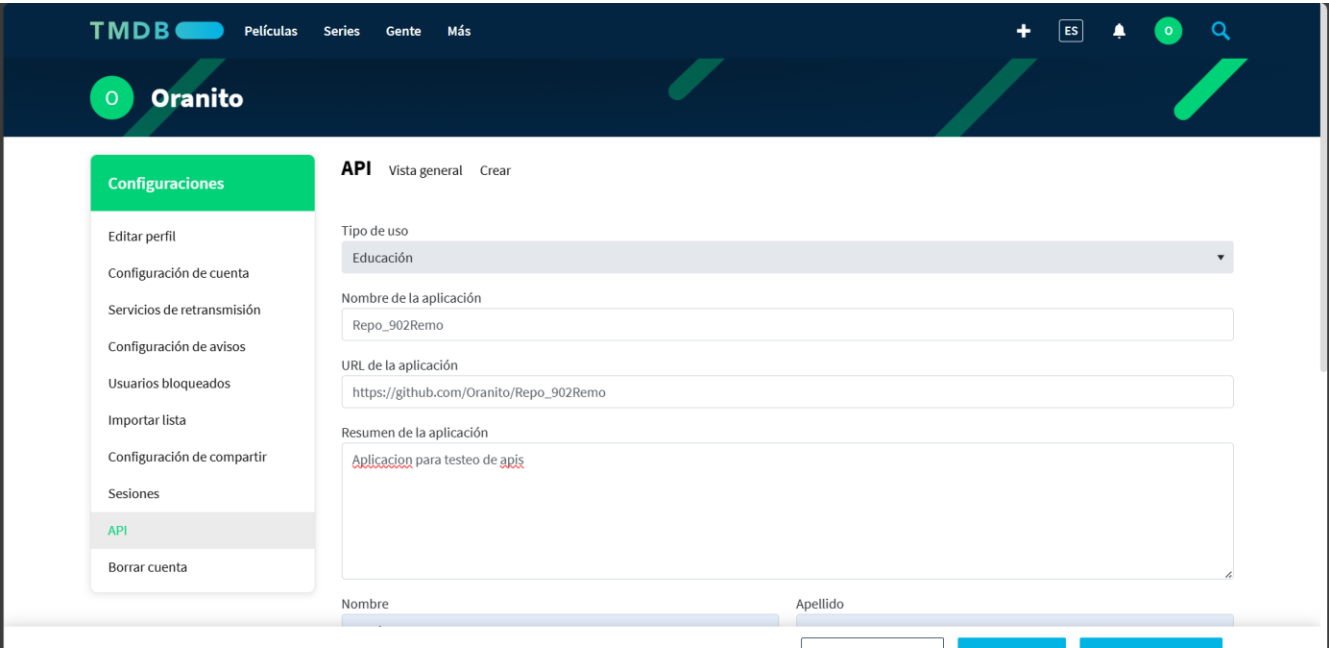
- **Sergio Serrano Cuevas**
- **Alejandra Gaspar Ramírez**



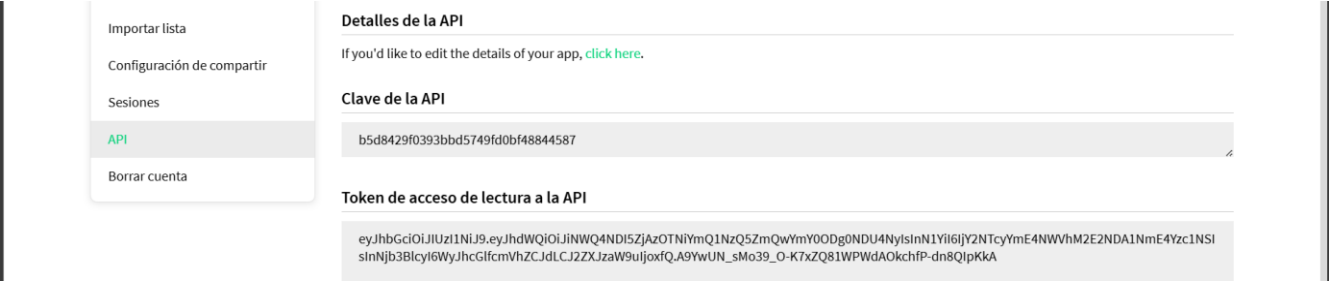
El primer paso es registrarnos en la página TMDB



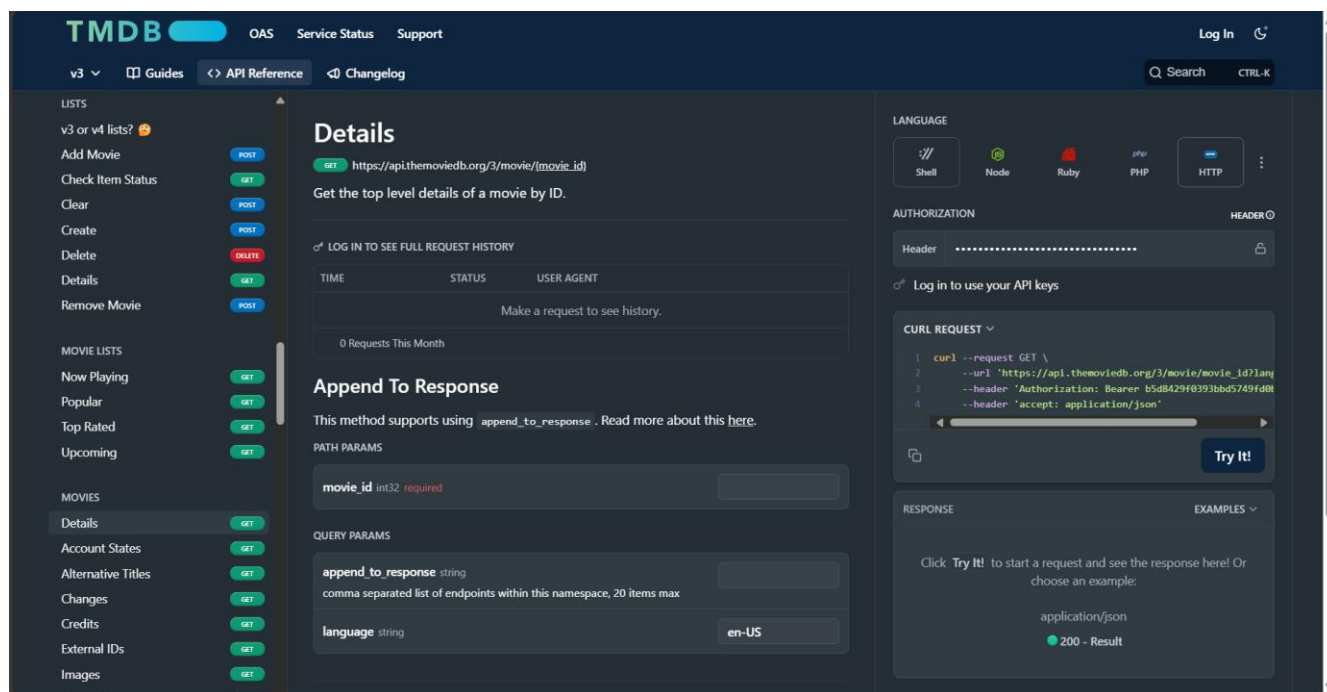
Después vamos a solicitar un api personal y de uso educativo



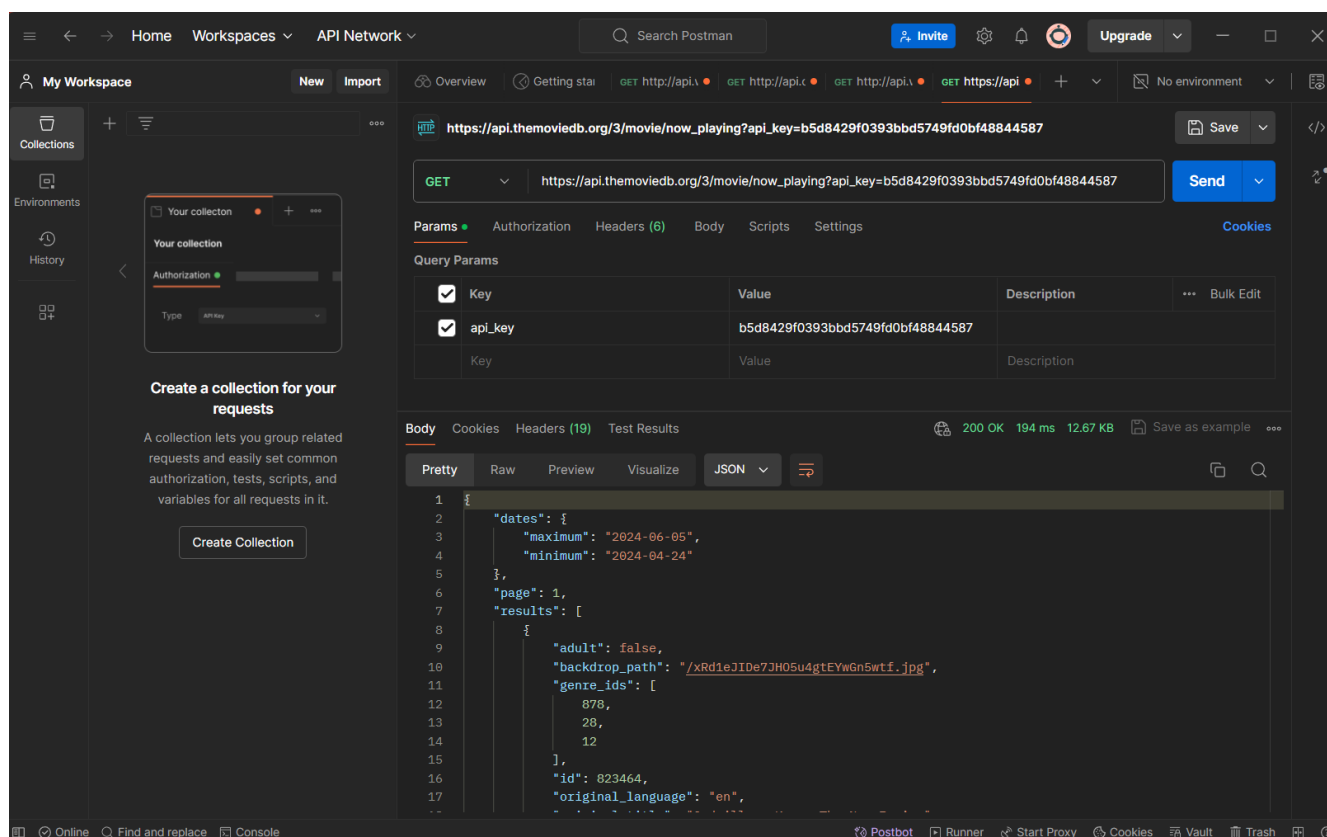
Ahora ya podemos ver nuestra llave de acceso a la aplicación.



Ahora vamos a realizar una llamada en postman para comprobar si nos da acceso en la petición

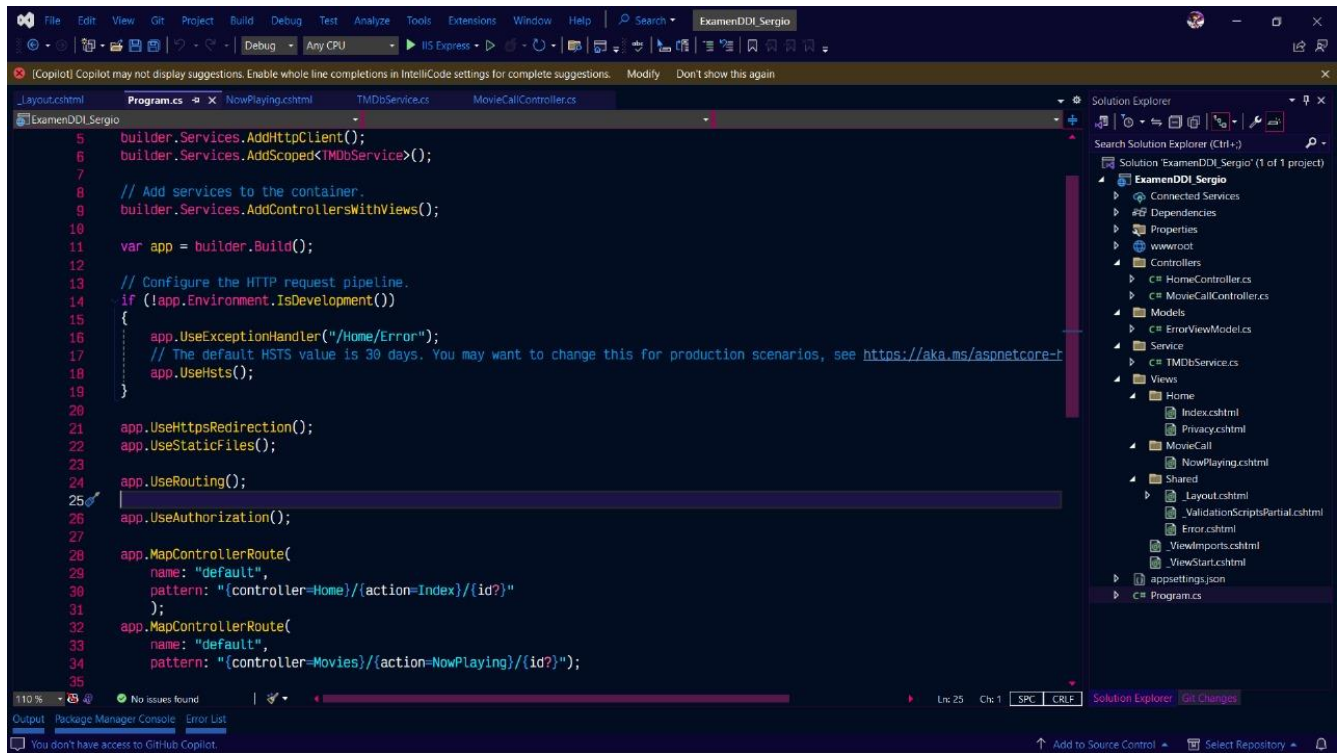


Comprobamos que la api_key funcione mediante una petición en postman



Ahora vamos a crear una aplicación Asp.Net Core Web App (MVC).

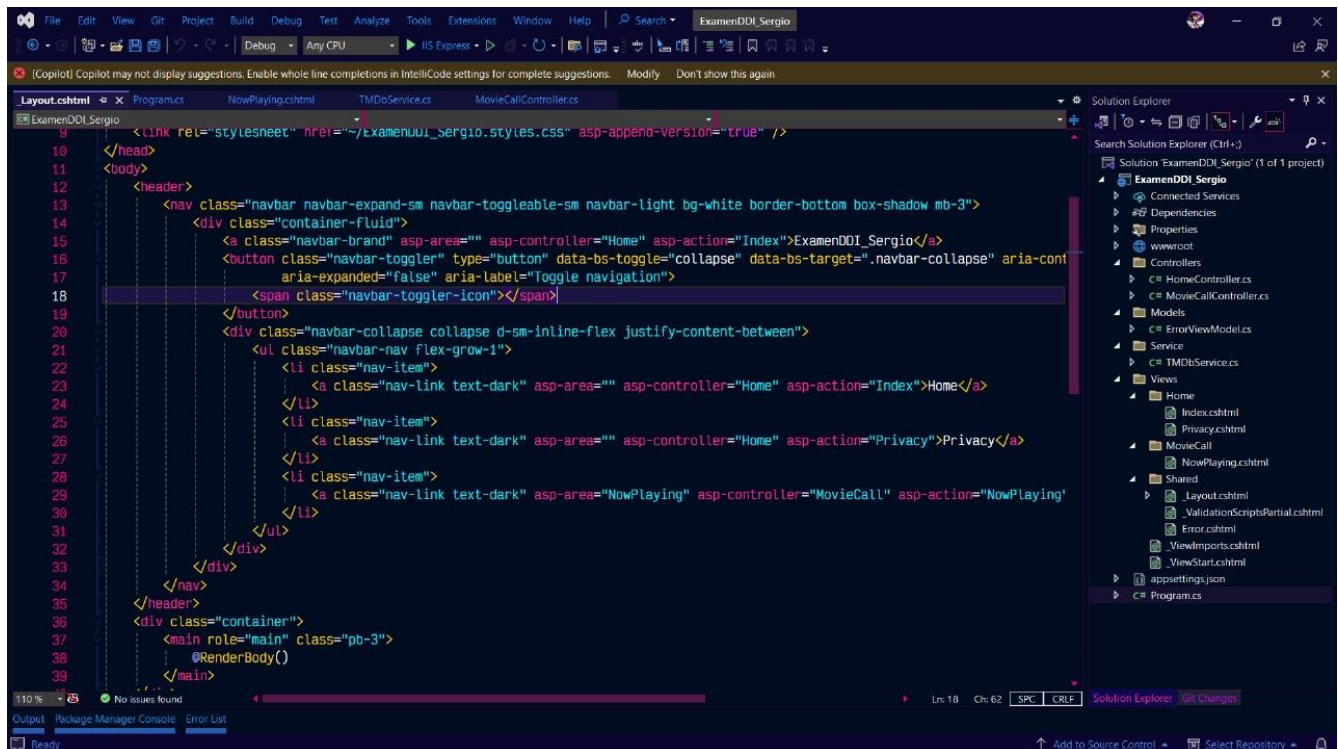
Una vez creada la aplicación se agregará el servicio http y scoped, además de enrutar el controlador en la clase Program.cs



The screenshot shows the Visual Studio IDE with the 'Program.cs' file open. The code configures the ASP.NET Core application. The 'builder.Services' section adds an HttpClient and a scoped service 'TmdbService'. The 'app' variable is built and configured with exception handling, HSTS, HTTPS redirection, static files, routing, and authorization. The routing section maps two controller routes: a default route for the 'Home' controller's 'Index' action, and another default route for the 'Movies' controller's 'NowPlaying' action. The Solution Explorer on the right shows the project structure, including controllers, models, services, and views.

```
5 builder.Services.AddHttpClient();
6 builder.Services.AddScoped<TmdbService>();
7
8 // Add services to the container.
9 builder.Services.AddControllersWithViews();
10
11 var app = builder.Build();
12
13 // Configure the HTTP request pipeline.
14 if (app.Environment.IsDevelopment())
15 {
16     app.UseExceptionHandler("/Home/Error");
17     // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts
18     app.UseHsts();
19 }
20
21 app.UseHttpsRedirection();
22 app.UseStaticFiles();
23
24 app.UseRouting();
25
26 app.UseAuthorization();
27
28 app.MapControllerRoute(
29     name: "default",
30     pattern: "{controller=Home}/{action=Index}/{id?}");
31
32 app.MapControllerRoute(
33     name: "default",
34     pattern: "{controller=Movies}/{action=NowPlaying}/{id?}");
35
```

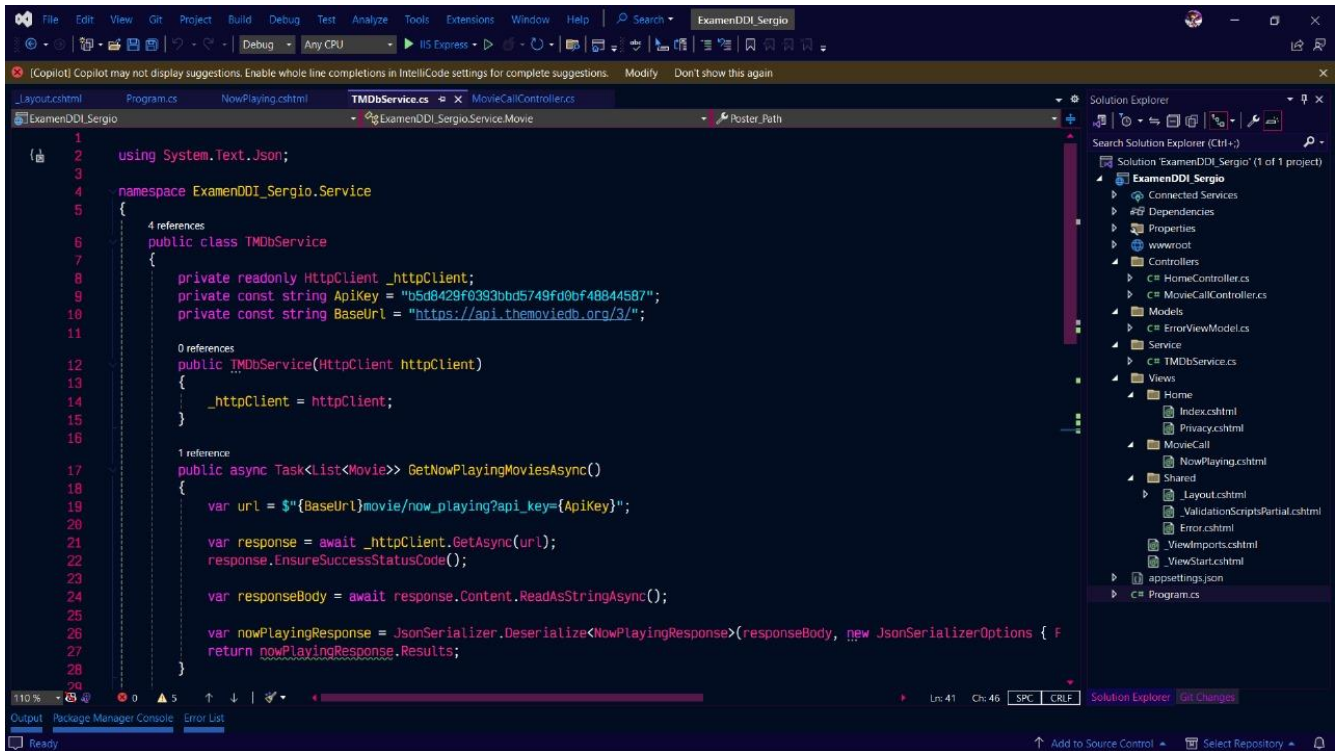
Se agrega al navbar (opcional)



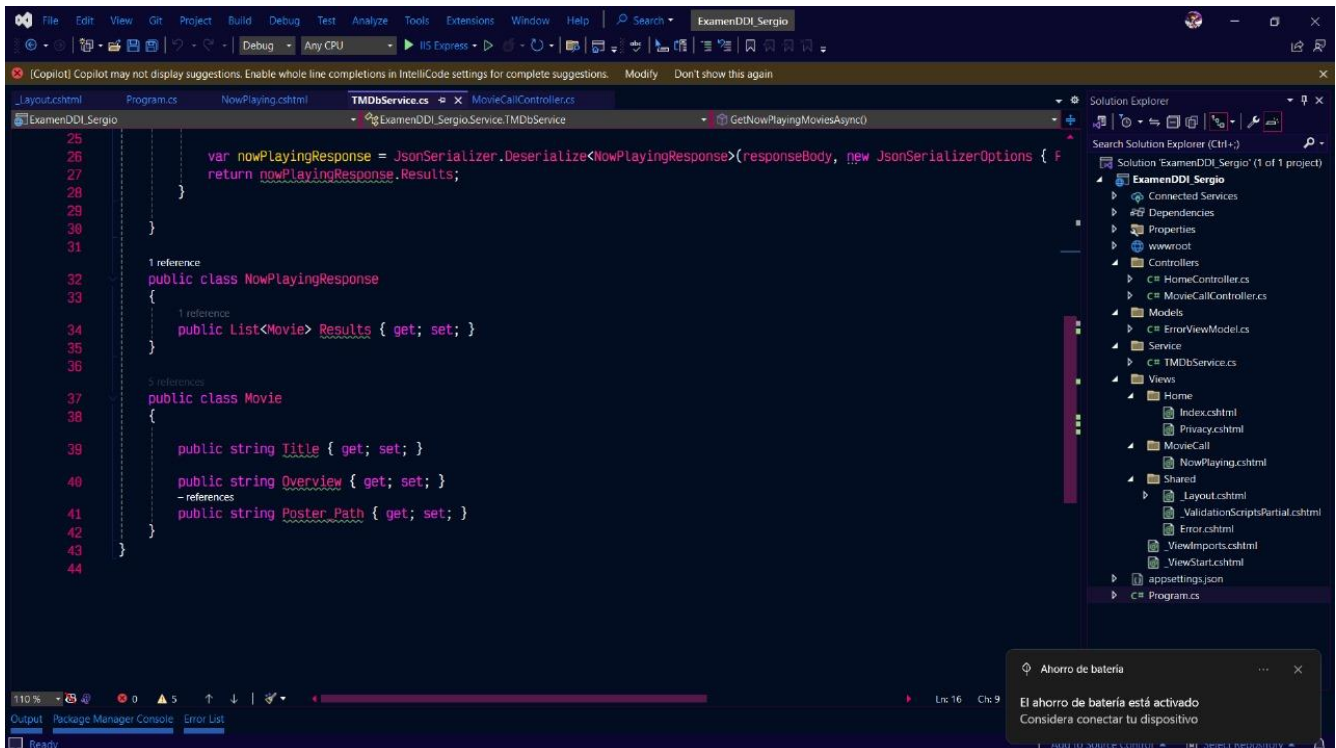
The screenshot shows the Visual Studio IDE with the 'Layout.cshtml' file open. The code defines the layout for the application, including a link to the stylesheets, a header section with a navigation bar, and a main content area. The navigation bar is a Bootstrap navbar with a toggle button and links for 'Home', 'Privacy', and 'NowPlaying'. The 'NowPlaying' link is highlighted. The main content area is rendered using the '@RenderBody()' method. The Solution Explorer on the right shows the project structure, including controllers, models, services, and views.

```
9 <link rel="stylesheet" href="/ExamenDDI_Sergio.styles.css" asp-append-version="true" />
10 </head>
11 <body>
12 <header>
13 <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">
14 <div class="container-fluid">
15 <a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">ExamenDDI_Sergio</a>
16 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse" aria-expanded="false" aria-label="Toggle navigation">
17 <span class="navbar-toggler-icon"></span>
18 </button>
19 <div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
20 <ul class="navbar-nav flex-grow-1">
21 <li class="nav-item">
22 <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>
23 </li>
24 <li class="nav-item">
25 <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>
26 </li>
27 <li class="nav-item">
28 <a class="nav-link text-dark" asp-area="NowPlaying" asp-controller="MovieCall" asp-action="NowPlaying">NowPlaying</a>
29 </li>
30 </ul>
31 </div>
32 </div>
33 </header>
34 <div class="container">
35 <main role="main" class="pb-3">
36 @RenderBody()
37 </main>
38 </div>
39
```


Se agrega una clase con los modelos y servicios a usar

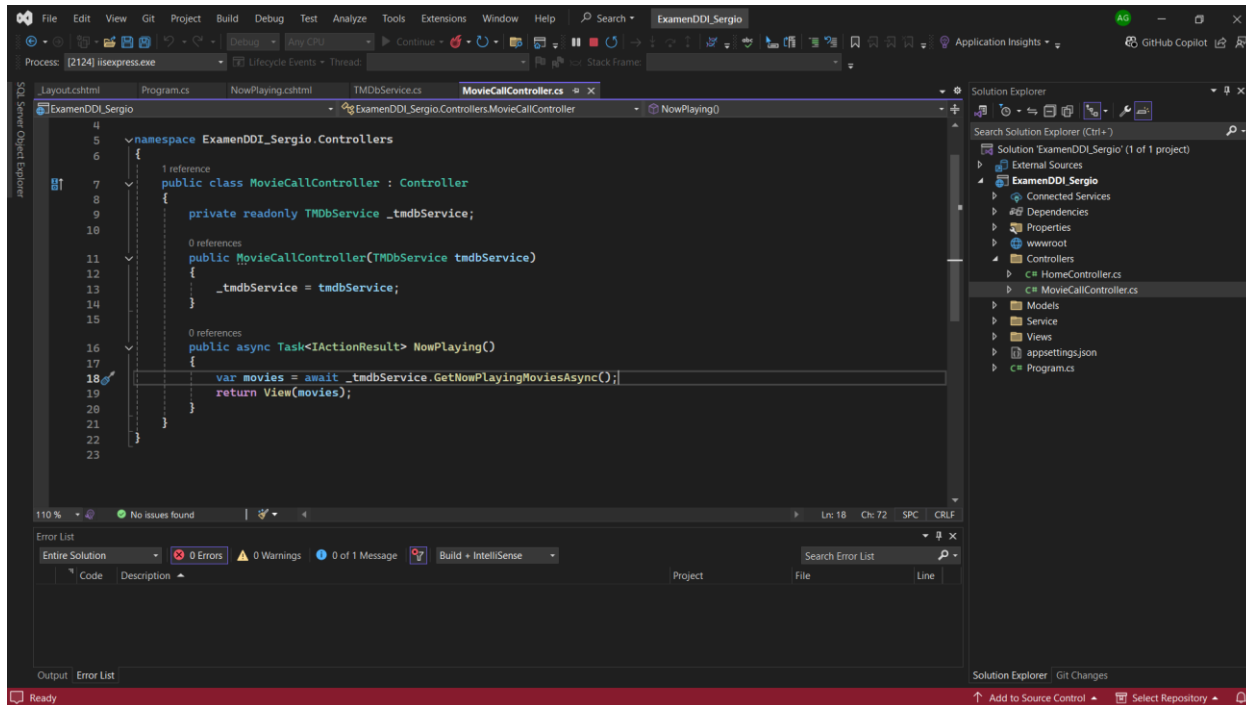


```
1 using System.Text.Json;
2
3 namespace ExamDDI_Sergio.Service
4 {
5     4 references
6     {
7         public class TmdbService
8         {
9             private readonly HttpClient _httpClient;
10            private const string ApiKey = "b5d8429f0393bbd5749fd06f48844587";
11            private const string BaseUrl = "https://api.themoviedb.org/3/";
12
13            0 references
14            public TmdbService(HttpClient httpClient)
15            {
16                _httpClient = httpClient;
17            }
18
19            1 reference
20            public async Task<List<Movie>> GetNowPlayingMoviesAsync()
21            {
22                var url = $"{BaseUrl}movie/now_playing?api_key={ApiKey}";
23
24                var response = await _httpClient.GetAsync(url);
25                response.EnsureSuccessStatusCode();
26
27                var responseBody = await response.Content.ReadAsStringAsync();
28
29                var nowPlayingResponse = JsonSerializer.Deserialize<NowPlayingResponse>(responseBody, new JsonSerializerOptions { F
30            }
31        }
32    }
33 }
```

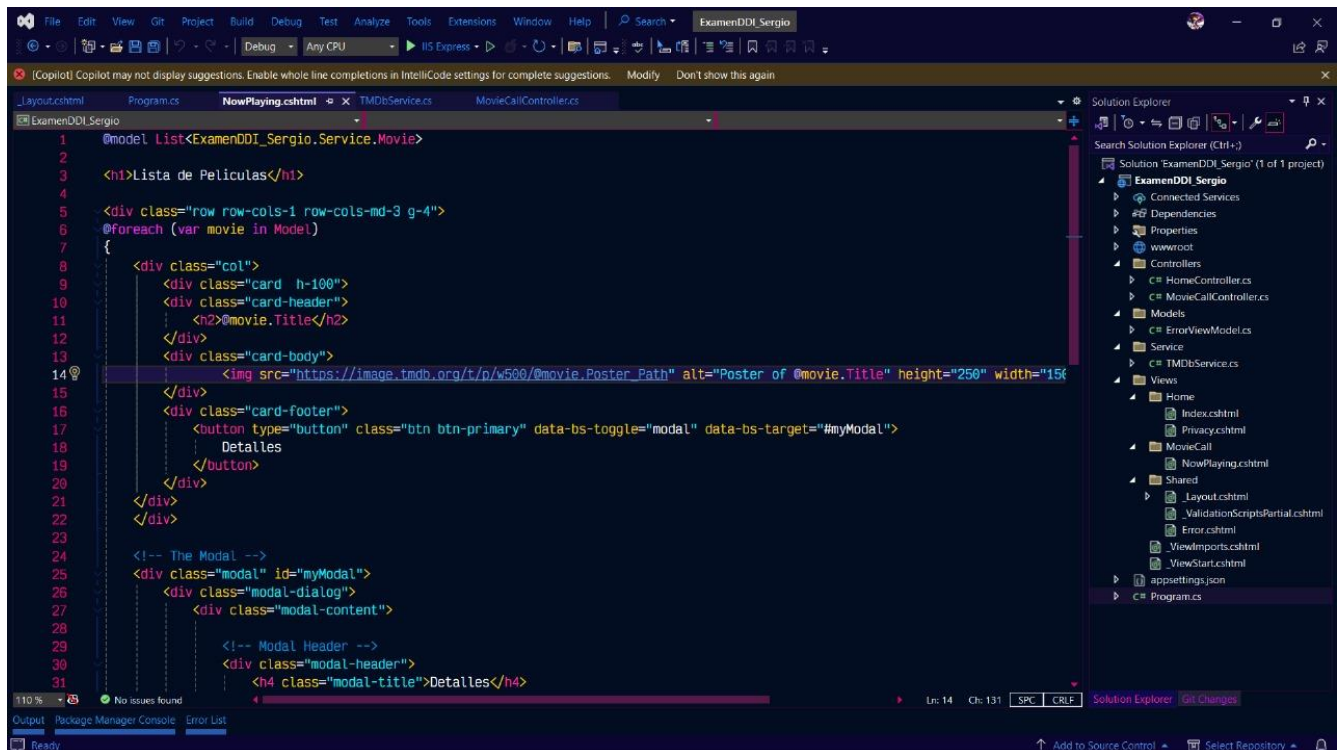


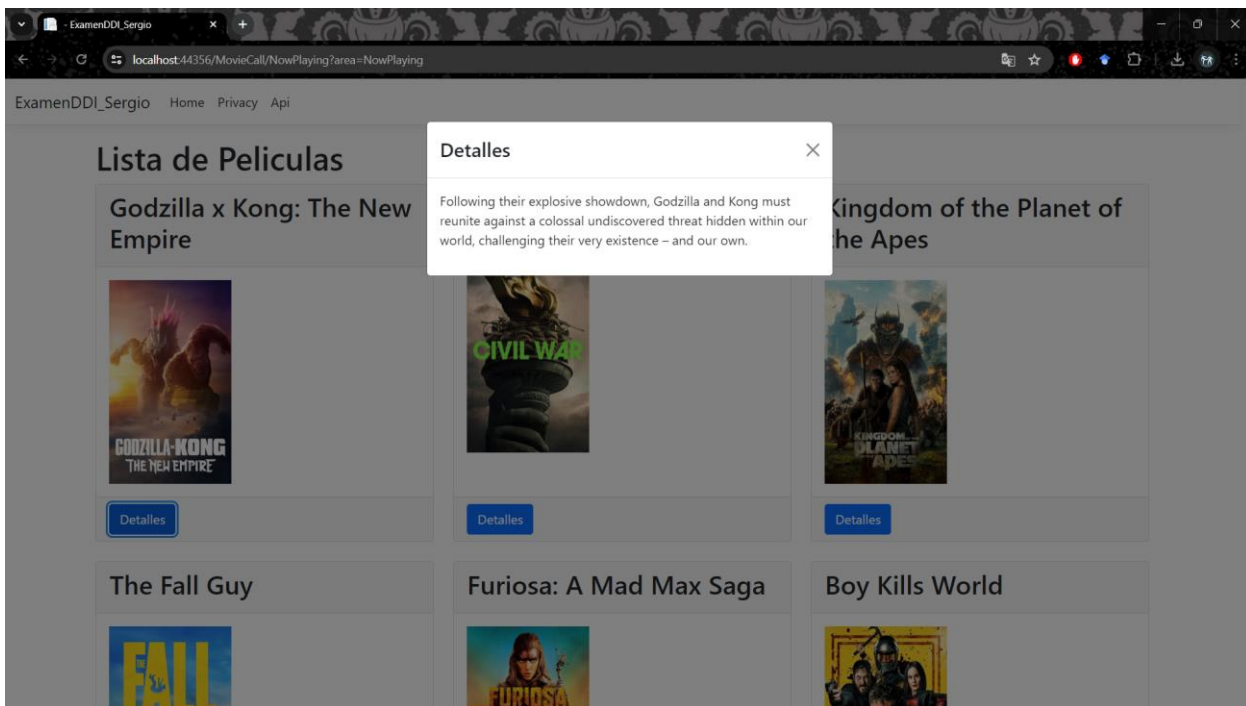
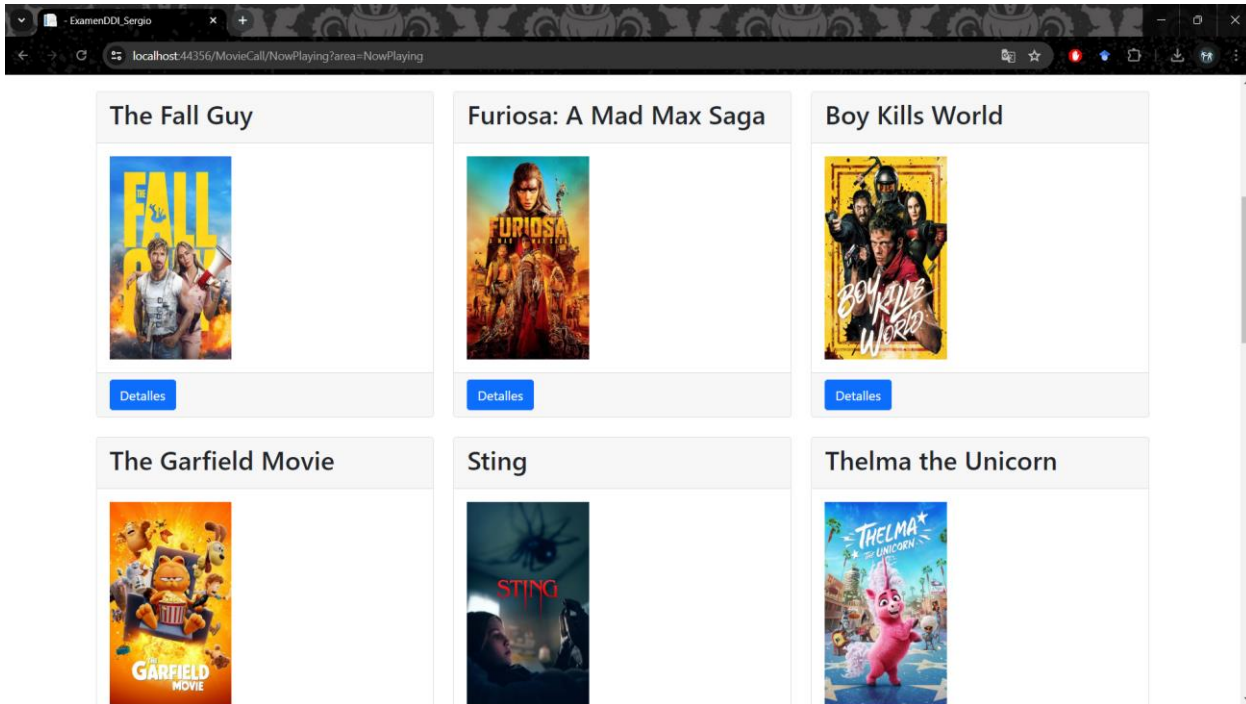
```
25
26 var nowPlayingResponse = JsonSerializer.Deserialize<NowPlayingResponse>(responseBody, new JsonSerializerOptions { F
27 return nowPlayingResponse.Results;
28 }
29
30
31
32 1 reference
33 {
34     1 reference
35     public List<Movie> Results { get; set; }
36 }
37
38 5 references
39 {
40     public class Movie
41     {
42         public string Title { get; set; }
43         public string Overview { get; set; }
44         ~ references
45         public string PosterPath { get; set; }
46     }
47 }
```

Posteriormente se creará el controlador



Una vez terminado el controlador, se creará la vista para mostrar la lista de películas






ExamenDDI_Sergio

HomePrivacyApi


Lista de Peliculas

Godzilla x Kong: The New Empire




Detalles

Civil War




Detalles

Kingdom of the Planet of the Apes




Detalles

The Fall Guy




Detalles

Furiosa: A Mad Max Saga



Detalles

Boy Kills World



Detalles

Detalles

Following their explosive showdown, Godzilla and Kong must reunite against a colossal undiscovered threat hidden within our world, challenging their very existence – and our own.