

# Informe de Entrega Sprint III

## Equipo Hex Data

### Carlos Zabala – Administrador de Configuración

El presente Pdf es para documentar los pasos que se realizaron para la creación y puesta en funcionamiento del proyecto de películas del Sprint III. El elemento importante de éste proyecto era poder aplicar un servicio inyectado a la página web, para no tener que cambiarla cada que la información varíe sino poder hacerlo desde el servicio que se está invocando.

#### 1. Creación del Proyecto HexDataMovies:

```
drwxr-xr-x 5 saint saint 4096 sep 21 06:59 PecesProy
-rw-r--r-- 1 saint saint 16640 sep 24 16:46 'Películas para SprintIII.ods'
-rw-r--r-- 1 saint saint 2686 ene 12 2016 proj1.html
-rw-r--r-- 1 saint saint 3039 ene 12 2016 projects.html
drwxr-xr-x 6 saint saint 4096 sep 16 22:32 ProyPelículas
drwxr-xr-x 6 saint saint 4096 sep 22 18:07 RocketProy
-rw-r--r-- 1 saint saint 3554 ene 12 2016 singlepost.html
[saint@saint-linux Documentos]$ dotnet new blazorwasm -o HexDataMovies --hosted
The template "Blazor WebAssembly App" was created successfully.
This template contains technologies from parties other than Microsoft, see https://aka.ms/aspnetcore/5.0-third-party-notices for details.

Processing post-creation actions...
Running 'dotnet restore' on HexDataMovies/HexDataMovies.sln...
  Determining projects to restore...
    Restored /home/saint/Documentos/HexDataMovies/Shared/HexDataMovies.Shared.csproj (in 220 ms).
    Restored /home/saint/Documentos/HexDataMovies/Server/HexDataMovies.Server.csproj (in 399 ms).
    Restored /home/saint/Documentos/HexDataMovies/Client/HexDataMovies.Client.csproj (in 135 ms).
Restore succeeded.

[saint@saint-linux Documentos]$ cd HexDataMovies/
[saint@saint-linux HexDataMovies]$ code .
```

#### 2. Creación de la Clase Movie:

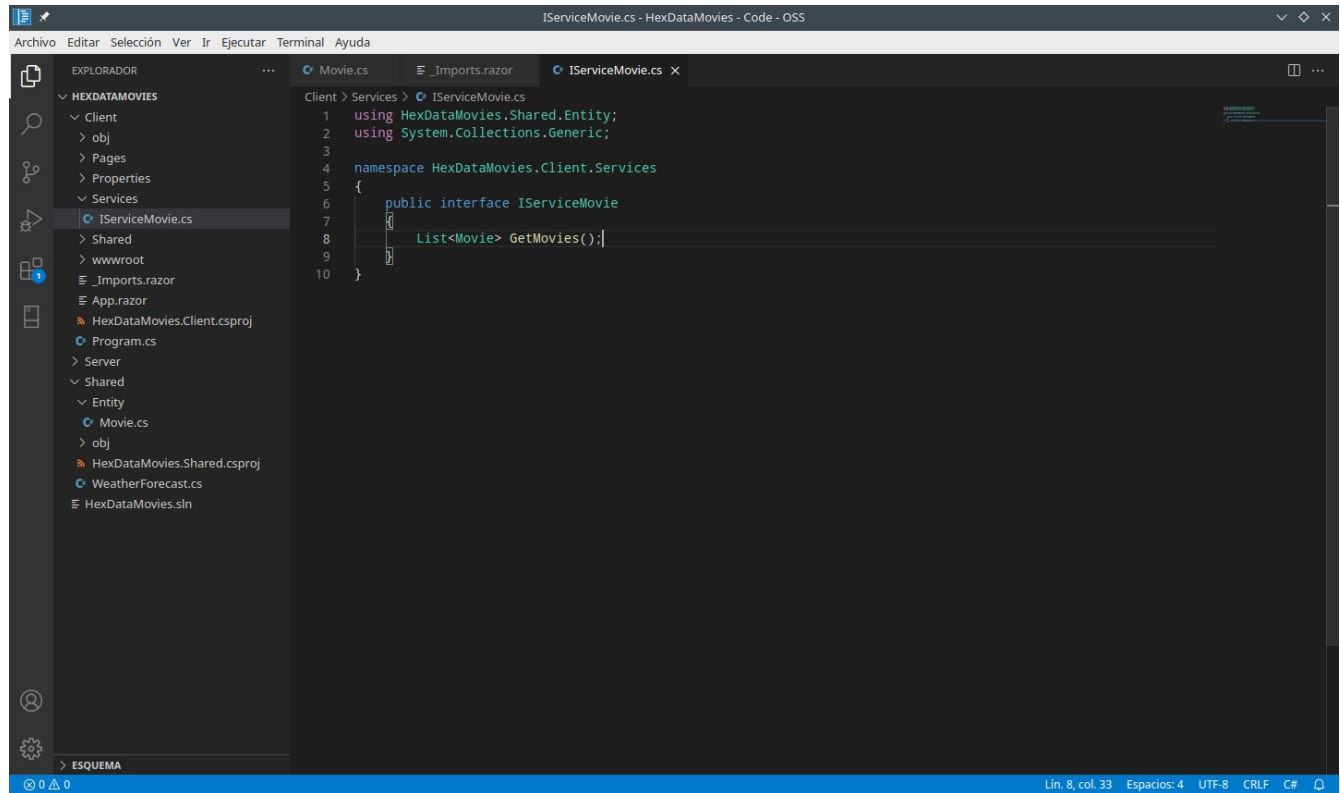
The screenshot shows the Visual Studio Code interface with the 'Movie.cs' file open in the editor. The file is located in the 'Shared > Entity > Movie.cs' path. The code defines a 'Movie' class with the following properties:

```
using System;

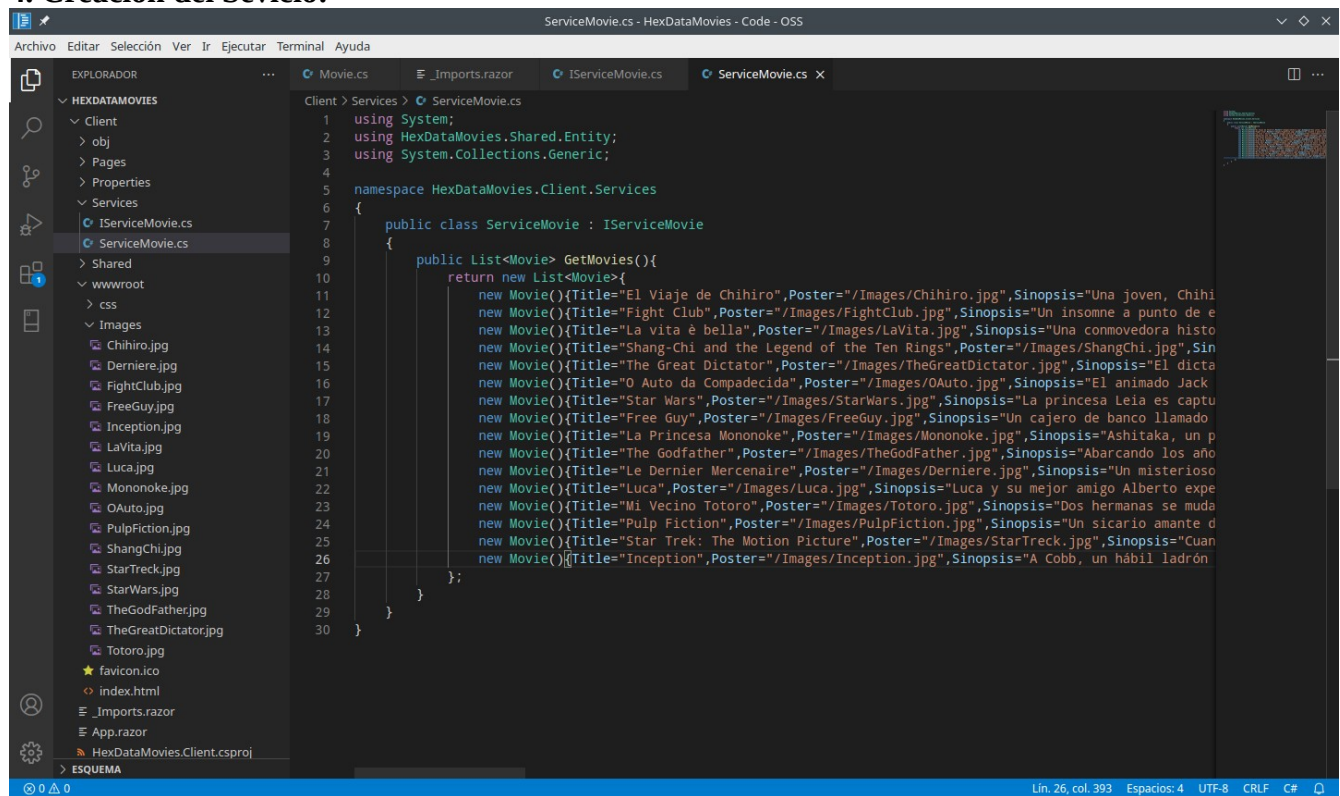
namespace HexDataMovies.Shared.Entity
{
    public class Movie
    {
        public string Title {get;set;}
        public string Poster {get;set;}
        public string Sinopsis {get;set;}
        public string Genero {get;set;}
        public string Language {get;set;}
        public string Country {get;set;}
        public float Score {get;set;}
        public TimeSpan RunningTime {get;set;}
        public DateTime Premier {get;set;}
    }
}
```

The left sidebar shows the 'EXPLORADOR' (Explorer) view with the project structure. The 'HEXDATAMOVIES' folder is expanded, showing subfolders like 'Shared', 'Server', and 'Client'. The 'Entity' folder under 'Shared' is selected, and 'Movie.cs' is the active file.

### 3. Creación de la Interfaz del servicio:

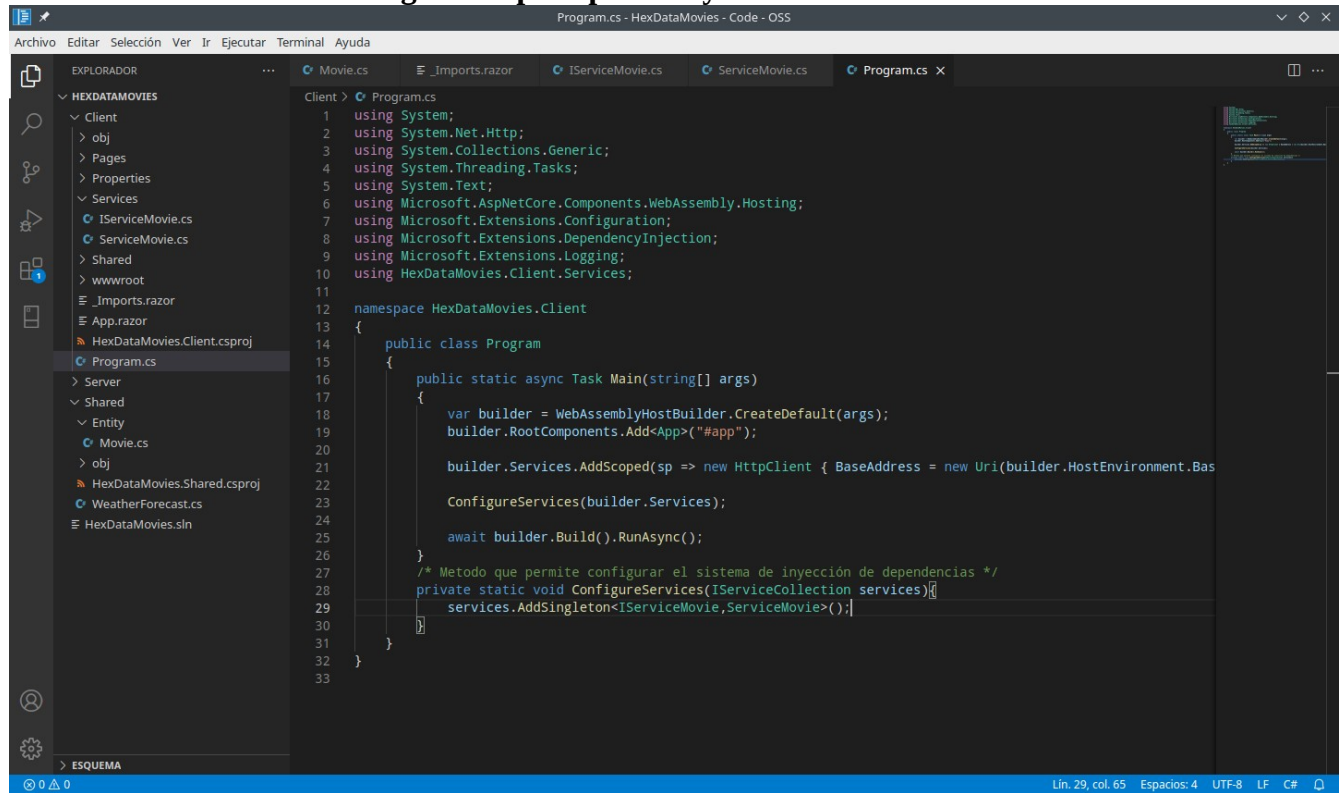


### 4. Creación del Servicio:



Aca hay un ejemplo de los datos de películas, estos fueron sacados de TMDB para popular la lista, en el GitHub está el código fuente para poder observar las partes que no salieron en la captura de pantalla.

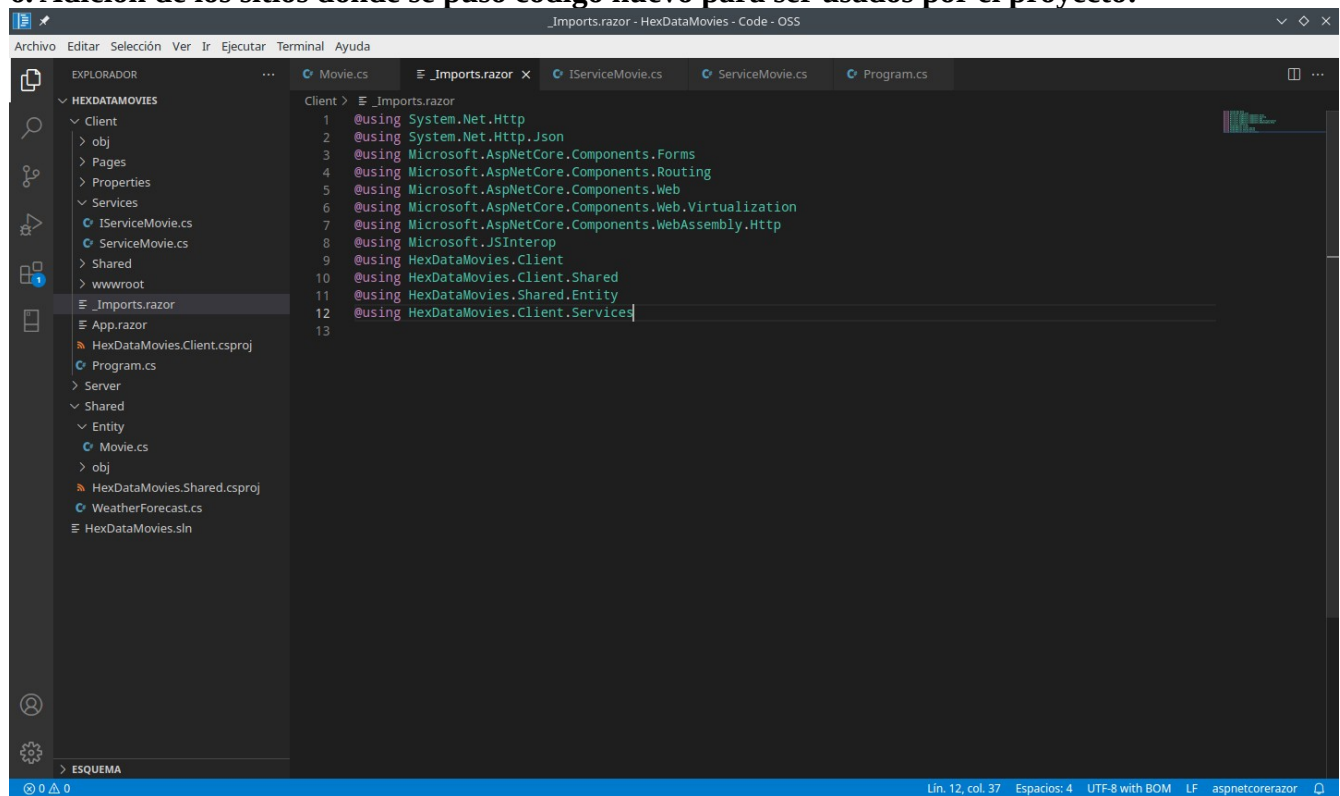
## 5. Cambios en el archivo Program.cs para poder inyectar el servicio:



```
Program.cs - HexDataMovies - Code - OSS
Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
EXPLORADOR
HEXDATAMOVIES
Client
obj
Pages
Properties
Services
IServiceMovie.cs
ServiceMovie.cs
Shared
wwwroot
_Imports.razor
App.razor
HexDataMovies.Client.csproj
Program.cs
Server
Shared
Entity
Movie.cs
obj
HexDataMovies.Shared.csproj
WeatherForecast.cs
HexDataMovies.sln
ESQUEMA
0 0 0
Lín. 29, col. 65 Espacios: 4 UTF-8 LF C#
```

```
Client > Program.cs
1 using System;
2 using System.Net.Http;
3 using System.Collections.Generic;
4 using System.Threading.Tasks;
5 using System.Text;
6 using Microsoft.AspNetCore.Components.WebAssembly.Hosting;
7 using Microsoft.Extensions.Configuration;
8 using Microsoft.Extensions.DependencyInjection;
9 using Microsoft.Extensions.Logging;
10 using HexDataMovies.Client.Services;
11
12 namespace HexDataMovies.Client
13 {
14     public class Program
15     {
16         public static async Task Main(string[] args)
17         {
18             var builder = WebAssemblyHostBuilder.CreateDefault(args);
19             builder.RootComponents.Add<App>("#app");
20
21             builder.Services.AddScoped(sp => new HttpClient { BaseAddress = new Uri(builder.HostEnvironment.BaseAddress) });
22
23             ConfigureServices(builder.Services);
24
25             await builder.Build().RunAsync();
26         }
27
28         /* Metodo que permite configurar el sistema de inyección de dependencias */
29         private static void ConfigureServices(IServiceCollection services){
30             services.AddSingleton<IServiceMovie, ServiceMovie>();
31         }
32     }
33 }
```

## 6. Adición de los sitios donde se puso código nuevo para ser usados por el proyecto:

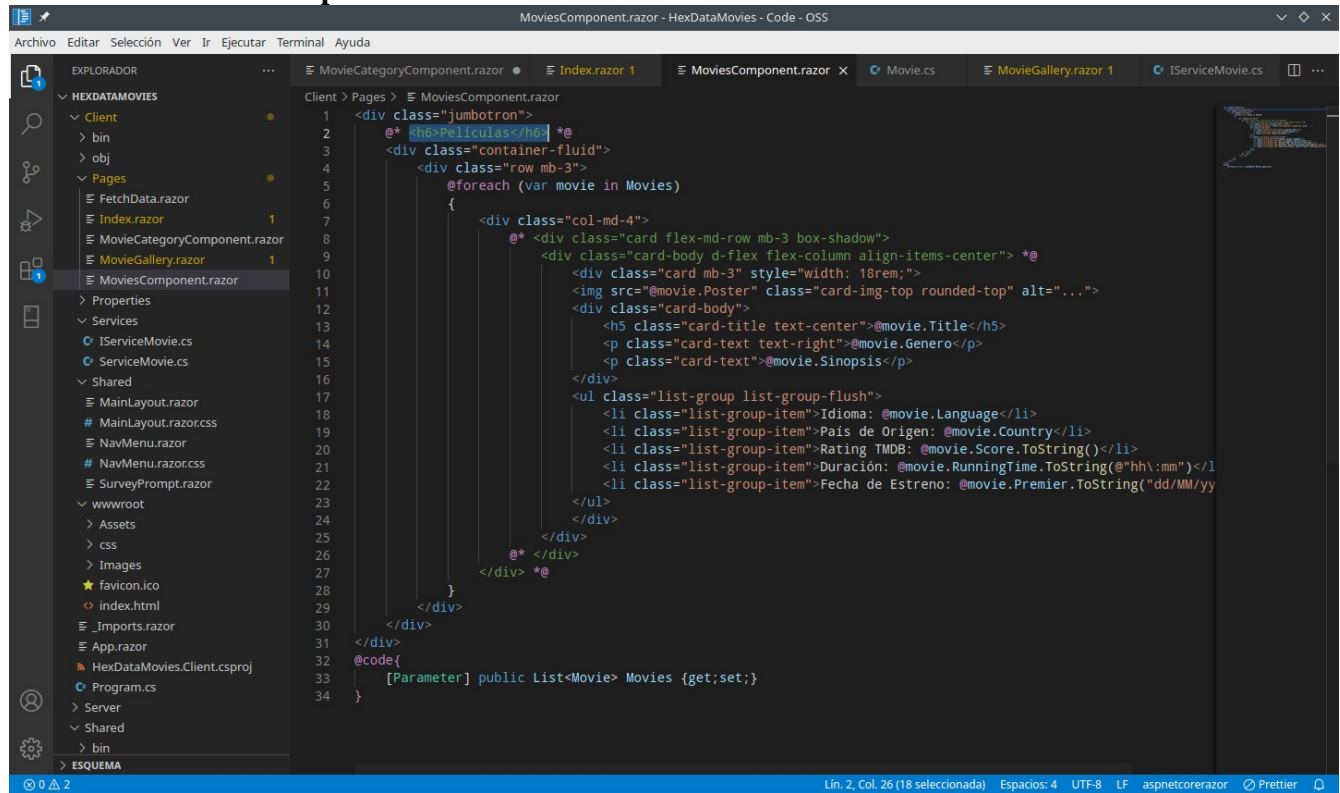


```
_Imports.razor - HexDataMovies - Code - OSS
Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
EXPLORADOR
HEXDATAMOVIES
Client
obj
Pages
Properties
Services
IServiceMovie.cs
ServiceMovie.cs
Shared
wwwroot
_Imports.razor
App.razor
HexDataMovies.Client.csproj
Program.cs
Server
Shared
Entity
Movie.cs
obj
HexDataMovies.Shared.csproj
WeatherForecast.cs
HexDataMovies.sln
ESQUEMA
0 0 0
Lín. 12, col. 37 Espacios: 4 UTF-8 with BOM LF aspNetCore.razor
```

```
Client > _Imports.razor
1 @using System.Net.Http
2 @using System.Net.Http.Json
3 @using Microsoft.AspNetCore.Components.Forms
4 @using Microsoft.AspNetCore.Components.Routing
5 @using Microsoft.AspNetCore.Components.Web
6 @using Microsoft.AspNetCore.Components.Web.Virtualization
7 @using Microsoft.AspNetCore.Components.WebAssembly.Http
8 @using Microsoft.JSInterop
9 @using HexDataMovies.Client
10 @using HexDataMovies.Client.Shared
11 @using HexDataMovies.Shared.Entity
12 @using HexDataMovies.Client.Services
13
```

Estas adiciones fueron en el archivo \_Imports.razor, se adicionó Entity y Services.

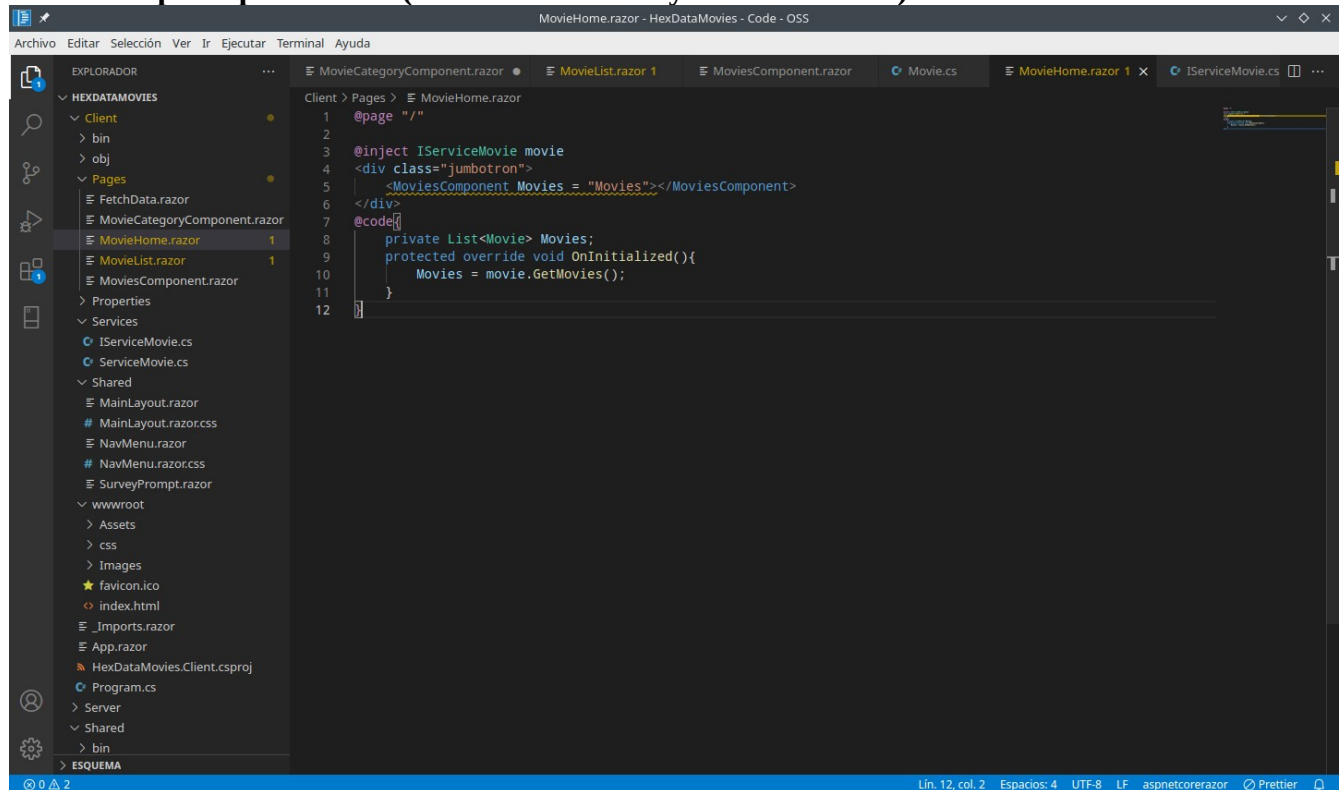
## 7. Archivo MoviesComponent.razor



```
Client > Pages > MoviesComponent.razor
1 <div class="jumbotron">
2     @* <h2>Películas</h2> *@
3 <div class="container-fluid">
4     <div class="row mb-3">
5         @foreach (var movie in Movies)
6         {
7             <div class="col-md-4">
8                 @* <div class="card flex-md-row mb-3 box-shadow">
9                     <div class="card-body d-flex flex-column align-items-center"> *@
10                    <div class="card mb-3" style="width: 18rem;">
11                        
12                        <div class="card-body">
13                            <h5 class="card-title text-center">@movie.Title</h5>
14                            <p class="card-text text-right">@movie.Genero</p>
15                            <p class="card-text">@movie.Sinopsis</p>
16                        </div>
17                        <ul class="list-group list-group-flush">
18                            <li class="list-group-item">Idioma: @movie.Language</li>
19                            <li class="list-group-item">País de Origen: @movie.Country</li>
20                            <li class="list-group-item">Rating TMDb: @movie.Score.ToString()</li>
21                            <li class="list-group-item">Duración: @movie.RunningTime.ToString(@"hh:mm")</li>
22                            <li class="list-group-item">Fecha de Estreno: @movie.Premier.ToString("dd/MM/yy")</li>
23                        </ul>
24                    </div>
25                </div>
26            </div>
27            @* </div> *@
28        }
29    </div>
30 </div>
31 </div>
32 @code{
33     [Parameter] public List<Movie> Movies {get;set;}
34 }
```

En este se está organizando la información a mostrar en el main. Está organizada por medio de cards de bootstrap, el diseño puede mejorarse. Por premuras por los tiempos de entrega no le pude trabajar más.

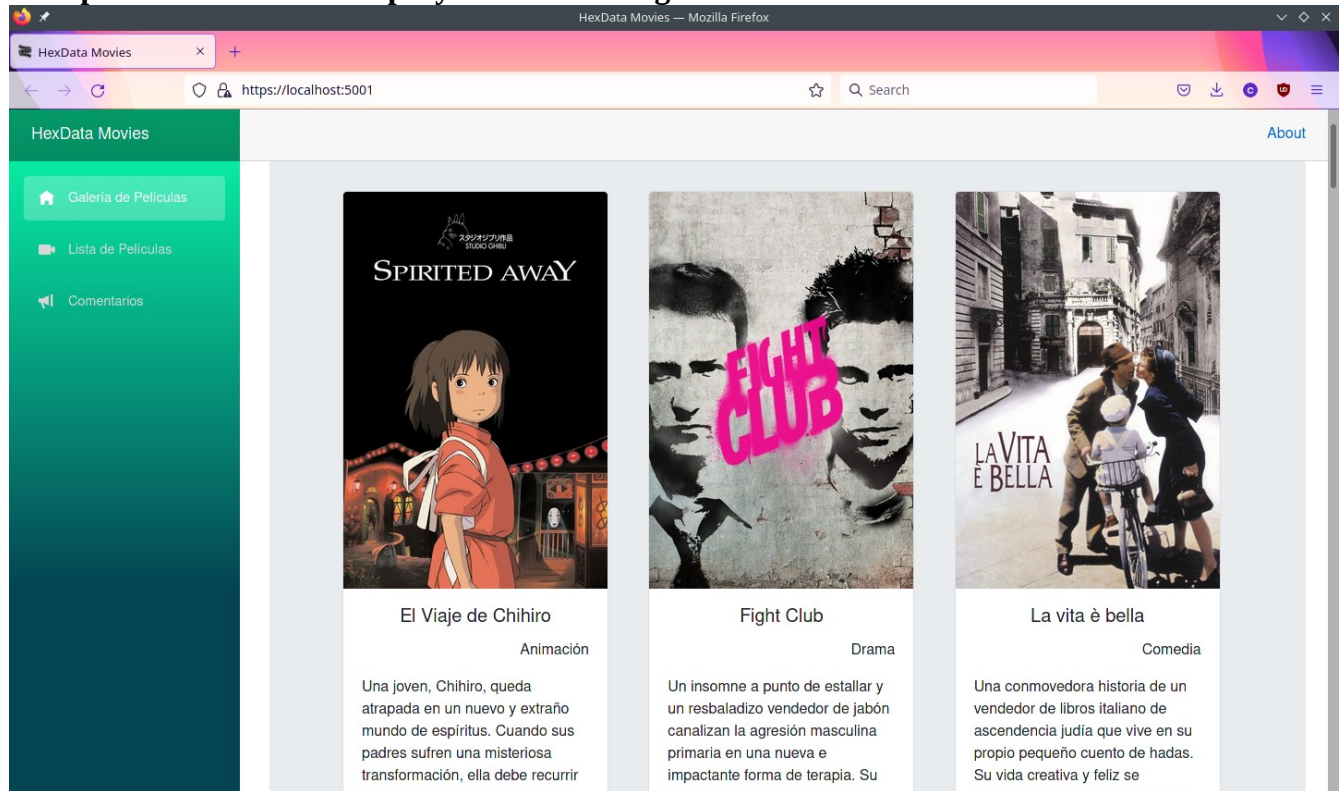
## 8. Archivo principal de html (en donde se va a inyectar el servicio):



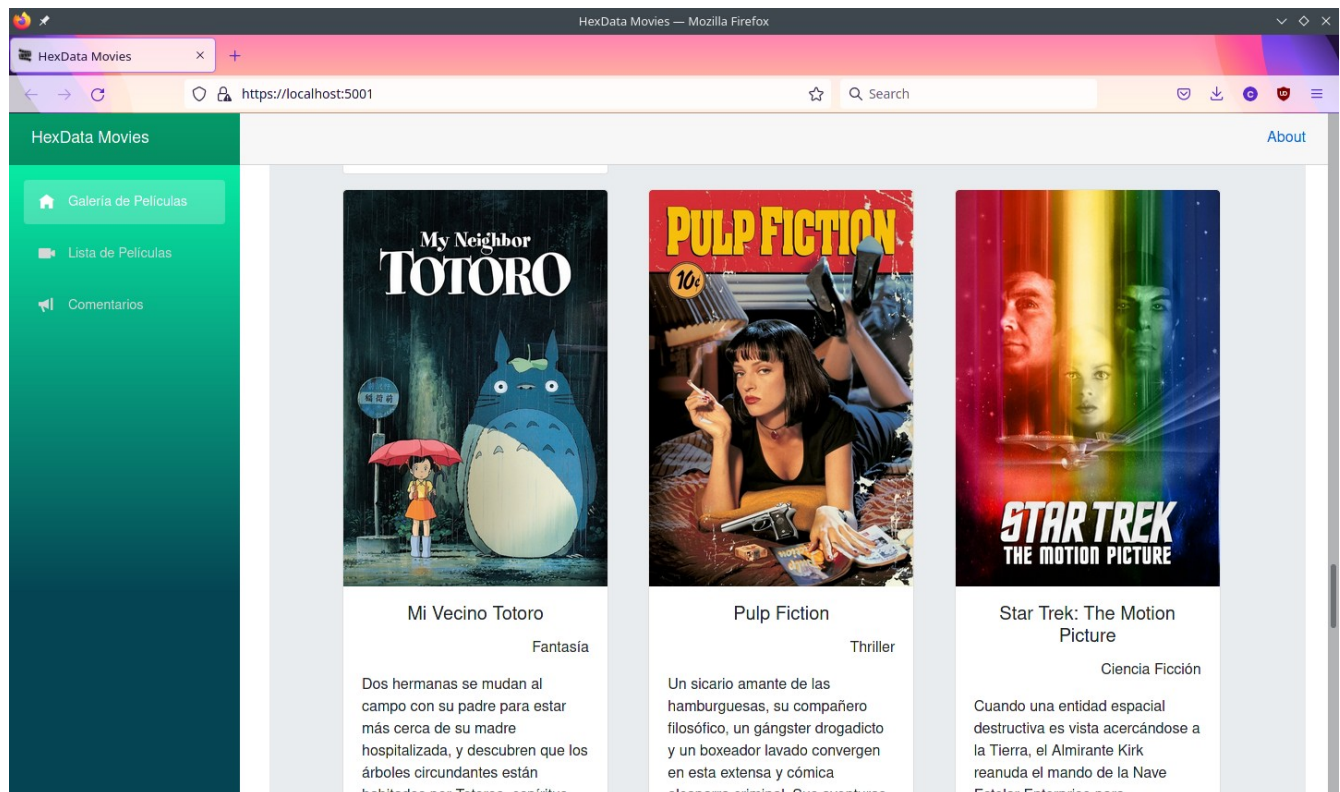
```
Client > Pages > MovieHome.razor
1 @page "/"
2
3 @inject IServiceMovie movie
4 <div class="jumbotron">
5     <MoviesComponent Movies = "Movies"></MoviesComponent>
6 </div>
7 @code{
8     private List<Movie> Movies;
9     protected override void OnInitialized(){
10         Movies = movie.GetMovies();
11     }
12 }
```



## 9. Capturas de Pantalla del proyecto en el Navegador:



El favicon se cambió para mostrar una tira de celuloide, al nombre de la app se le adicionó un espacio y los colores de la barra de navegación degradada se cambiaron.



Me gusta más en grupos de a tres, aunque para la app de sprints posteriores deben ser mas resultados por fila para aprovechar el espacio.