

Screenshots

1. Creación proyecto web (sección 3.1)

Configure su nuevo proyecto

Aplicación web de ASP.NET Core (Modelo-Vista-Controlador) C# Linux macOS Windows

Nombre del proyecto

laboratorio3

Ubicación

C:\Users\andre\Documents\UCR\laboratorios_ing_e_23a\Laboratorios\laboratorio3\ ...

Nombre de la solución ⓘ

laboratorio3

☐ Colocar la solución y el proyecto en el mismo directorio

Proyecto se creará en "C:\Users\andre\Documents\UCR\laboratorios_ing_e_23a\Laboratorios\laboratorio3\laboratorio3\laboratorio3\."

Andres Matarrita C04668

Información adicional

Aplicación web de ASP.NET Core (Modelo-Vista-Controlador) C# Linux

Framework ⓘ

.NET 6.0 (Compatibilidad a largo plazo)

Authentication de campo ⓘ

Ninguno

☒ Configurar para HTTPS ⓘ

☐ Habilitar Docker ⓘ

Sistema operativo de Docker ⓘ

Linux

☐ No usar instrucciones de nivel superior ⓘ

Andres Matarrita C04668

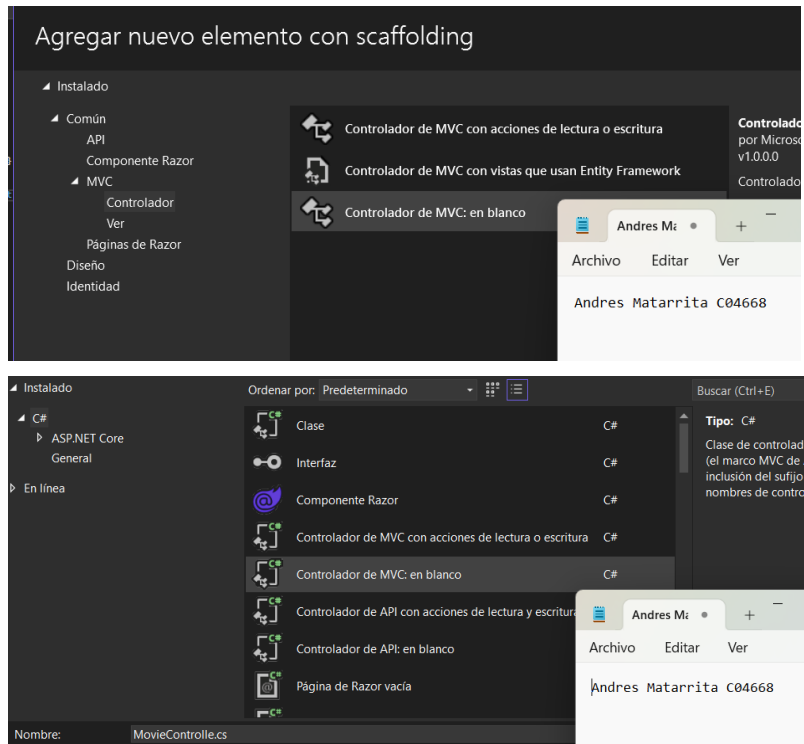
2. Creación del primer modelo (sección 3.2)

```
namespace laboratorio3.Models
{
    0 referencias
    public class MovieModel
    {
        0 referencias
        public int Id { get; set; }
        0 referencias
        public string Name { get; set; }
        0 referencias
        public string Genre { get; set; }
        0 referencias
        public DateTime ReleaseDate { get; set; }
    }
}
```

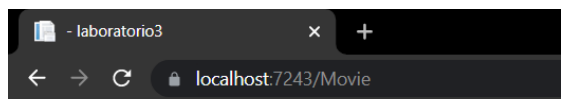
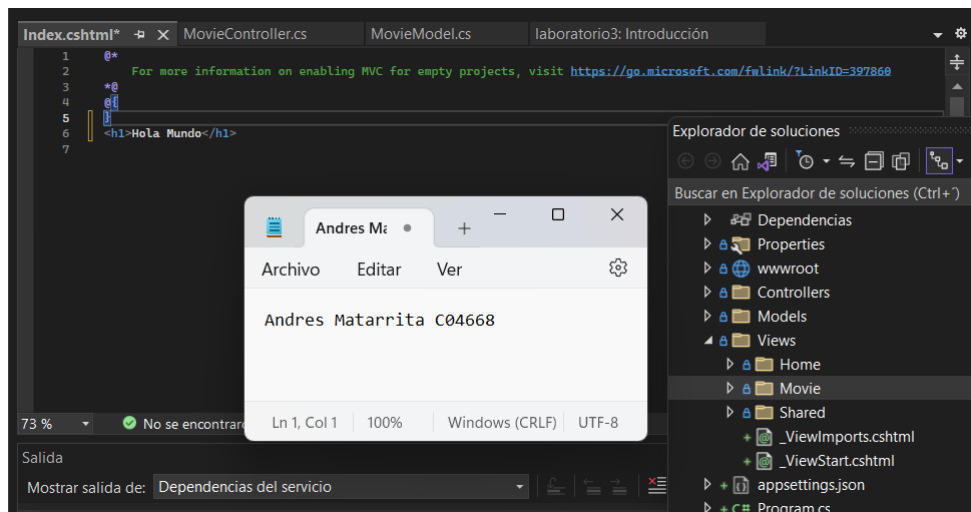
Andres Matarrita ha iniciado sesión

Configuración de la cuenta...

3. Creación del primer controlador (sección 3.3)



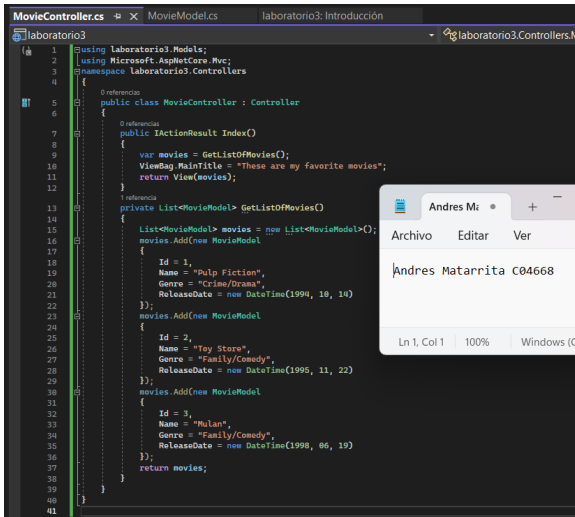
4. Creación de la primera vista (sección 3.4)



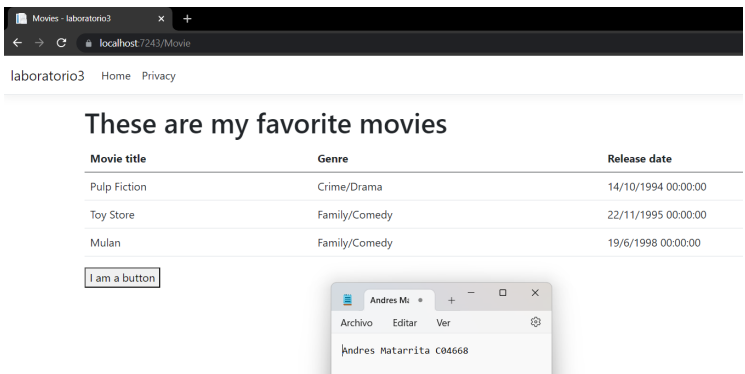
laboratorio3 Home Privacy



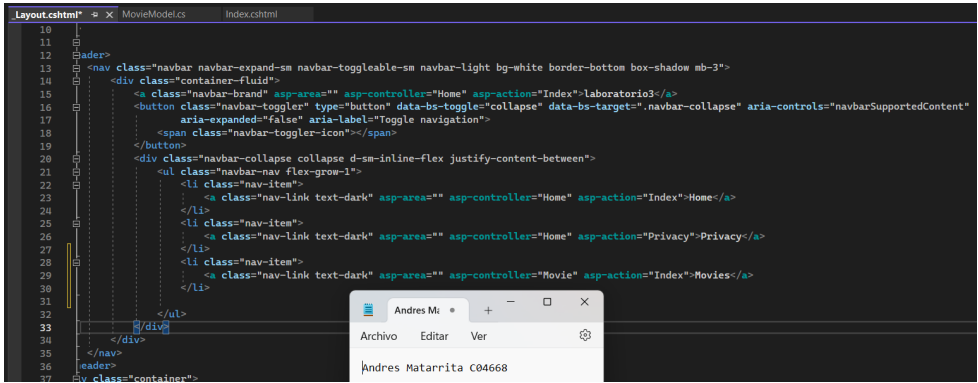
5. Creación de la nueva vista (sección 4.1)



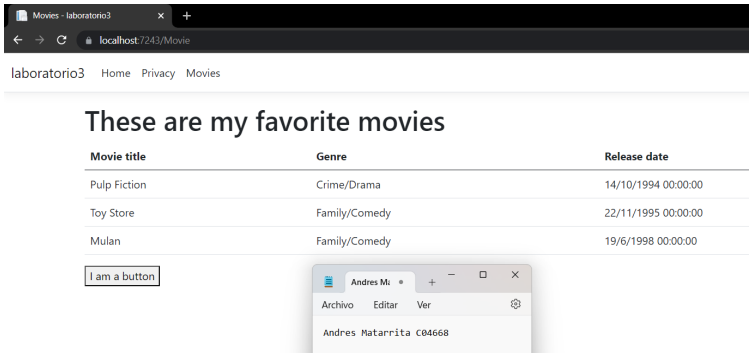
```
1 using Laboratorio3.Models;
2 using Microsoft.AspNetCore.Mvc;
3 namespace Laboratorio3.Controllers
4 {
5     public class MovieController : Controller
6     {
7         public IActionResult Index()
8         {
9             var movies = GetListOfMovies();
10            ViewBag.MainTitle = "These are my favorite movies";
11            return View(movies);
12        }
13
14        private List<MovieModel> GetListOfMovies()
15        {
16            List<MovieModel> movies = new List<MovieModel>();
17            movies.Add(new MovieModel
18            {
19                Id = 1,
20                Name = "Pulp Fiction",
21                Genre = "Crime/Drama",
22                ReleaseDate = new DateTime(1994, 10, 14)
23            });
24            movies.Add(new MovieModel
25            {
26                Id = 2,
27                Name = "Toy Store",
28                Genre = "Family/Comedy",
29                ReleaseDate = new DateTime(1995, 11, 22)
30            });
31            movies.Add(new MovieModel
32            {
33                Id = 3,
34                Name = "Mulan",
35                Genre = "Family/Comedy",
36                ReleaseDate = new DateTime(1998, 06, 19)
37            });
38            return movies;
39        }
40    }
41 }
```



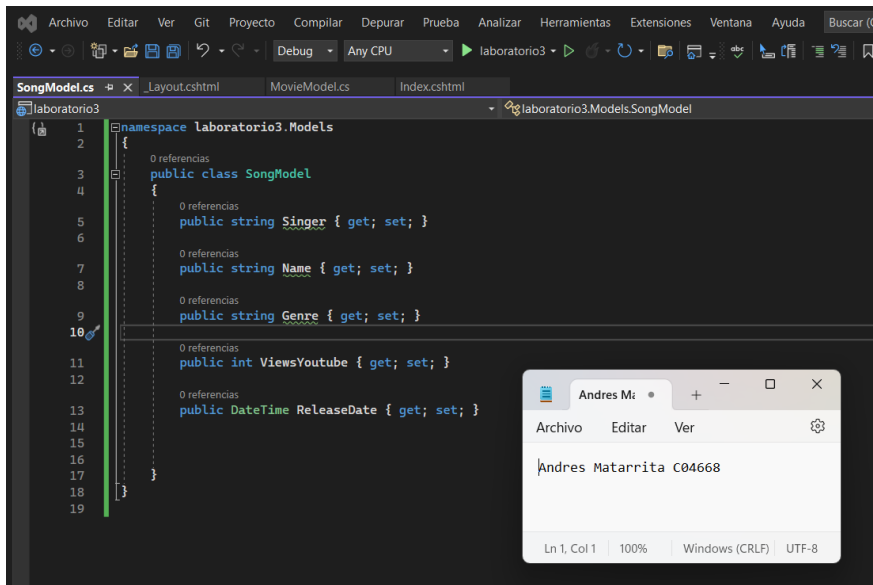
6. Modificación del layout (sección 4.2)



```
10 <div>
11 <div>
12 <div>
13 <div>
14 <div>
15 <div>
16 <div>
17 <div>
18 <div>
19 <div>
20 <div>
21 <div>
22 <div>
23 <div>
24 <div>
25 <div>
26 <div>
27 <div>
28 <div>
29 <div>
30 <div>
31 <div>
32 <div>
33 <div>
34 <div>
35 <div>
36 <div>
37 <div>
```

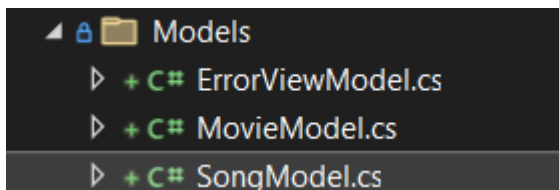


7. Modelo de la nueva funcionalidad que usted agregó (sección 5)

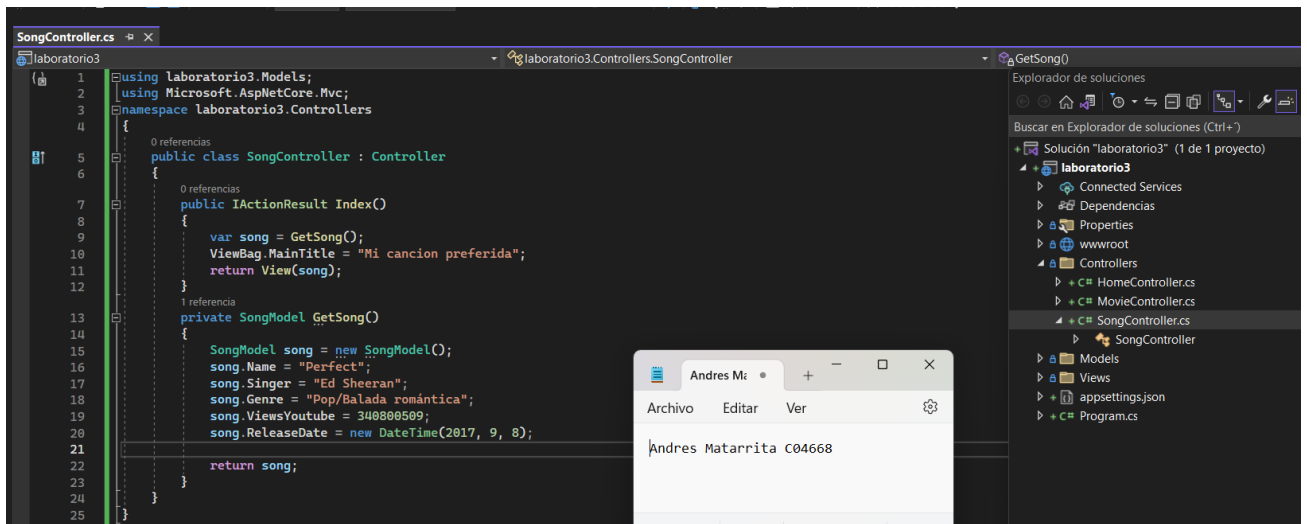


```
1 namespace laboratorio3.Models
2 {
3     public class SongModel
4     {
5         public string Singer { get; set; }
6
7         public string Name { get; set; }
8
9         public string Genre { get; set; }
10
11         public int ViewsYoutube { get; set; }
12
13         public DateTime ReleaseDate { get; set; }
14     }
15 }
16
17
18
19
```

Andres Matarrita C04668



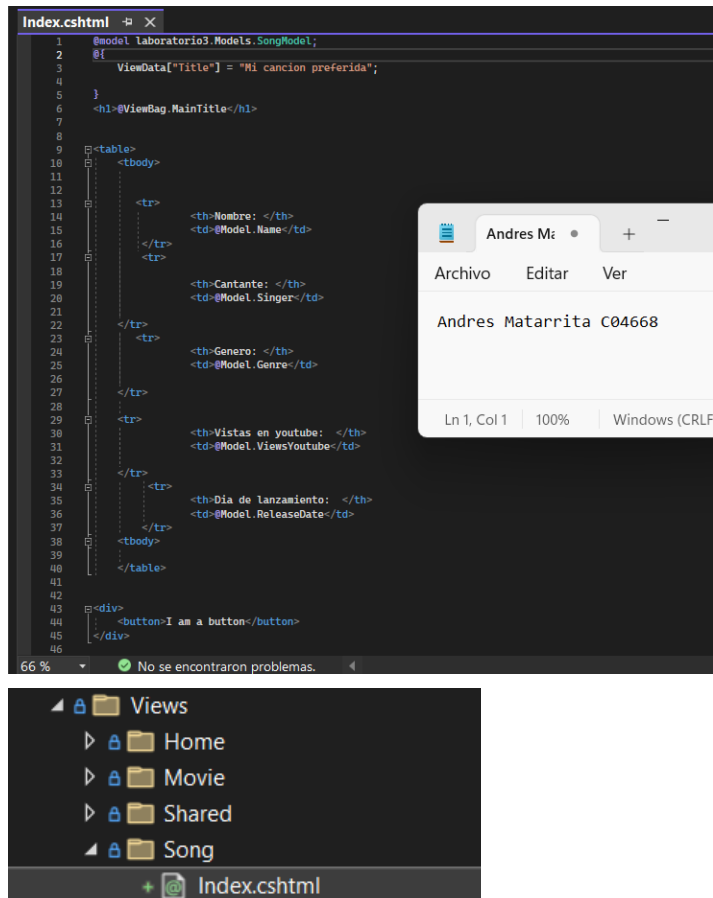
8. Controlador de la nueva funcionalidad que usted agregó (sección 5)



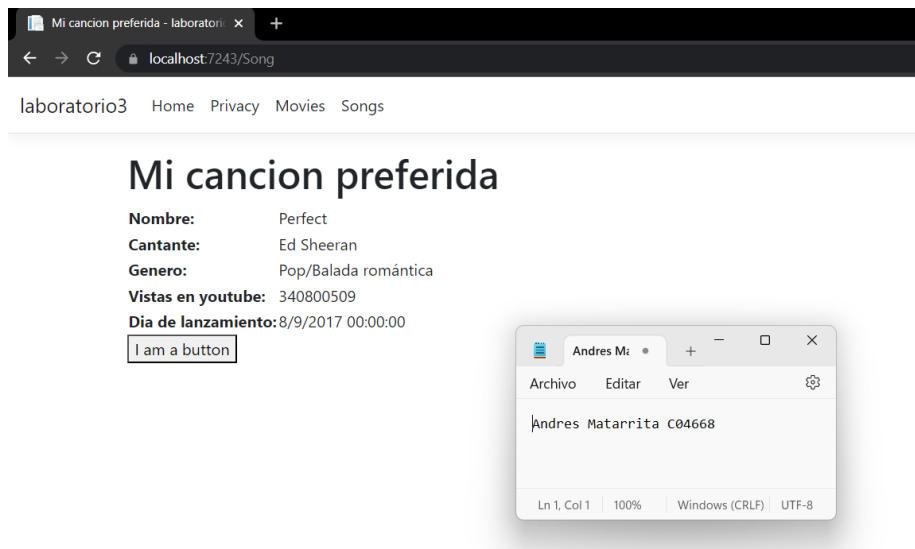
```
1 using laboratorio3.Models;
2 using Microsoft.AspNetCore.Mvc;
3 namespace laboratorio3.Controllers
4 {
5     public class SongController : Controller
6     {
7         public IActionResult Index()
8         {
9             var song = GetSong();
10             ViewBag.MainTitle = "Mi cancion preferida";
11             return View(song);
12         }
13
14         private SongModel GetSong()
15         {
16             SongModel song = new SongModel();
17             song.Name = "Perfect";
18             song.Singer = "Ed Sheeran";
19             song.Genre = "Pop/Balada romántica";
20             song.ViewsYoutube = 340800509;
21             song.ReleaseDate = new DateTime(2017, 9, 8);
22
23             return song;
24         }
25     }
26 }
```

Andres Matarrita C04668

9. Vista de la nueva funcionalidad que usted agregó (sección 5)



10. Página web de la nueva funcionalidad que usted agregó (sección 5)



Agregue un pequeño resumen, no más de 200 palabras indicando claramente:

- 1. Dos cosas que no sabía y aprendió en el laboratorio**
- 2. Una cosa que se le hizo difícil de realizar y explique por qué fue difícil.**
- 3. Una cosa que se le hizo fácil de realizar y explique por qué fue fácil.**
- 4. Indique cuánto tiempo tardó en realizar el laboratorio**

Al realizar este proyecto aprendí un poco sobre la sintaxis de HTML, debido a que desconocía totalmente este lenguaje. Para poder realizar el último ejercicio de crear una “vista” para demostrar atributos de mi canción preferida, tuve que investigar y estudiar por mi cuenta para cumplir con dicha tarea. También pude aprender la noción de cómo funciona el modelo MVC por debajo en una página web y lo importante que es esta separación de funcionalidades para la escalabilidad del proyecto. Me resultó muy difícil el entender el funcionamiento de HTML y su sintaxis debido a que es la primera vez que lo utilizo. Por otra parte, me resultó fácil realizar un “Modelo” debido a que en el curso de “diseño de software” diseñé modelos muy similares a este, además de que ya conozco ligeramente C#. Para este laboratorio me tomaron 3 horas y media debido a que tomé mi tiempo para entender este patrón de MVC.

Enlace repositorio github:

https://github.com/AndresMatarritaC04668/c04668_ci0126_23a.git

Resumen de al menos 400 y no más de 500 palabras en relación con el tutorial de la sección 4. Explique en sus propias palabras en qué consiste la arquitectura MVC. Para cada uno de los tres conceptos modelo, vista, y controlador, explique en sus propias palabras: ¿Qué es?, ¿Cuál es su función en la arquitectura MVC?

En el tutorial se habla acerca de que ASP.NET Core, la arquitectura MVC (Modelo-Vista-Controlador) es un patrón de diseño común que divide la aplicación en tres componentes que están interconectados: el modelo, la vista y el controlador. El componente del modelo representa los datos y la lógica comercial de la aplicación, y es responsable de recuperar y almacenar datos. El componente de vista representa la interfaz de usuario de la aplicación y es responsable de mostrar los datos al usuario. El componente del controlador maneja las solicitudes del usuario y decide qué datos del modelo recuperar y qué vista mostrar al usuario. La implementación de MVC de ASP.NET Core incluye funciones como enrutamiento, filtros de acción, enlace de modelos y validación. Estas funciones permiten a los desarrolladores crear aplicaciones web fáciles de mantener, probar y escalar. En general, el uso del patrón MVC en ASP.NET Core puede proporcionar una separación clara de las preocupaciones entre los diferentes componentes de la aplicación, lo que facilita la creación y el mantenimiento de aplicaciones web sólidas y escalables.

En lo personal yo entiendo la arquitectura MVC como un patrón de diseño de software que separa una aplicación en tres componentes interconectados: modelo, vista y controlador. Por otra parte, de acuerdo con lo leído, defino personalmente sus componentes como:

El modelo: este componente es responsable de administrar los datos y la lógica comercial de la aplicación. Además de representar la estructura de los datos y cómo se almacenan, procesan y recuperan. El modelo notifica al controlador de cualquier cambio en los datos y el controlador actualiza la vista en consecuencia.

La vista: Este componente se encarga de mostrar los datos al usuario de una forma amigable. Representa la interfaz de usuario y proporciona una forma para que los usuarios interactúen con la aplicación. La vista se comunica con el controlador, le informa sobre las acciones del usuario y actualiza los datos que se muestran en función de los cambios realizados por el controlador.

El controlador: este componente es responsable de manejar las solicitudes de los usuarios y actúa como intermediario entre el modelo y la vista. El controlador recibe información del usuario a través de la vista, actualiza el modelo en consecuencia y actualiza la vista con la información más reciente.

En resumen, el modelo, la vista y el controlador son los tres componentes principales de la arquitectura MVC, cada uno con una función específica. El modelo administra los datos y la lógica comercial, la vista muestra los datos de una manera fácil de usar y el controlador actúa como intermediario entre los dos, manejando las solicitudes de los usuarios y manteniendo sincronizados los datos de la aplicación y la interfaz de usuario.