

***UNIVERSIDAD DE COSTA RICA***

***Ingeniería de Software***

***CI-0126***

***Laboratorios***

***Reporte Lab. 3***

***ELABORADO POR***

***Josué Retana Rodríguez C06440***

***PROFESOR ASESOR***

***Allan Berrocal Rojas***

***2023***

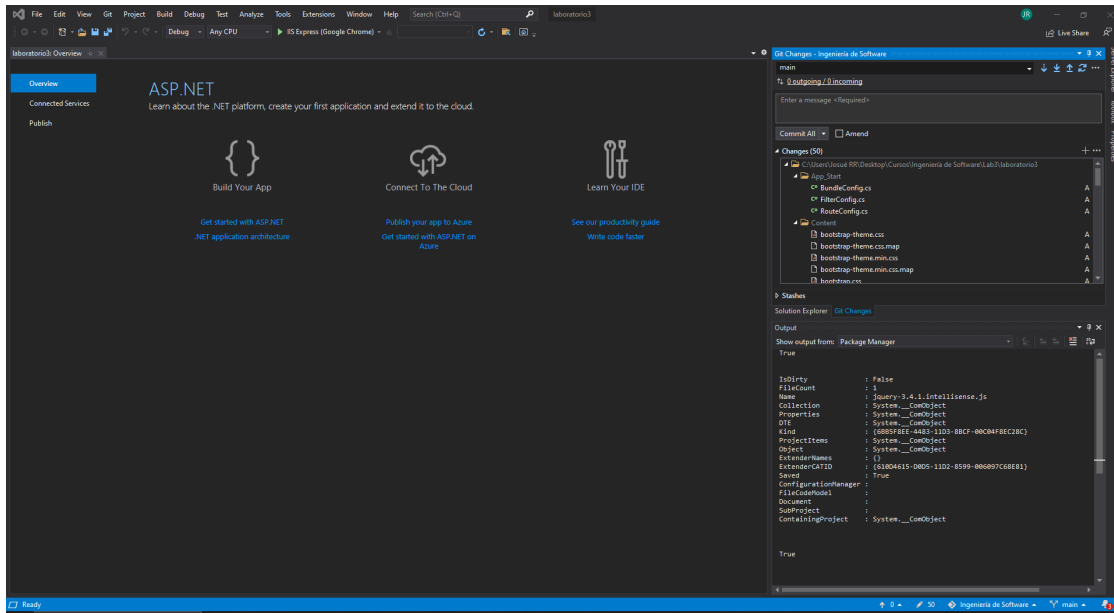
# Reporte Laboratorio 3

## Repositorio

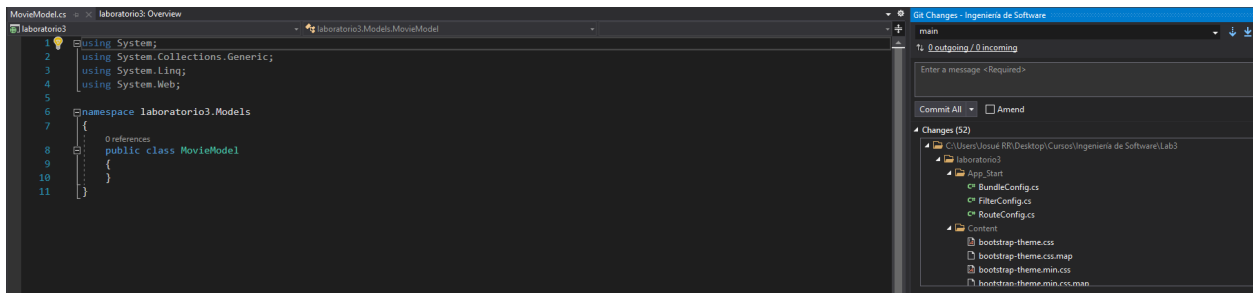
[https://github.com/JosueRR/IngenieriaSoftware\\_23a](https://github.com/JosueRR/IngenieriaSoftware_23a)

## Evidencias

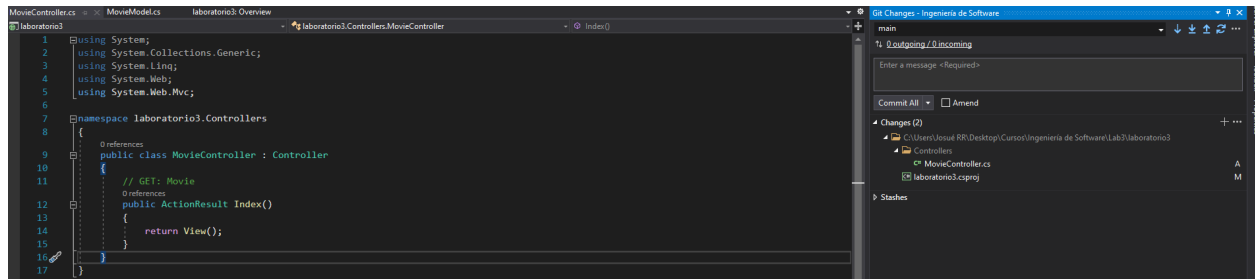
### Creación proyecto web



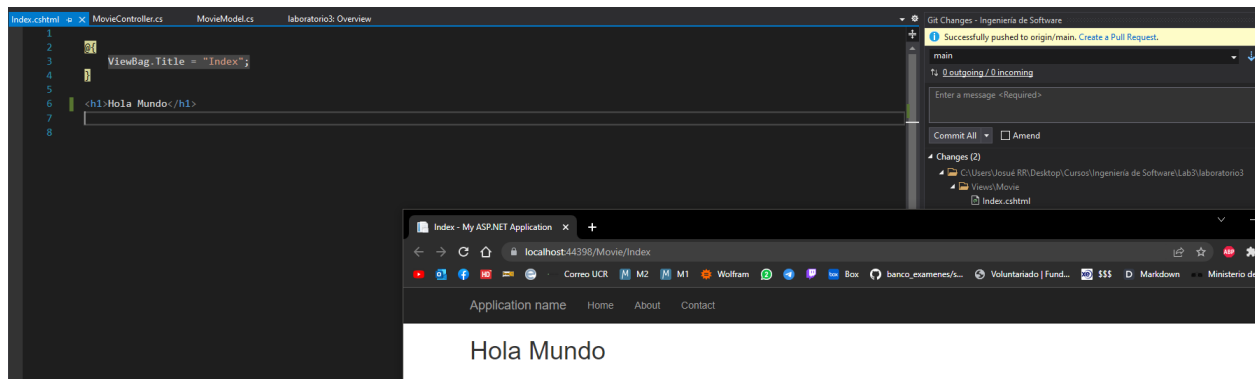
### Creación del primer modelo



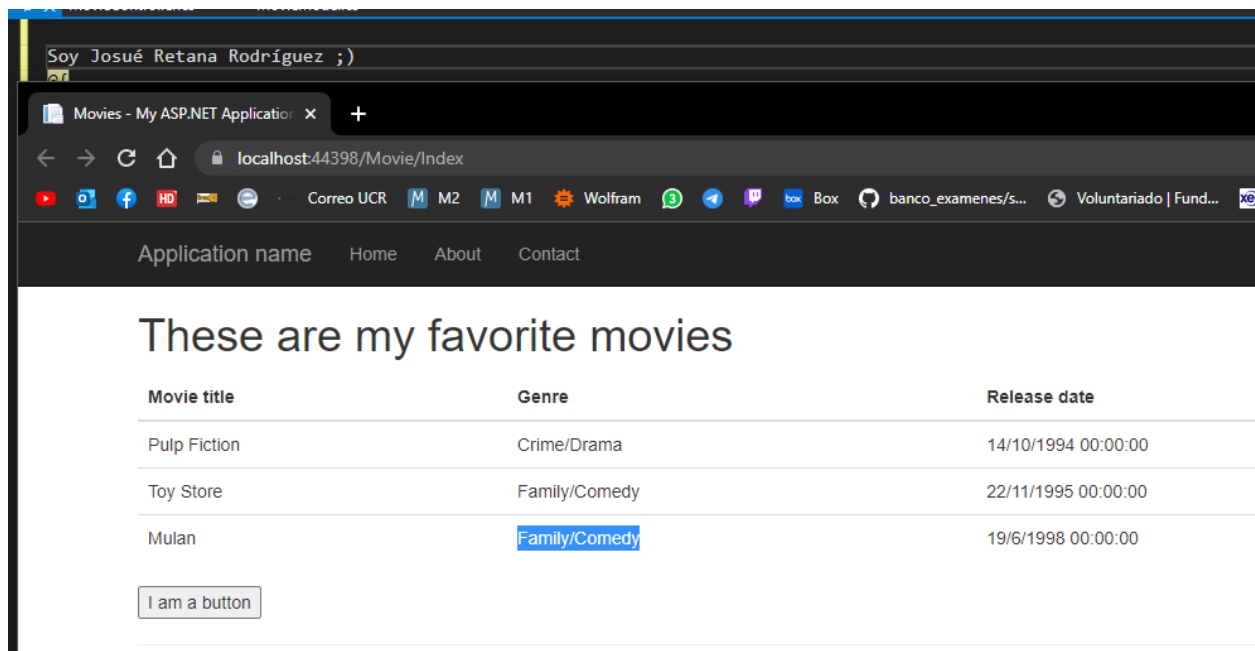
### Creación del primer controlador



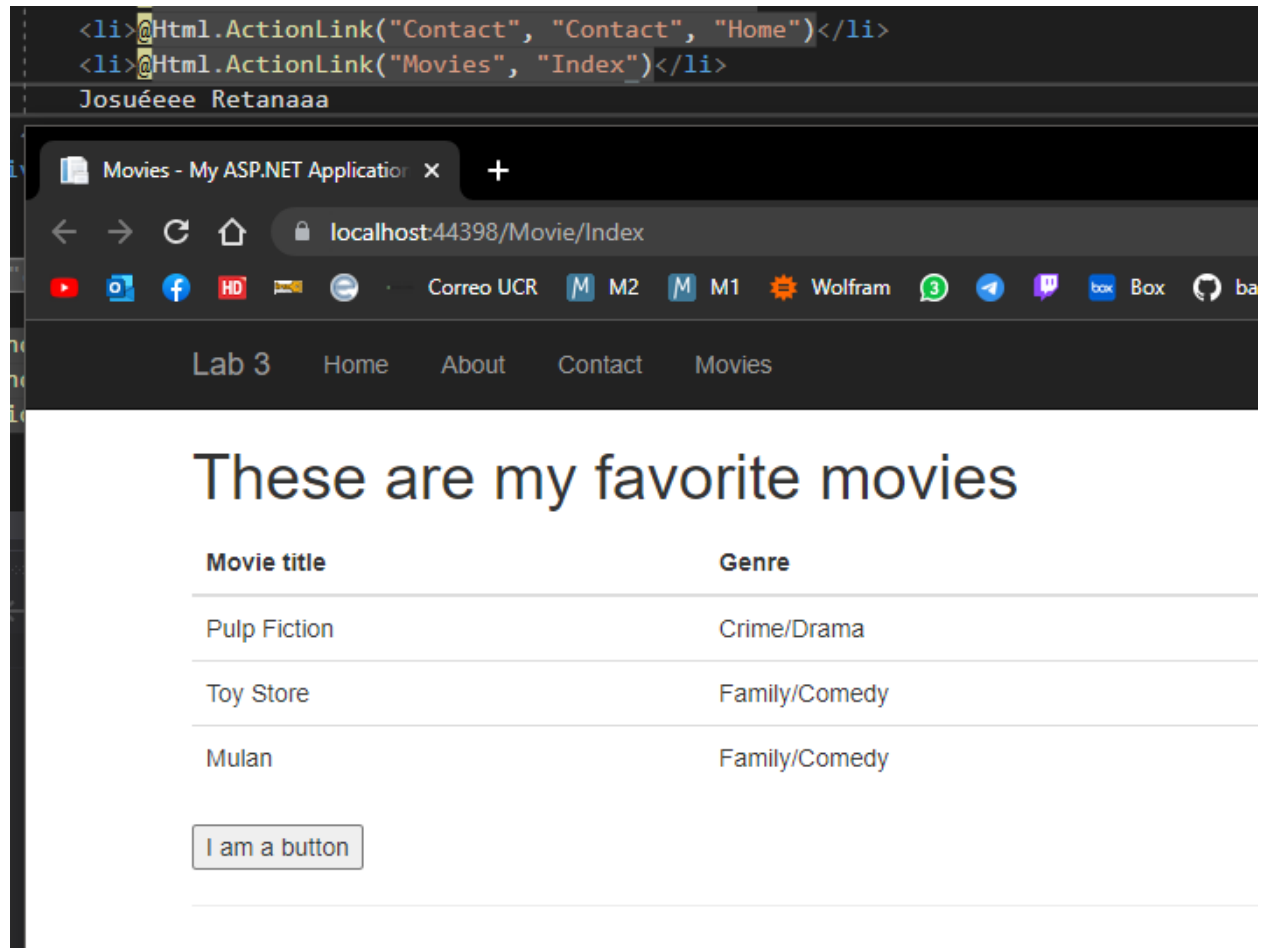
## Creación de la primera vista



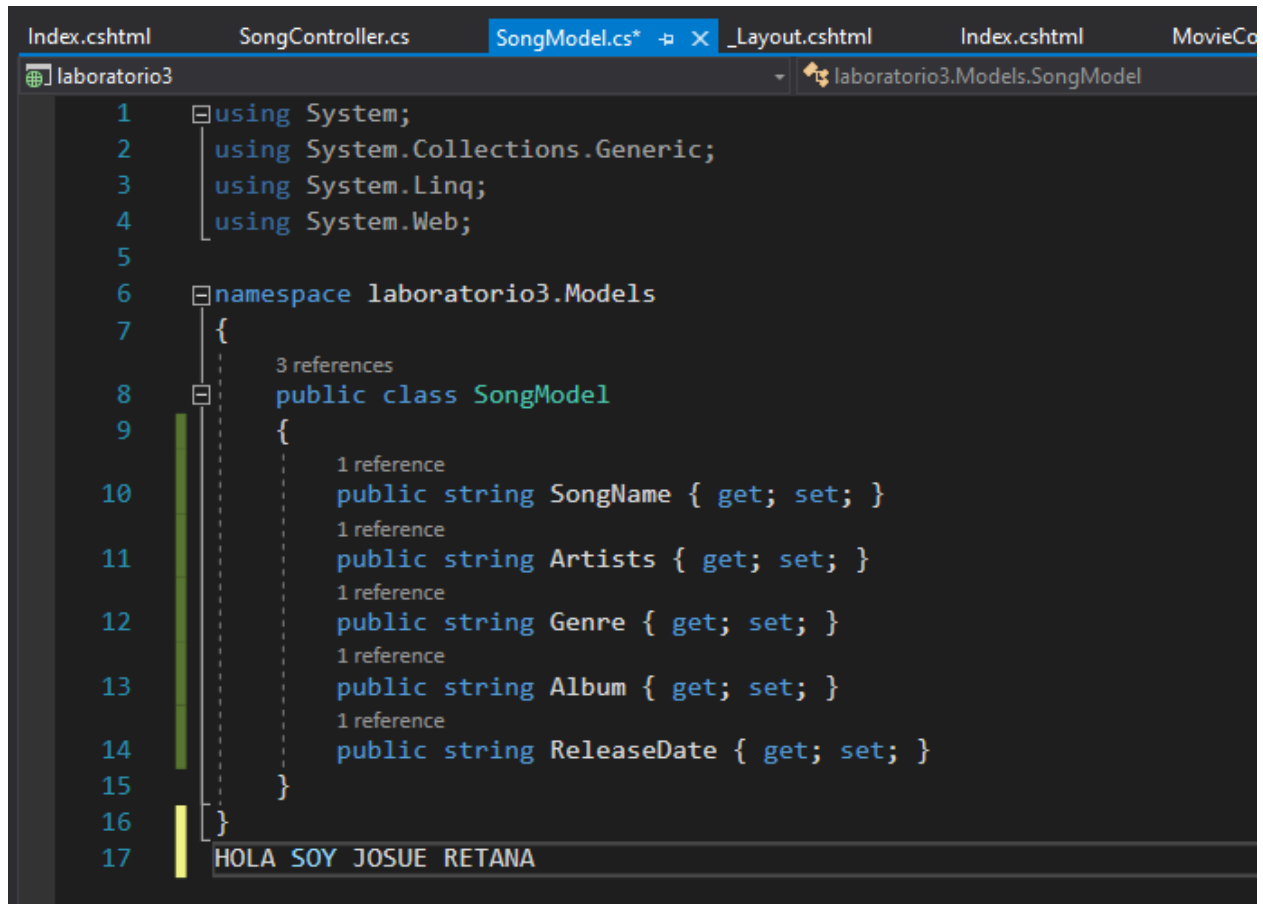
## Creación de la nueva vista



## Modificación del layout



Modelo de la nueva funcionalidad que usted agregó



```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5
6  namespace laboratorio3.Models
7  {
8      public class SongModel
9      {
10         public string SongName { get; set; }
11         public string Artists { get; set; }
12         public string Genre { get; set; }
13         public string Album { get; set; }
14         public string ReleaseDate { get; set; }
15     }
16 }
17 HOLA SOY JOSUE RETANA
```

Controlador de la nueva funcionalidad que usted agregó

```
Index.cshtml SongController.cs* SongModel.cs* _Layout.cshtml Index.cshtml MovieController.cs MovieModel.cs
laboratorio3
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.Mvc;
6 using laboratorio3.Models;
7
8 namespace laboratorio3.Controllers
9 {
10     0 references
11     public class SongController : Controller
12     {
13         // GET: Song
14         0 references
15         public ActionResult Index()
16         {
17             var song = GetSongInfo();
18             ViewBag.MainTitle = "A Very Good Song :>";
19             return View(song);
20         }
21         1 reference
22         private SongModel GetSongInfo()
23         {
24             SongModel song = new SongModel();
25             song.SongName = "Sale el Sol";
26             song.Artists = "Young Miko";
27             song.Genre = "HipHop/Trap";
28             song.Album = "None";
29             song.ReleaseDate = "2022, 12, 5";
30             return song;
31         }
32     }
33     HOLA SOY JOSUE RETANA
```

Vista de la nueva funcionalidad que usted agregó

```
index.cshtml* x SongController.cs SongModel.cs _Layout.cshtml Index.cshtml MovieController.cs
1 @{}
2   ViewData["Title"] = "Song";
3 }
4
5 <h1>@ViewBag.MainTitle</h1>
6 <div>
7   <table class="table">
8     <thead>
9       <tr>
10        <th>Song Name </th>
11        <th>Artist </th>
12        <th>Genre</th>
13        <th>Album</th>
14        <th>Release Date</th>
15      </tr>
16    </thead>
17    <tbody>
18      <tr>
19        <td>@Model.SongName</td>
20        <td>@Model.Artists</td>
21        <td>@Model.Genre</td>
22        <td>@Model.Album</td>
23        <td>@Model.ReleaseDate</td>
24      </tr>
25    </tbody>
26  </table>
27 </div>
28
29 <div>
30   <button>Me gusta!</button>
31 </div>
32 HOLA SOY JOSUE RETANA
```

Página web de la nueva funcionalidad que usted agregó

```
30 <button>Me gusta!</button>
31 </div>
32
33
34 SOY JOSUE RETANA MUCHO GUSTO JOSUE RETANA RODRIGUEZ
35
```

Song - My ASP.NET Application

localhost:44398/Song

Correo UCRM2M1WolframBoxbanco\_exámenes/s...Voluntariado | Fund...\$\$\$MarkdownMinisterio de Relaci...

Lab 3HomeAboutContactMoviesSong

## A Very Good Song :)

Song Name	Artist	Genre	Album	Release Date
Sale el Sol	Young Miko	HipHop/Trap	None	2022, 12, 5

Me gusta!

© 2023 - My ASP.NET Application



## ***Resumen MVC***

Implementada con frecuencia en la programación de computadoras, la arquitectura de software Model-View-Controller (MVC) facilita una asignación distinta de funciones en la estructura de codificación. El diseño de MVC consta de tres entidades principales: modelo, vista y controlador, todas las cuales trabajan en colaboración para construir una aplicación confiable y adaptable.

En este marco de trabajo, el modelo sirve como la representación esencial de los datos y la lógica comercial detrás de una aplicación al tiempo que permite la organización, el procesamiento y la ejecución de las operaciones necesarias en la información. Las responsabilidades entre las partes de una aplicación MVC se dividen en tres partes: Modelo, Vista y Control.

La capacidad de comunicarse con fuentes de datos externas, incluidas bases de datos y servicios web, se encuentra bajo responsabilidad del Modelo. Sirve como un puente utilizando interfaces adecuadas para manipular diferentes tipos de información relevante para nuestro dominio de destino mientras encapsula todas las reglas comerciales dentro de él.

La presentación de esta información en varios componentes gráficos como ventanas o botones se lleva a cabo por otra parte llamada Vista. Este se comunica con el controlador y recibe información por parte del modelo.

El componente intermediario crucial dentro de una arquitectura MVC es su capa de controlador: este elemento coordina la actividad entre los modelos y las vistas mientras centraliza la lógica utilizada para procesar las solicitudes de datos entrantes realizadas por los usuarios que interactúan con las aplicaciones de software diseñadas según los principios de MVC.

En MVC, el controlador administra las interacciones del usuario y organiza la comunicación entre el modelo y la vista, mientras que el modelo representa los datos y la lógica comercial y la vista sirve como interfaz de usuario.

En resumen como funciona es que el usuario interactúa con la vista a través del controlador, que luego actualiza el modelo y la vista en función de las acciones del usuario. Al final estos tres componentes funcionan en un proceso “cíclico”. En general esta arquitectura es útil porque genera modularidad en el código, y esto siempre es bueno porque es más fácil de encontrar errores en el programa; y mantener y modificar a futuro el programa.

## ***Resumen***

### **Dos cosas que no sabía y aprendió**

Aprendí a utilizar ASP.NET de forma general, nunca lo había utilizado y fue interesante intentar que todos los componentes funcionen de forma junta. Tampoco sabía cómo funcionaba C# ya que nunca lo había utilizado, aunque fue algo superficial fue interesante utilizar este nuevo lenguaje.

### **Una cosa que se le hizo difícil de realizar y explique por qué fue difícil.**

Lo más difícil fue desplegar la información ya que no tengo experiencia con cshtml y esto es algo tan básico que inclusive me fue difícil encontrar un buen tutorial sobre esto.

### **Una cosa que se le hizo fácil de realizar y explique por qué fue fácil.**

Lo más sencillo fue crear el esqueleto del modelo-vista-controlador, ya que era similar al ejemplo realizado en el laboratorio.

### **Indique cuánto tiempo tardó en realizar el laboratorio.**

Aproximadamente 3 horas.

## **Referencias**

Qué es MVC. (n.d.). Desarrollo Web.

<https://desarrolloweb.com/articulos/que-es-mvc.html>

Hernandez, R. D. (2021). El patrón modelo-vista-controlador: Arquitectura y frameworks explicados. *freeCodeCamp.org*.

<https://www.freecodecamp.org/espanol/news/el-modelo-de-arquitectura-view-controller-pattern/>

Rick-Anderson. (2022, September 21). *ASP.NET*. Microsoft Learn.

<https://learn.microsoft.com/es-es/aspnet/mvc/overview/older-versions-1/overview/asp-net-mvc-overview>