Desafío Practico N.1, Desarrollo de Software Empresarial

Oswaldo Alberto Valle Serrano VS161940

Estructura del sistema

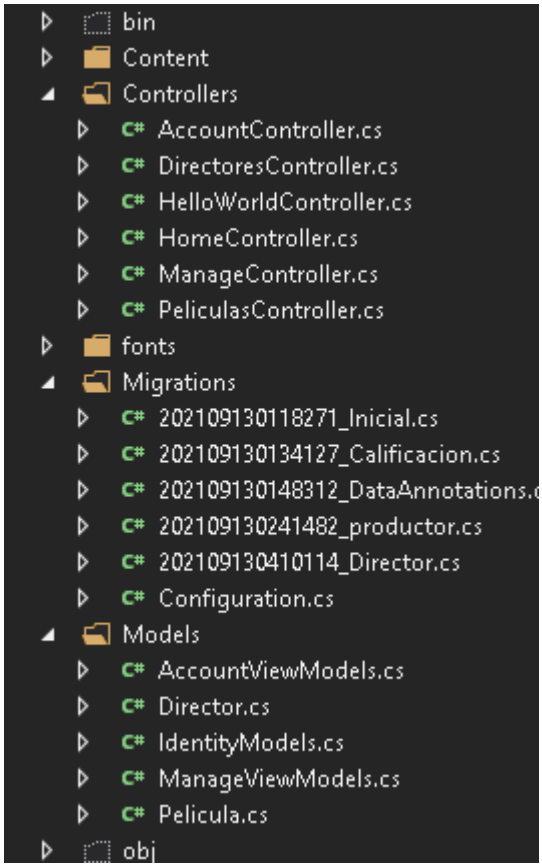


Figura 1: Controles y Modelos creados, con el uso de Code First para migrar la BDD

```
13  public class PeliculasController : Controller
14
15      private PeliculaDBContext db = new PeliculaDBContext();
16
17      // GET: Peliculas
        0 referencias
18      public ActionResult Index(string buscarString, string generoPelicula, string listaDirector)
19      {
20          var GeneroList = new List<string>();
21          var GeneroQry = from d in db.Peliculas
22                          orderby d.Genero
23                          select d.Genero;
24          GeneroList.AddRange(GeneroQry.Distinct());
25          ViewBag.generoPelicula = new SelectList(GeneroList, "accion");
26          ViewBag.generoPelicula = new SelectList(GeneroList);
27          var peliculas = from p in db.Peliculas
28                          select p;
29
30          var Genero2List = new List<string>();
31          var Genero2Qry = from d in db.Directors
32                          orderby d.Nombre
33                          select d.Nombre;
34          Genero2List.AddRange(Genero2Qry.Distinct());
35          ViewBag.listaDirector = new SelectList(Genero2List, "Luis");
36          ViewBag.listaDirector = new SelectList(Genero2List);
37
38          if (!String.IsNullOrEmpty(buscarString))
39          {
40              peliculas = peliculas.Where(s => s.Titulo.Contains(buscarString));
41          }
42          if (!string.IsNullOrEmpty(generoPelicula))
43          {
44              peliculas = peliculas.Where(x => x.Genero == generoPelicula);
45          }
46          if (!string.IsNullOrEmpty(listaDirector))
47          {
48              peliculas = peliculas.Where(y => y.productor == listaDirector);
49          }
50          return View(peliculas);
51      }
```

Figura 2, Controlador Película

```
Index.cshtml   Create.cshtml*   Pelicula.cs #   Configuration.cs #   DirectoresController.cs #   PeliculasController.cs #   20210913041014_Director.cs #
Guia1_DSE   Guia1_DSE.Controllers.DirectoresController   Details(int? id)
9   using MvcPelicula.Models;
10
11  namespace Guia1_DSE.Controllers
12  {
        0 referencias
13      public class DirectoresController : Controller
14      {
15          private PeliculaDBContext db = new PeliculaDBContext();
16
17          // GET: Directores
            0 referencias
18          public ActionResult Index()
19          {
20              return View(db.Directors.ToList());
21          }
22
23          // GET: Directores/Details/5
            0 referencias
24          public ActionResult Details(int? id)
25          {
26              if (id == null)
27              {
28                  return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
29              }
30              Director director = db.Directors.Find(id);
31              if (director == null)
32              {
33                  return HttpNotFound();
34              }
35              return View(director);
36          }
37
38          // GET: Directores/Create
            0 referencias
39          public ActionResult Create()
40          {
41              return View();
42          }
43
44          // POST: Directores/Create
45          // Para protegerse de ataques de publicación excesiva, habilite las propiedades específicas a las que desea enlazarse. Para obtener
```

Figura 3, Controlador para directores

Index.cshtml | Create.cshtml* | Pelicula.cs | Configuration.cs | DirectoresController.cs

Guia1_DSE | Guia1_DSE.Migrations.Configuration

```
        {
            AutomaticMigrationsEnabled = false;
        }

        0 referencias
        protected override void Seed(MvcPelicula.Models.PeliculaDBContext context)
        {
            context.Peliculas.AddOrUpdate(i => i.Titulo,
    new Pelicula
    {
        Titulo = "John Wick",
        FechaLanzamiento = DateTime.Parse("2014-10-24"),
        Genero = "Accion",
        Calificacion = "D",
        Precio = 6.99M,
        productor = "Keanu Rives"
    },
    new Pelicula
    {
        Titulo = "John Wick II",
        FechaLanzamiento = DateTime.Parse("2015-10-26"),
        Genero = "Accion",
        Calificacion = "D",
        Precio = 6.99M,
        productor = "Keanu Rives"
    },
    new Pelicula
    {
        Titulo = "John Wick III",
```

```
);

        context.Directors.AddOrUpdate(i => i.ID,
            new MvcPelicula.Models.Director
            {
                Nombre = "Luis Flores",
                Edad = 60
            }



        );



    }
```

Figura 4 y 5, Estructura de las tablas dentro de la migración.

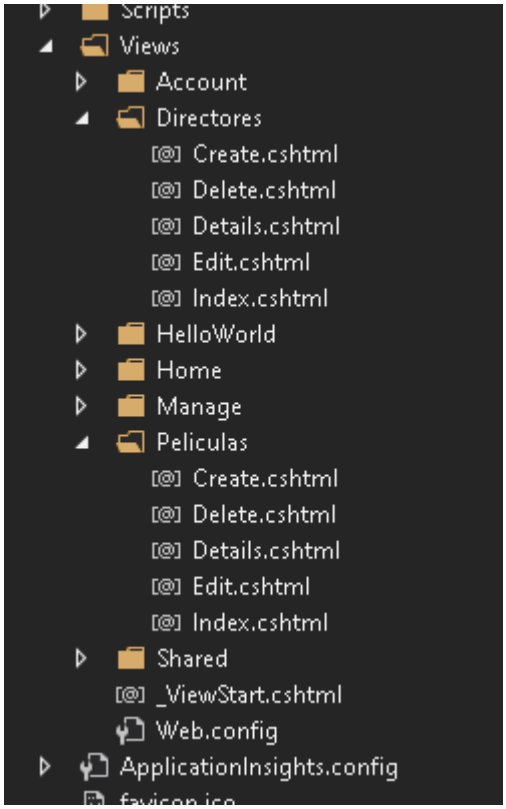


Figura 6, Carpetas View para CRUD de películas y Directores

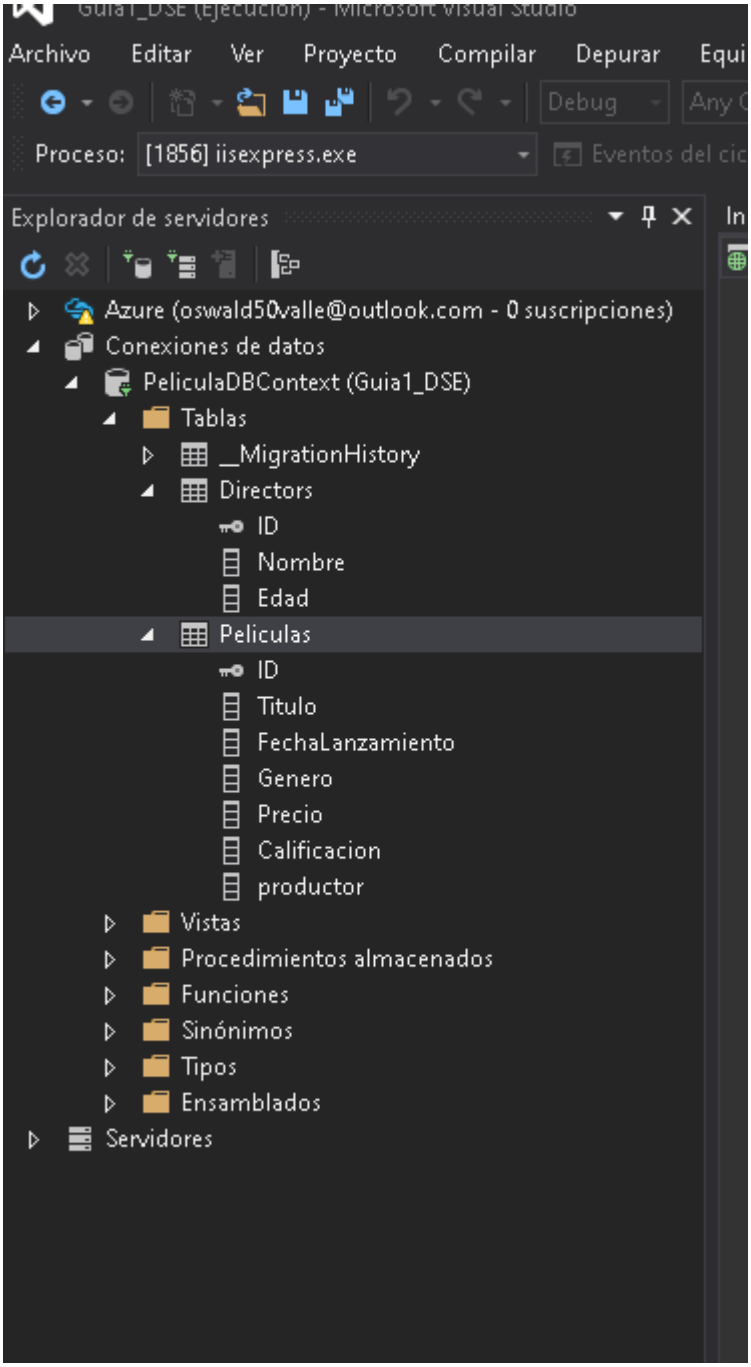


Figura 7, Base de datos con Code First.

Index.cshtml | Create.cshtml | Pelicula.cs | Configuration.cs | DirectoresController.cs | PeliculasCon

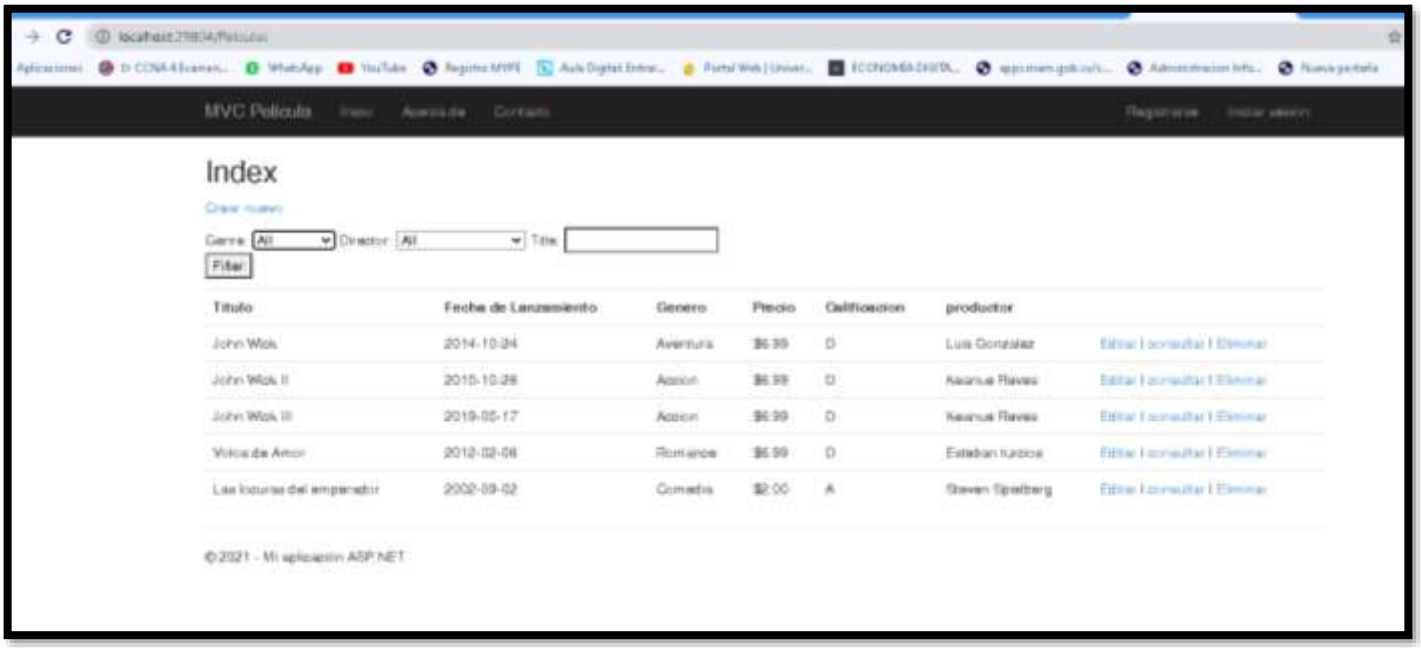
Guia1_DSE | MvcPelicula.Models.Pelicula | Calificacion

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Data.Entity;
using System.Web;
using System.ComponentModel.DataAnnotations;
namespace MvcPelicula.Models
{
    17 referencias
    public class Pelicula
    {
        4 referencias
        public int ID { get; set; }
        [StringLength(60, MinimumLength = 3)]

        14 referencias
        public string Titulo { get; set; }

        [Display(Name = "Fecha de Lanzamiento")]
        [DataType(DataType.Date)]
        [DisplayFormat(DataFormatString = "{0:yyyy-MM-dd}", ApplyFormatInEditMode
= true)]

        12 referencias
        public DateTime FechaLanzamiento { get; set; }
        [RegularExpression(@"^[A-Z]+[a-zA-Z'\s]*$")]
        [Required]
        [StringLength(30)]

        15 referencias
        public string Genero { get; set; }
        [Range(1, 100)]
        [DataType(DataType.Currency)]

        12 referencias
        public decimal Precio { get; set; }

        [RegularExpression(@"^[A-Z]+[a-zA-Z'\s]*$")]
        [StringLength(5)]
```

```
    15 referencias
    public string Genero { get; set; }
    [Range(1, 100)]
    [DataType(DataType.Currency)]

    12 referencias
    public decimal Precio { get; set; }

    [RegularExpression(@"^[A-Z]+[a-zA-Z'\s]*$")]
    [StringLength(5)]

    12 referencias
    public string Calificacion { get; set; }
    13 referencias
    public string productor { get; set; }
}
10 referencias
public class Director
{
    4 referencias
    public int ID { get; set; }

    5 referencias
    public string Nombre { get; set; }

    3 referencias
    public int Edad { get; set; }
}
6 referencias
public class PeliculaDBContext : DbContext
{
    9 referencias
    public DbSet<Pelicula> Peliculas { get; set; }

    9 referencias
    public System.Data.Entity.DbSet<MvcPelicula.Models.Director> Directors { get; set; }
}
```

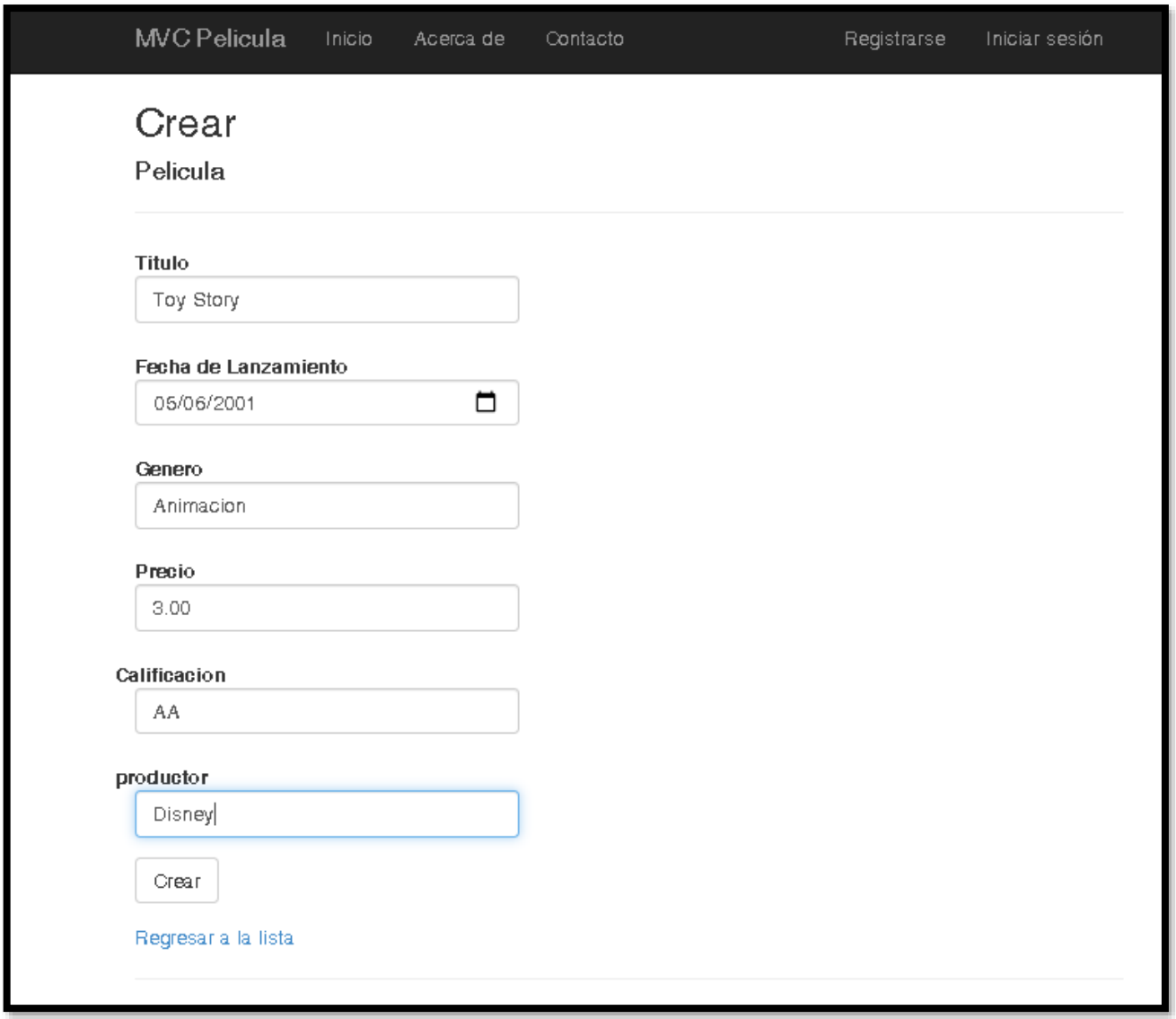
Figura 8 y 9, Modelo Películas, que incluye las clases Películas y Directores para la BDD

Funcionamiento del sistema



Como primera imagen podemos visualizar la página de películas, donde visualizamos las películas que hemos ingresado al sistema, a su vez esta información puede filtrarse de 3 maneras:

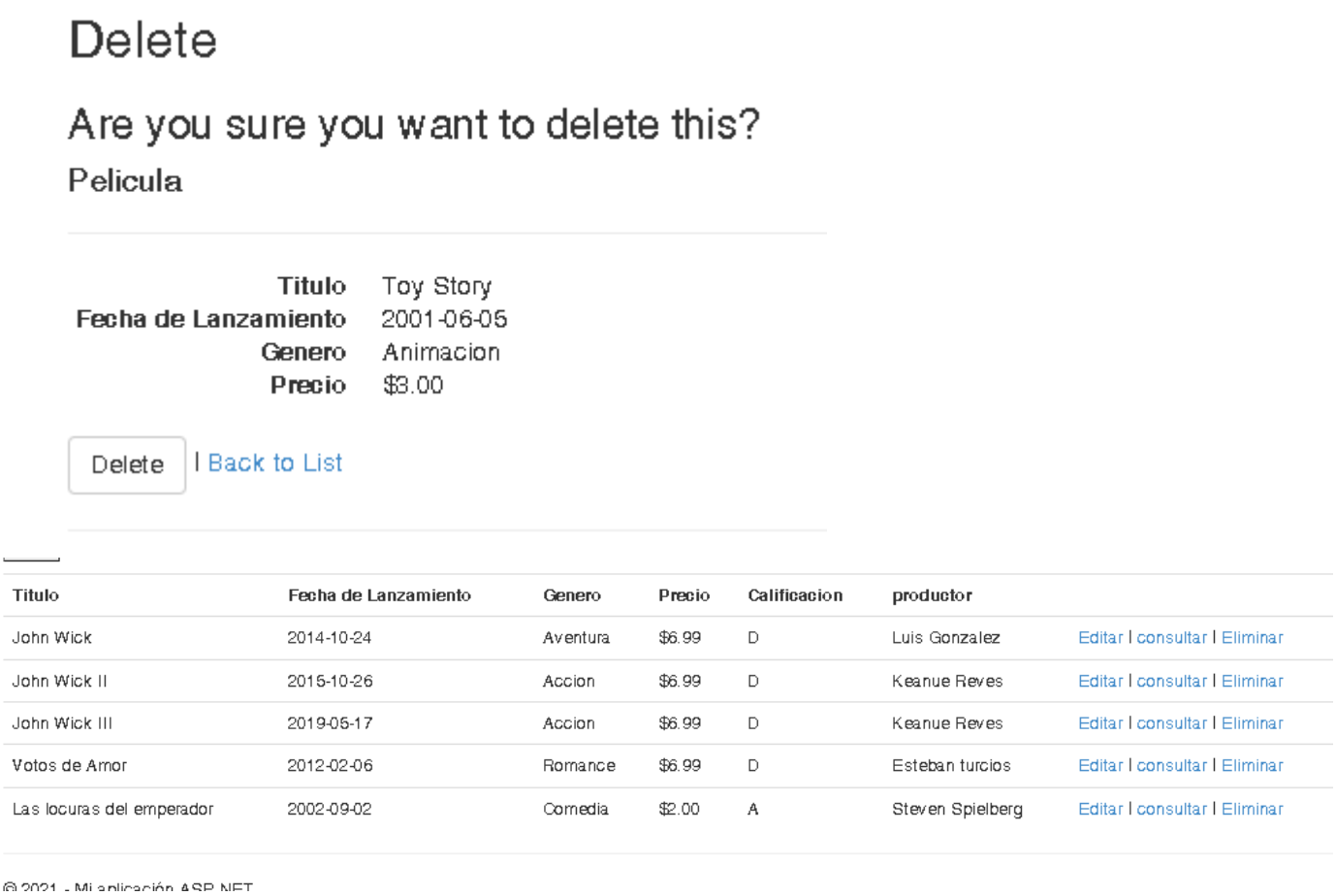
- Genero
- Director
- Título de la película



Al darle al botón Crear Nuevo, ingresaremos una película, completando los datos que se presentan en la imagen.

Nos regresara a la pantalla de inicio y podremos ver nuestra película ingresada

A su vez se puede modificar y consultar la información de la película.



Y así mismo eliminarse de la base de datos.

MVC Pelicula Inicio Acerca de Contacto

Productores

Create New

Nombre	Edad	
Luis Flores	60	Edit | Details | Delete
Michael	29	Edit | Details | Delete
Luis Gonzalez	49	Edit | Details | Delete
Keanue Reves	42	Edit | Details | Delete
Steven Spielberg	62	Edit | Details | Delete
Luis Esteban	49	Edit | Details | Delete

© 2021 - Mi aplicación ASP.NET

Así mismo existe una página CRUD para los Productores. En las siguientes imágenes se muestran su funcionamiento:

Create

Director

Nombre Armando Rodriguez

Edad 39

Create

Back to List

© 2021 - Mi aplicación ASP.NET

Details

Director

Nombre Armando Rodriguez

Edad 39

Edit | Back to List

© 2021 - Mi aplicación ASP.NET

Edit

Director

Nombre Carlos Rodriguez

Edad 39

Save

Back to List

© 2021 - Mi aplicación ASP.NET

MVC Pelicula Inicio Acerca de Contacto

Delete

Are you sure you want to delete this?

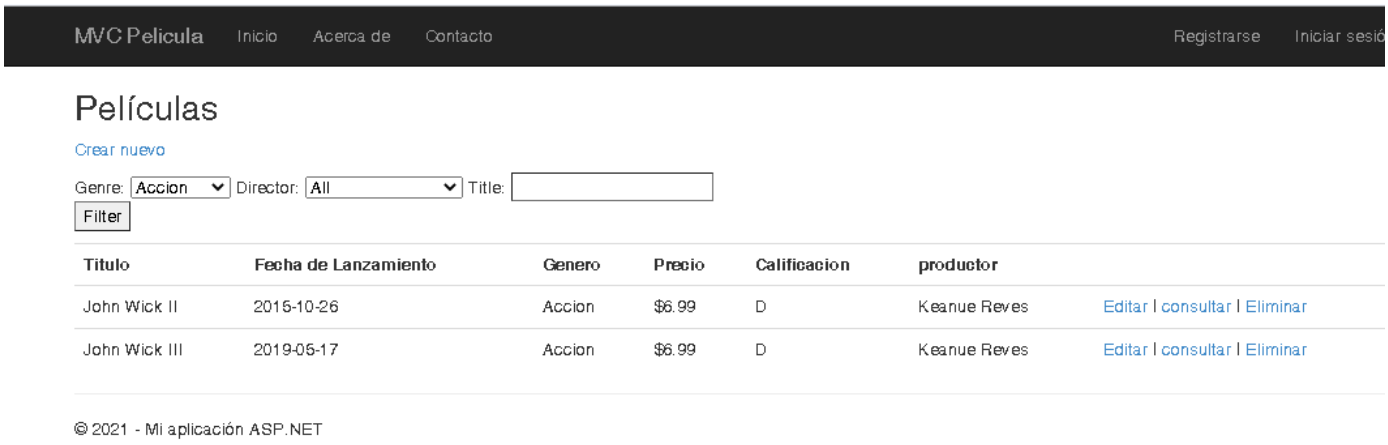
Director

Nombre Carlos Rodriguez

Edad 39

Delete | Back to List

© 2021 - Mi aplicación ASP.NET



Como se dijo anteriormente, la página puede filtrar por varias maneras, la primera es por el género de la película.

La segunda es por el Director, donde la lista se autocompleta con la lista de nombres de la base de datos.

La tercera es por el título de la película.

Y también se puede por multifiltrado de datos.