

Desafío práctico 1 – DSE

Nombre: José Ricardo Majano De Paz

Carnet: MD170149

Enlace del proyecto: <https://github.com/RicardoM99160/Guias-DSE/tree/master/Desafio%201/MvcPelícula>

Script Base de Datos

Modelos

Película.cs

```
using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Web;

namespace MvcPelícula.Models
{
    public class Película
    {
        public int ID { get; set; }

        [StringLength(60, MinimumLength = 3)]
        public string Titulo { get; set; }

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

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

        [Range(1, 100)]
        [DataType(DataType.Currency)]
        public decimal Precio { get; set; }

        [RegularExpression(@"^[A-Z]+[a-zA-Z'\s]*$")]
        [StringLength(5)]
        public string Calificacion { get; set; }

        [StringLength(60, MinimumLength = 3)]
        public string Productor { get; set; }

        public int? DirectorID { get; set; }

        public virtual Director Director { get; set; }
    }

    public class PelículaDbContext: DbContext
    {
        public DbSet<Película> Películas { get; set; }
        public DbSet<Director> Directores { get; set; }
    }
}
```

Director.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data.Entity;
using System.ComponentModel.DataAnnotations;

namespace MvcPelicula.Models
{
    public class Director
    {
        public int DirectorID { get; set; }

        [RegularExpression(@"^[A-Z]+[a-zA-Z'\s]*$")]
        [Required]
        [StringLength(60, MinimumLength = 3)]
        public string Nombre { get; set; }

        public virtual ICollection<Pelicula> Peliculas { get; set; }
    }
}
```

Script tablas

Directors

```
CREATE TABLE [dbo].[Directors] (
    [DirectorID] INT IDENTITY (1, 1) NOT NULL,
    [Nombre] NVARCHAR (60) NOT NULL,
    CONSTRAINT [PK_dbo.Directors] PRIMARY KEY CLUSTERED ([DirectorID] ASC)
);
```

Peliculas

```
CREATE TABLE [dbo].[Peliculas] (
    [ID] INT IDENTITY (1, 1) NOT NULL,
    [Titulo] NVARCHAR (60) NULL,
    [FechaLanzamiento] DATETIME NOT NULL,
    [Genero] NVARCHAR (30) NOT NULL,
    [Precio] DECIMAL (18, 2) NOT NULL,
    [Calificacion] NVARCHAR (5) NULL,
    [Productor] NVARCHAR (60) NULL,
    [DirectorID] INT NULL,
    CONSTRAINT [PK_dbo.Peliculas] PRIMARY KEY CLUSTERED ([ID] ASC),
    CONSTRAINT [FK_dbo.Peliculas_dbo.Directors_DirectorID] FOREIGN KEY ([DirectorID]) REFERENCES
[dbo].[Directors] ([DirectorID])
);
```

Migraciones

Configuration.cs

```
namespace MvcPelicula.Migrations
{
    using System;
    using System.Collections.Generic;
    using System.Data.Entity;
    using System.Data.Entity.Migrations;
    using System.Linq;
    using MvcPelicula.Models;

    internal sealed class Configuration : DbMigrationsConfiguration<MvcPelicula.Models.PeliculaDBContext>
    {
        public Configuration()
        {
            AutomaticMigrationsEnabled = false;
        }

        protected override void Seed(MvcPelicula.Models.PeliculaDBContext context)
        {
            var directores = new List<Director>
            {
                new Director
                {
                    Nombre = "Chad Stahelski"
                },
                new Director
                {
                    Nombre = "Michael Sucsy"
                },
                new Director
                {
                    Nombre = "Brad Silberling"
                }
            };

            directores.ForEach(p => context.Directores.AddOrUpdate(d => d.Nombre, p));
            context.SaveChanges();

            context.Peliculas.AddOrUpdate(i => i.Titulo,
                new Pelicula
                {
                    Titulo = "John Wick", FechaLanzamiento = DateTime.Parse("2014-10-24"), Genero = "Accion", Precio = 6.99M,
                    Calificacion = "D", Productor = "David Leicht",
                    DirectorID = directores.Single(s => s.Nombre == "Chad Stahelski").DirectorID, Director = directores.Single(s
=> s.Nombre == "Chad Stahelski")
                },
                new Pelicula
                {
                    Titulo = "John Wick II", FechaLanzamiento = DateTime.Parse("2015-10-26"), Genero = "Accion", Precio = 6.99M,
                    Calificacion = "D", Productor = "David Leicht",
                    DirectorID = directores.Single(s => s.Nombre == "Chad Stahelski").DirectorID, Director = directores.Single(s
=> s.Nombre == "Chad Stahelski")
                },
                new Pelicula
                {
                    Titulo = "John Wick III", FechaLanzamiento = DateTime.Parse("2019-05-17"), Genero = "Accion", Precio = 6.99M,
                    Calificacion = "D", Productor = "Basil Iwanyk",
                    DirectorID = directores.Single(s => s.Nombre == "Chad Stahelski").DirectorID, Director = directores.Single(s
=> s.Nombre == "Chad Stahelski")
                },
                new Pelicula
                {
                    Titulo = "Votos de Amor", FechaLanzamiento = DateTime.Parse("2012-02-06"), Genero = "Romance", Precio =
6.99M, Calificacion = "A", Productor = "Jonathan Glickman",
                    DirectorID = directores.Single(s => s.Nombre == "Michael Sucsy").DirectorID, Director = directores.Single(s
=> s.Nombre == "Michael Sucsy")
                }
            );
        }
    }
}
```

Lógica de negocios

Controladores

DirectoresController.cs

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using MvcPelicula.Models;

namespace MvcPelicula.Controllers
{
    public class DirectoresController : Controller
    {
        private PeliculaDBContext db = new PeliculaDBContext();

        // GET: Directores
        public ActionResult Index()
        {
            return View(db.Directores.ToList());
        }

        // GET: Directores/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Director director = db.Directores.Find(id);
            if (director == null)
            {
                return HttpNotFound();
            }
            return View(director);
        }

        // GET: Directores/Create
        public ActionResult Create()
        {
            return View();
        }

        // POST: Directores/Create
        // Para protegerse de ataques de publicación excesiva, habilite las propiedades específicas a las que
        desea enlazarse. Para obtener
        // más información vea https://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include = "DirectorID,Nombre")] Director director)
        {
            if (ModelState.IsValid)
            {
                db.Directores.Add(director);
                db.SaveChanges();
                return RedirectToAction("Index");
            }

            return View(director);
        }

        // GET: Directores/Edit/5
```

```

public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Director director = db.Directores.Find(id);
    if (director == null)
    {
        return HttpNotFound();
    }
    return View(director);
}

// POST: Directores/Edit/5
// Para protegerse de ataques de publicación excesiva, habilite las propiedades específicas a las que
desea enlazarse. Para obtener
// más información vea https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include = "DirectorID,Nombre")] Director director)
{
    if (ModelState.IsValid)
    {
        db.Entry(director).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(director);
}

// GET: Directores/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Director director = db.Directores.Find(id);
    if (director == null)
    {
        return HttpNotFound();
    }
    return View(director);
}

// POST: Directores/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    Director director = db.Directores.Find(id);
    db.Directores.Remove(director);
    db.SaveChanges();
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
}

```

PeliculasController.cs

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using MvcPelicula.Models;

namespace MvcPelicula.Controllers
{
    public class PeliculasController : Controller
    {
        private PeliculaDbContext db = new PeliculaDbContext();

        // GET: Peliculas
        public ActionResult Index(string buscarString, string generoPelicula, string precioPelicula,
            string directorPelicula)
        {
            var directores = from d in db.Directores select d;
            var directoresLst = new List<Director>();
            directoresLst.AddRange(directores.Distinct());

            var peliculas = from p in db.Peliculas select p;

            foreach (var pelicula in peliculas)
            {
                pelicula.Director = directoresLst.Single(s => s.DirectorID == pelicula.DirectorID);
            }

            var GeneroLst = new List<string>();
            var GeneroQry = from d in db.Peliculas orderby d.Genero select d.Genero;
            GeneroLst.AddRange(GeneroQry.Distinct());
            ViewBag.generoPelicula = new SelectList(GeneroLst, "Acción");

            var fromDatabaseEF = new SelectList(db.Directores.ToList(), "DirectorID", "Nombre");
            ViewBag.directorPelicula = fromDatabaseEF;

            if (!String.IsNullOrEmpty(buscarString))
            {
                peliculas = peliculas.Where(s => s.Titulo.Contains(buscarString));
            }

            if (!String.IsNullOrEmpty(generoPelicula))
            {
                peliculas = peliculas.Where(x => x.Genero == generoPelicula);
            }

            if (!String.IsNullOrEmpty(precioPelicula))
            {
                decimal precio = Decimal.Parse(precioPelicula);
                peliculas = peliculas.Where(y => y.Precio == precio);
            }

            if (!String.IsNullOrEmpty(directorPelicula))
            {
                int directorID = int.Parse(directorPelicula);
                peliculas = peliculas.Where(x => x.DirectorID == directorID);
            }
        }
    }
}
```

```

        return View(peliculas);
    }

    // GET: Peliculas/Details/5
    public ActionResult Details(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Pelicula pelicula = db.Peliculas.Find(id);

        Director director = db.Directores.Find(pelicula.DirectorID);
        ViewData["director"] = director.Nombre;

        if (pelicula == null)
        {
            return HttpNotFound();
        }
        return View(pelicula);
    }

    // GET: Peliculas/Create
    public ActionResult Create()
    {
        var directores = from d in db.Directores select d.Nombre;
        var fromDatabaseEF = new SelectList(db.Directores.ToList(), "DirectorID", "Nombre");
        ViewData["directores"] = fromDatabaseEF;

        return View();
    }

    // POST: Peliculas/Create
    // Para protegerse de ataques de publicación excesiva, habilite las propiedades específicas a
    las que desea enlazarse. Para obtener
    // más información vea https://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Create([Bind(Include =
    "ID,Titulo,FechaLanzamiento,Genero,Precio,Calificacion,Productor")] Pelicula pelicula, int directorID)
    {
        var directores = from d in db.Directores select d;
        var directoresLst = new List<Director>();
        directoresLst.AddRange(directores.Distinct());

        Director directorSelec = directoresLst.Single(s => s.DirectorID == directorID);
        pelicula.Director = directorSelec;
        pelicula.DirectorID = directorSelec.DirectorID;

        if (ModelState.IsValid)
        {
            db.Peliculas.Add(pelicula);
            db.SaveChanges();
            return RedirectToAction("Index");
        }

        return View(pelicula);
    }

    // GET: Peliculas/Edit/5
    public ActionResult Edit(int? id)
    {
        if (id == null)

```

```

    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }

    Pelicula pelicula = db.Peliculas.Find(id);

    Director director = db.Directores.Find(pelicula.DirectorID);
    pelicula.Director = director;

    var directores = from d in db.Directores select d.Nombre;
    var fromDatabaseEF = new SelectList(db.Directores.ToList(), "DirectorID", "Nombre",
director.Nombre);
    ViewData["directores"] = fromDatabaseEF;

    if (pelicula == null)
    {
        return HttpNotFound();
    }
    return View(pelicula);
}

// POST: Peliculas/Edit/5
// Para protegerse de ataques de publicación excesiva, habilite las propiedades específicas a
las que desea enlazarse. Para obtener
// más información vea https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
"ID,Titulo,FechaLanzamiento,Genero,Precio,Calificacion,Productor")] Pelicula pelicula, int directorID)
{
    var directores = from d in db.Directores select d;
    var directoresLst = new List<Director>();
    directoresLst.AddRange(directores.Distinct());

    Director directorSelec = directoresLst.Single(s => s.DirectorID == directorID);
    pelicula.Director = directorSelec;
    pelicula.DirectorID = directorSelec.DirectorID;

    if (ModelState.IsValid)
    {
        db.Entry(pelicula).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(pelicula);
}

// GET: Peliculas/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Pelicula pelicula = db.Peliculas.Find(id);

    Director director = db.Directores.Find(pelicula.DirectorID);
    ViewData["director"] = director.Nombre;

    if (pelicula == null)
    {
        return HttpNotFound();
    }
}

```



```

    }
    return View(pelicula);
}

// POST: Peliculas/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    Pelicula pelicula = db.Peliculas.Find(id);
    db.Peliculas.Remove(pelicula);
    db.SaveChanges();
    return RedirectToAction("Index");
}

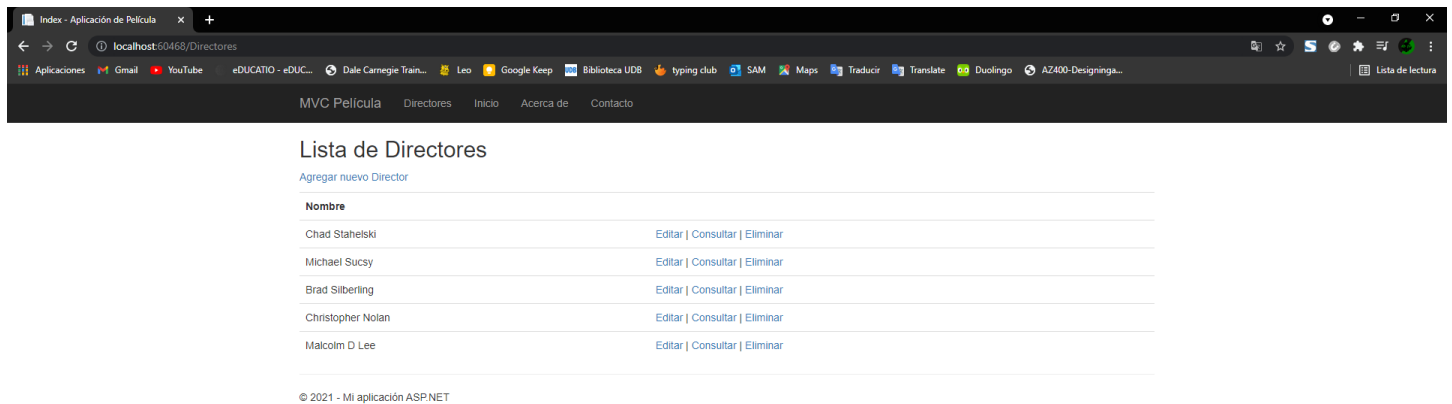
protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
}
}

```

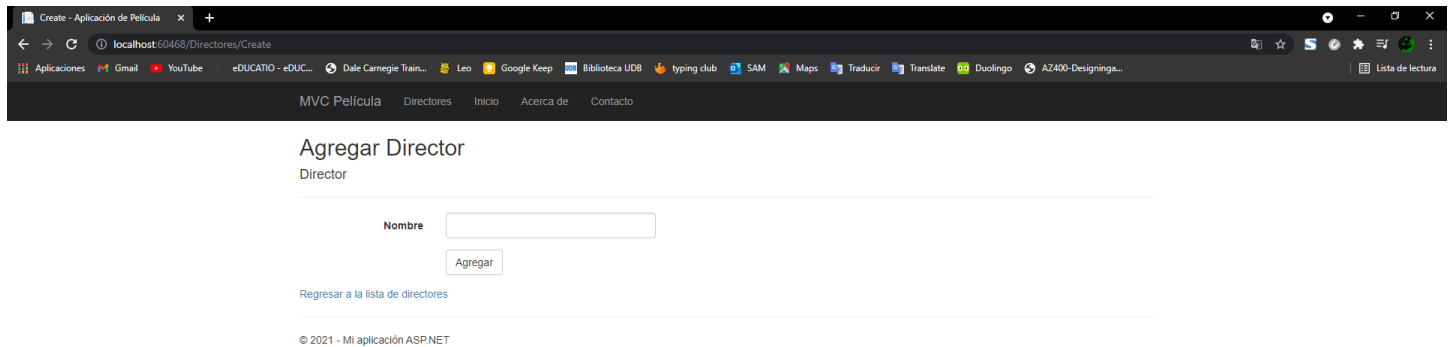
Capturas de pantalla

CRUD Directores

Index



Agregar



Consultar

Details - Aplicación de Película

localhost:60468/Directores/Details/7

Aplicaciones Gmail YouTube eDUCATIO - eDUC... Dale Carnegie Train... Leo Google Keep Biblioteca UDB typing club SAM Maps Traducir Translate Duolingo AZ400-Designing...

MVC Película Directores Inicio Acerca de Contacto

Detalles

Director

Nombre	Malcolm D Lee
--------	---------------

[Editar](#) | [Regresar a la lista de Directores](#)

© 2021 - Mi aplicación ASP.NET

Editar

Edit - Aplicación de Película

localhost:60468/Directores/Edit/7

Aplicaciones Gmail YouTube eDUCATIO - eDUC... Dale Carnegie Train... Leo Google Keep Biblioteca UDB typing club SAM Maps Traducir Translate Duolingo AZ400-Designing...

MVC Película Directores Inicio Acerca de Contacto

Editar Director

Director

Nombre	<input type="text" value="Malcolm D Lee"/>
--------	--

[Regresar a la lista de Directores](#)

© 2021 - Mi aplicación ASP.NET

Eliminar

Delete - Aplicación de Película

localhost:60468/Directores/Delete/7

Aplicaciones Gmail YouTube eDUCATIO - eDUC... Dale Carnegie Train... Leo Google Keep Biblioteca UDB typing club SAM Maps Traducir Translate Duolingo AZ400-Designing...

MVC Película Directores Inicio Acerca de Contacto

Eliminar Director

¿Estas seguro de eliminar a este Director?

Director

Nombre	Malcolm D Lee
--------	---------------

| [Regresar a la lista de Directores](#)

© 2021 - Mi aplicación ASP.NET

CRUD películas

Index

Index - Aplicación de Película

localhost:60468/Películas

Aplicaciones Gmail YouTube eDUCATIO - eDUC... Dale Carnegie Train... Leo Google Keep Biblioteca UDB typing club SAM Maps Traducir Translate Duolingo AZ400-Designing...

MVC Película Directores Inicio Acerca de Contacto

Agregar nueva

Género: Todos

Título:

Precio:

Director: Todos

Filtrar

Título	Fecha de Lanzamiento	Genero	Precio	Calificacion	Productor	Director	
John Wick	2014-10-24	Accion	\$6.99	D	David Leicht	Chad Stahelski	Editar Consultar Eliminar
John Wick II	2015-10-26	Accion	\$6.99	D	David Leicht	Chad Stahelski	Editar Consultar Eliminar
John Wick III	2019-05-17	Accion	\$6.99	D	Basil Iwanyk	Chad Stahelski	Editar Consultar Eliminar
Votos de Amor	2012-02-06	Romance	\$6.99	A	Jonathan Glickman	Michael Sucsy	Editar Consultar Eliminar
Ciudad de Angeles	1998-04-10	Romance	\$6.99	A	Robert Cavallo	Brad Silberling	Editar Consultar Eliminar
Interstellar	2014-10-26	Ciencia Ficcio	\$99.99	D	Emma Thomas	Christopher Nolan	Editar Consultar Eliminar
Space Jam: A New Legacy	2021-07-16	Comedia	\$1.00	A	Ryan Coogler	Malcolm D Lee	Editar Consultar Eliminar

© 2021 - Mi aplicación ASP.NET

Agregar

Create - Aplicación de Película

localhost:60468/Películas/Create

Aplicaciones Gmail YouTube eDUCATIO - eDUC... Dale Carnegie Train... Leo Google Keep Biblioteca UDB typing club SAM Maps Traducir Translate Duolingo AZ400-Designing...

MVC Película Directores Inicio Acerca de Contacto

Agregar Nueva Película

Película

Título

Fecha de Lanzamiento

dd/mm/aaaa

Genero

Precio

Calificacion

Productor

Director

Chad Stahelski

Agregar

[Regresar a la lista de películas](#)

© 2021 - Mi aplicación ASP.NET

Consultar

Details - Aplicación de Película

localhost:60468/Películas/Details/11

Aplicaciones Gmail YouTube eDUCATIO - eDUC... Dale Carnegie Train... Leo Google Keep Biblioteca UDB typing club SAM Maps Traducir Translate Duolingo AZ400-Designing...

MVC Película Directores Inicio Acerca de Contacto

Detalles

Película

Título

Fecha de Lanzamiento

Genero

Precio

Calificacion

Productor

Director

Interstellar

2014-10-26

Ciencia Ficcio

\$99.99

D

Emma Thomas

Christopher Nolan

[Editar](#) | [Regresar a la lista de películas](#)

© 2021 - Mi aplicación ASP.NET

Editar

Edit - Aplicación de Película

localhost:60468/Películas/Edit/11

Aplicaciones Gmail YouTube eDUCATIO - eDUC... Dale Carnegie Train... Leo Google Keep Biblioteca UDB typing club SAM Maps Traducir Translate Duolingo AZ400-Designing...

MVC Película Directores Inicio Acerca de Contacto

Editar

Película

Título

Interstellar

Fecha de Lanzamiento

26/10/2014

Genero

Ciencia Ficción

Precio

99.99

Calificación

D

Productor

Emma Thomas

Director

Christopher Nolan

Guardar

Regresar a la lista de películas

© 2021 - Mi aplicación ASP.NET

Eliminar

Delete - Aplicación de Película

localhost:60468/Películas/Delete/11

Aplicaciones Gmail YouTube eDUCATIO - eDUC... Dale Carnegie Train... Leo Google Keep Biblioteca UDB typing club SAM Maps Traducir Translate Duolingo AZ400-Designing...

MVC Película Directores Inicio Acerca de Contacto

Eliminar

¿Está seguro de eliminar esta película?

Película

Título

Interstellar

Fecha de Lanzamiento

2014-10-26

Genero

Ciencia Ficción

Precio

\$99.99

Calificación

D

Productor

Emma Thomas

Director

Christopher Nolan

Delete

Regresar a la lista de películas

© 2021 - Mi aplicación ASP.NET

Funcionamiento del sistema

1. Agregar Director

Create - Aplicación de Película

localhost:60468/Directores/Create

Aplicaciones Gmail YouTube eDUCATIO - eDUC... Dale Carnegie Train... Leo Google Keep Biblioteca UDB typing club SAM Maps Traducir Translate Duolingo AZ400-Designing...

MVC Película Directores Inicio Acerca de Contacto

Agregar Director

Director

Nombre

Ridley Scott

Agregar

Regresar a la lista de directores

© 2021 - Mi aplicación ASP.NET

Index - Aplicación de Película

localhost:60468/Directores

AplicacionesGmailYouTubeeDUCATIO - eDUC...Dale Carnegie Train...LeoGoogle KeepBiblioteca UDBtyping clubSAMMapsTraducirTranslateDuolingoAZ400-Designing...

MVC PelículaDirectoresInicioAcerca deContacto

Lista de Directores

[Agregar nuevo Director](#)

Nombre	
Chad Stahelski	Editar Consultar Eliminar
Michael Sucsy	Editar Consultar Eliminar
Brad Silberling	Editar Consultar Eliminar
Christopher Nolan	Editar Consultar Eliminar
Malcolm D Lee	Editar Consultar Eliminar
Ridley Scott	Editar Consultar Eliminar

© 2021 - Mi aplicación ASP.NET

2. Agregar película

Create - Aplicación de Película

localhost:60468/Peliculas/Create

AplicacionesGmailYouTubeeDUCATIO - eDUC...Dale Carnegie Train...LeoGoogle KeepBiblioteca UDBtyping clubSAMMapsTraducirTranslateDuolingoAZ400-Designing...

MVC PelículaDirectoresInicioAcerca deContacto

Agregar Nueva Película

Película

Título

Mision Rescate

Fecha de Lanzamiento

02/10/2015

Genero

Aventura

Precio

50.0

Calificacion

C

Productor

Simon Kinberg

Director

Ridley Scott

Agregar

[Regresar a la lista de películas](#)

© 2021 - Mi aplicación ASP.NET

Index - Aplicación de Película

localhost:60468/Peliculas

AplicacionesGmailYouTubeeDUCATIO - eDUC...Dale Carnegie Train...LeoGoogle KeepBiblioteca UDBtyping clubSAMMapsTraducirTranslateDuolingoAZ400-Designing...

MVC PelículaDirectoresInicioAcerca deContacto

Index

[Agregar nueva](#)

Genero: Todos

Título:

Precio:

Director: Todos

Filtrar

Titulo	Fecha de Lanzamiento	Genero	Precio	Calificacion	Productor	Director	
John Wick	2014-10-24	Accion	\$6.99	D	David Leicht	Chad Stahelski	Editar Consultar Eliminar
John Wick II	2015-10-26	Accion	\$6.99	D	David Leicht	Chad Stahelski	Editar Consultar Eliminar
John Wick III	2019-05-17	Accion	\$6.99	D	Basil Iwanyk	Chad Stahelski	Editar Consultar Eliminar
Votos de Amor	2012-02-06	Romance	\$6.99	A	Jonathan Glickman	Michael Sucsy	Editar Consultar Eliminar
Ciudad de Angeles	1998-04-10	Romance	\$6.99	A	Robert Cavallo	Brad Silberling	Editar Consultar Eliminar
Interstellar	2014-10-26	Ciencia Ficción	\$99.99	D	Emma Thomas	Christopher Nolan	Editar Consultar Eliminar
Space Jam: A New Legacy	2021-07-16	Comedia	\$1.00	A	Ryan Coogler	Malcolm D Lee	Editar Consultar Eliminar
Mision Rescate	2015-10-02	Aventura	\$50.00	C	Simon Kinberg	Ridley Scott	Editar Consultar Eliminar

© 2021 - Mi aplicación ASP.NET

Funcionamiento de Filtros

Listado de películas por Director

Index - Aplicación de Película

localhost:60468/Peliculas?generoPelicula=&buscarString=&precioPelicula=&directorPelicula=6

Aplicaciones Gmail YouTube eDUCATIO - eDUC... Dale Carnegie Train... Leo Google Keep Biblioteca UD8 typing club SAM Maps Traducir Translate Duolingo AZ400-Designinga...

MVC Película Directores Inicio Acerca de Contacto

Index

Agregar nueva

Género: Todos Titulo:

Precio:

Director: Christopher Nolan

Filtrar

Titulo	Fecha de Lanzamiento	Genero	Precio	Calificacion	Productor	Director	
Interestelar	2014-10-26	Ciencia Ficción	\$99.99	D	Emma Thomas	Christopher Nolan	Editar Consultar Eliminar
Inception	2010-07-09	Drama	\$99.99	C	Emma Thomas	Christopher Nolan	Editar Consultar Eliminar

© 2021 - Mi aplicación ASP.NET

Listado de películas por Director y por Género

Index - Aplicación de Película

localhost:60468/Peliculas?generoPelicula=Ciencia+Ficcion&buscarString=&precioPelicula=&directorPelicula=6

Aplicaciones Gmail YouTube eDUCATIO - eDUC... Dale Carnegie Train... Leo Google Keep Biblioteca UD8 typing club SAM Maps Traducir Translate Duolingo AZ400-Designinga...

MVC Película Directores Inicio Acerca de Contacto

Index

Agregar nueva

Género: Ciencia Ficción Titulo:

Precio:

Director: Christopher Nolan

Filtrar

Titulo	Fecha de Lanzamiento	Genero	Precio	Calificacion	Productor	Director	
Interestelar	2014-10-26	Ciencia Ficción	\$99.99	D	Emma Thomas	Christopher Nolan	Editar Consultar Eliminar

© 2021 - Mi aplicación ASP.NET

Index - Aplicación de Película

localhost:60468/Peliculas?generoPelicula=Drama&buscarString=&precioPelicula=&directorPelicula=6

Aplicaciones Gmail YouTube eDUCATIO - eDUC... Dale Carnegie Train... Leo Google Keep Biblioteca UD8 typing club SAM Maps Traducir Translate Duolingo AZ400-Designinga...

MVC Película Directores Inicio Acerca de Contacto

Index

Agregar nueva

Género: Drama Titulo:

Precio:

Director: Christopher Nolan

Filtrar

Titulo	Fecha de Lanzamiento	Genero	Precio	Calificacion	Productor	Director	
Inception	2010-07-09	Drama	\$99.99	C	Emma Thomas	Christopher Nolan	Editar Consultar Eliminar

© 2021 - Mi aplicación ASP.NET