**Desafío práctio 1 – DSE**

**Nombre:** José Ricardo Majano De Paz **Carnet:** MD170149

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

# Script Base de Datos

## Modelos

### Pelicula.cs

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Web;

namespace MvcPelicula.Models

{

public class Pelicula

{

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 PeliculaDBContext: DbContext

{

public DbSet<Pelicula> Peliculas { 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);

}

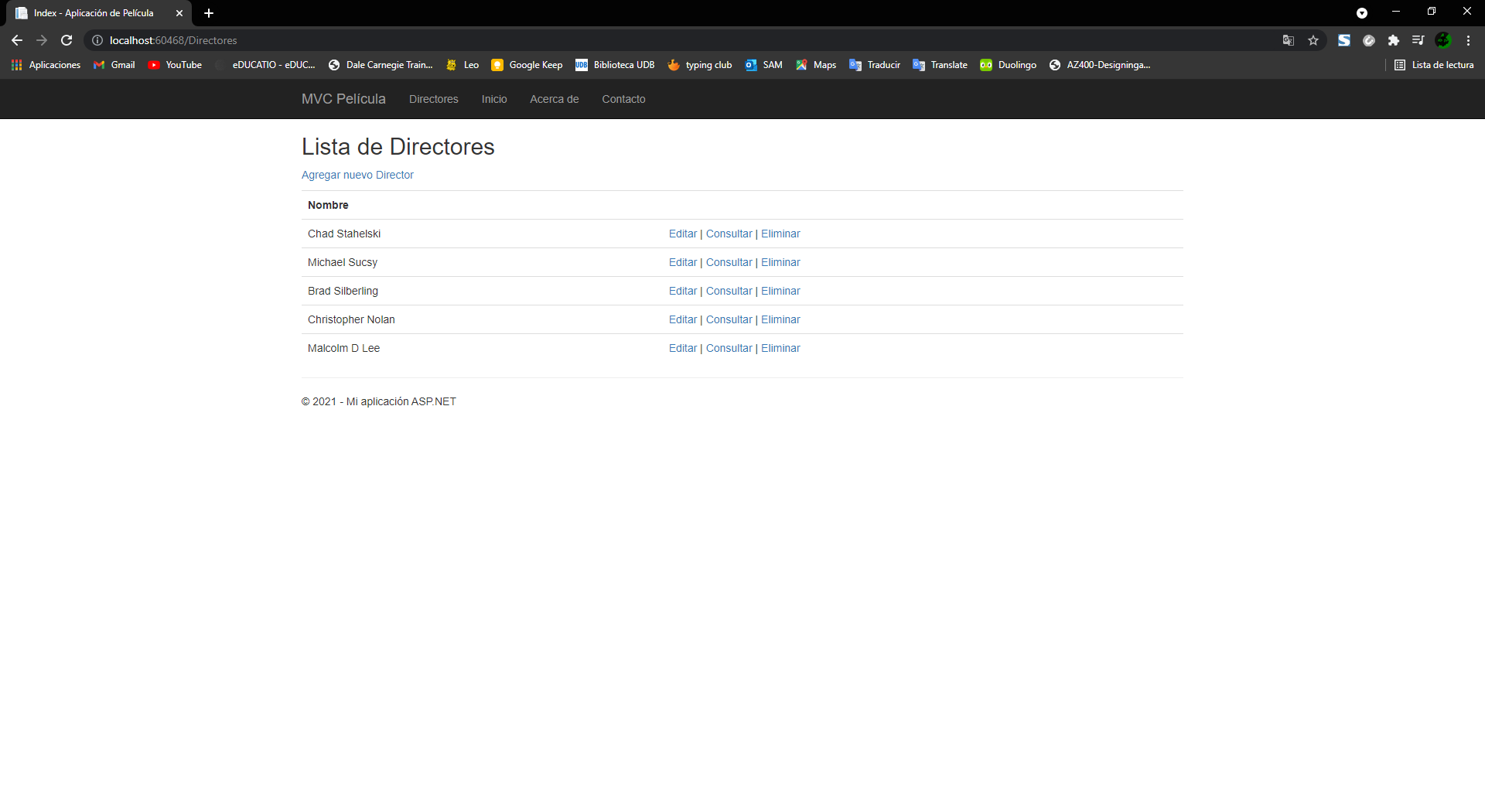
}

}

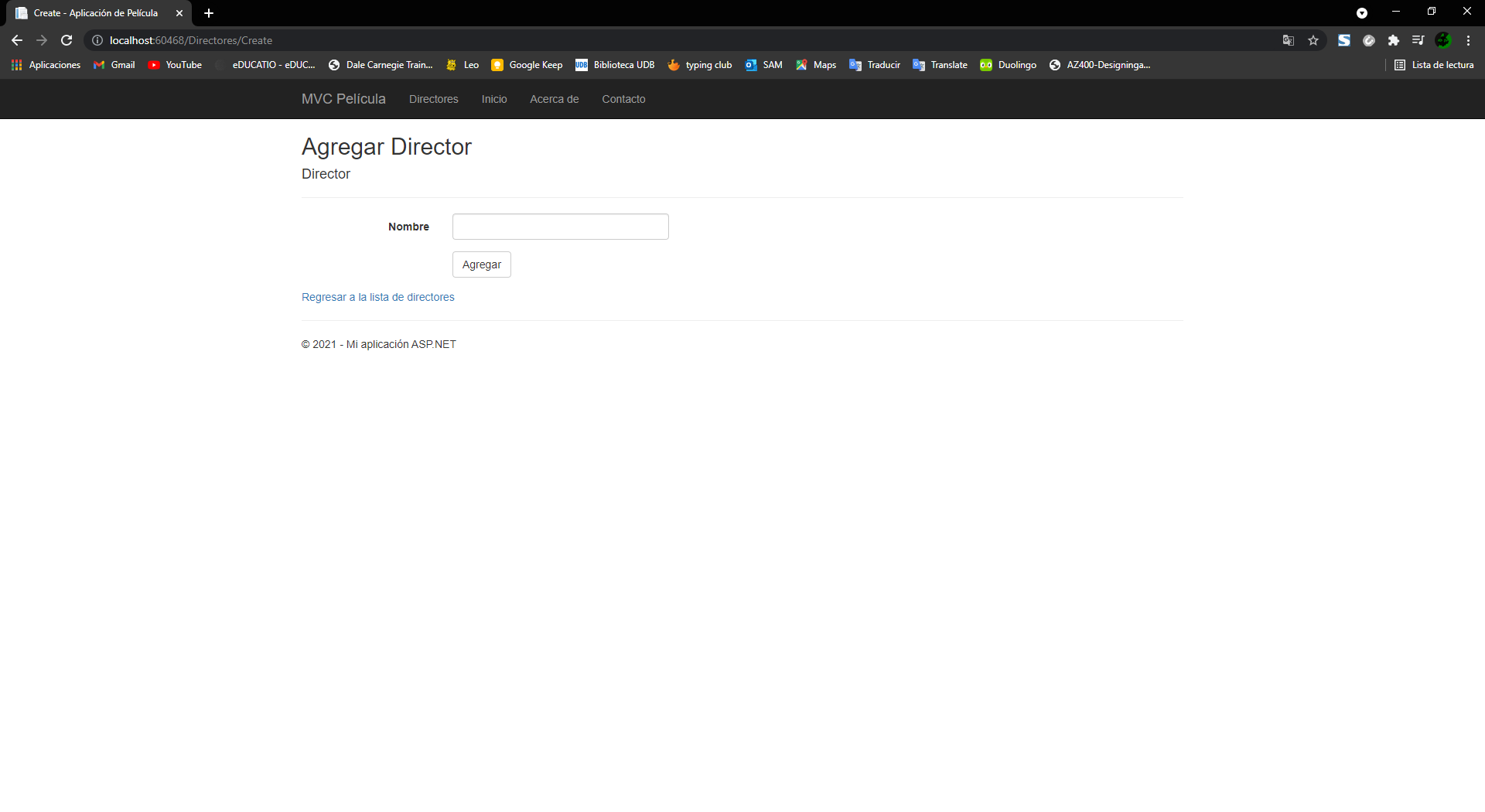
# Capturas de pantalla

## CRUD Directores

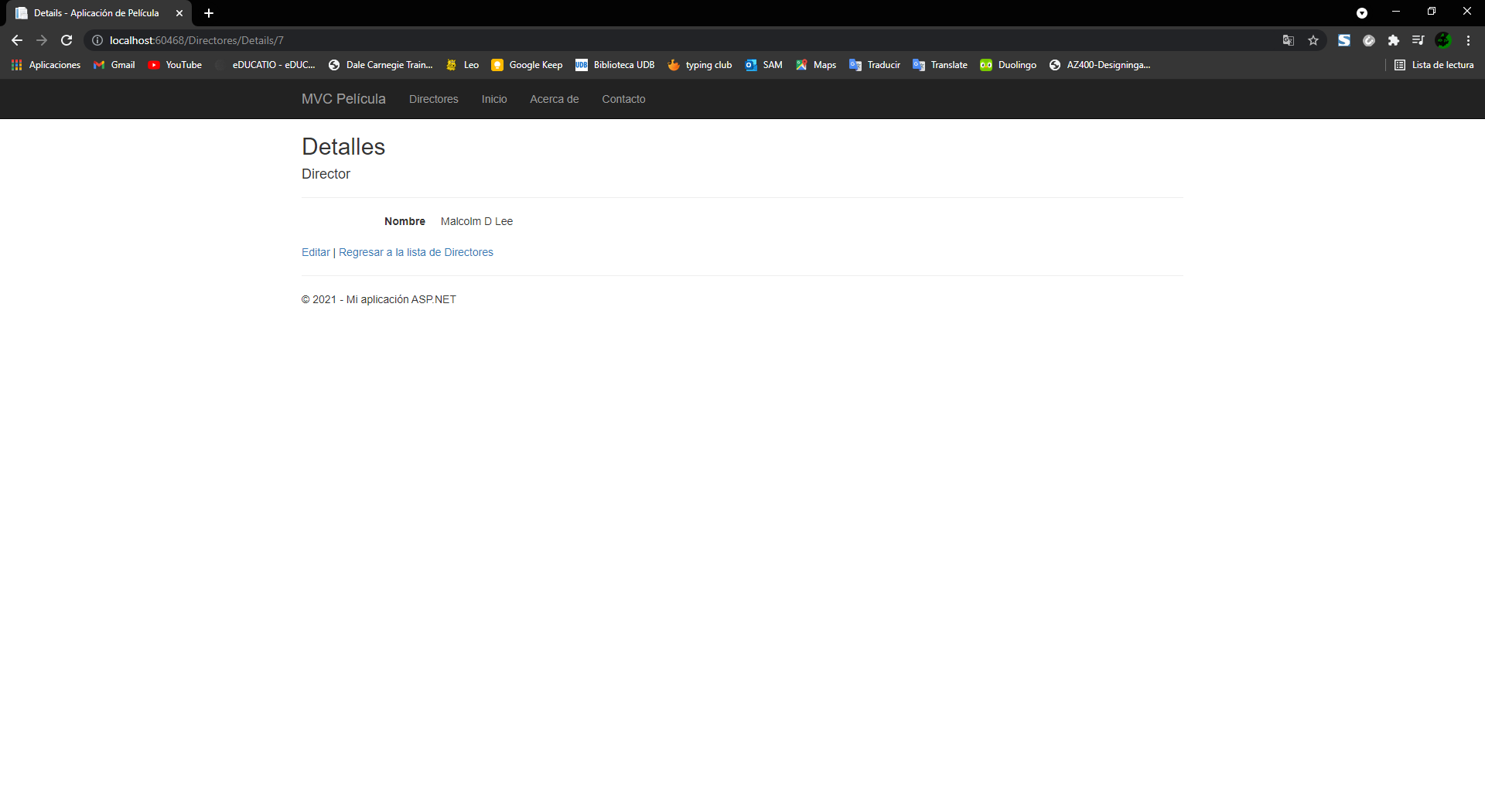
### Index



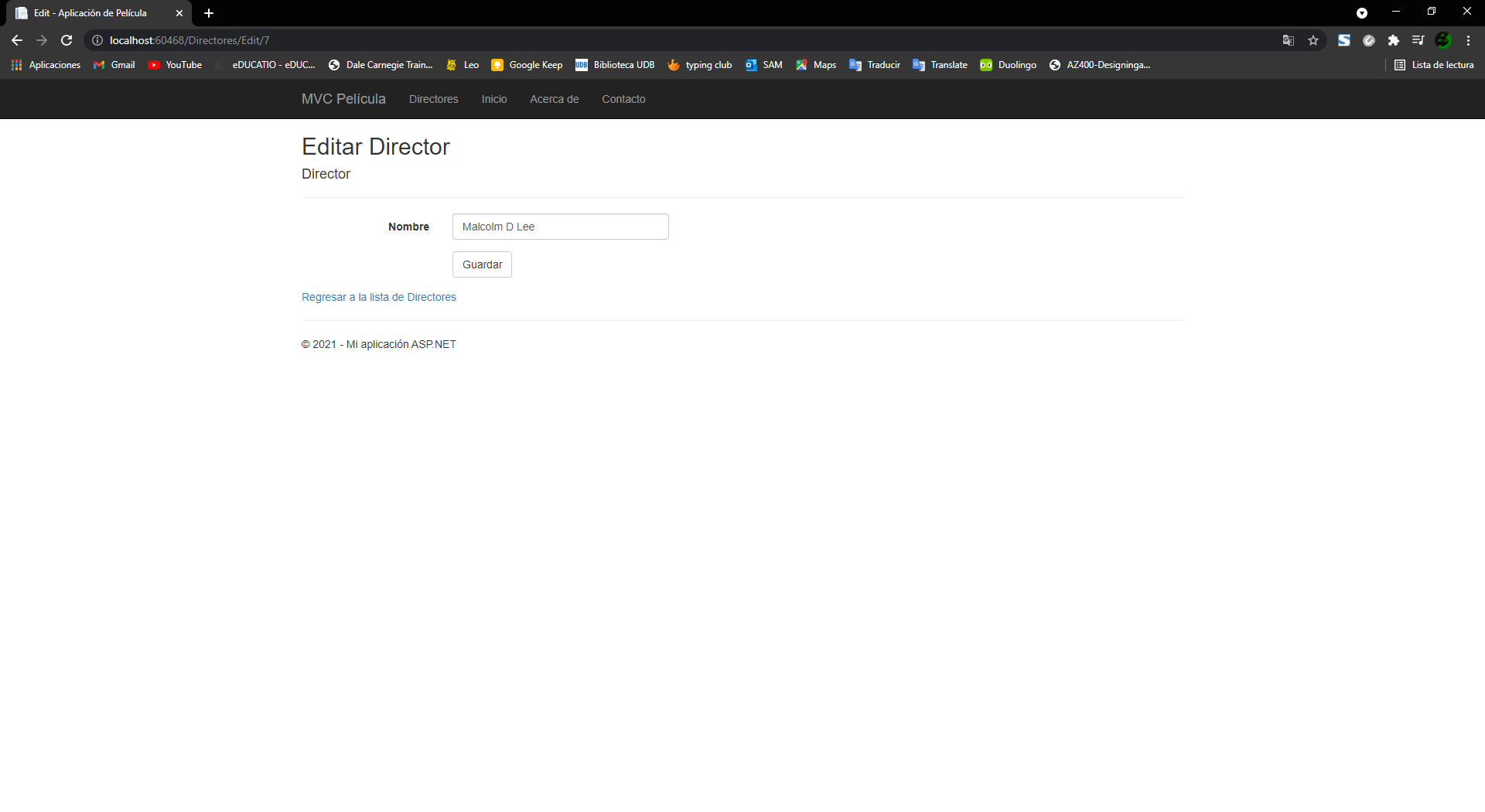
### Agregar



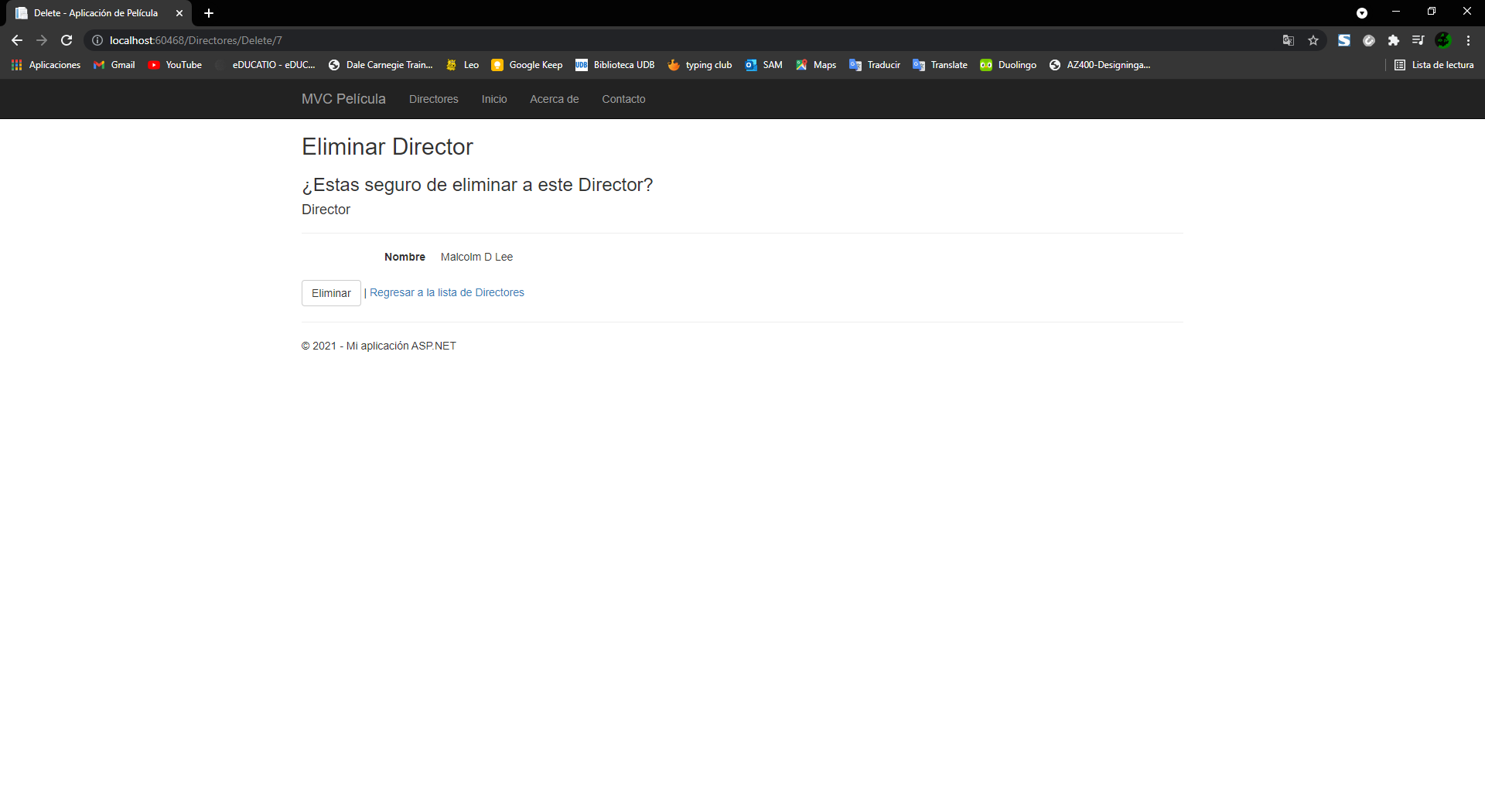
### Consultar



### Editar

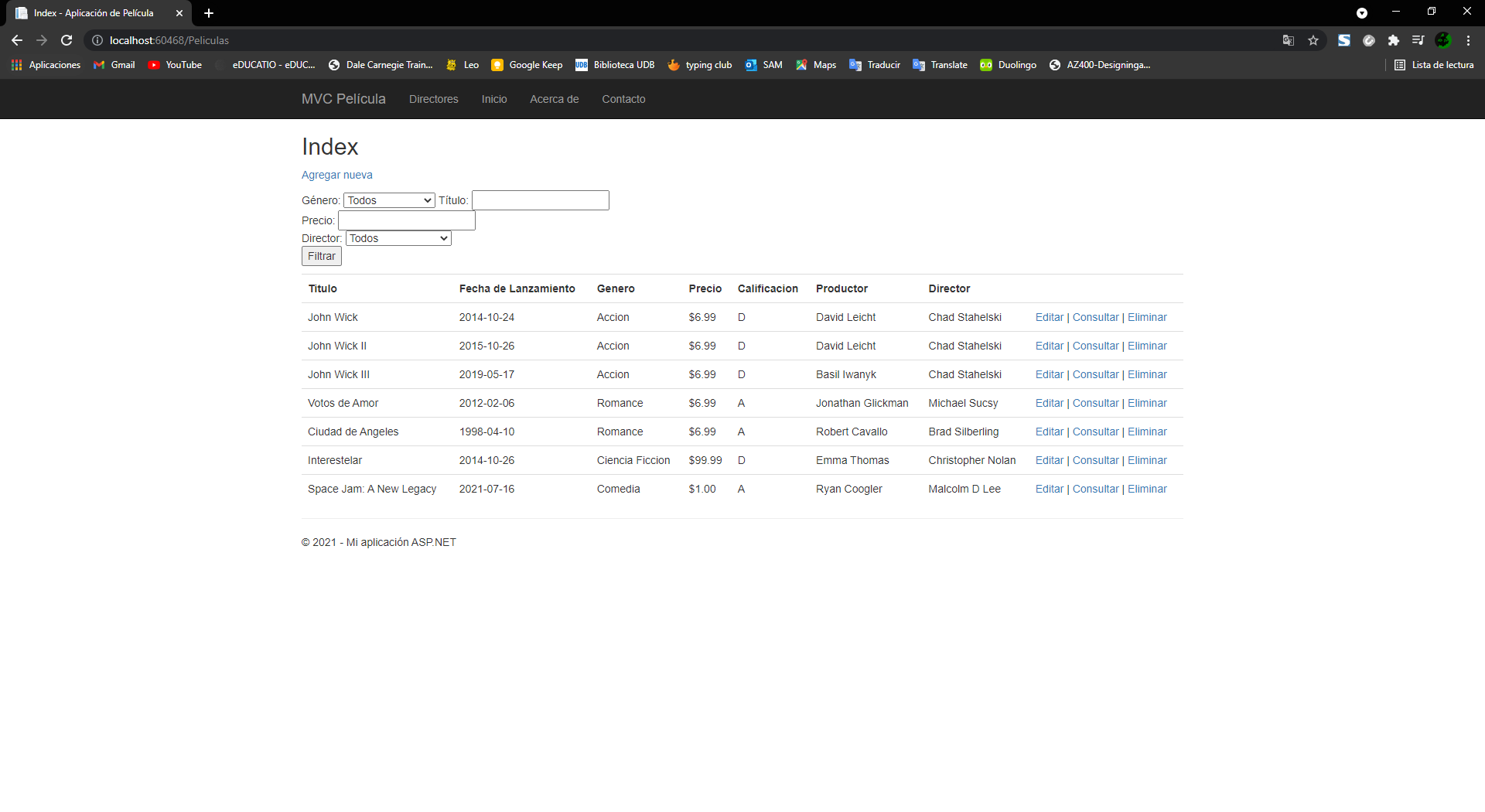


### Eliminar

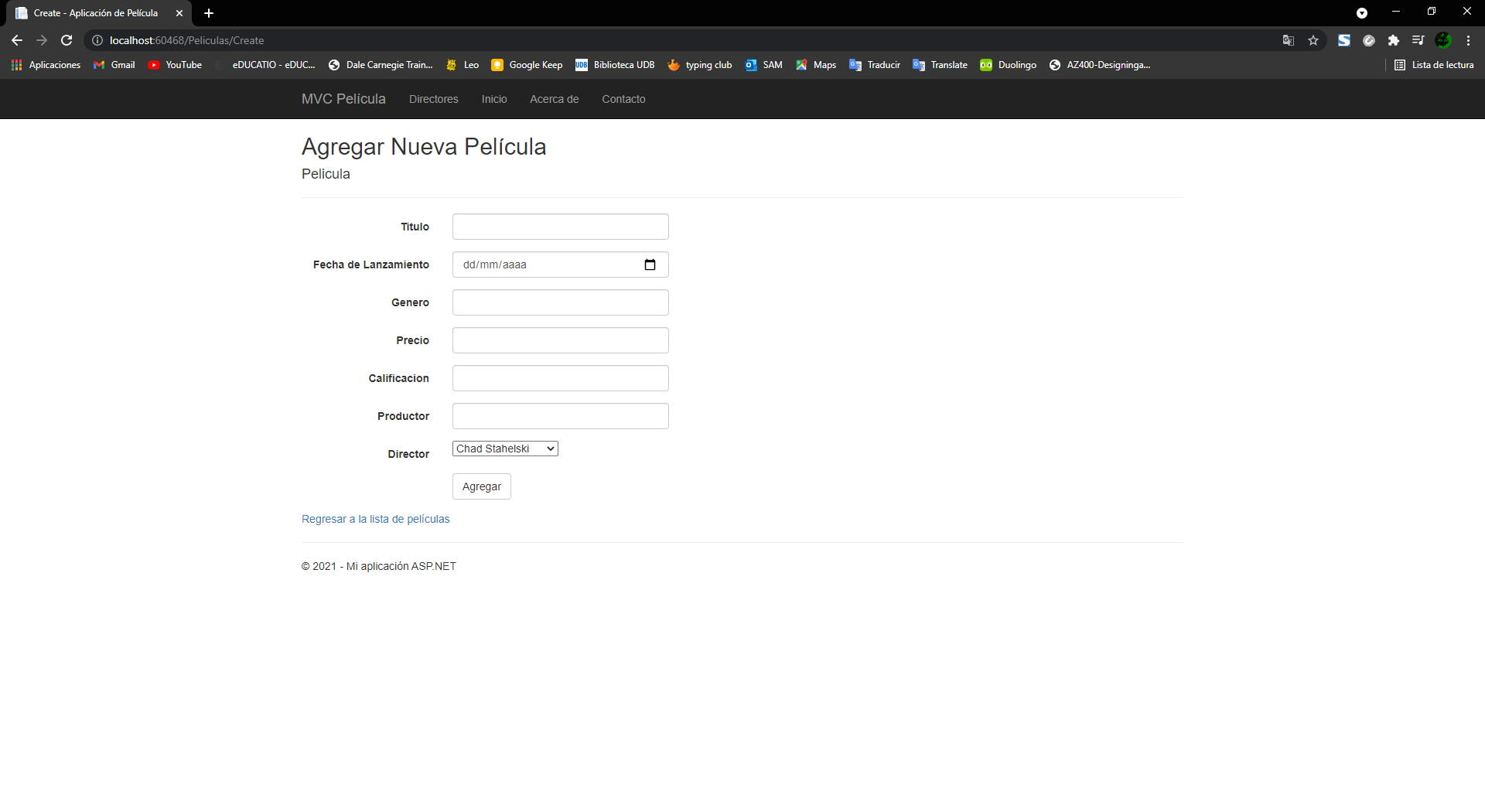


## CRUD películas

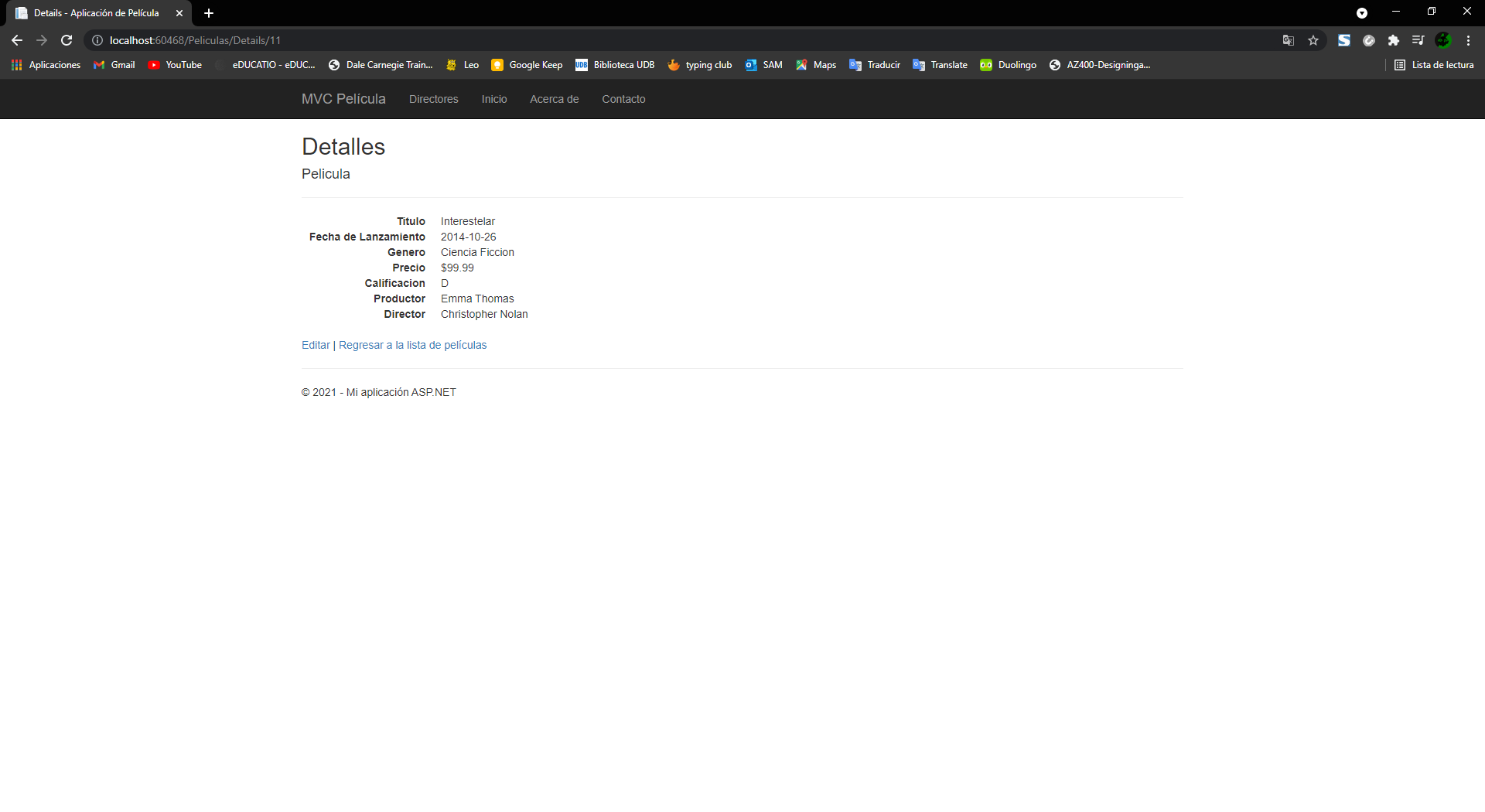
### Index



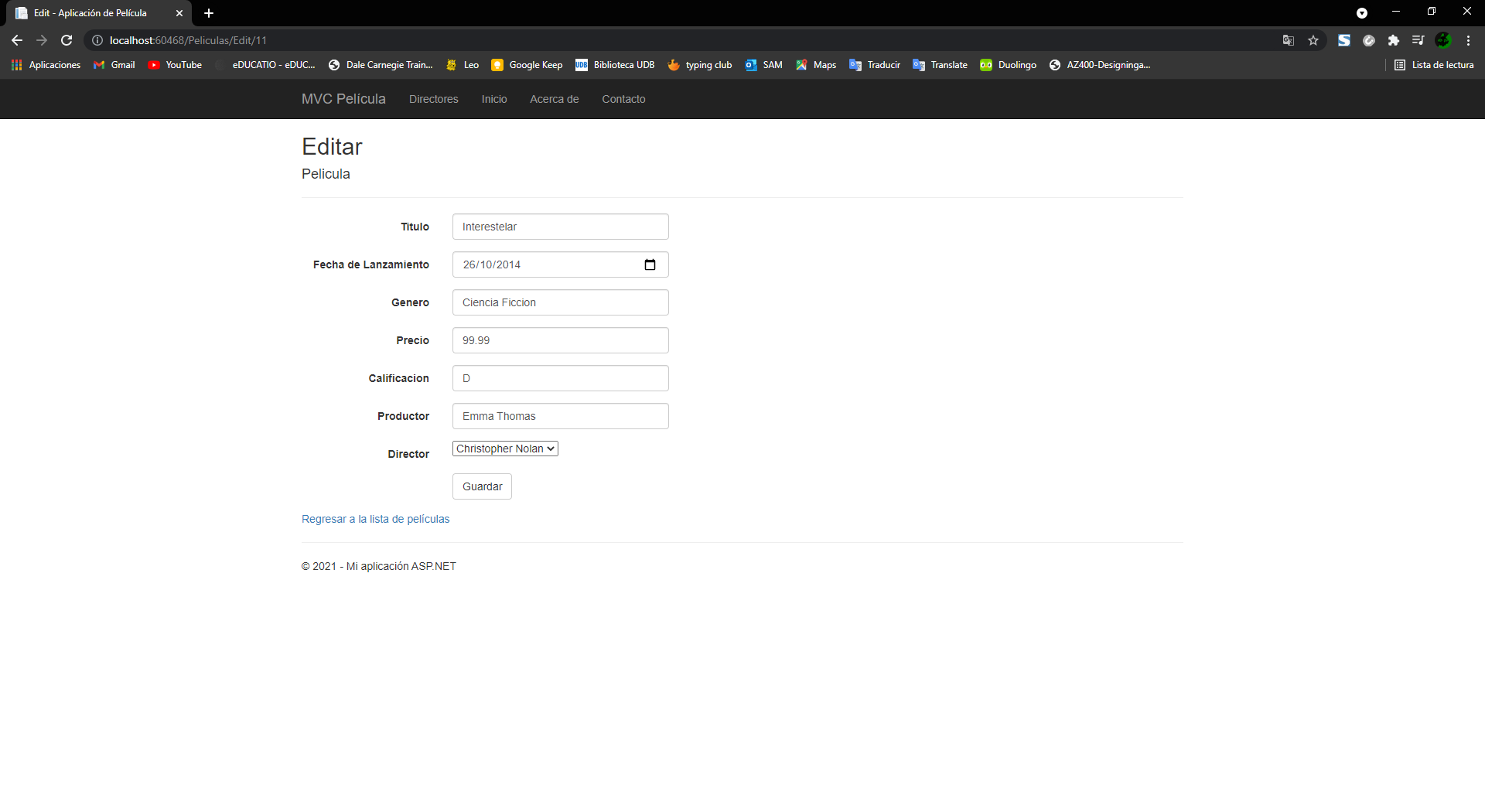
### Agregar



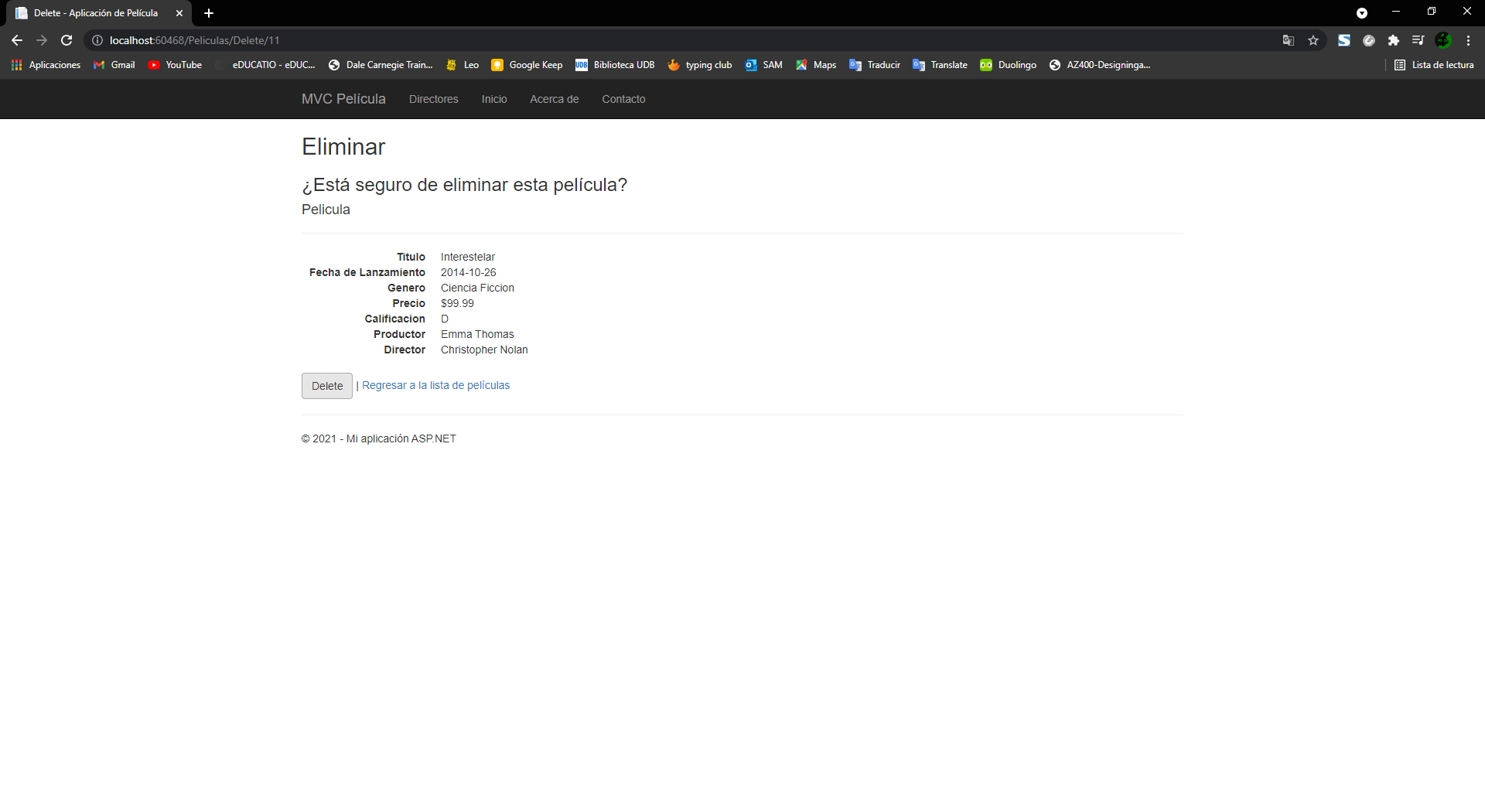
### Consultar



### Editar

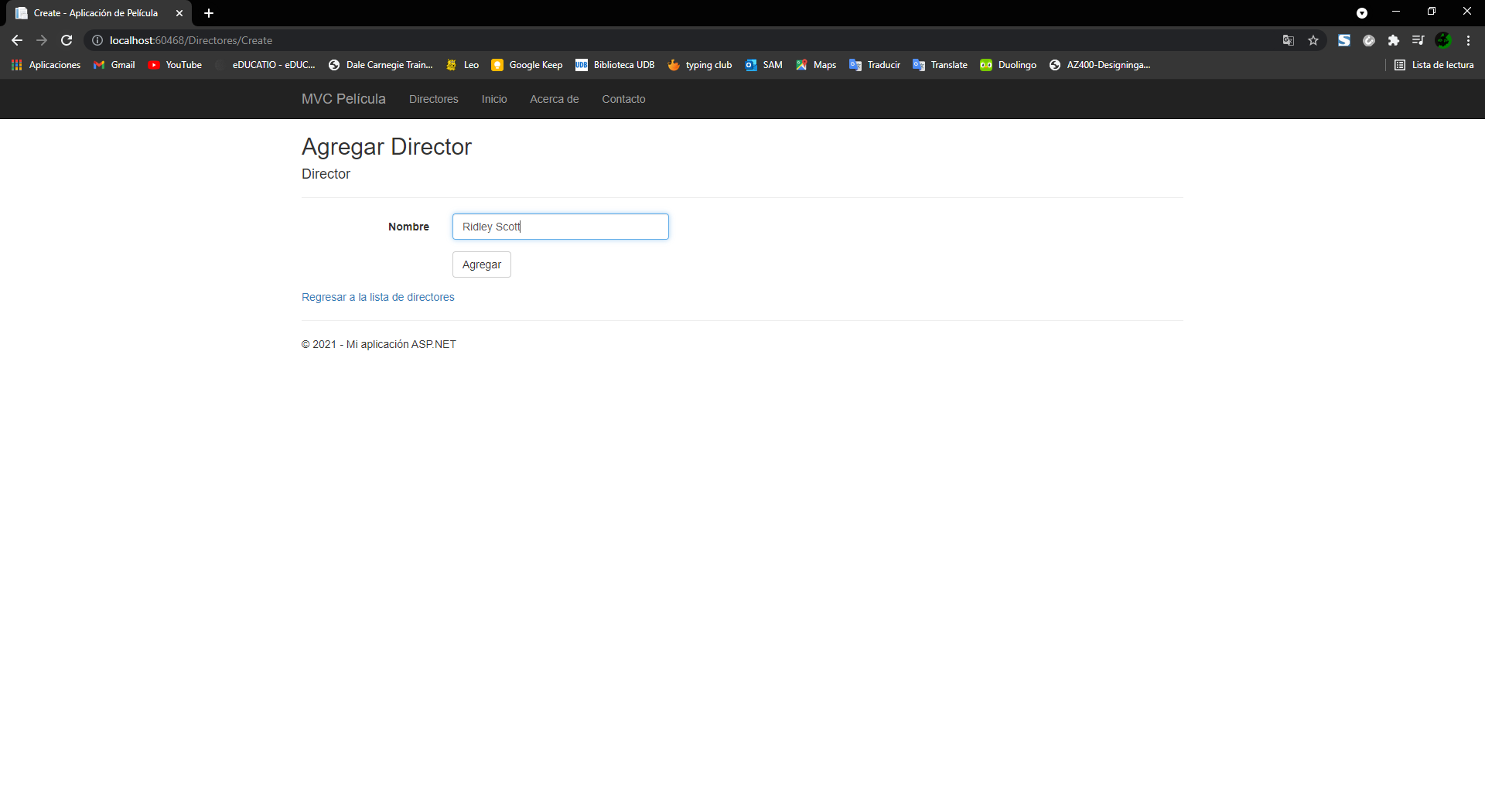


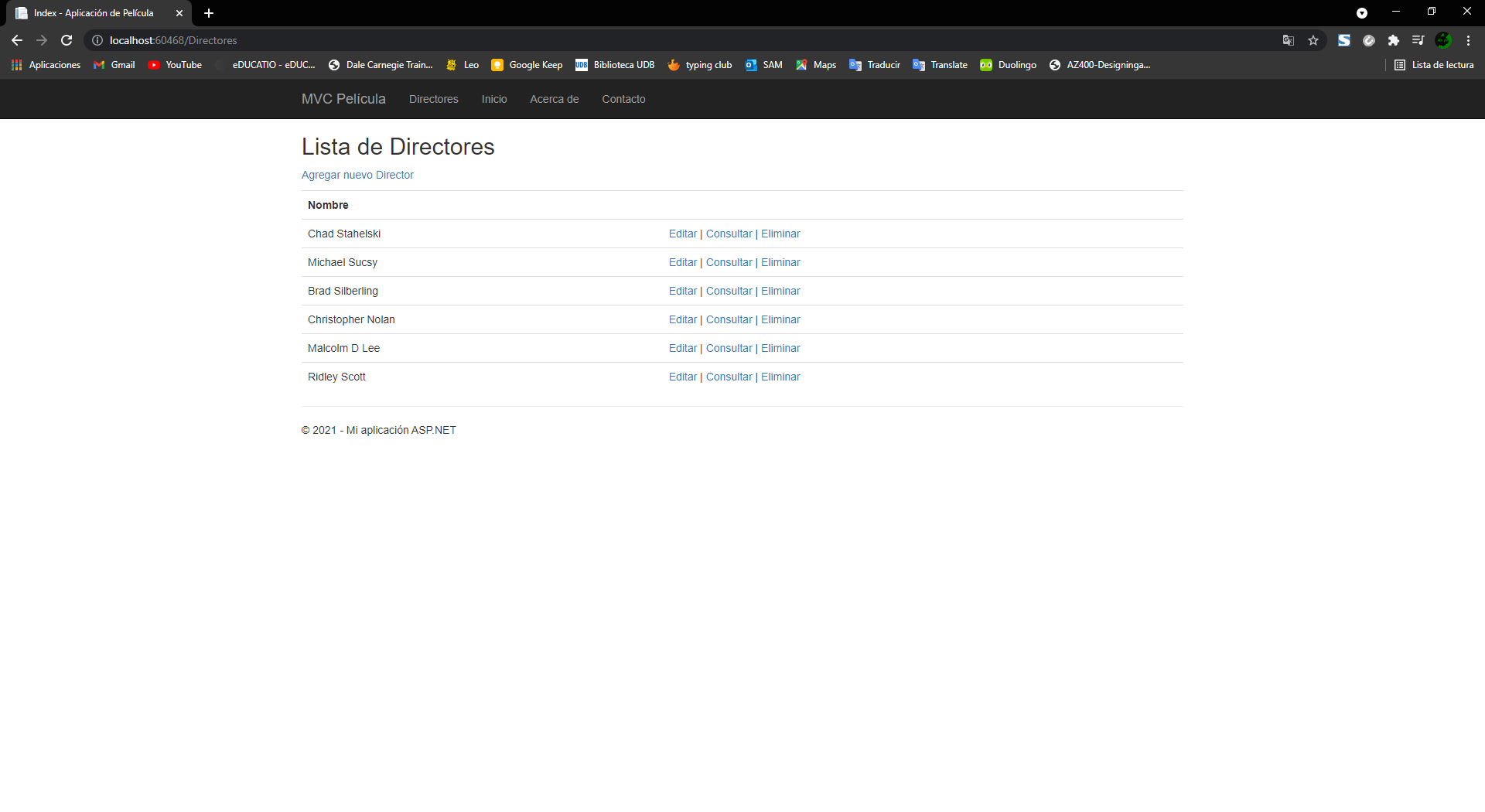
### Eliminar



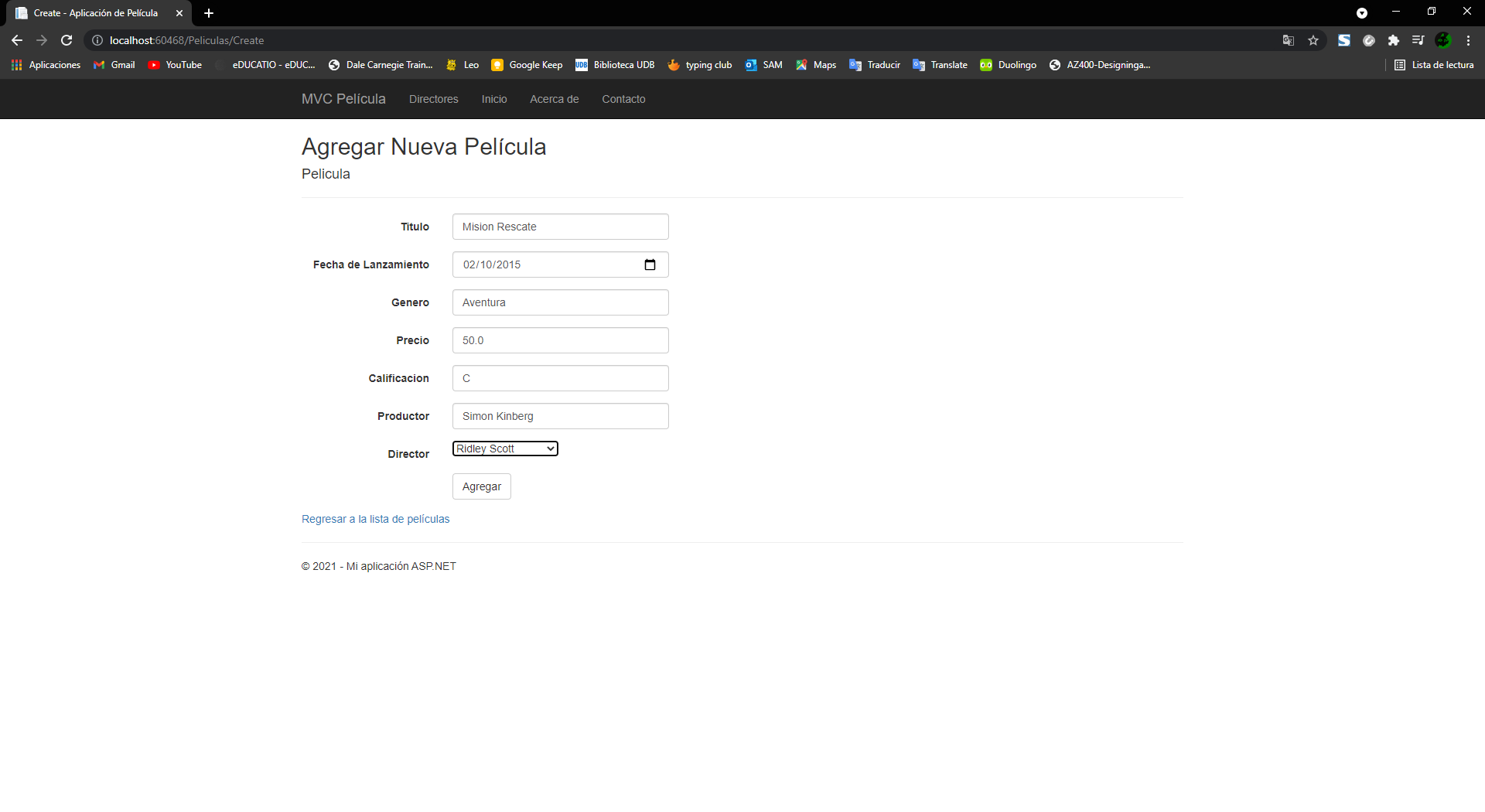
## Funcionamiento del sistema

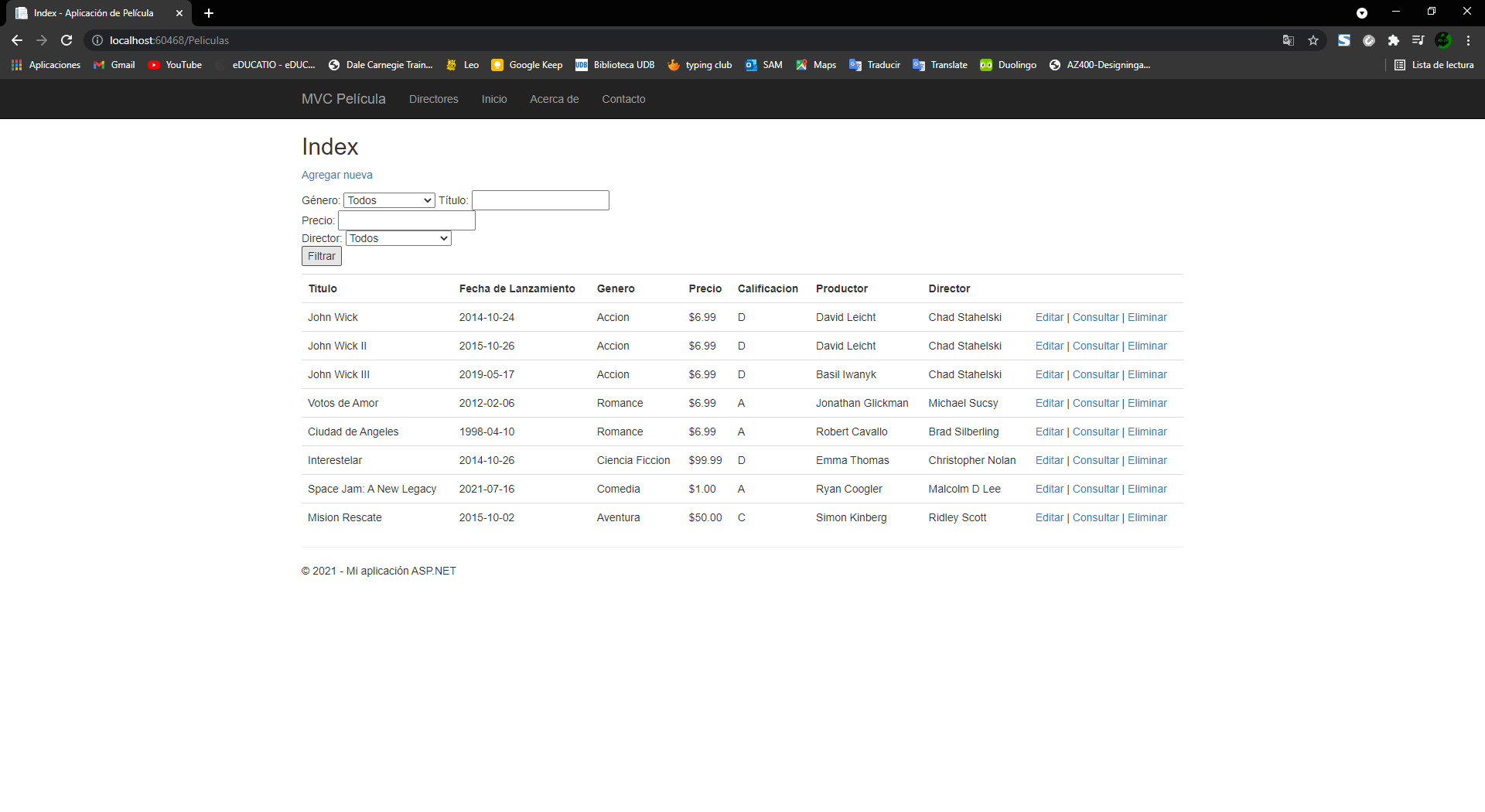
### Agregar Director





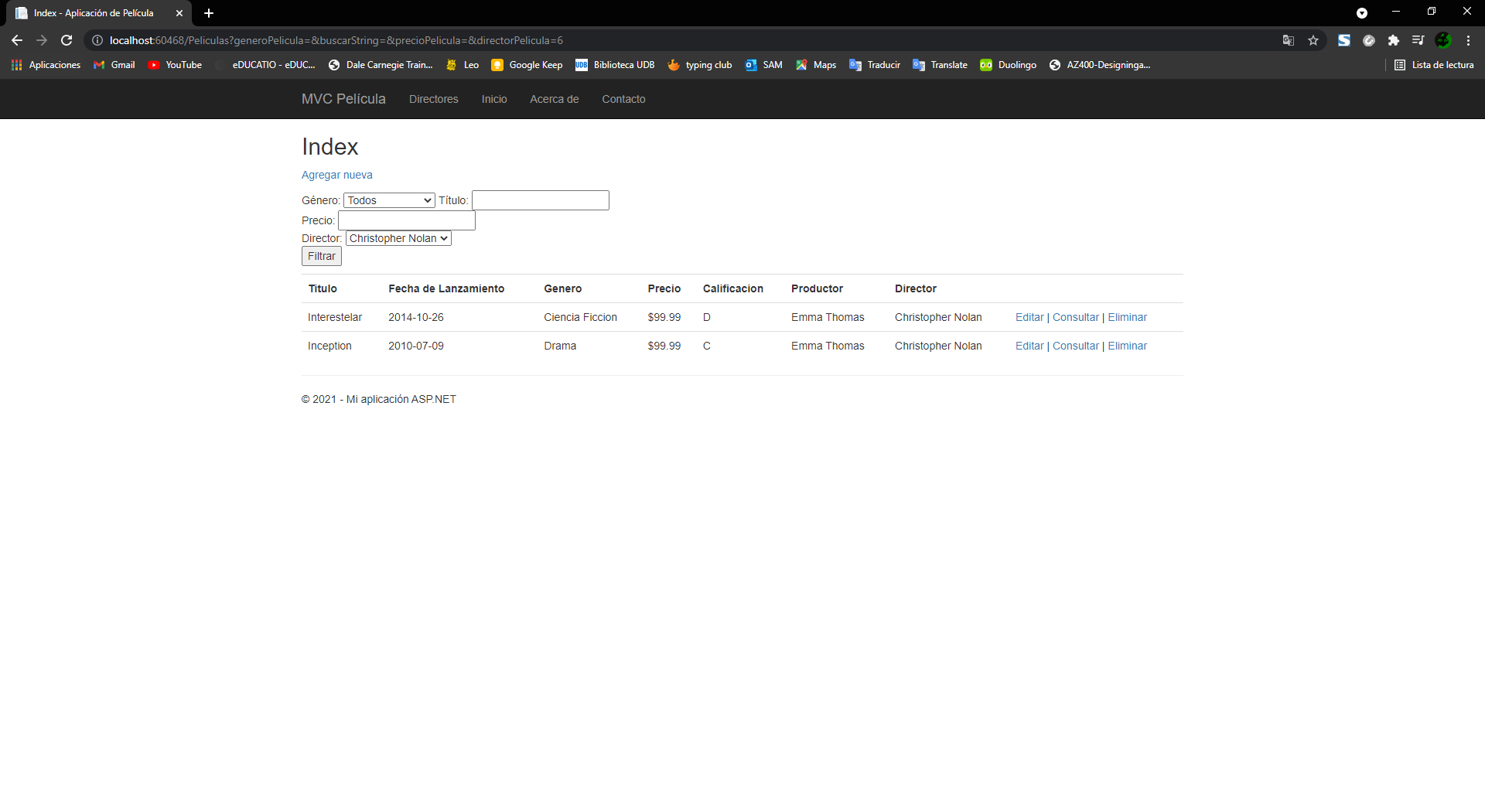
### Agregar película





## Funcionamiento de Filtros

### Listado de películas por Director



### Listado de películas por Director y por Género

