

Laboratorio 5¹

Conexión con una Base de Datos

Contenidos

1. Requisitos	1
2. Objetivos	1
3. Conexión con una base de datos	1
3.1. Conectarse a una base de datos en un servidor remoto	1
3.2. Conexión con una base de datos local	3
3.2.1. Connection strings	4
3.2.2. Creación de una tabla sobre la base de datos	6
4. Manejo de datos	7
4.1. Creación de un modelo	7
4.2. Manipulación de los datos	8
4.3. Despliegue y visualización de los datos	12
5. Entregables del laboratorio	15

¹ Basado en material previamente desarrollado en cursos de Gustavo López y Rebeca Obando en la Universidad de Costa Rica



1. Requisitos

1. Microsoft Visual studio 2019 o mayor instalado (cualquier versión eg. *Community*, *Enterprise*)
2. Cuenta en GitHub
3. SQL server instalado

2. Objetivos

1. Aprender los principios básicos que conllevan a la conexión de un proyecto con una base de datos, ya sea remota o local.
2. Interiorizar conceptos que se manejan a la hora de desplegar datos provenientes de una base de datos

3. Conexión con una base de datos

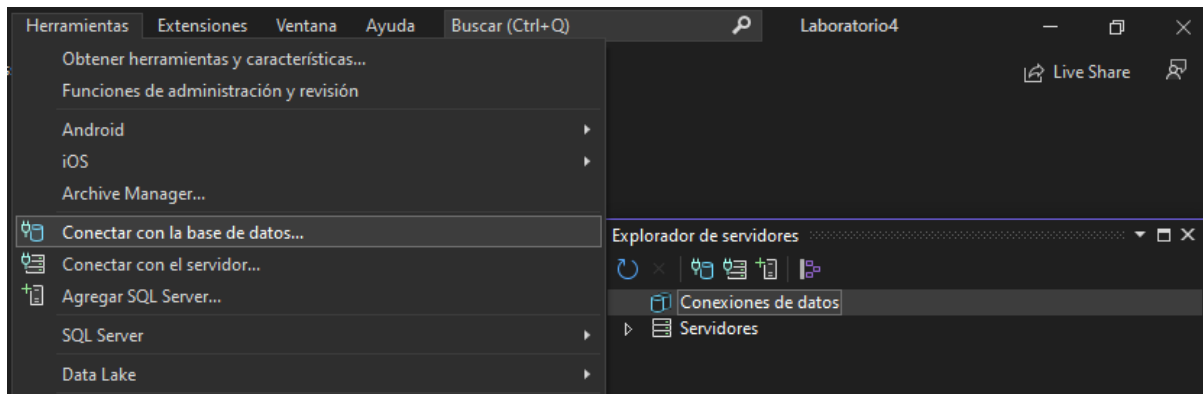
Al igual que en el laboratorio anterior, en este laboratorio debe crear un nuevo proyecto con el nombre **Laboratorio5** en el repositorio del curso dentro de un directorio llamado **laboratorio5**. Utilice el mismo tipod de proyecto del laboratorio3, **ASP.NET MVC Core**

Existen dos formas de conectar nuestro proyecto con una base de datos esto puede ser con una base de datos externa o remota, o con una base de datos local. En este laboratorio veremos ambas formas.

- 3.1. Conectarse a una base de datos en un servidor remoto
 - a. Establezca conexión con la VPN de la ECCI

Pensemos que nuestra base de datos remota sería similar a la que se nos asignó para el **PI**. En este caso debe tener conocimiento de la dirección IP, el nombre de usuario y la contraseña. Una vez teniendo estos datos siga los siguientes pasos.

- b. Ingrese a **Visual Studio**, al proyecto que generó anteriormente y busque la pestaña de **"Server explorer"**.



- c. En **Server explorer**, haga clic derecho sobre **Data connections** y presione **Add Connection**. Una vez hecho esto debería verse un cuadro como el siguiente, el cual debe llenar con la información solicitada anteriormente.

Agregar conexión

Especifique la información para conectarse al origen de datos seleccionado o haga clic en "Cambiar" para elegir otro origen o proveedor de datos.

Origen de datos:
Microsoft SQL Server (SqlClient) Cambiar...

Nombre del servidor:
Actualizar

Conexión con el servidor
Autenticación: Autenticación de SQL Server

Nombre de usuario:

Contraseña:

☐ Guardar mi contraseña

Establecer conexión con una base de datos

☒ Seleccionar o escribir el nombre de la base de datos:

☐ Adjuntar un archivo de base de datos:
 Examinar...

Nombre lógico:

Avanzadas...

Probar conexión Aceptar Cancelar

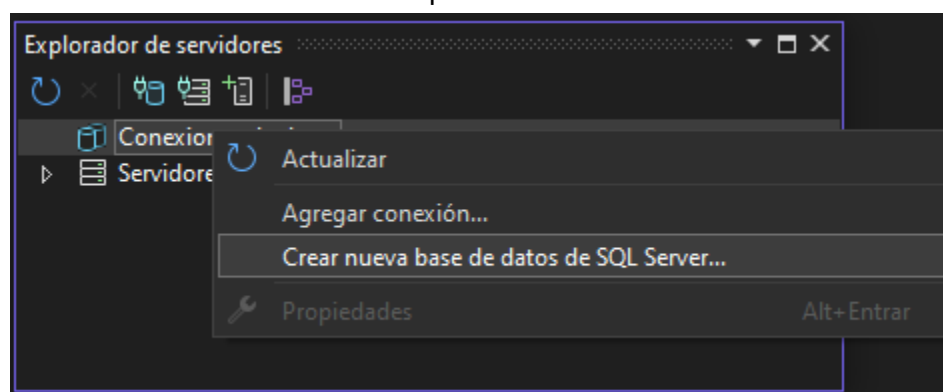
1. En Nombre del servidor, debe colocar la IP que se le proporcionó en el curso de bases de datos.
2. En Autenticación, coloque que desea autenticarse por medio de “**Autenticación de SQL server**”.
3. En usuario y contraseña, coloque las que se le proporcionaron para el servidor ingresado anteriormente.
4. En la opción de seleccionar o escribir el nombre de la base de datos se le van a desplegar aquellas donde usted tenga acceso. Seleccione la que le corresponde.
5. De clic en **Probar Conexión**. Si realizó todo de forma correcta, debería de salir un cuadro indicando que la prueba de conexión fue exitosa.

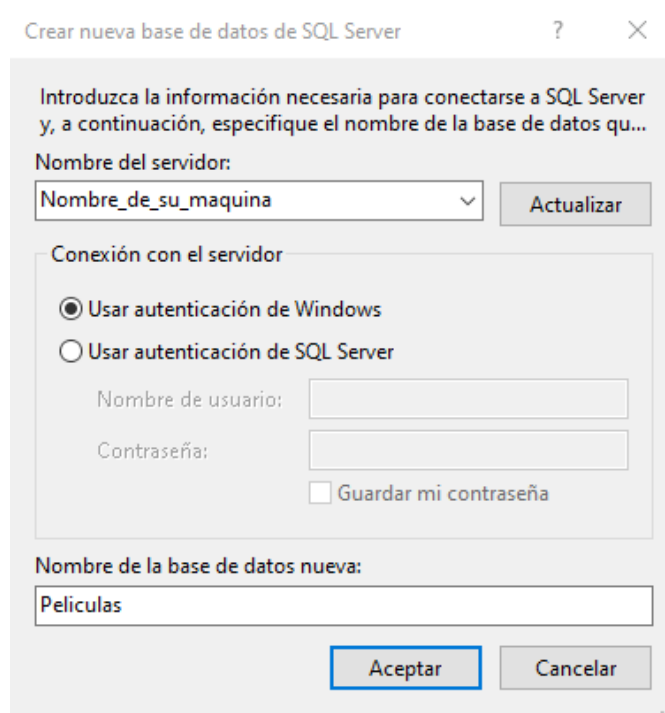
Ahora, presione el botón “**Aceptar**” para agregar la conexión. Una vez hecho esto, en el **Server Explorer** deberá aparecer su base de datos.

Si usted desea cerrar la conexión con la base de datos, solo debe presionar clic derecho sobre la base de datos y seleccionar **Cerrar conexión**. Si en vez de esto desea volver a abrirla solo es necesario dar doble clic en la base de datos.

3.2. Conexión con una base de datos local

La conexión con este tipo de bases de datos es similar a la vista en el punto anterior, sin embargo, para estas no se necesita una conexión con una VPN como fue la de la ECCI, cabe destacar que el encargado de la creación de la base de datos será usted. Para conectar la base de datos al proyecto debe ingresar al **Server Explorer** y dar clic derecho sobre **Data Connections** y seleccionar la opción **Create New SQL Server Database**. La siguiente imagen muestra el resultado de haber realizado los pasos antes mencionados.





En la casilla de “**Nombre del servidor**”, debe colocar el nombre de su computadora. Existen formas de averiguarlo a partir de línea de comandos (escribiendo el comando `hostname`) o a través de **SQL server**. En “**Conexión con el servidor**”, coloque que desea utilizar la autenticación de **Windows**, y finalmente en “**Nombre de la base de datos nueva**”, inserte el nombre que desea colocarle a su base de datos local. En este laboratorio crearemos una llamada **Películas**.

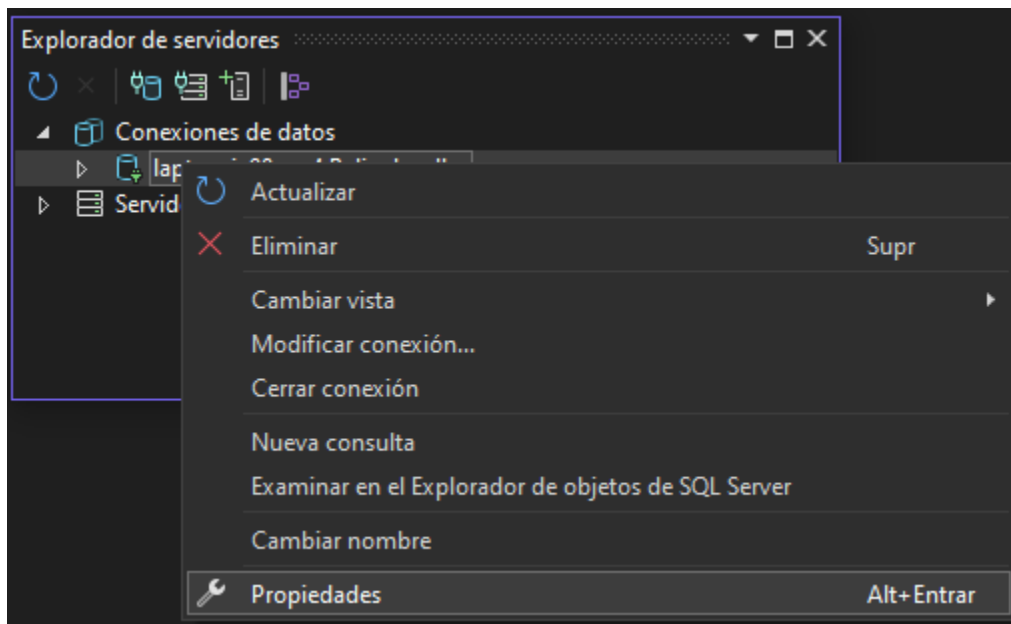
3.2.1. Connection strings

Debe saber que existe otra manera de conectar con una base de datos, esta se realiza por medio de un **connection string**. Este es una hilera de texto que especifica al programa como conectarse a la fuente de datos, comúnmente está compuesto de una serie de atributos que pueden variar pero su formato siempre es el mismo. Esta “**cadena**” se agrega y configura en el archivo **Web.config**.

Investigue las ventajas de realizar la conexión con una base de datos mediante el connection string.

Para obtener el “**connection string**” vaya al **Server Explorer** y haga clic derecho sobre la

base de datos “**Películas**” y seleccione la opción **Propiedades**. La siguiente imagen muestra el resultado.



En las Propiedades muestran información de la base de datos como se ve en la siguiente imagen, entre ellos el **Connection string** (Cadena de conexión) de la misma. Vamos a seleccionar el texto del **connection string** y lo vamos a copiar.

(Identidad)	
(Nombre)	Películas
Conexión	
Cadena de conexión	Data Source=LAPTOP-IV98VCG4;Initial Catalog=f
Estado	Abrir
Proveedor	Proveedor de datos .NET Framework para servid
Tipo	Microsoft SQL Server
Versión	15.00.2000

Para establecer la conexión entre nuestra aplicación web, y la base de datos que recién creamos, vamos a localizar un archivo llamado **appsettings.json**. En este archivo es donde vamos a decirle a nuestro programa que utilice el **connection string** de nuestra base de datos para extraer la información necesaria. La forma en que lo colocamos se observa en la siguiente imagen.

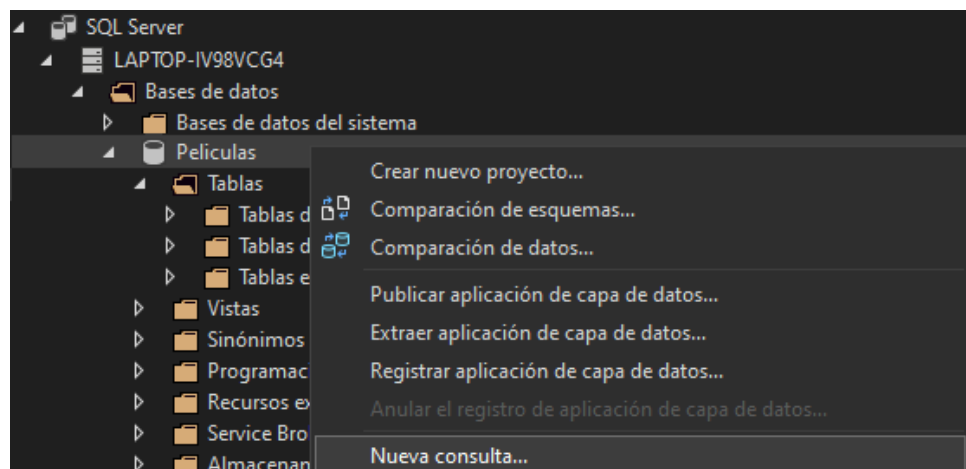


```
appsettings.json
Esquema: https://json.schemastore.org/appsettings.json
1 {
2   "Logging": {
3     "LogLevel": {
4       "Default": "Information",
5       "Microsoft": "Warning",
6       "Microsoft.Hosting.Lifetime": "Information"
7     }
8   },
9   "AllowedHosts": "*",
10
11  "ConnectionStrings": {
12    "ContextoDePeliculas": "Data Source=LAPTOP-IV98VCG4;Initial Catalog=Peliculas;Integrated Security=True;Pooling=False"
13  }
14 }
```

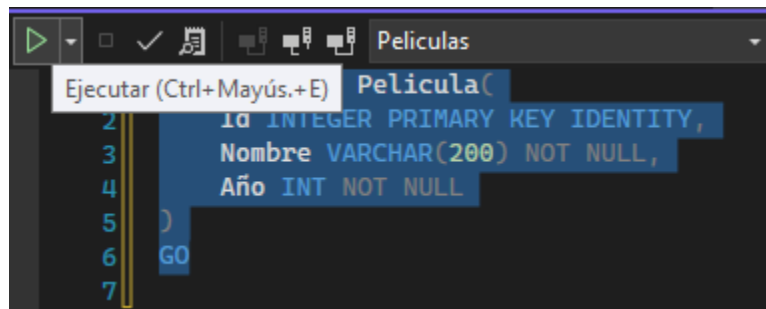
Con esto ya tenemos conectada nuestra base de datos a nuestro código. Es importante recordar que pueden existir múltiples conexiones de base de datos en este archivo.

3.2.2. Creación de una tabla sobre la base de datos

Para poder ingresar datos a la base, debemos generar una consulta y crear una tabla de una base de datos SQL, tal como se aprendió en el curso de bases de datos. Para esto ingrese al menú de **“Explorador de objetos de SQL server”**, seleccione la pestaña de SQL server, despliegue la que contiene el nombre de su máquina y busque la base de datos que creó. Sobre ella de clic derecho y presione la opción de **“Nueva consulta”**.

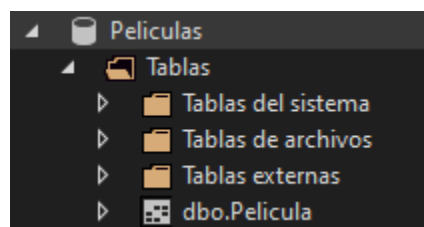


Lo anterior va a generar un archivo con extensión **.sql** donde podremos colocar el código para generar la tabla de **Película** con dos atributos. Presione ejecutar y verifique que la creación de la tabla fue correcta.



```
CREATE TABLE Pelicula(  
    Id INTEGER PRIMARY KEY IDENTITY,  
    Nombre VARCHAR(200) NOT NULL,  
    Año INT NOT NULL  
)  
GO
```

Si la generación fue exitosa, la tabla debería de visualizarse de la siguiente forma en el Explorador de objetos de SQL server.



4. Manejo de datos

El manejo de datos en una base de datos es una parte importante a conocer en la producción del Software, esto puede llevar desde la creación, eliminación, muestra o consulta hasta la edición de aquella información que tenemos en dicha base de datos. En esta sección de este laboratorio nos centraremos en el manejo de datos desde la perspectiva de creación y visualización.

4.1. Creación de un modelo

Para representar los datos es necesario la creación de un modelo (como fue aprendido en el laboratorio 3). Como sugerencia nombre el archivo **PeliculaModelo.cs** y agregue el siguiente código.

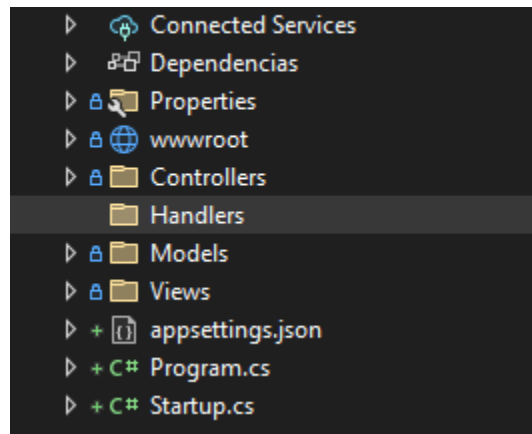


Con esto lograremos que en nuestro proyecto exista una base de datos en la cual tendremos una tabla Películas, es por este motivo que solo existe el modelo películas. Sin embargo, cuando existan varias tablas, no es necesario tener un modelo por tabla si no se tiene que unificar los datos de las diferentes tablas en un mismo modelo, de manera tal que se logre la representación del objeto que estamos trabajando en ese momento.

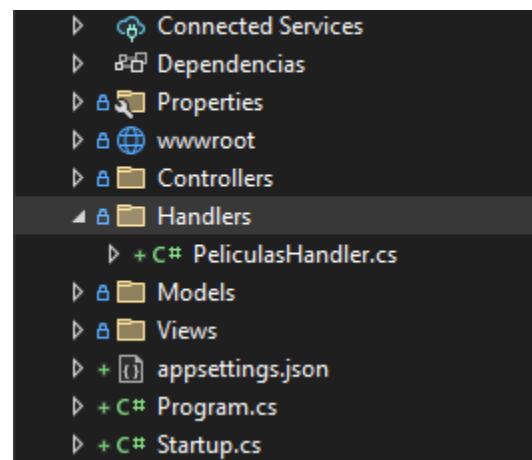
El objetivo del modelo es entonces responder a las necesidades según los datos que se desean representar.

4.2. Manipulación de los datos

Una vez que los datos fueron ingresados a la base de datos, se debe proceder a generar el “enlace” entre estos y el paso donde se despliegan en la página web. Para esto, comenzamos creando una carpeta en el proyecto llamada “**Handlers**”, tal y como se observa en la siguiente imagen.



Dentro de la carpeta recién creada, vamos a crear una nueva clase llamada **"PelículasHandler.cs"**, en la cual vamos a pegar el código que va a generar la conexión con nuestra base de datos, y a la vez generar la consulta para extraer los datos que queremos desplegar.



El siguiente código será el que deben de pegar en la clase recién creada. Analice las partes para que pueda comprender a fondo que está sucediendo.

```
using System;
using System.Collections.Generic;
using Laboratorio4.Models;
using System.Data;
using System.Data.SqlClient;
using Microsoft.AspNetCore.Builder;
```



```
using Microsoft.Extensions.Configuration;

namespace Laboratorio5.Handlers
{
    public class PeliculasHandler
    {
        private SqlConnection conexion;
        private string rutaConexion;
        public PeliculasHandler()
        {
            var builder = WebApplication.CreateBuilder();
            rutaConexion =
builder.Configuration.GetConnectionString("ContextoDePeliculas");
            conexion = new SqlConnection(rutaConexion);
        }

        private DataTable CrearTablaConsulta(string consulta)
        {
            SqlCommand comandoParaConsulta = new SqlCommand(consulta,
conexion);
            SqlDataAdapter adaptadorParaTabla = new
SqlDataAdapter(comandoParaConsulta);
            DataTable consultaFormatoTabla = new DataTable();
            conexion.Open();
            adaptadorParaTabla.Fill(consultaFormatoTabla);
            conexion.Close();
            return consultaFormatoTabla;
        }

        public List<PeliculaModel> ObtenerPeliculas()
        {
            List<PeliculaModel> peliculas = new List<PeliculaModel>();
            string consulta = "SELECT * FROM Pelicula";
            DataTable tablaResultado = CrearTablaConsulta(consulta);
            foreach (DataRow columna in tablaResultado.Rows)
            {
```



```
        peliculas.Add(  
            new PeliculaModel  
            {  
                ID = Convert.ToInt32(columna["Id"]),  
                Nombre = Convert.ToString(columna["Nombre"]),  
                Año = Convert.ToInt32(columna["Año"]),  
            });  
    }  
    return peliculas;  
}  
}
```

A continuación, se presenta una breve explicación del código que usted acaba de copiar:

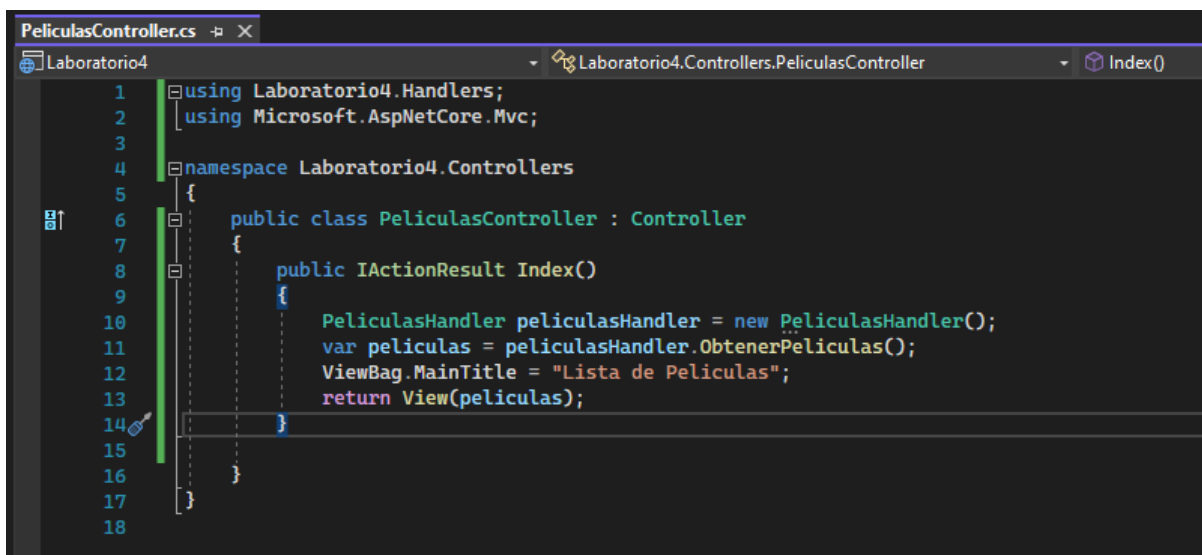
1. El **string de conexión** que habíamos copiado con anterioridad en el **appsettings.json** será uno de los atributos privados de la clase. Se lo debemos de indicar al código para que las librerías de SQL sepan donde deben conectarse para extraer la información.
2. El nombre del **connection string** que copiamos en pasos anteriores se debe pasar como parámetro al **GetConnectionString** con el fin de que este sea extraído y se pueda inicializar la conexión. Recuerde que puede usar distintas connection strings aquí, dependiendo del nombre que usted pusiera en su **appsettings.json**.
3. Recuerde generar esta conexión desde el constructor del **Handler** para que no repita código cada vez que desea conectarse a la base de datos.
4. El objeto **DataTable** es una representación de las tablas que por lo general se obtienen de realizar una consulta SQL, es decir, funciona como una representación en forma de matriz. El método creado "**CrearTablaConsulta**" esto es lo que realiza, este tiene como principal función el de llenar el objeto **DataTable** con la información extraída en la consulta.
5. En el caso del método para obtener las películas, las consultas se realizan a través de un **String**, mediante el cual también se pueden realizar consultas complejas, sin embargo, para este caso solo se ejemplifica con un caso sencillo. Se da como recomendación el tener cuidado siempre de dejar espacios entre cada sentencia que se escriba.

4.3. Despliegue y visualización de los datos

Ya tenemos la conexión con la base de datos, sin embargo es necesario llenar la misma con datos, y establecer una vista para que los mismos sean desplegados en la página web. El primer paso para esto es generar una consulta en la base de datos (como se aprendió en pasos anteriores) y llenarla con datos. Puede utilizar el siguiente código.

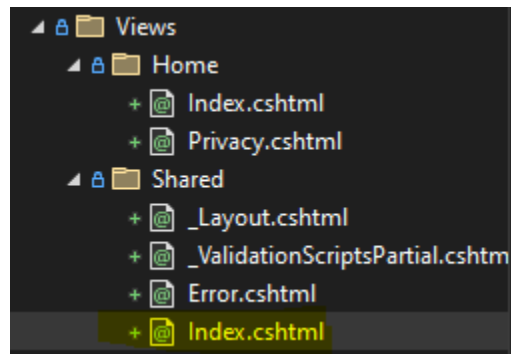
```
INSERT INTO [dbo].[Pelicula] ([Nombre],[Año])  
VALUES('Guardians of the Galaxy Vol 3', '2023'),  
('Spider-Man: No Way Home', '2022')  
GO
```

Luego, debemos generar un controlador tal y como se ha aprendido en laboratorios anteriores. Este controlador va a llamar a la función del **Handler** que programamos anteriormente, el cual extraía los datos de la base de datos. Un ejemplo de este código se muestra a continuación a modo de ilustración.



```
PeliculasController.cs  
Laboratorio4  
Laboratorio4.Controllers.PeliculasController  
Index()  
1 using Laboratorio4.Handlers;  
2 using Microsoft.AspNetCore.Mvc;  
3  
4 namespace Laboratorio4.Controllers  
5 {  
6     public class PeliculasController : Controller  
7     {  
8         public IActionResult Index()  
9         {  
10             PeliculasHandler peliculasHandler = new PeliculasHandler();  
11             var peliculas = peliculasHandler.ObtenerPeliculas();  
12             ViewBag.MainTitle = "Lista de Peliculas";  
13             return View(peliculas);  
14         }  
15     }  
16 }  
17  
18
```

Además de esto, debemos generar una **vista con html** como se aprendió en laboratorios anteriores, junto con una sección en el archivo **Layout** para que muestre una opción en el menú. Ejemplos de esto se pueden observar en las siguientes dos imágenes.



```
Index.cshtml
1  @model List<Laboratorio4.Models.PeliculaModel>;
2
3  @{
4      ViewData["Title"] = "Películas";
5  }
6
7  <h1>@ViewBag.MainTitle</h1>
8
9  <div>
10     <table class="table">
11         <thead>
12             <tr>
13                 <th>Título de la película</th>
14                 <th>Año de salida</th>
15             </tr>
16         </thead>
17         <tbody>
18             @foreach (var item in Model)
19             {
20                 <tr>
21                     <td>@item.Nombre</td>
22                     <td>@item.Año</td>
23                 </tr>
24             }
25         </tbody>
26     </table>
27
28 </div>
```



```
<div class="navbar-collapse collapse d-sm-inline-flex flex-sm-row-reverse">
  <ul class="navbar-nav flex-grow-1">
    <li class="nav-item">
      <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>
    </li>
    <li class="nav-item">
      <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>
    </li>
    <li class="nav-item">
      <a class="nav-link text-dark" asp-area="" asp-controller="Películas" asp-action="Index">Películas</a>
    </li>
  </ul>
</div>
```

Si todo lo anterior se llevó a cabo de manera correcta, ya debería de poder correr su solución y verificar que los datos se desplieguen de manera correcta. En la siguiente imagen se observa como se desplegaron los datos que estaban contenidos en la tabla películas.

Laboratorio4 Home Privacy Películas

Título de la película	Año de salida
Guardians of the Galaxy Vol 3	2023
Spider-Man: No Way Home	2022

Con esto, se concluye la última sección del laboratorio sobre conexión a las bases de datos.

5. Entregables del laboratorio

0.25% Todos los entregables de este laboratorio deben estar en su repositorio de GitHub en un directorio con nombre **laboratorio5**.

0.25% Cree un documento ordenado con su nombre y número de carné. Utilice el nombre **ReporteLab5_CARNE.pdf** (su número de carné). No entregue archivos en Word o en ningún otro formato editable. Utilice PDF.

80% El reporte debe tener una serie de **screenshots** que evidencien la realización de los pasos principales del laboratorio. Los siguientes **screenshots** son requeridos (y tienen el mismo valor). Puede incluir más si agregan valor al reporte.

1. Conexión con la base de datos remota (sección 1.1)
2. Conexión con la base de datos local (sección 1.2)
3. Cadena de conexión donde se vea nombre de base de datos local (sección 1.2.1)
4. Captura de la tabla Película sobre la base de datos local (sección 1.2.2)
5. Código del modelo generado (sección 2.1)
6. Captura de una porción del Handler (sección 2.2)
7. Captura de la tabla Película con los datos ingresados (sección 2.3)
8. Despliegue de la información en la página web (sección 2.3)

En los screenshots debe aparecer su nombre de usuario de manera clara e.g. nombre y apellidos, correo electrónico, usuario de windows, o usuario de GitHub que refleje su nombre claramente y no un pseudónimo o nombre críptico.

0.5% Agregue al reporte el enlace a su repositorio de GitHub.

9% 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.

10% Responder a la pregunta que se realiza en la sección 3.2.1 en relación con los connection strings, con una extensión máxima de 300 palabras.