



Ingeniería de software

Laboratorio 5: Conexión a base de datos

Resumen

El objetivo de este laboratorio es poder conectar a la base de datos y desplegar los datos en un proyecto MVC.

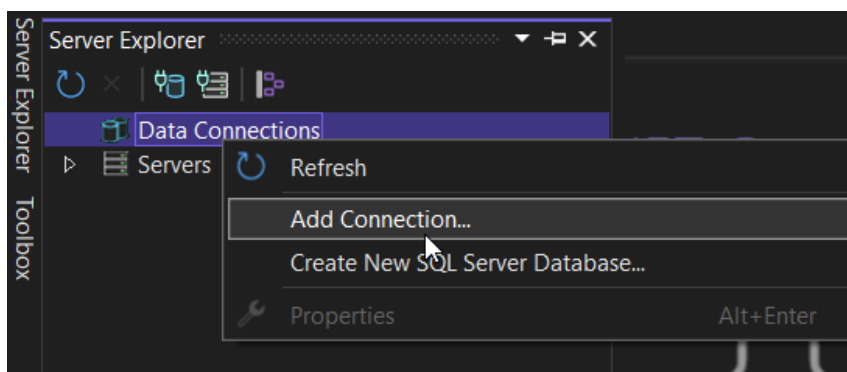
Primera parte - Conectando a una base de datos

Primero, debe crear un nuevo proyecto del tipo ASP.NET MVC Core (mismo tipo de proyecto utilizado en el lab 3) y nombre el proyecto como Laboratorio5, vincule este proyecto a nuevo repositorio en github.

Existen diferentes maneras de conectarse a una base de datos. Puede ser conectado a una base de datos local, es decir, que está en su propia computadora o bien puede conectarse a una base de datos que está alojada en un servidor remoto, como la base de datos que se le asignó al equipo en el curso de PI, a la cual ingresa mediante una dirección IP con una conexión vía VPN a la ECCI. En ambos casos, la manera de establecer la conexión es similar.

Conectando a una base de datos en un servidor remoto

1. Conéctese a la VPN de la ECCI.
2. En visual studio abra una el Server explorer. De click derecho sobre Data connections -> Add new connection





3. Seleccione la opción de data source: **Microsoft SQL Server**
4. Llene el formulario utilizando la siguiente información:
 - a. Server name: utilice alguna de las direcciones IP que se le proporcionaron en el curso de bases de datos
 - b. Authentication: seleccione la opción que se muestra en la siguiente figura.
 - c. Ingrese sus credenciales: su usuario y su respectiva contraseña
 - d. Select or enter database name: cuando presione el menú desplegable, aquí se mostrarán todas las bases de datos a las cuales tiene acceso (lectura y/o escritura) dentro del servidor, seleccione la base que le fue asignada
 - e. Finalmente de click en Test Connection

Add Connection ? X

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft SQL Server (SqlClient) Change...

Server name: Refresh

Log on to the server

Authentication: SQL Server Authentication

User name: my user name

Password: my password

☐ Save my password

Connect to a database

☒ Select or enter a database name:

☐ Attach a database file: Browse...

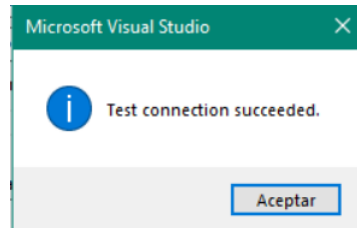
Logical name:

Advanced...

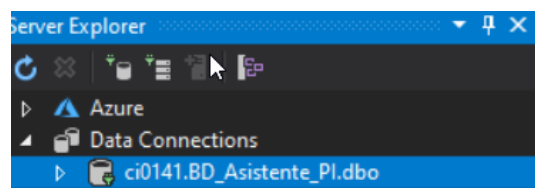
Test Connection OK Cancel



Debe obtener un mensaje de “Test Connection succeeded”



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



Para cerrar su conexión, presione click derecho sobre su base de datos y seleccione la opción Close Connection, para volver a abrir la conexión basta con dar doble click sobre la base de datos. Hasta este momento lo único que se ha hecho es añadir la conexión de la base de datos que está en el servidor remoto, pero para empezar a hacer consultas y demás es necesario realizar otros pasos que se presentan más adelante.

Conectando a una base de datos local

Para este caso no ocupa estar conectado a la VPN de la ecci sin embargo el encargado de crear la base de datos será usted. Para conectar la base de datos al proyecto solo debe ingresar al Server Explorer y dar click derecho sobre Data Connections y seleccione la opción Create New SQL Server Database. Se va abrir una ventana como la que se muestra en la figura a continuación.



En el server name debe poner el nombre de su máquina (es el mismo nombre que va a aparecer en SQL server management studio cuando instalo SQL server). En el nombre de new database name ponga “Países”

Connection strings

Otra manera de agregar una conexión a la base de datos es por medio de un **connection string**. El Connection String se agrega y configura en el archivo Web.config el cual se puede acceder desde el Solution Explorer. El Connection String en sí es la manera de especificar al programa la fuente de los datos, con lo cual se puede establecer la conexión.

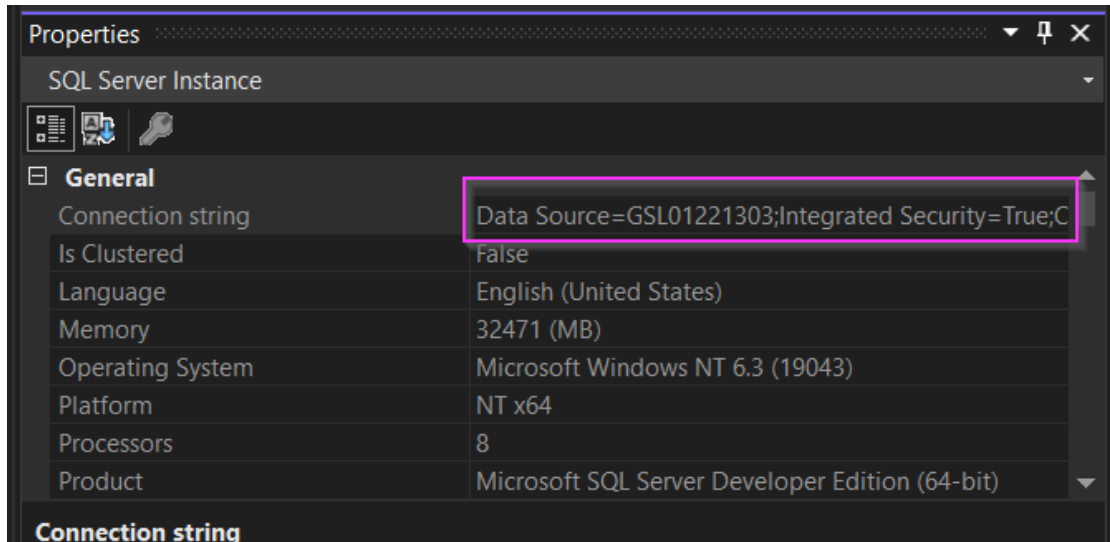
Usar el Connection String de la manera que se va mostrar a continuación, tiene la ventaja que cuando usted desea realizar consultas a una base de datos, no necesita repetir el proceso de ingresar esta hilera a cada uno de los comandos de SQL que se detallarán más adelante. Además, el Connection String no es único, usted puede tener varios en la configuración de su proyecto y nombrarlos de manera diferente, de modo que usted puede tener conexión con su base de datos del servidor remoto y local al mismo tiempo, incluso con otras bases de datos un mismo servidor o de distintos servidores.

¿Cómo obtener el connection string?

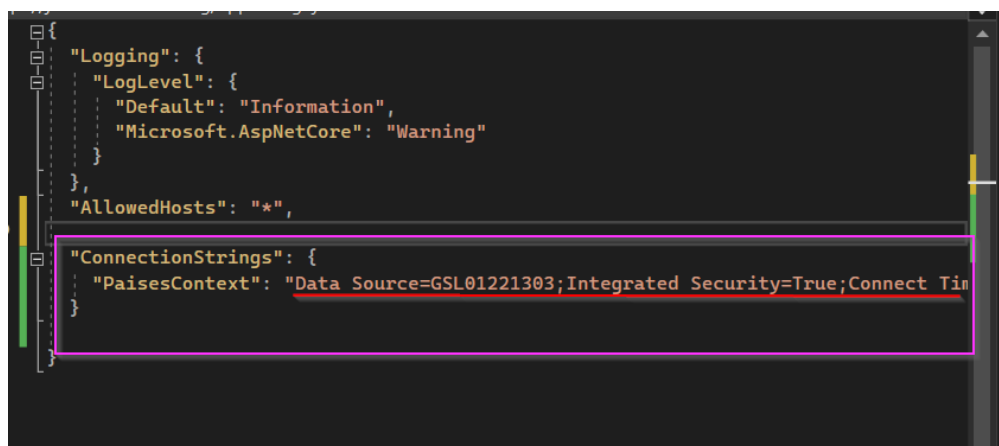
Desde el Server Explorer dé click derecho sobre la base de datos Países recién creada y seleccione la opción Properties. La siguiente figura muestra el resultado de seleccionar esta



opción, usted observará una fila que justamente se llama Connection String, la cual en la columna de al lado tiene una hilera, copie la hilera completa



Ingresa al Solution Explorer y ubique el archivo llamado appsettings.json, abra el archivo y al final del código, agregue el connection string de la siguiente manera:

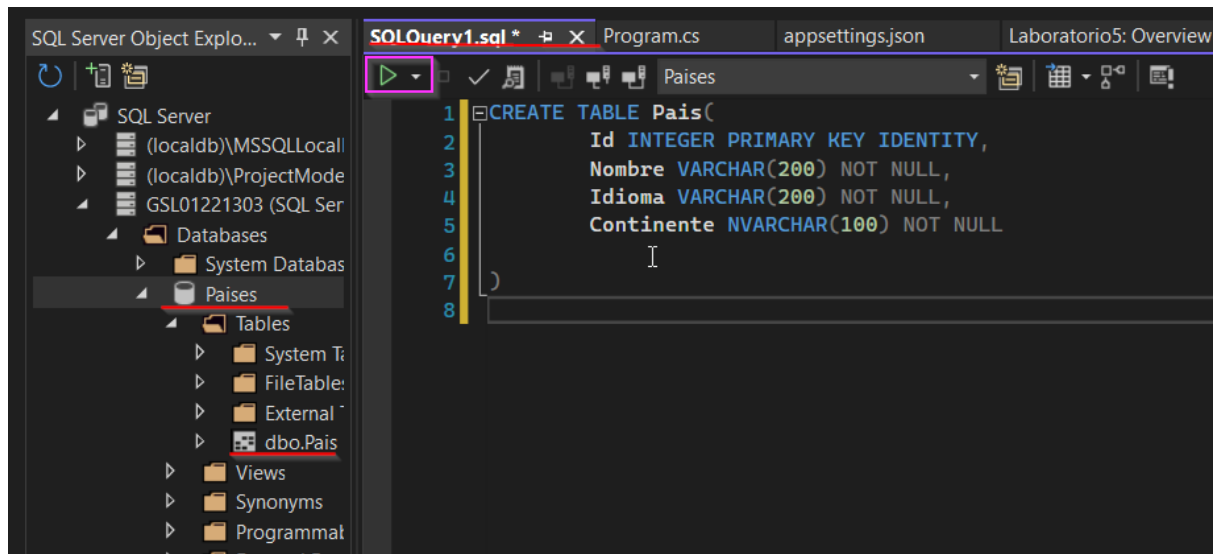


Creando una tabla en la base de datos

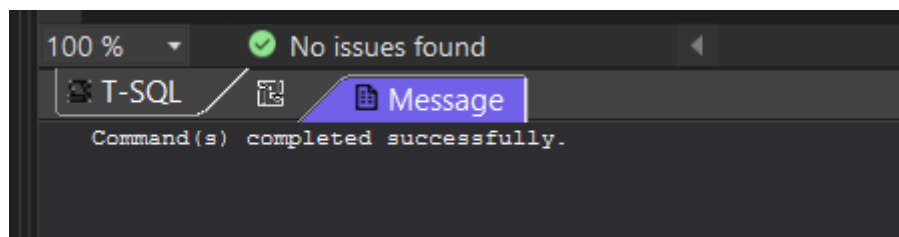
En este apartado no se dará mayor detalle, debido a que únicamente se creará una tabla con algunos atributos de la manera que usted ya aprendió en el curso de bases de datos. Dé click derecho sobre su base de datos Países y seleccione la opción New query, copie el siguiente código:



```
CREATE TABLE Pais(  
    Id INTEGER PRIMARY KEY IDENTITY,  
    Nombre VARCHAR(200) NOT NULL,  
    Idioma VARCHAR(200) NOT NULL,  
    Continente NVARCHAR(100) NOT NULL  
)  
GO
```



Presione el botón verde que está en la parte superior izquierda de la pantalla (figura anterior) y posteriormente recibirá un mensaje que refleja éxito en la creación de la tabla.



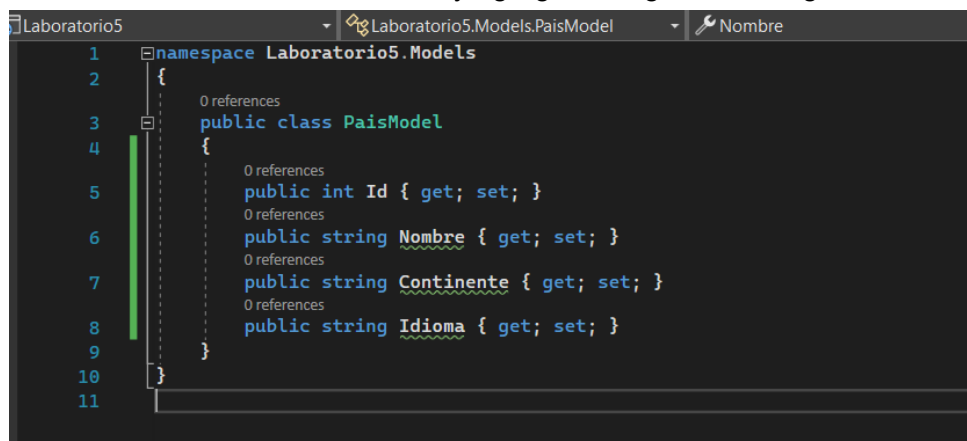


Segunda parte - Manipulando los datos

La manipulación de los datos almacenados en la base de datos se puede interpretar de distintas maneras como, por ejemplo: insertar nuevos datos, modificar los datos existentes, desplegar algunos datos o eliminarlos. En esta parte del laboratorio se va a centrar en la visualización y creación de los datos

Cree un modelo

Para lograr lo que se pretende, es necesario tener una manera de representar los datos en el sistema y para eso se utilizará un modelo, siguiendo la arquitectura en la cual se basa el tipo de proyecto que se está utilizando (MVC). Como usted aprendió en el laboratorio 3, cree un nuevo modelo llamado PaisModel, y agregue el siguiente código al modelo:

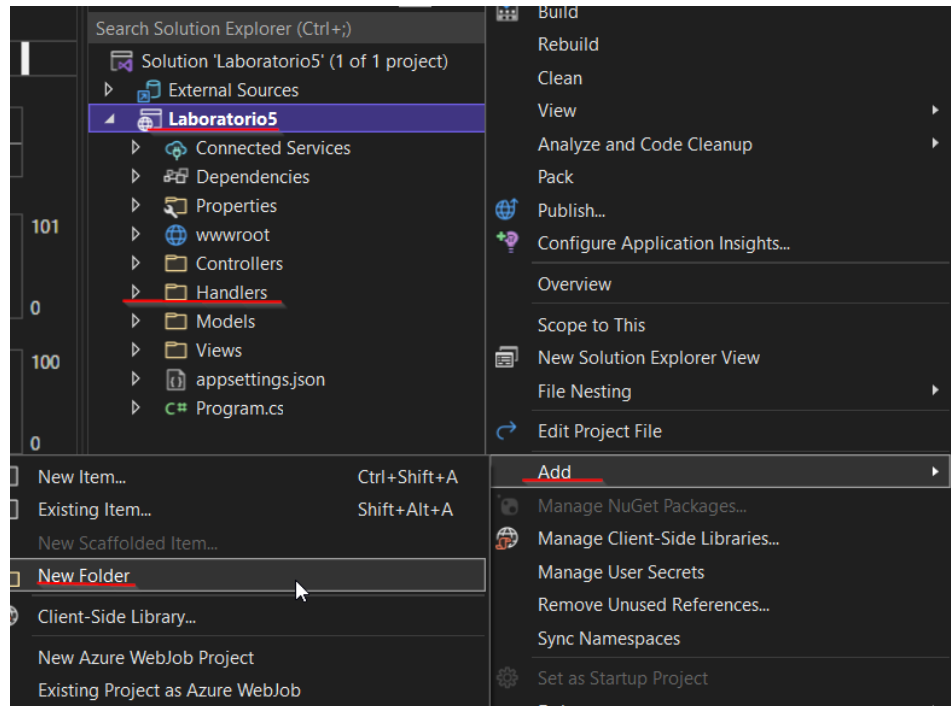


```
1 namespace Laboratorio5.Models
2 {
3     0 references
4     public class PaisModel
5     {
6         0 references
7         public int Id { get; set; }
8         0 references
9         public string Nombre { get; set; }
10        0 references
11        public string Continente { get; set; }
12        0 references
13        public string Idioma { get; set; }
14    }
15 }
```

Nota: en este caso la base de datos solo cuenta con una tabla país, es pertinente destacar que por esta razón el único modelo existente es el modelo país. No obstante, cuando existen más tablas, esto no implica que se deba crear un modelo por cada tabla, lo que se hace es unificar los datos de las distintas tablas en un modelo de manera que se logre una representación del objeto en cuestión o bien si para un mismo objeto existen variantes, se crean modelos que responden a las necesidades según los datos que desean presentar.

Establecer la conexión y manipular los datos

Ahora se va a crear una clase que va a servir para hacer la conexión a la base de datos, y por medio de los métodos ejecutar consultas, insertar datos y demás operaciones con los datos. Dé click derecho sobre su archivo de proyecto y seleccione la opción Add para crear una nueva carpeta, nombre la carpeta como Handlers.



Dé click derecho sobre la carpeta Handlers y agregue una nueva clase de C#, nombre la clase como PaísesHandler. En esta clase se realizará lo necesario para establecer la conexión a la base de datos y manipularla. Agregue el siguiente código:

```
using Laboratorio5.Models;
using System.Data;
using System.Data.SqlClient;

namespace Laboratorio5.Handlers
{
    public class PaísesHandler
    {
        private SqlConnection conexion;
        private string rutaConexion;
        public PaísesHandler()
        {
            var builder = WebApplication.CreateBuilder();
            rutaConexion = builder.Configuration.GetConnectionString("PaísesContext");
            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<PaisModel> ObtenerPaises()
    {
        List<PaisModel> paises = new List<PaisModel>();
        string consulta = "SELECT * FROM dbo.Pais ";
        DataTable tablaResultado = CrearTablaConsulta(consulta);
        foreach (DataRow columna in tablaResultado.Rows)
        {
            paises.Add(
                new PaisModel
                {
                    Id = Convert.ToInt32(columna["Id"]),
                    Nombre = Convert.ToString(columna["Nombre"]),
                    Idioma = Convert.ToString(columna["Idioma"]),
                    Continente = Convert.ToString(columna["Continente"]),
                });
        }
        return paises;
    }
}
```

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

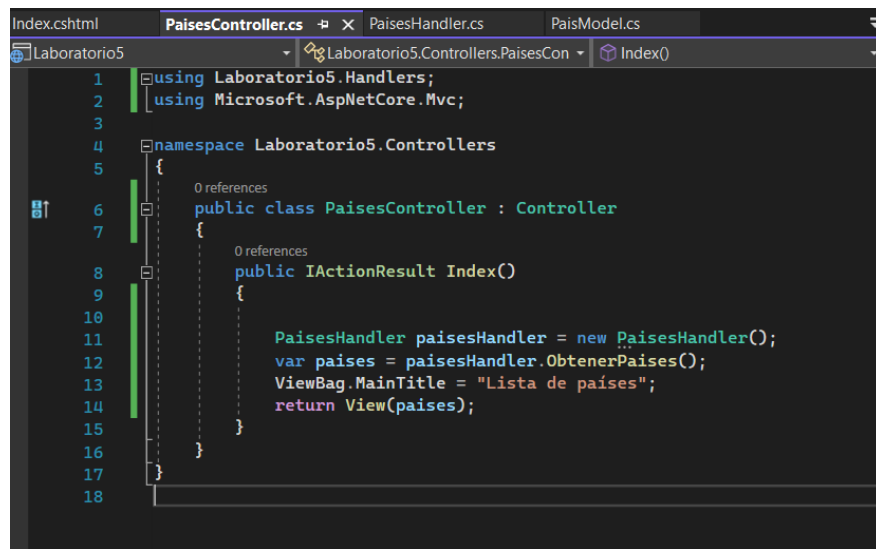
1. Observe que la clase tiene dos atributos privados, uno de ellos es básicamente el Connection String (rutaConexion) el cual se utiliza a su vez para inicializar la conexión con el servidor. Observe que en el encabezado se incluyen varias librerías, una de ellas es un cliente de SQL y es la que brinda los elementos para establecer la conexión y crear las consultas.
2. Observe que el constructor, para obtener el Connection String, se pasa por parámetro el nombre de alguno de los Connection String que fue agregado en el appsettings.json. En este caso se utiliza el de la base de datos Paises, pero usted podría conectarse a cualquier otra base de datos con solo cambiar el este valor por el nombre de otro Connection String que usted haya agregado. Note además que, al



- inicializar un nuevo `PaísesHandler` para manipular algunos datos, por medio del constructor, la conexión estará automáticamente creada.
3. Si la conexión no se crea desde el constructor (por ejemplo), usted debería estar inicializando esta conexión cada vez que quiera realizar una nueva consulta, lo cual genera repetición de código.
 4. El objeto `DataTable` es una representación matricial de las tablas que usted habitualmente obtiene cuando ejecuta una consulta en SQL server, el método `crearTablaConsulta` esto es lo que realiza, observe que se crea un comando el cual necesita el query que debe procesar y la conexión a la base de datos donde se procesa. El adaptador, en este caso, tiene la función de llenar el `DataTable` con los datos obtenidos en la consulta.
 5. Con respecto al método obtener países, observe que las consultas (no es la única manera) se crean a través de un String convencional. Usted en este String puede escribir un query tan complejo como se lo pueda imaginar, aunque en este caso se ejemplifica con una consulta básica. Las herramientas de Visual Studio le permiten escribir estas hileras de manera cómoda, ya que usted puede pulsar enter al final de la primera línea y automáticamente se crea un espacio para seguir escribiendo. Tenga cuidado de dejar espacios entre cada sentencia que escriba, ya que el analizador de consultas del adaptador lo que hace al final es procesar lo usted escriba en el String, por lo tanto, si usted escribe algo como `"Select * From Pais Where Id = 1"` esto producirá errores.
 6. Finalmente observe que la tabla de datos se puede iterar de una manera sencilla con un `for each`, en este caso se recorre cada fila (tupla) de la tupla y con la sintaxis utilizada `fila[nombre atributo]` se accede al respectivo atributo de la tupla.

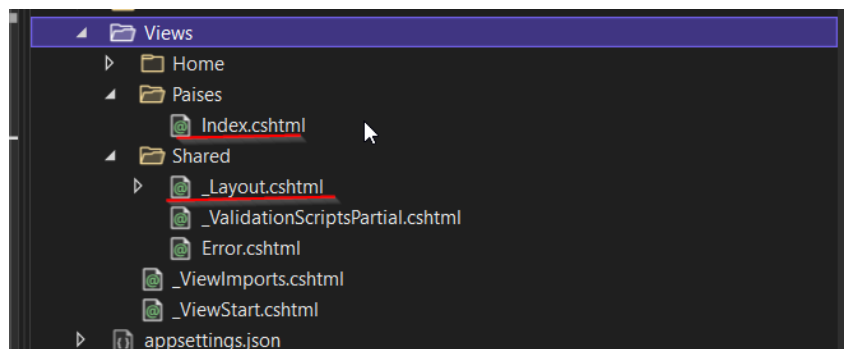
Desplegando los datos

PaísesHandler puede conectarse a la base de datos y obtener los datos de los países y retornar modelos. Ahora es necesario desplegar estos datos, para eso vamos a crear un controlador y vista similar a lo que se hizo en el laboratorio 3. Cree un PaísesController y haga genere el método index, similar a la siguiente imagen:



```
1 using Laboratorio5.Handlers;
2 using Microsoft.AspNetCore.Mvc;
3
4 namespace Laboratorio5.Controllers
5 {
6     public class PaísesController : Controller
7     {
8         public IActionResult Index()
9         {
10             PaísesHandler paísesHandler = new PaísesHandler();
11             var países = paísesHandler.ObtenerPaíses();
12             ViewBag.MainTitle = "Lista de países";
13             return View(países);
14         }
15     }
16 }
17
18
```

Cree una vista index para mostrar los países, de la misma manera que lo genero en el Laboratorio 3 (creando una tabla). También modifique el Layout para que tenga una opción países en el menú de la aplicación.





Antes de correr la solución, agregue algunos datos a la tabla país de su base de datos, para esto cree un nuevo query y corralo de la misma manera que hizo para crear la tabla. Puede usar el siguiente código como ejemplo:

```
INSERT INTO [dbo].[País] ([Nombre],[Idioma] ,[Continente])  
VALUES('Costa Rica', 'Español', 'América'),  
('Argentina', 'Español', 'América'),  
('Canada', 'Inglés/Frances', 'América'),  
('Francia', 'Frances', 'Europa'),  
('España', 'Español', 'Europa')
```

GO

Una vez haya hecho la inserción, verifique por medio de un Select que se hayan agregado correctamente, para ello puede borrar el código copiado en el query y escribir el Select * o bien crear un nuevo query.

Finalmente el resultado final de este laboratorio debe ser el siguiente:

Laboratorio5 Home Privacy Países

Lista de países

Id	Nombre	Idioma	Continente
1	Costa Rica	Español	América
2	Argentina	Español	América
3	Canada	Inglés/Frances	América
4	Francia	Frances	Europa
5	España	Español	Europa

NOTA: No borre este laboratorio de su cuenta github. El laboratorio 6 es sobre formularios y usará de base este laboratorio.



Entregable

Para este laboratorio se debe compartir el repositorio usado con el asistente y la profesora a los siguientes usuarios de github:

- rebeca-ov
- ChristianRojasRios

Cree un pequeño documento con diferentes screenshots donde demuestre:

1. Conexión exitosa a la base de datos (debe mostrar el nombre de su computadora local o la base de datos dada en el curso de bases de datos)
2. Screenshot de el **Select * from Pais** donde se enseñen que realmente se crearon los datos con éxito. Ya sea en la base de datos local o en la base de datos personal del curso de bases de datos.
3. Resultado final al correr el laboratorio en su computadora.

Referencias

- Laboratorio de Ingeniería de software, realizado por Edwin Brenes
- MVC .NET Core tutorials
<https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/adding-model?view=aspnetcore-6.0&tabs=visual-studio>