



Actividad 1.15. Trabajar con una BdD SQL

Crea una Aplicación de WPF para C# XAML, la cual trabajará con una serie de datos de una Base de Datos SQL que tienes también que crear.

Para esta actividad debéis entregar una **memoria con las explicaciones y pantallazos** correspondientes a la creación de esta aplicación:

1. **Crea un proyecto** llamado **"GestionEmpleados2023"** desde Visual Studio.
2. **Crea una base de datos de SQL** que se llame **"GestionEmpleados"** desde Visual Studio.
3. **Crea una tabla** de nombre **"Empleados"** con el siguiente diseño y con los siguientes datos:

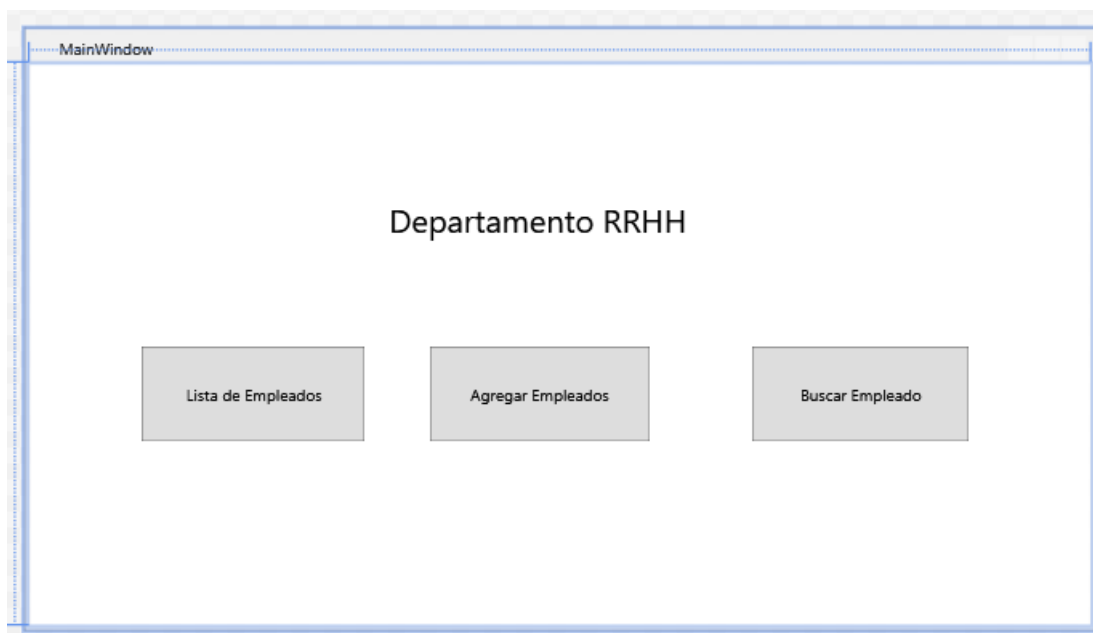
dbo.Empleados [Diseño] ListaEmpleados.xaml.cs ListaEmpleados.xaml				
Actualizar Archivo de script: dbo.Empleados.sql				
	Nombre	Tipo de datos	Permitir valores NULL	Predeterminado
	Id	int	<input type="checkbox"/>	
	Nombre	nvarchar(MAX)	<input checked="" type="checkbox"/>	
	Apellidos	nvarchar(50)	<input checked="" type="checkbox"/>	
	EsUsuario	bit	<input checked="" type="checkbox"/>	
	Edad	int	<input checked="" type="checkbox"/>	
			<input type="checkbox"/>	

Diseñar	T-SQL
---------	-------

```
1 CREATE TABLE [dbo].[Empleados] (  
2     [Id] INT IDENTITY (1, 1) NOT NULL,  
3     [Nombre] NVARCHAR (MAX) NULL,  
4     [Apellidos] NVARCHAR (50) NULL,  
5     [EsUsuario] BIT NULL,  
6     [Edad] INT NULL,  
7     PRIMARY KEY CLUSTERED ([Id] ASC)  
8 )
```

4. **Añade 5 registros** de manera libre.

5. **Diseña una MainWindow** similar a esta desde la cual accederemos navegando por botones al resto de ventanas:



6. **Diseña una nueva ventana** de nombre **"ListaEmpleados"** donde aparezca una **etiqueta** con el título **"Listado de Empleados"** y añade un **DataGrid** donde se **mostrarán** automáticamente los **datos de la tabla "Empleados"**.

```
<DataGrid x:Name="dataGrid" AutoGenerateColumns="True" />
```

7. Introduce el código necesario en la ventana **ListaEmpleados.xaml.cs**, y añade en la **memoria las explicaciones de los pasos** que vas realizando:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Configuration;
4 using System.Data;
5 using System.Data.SqlClient;
6 using System.Linq;
7 using System.Windows;
8
9 namespace GestionEmpleados2023
10 {
11     5 referencias | 0 cambios | 0 autores, 0 cambios
12     public class Empleado
13     {
14         1 referencia | 0 cambios | 0 autores, 0 cambios
15         public string Nombre { get; set; }
16         1 referencia | 0 cambios | 0 autores, 0 cambios
17         public string Apellidos { get; set; }
18         1 referencia | 0 cambios | 0 autores, 0 cambios
19         public bool EsUsuario { get; set; }
20         1 referencia | 0 cambios | 0 autores, 0 cambios
21         public int Edad { get; set; }
22     }
23 }
```



3 referencias | 0 cambios | 0 autores, 0 cambios

public partial class **GestionEmpleados2023**

```
{  
    private SqlConnection conexionConSql;  
  
    1 referencia | 0 cambios | 0 autores, 0 cambios  
    public GestionEmpleados2023()  
    {  
        EstablecerConexion();  
    }  
  
    2 referencias | 0 cambios | 0 autores, 0 cambios  
    private void EstablecerConexion()  
    {  
        string CadenaDeConexion = ConfigurationManager.ConnectionStrings["GestionEmpleados2023.Properties.Settings.GestionEmpleadosConnectionString"].ConnectionString;  
        conexionConSql = new SqlConnection(CadenaDeConexion);  
    }  
  
    1 referencia | 0 cambios | 0 autores, 0 cambios  
    public List<Empleado> ObtenerEmpleados()  
    {  
        EstablecerConexion();  
  
        string consulta = "SELECT * FROM EMPLEADOS";  
        DataTable Empleados = new DataTable();  
  
        List<Empleado> listaEmpleados = new List<Empleado>();  
  
        SqlDataAdapter adaptador = new SqlDataAdapter(consulta, conexionConSql);  
  
        using (adaptador)  
        {  
            adaptador.Fill(Empleados);  
        }  
  
        listaEmpleados = Empleados.AsEnumerable().Select(row => new Empleado  
        {  
            Nombre = row.Field<string>("Nombre"),  
            Apellidos = row.Field<string>("Apellidos"),  
            EsUsuario = (row["EsUsuario"] != DBNull.Value) ? row.Field<bool>("EsUsuario") : false,  
            Edad = row.Field<int>("Edad")  
        }).ToList();  
  
        return listaEmpleados;  
    }  
}
```

4 referencias | 0 cambios | 0 autores, 0 cambios

public partial class **ListaEmpleados** : Window

```
{  
    private GestionEmpleados2023 gestionEmpleados;  
  
    1 referencia | 0 cambios | 0 autores, 0 cambios  
    public ListaEmpleados()  
    {  
        InitializeComponent();  
        gestionEmpleados = new GestionEmpleados2023();  
        CargarEmpleadosEnDataGrid();  
    }  
  
    // Lógica para cargar y mostrar empleados en el DataGrid  
    1 referencia | 0 cambios | 0 autores, 0 cambios  
    private void CargarEmpleadosEnDataGrid()  
    {  
        List<Empleado> empleados = gestionEmpleados.ObtenerEmpleados();  
        dataGrid.ItemsSource = empleados;  
    }  
}
```

8. **Crea la interacción** entre la MainWindow y la ventana **ListaEmpleados**.

9. **Comprueba** que al ejecutar la Aplicación **muestra los datos de la tabla** que habías introducido.

10. **Crea una nueva ventana** llamada **"AgregarEmpleado"**.
11. **Diseña la ventana** para que tenga una forma similar a esta:

A screenshot of a Windows application window titled "Agregar Empleado". The window contains the following elements: a title bar with the text "Agregar Empleado"; a label "Nombre:" followed by a text input field; a label "Apellidos:" followed by a text input field; a label "Es Usuario:" followed by a checkbox; a label "Edad:" followed by a text input field; and a button labeled "Agregar Empleado" at the bottom.

12. Agregar el código necesario para **introducir datos** en la tabla SQL creada:

```
1  using System;
2  using System.Configuration;
3  using System.Data.SqlClient;
4  using System.Windows;
5
6  namespace GestionEmpleados2023
7  {
8      4 referencias | 0 cambios | 0 autores, 0 cambios
9      public partial class AgregarEmpleado : Window
10     {
11         1 referencia | 0 cambios | 0 autores, 0 cambios
12         public AgregarEmpleado()
13         {
14             InitializeComponent();
15         }
16
17         1 referencia | 0 cambios | 0 autores, 0 cambios
18         private void AgregarEmpleado_Click(object sender, RoutedEventArgs e)
19         {
20             string nombre = txtNombre.Text;
21             string apellidos = txtApellidos.Text;
22             bool esUsuario = chkEsUsuario.IsChecked ?? false;
23             int edad;
24
25             if (int.TryParse(txtEdad.Text, out edad))
26             {
27                 AgregarEmpleadoString(nombre, apellidos, esUsuario, edad);
28
29                 // Cerramos la ventana actual después de agregar el empleado
30                 Close();
31             }
32             else
33             {
34                 MessageBox.Show("Por favor, ingrese una edad válida.", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
35             }
36         }
37     }
38 }
```



```
1 referencia | 0 cambios | 0 autores, 0 cambios
private void AgregarEmpleadoString(string nombre, string apellidos, bool esUsuario, int edad)
{
    // Conexión a la base de datos
    using (SqlConnection conexion = new SqlConnection(ConfigurationManager.ConnectionStrings["GestionEmpleados2023.Properties.Settings.GestionEmpleadosConnectionString"].ConnectionString))
    {
        // Comando SQL para la inserción
        string consulta = "INSERT INTO EMPLEADOS (Nombre, Apellidos, EsUsuario, Edad) VALUES (@Nombre, @Apellidos, @EsUsuario, @Edad)";

        // Configurar y ejecutar el comando
        using (SqlCommand cmd = new SqlCommand(consulta, conexion))
        {
            cmd.Parameters.AddWithValue("@Nombre", nombre);
            cmd.Parameters.AddWithValue("@Apellidos", apellidos);
            cmd.Parameters.AddWithValue("@EsUsuario", esUsuario);
            cmd.Parameters.AddWithValue("@Edad", edad);

            try
            {
                conexion.Open();
                cmd.ExecuteNonQuery();
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error al agregar empleado: {ex.Message}", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
            }
        }
    }
}
```

13. **Añade las interacciones** a esa nueva Ventana **"AgregarEmpleado"** desde la **MainWindow**.
14. **Comprueba** que puedes **agregar empleados** a la tabla principal y que se añaden al listado de empleados. Vete poniendo los pantallazos en la **memoria**.
15. Ahora, **investiga e implementa** sobre la ventana **"ListaEmpleados"**, como hacer que un **botón borre un registro seleccionado**. Llama al botón **"Eliminar Registro"**
16. **Añade** también una **nueva ventana** a la que se pueda acceder desde la **MainWindow** en la que haya un **formulario** desde el cual se pueden hacer **búsquedas** de cualquiera de los **campos de la tabla "Empleados"** y en donde aparezca el **resultado de la búsqueda** (o un mensaje si no encuentra **nada** en la búsqueda). Desde ahí podremos **actualizar/modificar** el registro del **Empleado**.
17. **Sube** la memoria realizada al **Moodle de la actividad** (también tienen que estar incluidos los códigos de la aplicación)