

|   |   |                     |   |   |
|---|---|---------------------|---|---|
|            | <b>INSTITUTO TECNOLÓGICO DE NUEVO LAREDO</b><br><b>ING. EN SISTEMAS COMPUTACIONALES</b> |                     |   |  |
|   | <b>MATERIA:</b><br>Programación Orientada a<br>Objetos (C#)                             | <b>UNIDAD:</b><br>6 | <b>PRÁCTICA:</b><br>4                   |   |
| <b>NOMBRE DE LA PRÁCTICA:</b> Listado de bases de datos de MS Access con aplicaciones en C# |   |                     |   |   |
| <b>MAESTRO:</b> Dr. Ramón Roque Hernández,<br>M.C. Bruno López Takeyas                      |   |                     | <b>EMAIL:</b> bruno.lt@nlaredo.tecnm.mx |   |

|   |
|---|
| <b>OBJETIVO:</b> El estudiante elaborará un programa que muestre datos de una base de datos de MS Access a través de un programa en C#  |
| <b>MATERIAL Y EQUIPO NECESARIO:</b> <ul style="list-style-type: none"> <li>• Microsoft Access</li> <li>• Microsoft Visual C#</li> </ul> |

### EJERCICIO: LISTADO DE REGISTROS DE LA TABLA EMPLEADOS (MODO CONECTADO)

1. Para este ejercicio, se requiere tener el archivo `Empleados.mdb` en `C:\`

Esta es una base de datos creada en Microsoft Access, previamente en el ejercicio de Altas a la tabla de `Empleados`. Para fines de esta práctica, puede descargar este archivo desde el sitio web de la materia. Es necesario asegurarse que el archivo se llame `Empleados.mdb` y se encuentre ubicado en `C:\`

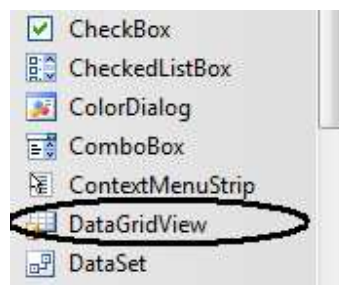
(Si desea ponerlo en otra ubicación, asegúrese de cambiar dentro del programa la cadena de conexión donde se menciona la ruta y el nombre del archivo).

Nota: Si usted utiliza Windows Vista, se recomienda utilizar la base de datos dentro de un Folder (Por ejemplo: `C:\BaseDeDatos\Empleados.mdb` en lugar de en la raíz), esto debido al sistema de permisos que se maneja en ese sistema operativo. En este caso, cambie también la cadena de conexión en este ejercicio para incorporar la ruta correcta del archivo.


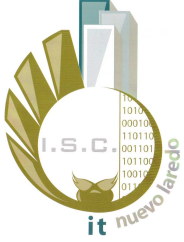
2. Crear un nuevo proyecto Windows en C#.NET. En la Forma (`Form1`), colocar 3 botones de comando y un `dataGridView` de la siguiente manera:

|   |   |                     |   |   |
|---|---|---------------------|---|---|
|            | <b>INSTITUTO TECNOLÓGICO DE NUEVO LAREDO</b><br><b>ING. EN SISTEMAS COMPUTACIONALES</b> |                     |   |  |
|   | <b>MATERIA:</b><br>Programación Orientada a<br>Objetos (C#)                             | <b>UNIDAD:</b><br>6 | <b>PRÁCTICA:</b><br>4                   |   |
| <b>NOMBRE DE LA PRÁCTICA:</b> Listado de bases de datos de MS Access con aplicaciones en C# |   |                     |   |   |
| <b>MAESTRO:</b> Dr. Ramón Roque Hernández,<br>M.C. Bruno López Takeyas                      |   |                     | <b>EMAIL:</b> bruno.lt@nlaredo.tecnm.mx |   |

NOTA: El `dataGridView` es un control de Windows como cualquier otro, que puede arrastrarse a la forma desde el toolbox. Para ayudar a ubicarlo se muestra la siguiente imagen:



El `dataGridView` principalmente sirve para desplegar una cuadrícula que muestra datos (aunque también es posible modificarlos, si se desea). Un `dataGridView` permite configurar los renglones y las columnas que se muestran en ella.

|   |   |                     |   |   |
|---|---|---------------------|---|---|
|            | <b>INSTITUTO TECNOLÓGICO DE NUEVO LAREDO</b><br><b>ING. EN SISTEMAS COMPUTACIONALES</b> |                     |   |  |
|   | <b>MATERIA:</b><br>Programación Orientada a<br>Objetos (C#)                             | <b>UNIDAD:</b><br>6 | <b>PRÁCTICA:</b><br>4                   |   |
| <b>NOMBRE DE LA PRÁCTICA:</b> Listado de bases de datos de MS Access con aplicaciones en C# |   |                     |   |   |
| <b>MAESTRO:</b> Dr. Ramón Roque Hernández,<br>M.C. Bruno López Takeyas                      |   |                     | <b>EMAIL:</b> bruno.lt@nlaredo.tecnm.mx |   |



3. Cambiar las siguientes propiedades:

En la Forma (Form1)

Text → Listado

En el dataGridView

Name → dataGridView1 (Por default ya aparece así, solo checar que tenga ese nombre)


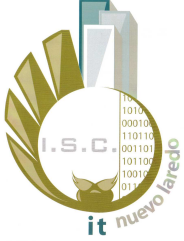
ReadOnly → True

4. Dar doble click en el botón “Regresar” y teclear lo siguiente:

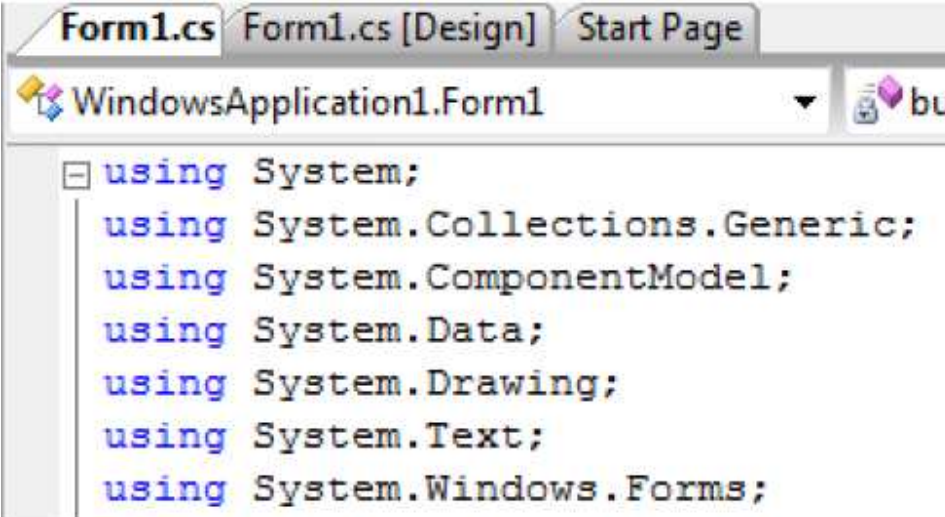
```
this.Close();
```

5. Dar doble click en el botón “Limpiar” y teclear lo siguiente:

```
dataGridView1.DataSource = null;
```

|   |   |                     |   |   |
|---|---|---------------------|---|---|
|            | <b>INSTITUTO TECNOLÓGICO DE NUEVO LAREDO</b><br><b>ING. EN SISTEMAS COMPUTACIONALES</b> |                     |   |  |
|   | <b>MATERIA:</b><br>Programación Orientada a<br>Objetos (C#)                             | <b>UNIDAD:</b><br>6 | <b>PRÁCTICA:</b><br>4                   |   |
| <b>NOMBRE DE LA PRÁCTICA:</b> Listado de bases de datos de MS Access con aplicaciones en C# |   |                     |   |   |
| <b>MAESTRO:</b> Dr. Ramón Roque Hernández,<br>M.C. Bruno López Takeyas                      |   |                     | <b>EMAIL:</b> bruno.lt@nlaredo.tecnm.mx |   |

6. Buscar arriba, al principio del código, la sección donde se colocan las sentencias `using`. Sólo busque la siguiente sección, no cambie ni borre nada:



```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

```

Inmediatamente después, agregue esta línea: (NOTA: Esta línea es la referencia a las clases necesarias para acceso a datos mediante ADO.NET en este programa.).


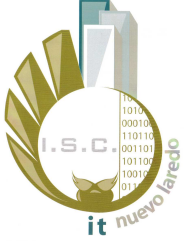
```
using System.Data.OleDb;
```

7. Regresar al modo de diseño de la forma, dar doble click en el botón "LISTAR" y teclear lo siguiente:

```

try
{
    string cc = "provider=Microsoft.Jet.OLEDB.4.0;data
source=C:\\Empleados.mdb;";
    OleDbConnection cn = new OleDbConnection(cc);
    cn.Open();
    string comando = "Select * from Empleados";
    OleDbCommand cm = new OleDbCommand(comando, cn);
    OleDbDataReader dr = cm.ExecuteReader();

```

|   |   |                     |   |   |
|---|---|---------------------|---|---|
|            | <b>INSTITUTO TECNOLÓGICO DE NUEVO LAREDO</b><br><b>ING. EN SISTEMAS COMPUTACIONALES</b> |                     |   |  |
|   | <b>MATERIA:</b><br>Programación Orientada a<br>Objetos (C#)                             | <b>UNIDAD:</b><br>6 | <b>PRÁCTICA:</b><br>4                   |   |
| <b>NOMBRE DE LA PRÁCTICA:</b> Listado de bases de datos de MS Access con aplicaciones en C# |   |                     |   |   |
| <b>MAESTRO:</b> Dr. Ramón Roque Hernández,<br>M.C. Bruno López Takeyas                      |   |                     | <b>EMAIL:</b> bruno.lt@nlaredo.tecnm.mx |   |

```

if (dr.HasRows == true)
{
    DataTable dt = new DataTable();
    dt.Load(dr);
    dataGridView1.DataSource = dt;
    dataGridView1.Columns[2].DefaultCellStyle.Format = "c2";
    dataGridView1.Columns[2].DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleRight;
    dataGridView1.AutoResizeColumns();
}
else
{
    MessageBox.Show(" No hay datos para mostrar! ");
}
cn.Close();
}
catch (Exception ex)
{
    MessageBox.Show(" Ha habido un error! " + ex.Message);
}

```

8. En este punto, el programa debe mostrar los registros que se encuentran en la tabla Empleados. Entre a Access, y verifique que los registros que se muestran en pantalla sean los que se efectivamente se encuentran grabados en la tabla Empleados.

**[Fin de la práctica].**

### EXPLICACIÓN DEL CÓDIGO PARA EL LISTADO DE DATOS (BOTÓN "LISTAR")

Observe que todo el código se encuentra dentro de un bloque `try..catch`. Es una practica común que todas las operaciones realizadas con una base de datos se encierren siempre en un

|   |   |                     |   |   |
|---|---|---------------------|---|---|
|            | <b>INSTITUTO TECNOLÓGICO DE NUEVO LAREDO</b><br><b>ING. EN SISTEMAS COMPUTACIONALES</b> |                     |   |  |
|   | <b>MATERIA:</b><br>Programación Orientada a<br>Objetos (C#)                             | <b>UNIDAD:</b><br>6 | <b>PRÁCTICA:</b><br>4                   |   |
| <b>NOMBRE DE LA PRÁCTICA:</b> Listado de bases de datos de MS Access con aplicaciones en C# |   |                     |   |   |
| <b>MAESTRO:</b> Dr. Ramón Roque Hernández,<br>M.C. Bruno López Takeyas                      |   |                     | <b>EMAIL:</b> bruno.lt@nlaredo.tecnm.mx |   |

bloque `try . .catch` para prevenir que un error inesperado altere la ejecución del programa. De esta manera, si ocurre alguna excepción, el usuario obtiene el respectivo aviso, y sigue con la operación normal de su programa.

Para realizar este listado, usaremos el modo “conectado” de ADO.NET. En este modo conectado, los objetos principales son el `Connection`, `Command`, y `DataReader`. Recordemos que el `Connection` es el encargado de conectar, y mantener un enlace con la base de datos . El `Command` es una vía para ejecutar sentencias SQL en la base de datos. El `DataReader` es un objeto donde temporalmente se almacenan los registros que son el resultado de la consulta ejecutada. El `DataReader` proporciona una manera limitada de almacenamiento, pero ofrece la ventaja de que no consume muchos recursos y funciona rápidamente.


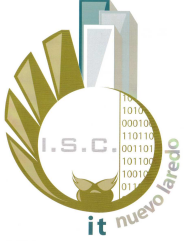
El primer paso es construir la cadena de conexión. Observe que se utiliza una variable `string` llamada `cc` para almacenar los datos de la cadena de conexión, la cual contiene todos los detalles que necesita ADO.NET para conectarse a la base de datos. Observe como en esta cadena de conexión se menciona que el archivo se encuentra en `C:\` y se llama `Empleados.mdb`.

Nota: Se coloca doble diagonal (`c:\empleados.mdb`) pues la diagonal (`\`) es un carácter especial en C#, y su sintaxis requiere que se coloque doble (`\\`) para que C# lo interprete como tal.

```
string cc = "provider=Microsoft.Jet.OLEDB.4.0;data
           source=C:\\Empleados.mdb;";
```

El siguiente paso es crear una conexión hacia esta base de datos (descrita en la cadena de conexión anterior) y posteriormente abrirla:

```
OleDbConnection cn = new OleDbConnection(cc);
cn.Open();
```

|   |   |                     |   |   |
|---|---|---------------------|---|---|
|            | <b>INSTITUTO TECNOLÓGICO DE NUEVO LAREDO</b><br><b>ING. EN SISTEMAS COMPUTACIONALES</b> |                     |   |  |
|   | <b>MATERIA:</b><br>Programación Orientada a<br>Objetos (C#)                             | <b>UNIDAD:</b><br>6 | <b>PRÁCTICA:</b><br>4                   |   |
| <b>NOMBRE DE LA PRÁCTICA:</b> Listado de bases de datos de MS Access con aplicaciones en C# |   |                     |   |   |
| <b>MAESTRO:</b> Dr. Ramón Roque Hernández,<br>M.C. Bruno López Takeyas                      |   |                     | <b>EMAIL:</b> bruno.lt@nlaredo.tecnm.mx |   |

El siguiente paso es colocar en una variable la sentencia SQL con la cual se va a llenar el DataSet para el listado que se desea obtener. En este caso, la sentencia `Select * from empleados` es suficiente para traer todos los campos de los registros de la tabla Empleados.

```
string comando = "Select * from empleados";
```

Ahora se debe crear un objeto `Command` para poder enviar esta consulta, utilizando la conexión que previamente se ha creado y abierto hacia la base de datos que se está utilizando:

```
OleDbCommand cm = new OleDbCommand(comando, cn);
```

Ahora se debe crear un objeto `DataReader` para almacenar el resultado de la consulta que ejecutará el objeto `Command` creado anteriormente.


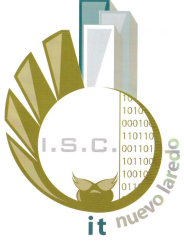
Nota: Observe que el método `ExecuteReader` indica que se espera un resultado después de ejecutar la consulta; dicho resultado se debe almacenar en un `DataReader`.

```
OleDbDataReader dr = cm.ExecuteReader();
```

El siguiente paso es verificar si la consulta entrega resultados (Es posible que no haya ningún dato en la tabla de Empleados). Si el `DataReader` tiene renglones (`HasRows = true`), significa que por lo menos hay un renglón que debe mostrarse. En el caso contrario (`else`), no hay ningún dato para mostrar.

```
if (dr.HasRows == true)
{
/* Ahora bien, si hay datos para mostrar, el DataReader debe ser
cargado a una estructura llamada DataTable, que permite que el
DataGridView muestre los datos directamente de ella. Nota: El
DataGridView NO puede mostrar los datos directamente del DataReader. */
```



|   |   |                     |   |   |
|---|---|---------------------|---|---|
|            | <b>INSTITUTO TECNOLÓGICO DE NUEVO LAREDO</b><br><b>ING. EN SISTEMAS COMPUTACIONALES</b> |                     |   |  |
|   | <b>MATERIA:</b><br>Programación Orientada a<br>Objetos (C#)                             | <b>UNIDAD:</b><br>6 | <b>PRÁCTICA:</b><br>4                   |   |
| <b>NOMBRE DE LA PRÁCTICA:</b> Listado de bases de datos de MS Access con aplicaciones en C# |   |                     |   |   |
| <b>MAESTRO:</b> Dr. Ramón Roque Hernández,<br>M.C. Bruno López Takeyas                      |   |                     | <b>EMAIL:</b> bruno.lt@nlaredo.tecnm.mx |   |

```

DataTable dt = new DataTable();
dt.Load(dr);
dataGridView1.DataSource = dt;
/*Estas líneas sirven para colocar formato especial a la tercera
columna (Recordar que las columnas se numeran comenzando en cero). La
tercera columna debe mostrar un signo de pesos, dos decimales y la
cantidad debe estar cargada a la derecha de la celda donde se
muestra.*/
dataGridView1.Columns[2].DefaultCellStyle.Format = "c2";
dataGridView1.Columns[2].DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleRight;
/*Esta instrucción permite que automáticamente se ajuste el ancho de
las columnas del DataGridView */
dataGridView1.AutoSizeColumnsMode();
}
else
{
    MessageBox.Show(" No hay datos para mostrar! ");
}
}

```

Finalmente, se debe cerrar la conexión que fue abierta anteriormente:

```

cn.Close();

```