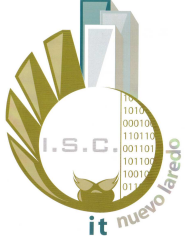
	INSTITUTO TECNOLÓGICO DE NUEVO LAREDO ING. EN SISTEMAS COMPUTACIONALES			
	MATERIA: Programación Orientada a Objetos (C#)	UNIDAD: 6	PRÁCTICA: 1	
NOMBRE DE LA PRÁCTICA: Altas en bases de datos de MS Access con aplicaciones en C#				
MAESTRO: Dr. Ramón Roque Hernández, M.C. Bruno López Takeyas			EMAIL: bruno.lt@nlaredo.tecnm.mx	

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

EJERCICIO: ALTAS A LA TABLA DE EMPLEADOS (MODO CONECTADO)

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


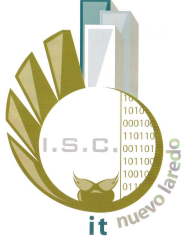
Esta es una base de datos creada en Microsoft Access. Para fines de esta práctica, puede descargar este archivo desde el sitio web de la materia, o bien, puede crear la base de datos desde cero, utilizando Microsoft Access. En cualquiera de los dos casos, es necesario asegurarse que el archivo se llame `Empleados.mdb` (ó `Empleados.accdb`) 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 del botón "Agregar" en este ejercicio para incorporar la ruta correcta del archivo. Si decide crear la base de datos en Access desde cero, utilice la siguiente estructura: (Nota: Mantener los mismos nombres y tipos de datos).

Nombre de la Base de datos: `EMPLEADOS.MDB`

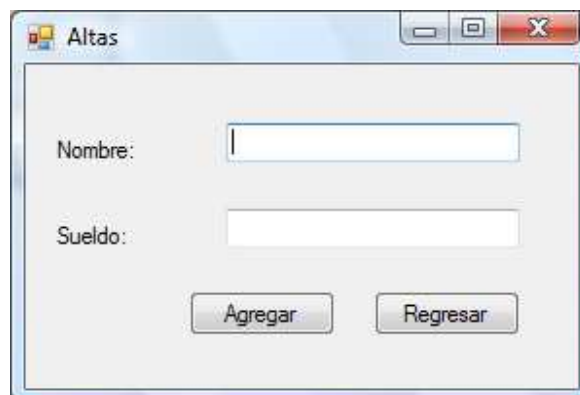
Nombre de la Tabla: `EMPLEADOS`

Llave principal: `Clave`

	INSTITUTO TECNOLÓGICO DE NUEVO LAREDO ING. EN SISTEMAS COMPUTACIONALES			
	MATERIA: Programación Orientada a Objetos (C#)	UNIDAD: 6	PRÁCTICA: 1	
NOMBRE DE LA PRÁCTICA: Altas en bases de datos de MS Access con aplicaciones en C#				
MAESTRO: Dr. Ramón Roque Hernández, M.C. Bruno López Takeyas			EMAIL: bruno.lt@nlaredo.tecnm.mx	

Nombre del campo	Tipo de dato
Clave	Autonumérico
Nombre	Texto (tamaño 100)
Sueldo	Moneda

2. Crear un nuevo proyecto Windows en C#.NET, en la Forma (Form1) , colocar los siguientes controles:



3. Cambiar las siguientes propiedades:

En la Forma (Form1):

Text → Altas


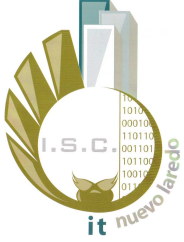
En el textBox para Nombre:

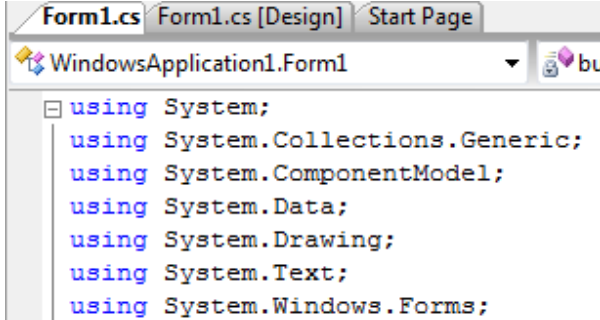
Name → txtNombre

En el textBox para Sueldo:

Name → txtSueldo

4. Dar doble click en el botón “AGREGAR”, y buscar arriba, al principio del código, la sección donde se colocan las sentencias using. Solo busque la siguiente sección, no cambie ni borre nada:

	INSTITUTO TECNOLÓGICO DE NUEVO LAREDO ING. EN SISTEMAS COMPUTACIONALES			
	MATERIA: Programación Orientada a Objetos (C#)	UNIDAD: 6	PRÁCTICA: 1	
NOMBRE DE LA PRÁCTICA: Altas en bases de datos de MS Access con aplicaciones en C#				
MAESTRO: Dr. Ramón Roque Hernández, M.C. Bruno López Takeyas			EMAIL: bruno.lt@nlaredo.tecnm.mx	



```

Form1.cs Form1.cs [Design] Start Page
WindowsApplication1.Form1
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 indica que se va a utilizar acceso a datos mediante ADO.NET en este programa.).


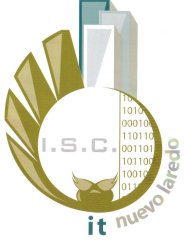
```
using System.Data.OleDb;
```

5. Regresar al modo de diseño de la Forma, dar doble click nuevamente en el botón “AGREGAR” y colocar el siguiente código:

```

string nombre = txtNombre.Text;
double sueldo = double.Parse(txtSueldo.Text);
string cc = "Provider=Microsoft.Jet.OLEDB.4.0;" +
"Data source = c:\\empleados.mdb";
OleDbConnection cn = new OleDbConnection(cc);
cn.Open();
string sql = string.Format("insert into EMPLEADOS (Nombre,Sueldo)
values ('{0}' ,
{1})", nombre, sueldo);
OleDbCommand comando = new OleDbCommand(sql, cn);
int renglones = comando.ExecuteNonQuery();
if (renglones > 0)
{
    MessageBox.Show(" Datos almacenados exitosamente ");
}
else
{
    MessageBox.Show(" Los datos NO han sido almacenados ");
}
cn.Close();

```

	INSTITUTO TECNOLÓGICO DE NUEVO LAREDO ING. EN SISTEMAS COMPUTACIONALES			
	MATERIA: Programación Orientada a Objetos (C#)	UNIDAD: 6	PRÁCTICA: 1	
NOMBRE DE LA PRÁCTICA: Altas en bases de datos de MS Access con aplicaciones en C#				
MAESTRO: Dr. Ramón Roque Hernández, M.C. Bruno López Takeyas			EMAIL: bruno.lt@nlaredo.tecnm.mx	

```
txtNombre.Text = " ";
txtSueldo.Text = " ";
txtNombre.Focus();
```

6. En el botón “REGRESAR”, colocar lo siguiente:
`this.Close();`

7. Ejecutar el programa, e introducir algunos registros.

8. Ir a Microsoft Access y verificar que los datos ingresados en el programa realizado en C#, se hayan almacenado en la base de datos de Access.

EXPLICACIÓN DEL CÓDIGO PARA INSERTAR DATOS (BOTÓN “AGREGAR”)


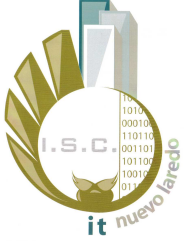
El primer paso, es almacenar en variables los datos que el usuario desea insertar en la base de datos en la tabla Empleados:

```
string nombre = txtNombre.Text;
double sueldo = double.Parse(txtSueldo.Text);
```

Luego, se debe preparar la cadena de conexión para la base de datos. 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;" +
```

	INSTITUTO TECNOLÓGICO DE NUEVO LAREDO ING. EN SISTEMAS COMPUTACIONALES			
	MATERIA: Programación Orientada a Objetos (C#)	UNIDAD: 6	PRÁCTICA: 1	
NOMBRE DE LA PRÁCTICA: Altas en bases de datos de MS Access con aplicaciones en C#				
MAESTRO: Dr. Ramón Roque Hernández, M.C. Bruno López Takeyas		EMAIL: bruno.lt@nlaredo.tecnm.mx		

```
"Data source = c:\\empleados.mdb";
```

Después, se declara un objeto llamado `cn` que es del tipo `OleDbConnection` y representa la conexión o enlace necesario con la base de datos, para realizar la operación de inserción de datos. Al crearlo, se proporciona como parámetro la variable `cc` que contiene la cadena de conexión.

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

Posteriormente se debe abrir la conexión:

```
cn.Open();
```


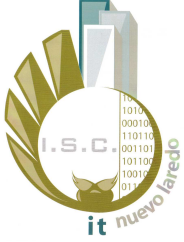
Ahora, se prepara la sentencia SQL para insertar los datos introducidos por el usuario en la tabla Empleados. Observe como la variable `sql` es de tipo `string` y se va construyendo con los datos que el usuario tecleó. Se utiliza el método `string.Format` pues resulta más sencillo incorporar los valores correspondientes al comando SQL. Notar que `{0}` y `{1}` son parámetros de sustitución que `string.Format` reemplaza respectivamente con los valores de las variables `nombre` y `sueldo`.

Por ejemplo, si el usuario introdujo: Nombre: Juan Salario: 1500 se construye un comando de la siguiente manera: `Insert into Empleados (Nombre, Sueldo) Values('Juan', 1500)`

```
string sql = string.Format("insert into EMPLEADOS (Nombre,Sueldo)
values ('{0}' , {1})",nombre, sueldo);
```

Ahora, se declara un objeto de tipo `OleDbCommand`, llamado `comando`, que será el encargado de ejecutar en la base de datos, la sentencia SQL que se acaba de construir. El objeto `comando` requiere que se le indiquen dos parámetros: Cual es la sentencia SQL que se va a ejecutar y cual conexión se utilizará:

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

	INSTITUTO TECNOLÓGICO DE NUEVO LAREDO ING. EN SISTEMAS COMPUTACIONALES			
	MATERIA: Programación Orientada a Objetos (C#)	UNIDAD: 6	PRÁCTICA: 1	
NOMBRE DE LA PRÁCTICA: Altas en bases de datos de MS Access con aplicaciones en C#				
MAESTRO: Dr. Ramón Roque Hernández, M.C. Bruno López Takeyas			EMAIL: bruno.lt@nlaredo.tecnm.mx	

Acto seguido, se ejecuta la sentencia SQL de inserción de datos mediante el objeto `OleDbCommand` que se creó en la línea anterior. Observe como se utiliza una variable entera llamada `renglones` para conocer el número de renglones que fueron insertados en la tabla:

```
int renglones = comando.ExecuteNonQuery();
```

Luego, mediante un IF se determina si los datos efectivamente se insertaron en la tabla. Si la variable `renglones` contiene un valor mayor que cero, significa que por lo menos un renglon fue agregado en la tabla:

```
if (renglones > 0)
{
    MessageBox.Show(" Datos almacenados exitosamente ");
}
else
{
    MessageBox.Show(" Los datos NO han sido almacenados ");
}
```

Finalmente, la conexión se cierra para liberar los recursos del sistema:

```
cn.Close();
```

Y ahora, el programa limpia los `textBox` y coloca el cursor en `txtNombre` para que el usuario continúe capturando datos si lo desea:

```
txtNombre.Text = "";
txtSueldo.Text = "";
txtNombre.Focus();
```