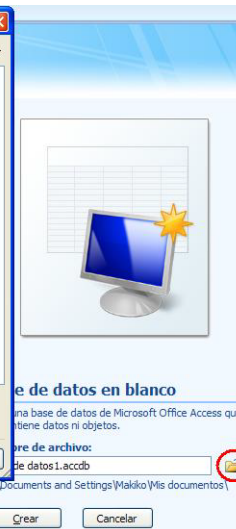
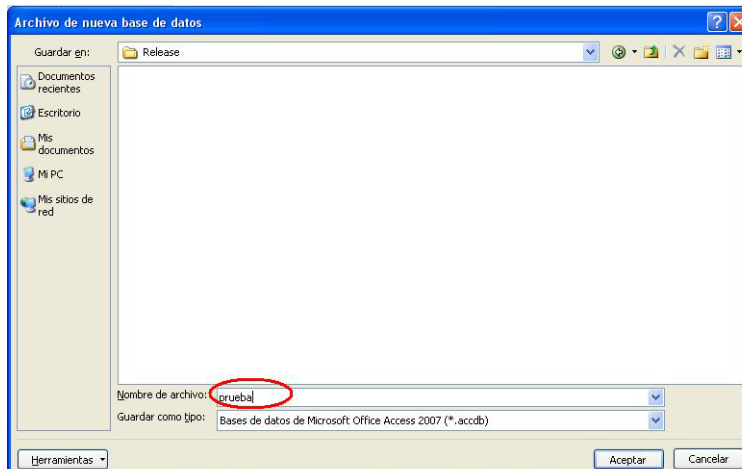
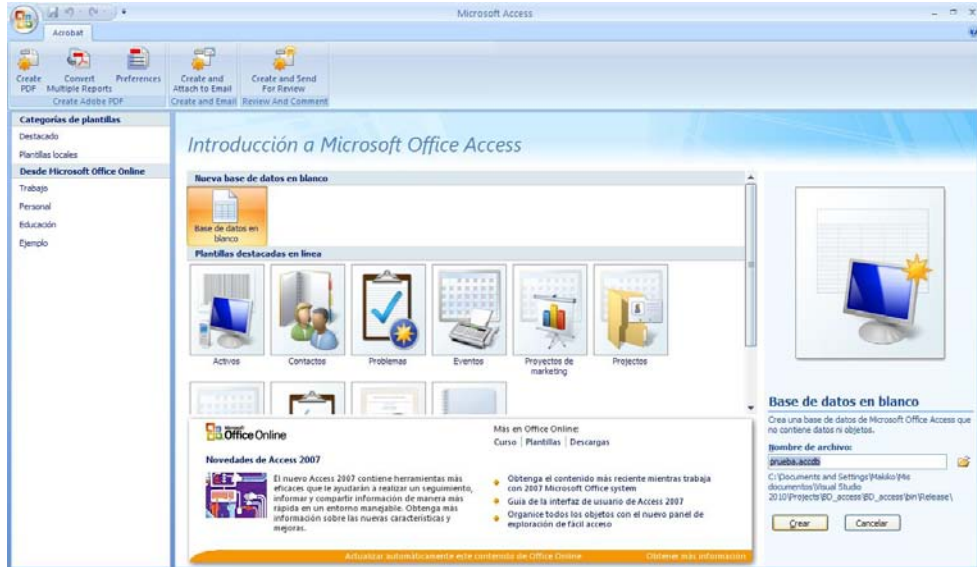


# Bases de Datos Access y C#. Insert, select, update, delete

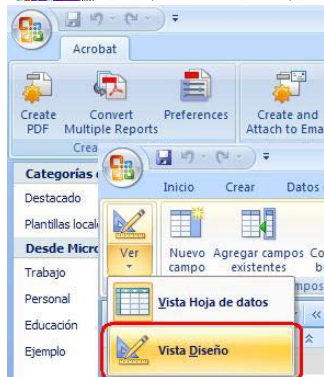
## 1. Crea una base de Datos en Access

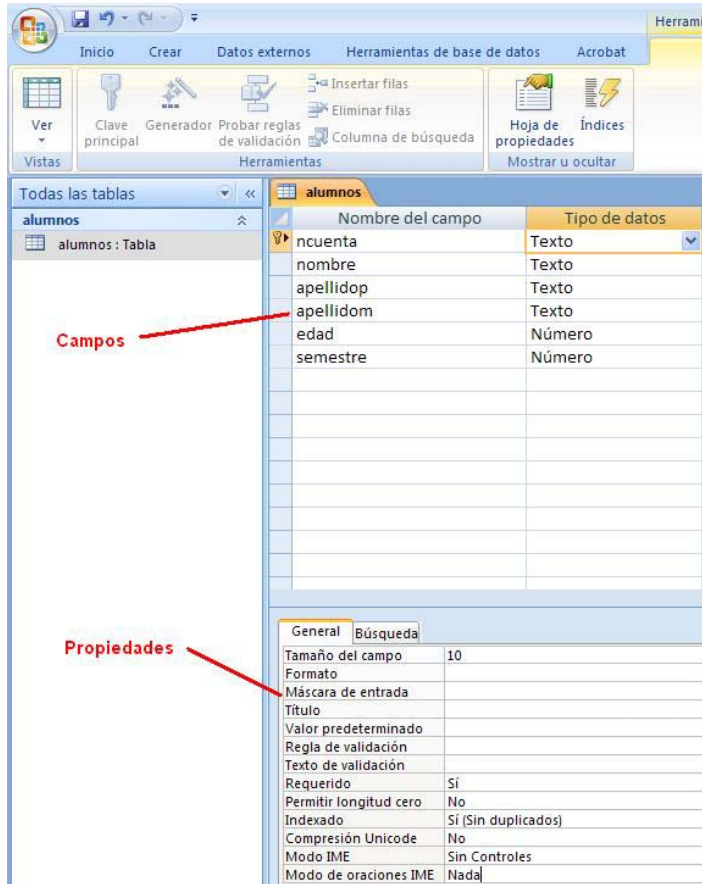


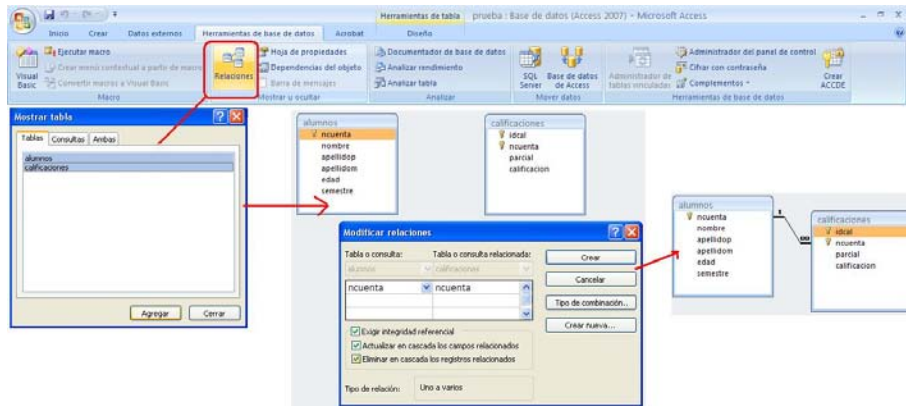
El nuevo Access 2007 contiene herramientas más eficaces que le ayudarán a realizar un seguimiento, informar y compartir información de manera más rápida en un entorno manejable. Obtenga más

- Obtenga el contenido más reciente mientras trabaja con 2007 Microsoft Office system
- Guía de la interfaz de usuario de Access 2007

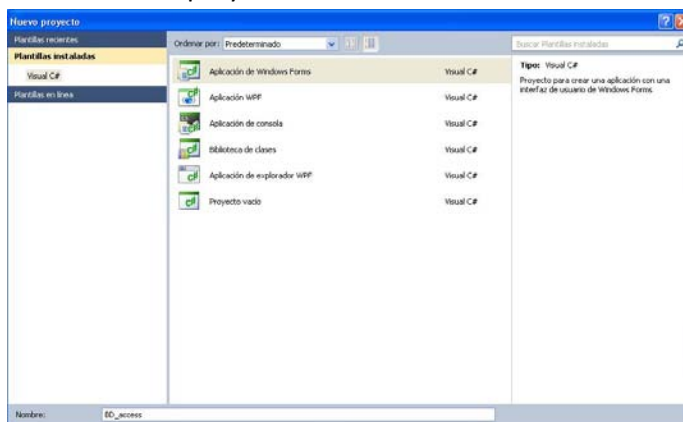
Comience todos los objetos con el nuevo panel de







## 2. Crea un Nuevo proyecto en C#



## 3. Agrega una nueva clase llamada **Database** y a continuación escribe el siguiente código:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;
using System.Data.OleDb;

namespace bd_access2
{
    class Database
    {
        private string StrConexion;

        private OleDbConnection Conexion;
        private OleDbDataAdapter Adapter;
        private DataSet miDataSet = new DataSet();

        public void IniciarConexion(string DataBase)
        {
            //Creo la cadena de conexion para Office 2007
            StrConexion = "Provider=Microsoft.ACE.OLEDB.12.0; Data Source = " + DataBase;

            //Objeto conexion
            Conexion = new OleDbConnection(StrConexion);
        }
    }
}
```

```
public int ejecutar_sql(string sql)
{
    //inserto en la BD
    int i = 0;
    try
    {
        Conexion.Open();
        OleDbCommand cmd = new OleDbCommand(sql, Conexion);
        i = cmd.ExecuteNonQuery();
    }
    catch
    {
        i = -1;
    }
    return i;
}

public DataTable consultar(string sql, string tabla)
{
    Adapter = new OleDbDataAdapter(sql, Conexion);

    //Creo el DataTable
    DataTable dt = new DataTable();

    //Relleno el adaptador con los datos en memoria
    Adapter.Fill(dt);

    return dt;
}
}
```

4. Crea una nueva clase llamada **Alumnos** y escribe el siguiente código:

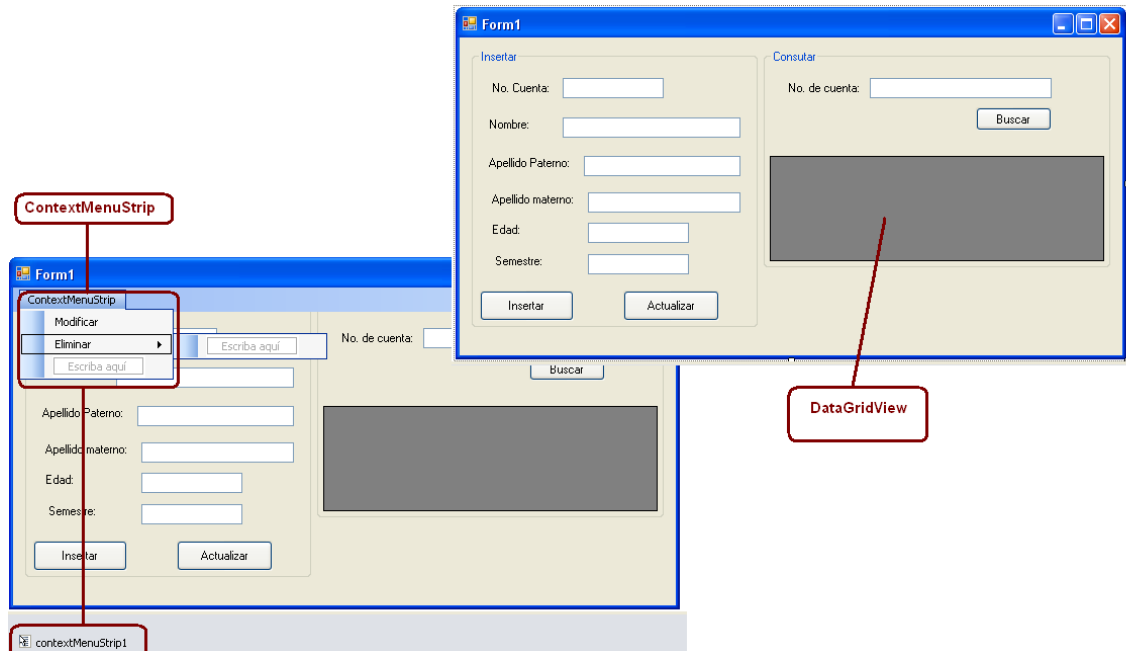
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace bd_access2
{
    class Alumnos
    {
        public string ncuenta;
        public string nombre;
        public string apellido;
        public string apellidom;
        public int edad;
        public int semestre;

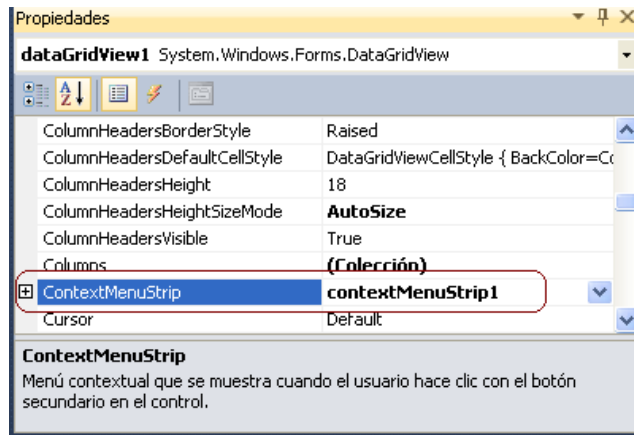
        public Alumnos()
        {
        }

        public Alumnos(string ncuenta, string nombre, string apellido, string apellidom, int edad, int semestre)
        {
            this.ncuenta = ncuenta;
            this.nombre = nombre;
            this.apellido = apellido;
            this.apellidom = apellidom;
            this.edad = edad;
            this.semestre = semestre;
        }
    }
}
```

5. Edita el formulario agregando los siguientes componentes:



6. Edita la propiedad ContextMenuStrip del DataGridView1. (Le permitirá mostrar un menú flotante al dar clic con el botón derecho sobre el DataGridView).



7. Edita el código del Form1.cs

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

namespace bd_access2
{
    public partial class Form1 : Form
    {
        Database DB = new Database();
        String sql;

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            DB.IniciarConexion("../prueba.accdb");
            //Creo el miembro de datos del DataGridView
            dataGridView1.DataMember = "alumnos";
        }

        private Alumnos asig_alum_text()
        {
            string ncta, nombre, apellidop, apellidom;
            int edad = 0, semestre;
            ncta = txtnCta.Text;
            nombre = txtNombre.Text;
            apellidop = txtApellidoop.Text;
            apellidom = txtApellidoom.Text;
            try
```



```

private void eliminarToolStripMenuItem_Click(object sender, EventArgs e)
{
    //Para eliminar un alumno
    int renglon = dataGridView1.CurrentRow.RowIndex;
    int ncuenta = Convert.ToInt32(dataGridView1[0, renglon].Value.ToString());
    if (ncuenta != -1)
    {
        if (MessageBox.Show("Esta seguro que desea eliminar el alumno " + dataGridView1[1,
renglon].Value.ToString(), "Eliminar registro", MessageBoxButtons.YesNo) ==
DialogResult.Yes)
            DB.ejecutar_sql("DELETE FROM alumnos WHERE Cuenta=" + ncuenta + "");
        dataGridView1.Rows.RemoveAt(renglon);
    }
}

private void btnActualizar_Click(object sender, EventArgs e)
{
    Alumnos a = asig_alum_text();
    sql = "UPDATE alumnos SET nombre=" + a.nombre + ",";
    sql += " apellidop=" + a.apellidop + ",apellidom=" + a.apellidom + ",";
    sql += " edad=" + a.edad + ", semestre=" + a.semestre + "";
    sql += " WHERE ncuenta=" + a.ncuenta + """;

    int update = DB.ejecutar_sql(sql);
    if (update == 1) //Si se logro la insercion limpio el formulario
    {
        MessageBox.Show("Se actualizaron correctamente sus datos");
        foreach (Control txt in this.Controls)
        {
            if (txt.GetType() == typeof(TextBox))
                txt.Text = "";
        }
    }
    else
        MessageBox.Show("Hubo un error al actualizar los datos");
}
}
}
}

```

#### NOTAS:

- No olviden generar todo el código de los eventos.
- Verifiquen bien el nombre de sus controles.
- Integren un módulo de validación de datos.
- Agreguen control de errores en las partes que lo requieran.