

Estructuras Dinámicas: Listas

Guía De Ejercicio Práctico

Para el diseño del formulario en **Windows Forms** utilizando **Visual Studio 2012**, debes agregar los siguientes controles básicos para que el CRUD funcione correctamente:

1. Entradas de Texto (TextBoxes)

Estos componentes permiten al usuario ingresar la información del estudiante:

- **txtId**: Utilizado para capturar el número de identificación del estudiante.
- **txtNombre**: Utilizado para ingresar el nombre completo.
- **txtCarrera**: Utilizado para ingresar la carrera profesional.

2. Acciones (Buttons)

Botones para disparar la lógica de programación del sistema:

- **btnAgregar**: Ejecuta la creación de un nuevo objeto en la lista.
- **btnConsultar**: Permite buscar un estudiante específico por su ID.
- **btnEliminar**: Quita al estudiante seleccionado de la lista.
- **btnModificar**: (Adicional) Para actualizar los datos de un registro existente.
- **btnGuardar**: (Adicional) Para persistir los cambios en el archivo físico.

3. Visualización (ListBox)

- **lstEstudiantes** (o **lstDatos**): Es el control donde se "amarrará" la lista de objetos para mostrar los registros al usuario en tiempo real.

Sugerencia de diseño: Al trabajar en **Visual Studio 2012**, recuerda que el diseñador de formularios a veces requiere que compiles el proyecto (**Ctrl + Shift + B**) para que los cambios en los componentes y sus propiedades se reflejen correctamente en la vista de diseño.

Visual Studio 2012 (.NET 4.5), es necesario ajustar la sintaxis de C#, ya que esta versión no soporta características modernas como la interpolación de cadenas (`$""`) o la librería nativa `System.Text.Json`.

Para crear el proyecto y los archivos necesarios en **Visual Studio 2012** siguiendo el material de clase, sigue este paso a paso:

Paso 1: Crear el Proyecto en Visual Studio 2012

1. Abre Visual Studio 2012.
 2. Ve a **Archivo > Nuevo > Proyecto**.
 3. En la columna izquierda, selecciona **Visual C# > Windows**.
 4. Selecciona **Aplicación de Windows Forms**.
 5. Nombra el proyecto como MiAppWinForms y haz clic en **Aceptar**.
-

Paso 2: Crear la Clase de Datos (Modelo)

Este archivo define la estructura de lo que vamos a guardar.

En el **Explorador de Soluciones**, haz clic derecho sobre el nombre de tu proyecto.

1. Selecciona **Agregar > Clase**.
 2. Nombra el archivo como Estudiante.cs.
 3. Sustituye el código por el siguiente (adaptado a VS 2012):
-

Paso 3: Diseñar la Interfaz (UI)

1. Haz doble clic en Form1.cs para ver el diseñador.
 2. Abre el **Cuadro de Herramientas** (Toolbox) y arrastra los siguientes componentes:
 3. **3 TextBoxes**: Nómbralos en la propiedad (Name) como txtId, txtNombre y txtCarrera.
 4. **4 Buttons**: Nómbralos como btnAgregar, btnConsultar, btnEliminar y btnModificar.
 5. **1 ListBox**: Nómbralo como lstEstudiantes.
-

Paso 4: Programar la Lógica del Formulario

1. Haz clic derecho sobre el formulario y selecciona **Ver código** (o presiona F7).
2. Asegúrate de incluir las directivas necesarias al principio del archivo:

```
C#  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Windows.Forms;  
using System.IO;
```

3. Dentro de la clase Form1, declara la lista a nivel global:

```
C#  
]private List<Estudiante> listaEstudiantes = new List<Estudiante>();
```

4. Copia los métodos de **Agregar**, **Eliminar** y **Modificar**, asegurándote de usar `string.Format` para los mensajes en lugar de \$.

Paso 5: Implementar la Persistencia (CSV)

Para que los datos no se borren al cerrar el programa en VS 2012:

1. Crea el método `GuardarDatos()` usando `StreamWriter` para escribir en un archivo llamado `datos.csv`.
2. Llama a este método al final de cada operación CRUD para automatizar el guardado.

El material de clase proporcionado, se hace el uso de `{get; set;}` se refiere a las **Propiedades Auto-implementadas** de C#.

Aquí tienes una breve explicación técnica:

¿Qué es `get; set;`?

Es una forma abreviada de definir los atributos de una clase. Permite que una variable sea leída (`get`) y escrita (`set`) sin tener que declarar manualmente un campo privado.

Aplicación en el Modelo (Estudiantes.cs)

Dentro de la clase Estudiante, utilizamos esta sintaxis para definir los datos que queremos gestionar en la lista dinámica:

- **public int Id { get; set; }**: Permite almacenar y recuperar el número único del estudiante.
- **public string Nombre { get; set; }**: Define la propiedad para el nombre del alumno.
- **public string Carrera { get; set; }**: Define la propiedad para la carrera profesional.

¿Por qué se usa así?

1. **Limpieza de código**: En Visual Studio 2012, estas propiedades funcionan perfectamente y evitan escribir decenas de líneas de código repetitivo.
2. **Encapsulamiento**: Aunque parece una variable pública simple, internamente C# crea un campo privado seguro para proteger el dato.
3. **Compatibilidad con la Lista**: Al trabajar con una List<Estudiante>, podemos acceder directamente a estas propiedades para realizar búsquedas con **LINQ** (como est.Id) o mostrarlas en el formulario.

Importancia en el CRUD

Estas propiedades son las que permiten que las funciones de la lista operen correctamente:

1. **Búsqueda (Read/Update)**: LINQ utiliza el get de Id para encontrar coincidencias: lista.FirstOrDefault(est => est.Id == idBusqueda).
2. **Asignación (Create/Update)**: El programa usa el set para guardar los datos de los TextBoxes en el objeto: nuevo.Nombre = txtNombre.Text.
3. **Visualización**: El método ToString() accede a estos valores para mostrar la información en el ListBox.

¿Qué es this?

Se utiliza para referirse a los miembros (propiedades, campos o métodos) de la propia clase desde dentro de ella. Es como si el objeto dijera "mi propio...".

2. Uso en la clase Estudiante

Lo utilizamos dentro del método ToString() para dejar claro que estamos accediendo a las propiedades del objeto que se está convirtiendo a texto

Lógica del Programa

1. El Modelo de Datos (Estudiante.cs)

En VS 2012, las propiedades auto-implementadas funcionan correctamente, pero debemos usar `string.Format` para la visualización.

2. Formulario CRUD (Form1.cs)

Esta versión utiliza **C# 5.0**. Se ha sustituido el manejo de JSON por **Texto Plano (CSV)** para evitar la instalación de paquetes externos como *Newtonsoft.Json* en versiones antiguas de NuGet.

Actualización (Update) y Selección

Para que sea profesional, los datos deben cargarse al hacer clic en la lista.

3. Persistencia de Datos (CSV)

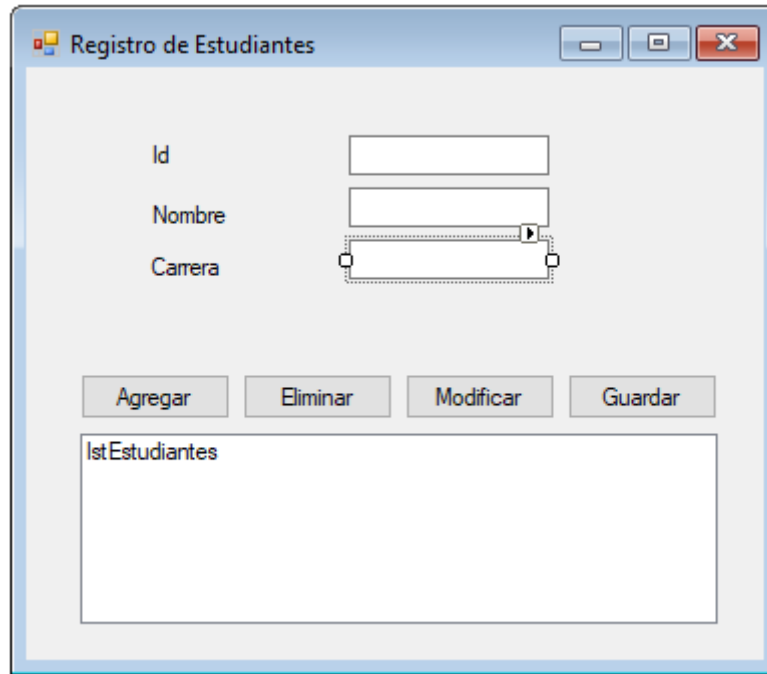
Como `System.Text.Json` no está disponible en .NET 4.5, usaremos archivos delimitados por punto y coma (;).

Tips

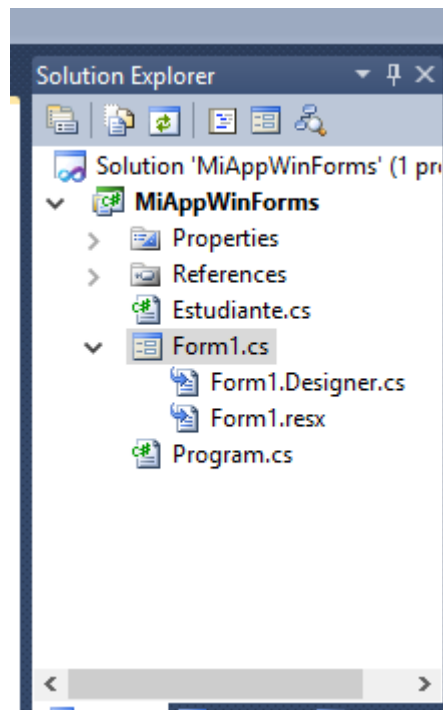
- **Adiós al \$:** Recuerda a los alumnos que la interpolación de cadenas les marcará error; deben concatenar con `+` o usar `string.Format`.
- **Referencia IO:** Para usar archivos, es obligatorio tener `using System.IO;`
- **Diseñador "Pegado":** Si el diseño visual no carga, recomienda usar **Ctrl + Shift + B** para compilar antes de abrir la vista de diseño.
- **Try-Catch:** Enfatiza que el manejo de excepciones es vital para evitar que el programa "trueene" ante errores del usuario.

Código Funcional:

Para que el proyecto funcione en **Visual Studio 2010 Ultimate**



The screenshot shows a Windows application window titled "Registro de Estudiantes". The window contains three text input fields labeled "Id", "Nombre", and "Carrera". Below these fields are four buttons: "Agregar", "Eliminar", "Modificar", and "Guardar". At the bottom of the window is a list box containing the text "lstEstudiantes".



Form1.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;
using System.IO;

namespace MiAppWinForms
{
    public partial class Form1 : Form
    {
        // LA LISTA (Nuestra base de datos en memoria)
        private List<Estudiante> listaEstudiantes = new List<Estudiante>();

        public Form1()
        {
            InitializeComponent();
        }

        // --- C: CREATE (Agregar) ---
        private void btnAgregar_Click(object sender, EventArgs e)
        {
            try
            {
                int idInput = int.Parse(txtId.Text);

                // Validación de duplicados usando LINQ
                if (listaEstudiantes.Any(est => est.Id == idInput))
                {
                    MessageBox.Show("El ID ya existe en el sistema.");
                    return;
                }

                Estudiante nuevo = new Estudiante();
                nuevo.Id = idInput;
                nuevo.Nombre = txtNombre.Text;
                nuevo.Carrera = txtCarrera.Text;

                listaEstudiantes.Add(nuevo);
                ActualizarInterfaz();
                LimpiarCampos();
            }
            catch (FormatException)
            {
                MessageBox.Show("Error: El ID debe ser un número entero.");
            }
        }

        // --- U: UPDATE (Modificar) ---
        private void btnModificar_Click(object sender, EventArgs e)
        {
            try
            {
                int idBusqueda = int.Parse(txtId.Text);
                // Buscar por referencia
            }
        }
    }
}
```

```

idBusqueda);

var encontrado = listaEstudiantes.FirstOrDefault(est => est.Id ==

if (encontrado != null)
{
    encontrado.Nombre = txtNombre.Text;
    encontrado.Carrera = txtCarrera.Text;
    ActualizarInterfaz();
    MessageBox.Show("Datos actualizados correctamente.");
}
else
{
    MessageBox.Show("No se encontró el estudiante para modificar.");
}
}
catch (Exception ex)
{
    MessageBox.Show("Error: " + ex.Message);
}
}

// --- D: DELETE (Eliminar con Confirmación) ---
private void btnEliminar_Click(object sender, EventArgs e)
{
    if (lstEstudiantes.SelectedItem != null)
    {
        // Casteo de object a Estudiante
        Estudiante seleccionado = (Estudiante)lstEstudiantes.SelectedItem;

        // Confirmación de eliminación
        DialogResult respuesta = MessageBox.Show("¿Desea eliminar a " +
seleccionado.Nombre + "?",
                                                "Confirmar", MessageBoxButtons.YesNo,
MessageBoxIcon.Warning);

        if (respuesta == DialogResult.Yes)
        {
            listaEstudiantes.Remove(seleccionado);
            ActualizarInterfaz();
        }
    }
    else
    {
        MessageBox.Show("Por favor, seleccione un estudiante de la lista.");
    }
}

// --- PERSISTENCIA: Guardar en archivo CSV ---
private void btnGuardar_Click(object sender, EventArgs e)
{
    try
    {
        using (StreamWriter sw = new StreamWriter("datos.csv"))
        {
            foreach (var est in listaEstudiantes)
            {
                // Usamos string.Format porque VS 2010 no tiene $
            }
        }
    }
}

```

```

        sw.WriteLine(string.Format("{0};{1};{2}", est.Id, est.Nombre,
est.Carrera));
    }
}
    MessageBox.Show("Archivo guardado con éxito.");
}
catch (Exception ex)
{
    MessageBox.Show("Error al guardar: " + ex.Message);
}
}

// --- MÉTODOS DE APOYO ---
private void ActualizarInterfaz()
{
    lstEstudiantes.DataSource = null; // Forzar refresco
    lstEstudiantes.DataSource = listaEstudiantes;
}

private void LimpiarCampos()
{
    txtId.Clear();
    txtNombre.Clear();
    txtCarrera.Clear();
    txtId.Focus();
}

private void Form1_Load(object sender, EventArgs e)
{
}
}
}

```

Estudiante.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System;

namespace MiAppWinForms
{
    public class Estudiante
    {
        // Propiedades de la clase
        public int Id { get; set; }
        public string Nombre { get; set; }
        public string Carrera { get; set; }

        // Este método permite que el ListBox muestre los datos correctamente
        public override string ToString()
        {
            // Usamos string.Format porque VS 2010 no soporta el símbolo $
            return string.Format("{0} - {1} ({2})", this.Id, this.Nombre, this.Carrera);
        }
    }
}

```

```
}
```

Form1.Designer.cs

```
namespace MiAppWinForms
{
    partial class Form1
    {
        private System.ComponentModel.IContainer components = null;

        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        private void InitializeComponent()
        {
            this.txtId = new System.Windows.Forms.TextBox();
            this.txtNombre = new System.Windows.Forms.TextBox();
            this.txtCarrera = new System.Windows.Forms.TextBox();
            this.Label1 = new System.Windows.Forms.Label();
            this.Nombre = new System.Windows.Forms.Label();
            this.Carrera = new System.Windows.Forms.Label();
            this.btnAgregar = new System.Windows.Forms.Button();
            this.btnEliminar = new System.Windows.Forms.Button();
            this.btnModificar = new System.Windows.Forms.Button();
            this.btnGuardar = new System.Windows.Forms.Button();
            this.lstEstudiantes = new System.Windows.Forms.ListBox();
            this.SuspendLayout();
            //
            // txtId
            //
            this.txtId.Location = new System.Drawing.Point(161, 34);
            this.txtId.Name = "txtId";
            this.txtId.Size = new System.Drawing.Size(100, 20);
            this.txtId.TabIndex = 0;
            //
            // txtNombre
            //
            this.txtNombre.Location = new System.Drawing.Point(161, 60);
            this.txtNombre.Name = "txtNombre";
            this.txtNombre.Size = new System.Drawing.Size(100, 20);
            this.txtNombre.TabIndex = 1;
            //
            // txtCarrera
            //
            this.txtCarrera.Location = new System.Drawing.Point(161, 86);
            this.txtCarrera.Name = "txtCarrera";
            this.txtCarrera.Size = new System.Drawing.Size(100, 20);
            this.txtCarrera.TabIndex = 2;
            //
        }
    }
}
```

```
// Label1
//
this.Label1.AutoSize = true;
this.Label1.Location = new System.Drawing.Point(60, 37);
this.Label1.Name = "Label1";
this.Label1.Size = new System.Drawing.Size(16, 13);
this.Label1.Text = "Id";
//
// Nombre
//
this.Nombre.AutoSize = true;
this.Nombre.Location = new System.Drawing.Point(60, 67);
this.Nombre.Name = "Nombre";
this.Nombre.Size = new System.Drawing.Size(44, 13);
this.Nombre.Text = "Nombre";
//
// Carrera
//
this.Carrera.AutoSize = true;
this.Carrera.Location = new System.Drawing.Point(60, 93);
this.Carrera.Name = "Carrera";
this.Carrera.Size = new System.Drawing.Size(41, 13);
this.Carrera.Text = "Carrera";
//
// btnAgregar
//
this.btnAgregar.Location = new System.Drawing.Point(27, 153);
this.btnAgregar.Name = "btnAgregar";
this.btnAgregar.Size = new System.Drawing.Size(75, 23);
this.btnAgregar.TabIndex = 6;
this.btnAgregar.Text = "Agregar";
this.btnAgregar.Click += new System.EventHandler(this.btnAgregar_Click);
//
// btnEliminar
//
this.btnEliminar.Location = new System.Drawing.Point(108, 153);
this.btnEliminar.Name = "btnEliminar";
this.btnEliminar.Size = new System.Drawing.Size(75, 23);
this.btnEliminar.TabIndex = 8;
this.btnEliminar.Text = "Eliminar";
this.btnEliminar.Click += new System.EventHandler(this.btnEliminar_Click);
//
// btnModificar
//
this.btnModificar.Location = new System.Drawing.Point(189, 153);
this.btnModificar.Name = "btnModificar";
this.btnModificar.Size = new System.Drawing.Size(75, 23);
this.btnModificar.TabIndex = 9;
this.btnModificar.Text = "Modificar";
this.btnModificar.Click += new System.EventHandler(this.btnModificar_Click);
//
// btnGuardar
//
this.btnGuardar.Location = new System.Drawing.Point(270, 153);
this.btnGuardar.Name = "btnGuardar";
this.btnGuardar.Size = new System.Drawing.Size(75, 23);
this.btnGuardar.TabIndex = 10;
this.btnGuardar.Text = "Guardar";
```

```

this.btnGuardar.Click += new System.EventHandler(this.btnGuardar_Click);
//
// lstEstudiantes
//
this.lstEstudiantes.Location = new System.Drawing.Point(27, 183);
this.lstEstudiantes.Name = "lstEstudiantes";
this.lstEstudiantes.Size = new System.Drawing.Size(318, 95);
this.lstEstudiantes.TabIndex = 11;
//
// Form1
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(368, 296);
this.Controls.Add(this.lstEstudiantes);
this.Controls.Add(this.btnGuardar);
this.Controls.Add(this.btnModificar);
this.Controls.Add(this.btnEliminar);
this.Controls.Add(this.btnAgregar);
this.Controls.Add(this.Carrera);
this.Controls.Add(this.Nombre);
this.Controls.Add(this.Label1);
this.Controls.Add(this.txtCarrera);
this.Controls.Add(this.txtNombre);
this.Controls.Add(this.txtId);
this.Name = "Form1";
this.Text = "Registro de Estudiantes";
this.ResumeLayout(false);
this.PerformLayout();
}

#endregion

private System.Windows.Forms.TextBox txtId;
private System.Windows.Forms.TextBox txtNombre;
private System.Windows.Forms.TextBox txtCarrera;
private System.Windows.Forms.Label Label1;
private System.Windows.Forms.Label Nombre;
private System.Windows.Forms.Label Carrera;
private System.Windows.Forms.Button btnAgregar;
private System.Windows.Forms.Button btnEliminar;
private System.Windows.Forms.Button btnModificar;
private System.Windows.Forms.Button btnGuardar;
private System.Windows.Forms.ListBox lstEstudiantes;
}
}

```

Para **Visual Studio 2012**, utilizaremos el **.NET Framework 4.5**. A diferencia de la versión 2010, aquí ya tenemos un entorno más estable, pero recuerda que **sigue sin existir la interpolación de cadenas** (\$), por lo que usaremos `string.Format`.

Aquí tienes los 3 archivos necesarios con el código **completamente limpio**, sin etiquetas de "cite" y listo para copiar y pegar:

1. El Modelo: `Estudiante.cs`

Crema una clase nueva llamada `Estudiante.cs` y pega esto:

```
C#
using System;

namespace MiAppWinForms
{
    public class Estudiante
    {
        public int Id { get; set; }
        public string Nombre { get; set; }
        public string Carrera { get; set; }

        public override string ToString()
        {
            // VS 2012 usa string.Format (el $ no funciona aquí)
            return string.Format("{0} - {1} ({2})", this.Id, this.Nombre,
this.Carrera);
        }
    }
}
```

2. La Lógica: `Form1.cs`

Haz clic derecho en tu formulario -> **Ver Código** y reemplaza todo por esto:

```
C#
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;
using System.IO;

namespace MiAppWinForms
{
    public partial class Form1 : Form
    {
        // Lista dinámica de objetos
    }
}
```

```

        private List<Estudiante> listaEstudiantes = new
List<Estudiante>();

    public Form1()
    {
        InitializeComponent();
    }

    private void btnAgregar_Click(object sender, EventArgs e)
    {
        try
        {
            int id = int.Parse(txtId.Text);

            if (listaEstudiantes.Any(est => est.Id == id))
            {
                MessageBox.Show("Este ID ya existe.");
                return;
            }

            Estudiante nuevo = new Estudiante();
            nuevo.Id = id;
            nuevo.Nombre = txtNombre.Text;
            nuevo.Carrera = txtCarrera.Text;

            listaEstudiantes.Add(nuevo);
            ActualizarInterfaz();
            LimpiarCampos();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Error: " + ex.Message);
        }
    }

    private void btnEliminar_Click(object sender, EventArgs e)
    {
        if (lstEstudiantes.SelectedItem != null)
        {
            Estudiante seleccionado =
(Estudiante)lstEstudiantes.SelectedItem;
            listaEstudiantes.Remove(seleccionado);
            ActualizarInterfaz();
        }
    }

    private void btnGuardar_Click(object sender, EventArgs e)
    {
        try
        {
            using (StreamWriter sw = new StreamWriter("datos.csv"))
            {
                foreach (var est in listaEstudiantes)
                {

```

```

        sw.WriteLine(string.Format("{0};{1};{2}", est.Id,
est.Nombre, est.Carrera));
    }
}
MessageBox.Show("Guardado con éxito en CSV.");
}
catch (Exception ex)
{
    MessageBox.Show("Error al guardar: " + ex.Message);
}
}

private void ActualizarInterfaz()
{
    lstEstudiantes.DataSource = null;
    lstEstudiantes.DataSource = listaEstudiantes;
}

private void LimpiarCampos()
{
    txtId.Clear();
    txtNombre.Clear();
    txtCarrera.Clear();
    txtId.Focus();
}
}
}
}

```

3. El Diseño: Form1.Designer.cs

Este archivo es delicado. Asegúrate de que los nombres de los controles en tu diseño coincidan. Reemplaza el contenido de Form1.Designer.cs con este:

C#

```

namespace MiAppWinForms
{
    partial class Form1
    {
        private System.ComponentModel.IContainer components = null;

        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        private void InitializeComponent()
        {
            this.txtId = new System.Windows.Forms.TextBox();
            this.txtNombre = new System.Windows.Forms.TextBox();

```

```

this.txtCarrera = new System.Windows.Forms.TextBox();
this.btnAgregar = new System.Windows.Forms.Button();
this.btnEliminar = new System.Windows.Forms.Button();
this.btnGuardar = new System.Windows.Forms.Button();
this.lstEstudiantes = new System.Windows.Forms.ListBox();
this.SuspendLayout();

// txtId
this.txtId.Location = new System.Drawing.Point(120, 30);
this.txtId.Name = "txtId";
this.txtId.Size = new System.Drawing.Size(150, 20);

// txtNombre
this.txtNombre.Location = new System.Drawing.Point(120, 60);
this.txtNombre.Name = "txtNombre";
this.txtNombre.Size = new System.Drawing.Size(150, 20);

// txtCarrera
this.txtCarrera.Location = new System.Drawing.Point(120, 90);
this.txtCarrera.Name = "txtCarrera";
this.txtCarrera.Size = new System.Drawing.Size(150, 20);

// btnAgregar
this.btnAgregar.Location = new System.Drawing.Point(30, 130);
this.btnAgregar.Name = "btnAgregar";
this.btnAgregar.Size = new System.Drawing.Size(75, 23);
this.btnAgregar.Text = "Agregar";
this.btnAgregar.Click += new
System.EventHandler(this.btnAgregar_Click);

// btnEliminar
this.btnEliminar.Location = new System.Drawing.Point(115,
130);
this.btnEliminar.Name = "btnEliminar";
this.btnEliminar.Size = new System.Drawing.Size(75, 23);
this.btnEliminar.Text = "Eliminar";
this.btnEliminar.Click += new
System.EventHandler(this.btnEliminar_Click);

// btnGuardar
this.btnGuardar.Location = new System.Drawing.Point(200, 130);
this.btnGuardar.Name = "btnGuardar";
this.btnGuardar.Size = new System.Drawing.Size(75, 23);
this.btnGuardar.Text = "Guardar";
this.btnGuardar.Click += new
System.EventHandler(this.btnGuardar_Click);

// lstEstudiantes
this.lstEstudiantes.Location = new System.Drawing.Point(30,
170);
this.lstEstudiantes.Name = "lstEstudiantes";
this.lstEstudiantes.Size = new System.Drawing.Size(245, 95);

// Form1

```

```

        this.ClientSize = new System.Drawing.Size(310, 300);
        this.Controls.Add(this.lstEstudiantes);
        this.Controls.Add(this.btnGuardar);
        this.Controls.Add(this.btnEliminar);
        this.Controls.Add(this.btnAgregar);
        this.Controls.Add(this.txtCarrera);
        this.Controls.Add(this.txtNombre);
        this.Controls.Add(this.txtId);
        this.Name = "Form1";
        this.Text = "CRUD Estudiantes 2012";
        this.ResumeLayout(false);
        this.PerformLayout();
    }

    private System.Windows.Forms.TextBox txtId;
    private System.Windows.Forms.TextBox txtNombre;
    private System.Windows.Forms.TextBox txtCarrera;
    private System.Windows.Forms.Button btnAgregar;
    private System.Windows.Forms.Button btnEliminar;
    private System.Windows.Forms.Button btnGuardar;
    private System.Windows.Forms.ListBox lstEstudiantes;
}
}

```

Pasos finales para que funcione:

1. Asegúrate de que tu proyecto se llame `MiAppWinForms`. Si se llama distinto, cambia la primera línea (`namespace ...`) en todos los archivos.
2. Presiona **F5** para ejecutar.
3. Si te pide guardar antes de ejecutar, dale a "Sí a todo".