



INSTITUTO
KHIPU

Semestre III

Sesión 23

PROGRAMA DE ESTUDIOS

DESARROLLO DE SISTEMAS DE INFORMACIÓN

**HERRAMIENTAS DE
PROGRAMACIÓN C#**

Tema

**EJERCICIOS UTILIZANDO CONTROLES:
COMBOBOX Y LISTBOX**

COMBOBOX Y LISTBOX

El **ComboBox** y el **ListBox** son controles comunes en formularios de Windows

Forms que permiten mostrar listas de elementos. Aquí te explico brevemente sus diferencias y ejemplos de uso con Programación Orientada a Objetos (POO) en un formulario de Windows Forms.

ComboBox:

- Muestra una lista desplegable de opciones, de las cuales solo una puede ser seleccionada a la vez.
- Se usa cuando el espacio en pantalla es limitado y el usuario debe seleccionar un solo valor de una lista de opciones.

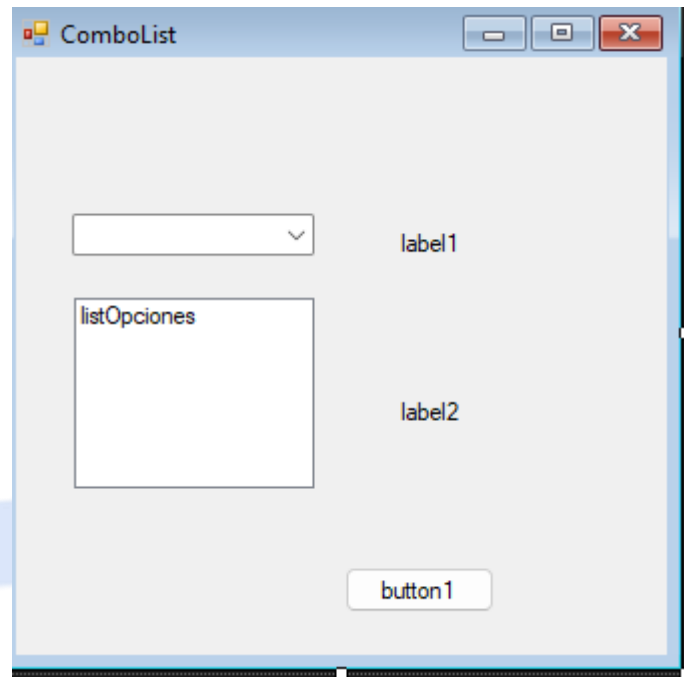
ListBox:

- Muestra una lista de elementos en una caja, donde el usuario puede seleccionar uno o varios elementos (según la configuración).
- Se usa cuando se quieren mostrar varias opciones y puede que el usuario deba seleccionar más de una.

A continuación, se muestra cómo utilizar estos dos controles en un formulario con un enfoque POO:

Ejemplo con ComboBox y ListBox:

Diseño tentativo



Código para la clase "cDatos"

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

namespace compraDescuento
{
    public class cDatos
    {
        public List<string> ObtenerOpciones()
        {
            return new List<string> { "Opción 1", "Opción 2", "Opción 3", "Opción 4" };
        }
    }
}
```

Código para la clase principal

```
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
```

```
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace compraDescuento
{
    public partial class ComboList : Form
    {
        public ComboList()
        {
            InitializeComponent();

            // Llenar ComboBox y ListBox con datos de la clase Datos
            cDatos datos = new cDatos();
            foreach (var opcion in datos.ObtenerOpciones())
            {
                cmbOpciones.Items.Add(opcion);
                listOpciones.Items.Add(opcion);
            }

            // Agregar controles al formulario
            this.Controls.Add(cmbOpciones);
            this.Controls.Add(listOpciones);
            this.Controls.Add(btnMostrarSeleccion);
            this.Controls.Add(lblSeleccionComboBox);
            this.Controls.Add(lblSeleccionListBox1);
        }

        private void btnMostrarSeleccion_Click_1(object sender, EventArgs e)
        {
            // Mostrar la selección del ComboBox
            if (cmbOpciones.SelectedItem != null)
            {
                lblSeleccionComboBox.Text = "Selección ComboBox: " +
                cmbOpciones.SelectedItem.ToString();
            }

            // Mostrar las selecciones del ListBox
            if (listOpciones.SelectedItems.Count > 0)
            {
                string seleccionListBox = "Selección ListBox: ";
                foreach (var item in listOpciones.SelectedItems)
                {
                    seleccionListBox += item.ToString() + " ";
                }
            }
        }
    }
}
```

```
        lblSeleccionListBox1.Text = seleccionListBox;  
    }  
}  
}
```

Diferencias entre ComboBox y ListBox:

- **ComboBox:** Útil cuando quieres ahorrar espacio en la interfaz y necesitas que el usuario seleccione solo un elemento de una lista.
- **ListBox:** Útil cuando necesitas mostrar múltiples elementos y permitir que el usuario seleccione uno o varios elementos simultáneamente.

Este ejercicio muestra cómo puedes integrar ambos controles en una aplicación, haciendo uso de POO para separar la lógica de negocio (datos) de la interfaz de usuario.

EJERCICIO

Resolver con clases

Este ejercicio consiste en diseñar una aplicación que, al ingresar un número en el textBox, y hacer clic en un botón “Agregar”, se cargara en un comboBox y en un listBox. En medio de dos listBox, hay 4 botones para pasar los items de uno a otro, ya sea uno por uno o todos al mismo tiempo.

Ejemplo del diseño de formulario:

