

CARRERA PROFESIONAL

DESARROLLO DE SISTEMAS DE INFORMACIÓN

**HERRAMIENTAS DE
PROGRAMACION C#**

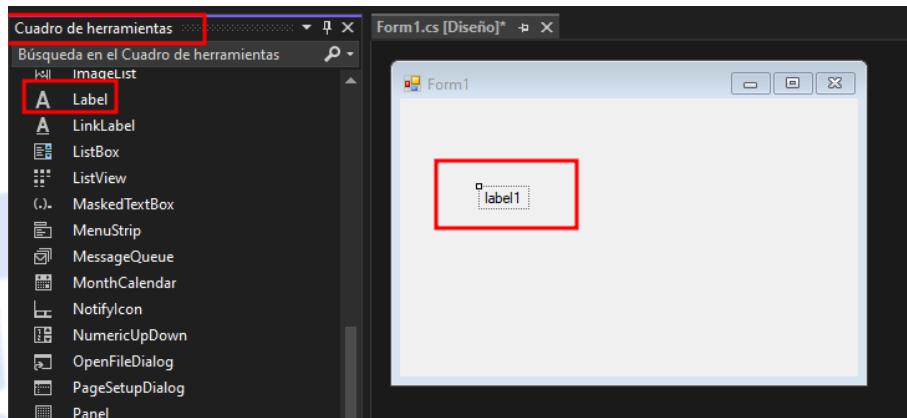
Tema

**HERRAMIENTAS BASICAS – Label,
Textbox, Button.**

HERRAMIENTAS BASICAS

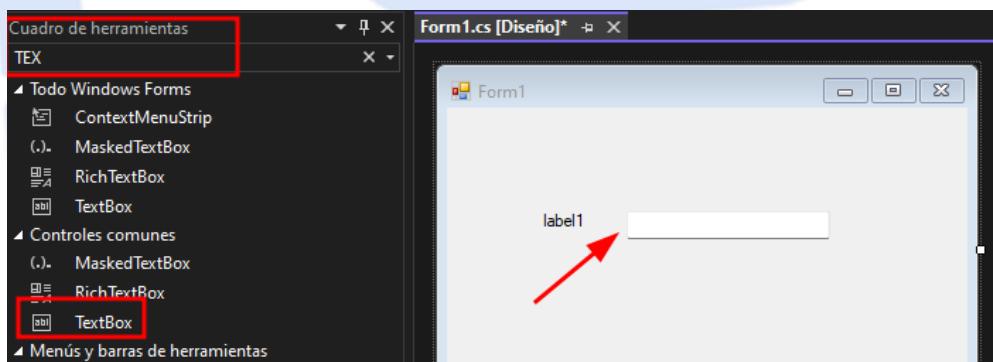
LABEL:

- Un control utilizado para mostrar texto estático en la interfaz del usuario.
- Propiedad clave: Text para definir el contenido del texto.



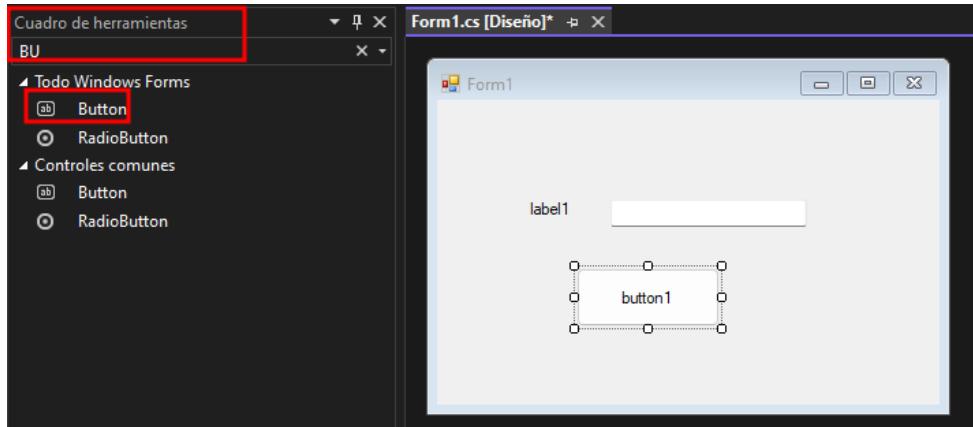
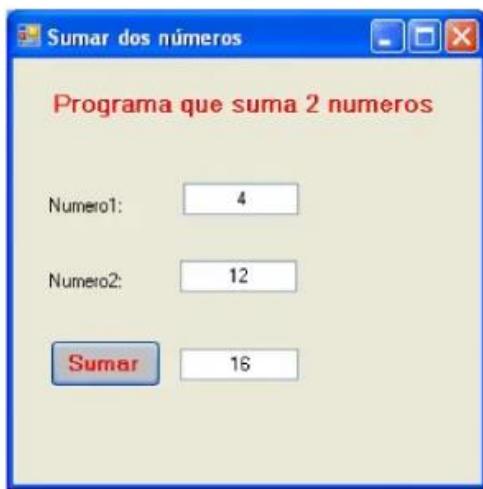
TEXTBOX:

- Utilizado para que el usuario ingrese texto.
- Propiedades clave: Text para obtener o establecer el valor del texto, PasswordChar para usarlo en campos de contraseña.



BUTTON:

- Un control que el usuario puede hacer clic para ejecutar una acción.
- Eventos clave: Click para definir qué pasa cuando el botón es presionado.

**EJEMPLO:****CODIGO:**

```
public class Calculadora
{
    // Método que suma dos números
    public int Sumar(int a, int b)
    {
        return a + b;
    }
}
```

```
using System;
using System.Windows.Forms;

namespace SumaNumerosApp
{
    public partial class Form1 : Form
    {
        private Calculadora calculadora;

        public Form1()
        {
            InitializeComponent();
            // Instanciamos la clase Calculadora
            calculadora = new Calculadora();
        }

        private void btnSumar_Click(object sender, EventArgs e)
        {
            // Convertimos el texto ingresado en los TextBox a enteros
            int numero1 = Convert.ToInt32(txtNumero1.Text);
            int numero2 = Convert.ToInt32(txtNumero2.Text);

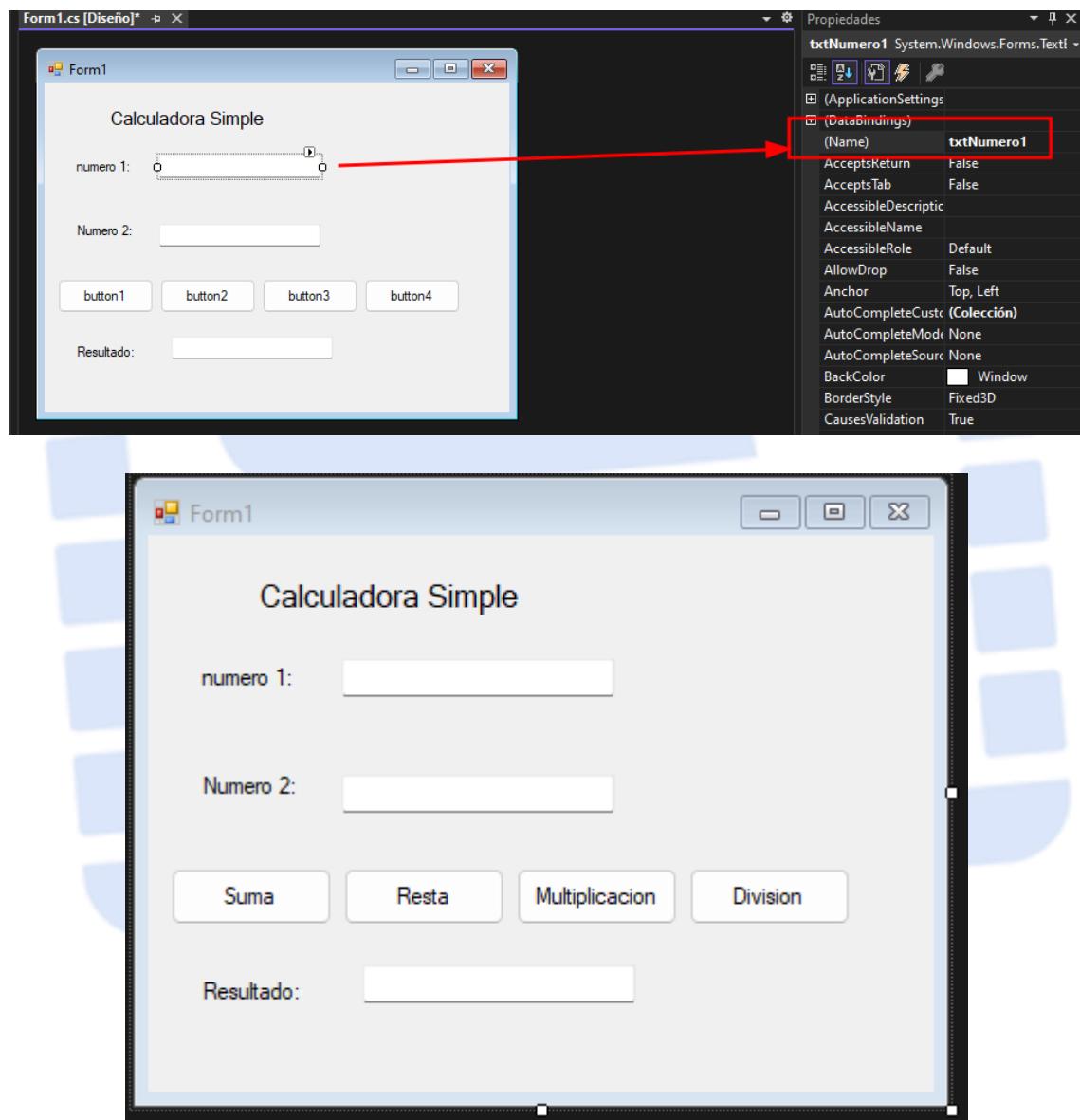
            // Llamamos al método Sumar de la clase Calculadora
            int resultado = calculadora.Sumar(numero1, numero2);

            // Mostramos el resultado en el Label
            lblResultado.Text = "Resultado: " + resultado.ToString();
        }
    }
}
```

EJERCICIO RESUELTO

Calculadora Simple usando Clases

Objetivo: Implementar una calculadora simple que permita sumar, restar, multiplicar y dividir dos números. La lógica de la calculadora se debe manejar mediante una clase separada.



Código para la clase cCalculadora

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

namespace TiendaGolosinas
{
    public class cCalculadora
    {
        public int Sumar(int a, int b) => a + b;
        public int Restar(int a, int b) => a - b;
        public int Multiplicar(int a, int b) => a * b;
        public double Dividir(int a, int b)
        {
            if (b == 0)
                throw new DivideByZeroException("No se puede dividir por cero");
            return (double)a / b;
        }
    }
}
```

Código para el formulario

```
using System;
using System.Windows.Forms;

namespace TiendaGolosinas
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private cCalculadora calculadora = new cCalculadora();

        private void button1_Click(object sender, EventArgs e)
        {
            int num1 = Convert.ToInt32(txtNumero1.Text);
            int num2 = Convert.ToInt32(txtNumero2.Text);
            int resultado = calculadora.Sumar(num1, num2);
            txtResultado.Text = "Resultado: " + resultado;
        }
    }
}
```

Implementar los botones: Resta, Multiplicación y División

EJERCICIOS PROPUESTOS

Ejercicio 1: Validación de Campos Obligatorios

Enunciado:

Desarrolla un formulario que permita al usuario ingresar datos personales (nombre, dirección, correo electrónico). Crea una clase Persona con las propiedades necesarias, y un método ValidarCampos() que verifique que todos los campos obligatorios estén llenos y que el correo tenga un formato válido. El formulario debe mostrar un mensaje de error en un Label si falta algún dato, y debe mostrar "Registro exitoso" si todos los datos son correctos.

Ejercicio 2: Sistema de Notas con Validación de Rangos

Enunciado:

Desarrolla una aplicación que permita a un profesor ingresar las notas de varios estudiantes y calcular el promedio del grupo. La clase Estudiante debe tener una propiedad Nota y un método ValidarNota() que asegure que la nota esté entre 0 y 100. El formulario debe tener un TextBox para ingresar la nota de cada estudiante, un Button para agregar la nota a la lista, y un Label que muestre el promedio del grupo. Si se ingresa una nota fuera del rango, debe mostrarse un mensaje de error en el Label.

Ejercicio 3: Gestión de Inventario con Actualización en Tiempo Real**Enunciado:**

Crea una aplicación de Windows Forms para gestionar un inventario de productos.

Crea una clase Producto con propiedades como Nombre, Precio y Cantidad. La aplicación debe permitir agregar productos a una lista, actualizar la cantidad de un producto existente y mostrar el total del inventario en un Label. Usa un TextBox para ingresar el nombre del producto, y otros TextBox para ingresar el precio y la cantidad.

Al realizar cualquier cambio, el total del inventario debe actualizarse automáticamente.

Ejercicio 4: Simulador de Ventas con Descuentos Dinámicos**Enunciado:**

Desarrolla una aplicación para simular un sistema de ventas. Crea una clase Venta que permita aplicar descuentos según el monto total de la compra. La aplicación debe permitir agregar productos a una lista de compras, mostrar el total de la compra en un Label, y permitir aplicar un descuento en tiempo real según el monto. Si el total es mayor a una cantidad determinada (por ejemplo, \$100), se debe aplicar un descuento automático. Además, implementa la opción de introducir manualmente un código de descuento en un TextBox para aplicar descuentos especiales.



INSTITUTO
KHIPU