

# Introducción a la Programación Interactiva en Visual C#

Curso de Algoritmia y Programación

Luis E. Garreta U.

Universidad Autonoma de Occidente – Cali  
Depto. Operaciones y Sistemas  
Facultad de Ingeniería

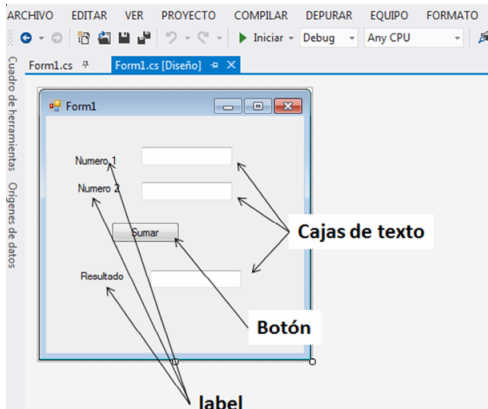
6 de septiembre de 2018

# Introducción a la programación de interfaces gráficas

- ▶ Visual C# es un lenguaje gráfico orientado a eventos, lo que significa que las acciones se ejecutan en respuesta a eventos.
- ▶ Un programa desarrollado en un lenguaje visual, tendrá dos componentes:
  - ▶ la interfaz gráfica (conocida como GUI por sus siglas en inglés – Graphic User Interface),
  - ▶ y el código o instrucciones del programa.

# Elementos de una GUIs simple

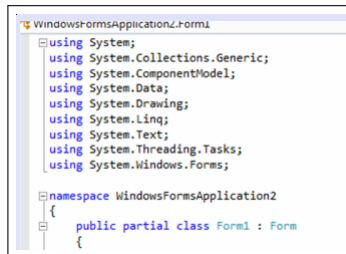
- ▶ Inicialmente se desarrollará la GUI, agregando esencialmente:
  - ▶ labels (etiquetas),
  - ▶ cajas de texto (para leer o mostrar datos) y
  - ▶ botones.



# Partes del Código en una GUI: Encabezados

El código del programa aparece en otra ventana y tiene una estructura similar a la siguiente:

- ▶ **Librerías:** inicialmente se encuentra una sección de varias instrucciones *using*, esta sección le indica al lenguaje cuales librerías<sup>4</sup> se van a emplear en el programa
- ▶ **namespace:** posteriormente aparece una definición del namespace que corresponde a la aplicación (si observa, este nombre es el mismo que usted dio a su programa al crear el proyecto);
- ▶ **Class:** luego continúa la definición de clase, que corresponde a la estructura empleada por el lenguaje y en la cual por ahora no ahondaremos.



```
WindowsFormsApplication2.Form1
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApplication2
{
    public partial class Form1 : Form
    {
```

# Partes del Código en una GUI: Métodos

Lo que sí es importante, es que:

- ▶ Dentro de la clase se ubican diferentes métodos que responden a los eventos de los elementos incorporados en la GUI.
- ▶ En general, serán de la misma forma: tienen el nombre del elemento (button1, label1, etc.) y la acción a la que responden (click).

```
namespace WindowsFormsApplication2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
        }

        private void label1_Click(object sender, EventArgs e)
        {
        }
    }
}
```

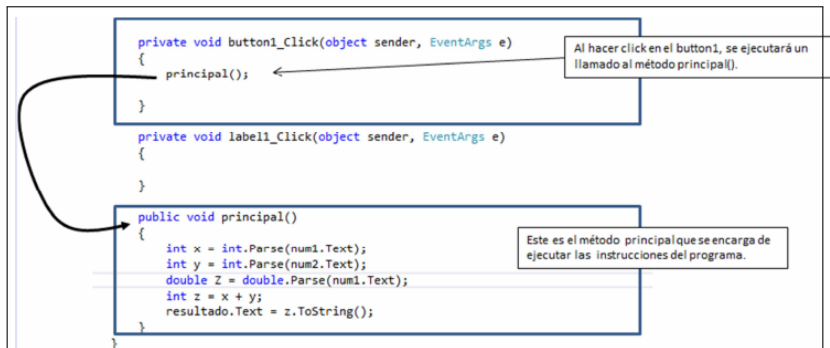
Las instrucciones del método (contenidas entre las llaves), se ejecutarán cuando el usuario de **click** en el boton llamado **button1**.

Las instrucciones del método (contenidas entre las llaves), se ejecutarán cuando el usuario de **click** en el label llamado **label1**.

# Estructura General de un Programa en Este Curso

Como convención en nuestro curso, la estructura general de los programas es:

- ▶ hay un botón que inicia la ejecución del programa (botón de calcular, ejecutar, etc.),
- ▶ y dentro del evento correspondiente se realizará un llamado al método principal(), en el cual se ubicarán las instrucciones necesarias



# Operaciones de Entrada (Teclado/Mouse)

Para escribir o mostrar datos en la pantalla, generalmente usaremos cajas de texto.

- ▶ Cuando se trabaja con cajas de texto, debe tenerse en cuenta que son eso: cajas de texto y por tanto, lo que se ingresa o se muestra en ellas es texto (o cadenas).
- ▶ Para trabajar con los datos que se han ingresado, deben transformarse al tipo requerido, de la siguiente forma:

Suponiendo que la caja de texto que contiene el dato se llama Control:

Lectura de cadenas	<i>String variable = Control.Text;</i> <b>Ej:</b> <i>String nombre = textBoxNombre.Text;</i>
Lectura de enteros	<i>int variable = int.Parse(Control.Text);</i> <b>Ej:</b> <i>int edad = int.Parse( textBoxEdad.Text );</i>
Lectura de reales	<i>double variable = double.Parse(Control.Text);</i> <b>Ej:</b> <i>double estatura = double.Parse( textBoxEstatura.Text);</i>
Lectura de caracteres	<i>char variable = char.Parse(Control.Text);</i> <b>Ej:</b> <i>char seguir = char.Parse( textBoxEstadoCivil.Text );</i>

# Operaciones de Salida (Pantalla)

Para escribir o mostrar datos en la pantalla, generalmente usaremos cajas de texto deshabilitadas (`Enabled=false`), para evitar que el usuario ingrese datos en ellas.

Suponiendo que la caja de texto que va a contener el dato se llama Control:

Mostrar una cadena fija o constante	<b>Control.Text</b> = "Constante de cadena"; Ej: <code>textBoxSaludo.Text = "Hello World";</code>
Mostrar una variable de tipo cadena	<b>Control.Text</b> = <i>variable</i> ; Ej: <code>textBoxNombre.Text = nombre;</code>
Mostrar una variable de tipo diferente a cadena (debe transformarse a texto)	<b>Control.Text</b> = <i>variable</i> .ToString( ); Ej: <code>textBoxNota.Text = nota.ToString( );</code>