
	GUÍA N° 2		
	Ingeniería Electrónica 2011 - III	Programación Aplicada Página 1 de 6	

Guía 2: ESTRUCTURAS DE PROGRAMACIÓN (C#)

I. Entorno Grafico (I.G.U) con C#

La plataforma .Net provee para todos los lenguaje que soporta un entorno gráfico por medio de formularios y controles. Cada formulario se comporta como una clase con atributos y métodos; cada control colocado sobre un formulario se comporta como un atributo del formulario; tanto controles y formularios poseen propiedades, eventos y métodos. Cada uno de estos elementos posee un evento principal por el cual se activa el control.

Ejercicio 1:

1. Cree un proyecto nuevo en Visual Studio .Net; como lenguaje seleccione C# y como plantilla seleccione “*Aplicación de Windows Forms*”.

2. Desde el cuadro de herramientas, seleccione “*Controles Comunes*” y Sobre el formulario coloque los siguientes controles:

- ✓ Una Etiqueta (Label)
- ✓ Un cuadro de texto (Textbox)
- ✓ Un botón (Button)
- ✓ Un cuadro de Chequeo (CheckBox)
- ✓ Un cuadro desplegable (Combobox)
- ✓ Un botón de selección (Radiobutton)
- ✓ Una lista de selección (Listbox).
- ✓ Un Cuadro de Imagen(PictureBox)
- ✓ Un Cuadro de Fecha(DateTimePicker)
- ✓ Un Visualizador WEB(WebBrowser)

3. Manipule alguna de sus propiedades más comunes (size, height, width, backcolor, font, text, location). Analice y observe los resultados.

4. Haga doble clic sobre cada control para acceder al editor de código. Observe que esta acción lo sitúa dentro del evento principal de cada control. Cree su propio listado apuntando cual es el evento principal de cada control y agregue una descripción de cuándo o como se desencadena

II. Introducción al Lenguaje C# (C Sharp)

C#, es un lenguaje .Net FrameWork que procede de C++, por lo que la sintaxis c# es muy similar a la de C/C++. Sus características principales son:

Clases: todo el código y los datos en C# deben ser incluidos en una clase. No se puede definir una variable fuera de la clase y no puede escribir ningún código que no este en una clase. La clase puede tener constructores que se ejecutan cuando se crea el objeto de la clase, y un destructor que se ejecuta cuando un objeto de la clase es destruido. C# permite agrupar sus clases en una colección de clases llamada espacio de nombre.

Tipos de datos: C# permite trabajar con dos tipos de datos: de valor y de referencia. Los de valor contienen valores reales. Los de referencia contienen referencias a valores almacenados en algún lugar de la memoria. Los tipos primitivos como char, int y float, junto con los valores y estructuras comentados, son tipos de valor. Los tipos de referencia tienen variables que tratan con objetos y matrices.

Ejercicio 2.

1. Cree un nuevo proyecto en .Net seleccione el lenguaje C# y en plantillas seleccione “Aplicación de Windows Forms”.
2. Sobre el formulario coloque una etiqueta (label), en la propiedad text escriba “Números”; coloque dos cuadros de texto (textBox); coloque dos botones (button) y en la propiedad text de cada uno escriba “mostrar”; coloque 4 etiquetas vacías
3. Haga doble clic sobre el botón para acceder al editor de código y escriba el siguiente código:

Boton 1	Boton 2
<pre> int n1, n2; int a; float b; double c; decimal d; n1 = int.Parse(textBox1.Text); n2 = int.Parse(textBox2.Text); a = n1 / n2; label2.Text = a.ToString(); b = n1 / n2; label3.Text = b.ToString(); c = n1 / n2; label4.Text = c.ToString(); d = n1 / n2; label5.Text = d.ToString(); </pre>	<pre> int a, n1, n2; float b, n3,n4; double c, n5,n6; decimal d, n7, n8; n1 = int.Parse(textBox1.Text); n2 = int.Parse(textBox2.Text); n3 = float.Parse(textBox1.Text); n4 = float.Parse(textBox2.Text); n5 = double.Parse(textBox1.Text); n6 = double.Parse(textBox2.Text); n7 = decimal.Parse(textBox1.Text); n8 = decimal.Parse(textBox2.Text); a = n1 / n2; label2.Text = a.ToString(); b = n3 / n4; label3.Text = b.ToString(); c = n5 / n6; label4.Text = c.ToString(); d = n7 / n8; label5.Text = d.ToString(); </pre>

4. Ingrese valores numéricos en los textbox, por ejemplo 10 y 9 - 10 y 7 - 10 y 3 ...etc, haga Click en cada botón, observe y Analice los resultados obtenidos.

III. ESTRUCTURAS CONDICIONALES EN C#

3.1 Condicional Simple (If): Cuando se cumple una determinada condición, se ejecuta una o un conjunto de instrucciones específicas. Su estructura es:

```

if (condición)
{
    Instrucción1;
    Instrucción2;
    ...
    Instrucciónn;
}

```

Ejercicio 3.1:

1. Cree un proyecto nuevo en C#, coloque una etiqueta y escriba Digite el numero; coloque una caja de texto vacía; coloque un botón y escriba calcular.
2. Haga doble clic sobre el botón para acceder al editor código y escriba el siguiente código:

```
int n, r;
n = int.Parse(textBox1.Text);
r = n % 2;
if (r != 0)
{
    MessageBox.Show("El numero no es par");
}
```



3.2 Condicionales Compuestos o Anidados (if – else)- (If – else if – else): se utilizan para evaluar diferentes condiciones lógicas en una misma estructura de toma de decisiones. Su estructura es:

If – else	If – else if - else
<pre>if(condición) { Instrucción1; Instrucción2; ... } else { Instrucción1; Instrucción2; ... }</pre>	<pre>if (condición 1) { Instrucción1; Instrucción2; ... } else if (condición 2) { Instrucción1; Instrucción2; ... } else { Instrucción1; Instrucción2; ... }</pre>

Ejercicio 3.2:

1. Utilice el proyecto anterior para el ejercicio.
2. Modifique el código del botón y escriba el siguiente código:

```
int n, r;
n = int.Parse(textBox1.Text);
r = n % 2;
if (r != 0){
    MessageBox.Show("El numero no es par");}
Else {
    MessageBox.Show("el numero es par");}
```

	GUÍA Nº 2		
	Ingeniería Electrónica 2011 - III	Programación Aplicada Página 4 de 6	

3.3 La instrucción switch: se utiliza cuando se requiere en misma estructura evaluar varias condiciones teniendo como base un valor o variable de referencia. Su estructura es:

```

switch (variable_a_evaluar)
{
    Case valor_1:
    {
        Instrucción1;
        Instrucción2;
        .....
    }
    Break;
    Case valor_2:
    {
        Instrucción1;
        Instrucción2;
        .....
    }
    Break;
    Case valor_n:
    {
        Instrucción1;
        Instrucción2;
        .....
    }
    Break;
}

```

Ejercicio 3.3:

1. Cree un proyecto nuevo en C#, coloque una etiqueta vacía; coloque una caja desplegable (combobox).
2. Haga doble clic sobre el formulario para acceder al editor de código al evento Load del formulario (Form1_Load...) y escriba el siguiente código:

```

comboBox1.Items.Add ("Seleccione");
comboBox1.Items.Add ("Azul");
comboBox1.Items.Add ("Amarillo");
comboBox1.Items.Add ("Verde");
comboBox1.Items.Add ("Rojo");



```

3. haga doble clic sobre el combo para acceder al editor de código y escriba el siguiente código:

```

switch (comboBox1.Text)
{
    case "Verde":
        label1.BackColor = System.Drawing.Color.Green;
        break;
    case "Azul":
        label1.BackColor = System.Drawing.Color.Blue;
        break;
    case "Amarillo":
        label1.BackColor = System.Drawing.Color.Yellow;
        break;
}

```

	GUÍA Nº 2		
	Ingeniería Electrónica 2011 - III	Programación Aplicada Página 5 de 6	

```

    case "Rojo":
        labell.BackColor = System.Drawing.Color.Red;
        break;
    default:
        MessageBox.Show("Por favor, Seleccione un Color");
        break;
}

```

Nota: si la variable a evaluar es de valor numérico los valores en el case no se deben colocar entre comillas.

IV. ESTRUCTURAS REPETITIVAS EN C#

Se usan cuando se requiere que un procedimiento se realice un número determinado de veces y se termine cuando se cumpla una condición específica.

4.1 La instrucción for(): Su estructura es:

```

for (variable_inicio; condición_de_fin; incremento)
{
    Instrucción1;
    Instrucción2;
    ...
}

```

4.2 La instrucción While(): su estructuras es

```

While (condición)
{
    Instrucción1;
    Instrucción2;
    Incremento;
}

```

4.3 La instrucción do – while(): su estructura es:

```

Do
{
    Instrucción1;
    Instrucción2;
    Incremento;
}
While (condición);

```

Ejercicio 4.



1. Cree un proyecto nuevo en C#. Sobre el formulario coloque una etiqueta que diga "Número a calcular su factorial"; una caja de texto vacía; tres botones que digan respectivamente: for, while, do_while; y finalmente una etiqueta(label) vacía.

2. Haga doble clic sobre el botón que dice for y escriba el siguiente código:

```

int n, f=1;
n = int.Parse(textBox1.Text);
for(int i=1;i<=n;i++)
    f*=i;
labell.Text = f.ToString();

```

	GUÍA Nº 2		
	Ingeniería Electrónica 2011 - III	Programación Aplicada Página 6 de 6	

3. Haga doble clic sobre el botón que dice while y escriba el siguiente código:

```
int n, f=1, i=1;
n = int.Parse(textBox1.Text);
while (i <= n)
{
    f *= i;
    i++;
}
label1.Text = f.ToString();
```

4. Haga doble clic sobre el botón que dice do_while y escriba el siguiente código:

```
int n, f=1, i=1;
n = int.Parse(textBox1.Text);
do
{
    f *= i;
    i++;
}
while (i <= n);
label1.Text = f.ToString();
```

TALLER:

1. Hacer el listado de los eventos principales de los controles de I.G.U .Net más importantes y descripción de cuándo o como se desencadena
2. Investigue cuales los principales tipos de datos permitidos en el lenguaje C# e incluya una breve descripción de cada uno y su correspondiente tamaño o espacio en memoria.
3. Realice una aplicación en entorno grafico tipo Windows en la cual satisfaga los siguientes requerimientos:
 - ✓ Permita ingresar un numero por medio del uso un cuadro de texto (TextBox)
 - ✓ Presente un "ComboBox" con las siguientes opciones: Seno – Coseno – Raíz Cuadrada – Serie de Fibonacci - Factorial – Potencia
 - ✓ Cuando el usuario seleccione una de las opciones y presione clic sobre un determinado botón, se debe validar que efectivamente se ha ingresado un número y luego mostrar en alguna etiqueta (label) el resultado correcto de acuerdo a la opción seleccionada.
4. Realice una aplicación en entorno grafico tipo Windows que calcule el valor de una resistencia a partir de su combinación de colores. Agregar un convertidor de unidades (MegaOhm – KiloOhm – Ohm... etc) que se pueda utilizar al momento de tener calculado el valor de la resistencia. Debe mostrar una imagen de la resistencia con los colores seleccionados antes de efectuar el cálculo correspondiente.