



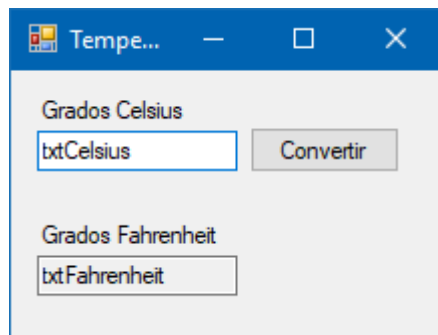
## PRÁCTICA 4: INTRODUCCIÓN A LAS FUNCIONES

### Objetivos:

- ✓ Programar funciones propias o personalizadas
- ✓ Crear funciones con listas de argumentos de longitud variable
- ✓ Aplicar la recursividad en la programación de funciones
- ✓ Aplicar la sobrecarga de funciones

1. Ejecutar Visual Studio .NET
2. Crear un nuevo proyecto de tipo Aplicación de Windows Forms (Windows Form App)
3. Crear un formulario para cada uno de los siguientes ejemplos.

**Ejemplo 1:** función personalizada con un parámetro (celsius), para convertir grados Celsius a Fahrenheit



1 reference

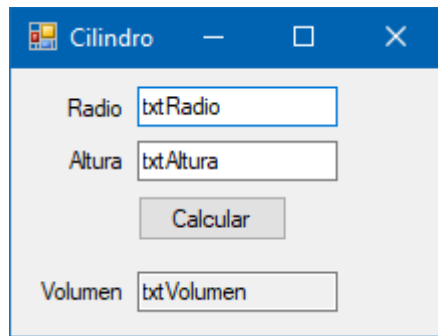
```
public static decimal CelAfah(decimal cel)
{
    decimal fah;
    fah = cel * 9 / 5 + 32;
    return fah;
}
```

1 reference

```
private void btnConvertir_Click(object sender, EventArgs e)
{
    decimal celsius = Convert.ToDecimal(txtCelsius.Text);
    decimal fahrenheit = CelAfah(celsius);
    txtFahrenheit.Text = String.Format("{0:N2}", fahrenheit);
}
```

**Ejemplo 2:** función personalizada con dos parámetros (radio y altura), para calcular el volumen de un cilindro

---

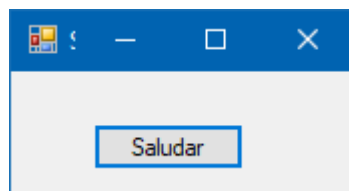


```
1 reference
public static double VolumenCilindro(double r, double h)
{
    double vol = Math.PI * Math.Pow(r, 2) * h;
    return vol;
}

1 reference
private void btnCalcular_Click(object sender, EventArgs e)
{
    double radio = Convert.ToDouble(txtRadio.Text);
    double altura = Convert.ToDouble(txtAltura.Text);
    double volumen = VolumenCilindro(radio, altura);
    txtVolumen.Text = String.Format("{0:N1}", volumen);
}
```

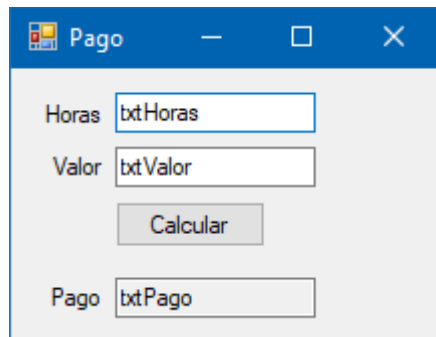
**Ejemplo 3:** ejemplo de función sin valor de retorno

---



```
1 reference
public static void Saludar()
{
    DateTime hoy = DateTime.Today;
    MessageBox.Show("Hola hoy es " + hoy.ToString("D"));
}

1 reference
private void btnSaludar_Click(object sender, EventArgs e)
{
    Saludar();
}
```

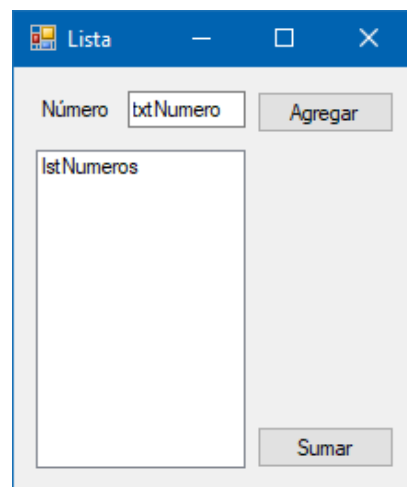


2 references

```
public static double CalcularPago(int h, double val = 7.5)
{
    return h * val - h * val * 0.1;
}
```

1 reference

```
private void btnCalcular_Click(object sender, EventArgs e)
{
    int horas = Convert.ToInt32(txtHoras.Text);
    double valor, pago;
    if (double.TryParse(txtValor.Text, out valor))
    {
        pago = CalcularPago(horas, valor);
    }
    else
    {
        pago = CalcularPago(horas);
    }
    txtPago.Text = String.Format("{0:C1}", pago);
}
```



1 reference

```
private void btnAgregar_Click(object sender, EventArgs e)
{
    lstNumeros.Items.Add(txtNumero.Text);
}
```

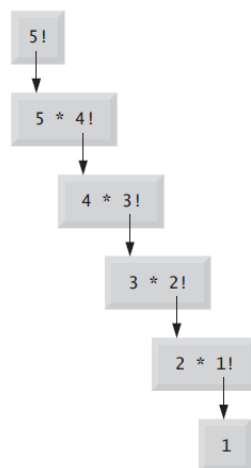
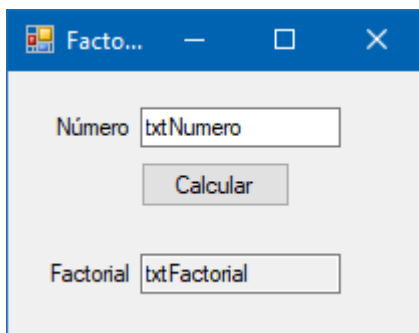
1 reference

```
public static int Sumar(params String[] numeros)
{
    int suma = 0;
    foreach (string numero in numeros)
    {
        suma += Convert.ToInt32(numero);
    }
    return suma;
}
```

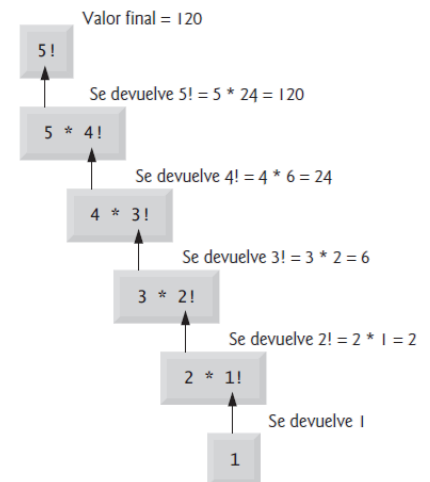
1 reference

```
private void btnSumar_Click(object sender, EventArgs e)
{
    String[] nums = new String[lstNumeros.Items.Count];
    lstNumeros.Items.CopyTo(nums, 0);
    MessageBox.Show("Suma = " + Sumar(nums));
}
```

#### Ejemplo 6: ejemplo de función recursiva, para calcular el factorial de un número



(a) Secuencia de llamadas recursivas.



(b) Valores devueltos de cada llamada recursiva.

2 references

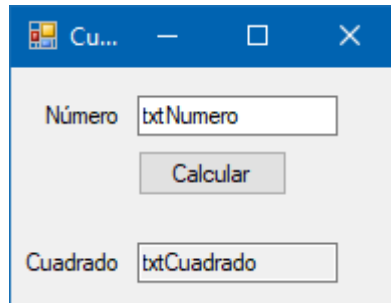
```
public static int Factorial(int n)
{
    if (n <= 1)
    {
        return 1;
    }
    else
    {
        return n * Factorial(n - 1);
    }
}
```

1 reference

```
private void btnCalcular_Click(object sender, EventArgs e)
{
    int num = Convert.ToInt32(txtNumero.Text);
    int facto = Factorial(num);
    txtFactorial.Text = String.Format("{0}", facto);
}
```

**Ejemplo 7:** ejemplo función calcularCuadrado sobrecargada (con 3 versiones)

---



0 references

```
public static int CalcularCuadrado(int num)
{
    return num * num;
}
```

1 reference

```
public static double CalcularCuadrado(double num)
{
    return num * num;
}
```

0 references

```
public static decimal CalcularCuadrado(decimal num)
{
    return num * num;
}
```

1 reference

```
private void btnCalcular_Click(object sender, EventArgs e)
{
    double numero = Convert.ToDouble(txtNumero.Text);
    double cuadrado = CalcularCuadrado(numero);
    txtCuadrado.Text = String.Format("{0}", cuadrado);
}
```



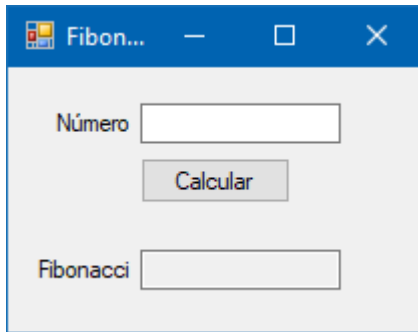
Colocar en el repositorio (GitHub) del portafolio de la **práctica4** la solución de los siguientes ejercicios.

**Indicaciones:** Agregar a la solución un proyecto de Visual C# de tipo Aplicación Windows Forms y programar los siguientes formularios.

1. Desarrollar una aplicación que capture 4 números en diferentes cuadros de texto, crear una función que reciba los números y retorne la sumatoria de dichos números para ser mostrada en una etiqueta.

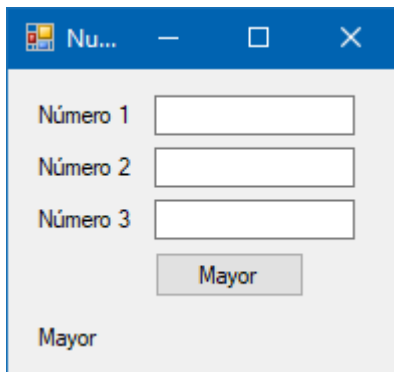
2. Desarrollar un programa que almacene en un ListBox cualquier cantidad de números ingresados por un cuadro de texto, que posea una función para buscar un determinado número y que indique cuantas veces aparece dicho número en la lista.

3. Escribir una aplicación que posea una función recursiva para calcular el Fibonacci de un número determinado capturado en un cuadro de texto.



The screenshot shows a standard Windows application window with a blue title bar containing the text 'Fibon...'. The main area of the window is light gray and contains three elements: a text input field with the label 'Número' to its left, a gray button with the text 'Calcular' below the first input field, and another text input field with the label 'Fibonacci' to its left.

4. Elaborar un programa que posea la función mayor con 2 versiones, la primera para devolver el mayor de dos números enteros y la segunda para devolver el mayor de tres números enteros. Demostrar su funcionamiento.



The screenshot shows a standard Windows application window with a blue title bar containing the text 'Nu...'. The main area of the window is light gray and contains five elements: three text input fields stacked vertically with labels 'Número 1', 'Número 2', and 'Número 3' to their left; a gray button with the text 'Mayor' below the third input field; and a label 'Mayor' at the bottom left of the window.