

ILERNA

Online

Videotutoría 5: aprendiendo a programar(III)

Módulo 03A: Programación



IDEs: Entornos de desarrollo de programas

- **Características:**
 - Son multiplataforma.
 - Actúan como soporte para diferentes lenguajes de programación.
 - Reconocen sintaxis.
 - Están integrados con sistemas de control de diferentes versiones.
 - Tienen un depurador.
 - Permiten importar y exportar proyectos.
 - Manejan diferentes idiomas.
 - Facilitan un manual de ayuda para el usuario.

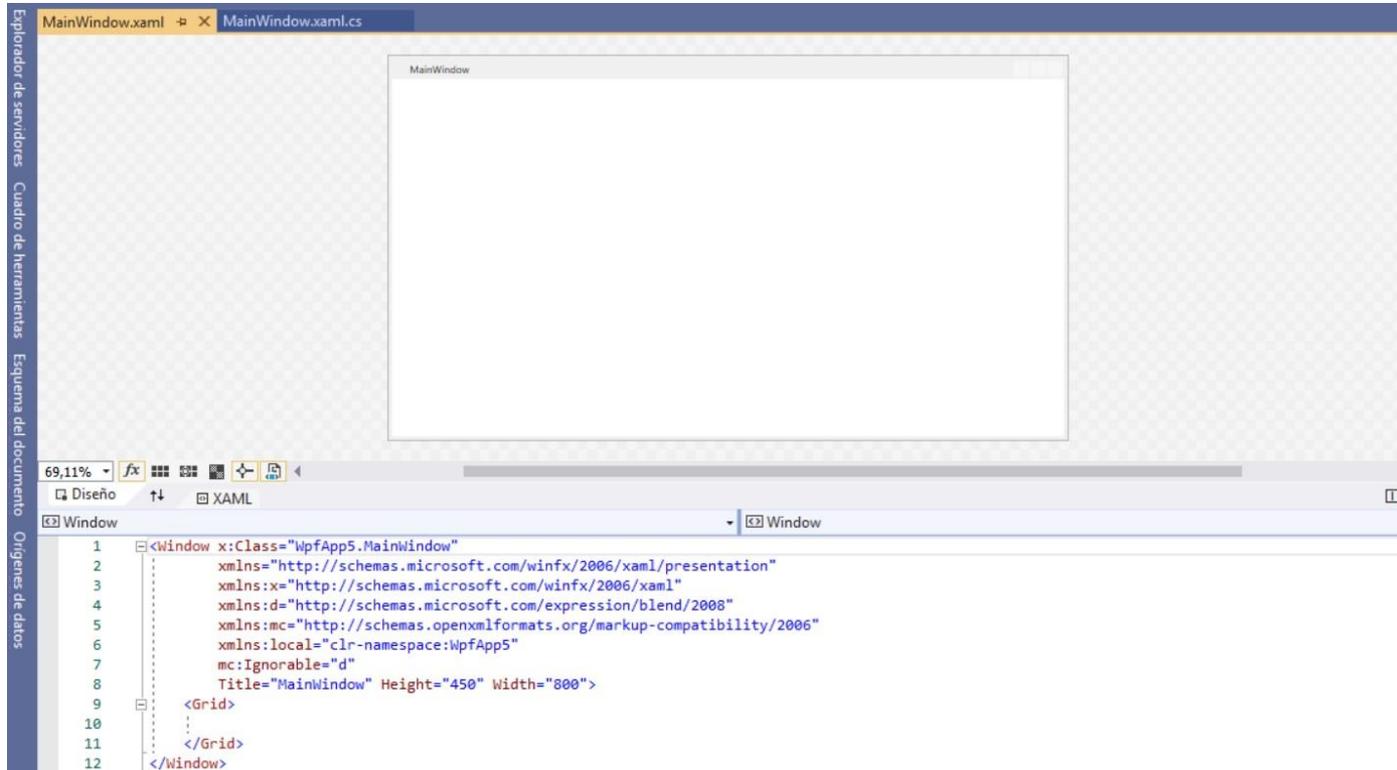
IDEs: XAML

- Interfaz de usuario básica para un programa:
- Mediante las herramientas de XAML en Visual Studio, como el Diseñador XAML, Caja de herramientas, el editor XAML, el panel Propiedades y Esquema del documento, para agregar controles y contenido a la interfaz de usuario.
- Mediante algunos de los paneles de diseño de XAML más comunes, como RelativePanel, Grid y StackPanel.

XAML:

```
<TextBlock x:Name="textBlock"  
    HorizontalAlignment="Left"  
    Margin="351,44,0,0"  
    TextWrapping="Wrap"  
    Text="TextBlock"  
    VerticalAlignment="Top"/>
```

IDEs: XAML



Ejercicios: For

- Realiza un programa que nos muestre la tabla de multiplicar desde 1 hasta n (número introducido por el usuario)

Ejercicios: For

```
static void Main(string[] args) {
    int j,i,n;

    Console.WriteLine("\n\n");

    Console.WriteLine("\n\n");

    Console.WriteLine("Introduce los números desde 1 : ");
    n= Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("Multiplicamos de 1 a {0} \n",n);
    for(i=1;i<=10;i++)
    {
        for(j=1;j<=n;j++)
        {
            if (j<=n-1)
                Console.WriteLine("{0}x{1} = {2}, ",j,i,i*j);
            else
                Console.WriteLine("{0}x{1} = {2}",j,i,i*j);

        }
        Console.WriteLine("\n");
    }
}
```

Ejercicios: Matrices

- Realiza un programa que el usuario introduzca los valores de una matriz de 3x3 y los muestre por pantalla

Ejercicios: Matrices

```
static void Main(string[] args)
{
    int i,j;
    int[,] arr1 = new int[3,3];

    Console.WriteLine("Introduce los elementos de la matriz :\n");
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            Console.WriteLine("elemento - [{0},{1}] : ",i,j);
            arr1[i,j] = Convert.ToInt32(Console.ReadLine());
        }
    }

    Console.WriteLine("\nLa matriz es : \n");
    for(i=0;i<3;i++)
    {
        Console.WriteLine("\n");
        for(j=0;j<3;j++)
            Console.WriteLine("{0}\t",arr1[i,j]);
    }
    Console.WriteLine("\n\n");
}
```

Ejercicios: Matrices

```
1  using System;
2  public class Exercisel2
3  {
4      public static void Main()
5      {
6
7          //ESCRIBIR UN TRIÁNGULO
8          int i,j,rows,k=1;
9
10         Console.Write("\n\n");
11         Console.Write("\n\n");
12
13         Console.Write("Mete el número de filas : ");
14         rows= Convert.ToInt32(Console.ReadLine());
15         for(i=1;i<=rows;i++)
16         {
17             for(j=1;j<=i;j++)
18                 Console.Write("{0} ",k++);
19             Console.Write("\n");
20         }
21     }
22 }
```

Ejercicios: enum

- Realiza un programa que nos muestre los días de la semana de forma numérica

Ejercicios: enum

```
7 namespace ConsoleApp16
8 {
9     0 referencias
10    class Program
11    {
12        2 referencias
13        enum Day { Sun, Mon, Tue, Wed, Thu, Fri, Sat };
14        0 referencias
15        static void Main(string[] args)
16        {
17
18            int x = (int)Day.Sun;
19            int y = (int)Day.Fri;
20
21            Console.WriteLine("{0}", x);
22            Console.WriteLine("{0}", y);
23
24            Console.ReadKey();
25        }
26    }
27 }
28 }
```

Ejercicios: String

```
static void Main (string[] args) {  
  
    string cadena="Hola";  
    for (int i=0;i<cadena.Length;i++)  
        Console.WriteLine("{0}", cadena[i]);  
  
    Console.ReadKey();  
  
}
```

Ejercicios: Foreach

Para matrices unidimensionales, la instrucción foreach procesa los elementos en orden creciente de índice, comenzando con el índice 0 y terminando con el índice Length - 1:

```
int[] numbers = { 4, 5, 6, 1, 2, 3, -2, -1, 0 };  
foreach (int i in numbers)  
{  
    System.Console.WriteLine("{0} ", i);  
}
```

Ejercicios: Listas

```
namespace ConsoleApp33
{
    Oreferencias
    class Program
    {
        Oreferencias
        static void Main(string[] args)
        {
            var fibNumbers = new List<int> { 0, 1, 1, 2, 3, 5, 8, 13 };
            int count = 0;
            foreach (int element in fibNumbers)
            {
                count++;
                Console.WriteLine($"Element #{count}: {element}");
            }
            Console.WriteLine($"Number of elements: {count}");
            Console.ReadKey();
        }
    }
}
```

Ejercicios a realizar

1.- Realice un programa que lea de una lista de valores introducida por teclado. A continuación, y sobre la lista, tiene que buscar los valores máximo y mínimo y escribirlos.

2.- Escribir un programa que lea una cadena de caracteres y a continuación visualice el símbolo y el valor Unicode de cada uno de los caracteres de la cadena.

Ej: Carácter 'H', código Unicode = 72

3.- Escribir un programa que lea una cadena de caracteres de la entrada estándar, la almacene en un objeto `StringBuilder` y utilizando un método "MinusculasMayusculas", convierta todos los caracteres de la cadena a mayúsculas (función parecida al método `ToUpper`)

Recibirá el método una cadena de texto a convertir. Esta cadena estará encapsulada en un objeto `StringBuilder` para poder modificarla

Ej: `void MinusculasMayusculas (StringBuilder str) { // cuerpo del método }`

4.- Utilizando el método `NextDouble` de la clase `Random` del espacio de nombres `System`, realizar un programa que muestre seis números aleatorios diferentes entre 1 y 49 ordenados ascendentemente.

5.- Realice un programa que partiendo de dos matrices de cadenas de caracteres ordenadas ascendentemente pedidas por teclado, construya y visualice una tercera matriz también ordenada ascendentemente.

La idea que se persigue es construir la tercera lista ordenada (no es correcto construirla y después ordenarla con el método `Sort`)

