

#4 – Operadores

En cualquier lenguaje de programación los operadores nos permiten trabajar con los datos guardados en las variables del programa, así por ejemplo podremos realizar entre esos datos operaciones matemáticas, relacionales y lógicas. Para cada una de las operaciones citadas existen una variedad de operadores. Los primeros que estudiaremos por ser quizá con los primeros que interactuamos en nuestra niñez serán los operadores aritméticos o matemáticos.

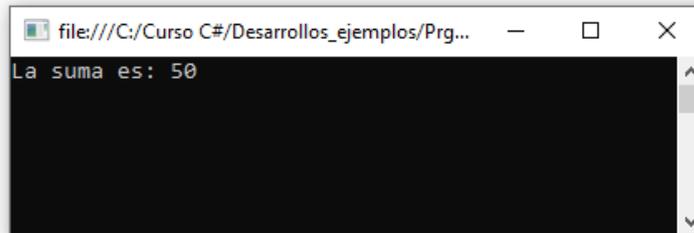
Operadores aritméticos de C#		
Operador	Operación que realiza	Expresión en C#
+	Suma	$x + y$
-	Resta	$x - y$
*	Multiplicación	$x * y$
/	División	x / y
%	Módulo (resto de una división)	$x \% y$
Operadores unarios		
Operador	Operación que realiza	Expresión en C#
++	Incremento	$x++$ equivalente a $(x = x + 1)$
--	Decremento	$x--$ equivalente a $(x = x - 1)$

A continuación veremos cómo implementar el uso de los operadores aritméticos, comenzaremos con la suma:

```
static void Main(string[] args)
{
    //Operador aritmético suma
    int a = 30;
    int b = 20;
    int c = a + b; //Aquí se implementa el operador
    Console.WriteLine("La suma es: " + c); //El signo + es de concatenación
    Console.ReadKey();
}
```

```
}
```

Salida por pantalla al ejecutar el código:



```
file:///C:/Curso C#/Desarrollos_ejemplos/Prg... - □ ×
La suma es: 50
```

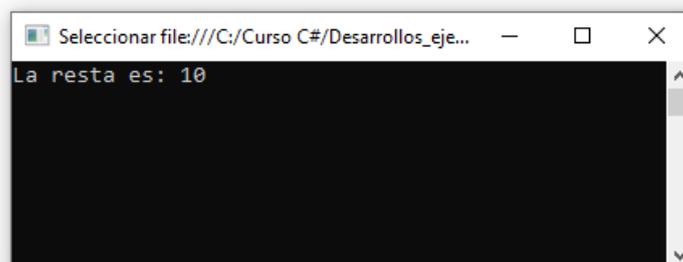
Operador aritmético resta:

```
static void Main(string[] args)
{
    //Operador aritmético resta
    int a = 30;
    int b = 20;
    int c = a - b; //Aquí se implementa el operador
    Console.WriteLine("La resta es: " + c); //El signo + es de concatenación
    Console.ReadKey();
}
```

Salida por pantalla al ejecutar el código, observar que en la línea

```
Console.WriteLine("La resta es: " + c);
```

El signo (+) antes de la variable c realiza la concatenación en la salida, es decir se muestra por pantalla la leyenda "La resta es: " y a continuación el contenido de c que para el caso es 10.



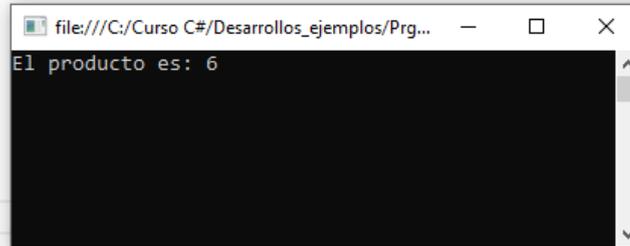
```
Seleccionar file:///C:/Curso C#/Desarrollos_eje... - □ ×
La resta es: 10
```

Operador aritmético de multiplicación:

```
static void Main(string[] args)
{
    //Operador aritmético multiplicación
    int a = 3;
    int b = 2;
    int c = a * b; //Aquí se implementa el operador
    Console.WriteLine("La resta es: " + c);
    Console.ReadKey();
}
```

}

Salida por pantalla:

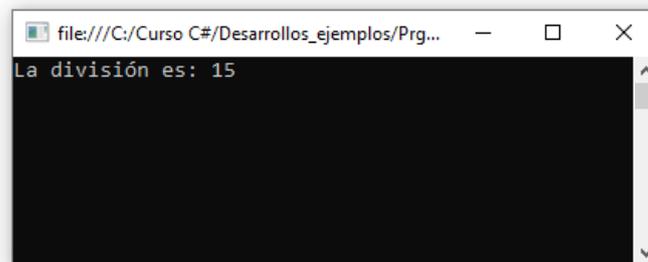


```
file:///C:/Curso C#/Desarrollos_ejemplos/Prg...
El producto es: 6
```

Operador aritmético de división:

```
static void Main(string[] args)
{
    //Operador aritmético división
    int a = 30;
    int b = 2;
    int c = a / b; //Aquí se implementa el operador
    Console.WriteLine("La división es: " + c);
    Console.ReadKey();
}
```

Salida por pantalla:

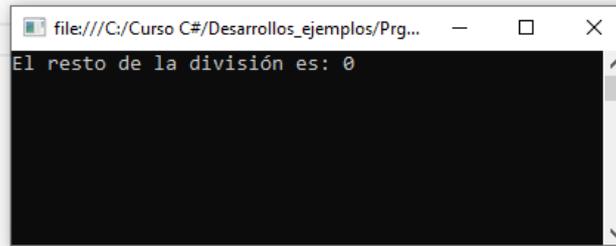


```
file:///C:/Curso C#/Desarrollos_ejemplos/Prg...
La división es: 15
```

Operador aritmético módulo:

```
static void Main(string[] args)
{
    //Operador aritmético módulo
    int a = 30;
    int b = 2;
    int c = a % b; //Aquí se implementa el operador
    Console.WriteLine("El resto de la división es: " + c);
    Console.ReadKey();
}
```

Salida por pantalla:

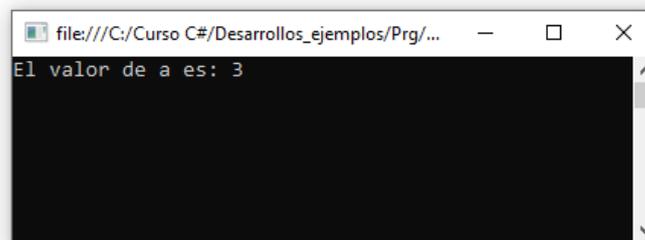


```
file:///C:/Curso C#/Desarrollos_ejemplos/Prg...
El resto de la división es: 0
```

El operador unario ++

```
static void Main(string[] args)
{
    //Operador unario ++
    int a = 1;
    a++; //la variable a vale 1 y se la incrementa en 1
    a++; //la variable a vale 2 y se la incrementa en 1
    Console.WriteLine("El valor de a es: " + a); //a vale 3
    Console.ReadKey();
}
```

Salida por pantalla:

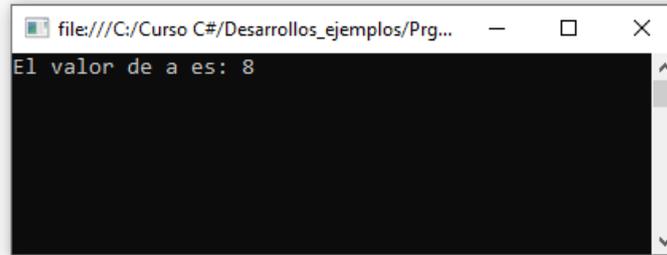


```
file:///C:/Curso C#/Desarrollos_ejemplos/Prg/...
El valor de a es: 3
```

Operador decremento --

```
static void Main(string[] args)
{
    //Operador unario --
    int a = 10;
    a--; //la variable a vale 10 y se la decrementa en 1
    a--; //la variable a vale 9 y se la decrementa en 1
    Console.WriteLine("El valor de a es: " + a); //a vale 8
    Console.ReadKey();
}
```

Salida por pantalla:



```
file:///C:/Curso C#/Desarrollos_ejemplos/Prg...
El valor de a es: 8
```

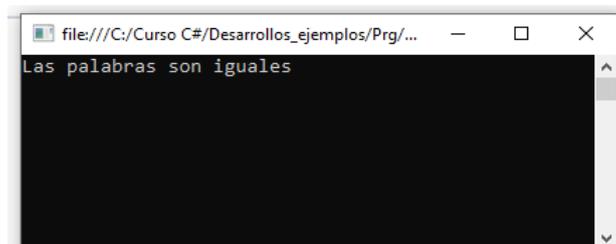
A continuación trataremos los operadores relacionales

Operadores relacionales de C#		
Operador	Operación que realiza en x e y	Expresión en C#
==	x es igual que y	x == y
!=	x es distinto que y	x != y
<	x es menor que y	x < y
>	x es mayor que y	x > y
<=	x es menor o igual que y	x <= y
>=	x es mayor o igual que y	x >= y

El operador relacional de igualdad ==

```
static void Main(string[] args)
{
    //Operador relacional igual que
    string cad1 = "texto";//la variable cad1 vale "texto"
    string cad2 = "texto";//la variable cad2 vale "texto"
    if (cad1 == cad2)
    {
        Console.WriteLine("Las palabras son iguales");//salida por pantalla
    }
    Console.ReadKey();
}
```

Salida por pantalla:



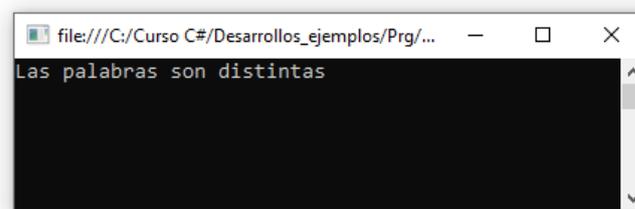
```
file:///C:/Curso C#/Desarrollos_ejemplos/Prg/...
Las palabras son iguales
```

El operador relacional es distinto que:

```
static void Main(string[] args)
{
    //Operador relacional distinto que
    string cad1 = "texto";//la variable cad1 vale "texto"
    string cad2 = "hola";//la variable cad2 vale "hola"

    if (cad1 != cad2)
    {
        Console.WriteLine("Las palabras son distintas");//salida por pantalla
    }
    Console.ReadKey();
}
```

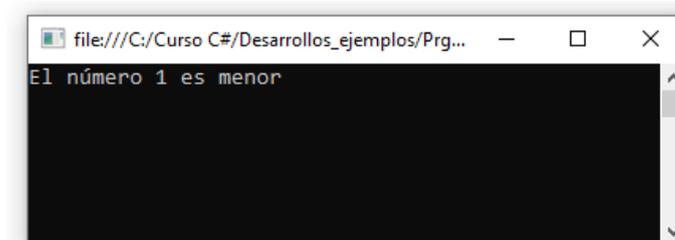
Salida por pantalla:



Uso del operador relacional menor que

```
static void Main(string[] args)
{
    //Operador relacional es menor que
    int numero1 = 20;
    int numero2 = 22;
    if (numero1 < numero2)
    {
        Console.WriteLine("El número 1 es menor");//salida por pantalla
    }
    Console.ReadKey();
}
```

Salida por pantalla:

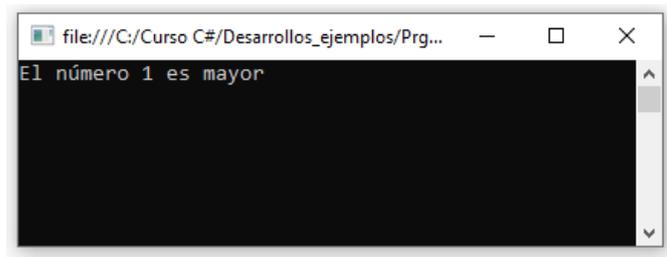


Uso del operador relacional es mayor que:

```
static void Main(string[] args)
{
    //Operador relacional es mayor que
```

```
int numero1 = 40;
int numero2 = 22;
if (numero1 > numero2)
{
    Console.WriteLine("El número 1 es mayor");//salida por pantalla
}
Console.ReadKey();
}
```

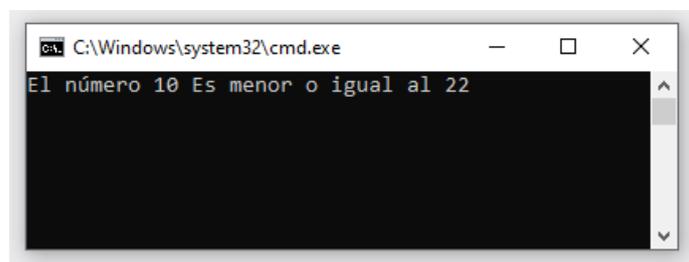
Salida por pantalla:



Uso del operador relacional es menor o igual que

```
static void Main(string[] args)
{
    //Operador relacional es menor o igual que
    int numero1 = 10;
    int numero2 = 22;
    if (numero1 <= numero2)
    {
        Console.WriteLine("El número " + numero1 + " Es menor o igual al " + numero2);
    }
    Console.ReadKey();
}
```

Salida por pantalla:

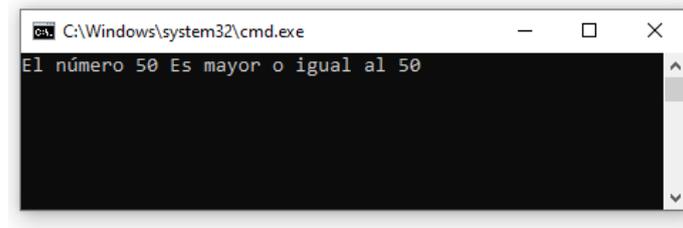


Uso del operador relacional mayor o igual que

```
static void Main(string[] args)
{
    //Operador relacional es mayor o igual que
    int numero1 = 50;
    int numero2 = 50;
    if (numero1 >= numero2)
    {
```

```
Console.WriteLine("El número " + numero1 + " Es mayor o igual al " + numero2);  
}  
Console.ReadKey();  
}
```

Salida por pantalla



Operadores lógicos a nivel de byte

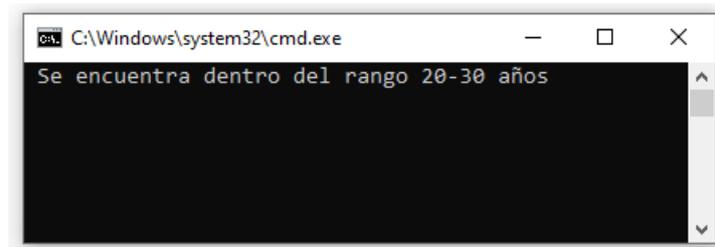
Operadores lógicos de C#		
Operador	Operación que realiza en x e y	Expresión en C#
&&	y lógico (and)	x && y
	o lógico (or)	x y
!	Negación (not)	!x

Uso del operador lógico && - (y)

El siguiente código permite determinar si una persona tiene entre 20 y 30 años inclusive:

```
static void Main(string[] args)  
{  
    //Operador lógico y  
    int edad = 26;  
    if (edad >= 20 && edad <= 30)  
    {  
        Console.WriteLine(" Se encuentra dentro del rango 20-30 años");  
    }  
    else  
    {  
        Console.WriteLine(" Se encuentra fuera del rango de edad");  
    }  
    Console.ReadKey();  
}
```

Salida por pantalla:



```
C:\Windows\system32\cmd.exe
Se encuentra dentro del rango 20-30 años
```

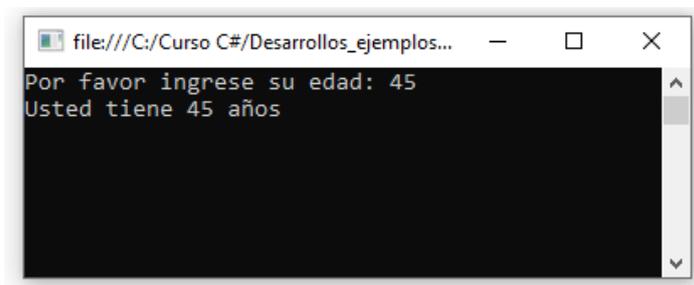
Uso del operador lógico || - (o)

El siguiente programa le indica al usuario que debe ingresar su edad y la verifica (valida) entre 0 y 120 años. Si el usuario ingresa una edad no válida el programa se lo informa mediante una leyenda, de lo contrario, si la edad es válida la muestra por pantalla.

```
static void Main(string[] args)
{
    //Operador lógico o
    int edad;
    Console.WriteLine("Por favor ingrese su edad: ");
    edad = Convert.ToInt32(Console.ReadLine());
    if (edad < 0 || edad > 120)
    {
        Console.WriteLine("Ingreso una edad no válida...");
    }
    else
    {
        Console.WriteLine("Usted tiene " + edad + " años");
    }

    Console.ReadKey();
}
```

Salida por pantalla:



```
file:///C:/Curso C#/Desarrollos_ejemplos...
Por favor ingrese su edad: 45
Usted tiene 45 años
```

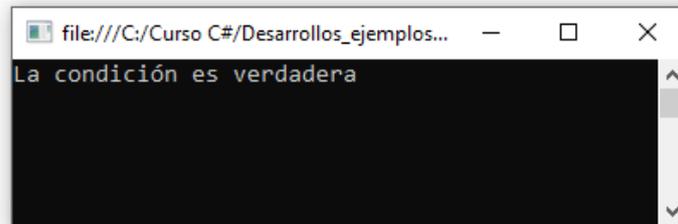
Uso del operador lógico ! – no

```
static void Main(string[] args)
{
    //Operador lógico not
    bool valor = false;

    if (!valor) //se evalua la condición negada, es decir no false
    {
```

```
        Console.WriteLine("La condición es verdadera");  
    }  
  
    Console.ReadKey();  
}
```

Salida por pantalla:



Queda agregar para los operadores lógicos aquellos que evalúan a nivel de bits, a ellos los dejaremos para tratarlos especialmente en otro pequeño apunte.