



Lenguaje básico C#

▼ Hola mundo

```
using System;

namespace Pruebas {
    class HolaMundo {
        public static void Main(string[] args) {
            Console.WriteLine("Hello World!");
        }
    }
}
```

▼ Tipos de datos

```
public static void Main(string[] args) {
    // Tipos de datos
    bool logico = false;
    char character = 'c';
    int num = 5;
    long largo = 22;
    float dec = 12.5f;
    double doble = 22.222;
    string cadena = "hola";
}
```

▼ Operadores aritméticos

Operador	Significado	Ejemplo
<code>+</code>	Suma	<code>a + b</code>
<code>-</code>	Resta	<code>a - b</code>
<code>*</code>	Multipliación	<code>a * b</code>
<code>/</code>	División	<code>a / b</code>
<code>%</code>	Módulo	<code>a % b</code>

▼ Operadores de asignación

Operador	Significado	Ejemplo
<code>=</code>	Asignación	<code>a = b</code>

Operador	Significado	Ejemplo
<code>+=</code>	Suma y asignación	<code>a += b</code> (<code>a = a + b</code>)
<code>-=</code>	Resta y asignación	<code>a -= b</code> (<code>a = a - b</code>)
<code>*=</code>	Multiplicación y asignación	<code>a *= b</code> (<code>a = a * b</code>)
<code>/=</code>	División y asignación	<code>a /= b</code> (<code>a = a / b</code>)
<code>%=</code>	Módulo y asignación	<code>a %= b</code> (<code>a = a % b</code>)

▼ Operadores relacionales

Operador	Significado	Ejemplo
<code>==</code>	Igualdad	<code>a == b</code>
<code>!=</code>	Distinto	<code>a != b</code>
<code><</code>	Menor que	<code>a < b</code>
<code>></code>	Mayor que	<code>a > b</code>
<code><=</code>	Menor o igual que	<code>a <= b</code>
<code>>=</code>	Mayor o igual que	<code>a >= b</code>

▼ Operadores lógicos

Operador	Significado	Ejemplo	Resultado
<code>&&</code>	y	<code>(7 > 2) && (2 < 4)</code>	Las dos condiciones son verdaderas
<code> </code>	o	<code>(7 > 2) (2 < 4)</code>	Al menos una de las condiciones es verdadera
<code>!</code>	no	<code>!(7 > 2)</code>	La condición es falsa

▼ Operadores especiales

Operador	Significado	Ejemplo
<code>++</code>	Incremento	<code>a++</code> (postincremento) <code>++a</code> (preincremento)
<code>--</code>	Decremento	<code>a--</code> (posdecremento) <code>--a</code> (predecremento)
<code>(tipo)expr</code>	Cast	<code>a = (int)b</code>
<code>+</code>	Concatenación de cadenas	<code>a = "str1" + "str2"</code>
<code>.</code>	Acceso a variables y	<code>a = obj.var1</code>

Operador	Significado	Ejemplo
	métodos	
()	Agrupación de expresiones	<code>a = (a + b) * c</code>

▼ Condicionales

```
public static void Main(string[] args) {
    int num = 21;

    if (num % 2 == 0) {
        Console.WriteLine("par");
    } else {
        Console.WriteLine("impar");
    }

    var edad = 7;

    if (edad < 12 || edad > 65) {
        Console.WriteLine("Recibe descuento");
    }

    var opcion = "salir";

    switch (opcion) {
        case "saludar":
            Console.WriteLine("Buenas");
            break;
        case "despedir":
            Console.WriteLine("Hasta luego");
            break;
        case "salir":
            Console.WriteLine("Saliendo del programa");
            break;
        default:
            Console.WriteLine("No se reconoce la opcion");
            break;
    }
}
```

▼ Bucle while y do while

```
public static void Main(string[] args) {
    var i = 0;

    while (i < 10) {
        Console.WriteLine(i);
        i++;
    }

    var condicion = false;

    do {
        Console.WriteLine("Hola");
    } while (condicion);
}
```

```
while (condicion);  
}
```

▼ Bucle for

```
public static void Main(string[] args) {  
    for (int i = 0; i < 10; i++) {  
        Console.WriteLine(i);  
    }  
}
```

▼ Clase string

```
public static void Main(string[] args) {  
    string cadena = "Hola mundo";  
  
    Console.WriteLine("Longitud de la cadena: " + cadena.Length);  
    Console.WriteLine("Indice de subcadena: " + cadena.IndexOf("mundo"));  
    Console.WriteLine("Devolver subcadena: " + cadena.Substring(0, 4));  
    Console.WriteLine("Comparar dos cadenas: " + cadena.Equals("Hola mundo"));  
    Console.WriteLine("Empieza con una subcadena: " + cadena.StartsWith("Hola"));  
    Console.WriteLine("Acaba con una subcadena: " + cadena.EndsWith("mundo"));  
    Console.WriteLine("Reemplazar una palabra: " + cadena.Replace("Hola", "Hello"));  
    Console.WriteLine("Reemplazar un caracter: " + cadena.Replace("o", "a"));  
  
    string cadena2 = "      muchos espacios      "  
    Console.WriteLine("Borrar espacios: " + cadena2.Trim());  
  
    string cadena3 = "";  
    Console.WriteLine("Cadena nula o vacia: " + String.IsNullOrEmpty(cadena3));  
  
    string cadena4 = "  ";  
    Console.WriteLine("Cadena nula o espacios en blanco: " + String.IsNullOrWhiteSpace(cadena4));  
}
```

▼ Arrays

```
class MainClass  
{  
    public static void Main(string[] args) {  
        int[] numeros = {4, 1, -6, 12};  
  
        for (var i = 0; i < numeros.Length; i++) {  
            Console.WriteLine(numeros[i]);  
        }  
    }  
}
```

▼ Listas

```
public static void Main(string[] args) {  
    List<int> numeros = new List<int>();  
}
```

```
numeros.Add(1);  
numeros.Add(2);  
numeros.Add(3);  
  
foreach (int numero in numeros) {  
    Console.WriteLine(numero);  
}  
}
```