

## C# por Consola

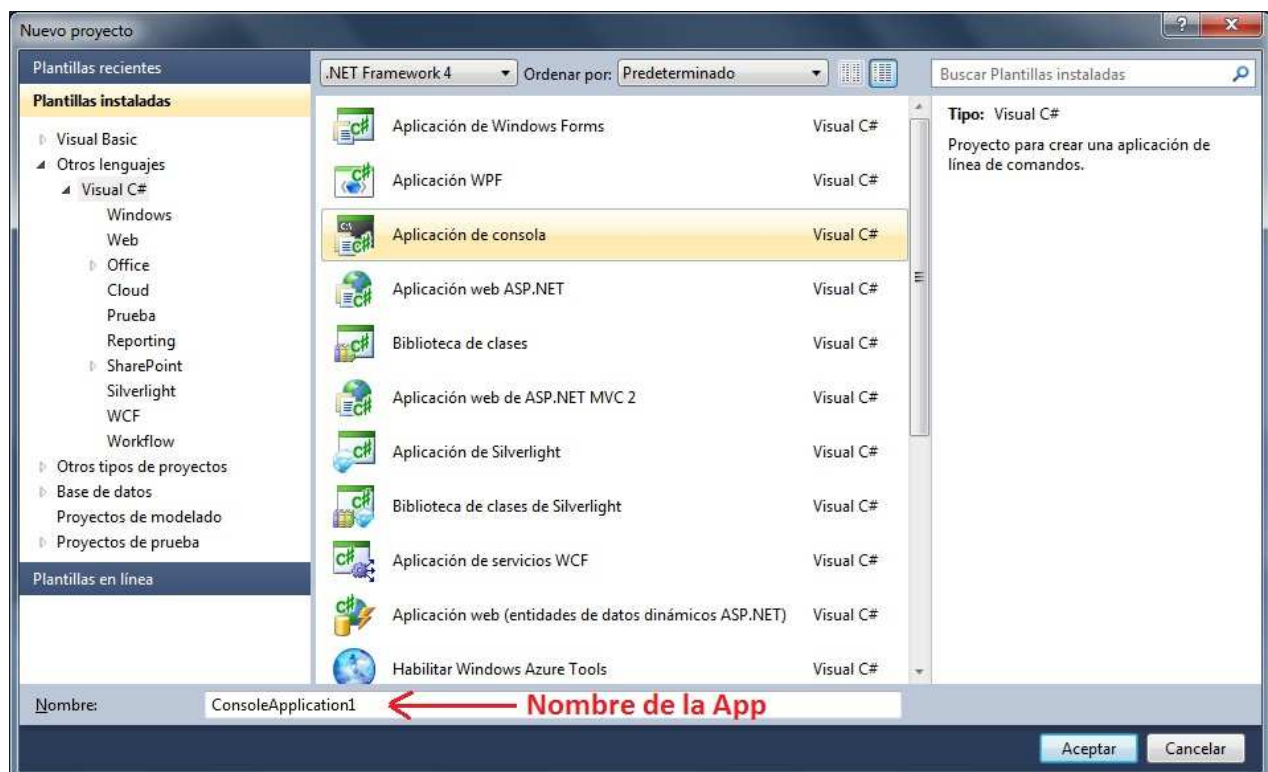
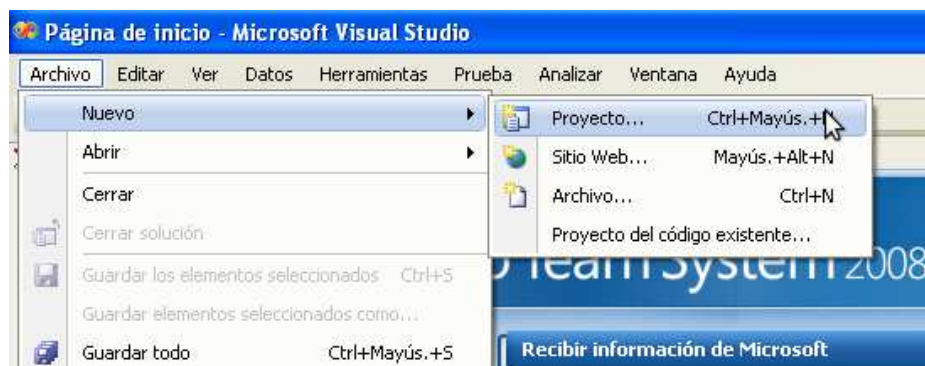
**C#** (leído como "C Sharp"). La sintaxis y estructuración de C# es muy similar a la C++, ya que la intención de Microsoft con C# es facilitar la migración de códigos escritos en estos lenguajes a C#. Se equipara en productividad a Visual Basic.

Se puede definir una aplicación de consola como aquella que se ejecuta en una ventana de MS-DOS, es decir, en línea de comandos.

Lo más común dentro del desarrollo bajo la plataforma .Net es la creación de aplicaciones Web o aplicaciones Windows sin embargo la mejor forma de sentar unas bases firmes acerca de la programación orientada a objetos es comenzar construyendo aplicaciones sencillas de consola.

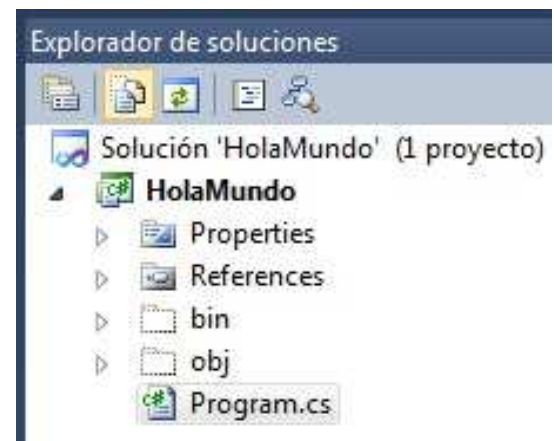
### Aplicación básica Hola Mundo!

Archivo → Nuevo Proyecto → Visual C# → Aplicación de Consola



Esta acción creará el nuevo Proyecto en la solución. Esto genera los elementos "Properties, References y Program.c".

Program.c es el archivo en el cual debemos agregar el código para cumplir nuestra finalidad (imprimir "Hola Mundo").



El siguiente código muestra el contenido de Program.cs

```
using System; // Namespace Importados
using System.Collections.Generic; // son un conjunto de clases y librerías
using System.Linq; // para ser reutilizados, ejemplo el System
using System.Text; // tiene la clase Console y el método WriteLine

namespace HolaMundo
{
    class Program // define la clase Program
    {
        static void Main(string[] args) // método Main: es lo 1º en ejecutarse
        { // static -> indica que el método main es propio de la clase y no del objeto
            // void -> indica que el método main no devuelve ningún valor

            Console.WriteLine("Hola Mundo");
            Console.Read(); // lee, se usa para realizar pausa
        }
    }
}
```

Explicación de los imports que aparecen por defecto:

**using System;** Contiene clases fundamentales y clases base que definen tipos de datos de referencia y de valor de uso frecuente, eventos y controladores de eventos, interfaces, atributos y excepciones de procesamiento.

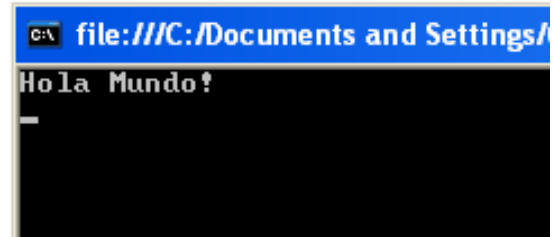
**using System.Collections.Generic;** Contiene interfaces y clases que definen colecciones genéricas, permitiendo que los usuarios creen colecciones con establecimiento inflexible de tipos para proporcionar una mayor seguridad de tipos y un rendimiento mejor que los de las colecciones no genéricas con establecimiento inflexible de tipos.

**using System.Linq;** Provee clases e interfaces que soportan queries que usan Language-Integrated Query (LINQ)

**using System.Text;** Contiene clases que representan codificaciones de caracteres ASCII, Unicode, UTF-7, y UTF-8; clases base abstractas para la conversión de bloques de caracteres en bloques de bytes y viceversa; y una clase auxiliar que manipula y da formato a objetos String sin necesidad de crear instancias intermedias de String.

Falta generar (boton derecho sobre la solución y "generar solución").

Esto sería como Compilar) y ejecutar (F5, o en la barra de herramientas ir a Depurar-> Iniciar depuración). y Listo! **Hola Mundo!**



## Métodos útiles para trabajar con la consola

Veamos los más usados:

### **1- Console.WriteLine();**

Escribe una nueva línea en la consola, es como presionar Enter y escribir una línea. Por ejemplo, podemos hacer:

Console.WriteLine("Lo que quiera q salga en la consola");  
También podremos imprimir las variables de nuestro programa, para esto hay dos formas, pero veamos mejor dos ejemplos que son lo mismo:

```
//declaramos dos variables
string nombre = "Invitado";
int edad = 22;
//Imprimimos una linea en la consola de dos formas distintas
Console.WriteLine("Tu nombre es: "+nombre+" y tienes " + edad + " años.");
//En C# se usa el operador + para concatenar cadenas
Console.WriteLine("Tu nombre es {0} y tienes {1} años", nombre, edad);
```

Como ven la segunda vía es mucho más simple, solo debes recordar que el número que está entre corchetes, es el índice del parámetro que se pasa al final del método.

### **2- Console.Write()**

Hace casi lo mismo que Console.WriteLine() pero sin cambiar de línea, o sea... concatena el texto que le pasemos al método con lo que se haya escrito hasta ese momento en la consola.

### **3- Console.ReadLine()**

Se usa para leer lo que el usuario escribió en la consola, y podemos guardarlo en una variable para después usarlo en nuestro programa. No hay que pasarle ningún parámetro y el programa solo continuará cuando el usuario teclee alguna línea y presione Enter. Podemos hacer algo como esto:

```
Console.Write("escribe tu nombre: ");
string nombre2 = Console.ReadLine();
```

Importante: Si lo que queremos es recoger lo que el usuario escriba como un tipo en específico, por ejemplo, un entero, deberemos usar el método estático `int.Parse`, por ejemplo

```
Console.Write("escribe tu numero: ");
int x = int.Parse(Console.ReadLine());
```

Ya que lo que devuelve `Console.ReadLine()` es un string.

## 4-Console.ReadKey()

Este método resulta importante algunas veces, sobre todo cuando queremos darle varias opciones al usuario (a, b, c, d). Aunque se puede utilizar Console.ReadLine(). En este caso, este método lo que devuelve es un objeto de tipo ConsoleKeyInfo, y se puede utilizar de la siguiente forma, por ejemplo:

```
Console.WriteLine("Presiona una tecla");  
ConsoleKeyInfo c = Console.ReadKey();  
Console.WriteLine("Presionaste la tecla "+ c.KeyChar);
```

## Declaración y Tipos de variables

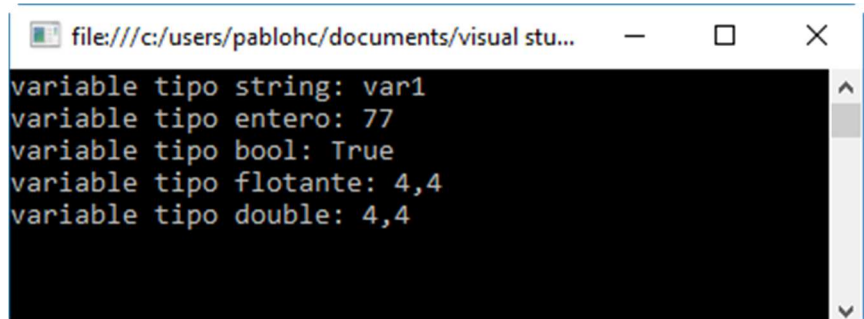
```
string charr = "var1";  
int ent = 77;  
bool its = true; // para datos decimales float y double  
float vflot = 4.4f; // 4 byte (32 bit) para la precisión de un numero decimal  
double doble = 4.4; // 8 byte (64 bit) para la precisión de un numero decimal
```

Los de uso más común son: char, float, int y bool

### Ejercicio 1

Realice una aplicación por consola en la que declare distintos tipos de datos en C# y muestre por pantalla.

```
7 namespace Ejercicios_Clas_1  
8 {  
9     class Program  
10    {  
11        static void Main(string[] args)  
12        {  
13            // ejercicio 1  
14            string charr = "var1";  
15            int ent = 77;  
16            bool its = true; // para datos decimales float y double  
17            float vflot = 4.4f; // 4 byte (32 bit) para la precisión de un numero decimal  
18            double doble = 4.4; // 8 byte (64 bit) para la precisión de un numero decimal  
19  
20            Console.WriteLine("variable tipo string: " + charr);  
21            Console.WriteLine("variable tipo entero: " + ent);  
22            Console.WriteLine("variable tipo bool: " + its);  
23            Console.WriteLine("variable tipo flotante: " + vflot);  
24            Console.WriteLine("variable tipo double: " + doble);  
25            Console.Read();  
26        }  
27    }  
28 }
```



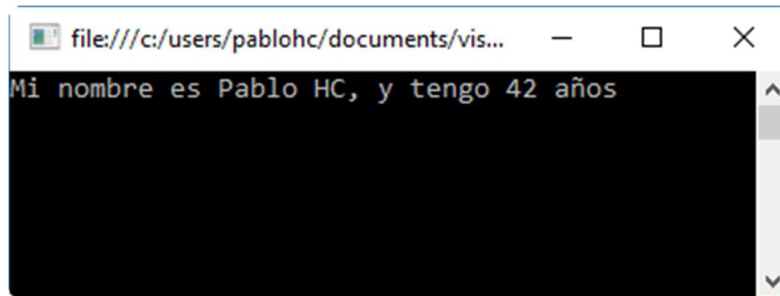
```
file:///c:/users/pablohc/documents/visual stu...  
variable tipo string: var1  
variable tipo entero: 77  
variable tipo bool: True  
variable tipo flotante: 4,4  
variable tipo double: 4,4
```



## Ejercicio 2

Declare en un string su nombre y en un entero su edad y que lo muestre por pantalla después de las otras variables

```
string nombre = "Pablo HC";  
int edad = 42;  
  
Console.WriteLine("Mi nombre es {0}, y tengo {1} años", nombre, edad);
```

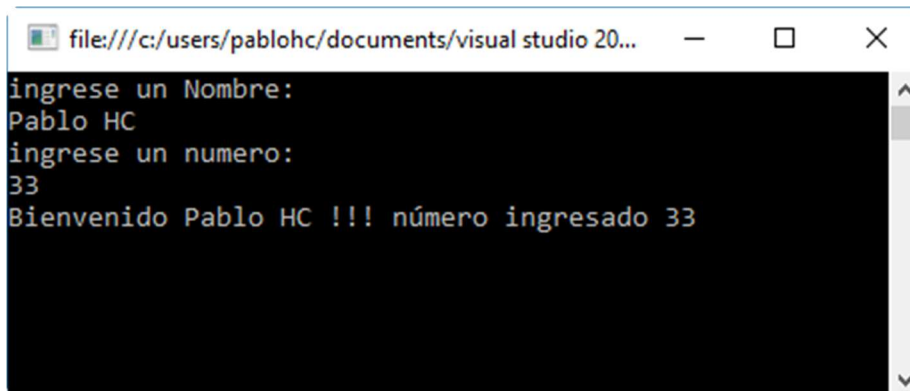


```
file:///c:/users/pablohc/documents/vis...  
Mi nombre es Pablo HC, y tengo 42 años
```

## Ejercicio 3

Cree una aplicación que solicite su nombre y un número por pantalla y luego los muestre en un saludo de Bienvenida

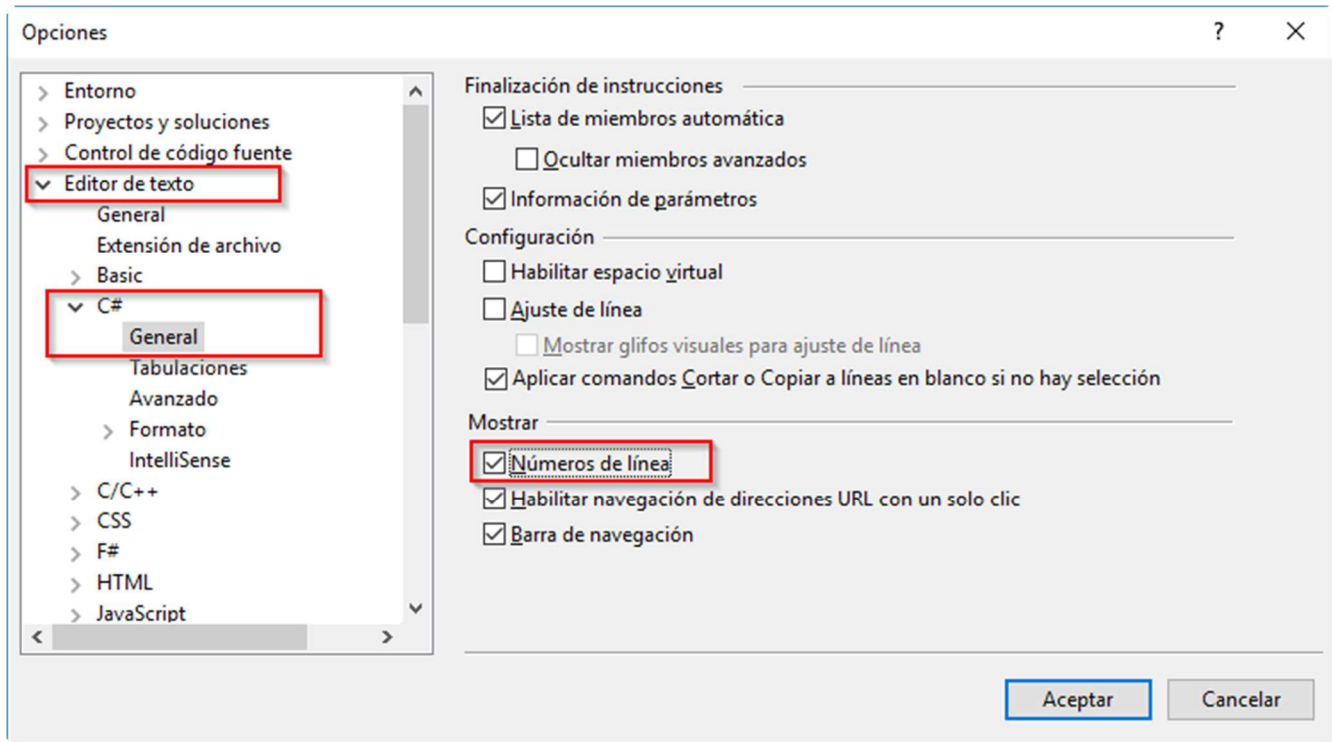
```
Console.WriteLine("ingrese un Nombre:");  
string nombre = Console.ReadLine();  
Console.WriteLine("ingrese un numero:");  
int num = int.Parse(Console.ReadLine());  
Console.WriteLine("Bienvenido {0} !!! número ingresado {1}", nombre, num);  
Console.Read();
```



```
file:///c:/users/pablohc/documents/visual studio 20...  
ingrese un Nombre:  
Pablo HC  
ingrese un numero:  
33  
Bienvenido Pablo HC !!! número ingresado 33
```

## Para mostrar los números de línea en el código

En el menú Herramientas, elija Opciones:



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Ejercicios_Clase_1
8 {
9     class Program
10     {
11         static void Main(string[] args)
12         {
13
14         }
15     }
16 }
17
18
```

## Operadores en C#

### Operaciones aritméticas básicas

+	Suma
-	Resta, negación
*	Multiplicación
/	División
%	Resto de la división ("módulo")

**Ejercicio 4:** Ingresa por teclado 2 números y calcule las operaciones básicas de estos

```
int num1, num2, suma, resta, multi;
double div;

Console.WriteLine("primer num: ");
num1 = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("segundo num: ");
num2 = Convert.ToInt32(Console.ReadLine());
suma = num1 + num2;
resta = num1 - num2;
multi = num1 * num2;
div = (double)num1 / (double)num2;

Console.WriteLine("la suma es: {0}", suma);
Console.WriteLine("la resta es: {0}", resta);
Console.WriteLine("la multi es: " + multi);
Console.WriteLine("la division es: " + div);
Console.Read();
```

### **Nota:**

Convert.ToInt32 realiza la conversión a número entero

**Operadores relacionales: <, <=, >, >=, ==, !=**

Operador	Operación
<	Menor que
>	Mayor que
<=	Menor o igual que
>=	Mayor o igual que
==	Igual a
!=	No igual a (distinto de)

**Operadores lógicos: &&, ||, !**

Operador	Significado
&&	Y
	O
!	No

**Ejemplos:**

```
if ((opcion==1) && (usuario==2)) ...
if ((opcion==1) || (opcion==3)) ...
if (!(opcion==opcCorrecta) || (tecla==ESC)) ...
```

**Estructuras de Control****Instrucción IF**

```
if (<condición>){
    <instruccionesIf>}
else{
    <instruccionesElse>}
```

**Ejercicio 5:** Ingresa por teclado 1 número y diga si es mayor que 0

```
int num1;
Console.Write("primer num: ");
num1 = Convert.ToInt32(Console.ReadLine());

if (num1 >= 0) {
    Console.WriteLine("el num es MAYOR que cero");
} else {
    Console.WriteLine("el num es Menor que cero");
}
Console.Read();
```



**Ejercicio 6:** Ingrese un número y usando el resto verifique si es par o Impar

```
float num;
Console.Write("Ingrese número para ver si es Par o Impar:");
num = float.Parse(Console.ReadLine());
if (num % 2 == 0) {
    Console.Write("El {0} es par", num);
}else {
    Console.Write("El {0} es impar", num);
}
Console.Read();
```

## Instrucción switch

La instrucción switch permite ejecutar unos u otros bloques de instrucciones según el valor de una cierta expresión. Su estructura es:

```
switch (expresión)
{
    case valor1: sentencia1;
        break;
    case valor2: sentencia2;
        sentencia2b;
        break;
    ...
    case valorN: sentenciaN;
        break;
    default:
        otraSentencia;
        break;
};
```

**Ejercicio 7:** Ingrese por teclado un nombre y salude de forma personalizada si somos "Juan" o "Pedro" o "Pepe"

```
string nombre;
Console.WriteLine("Introduce tu nombre");
nombre = Console.ReadLine();
switch (nombre)
{
    case "Juan": Console.WriteLine("Bienvenido, Juan");
        break;
    case "Pedro": Console.WriteLine("Que tal estas, Pedro");
        break;
    case "Pepe": Console.WriteLine("Órale, Pepe");
        break;
    default: Console.WriteLine("Quien eres?...desconocido.");
        break;
}
Console.Read();
```

**Ejercicio Creando un Menú simple usando Switch:**

Creamos un método llamado menú y se llama desde el Main:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Ejer_Clas_2_Menu
8 {
9     class Program
10     {
11         static void Main(string[] args)
12         {
13             Menu(); // llama al Método menú
14         }
15
16         static void Menu() {
17             int valor;
18             Console.WriteLine("1. Opcion 1");
19             Console.WriteLine("2. Opcion 2");
20             Console.WriteLine("3. Salir");
21             Console.WriteLine("Seleccione una opción: ");
22             valor = Convert.ToInt32(Console.ReadLine());
23
24             switch (valor)
25             {
26                 case 1:
27                     Console.WriteLine("1. Opcion 1");
28                     Console.ReadLine();
29                     break;
30                 case 2:
31                     Console.WriteLine("2. Opcion 2");
32                     Console.ReadLine();
33                     break;
34                 case 3:
35                     break;
36                 default:
37                     Console.WriteLine("Ingresa una Opción");
38                     break;
39             } // cierra switch
40         } // cierra Menu
41     }
42 }
43 }
```

**Realice las siguientes Tareas Usando un Menú case por cada tarea:**

**Tarea 1:** Ingrese dos números por teclado y que la suma de estos dos números se multiplique por un tercer número que Ud. Ingrese después de mostrar el resultado de dicha suma.

**Tarea 2:** Ingrese 4 números y muestre su promedio

**Tarea 3 Instrucción IF:** Ingrese por teclado 2 números y diga cuál de ellos es mayor

**Tarea 4 Instrucción IF:** Ingrese por teclado 2 números y realice una operación básica según la opción escogida