

## Programación básica

- . Variables
- . Expresiones booleanas y condicionales
- . Loops
- . Arrays y listas
- . Métodos

## IDE = Integrated development environment

Incluye:

- . Editor de código
- . Compilador
- . Herramientas para facilitar la programación

Primer programa en C# que imprime una cadena de Texto

```
1 // Programa que imprime la cadena de texto en C#  
2 Console.WriteLine("Hola Mundo!");
```

```
// Método que sirve para que el ejecutable se cierre  
// hasta pulsar una tecla  
Console.ReadKey();
```

Tipos de comentarios en C#

```
// Comentario de línea  
  
/*  
 * Comentario de bloque o multilínea  
 */
```

Tipos de datos básicos

# TIPOS DE DATOS

- int** **Números enteros.** Ejemplos: 0, 1, 50, 500000, -1, -25.  
Ejemplo en juegos: cantidad de munición.
- float** **Números con parte decimal.** Ejemplos: 1.1f, 20.5f, -5.04f.  
Ejemplo en juegos: distancia.
- string** **Cadenas de texto.** Ejemplos: "Mario", "Espada legendaria".  
Ejemplo en juegos: nombre de un personaje.
- bool** Solamente los valores **verdadero y falso** (true y false).  
Ejemplo en juegos: si un personaje está vivo o muerto.

Practicando

Código

```
// Creando variables
// Variable entera
// Declaración de variable
int myFirstVar;
// Inicialización de variable
myFirstVar = 1000;

// Las dos cosas en una sola línea
int coins = 50;

// Imprimiendo las variables
Console.WriteLine(myFirstVar);
Console.WriteLine(coins);
```

Código

```
// Sumando los valores anteriores
Console.WriteLine(myFirstVar + coins);
```

Código declarando variable entera, flotante, cadena y booleana.

Salida

```
Consola de depuración de Microsoft Visual Studio
1000
50
```

Salida

```
Consola de depuración de Microsoft Visual Studio
1050
```

```
// Declaración de variable
int myFirstVar;
// Inicialización de variable
myFirstVar = 1000;

// Variable de tipo flotante
float myFloat = 10.5f;

// Variable tipo String
string myString = "Hola mundo!";
Console.WriteLine(myString);

// Variable tipo booleana
bool myBool = true;
```

¿Cómo elegir **nombres** para las variables?

Elegir un **nombre representativo** de su contenido

```
int h; // No representativo
int health; // Representativo
int playerHealth; // Representativo
```

Conversión del camello para nombrar variables, donde la primera palabra es en minúscula y si lleva más palabras estas deben empezar en mayúscula. Esto simula la joroba de un camello.

Usar **camelCase**

```
float distanceToGoal;
int playerHealth;
bool isGameOver;
```

Solo **letras, números y guión bajo**

`int health!; // No válido`



`int health; // Válido`

**No** se puede **empezar con un dígito**

`int 1playerHealth; // Inválido`

`int player1Health; // Válido`

`int playerHealth1; // Válido`

**No** se pueden **usar palabras clave de C#**

`int float; // No válido`

`int for; // No válido`

`int if; // No válido`

`int float1; // Válido`

**En lo posible, usar inglés**

El inglés es el lenguaje natural de la programación

## Ejercicio

Determinar el tipo de dato más adecuado para almacenar:

- . Un **puntaje**
- . El **tiempo de duración** de una animación
- . La **cantidad de enemigos** restantes en un nivel
- . Si un jugador está **vivo o muerto**

# Ejercicio

Determinar el tipo de dato más adecuado para almacenar:

- . Un **puntaje**
- . El **tiempo de duración** de una animación
- . La **cantidad de enemigos** restantes en un nivel
- . Si un jugador está **vivo o muerto**
- . El **nombre** de un ítem

Declarar e inicializar dichas variables.

Código del ejercicio

```
// Declarando variables e inicializando variables

int score = 100;
float run_animation_duration = 0.3f;
int number_of_enemies = 30;
bool player_is_alive = true;
string item_1 = "Espada";
```

Operadores aritméticos

Código

```
1 // Operadores aritméticos
2 int a = 10;
3 int b = 5;
4 int c;
5
6 Console.WriteLine("Operando los números " + a + " y " + b);
7 // Suma
8 c = a + b;
9 Console.WriteLine(a + " + " + b + " = " + c);
10
11 // Resta
12 c = a - b;
13 Console.WriteLine(a + " - " + b + " = " + c);
14
15 // Multiplicación
16 c = a * b;
17 Console.WriteLine(a + " * " + b + " = " + c);
18
19 // División
20 c = a / b;
21 Console.WriteLine(a + " / " + b + " = " + c);
22
23 // Módulo
24 c = a % 2;
25 Console.WriteLine(a + " % " + 2 + " = " + c);
26
27 Console.ReadKey();
```

Salida

```
Consola de depuración de Microsoft Visual Studio
Operando los números 10 y 5
10 + 5 = 15
10 - 5 = 5
10 * 5 = 50
10 / 5 = 2
10 % 2 = 0
```

Formas abreviadas de los operadores aritméticos

Código

```
1 // Formas abreviadas de los operadores aritméticos
2 int a = 10;
3
4 Console.WriteLine("Formas abreviadas de los operadores aritméticos");
5 // Suma
6 a += 5;
7 Console.WriteLine(a);
8
9 // Resta
10 a -= 5;
11 Console.WriteLine(a);
12
13 // Multiplicación
14 a *= 5;
15 Console.WriteLine(a);
16
17 // División
18 a /= 5;
19 Console.WriteLine(a);
20
21 // Módulo
22 a %= 2;
23 Console.WriteLine(a);
24
25 // Incremento;
26 a++;
27 Console.WriteLine(a);
28
29 // Decremento
30 a--;
31 Console.WriteLine(a);
32
33 Console.ReadKey();
```

Salida

```
Consola de depuración de Microsoft Visual Studio
Formas abreviadas de los operadores aritméticos
15
10
50
10
0
1
0
```



# Orden de evaluación en operaciones aritméticas

1. Paréntesis
2. Multiplicaciones, divisiones y módulo
3. Sumas y restas

Concatenación

Código

```
1 string player_name = "Alfredo";  
2 int gold_coins = 5;  
3  
4 Console.WriteLine("Hola " + player_name + " tienes " + gold_coins + " monedas de oro");  
5  
6 Console.ReadKey();
```

Salida

Selecciónar C:\Users\alfre\source\repos\programacion\_basica\programacion\_basica\bin\Debug\net8.0\programacion\_basica.exe  
Hola Alfredo tienes 5 monedas de oro

## Ejercicio

En una pista de carreras triangular, la **distancia** total de **una vuelta** se puede calcular como:

$$distance1 + distance2 + distance3$$



Definir **3 variables** para las 3 distancias e **inicializarlas** en 30, 40 y 50 respectivamente. Luego **calcular** la distancia total para dar **8 vueltas**.

Código

```

1 // Variables de las distancias
2 float distance_1 = 30f;
3 float distance_2 = 40f;
4 float distance_3 = 50f;
5
6 // Variable que calcula una vuelta
7 float lap = distance_1 + distance_2 + distance_3;
8
9 Console.WriteLine("La distancia de 8 vueltas es = " + (lap * 8));
10 Console.ReadKey();

```

Salida

Consola de depuración de Microsoft Visual Studio

```
La distancia de 8 vueltas es = 960
```

Entradas de Usuario

Código

```

1 // Entradas de usuario
2 Console.WriteLine("¿Cuál es tu nombre?");
3
4 string player_name = Console.ReadLine();
5 Console.WriteLine("Hola " + player_name);
6
7 Console.ReadKey();

```

Salida

Consola de depuración de Microsoft Visual Studio

```
¿Cuál es tu nombre?
Alfredo
Hola Alfredo
```

Conversiones o casteo

Código

```

1 // Conversiones de tipo o casteo
2 // De entero a float (casteo implícito o automática)
3 using System.Globalization;
4
5 float my_float = 5;
6 Console.WriteLine(my_float);
7
8 // De float a entero (casteo explícito)
9 int my_int = (int)2.5f;
10 Console.WriteLine(my_int);
11
12 // De entero a string / de float a string
13 int my_int_2 = 10;
14 float my_float_2 = 3.1416f;
15 string my_string = my_int_2.ToString();
16 string my_string_2 = my_float_2.ToString();
17
18 Console.WriteLine(my_string);
19 Console.WriteLine(my_string_2);
20
21 // De string a entero
22 string my_string_3 = "123";
23 int my_int_3 = Convert.ToInt32(my_string_3);
24 Console.WriteLine(my_int_3);
25
26 Console.ReadKey();

```



Salida

```
Consola de depuración de Microsoft Visual Studio
5
2
10
3.1416
123
```

## Ejercicio

Escribir un programa que **pida el nombre y el nivel** de un personaje como **entradas**, y que luego las muestre en pantalla.

Por ejemplo, si el usuario ingresó el nombre "Geralt" y el valor 15, que la consola muestre un mensaje como:

*Nombre del jugador: Geralt. Nivel: 15*

Código

```
1  string player_name;
2  int player_level;
3  string temporal;
4
5  Console.Write("Dime el nombre de tu personaje: ");
6  player_name = Console.ReadLine();
7
8  Console.Write("Dime el nivel de tu personaje: ");
9  temporal = Console.ReadLine();
10 player_level = Convert.ToInt32(temporal);
11
12 Console.WriteLine("Nombre: " + player_name);
13 Console.WriteLine("Nivel: " + player_level);
```

Salida

```
Consola de depuración de Microsoft Visual Studio
Dime el nombre de tu personaje: Alfredo
Dime el nivel de tu personaje: 10
Nombre: Alfredo
Nivel: 10
```

# Ejercicio

Declarar e inicializar **tres variables enteras** que almacenen valores de salud. Calcular el **promedio** de las tres saludes.

*Ayuda: El promedio se calcula sumando todos los valores (todas las saludes) dividido entre la cantidad de valores (tres en este caso).*

Código

```
1 // Ejercicio. Calcular el promedio de 3 números
2 int health_1 = 3;
3 int health_2 = 5;
4 int health_3 = 2;
5
6 int tola_health = health_1 + health_2 + health_3;
7 int average = tola_health / 3;
8
9 Console.WriteLine("Salud 1: " + health_1);
10 Console.WriteLine("Salud 2: " + health_2);
11 Console.WriteLine("Salud 3: " + health_3);
12 Console.WriteLine("Promedio: " + average);
13 Console.ReadKey();
14
```

Salida

```
Consola de depuración de Microsoft Visual Studio
Salud 1: 3
Salud 2: 5
Salud 3: 2
Promedio: 3
```