



# Programacion Estructurada C#

**Instructor:**  
**Carlos Daniel Gomez**  
**Daza**



[www.sena.edu.co](http://www.sena.edu.co)

# Elementos del lenguaje



## Elementos del lenguaje

<i>Elemento</i>	<i>Ejemplos</i>
palabraclave	<code>public class int</code>
literal	<code>7 18 14</code>
operador	<code>+ - * / % =</code>
identificador	<code>suma Console nota</code>
puntuación	<code>; { } ,</code>

# Literales



## Literales

<i>tipo</i>	<i>literales</i>
logicas	true false
reales	3.14159
enteras	7 4 14 42
caracter	'X'
cadena	"Hello"
nulo	null

# Operadores



## Operadores

<i>categoria</i>	<i>operadores</i>
aritméticos	+ - * / %
relacionales	== != < <= > >=
lógicos	&&    ! ?:
asignación	=

# Namespaces



- En aplicaciones grandes pueden existir conflictos en la nomenclatura de los tipos
- Los tipos se agrupan en categorías o Namespaces
- Los Namespaces deben de ser referenciados para utilizar los tipos incluidos en ellos
- Podemos incluir un namespace al principio del programa con la sentencia „using“
- Los Namespaces más utilizados son:

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

# Namespaces



```
class HolaMundo
{
    static void Main( )
    {
        System.Console.WriteLine("Hola Mundo");
    }
}
```

A screenshot of a Windows command prompt window. The title bar shows the file path 'C:\WINDOWS\system32\cmd.exe'. The window has a black background with white text. The first line of output is 'Hola Mundo' and the second line is 'Press any key to continue . . .'.

```
C:\WINDOWS\system32\cmd.exe
Hola Mundo
Press any key to continue . . .
```

# Signos de puntuacion



## Signos de puntuación

- En el código los espacios en blanco y los saltos de línea se ignoran
- El código está formado por sentencias
- Con punto y coma (;) indicamos el final de una sentencia
- Varias sentencias pueden estar agrupadas en un bloque delimitado por llaves {...}

# Signos de puntuacion



```
class Notas
{
static void Main( )
{
int teoria = 4;
int practica = 2;
int final;
final = teoria + practica;
Console.WriteLine(final);
}
}
```

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.exe'. The command prompt displays the number '6' on the first line and the text 'Press any key to continue . . .' on the second line, indicating the program has finished execution and is waiting for a key press.



# Declaraciones



## Declaraciones

- Las declaraciones introducen una variable en un bloque
- Una variable tiene un identificador y un tipo
- Al declarar una variable podemos asignarle un valor inicial
- En una sentencia podemos declarar varias variables de un mismo tipo
- Ejemplos
  - `int nota;`
  - `int nota = 5;`
  - `int practica = 4, teoria = 2, nota;`
  - `string nombre;`
  - `string apellido1 = "Cuartas", apellido2 = "Hernandez";`

# Expresiones



## Expresiones

- Las expresiones generan un valor
- No podemos utilizar una variable en una expresión a menos que antes le asignemos un valor

- Ejemplo:

```
int a = 4, b = 2, c;  
c = b * (a - 1);
```

- Ejemplo:

```
int a, b = 2, c;  
c = b * ((a = 4) - 1);
```

# Clase console



## Clase Console

- Se encuentra en el namespace **System**.
- Nos permite la entrada y salida de datos.
- **Console.ReadLine( );**  
Lee un dato del teclado al pulsar intro.  
Devuelve un string.
- **Console.Write(string);**  
Escribe un string en la consola.
- **Console.WriteLine(string);**  
Escribe un string en la consola y salta a la siguiente linea.
- Tanto Write como WriteLine aceptan también otros tipos además de string.

# Clase console



```
using System;
class Saludo
{
    static void Main( )
    {
        string nombre;
        Console.WriteLine("Como te llamas ?");
        nombre = Console.ReadLine();
        Console.Write("Hola, " );
        Console.WriteLine(nombre);
        Console.ReadLine();
    }
}
```

A screenshot of a Windows command prompt window. The title bar at the top shows the file path "C:\WINDOWS\system32\cmd.exe". The command prompt displays the text "Como te llamas ?" on the first line, "Carlos" on the second line (the user's input), and "Hola, Carlos" on the third line (the program's output).

```
C:\WINDOWS\system32\cmd.exe
Como te llamas ?
Carlos
Hola, Carlos
```

# Tipos de datos



## Tipos enteros

<i>tipo</i>	<i>System.</i>	<i>Rango</i>	<i>signo</i>	<i>bits</i>
sbyte	SByte	-128 a 127	si	8*
ushort	UInt16	0 a 65.535	no	16
uint	UInt32	0 a $4 \times 10^9$	no	32
ulong	UInt64	0 a $18 \times 10^{18}$	no	64
byte	Byte	0 a 255	no	8*
short	Int16	-32.768 a 32.767	si	16
int	Int32	$\pm 2 \times 10^9$	si	32
long	Int64	$\pm 9 \times 10^{18}$	si	64



# System.Int32



## System.Int32

- **int** es un alias para **System.Int32**
  - **int** suma = 14;
  - **System.Int32** suma = 14;
- **MaxValue** y **MinValue** nos dan el rango
  - **int.MaxValue** vale 2.147.483.647
  - **int.MinValue** vale -2.147.483.648
- **Convert.ToInt32(...)** para convertir cualquier tipo a **System.Int32**
  - **Convert.ToInt32("14")** vale 14 entero
  - **Convert.ToInt32(14.73)** vale 15 entero

# Tipos reales



## Tipos reales

<i>tipo</i>	<i>System.</i>	<i>Rango</i>	<i>dígitos sig.</i>	<i>bits</i>
float	Single	$\pm 3.4 \times 10^{38}$	7	32
double	Double	$\pm 1.7 \times 10^{308}$	15	64
decimal	Decimal	$\pm 7.9 \times 10^{28}$	28	128

# System.Double



## System.Double

- **double** es un alias para **System.Double**
- **Constantes**
  - **double.MaxValue** El mayor double  $1.7 \times 10^{308}$
  - **double.MinValue** El menor double  $-1.7 \times 10^{308}$
  - **double.Epsilon** El double más pequeño mayor que 0.
  - **double.NegativeInfinity** -Infinito.
  - **double.PositiveInfinity** +Infinito.
  - **double.NaN** (Not A Number) Cuando el resultado no es un número.
- **Métodos que devuelven true o false (bool)**
  - **double.IsInfinity(d)** Comprueba si d contiene el valor infinito
  - **double.IsNaN(d)** Comprueba si d contiene el valor NaN



# Secuencia de escape



- Representan caracteres especiales
- Comienzan por una barra inclinada (\)
  - `\n` Salto de línea
  - `\t` Tabulación
  - `\\` Barra inclinada

```
Console.Write("C:\\temp\\nC:\\\\temp");
```

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.exe'. The command prompt shows the current directory as 'C:' and the user has entered 'emp'. The prompt then shows 'C:\temp' followed by the text 'Press any key to continue . . .', indicating the command executed successfully and the program is waiting for a key press to continue.


```
C:\WINDOWS\system32\cmd.exe  
C: emp  
C:\tempPress any key to continue . . .
```

# System.String



```
internal class Program
{
    static void Main(string[] args)
    {
        string cadena, resultado;
        Console.WriteLine("Introduzca cadena de longitud > 2:");
        cadena = Console.ReadLine();
        resultado = "Longitud: " +cadena.Length +
            "\nSegundo carácter: " +cadena[1];
        Console.WriteLine(resultado);
        Console.ReadLine();

    }
}
```

 C:\Users\SENA\source\repos\Programas\Progra  
Introduzca cadena de longitud > 2:  
carlos  
Longitud: 6  
Segundo carácter: a

# Tipo booleano



Tipo booleano				
tipo	System.	Contiene	Ejemplo	bits
bool	Boolean	true o false	true	16

```
C:\WINDOWS\system32\cmd.exe
```

```
True
False
False
Press any key to continue . . .
```

internal class Program

```
{
    static void Main(string[] args)
    {

        int nota;
        bool suspenso, aprobado, matricula;
        nota = 4;
        suspenso = (nota < 5);
        aprobado = (nota >= 5);
        matricula = (nota == 10);

        Console.WriteLine(suspenso);
        Console.WriteLine(aprobado);
        Console.WriteLine(matricula);

    }
}
```

# Clase convert



## Clase Convert

- Se encuentra en el namespace **System**.
- Nos permite la conversión de tipos.
- **Convert.ToInt32(a);**  
Devuelve el valor de a convertido a int
- **Convert.ToDouble(b);**  
Devuelve el valor de b convertido a double
- **Convert.ToString(c);**  
Devuelve el valor de c convertido a string
- Conversiones explícitas vs implícitas

# Clase convert



```
internal class Program
{
    static void Main(string[] args)
    {

        Console.Write("Introduzca un valor entero: ");
        int a = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine(a);

        Console.Write("Introduzca un valor real: ");
        double d = Convert.ToDouble(Console.ReadLine());
        Console.WriteLine(d);

    }
}
```

C:\WINDOWS\system32\cmd.exe

```
Introduzca un valor entero: 8
8
Introduzca un valor real: 8
8
Press any key to continue . . .
```

# Clase Math



## Clase Math

- Se encuentra en el namespace **System**
- Nos permite realizar las operaciones matemáticas más frecuentes.
- **Constantes**
  - $\text{Math.PI} = 3.1415926535\dots$
  - $\text{Math.E} = 2.7182818285\dots$
- Para utilizar sus métodos:  
**`Math.NombreMetodo(argumento1, argumento2, ...);`**
- Ejemplo  
`double a = Math.Sqrt(49) + Math.PI;`

## Clase Math. Métodos usuales.

<b>Sqrt ( x )</b>	Raiz cuadrada de x.	<b>Sqrt ( 9 ) = 3</b>
<b>Pow ( x, y )</b>	x elevado a y.	<b>Pow ( 2, 7 ) = 128</b>
<b>Log ( x )</b>	Logaritmo de x en base e.	<b>Log ( 7.389 ) = 2</b>
<b>Exp ( x )</b>	e elevado a x.	<b>Exp ( 2 ) = 7.389</b>
<b>Sin ( x )</b>	Seno de x.	<b>Sin ( 0 ) = 0</b>
<b>Cos ( x )</b>	Coseno de x.	<b>Cos ( 0 ) = 1</b>
<b>Tan ( x )</b>	Tangente de x.	<b>Tan ( 0 ) = 0</b>
<b>Abs ( x )</b>	Valor absoluto de x.	<b>Abs ( -2.1 ) = 2.1</b>
<b>Ceiling ( x )</b>	Redondeo al entero superior.	<b>Ceiling ( 2.1 ) = 3</b>
<b>Floor ( x )</b>	Redondeo al entero inferior.	<b>Floor( 2.1 ) = 2</b>
<b>Max ( x, y )</b>	Máximo de x e y.	<b>Max ( 2.1, 1.5 ) = 2.1</b>
<b>Min ( x, y )</b>	Mínimo de x e y.	<b>Min( 2.1, 1.5 ) = 1.5</b>

# Caso de division entera



```
internal class Program
{
    static void Main(string[] args)
    {

        int total = 5;
        double mitad;
        mitad = Convert.ToDouble(total) / 2;
        // ó también: mitad = (double)total / 2;
        // ó también: mitad = total / 2.0;
        Console.WriteLine(mitad);

    }
}
```

---

 C:\WINDOWS\system32\cmd.exe

```
2.5
Press any key to continue . . .
```



# Otros operadores de asignación



## Otros operadores de asignación

Nos sirven para reducir código

**Suma = Suma + 14** es igual que **Suma += 14**

Suponiendo `int a = 10;`

Operador	Ejemplo	Explicación	Resultado
<b>+=</b>	<b>a += 5</b>	<b>a = a + 5</b>	a vale 15
<b>-=</b>	<b>a -= 6</b>	<b>a = a - 6</b>	a vale 4
<b>*=</b>	<b>a *= 4</b>	<b>a = a * 4</b>	a vale 40
<b>/=</b>	<b>a /= 2</b>	<b>a = a / 2</b>	a vale 5
<b>%=</b>	<b>a %= 3</b>	<b>a = a % 3</b>	a vale 1



# Operadores de incremento y decremento



## Operadores de incremento y decremento

- **Operador incremento (  $x++$  )**
  - Añade 1 a una variable
  - Es lo mismo que  $x = x + 1$
- **Operador decremento (  $x--$  )**
  - Resta 1 a una variable
  - Es lo mismo que  $x = x - 1$
- **PreIncremento y PostIncremento**
  - $x++$  ó  $x--$  Primero ejecutan la acción y luego añaden o restan uno a la variable.
  - $++x$  ó  $--x$  Primero añaden o restan uno a la variable y luego ejecutan la acción.

# Operadores de incremento y decremento



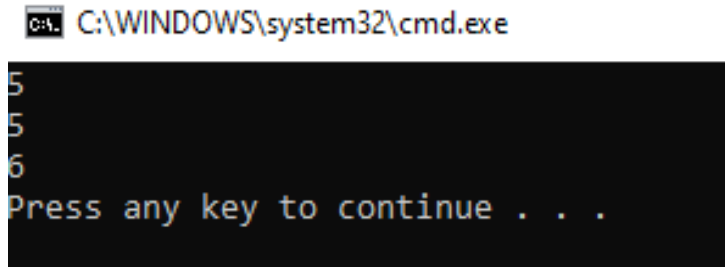
Operadores de incremento y decremento (II)		
Operador	Nombre	Ejemplo
++	PreIncremento	++a
Incrementa a en 1, luego utiliza el nuevo valor de a en la expresión		
++	PostIncremento	a++
Usa el valor de a en la expresión y después lo incrementa en 1		
--	PreDecremento	--a
Decrementa a en 1, luego utiliza el nuevo valor en la expresión		
--	PostDecremento	a--
Usa el valor de a en la expresión y después lo decrementa en 1		

# Operadores de incremento y decremento



## Post -incremento

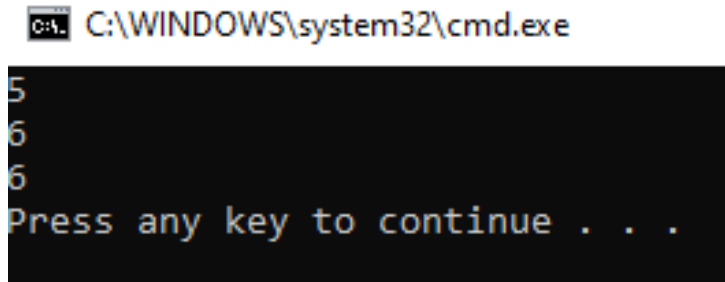
```
int c = 5;  
Console.WriteLine(c);  
Console.WriteLine(c++);  
Console.WriteLine(c);
```

A screenshot of a Windows command prompt window titled 'C:\WINDOWS\system32\cmd.exe'. The output shows the number 5 on the first line, 5 on the second line, and 6 on the third line, followed by the prompt 'Press any key to continue . . .'. This demonstrates that the value of 'c' is printed before being incremented in the post-increment operation.

```
C:\WINDOWS\system32\cmd.exe  
5  
5  
6  
Press any key to continue . . .
```

## Pre -incremento

```
int c = 5;  
Console.WriteLine(c);  
Console.WriteLine(++c);  
Console.WriteLine(c);
```

A screenshot of a Windows command prompt window titled 'C:\WINDOWS\system32\cmd.exe'. The output shows the number 5 on the first line, 6 on the second line, and 6 on the third line, followed by the prompt 'Press any key to continue . . .'. This demonstrates that the value of 'c' is incremented before being printed in the pre-increment operation.

```
C:\WINDOWS\system32\cmd.exe  
5  
6  
6  
Press any key to continue . . .
```

# Operadores de incremento y decremento



## Post -incremento

```
int c = 5;  
Console.WriteLine(c);  
Console.WriteLine(c++);  
Console.WriteLine(c);
```

A screenshot of a Windows command prompt window titled 'C:\WINDOWS\system32\cmd.exe'. The output shows the number 5 on the first line, 5 on the second line, and 6 on the third line, followed by the prompt 'Press any key to continue . . .'.

```
C:\WINDOWS\system32\cmd.exe  
5  
5  
6  
Press any key to continue . . .
```

## Pre -incremento

```
int c = 5;  
Console.WriteLine(c);  
Console.WriteLine(++c);  
Console.WriteLine(c);
```


A screenshot of a Windows command prompt window titled 'C:\WINDOWS\system32\cmd.exe'. The output shows the number 5 on the first line, 6 on the second line, and 6 on the third line, followed by the prompt 'Press any key to continue . . .'.

```
C:\WINDOWS\system32\cmd.exe  
5  
6  
6  
Press any key to continue . . .
```

# Precedencia y Asociatividad



## Precedencia y Asociatividad

 Precedencia más alta	operadores	Asociatividad	Tipo
	()	izquierda a derecha	parentesis
	++ --	derecha a izquierda	unario PostFijo
	++ -- + -	derecha a izquierda	unario Prefijo
	* / %	izquierda a derecha	multiplicación
	+ -	izquierda a derecha	adición
	< <= > >=	izquierda a derecha	relacional
	== !=	izquierda a derecha	igualdad
	?:	derecha a izquierda	condicional
	= += -= *= /= %=	derecha a izquierda	asignación

# Comentarios



## Comentarios

- Se utilizan para hacer aclaraciones sobre el código
- Los comentarios son ignorados por el compilador
- Comentarios en una linea `//...`
- Comentarios en múltiples líneas `/*...*/`

```
class HolaMundo
{
/* Este programa muestra por pantalla
la frase Hola mundo */
static void Main ( )
{
System.Console.WriteLine("Hola Mundo");
}
} // Aquí finaliza el programa
```

# Suma de 2 numeros



```
string Numero1, Numero2;  
int a, b, c;  
Console.WriteLine(  
    "Introduzca dos números: ");  
Numero1 = Console.ReadLine();  
Numero2 = Console.ReadLine();  
a = Convert.ToInt32(Numero1);  
b = Convert.ToInt32(Numero2);  
c = a + b;  
Console.WriteLine(c);
```

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.exe'. The prompt 'C:\>' is visible. The program's output is displayed in a monospaced font: 'Introduzca dos números:', followed by the user input '5' on the next line, '7' on the next line, and the result '12' on the next line. The final line of output is 'Press any key to continue . . .'.

```
C:\WINDOWS\system32\cmd.exe  
C:\>  
Introduzca dos números:  
5  
7  
12  
Press any key to continue . . .
```



# Estructuras de control



## Estructuras de control

- Por norma general en un programa se ejecuta una sentencia y después la que viene a continuación.
- Las estructuras de control nos permiten alterar este orden secuencial de ejecución.
  - Estructuras de selección.
    - if / else
    - switch
  - Estructuras de repetición.
    - while
    - do / while
    - for



# Estructura If



```
internal class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Introduzca nota: ");
        int Nota = Convert.ToInt32(
            Console.ReadLine());
        if (Nota >= 5)
        {
            Console.WriteLine("Aprobado");
        }
    }
}
```

C:\WINDOWS\system32\cmd.exe

```
Introduzca nota:
6
Aprobado
Press any key to continue . . .
```

# Estructura If/else



```
internal class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Introduzca nota: ");
        int Nota = Convert.ToInt32(
            Console.ReadLine());
        if (Nota >= 5)
        {
            Console.WriteLine("Aprobado");
        }
        else
        {
            Console.WriteLine("Suspenso");
        }
    }
}
```

C:\WINDOWS\system32\cmd.exe

```
Introduzca nota:
6
Aprobado
Press any key to continue . . .
```

# Estructura If anidados



```
Console.WriteLine("Introduzca nota: ");
int Nota = Convert.ToInt32(Console.ReadLine());
if (Nota < 5)
{
    Console.WriteLine("Suspenso");
}
else
{
    if (Nota < 7)
    {
        Console.WriteLine("Aprobado");
    }
    else
    {
        if (Nota < 9)
        {
            Console.WriteLine("Notable");
        }
        else
        {
            Console.WriteLine("Sobresaliente");
        }
    }
}
```

# Operador condicional (?:)



```
internal class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Introduzca nota: ");
        int Nota = Convert.ToInt32(
            Console.ReadLine());
        Console.WriteLine(Nota >= 5 ?
            "Aprobado" : "Suspenso");

    }
}
```

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.exe'. The prompt is 'C:\>'. The user has entered 'C:\WINDOWS\system32\cmd.exe' and pressed Enter. The output shows 'Introduzca nota:' followed by the user input '6'. The program then outputs 'Aprobado' and 'Press any key to continue . . .'.

```
C:\> C:\WINDOWS\system32\cmd.exe
Introduzca nota:
6
Aprobado
Press any key to continue . . .
```

# Estructura while



```
int Contador = 0;
    double Nota, Total = 0, Media;
    while (Contador < 10)
    {
        Console.WriteLine("Nota " +(Contador + 1) + ": ");
        Nota = Convert.ToDouble(Console.ReadLine());
        Total += Nota;
        Contador++;
    }
    Media = Total / 10;
    Console.WriteLine("La media es " +Media);
```

A screenshot of a Windows command prompt window titled 'C:\WINDOWS\system32\cmd.exe'. The window shows the output of a C# program that calculates the average of 10 numbers. The output is as follows:

```
Nota 1:
5
Nota 2:
4
Nota 3:
6
Nota 4:
7
Nota 5:
8
Nota 6:
7
Nota 7:
4
Nota 8:
6
Nota 9:
7
Nota 10:
7
La media es 6.1
Press any key to continue . . .
```

# Operadores logicos



```
Console.WriteLine("Introduzca nota: ");  
int Nota = Convert.ToInt32(  
    Console.ReadLine());  
if (Nota < 0 || Nota > 10)  
{  
    Console.WriteLine("Nota incorrecta");  
}  
if (Nota >= 7 && Nota < 9)  
{  
    Console.WriteLine("Notable");  
}
```

C:\WINDOWS\system32\cmd.exe

```
Introduzca nota:  
11  
Nota incorrecta  
Press any key to continue . . .
```

C:\WINDOWS\system32\cmd.exe

```
Introduzca nota:  
8  
Notable  
Press any key to continue . .
```

# La estructura do / while



## La estructura do / while

```
do
{
    ...
} while (condición);
```

- *condición* puede ser cualquier expresión que devuelva un tipo bool (*true* ó *false*)
- Lo que va entre llaves se ejecuta SIEMPRE una primera vez.
- Lo que va entre llaves se volverá a ejecutar mientras que la *condición* sea *true*.
- Dentro de la llaves tiene que haber algo que modifique el resultado de la *condición* o sino se repetiría indefinidamente.
- TERMINA EN PUNTO Y COMA.

```
int Contador = 1;
do
{
    Console.WriteLine(Contador);
    Contador++;
} while (Contador <= 5);
```

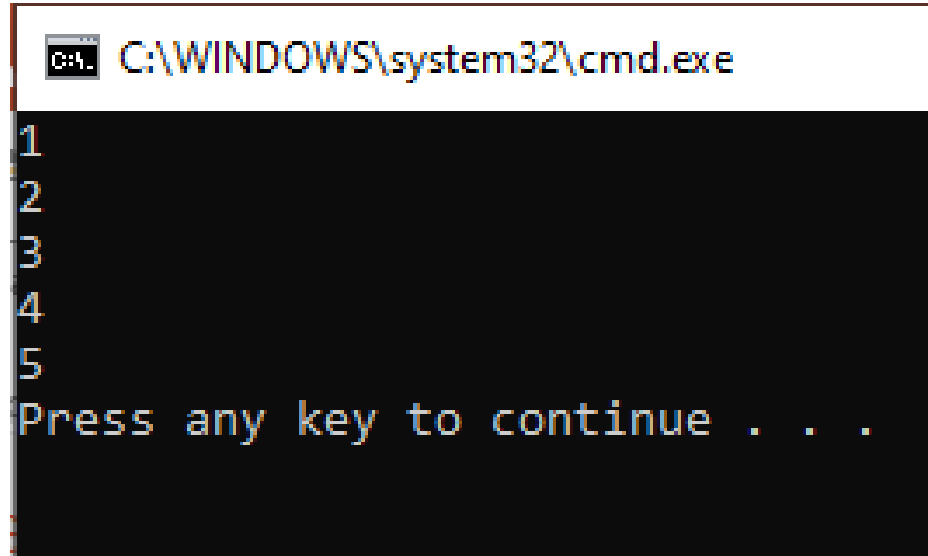
C:\WINDOWS\system32\cmd.exe

```
1
2
3
4
5
Press any key to continue . . .
```

# La estructura for



```
static void Main(string[] args)
{
    for (int a = 1; a <= 5; a++)
    {
        Console.WriteLine(a);
    }
}
```

A screenshot of a Windows command prompt window. The title bar shows the path "C:\WINDOWS\system32\cmd.exe". The command prompt displays the numbers 1 through 5 on separate lines, followed by the text "Press any key to continue . . .".

```
C:\WINDOWS\system32\cmd.exe
1
2
3
4
5
Press any key to continue . . .
```



# La estructura switch / case



```
Console.WriteLine("Introduzca día: ");
string Dia = Console.ReadLine();
Dia = Dia.ToUpper();
switch (Dia)
{
    case "SABADO":
        Console.WriteLine("Salir de copas.");
        break;
    case "DOMINGO":
        Console.WriteLine("Ir al cine.");
        break;
    default:
        Console.WriteLine("Ir a clase");
        break;
}
```

---

C:\WINDOWS\system32\cmd.exe

```
Introduzca día:
sabado
Salir de copas.
Press any key to continue . . .
```

# Bucles. break y continue



```
for (int a = 1; a <= 5; a++)  
{  
    if (a == 3)  
        break;  
    Console.WriteLine(a);  
}
```

C:\WINDOWS\system32\cmd.exe

```
1  
2  
Press any key to continue . . .
```

```
for (int a = 1; a <= 5; a++)  
{  
    if (a == 3)  
        continue;  
    Console.WriteLine(a);  
}
```

C:\WINDOWS\system32\cmd.exe

```
1  
2  
4  
5  
Press any key to continue . . .
```

## Bucles. break y continue

- *Bucle*: sentencias que se repiten.
- *break* y *continue* alteran el orden normal de ejecución de un bucle.
- *break* – Termina la ejecución del bucle en ese momento de forma definitiva.
- *continue* – Finaliza la ejecución del ciclo en ese momento y vuelve al principio del bucle para repetirlo.

# Clase random



EJEMPLO: Mostrar 20 números aleatorios del 1 al 6 colocando 5 en cada fila.

```
string resultado = "";
int valor;
Random creadorRandom = new Random();
for (int i = 1; i <= 20; i++)
{
    valor = creadorRandom.Next(1, 7);
    resultado += valor + " ";
    if (i % 5 == 0)
        resultado += "\n";
}
Console.WriteLine(resultado);
```

A screenshot of a console window with a black background and yellow text. It displays 20 random numbers from 1 to 6, arranged in 4 rows of 5 numbers each. The numbers are: Row 1: 1 5 1 5 2; Row 2: 2 6 1 5 6; Row 3: 4 3 3 5 4; Row 4: 3 4 6 6 4. Below the numbers, the text "Press any key to continue . . ." is displayed.

1 5 1 5 2  
2 6 1 5 6  
4 3 3 5 4  
3 4 6 6 4  
Press any key to continue . . .

# Vectores



```
string[] VECTOR = {  
    "LORENA",  
    "ESTEBAN",  
    "ARIEL"  
};  
byte K = 0;  
for (K = 0; K <= 2; K++)  
{  
    Console.WriteLine(VECTOR[K]);  
}  
Console.WriteLine();
```

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.exe'. The command prompt displays the output of the C# program: 'LORENA', 'ESTEBAN', and 'ARIEL' on three separate lines. Below the output, it says 'Press any key to continue . . .' with a red cursor on the first dot.

```
C:\WINDOWS\system32\cmd.exe  
LORENA  
ESTEBAN  
ARIEL  
Press any key to continue . . .
```

# Matrices



```
int l = 0;
int CAN = 0;
int POS = 0;
String cad;
Console.WriteLine("CUANTOS ELEMENTOS:");
cad = Console.ReadLine();
CAN = int.Parse(cad);
int[] VEC = new int[CAN + 1];
int[] NUEVO = new int[CAN];
// INGRESO
for (l = 1; l <= CAN; l++)
{
    Console.WriteLine("POSICIÓN {0} ==>", l);
    cad = Console.ReadLine();
    VEC[l] = int.Parse(cad);
}
do
{
    Console.WriteLine("POSICIÓN A ELIMINAR:");
    cad = Console.ReadLine();
    POS = int.Parse(cad);
} while (((POS < 1) | (POS > CAN)));
// PROCESO
// TRANSLADAMOS DATOS ANTES DE LA POSICIÓN
for (l = 1; l <= POS - 1; l++)
{
    NUEVO[l] = VEC[l];
}
// TRANSLADAMOS DATOS DESPUES DE LA POSICIÓN
for (l = POS + 1; l <= CAN; l++)
{
    NUEVO[l - 1] = VEC[l];
}
// SALIDA
Console.WriteLine();
Console.WriteLine("NUEVO ARREGLO");
for (l = 1; l <= CAN - 1; l++)
{
    Console.WriteLine(NUEVO[l]);
}
Console.WriteLine("Pulse una Tecla:");
Console.ReadLine();
```

A screenshot of a Windows command prompt window. The title bar shows the path "C:\WINDOWS\system32\cmd.exe". The command prompt displays the output of a C# program. The text is as follows:  
CUANTOS ELEMENTOS :4  
POSICIÓN 1 ==>12  
POSICIÓN 2 ==>3  
POSICIÓN 3 ==>5  
POSICIÓN 4 ==>7  
POSICIÓN A ELIMINAR:1  
  
NUEVO ARREGLO  
3  
5  
7  
Pulse una Tecla:

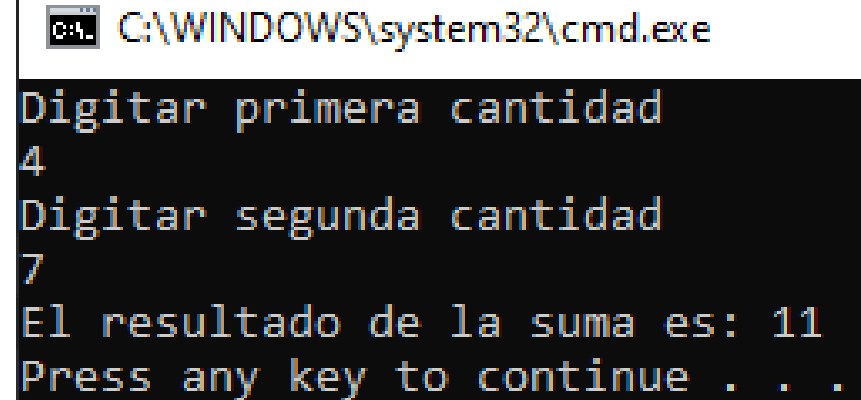
# Procedimientos-Funciones



```
static void Main(string[] args)
{
    suma(); // llamando al procedimiento llamado suma
}

static void suma()
{
    Double r, n1, n2;
    Console.WriteLine("Digitar primera cantidad");

    n1 = Double.Parse(Console.ReadLine());
    Console.WriteLine("Digitar segunda cantidad");
    n2 = Double.Parse(Console.ReadLine());
    r = n1 + n2;
    Console.WriteLine("El resultado de la suma es: " + r);
}
```

A screenshot of a Windows command prompt window. The title bar shows the path "C:\WINDOWS\system32\cmd.exe". The window contains the following text: "Digitar primera cantidad", "4", "Digitar segunda cantidad", "7", "El resultado de la suma es: 11", and "Press any key to continue . . .".

```
C:\WINDOWS\system32\cmd.exe
Digitar primera cantidad
4
Digitar segunda cantidad
7
El resultado de la suma es: 11
Press any key to continue . . .
```

# Procedimientos-Funciones



```
static void Main(string[] args)
{
    Double x, p, r;
    Console.WriteLine("Digitar la cantidad en dólares:");
    x = Double.Parse(Console.ReadLine());

    // llamadas a las funciones
    p = euros(x);
    Console.WriteLine("Los {0} dolares son {1} euros ", x, p);
    r = libras(x);
    Console.WriteLine("Los {0} dolares son {1} libras ", x, r);
    Console.WriteLine("\n");
    Console.WriteLine("-->Fin del programa");
    Console.ReadKey();
}

static Double euros(Double x)
{
    Double g;
    g = x* 1.15;
    return g;
}

static Double libras(Double x)
{
    Double v;
    v = x* 3.15;
    return v;
}
```

C:\WINDOWS\system32\cmd.exe

```
Digitar la cantidad en dólares:
80
Los 80 dolares son 92 euros
Los 80 dolares son 252 libras

-->Fin del programa
```



# GRACIAS

Línea de atención al ciudadano: 01 8000 910270  
Línea de atención al empresario: 01 8000 910682



[www.sena.edu.co](http://www.sena.edu.co)