

PROGRAMACIÓN EN C#

4. LOS CICLOS

1

4. LOS CICLOS

A. INSTRUCCIONES DE ITERACIÓN.

Un ciclo en programación es una forma de ejecutar una instrucción o un conjunto de ellas varias veces, dependiendo de una condición que será evaluada.

2

4. LOS CICLOS

B. LOS OPERADORES DE INCREMENTO Y DECREMENTO.

OPERADOR	SIGNO	PREFIJO	SUFIJO
Incremento	++	++i	i++
Decremento	--	--i	i--

```
Int i =10; //operando
Console.WriteLine("Valor inicial del operando: {0}", i); //10
//incremento como sufixo
Console.WriteLine("Valor con incremento: {0}", i++); //10
//incremento como prefijo
Console.WriteLine("Valor con incremento: {0}", ++i); //12
```



3

3

4. LOS CICLOS

B. LOS OPERADORES DE INCREMENTO Y DECREMENTO.

OPERADOR	SIGNO	PREFIJO	SUFIJO
Incremento	++	++i	i++
Decremento	--	--i	i--

```
Int in=20; //operando
Console.WriteLine("Valor inicial del operando: {0}", n); //20
//Decremento como sufixo
Console.WriteLine("Valor con decremento: {0}", n--); //20
//Decremento como prefijo
Console.WriteLine("Valor con decremento: {0}", --n); //18
```



4

4

4. LOS CICLOS

ASIGNACIÓN COMPUESTA.

ASIGNACIÓN COMPUESTA	FORMATO $x \text{ op } y$	EQUIVALENCIA $x = x \text{ op } y$
$+=$	$x += y$	$x = x + y$
$-=$	$x -= y$	$x = x - y$
$*=$	$x *= y$	$x = x * y$
$/=$	$x /= y$	$x = x / y$

$x = x + 5; \rightarrow x += 5;$
 $x = x - 3; \rightarrow x -= 3;$
 $x = x * 6; \rightarrow x *= 6;$
 $x = x / 5; \rightarrow x /= 5;$



5

5

4. LOS CICLOS

C. CUANDO USAR INSTRUCCIONES DE ITERACIÓN.

Código que nos hace la media de las notas de 3 alumnos.

```
//variables
double nota1, nota2, nota3, promedio;
Console.WriteLine("Nota del primer alumno: ");
nota1 = Convert.ToDouble(Console.ReadLine());
Console.WriteLine("Nota del segundo alumno: ");
nota2 = Convert.ToDouble(Console.ReadLine());
Console.WriteLine("Nota del tercer alumno: ");
nota3 = Convert.ToDouble(Console.ReadLine());
//Calculamos el promedio
promedio =(nota1 + nota2 + nota3)/3
Console.WriteLine("El promedio es: {0}", promedio);
```

¿Y SÍ TENEMOS
30 ALUMNOS?



6

6

4. LOS CICLOS

D. LA INSTRUCCIÓN DE ITERACIÓN FOR.

Se usa cuando necesitamos repetir una o varias instrucciones un número de veces que ya sabemos inicialmente.

```
for(//inicializador ; condición ; iterador)
{
    //bloque de código
}
```



7

7

4. LOS CICLOS

D. LA INSTRUCCIÓN DE ITERACIÓN FOR.

```
static void Main (string[] args)
{
    //variable de bucle (o de control)
    int i;
    for(i = 1 ; i <= 10 ; i++)
    {
        Console.WriteLine(i);
    }
}
```

→ Entra	For i++	→ Salido
i=1	1ra	i=2
i=2	2da	i=3
i=3	3ra	i=4
i=4	4ta	i=5
i=5	5ta	i=6
i=6	6ta	i=7
i=7	7ma	i=8
i=8	8va	i=9
i=9	9va	i=10
i=10	10ma	i=11
i=11	i<=10	sale



8

4. LOS CICLOS

CONTADOR

Un contador es una variable que suma a su valor inicial un conjunto de valores constantes.

```
int i, contador = 0;
for (i = 1; i <= 10; i++){
    contador++; //contador = contador + 1; //contador += 1;
    Console.WriteLine("Vuelta número {0}", contador);
}
Console.WriteLine("El número de vueltas es {0}", contador);
```



9

9

4. LOS CICLOS

ACUMULADOR

Acumulador es una variable que suma a su valor inicial un conjunto de valores variables.

```
int i, acumulador = 0, precio = 0;
for (i = 1; i <= 5; i++){
    Console.WriteLine("Da el precio del producto");
    precio = Convert.ToInt32(Console.ReadLine());
    acumulador = acumulador + precio; //acumulador += precio;
}
Console.WriteLine("Precio total es: {0}", acumulador);
```



10

10

TAREA #11

Modificar el programa de la media de las notas para que nos haga la media de las notas de un número variable de alumnos, utilizando un bucle for.



11

11

TAREA #12

Realiza un programa que calcule el factorial de un número cualquiera.

Factorial es la cantidad que resulta de la multiplicación de determinado número natural por todos los números naturales que le anteceden excluyendo el cero; se representa por $n!$

"el factorial de 4 es 24 (producto de $4 \times 3 \times 2 \times 1$)"



12

12

4. LOS CICLOS

E. INSTRUCCIÓN WHILE.

Se usa cuando necesitamos repetir una o varias instrucciones un número de veces que no sabemos inicialmente.

Se ejecuta 0 o un número de veces indefinido.

```
while (//condicion)
{
    //bloque de código
}
```



13

13

4. LOS CICLOS

D. INSTRUCCIÓN WHILE.

```
static void Maint (string[] args)
{
    byte numero = 1;
    while (numero <= 10)
    {
        Console.WriteLine(numero);
        numero++; //operador de incremento
    }
}
```



14

14

TAREA #13

Realiza un programa que pida a un usuario meter una contraseña y su confirmación. Mientras ambas contraseñas no coincidan se repetirá la petición de confirmación de contraseña. Cuando ambas sean iguales se mostrará un mensaje de "OK".

Usamos un bucle while y no for, porque no sabemos inicialmente el número de veces que el usuario se confundirá al confirmar la contraseña.



15

15

4. LOS CICLOS

F. INSTRUCCIÓN DO-WHILE.

Se usa cuando necesitamos repetir una o varias instrucciones un número de veces que no sabemos inicialmente.

Se ejecuta 1 o un número de veces indefinido.

```
do
{
    //bloque de código
} while (//condicion);
```



16

16

TAREA #14

Mejorar el programa que realiza las 4 operaciones básicas: suma, resta, multiplicación y división a partir de dos números, la operación deseada será la elegida por el usuario. Debe realizar los siguiente:

- 1 - Mostrar el menú.
- 2 – Hacer que el usuario escoja una opción.
- 3 – Si la opción no está dentro de las mostradas, volver a enseñar el menú hasta que escoja una opción válida
- 4 – Pedir 2 números con los que realizar la operaciones.
- 5 – Sumar, restar, multiplicar o dividir según sea el caso.
- 6 – Mostrar el resultado.



17

17

4. LOS CICLOS

INSTRUCCIONES DE ITERACIÓN ANIDADAS

```
int i, j;
    for(i = 1 ; i <= 10 ; i++)
    {
        for(j = 1 ; j <= 10 ; j++)
        {
            Console.WriteLine("{0} {1}",i,j);
        }
    }
```



18

18

TAREA #15

Realiza un programa que muestre las tablas de multiplicar del 1 al 10.



19

19

TAREA #16

Hacer un programa que calcule la potencia, ya sea negativa o positiva de cualquier exponente.



20

20

TAREA #17

Hacer un programa que calcule los números primos que existen entre el 1 y el 100

