

# CURSO DE PROGRAMACIÓN.NET

M.374.001.003



## ESTRUCTURAS DE CONTROL



## Ejercicios (estructuras iterativas)

---

1. Muestra por pantalla los números de 20 al 1 usando un bucle while, y después hazlo otra vez usando un bucle for.
2. Inicializa una variable entera a 0. Después, pregunta al usuario por un número y súmaselo a la variable inicial. Repite esto hasta que el usuario introduzca el número cero. Utiliza el bucle **do..while**.
3. Pide al usuario un número y calcula cuantas cifras tiene dicho número. Para ello puedes dividir el número entre 10 hasta que el resultado sea cero.
4. Pide al usuario un número y dibuja una línea con tantos asteriscos como el número introducido.
5. Muestra los números del 1 al 30 menos los divisibles entre 3 (resto da cero).
6. Muestra de 5 en 5, los números del 0 al 100.
7. Pregunta al usuario un número y muestra del 1 hasta ese número, en una misma línea, separados por comas (debes comprobar cuando es el último y no poner una coma en ese caso).

```
Dime un número: 7
1,2,3,4,5,6,7
```

8. Repite el ejercicio 2 pero con un bucle infinito (la condición nunca será falsa). En este caso tendrás que comprobar dentro del bucle cuando el usuario ha introducido un cero para salir (break). Si el usuario introduce un número negativo no lo sumes (usa continue para saltarte la suma).
9. Pide al usuario un número y dile los divisores de dicho número (resto da cero).

### Bucles anidados

10. Escribe en una misma línea los números del 1 al 9, 5 veces:

```
123456789123456789123456789123456789123456789
```

11. Escribe los números del 1 al 9, después del 1 al 8, del 1 al 7, y así sucesivamente en líneas diferentes.

```
123456789
12345678
1234567
123456
12345
1234
1234
123
```

12  
1

12. Pide al usuario un número y dibuja un cuadrado (mismo alto que ancho) con asteriscos según el número recibido.

```
Dime un número: 4
****
****
****
****
```

13. Pide al usuario un número y dibuja una escalera de bajada con asteriscos. La base será igual al número recibido.

```
Dime un número: 4
*
**
***
****
```

14. Haz lo mismo pero con una escalera de subida (para generar los huecos debes escribir el carácter de espacio).

```
Dime un número: 4
  *
  **
 ***
****
```

15. Pide al usuario un ancho y un alto y dibuja un rectángulo vacío.

```
Dime un ancho: 5
Dime un alto: 4
*****
*      *
*      *
*      *
*****
```

16. Pide al usuario un número que será la altura de una pirámide. Dibuja una pirámide con asteriscos con dicha altura

**Pista:** El ancho (base) de la pirámide será:  $(\text{altura} * 2 - 1)$ , y la posición del primer asterisco será el mismo que la altura si empiezas el bucle en 1 (o una menos si lo quieres empezar en 0). Crea 2 variables auxiliares con la posición del primer asterisco, una la irás decrementando y otra incrementando. Si la posición actual está entre esas 2 variables dibujas un asterisco, y si no, un espacio.

```
Dime la altura de la pirámide: 4
  *
 ***
*****
*****
```

17. Intenta hacer lo mismo pero con una pirámide hueca:

Dime la altura de la pirámide: 4

```
  *
 * *
*   *
*****
```

## 18. Dibuja ahora una pirámide invertida

Dime la altura de la pirámide: 4

```
*****
*****
***
*
```

## 19. Ahora dibuja un rombo (puedes dibujar una pirámide y posteriormente una pirámide invertida). Pide al usuario el ancho del rombo, que deberá ser un número impar (sigue pidiéndole un número hasta que introduzca uno impar).

Si quieres hacerlo con 2 pirámides, la altura de la primera sería  $(\text{ancho} + 1) / 2$

Dime la base del rombo: 5

```
  *
 ***
*****
 ***
  *
```

1. Realiza un programa de C# que pedirá una contraseña y la volverá a pedir hasta que se le introduzca la correcta que será 1111. Se debe hacer con while y con do while
2. Realiza un programa de C# que pedirá un número de forma repetitiva y dirá si el número introducido es par o impar. Dejará de pedir números cuando se introduzca un 0. Realiza este programa con while y con do while.
3. Pide un número y luego otro. Si el segundo es 0 lo tendrás que seguir pidiendo hasta que introduzca el usuario un número diferente de 0. Después de esto realizas la división y sacas el resultado por pantalla.
4. Crea un programa en C# que pida 2 números enteros y responda cuantos de ellos son positivos. No utilizar un contador sino && || !
5. Crea un programa en C# que pida 2 números enteros y responda cuantos de ellos son positivos. Utilizando un contador
6. Crea un programa en C# que pida números hasta que se introduzca un 0 y al final nos diga cuántos de ellos han sido positivos y cuantos negativos.
7. Crea un programa en C# que pida el login y el password al usuario hasta que introduzca los datos correctos: login 1 y contraseña 1234
8. Realiza un programa en C# que pida un usuario y contraseña de forma repetitiva hasta que se introduzca la correcta que será usuario:1212 y contraseña:3434. Se le darán al usuario sólo 3 intentos.
9. Realiza un programa en C# que a pida un número al usuario y cuente los números pares hasta ese número, contando de 1 en 1 y contando de 2 en 2.
10. Realiza un programa en C# que a pida un número al usuario y cuente los números pares e impares hasta ese número.
11. Realiza un programa en C# que a saque por pantalla los primeros n números cuadrados, siendo n un número proporcionado por el usuario. Por ejemplo si se introduce el número 4 el programa sacaría: 1, 4, 9, 16
12. Crea un programa en C# que responda si un número introducido por el usuario es primo o no.
13. Crea un programa en C# que nos diga cuántos números primos hay hasta un número suministrado por el usuario.

Por ejemplo:

Introduzca un número: 10

Los primos hasta 10 son: 1,3,5,7,

14. Realiza un programa en C# que imprima por pantalla un triángulo de las alturas que le suministre el usuario y que quede de la siguiente forma.

```
####  
###  
##  
#
```

15. Realiza un programa en C# que imprima por pantalla un triángulo invertido de las alturas que le suministre el usuario y que quede de la siguiente forma.

```
####  
###  
##  
#
```

16. Crea un programa que le pida un número al usuario y calcule su factorial. El factorial de un número se calcula multiplicando todos los números que van desde el 1 hasta dicho número. Por ejemplo, el factorial de 6 se calcularía multiplicando  $6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$
17. Crea un programa en C# que vaya pidiendo cantidades y precios hasta que se introduzca un 0 o en cantidad o en precio. Nos tendrá que ir mostrando el total de cada artículo y al final nos tendrá que mostrar el total final.

Introduzca cantidad: 5

Introduzca precio: 10

Total: 50

Introduzca cantidad: 7

Introduzca precio: 2

Total: 14

Introduzca cantidad: 2

Introduzca precio: 0

Total Final: 64

18. Crea un programa en C# que nos diga cuántos números primos hay entre 2 números suministrados por el usuario.
19. Crea un programa en C# que nos diga los divisores comunes de 2 números suministrados por el usuario
20. Realiza un programa en C# que pida números al usuario y no pare de pedirle hasta que el usuario introduzca un 0. Debe contar los números que se han introducido menos el 0 del final que no lo tiene que contar.
21. Realiza un programa en C# que pida números al usuario hasta que este introduzca un 0. Deberá contar los que son múltiplos de 5 y el 0 final no debe ser contado.

22. Realiza un programa en C# que pida al usuario dos números y saque todos los números pares entre esos 2 números:
- 1.- Utilizando un for que salte de 2 en 2
  - 2.- Utilizando un for que salte de 1 en 1 y usando continue
  - 3.- Utilizando un bucle infinito y usando break y continue
23. Crea un programa en C# que pregunte al usuario 2 números y responda, usando el operador condicional, lo siguiente:
- Si el primer número es positivo
  - Si el segundo número es positivo
  - Si los dos son positivos
  - Cual de ellos es el más pequeño
24. Crea un programa en C# que enseñe los múltiplos de 3 entre -30 y -10 que no son múltiplos de 7.
- Usando "for" contando de 1 en 1
  - Usando "for" contando de 3 en 3
  - Usando "while"
  - Usando "do while"
25. Crea un programa que pida al usuario 2 números enteros "n1" y "n2" y asigne a la variable "numerosPares" los valores 0, 1 o 2, dependiendo de cuantos de ellos sean pares.