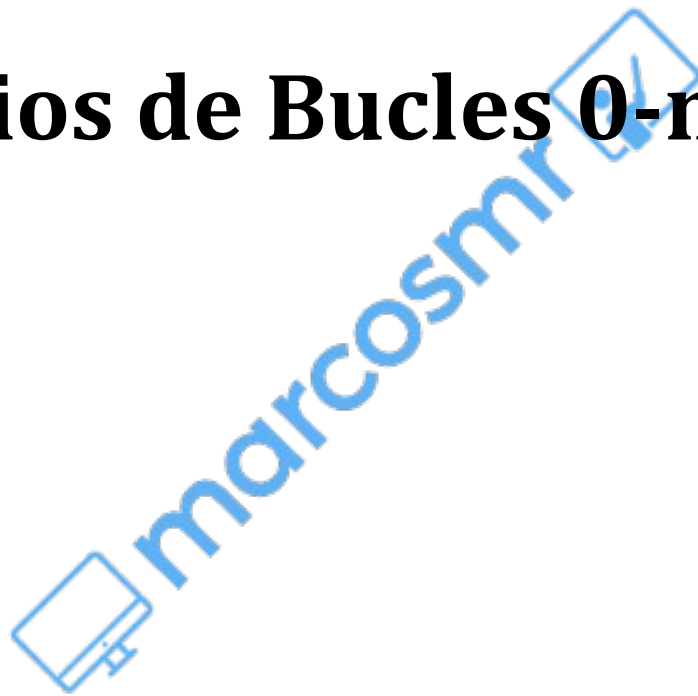


# Ejercicios de Bucles 0-n



## Relación de ejercicios

**Ejercicio 1.** Implementa un programa que saque por pantalla qué día y hora mientras que el usuario no escriba la palabra clave “Salir”. El usuario puede escribir dicha palabra combinando mayúsculas y minúsculas. Los mensajes por pantalla deben ser amigables para el usuario.

**Ejercicio 2.** Crea el siguiente menú e indica qué opción ha sido elegida por el usuario. Recuerda, de aquí en adelante, validar que los datos introducidos por el usuario son coherentes con nuestra lógica de negocio. Por ejemplo, el usuario únicamente debe poder introducir valores que estén entre 0 y 3 en el siguiente menú.

Menú:

```
GESTIÓN DEL CENTRO EDUCATIVO
*****
1. Gestión de alumnos
2. Gestión de docentes
3. Gestión de personal administrativo
0. SALIR
```

Ejemplo de salida tras la introducción de una opción por parte del usuario.

```
Indique el número de la opción deseada: 2
Ha elegido la opción: 2
```

**Ejercicio 3.** Debes implementar un programa cuya salida por pantalla sea la de la imagen. Para ello, tendrás que pedir por teclado los datos al usuario teniendo en cuenta lo siguiente:

- La temperatura mínima en Groenlandia es de -50 °C y la máxima es de 15 °C.
- La altura máxima del Everest es de 8848 m
- El valor de una casa está entre 100.000€ y 400.000€ en una determinada zona de Madrid.

Ejemplo de petición de entrada de datos:

```
¿Cuál es la temperatura actual de Groenlandia?
¿Hasta qué altitud has subido al Everest?
¿Cuánto te valió la casa?
```

Ejemplo de salida por pantalla tras la petición de datos al usuario:

```
La temperatura actual de Groenlandia es de 13 °C
Has subido hasta los 1500 m en el Everest
Tu casa te ha valido 100000 euros
```

**Ejercicio 4.** Implementa un juego de adivinar un número. El juego es para dos jugadores. Uno de ellos debe introducir un número (sin que lo vea el otro jugador). Posteriormente, el segundo jugador debe adivinar el número introducido por el primero a través de las pistas del programa. Cada vez que el segundo usuario introduzca un número para intentar acertarlo, el programa debe decir si el número a acertar es mayor o menor que el introducido.

**Ejercicio 5.** Modifica el programa anterior para que informe al jugador 2 del número de intentos que lleva y del número de intentos que ha tardado en adivinar el número.

**Ejercicio 6.** Modifica el programa anterior para que el número de intentos sea limitado. Una vez que el usuario (jugador 2) ha superado el número de intentos, se debe acabar el juego e indicarle cuál era el número que debía adivinar.

**Ejercicio 7.** Implementa un programa calcule el número de cifras que tiene un número entero positivo (Pista: se puede hacer dividiendo varias veces entre 10).

**Ejercicio 8.** Implementar un programa que lea un número positivo y compruebe si es un número perfecto. Un número es perfecto aquel en el que la suma de sus divisores (exceptuando el propio número) es igual al número. Por ejemplo el número 6 tiene de divisores: 1, 2 y 3. Si sumamos  $1+2+3 = 6$ , por tanto, el número 6 es perfecto. En cambio el número 24 tiene por divisores: 1, 2, 3, 4, 6, 8 y 12. Si sumamos sus divisores nos dará 36 y, por tanto, el número 24 no es perfecto.

**Ejercicio 9.** Implementa un programa que saque los n primeros números de la sucesión de Fibonacci. Los dos primeros números son el 1 y, para sacar el siguiente número (hijo de Fibonacci) hay que sumar los dos anteriores. El comienzo de la secuencia es el siguiente: 1, 1, 2, 3, 5, 8, 13, 21, 34,...

**Ejercicio 10.** Implementa un programa que saque los números de la sucesión de Fibonacci que sean menores o iguales a un determinado valor introducido por el usuario. Por ejemplo, si el usuario introduce el número 35, se debe mostrar la sucesión : 1, 1, 2, 3, 5, 8, 13, 21 y 34.

**Ejercicio 11.** Modifica el programa anterior para que si el usuario introduce un valor negativo, se le siga pidiendo que introduzca un valor mayor a 0 hasta que lo haga y poder calcular, de esa forma, la sucesión de Fibonacci deseada.