



## SESIÓN 3: Estructuras de control: Repetitivas, Iterativas o Bucles

### Objetivos

- Saber construir y utilizar una estructura repetitiva **while, for, do-while**.
- Saber elegir el tipo de bucle adecuado para un problema determinado.
- Ser capaces de construir bucles anidados.
- Evitar la construcción de bucles infinitos.

**Nota importante:** Siga el esquema de nombrado de paquetes que se indicó en la sesión 01 es decir: **org.ip.sesion03**. En ese paquete se crearán todos los programas que se proponen en la sesión dándoles un nombre al programa y que se indica en cada ejercicio **entre paréntesis y en negrita**.

Al final de la sesión, el alumno deberá cargar el trabajo realizado a su repositorio indicando la clave correspondiente a la sesión a través del mensaje del *Commit*.

### Ejercicios propuestos

1. Desarrolle un programa que lea desde teclado un número indeterminado de enteros (separados por espacios en blanco) y muestre cuantos son positivos, cuantos negativos, número total de valores leídos, suma de los valores positivos, suma de los valores negativos, media de todos los valores leídos (no se contabiliza el 0 para la media). El programa termina cuando la entrada es 0, denominado *valor centinela*. Prueba con una estructura *while* y con *for*. (**BucleCentinela**).

**Nota:** Ejemplo de uso del Scanner en Java

```
import java.util.Scanner;
class EjemploLectura {
    public static void main(String arg[]){
        String variableString;
        // Se declara e inicializa una instancia de la clase Scanner
        Scanner entrada=new Scanner(System.in);
        System.out.print("Ingrese un texto: ");
        variableString = entrada.next();
        // Para leer valores enteros, reales, etc. desde teclado, hay que declarar
        // las variables apropiadas y luego llamar a nextInt(), nextFloat(), etc.
        System.out.println("Texto introducido: " + variableString);
    }
}
```

Ejemplo de ejecución:



```
Introduce valores enteros, el programa termina si la entrada es 0
1 2 -1 3 0
El numero de positivos es 3
El numero de negativos es 1
El numero total de valores leidos es 4
La suma de positivos es 6
La suma de negativos es -1
La media de los valores es 1.25
```

2. Desarrolle un programa en Java que permita obtener el MCD (máximo común divisor) de dos enteros positivos haciendo uso del algoritmo de Euclides. **(Euclides)**. Para ello se divide el *dividendo* (x) entre el *divisor* (y) y se obtiene un *cociente* ( $q_1$ ) y un *resto* ( $r_1$ ) si dicho resto es distinto de 0 se divide nuevamente el anterior *divisor* entre el *resto* obtenido. Se continua el proceso hasta obtener un resto 0, el último divisor es el MCD.

$$\begin{array}{ccccccc}
 x \mid y & y \mid r_1 & r_1 \mid r_2 & \dots & r_{n-2} \mid r_{n-1} \\
 r_1 \quad q_1 & r_2 \quad q_2 & r_3 \quad q_3 & & r_n \quad q_n \\
 r_1 \neq 0 & r_2 \neq 0 & r_3 \neq 0 & \dots & r_n = 0 \Rightarrow r_{n-1} \text{ es el mcd}
 \end{array}$$

Ejemplo de ejecución:



```

Introduce el primer valor entero positivo
-3
Introduce el primer valor entero positivo
25
Introduce el segundo valor entero positivo
5
El MCD de 25 y 5 es 5

```

3. Desarrolle un programa en Java que muestre los  $i$  primeros términos de la serie de Fibonacci **(Fibonacci)**. La expresión de la serie de Fibonacci es la siguiente:

$$\begin{aligned}
 f_0 &= 0 \\
 f_1 &= 1 \\
 f_i &= f_{i-1} + f_{i-2} \quad \text{para } i \geq 2
 \end{aligned}$$

Ejemplo:  $f_0 = 0; f_1 = 1, f_2 = 1, f_3 = 2, f_4 = 3, f_5 = 5, f_6 = 8, f_7 = 13, f_8 = 21, f_9 = 34, f_{10} = 55, \dots$

Ejemplo de ejecución:



```


Hasta que termino de la serie de Fibonacci quieres mostrar?
10
0 1 1 2 3 5 8 13 21 34 55

```

## Trabajo autónomo

4. Desarrolle un programa en Java que pida por teclado un número comprendido entre 1 y 10. Asegúrese de que efectivamente el número introducido desde teclado se encuentra entre el 1 y el 10, caso de no ser así, se debe volver a pedir dicho número (uso del bucle **do-while**). Una vez introducido correctamente dicho número, se debe mostrar por pantalla la tabla de multiplicar de dicho número. (**TablaMultiplicar**)

Ejemplo de ejecución:




```
Introduzca un numero (de 1 a 10): 20
Introduzca un numero (de 1 a 10): -3
Introduzca un numero (de 1 a 10): 0
Introduzca un numero (de 1 a 10): 5

Tabla del 5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

5. Desarrolle un programa en Java (**Primo**) que muestre si un número entero positivo ( $>1$ ) es primo o no. En el archivo EjerciciosResueltos\_Tema01.pdf, visto en clase y cuyos códigos están en el repositorio de la asignatura. En el ejemplo número 12 muestra dos maneras de hacerlo y deja planteada una tercera (Segunda mejora). Resuelve esta última sobre la primera mejora, implementando las dos siguientes restricciones: (1) comprobando si es divisible entre 2, de no serlo no se comprobarían los divisores pares y, (2) se llegaría probando divisores hasta  $\sqrt{dato}$  (si *dato* es el número a comprobar si es primo o no), si al llegar hasta ese valor no hemos encontrado ningún divisor, no es necesario seguir comprobando. Prueba la ejecución para números primos y no primos.

Ejemplo de ejecución:



```
Introduce un numero (>1) para saber si es primo: 0
Introduce un numero (>1) para saber si es primo: 1
Introduce un numero (>1) para saber si es primo: 9199
ES PRIMO
```

6. Dado un número mayor que 0 y no superior a 10, *lado*, implemente un programa en Java que muestre por consola un *triángulo rectángulo de números*, tal y como se muestra en la siguiente figura. No escribir simplemente las cadenas de dígitos, una a continuación de otra. Conseguir una expresión para generar los dígitos correspondientes para cada línea, variando el valor del lado y haciendo uso de bucles. (**TrianguloRectanguloNumeros**)

Ejemplo de ejecución:



Triangulo rectangulo de numeros para un valor de lado = 10

```

          0
        1 0
      2 1 0
    3 2 1 0
  4 3 2 1 0
5 4 3 2 1 0
6 5 4 3 2 1 0
7 6 5 4 3 2 1 0
8 7 6 5 4 3 2 1 0
9 8 7 6 5 4 3 2 1 0
```

7. Dados un número mayor que cero, *lado*. Desarrolle un programa en Java que muestre por consola un cuadrado de asteriscos tal y como se indica en la siguiente figura. (**CuadradoAsteriscos**)

Ejemplo de ejecución:



Cuadrado de asteriscos de lado 7

```

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

8. Dado un número mayor que cero, *lado*, tal que  $3 < lado < 15$ . Desarrolle un programa en Java que muestre por consola un triángulo rectángulo de asteriscos tal y como se indica en la siguiente figura. (**TrianguloRectanguloAsteriscos**)

Ejemplo de ejecución:



Triangulo rectangulo de asteriscos de lado 7

```

* * * * *
* * * *
* * * *
* * *
* *
*

```