

# UD2Tarea4 - Do While

Andrés Pérez Guardiola - 1st DAW

**1. Nombra las diferencias entre los siguientes dos tipos de bucle. ¿Obtenemos el mismo resultado? En qué caso, ¿sería mejor usar un bucle while frente a uno do-while?**

a.

```
public static void main(String[] args) {  
    int contador = 1;  
  
    do {  
        System.out.println(contador);  
        contador++;  
    } while (contador <= 5);  
}
```

b.

```
public static void main(String[] args) {  
    int contador = 1;  
  
    while (contador <= 5)  
    {  
        System.out.println(contador);  
        contador++;  
    }  
}
```

Ahora mismo los dos contadores realizan lo mismo y se obtendrá la misma salida en ambos casos. Aunque en funcionamiento existen algunas diferencias.

En do-while no evalúa la condición `contador <= 5` en la primera iteración, sino que primero realiza las sentencias dentro del bloque `do`, y luego evalúa la condición.

En ambos casos la salida es la misma.

```
"C:\Program Files\Java\jdk-22\bin\  
1  
2  
3  
4  
5  
  
Process finished with exit code 0
```

Sobre si es mejor usar un bucle While frente un bucle do-while, creo que la respuesta es que depende de lo que se pretende hacer. Es posible que sea necesario realizar una tarea como mínimo una vez y, por tanto, do-while cumpliría con ese propósito. Pero es posible que se pretenda que la tarea se realice sólo si se cumple la condición. En ese caso, solo nos serviría el bucle while.

**2. ¿Son correctos los siguientes fragmentos de código? Explica si produce algún error, qué error produce y cómo solucionarlo. (imagina que las variables existen).**

**a. `do { } while();`**

No, no se puede ejecutar una sentencia do-while sin indicarle mínimo una condición.

**b. `do { } while(a<=b);`**

Si, pero sería un potencial bucle infinito al no cambiar el valor que vale a dentro del cuerpo de do.

**c. `do { } while(1);`**

La condición debe de ser un valor booleano. 1 no es un valor booleano, true si lo sería.

**d. `do while(m==n);`**

No sería correcto. do debe ir precedido de {} para indicar el bloque de sentencias que se ejecutarán dentro del mismo.

**3. Desarrolla un programa utilizando un bucle do-while que imprima “Hola caracola” 5 veces.**

```
public class Hola
{
    public static void main(String[] args) {
        int contador = 1;
        int limite = 5;
        String mensaje = "Hola caracola";

        do {
            System.out.println(mensaje);
            contador++;
        } while (contador <= limite);

    }
}
```

```
"C:\Program Files\Java\jdk-22\bin\
Hola caracola
Hola caracola
Hola caracola
Hola caracola
Hola caracola

Process finished with exit code 0
```

#### 4. Escribe las trazas del ejercicio anterior.

La primera iteración ya empieza con el contador = 2, porque en la primera ejecución el do incrementa el valor del contador a 2 antes de la primera evaluación de la condición.

Iteración	Variables	Condición (contador <= limite)	Acción
1ª	contador = 2 limite = 5	true	incrementa (3)
2ª	contador = 3 limite = 5	true	incrementa (4)
3ª	contador = 4 limite = 5	true	incrementa (5)
4ª	contador = 5 limite = 5	true	incrementa (6)
5ª	contador = 6 limite = 5	false	

## 5. A partir del siguiente código, responde a las preguntas.

```
public static void main(String[] args) {  
    int contador = 1;  
    do {  
        System.out.println(contador);  
        if(contador % 4 == 0)  
        {  
            System.out.println("Múltiplo de 4 encontrado. Salir!");  
            contador = 10;  
        }  
        contador++;  
    } while (contador <= 10);  
}
```

a) ¿Qué hace el código?

El código va incrementando el valor de contador en 1 hasta encontrar un número que sea múltiplo de 4 o alcanza la condición límite que sea mayor que 10. En el momento que encuentra un múltiplo de 4, este bucle se detiene.

b) ¿Cuántas veces se ejecutará las instrucciones dentro del bloque do?

Se ejecutará cuatro veces el bloque, ya que en la cuarta iteración se cumple la condición del if y el contador pasa a valer 10 y luego 11.

c) ¿Cuántas veces se ejecutará la condición?

Se ejecutará 4 veces, una por cada vez que se ejecuta el bloque do. Si fuera un bucle while en vez de uno tipo do, la condición se ejecutaría una vez más.

d) ¿Cuántas veces se ejecutará la condición como verdadera?

Se ejecutará 3 veces como verdadera en contador = 2, 3 y 4, y falsa con contador = a 11. Contador = 1 no se llega a evaluar ya que el incremento sucede en el do por primera vez pasando su valor a 2.

e) Crea una tabla con las trazas del programa.

Iteración	Variables	Condición (contador <= 10)	Acción (Se ejecuta antes de la condición)
1ª	contador = 1	true	incrementa (2)

2ª	contador = 2	true	incrementa (3)
3ª	contador = 3	true	incrementa (4)
4ª	contador = 4	false	incrementa a (11)

- f) ¿Qué conseguimos con contador = 10; en el programa?

Conseguimos romper el bucle haciendo que la condición del mismo deje de cumplirse.

- g) ¿Qué imprimiría el programa si cambiamos la primera sentencia por int contador = 0;?

Cero daría un resto cero al dividirlo por 4, pero esto es un falso resultado ya que cero no es múltiplo de 4. Este bucle sólo tiene sentido para valores enteros mayores de cero.

6. Escribe un programa que le pida al usuario introducir por teclado un número, y le muestre la tabla de multiplicar de dicho número.

```
import java.util.Scanner;

public class Main
{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Introduce un número: ");
        int numero = sc.nextInt();

        int contador = 1;

        do {
            System.out.println(numero + " x " + contador + " = " + numero * contador);
            contador = contador + 1;
        } while (contador <= 10);
    }
}
```

Resultado:

```
Introduce un número: 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

Process finished with exit code 0
```



**7. Escribe un programa que pida al usuario introducir un número, y muestre los números de forma descendente. Ejemplo: usuario inserta el 4. Output: 4 3 2 1.**

```
import java.util.Scanner;

public class Main
{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Introduce un número: ");
        int contador = sc.nextInt();

        do {
            System.out.println(contador);
            contador = contador - 1;
        } while (contador > 0);
    }
}
```

Resultado:

```
Introduce un número: 6
6
5
4
3
2
1

Process finished with exit code 0
```