

PROMETEO

Estructuras de Control

Índice

01

Bucles

Introducción

while

do-while

for

02

Salida anticipada

break

continue

03

Bucles anidados

Descripción



1

Bucles

PROMETEO

01

Introducción

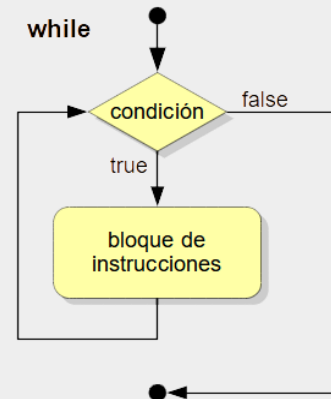
- Utilizaremos la estructura de **bucle** cuando necesitemos repetir un conjunto de instrucciones **siempre que se cumpla** una condición.
- El uso de esta estructura de iteración **simplifica** el código y **reduce el código duplicado**.
- Tipos de bucles en java:
 - while
 - do-while
 - for

01

while

- Antes de ejecutar el bloque de instrucciones, **se evalúa la condición** que rige la iteración. Si la evaluación resulta en verdadero, se ejecutan las instrucciones del bucle, en caso contrario no se ejecutan.
- Cada vez que se llega al final de las instrucciones del bucle, se vuelve a evaluar la condición inicial para determinar si seguir con las iteraciones o no.

¡Imprescindible garantizar que la condición llegará a ser false en algún momento!!



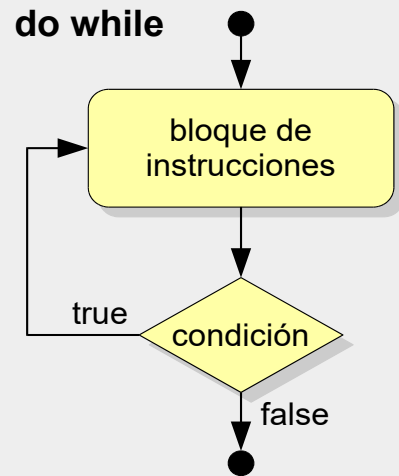
```
while (condición) {  
    bloque de instrucciones;  
}
```

01

do-while

- Se ejecuta el bloque de instrucciones y al finalizar la primera iteración **se evalúa la condición** que determina si continuar en el bucle o no.
- La diferencia con el 'while' es que en esta ocasión **se ejecuta siempre al menos una vez** el bloque de instrucciones.
- Respecto a la sintaxis, hay que poner ; tras la condición.

¡Imprescindible garantizar que la condición llegará a ser false en algún momento!!

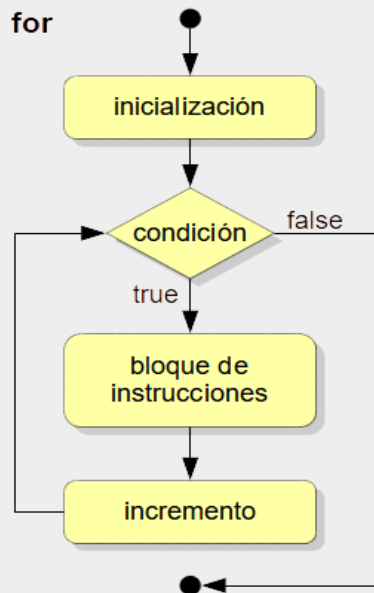


```
do {  
    bloque de instrucciones;  
} while (condición);
```

01 for

- Permite **controlar el número de iteraciones** con un contador
- La **inicialización** da valor a las variables que se utilizan para controlar el bucle (contador).
- En el **incremento** se modifican las variables 'contador'
- Las variables 'contador' se declaran y solo se utilizan en el 'for'

```
for (inicialización ; condición ; incremento){  
    bloque de instrucciones;  
}
```





2

Salidas
anticipadas

PROMETEO

02

break y continue

- En ocasiones interesa **finalizar un bucle antes de lo previsto** inicialmente. Existen dos opciones que permiten modificar el flujo de iteración:

break : interrumpe completamente la ejecución del bucle

```
i=1;
while (i <=10){
    System.out.println("La i vale " + i);
    if (i ==2= {
        break; // sale del bucle
    }
    i++;
}
```

continue : detiene la iteración actual, pero continúa con la siguiente, el bucle puede continuar.

```
i=0;
while (i <10){
    i++;
    if (i % 2==0) {
        continue; // salta a siguiente iteración
    }
    System.out.println("La i vale " + i);
}
```



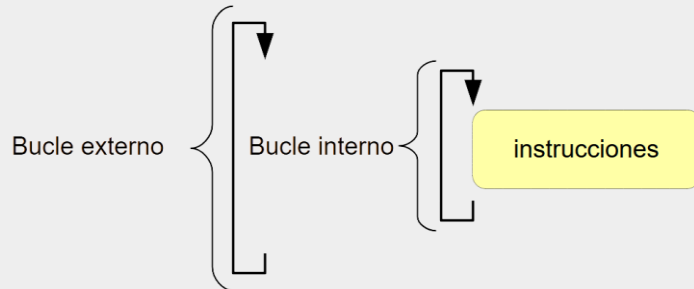
3

Bucles anidados

03

Bucles anidados

- La anidación de bucles consiste en incluir un bucle dentro de otro (entre las llaves)
- En el bloque de instrucciones de un bucle se puede incluir cualquier tipo de instrucciones , también otros bucles.
- En el anidamiento de bucles se pueden distinguir :
 - bucle **dependiente**: El bucle interno depende del externo
 - bucle **independiente** : El bucle interno no depende del externo





PROMETEO