



TÉCNICAS DE PROGRAMACIÓN (JAVA)

Presentación 07 – Estructuras repetitivas





Contenido:

- while
- do-while
- for
- break
- continue



Estructuras de Control repetitivo

- Permiten la ejecución de un bloque determinado de código.
- Son para ejecutar una determinada cantidad de veces una opción de código de programa.
- Se puede ejecutar una cantidad específica de veces o hasta que se cumpla una condición determinada.



Estructura while

- Permite ejecutar un bloque de código determinado en forma repetitiva “mientras” (while, en inglés) se cumpla la condición.
- La condición se evalúa al inicio del ciclo.
- Si la condición es válida, ingresa en el ciclo y ejecuta el código que pusimos en el mismo.
- Cuando deja de cumplirse la condición, el programa sale del while y continúa con el resto del programa.
- Se puede no entrar al bucle, dependiendo de la expresión booleana.

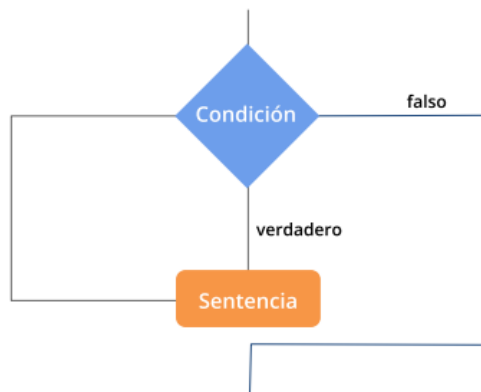


Estructura while

- El bucle while evalúa la condición booleana, si la misma es verdadera, se ejecutan las declaraciones dentro del bucle.
- En cada ciclo del bucle se evalúa la condición nuevamente, si la condición es falsa el ciclo termina.

Forma de uso:

```
while(expresionBooleana)
{
    sentencias;
}
```





Estructura do-while

- Esta sentencia es igual a la anterior, con la diferencia que se evalúa al final.
- **La condición se evalúa al final del ciclo.**
- El bloque de código se ejecuta al menos una vez.
- Cuando deja de cumplirse la condición, el programa sale del do-while y continúa con el resto del programa.
- Tiene como particularidad que el control está al final del bucle.
- **Se debe entrar al bucle al menos una vez.**



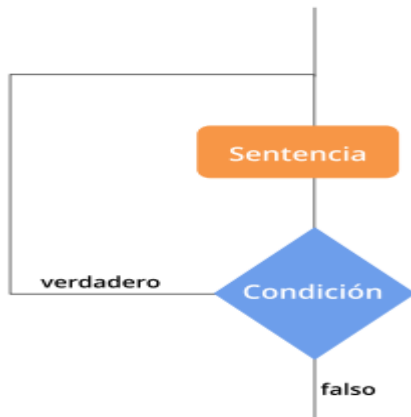
Estructura do-while

La declaración se ejecuta una vez y luego se evalúa la expresión. Si la expresión es verdadera, se repite la declaración hasta que la expresión se convierte en falsa.

La única diferencia práctica entre el while y do-while es que en el do-while su declaración siempre se ejecutará al menos una vez.

Forma de uso:

```
do  
{  
    sentencias;  
}  
while(expresionBooleana);
```





Estructura for

La sentencia “for” realiza un bucle un determinado número de veces.

La sentencia inicialización se ejecuta al comienzo del for. El incremental, después de las sentencias. La expresión Booleana se evalúa al comienzo de cada iteración.

Está compuesta por tres partes:

- **Inicialización** de la variable que utilizaremos en la condición (incondicionalmente se ejecuta solo una vez al principio del ciclo).
- Indicar la **condición** por la cual finalizará el bucle (se evalúa esta expresión al comienzo de cada iteración).
- **Modificación** de la variable, para no entrar en un bucle infinito (se ejecuta al final de cada iteración).



Estructura for

Forma de uso:

```
for (inicializacion; expresionBooleana;postAccion)
{
    sentencias;
}
```

