

SENTENCIAS REPETITIVAS O BUCLES

Los bucles, iteraciones o sentencias repetitivas modifican el flujo secuencial de un programa permitiendo la ejecución reiterada de una sentencia o sentencias. En Java hay tres tipos diferentes de bucles: `for`, `while` y `do-while`.

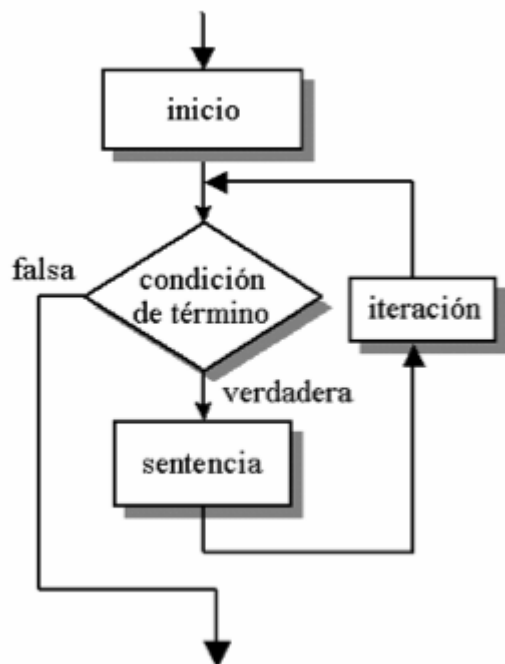
SENTENCIA FOR

Es un bucle o sentencia repetitiva que

- Ejecuta la sentencia de inicio.
- verifica la expresión booleana del término.

Sintaxis:

- Si es cierta, ejecuta la sentencia entre llaves y la sentencia de iteración para volver a verificar la expresión booleana de término.
- si es falsa, sale del bucle.



Sintaxis:

```
for (inicio; termino; iteracion)
```

```
sentencia;
```

O si se desean repetir varias sentencias:

```
for (inicio; termino; iteracion) {  
  
    sentencia_1;  
  
    sentencia_2;  
  
    sentencia_n;  
  
}
```

Las llaves sólo son necesarias si se quieren repetir varias sentencias, aunque se recomienda su uso porque facilita la lectura del código fuente y ayuda a evitar errores al modificarlo. Habitualmente, en la expresión lógica de término se verifica que la variable de control alcance un uso determinado. Por ejemplo:

```
for (i = valor_inicial; i <= valor_final; i++) {  
  
    sentencia;  
  
}
```

Es completamente legal en Java declarar una variable dentro de la cabecera de un bucle `for`. De esta forma la variable (local) sólo tiene ámbito dentro del bucle. Ejemplo sencillo:

```
System.out.println("Tabla de multiplicar del 5");  
  
for (int i =0 ; i <= 10; i++) {  
  
    System.out.println(5 + " * " + i + " = " + 5 * i);  
  
}
```

Salida por pantalla al ejecutar el código anterior:

```
5 * 0 = 0  
5 * 1 = 5  
5 * 2 = 10  
5 * 3 = 15
```

5 * 4 = 20

5 * 5 = 25

5 * 6 = 30

5 * 7 = 35

5 * 8 = 40

5 * 9 = 45

5 * 10 = 50

6.2 SENTENCIA WHILE

Es un bucle o sentencia repetitiva con una condicion al principio. Se ejecuta una sentencia mientras sea cierta una condición. La sentencia puede que no se ejecute ni una sola vez.

Sintaxis:

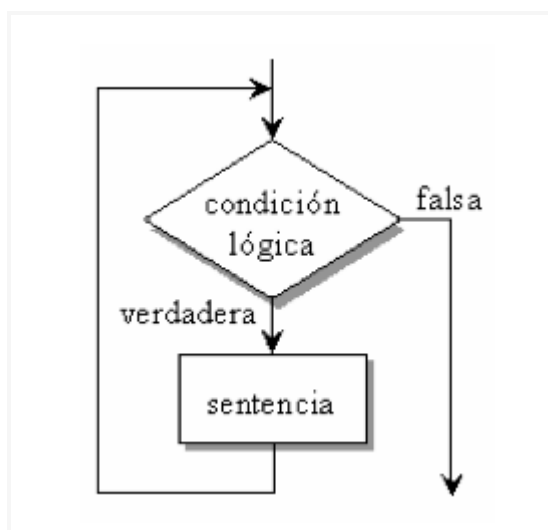
```
[inicializacion;]
```

```
while (expresionLogica) {
```

```
    sentencias;
```

```
    [iteracion;]
```

```
}
```



Flujograma de la sentencia while

Se recuerda que el vector `args` contiene todos los parámetros o argumentos indicados en la línea de comandos. El primer elemento de este vector es `args[0]`. El tamaño del vector puede determinarse añadiendo `.length` a su identificador. Como el índice del primer elemento del vector es 0, si el tamaño del vector es n , entonces el último elemento del vector tiene índice $n-1$. En el ejemplo anterior de ejecución del programa `eco`, `args[0]` toma como valor la cadena "Esto", `args[1]` vale "es", `args[2]` vale "una" y `args[3]` vale "prueba".

SENTENCIA DO-WHILE

Es un bucle o sentencia repetitiva con una condicion al final. Se ejecuta una sentencia mientras sea cierta una condición. En este caso, la sentencia se ejecuta al menos una vez.

Flujograma de la sentencia do/while

Sintaxis:

```
do {  
    sentencias;  
    [iteracion;]  
} while (expresionLogica);
```