ESTRUCTURAS DE CONTROL REPETITIVAS (BUCLES)

Introducción

Permiten repetir la ejecución de un bloque determinado de código

Son para ejecutar n cantidad de veces una opción de código de programa.

Se puede ejecutar n cantidad de veces o hasta que se cumpla una condición específica.

Bucle 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.

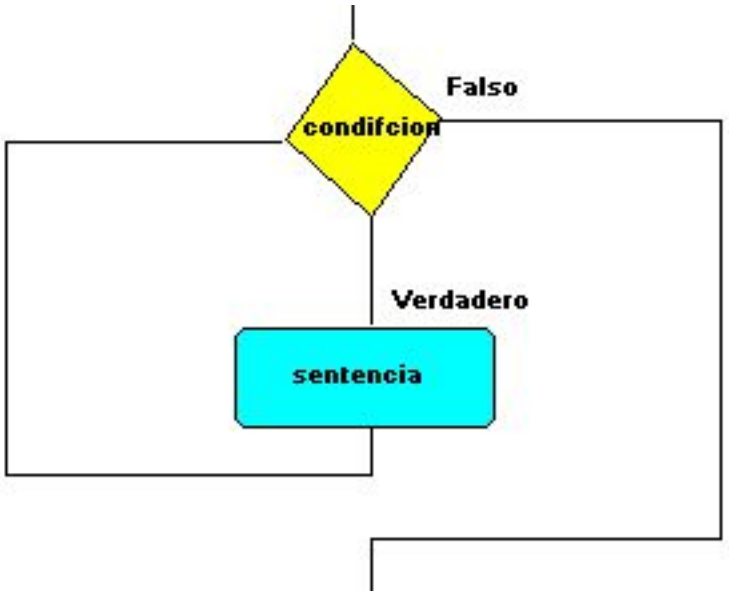
Si la condición es válida, ingresa en el ciclo y ejecuta el código que pusimos en el mismo.

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

Forma de uso:

[code]
while (expresionBooleana)
{
Sentencias;
}
[/code]



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.

Bucle 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.

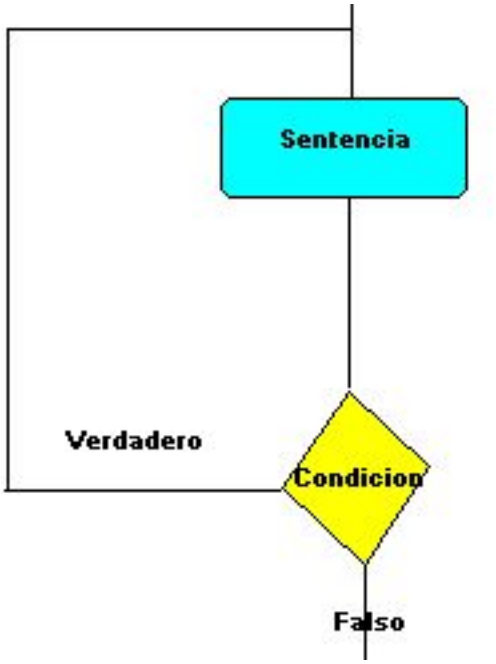
Cuando deja de cumplirse la condición, el programa sale del do while y continúa con el resto del programa.

Tiene como particularidad que el control está al final del bucle

Se debe entrar al bucle al menos una vez

Forma de uso:

[code]
do
{
Sentencias;
} while (expresionBooleana);
[/code]



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.

Bucle 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.

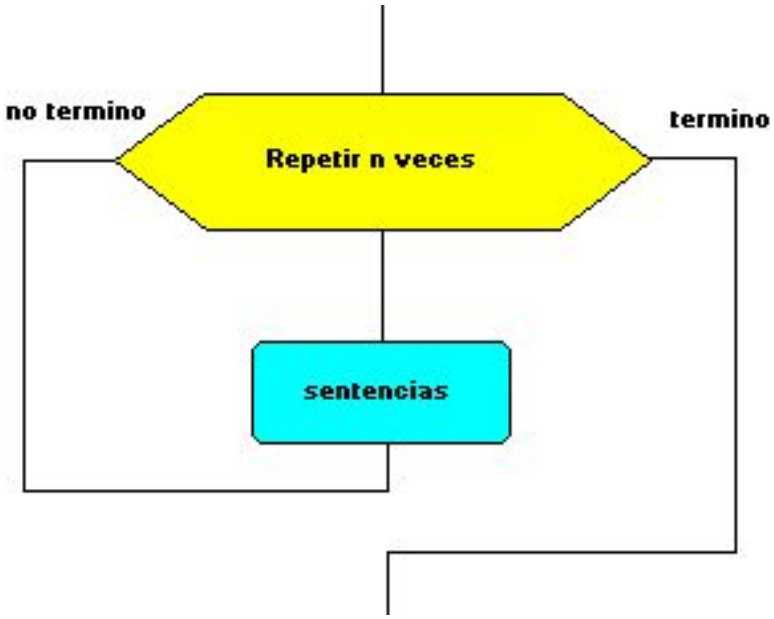
Está compuesta por tres partes:

- 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).

Forma de uso:

[code]
for (inicialización; expresionBooleana; postAccion)
{

Sentencias;
}
[/code]



Break y Continue

La manera estándar para salir de una estructura de bucle es que la condición de prueba principal se convierta en falsa. Los comandos especiales de break y continue ofrecen una salida lateral opcional de todos los tipos de bucles: for, while y do-while.

El comando break sale del bucle más interno que la contiene.

El comando continue salta hasta el final de la iteración actual del bucle más interno que lo contiene.

Sentencia break

A veces es necesario interrumpir la ejecución de un bucle for, while, o do...while.

Válido para bifurcaciones y bucles.

Hace que salga inmediatamente del bloque que se está ejecutando

Forma de uso:

[code]
for (inicialización; expresionBooleana; postAccion)
{
Sentencias;
break;
Sentencias;
}
[/code]

Sentencia continue

La sentencia continue, fuerza al bucle a comenzar la siguiente iteración desde el principio.

Válido solo para bucles

Fuerza una próxima iteración, sin ejecutar todas las sentencias que están después

Forma de uso:

[code]

```
for (inicialización; expresionBooleana; postAccion)
{
Sentencias;
continue;
Sentencias;
}
```
[/code]