Ciclos de Repetición

Un ciclo es una herramienta de la programación que nos permite repetir la ejecución de cierta instrucción o cierto conjunto de instrucciones tantas veces como necesitemos.

Es decir, al día de hoy, si quisiéramos mostrar un cartel "Hola mundo" en la pantalla 5 veces, podemos, pero tendríamos que hacer lo siguiente:

```
cout << "Hola mundo";
cout << "Hola mundo";
cout << "Hola mundo";
cout << "Hola mundo";
cout << "Hola mundo";
```

Es decir, escribir cinco veces la misma línea de código. Con un ciclo (también conocidos como bucles) puedo programar escribir esa instrucción una sola vez pero que se ejecute cinco veces. O las veces que necesite.

Ciclo FOR

Hay distintos tipos de ciclos, vamos a comenzar con el ciclo exacto: el ciclo FOR.

El FOR es un ciclo que nos permite ejecutar un conjunto de instrucciones una cantidad de veces establecida. Por ejemplo, cinco. Por ejemplo, diez. De allí su calificación de "ciclo exacto", porque dará exactamente la cantidad de vueltas que esté configurado para dar.

Para configurar un ciclo FOR vamos a necesitar contar con una variable, generalmente se usa como nombre para dicha variable las letras "x", "y" o "z" aunque también encontrarán muy frecuentemente que la variable del FOR se denomina "i" (de "iterar", que significa ciclar, o dar una vuelta al ciclo).

Una vez con la variable hay tres factores que se requiere configurar en un ciclo:

- Inicialización.
- Condición.
- Incremento.

Cada una de estas acciones ya son conocidas por separado, pero en este caso van a estar trabajando en conjunto para poder configurar lo que se denomina "reloj" del ciclo y que será lo que permita el funcionamiento del mecanismo del mismo.

Inicialización

Se trata de dar un valor inicial a la variable del ciclo. Esta instrucción se ejecutará una sola vez en toda la vida del ciclo y es para darle un punto de partida al mismo. Se puede inicializar en cualquier valor, incluso en el valor de otra variable, aunque en una configuración regular, se suele inicializar simplemente en cero.

Condición

La condición es la instrucción que determinará si se sigue iterando o no. Se ejecuta previo a comenzar cada vuelta del ciclo y si da verdadero, se ejecuta la vuelta. Cuando sea falso, se dará por terminado el ciclo. Aquí se puede asignar cualquier condición válida para un IF, ya que manejan exactamente el mismo formato, incluso se pueden agregar operadores lógicos. En una configuración regular se suele hacer una comparación del tipo "menor a" el valor cuya cantidad de vueltas se quiere dar. Por ejemplo "x < 5" si quiero dar cinco vueltas.

Incremento

El incremento es la instrucción que se encargará de modificar el valor de la variable del ciclo para que, eventualmente, la condición dé falso y el ciclo pueda concluir. De otro modo, la condición dará siempre verdadero y el ciclo no terminaría, lo que sería un problema (se conoce como ciclo que tiende a infinito). Generalmente se utiliza un incremento de a una unidad (x++) aunque también puede variar.

Funcionamiento

Cuando la ejecución de un programa se encuentra con un ciclo, lo primero que hace es reconocer cuál es la variable del mismo. Lo siguiente, y por única vez, es ejecutar la inicialización. Luego de eso ejecuta la condición y si la misma da verdadero, ingresa al ciclo y ejecuta todo lo que haya dentro. Dentro del ciclo pueden haber tantas instrucciones como necesitemos, incluso otros ciclos, pero esto lo veremos más adelante. Luego de haber ejecutado todas las instrucciones dentro del ciclo y haber llegado al final del mismo, el próximo paso es el incremento. Una vez realizado, se vuelve a evaluar la condición, y si sigue dando verdadero, sigue entrando al ciclo. Así hasta que la condición por fin dé falso.

Como se ha dicho antes, cada una de las configuraciones cuenta con lo que llamaremos "configuración estándar", sin embargo, se puede establecer valores de cualquier tipo en cada una de ellas siempre y cuando se cumpla con la estructura respectiva. Hay que tener en cuenta que la cantidad de vueltas que dará el ciclo depende directamente de la configuración de su reloj, con lo cual tenemos que tener muy claro qué es lo que estamos configurando y por qué.

Configuración estándar

Si queremos que un ciclo de diez vueltas, la configuración más sencilla sería:

- x = 0
- x < 10
- x++

Donde "x" arranca en cero, y en cada vuelta se preguntará si "x" es menor a diez. Mientras sea, se ingresa y se da una vuelta. Al final de cada vuelta se incrementará el valor de "x" en uno.

Configuración estándar

Hagamos en código el ejemplo de mostrar cinco veces el hola mundo, pero escribiendo una sola vez dicha instrucción:

```
for(x = 0; x < 5; x++){
   cout<<"Hola mundo";
}
```

Ciclo WHILE

Ya conocemos el Ciclo FOR. El mismo nos permite, a partir de una configuración previamente establecida, repetir un conjunto de instrucciones siempre y cuando conozcamos la cantidad de veces que necesitamos que el mismo se ejecute. Hay ocasiones en las que no se conocerá la cantidad de veces que un algoritmo deberá ser ejecutado; en dichos casos se utiliza otro tipo de ciclo, denominado ciclo inexacto, que repetirá el conjunto de instrucciones dado dependiendo de una condición establecida. Existen básicamente dos tipos de ciclos inexactos en la programación. Por un lado, el While y por otro lado el Do While.

El ciclo While (del inglés “mientras”) comenzará a ciclar siempre y cuando la condición dada sea verdadera, y seguirá ciclando hasta tanto esa situación no cambie.

```
int n = 0;
while(n<10){
   n++;
   cout<<n;
}
```

En el ejemplo se declara una variable inicializada en cero. Cuando el flujo se encuentra con el ciclo While, lo primero que se hace es evaluar la condición: “El contenido de la variable, ¿es menor a diez?”. Naturalmente en este momento la condición tiene como resultado un valor “TRUE”. Recordemos que una condición de este tipo puede adoptar sólo uno de dos valores (true o false. 1 o 0, etc.). Una vez dentro del ciclo, se ejecutan las instrucciones. Se incrementa el valor de la variable y se muestra en pantalla. Una vez concluías estas dos instrucciones, se evaluará la condición nuevamente. El ciclo del ejemplo dejará de girar cuando la variable “N” sea igual a 10 (diez).

Ciclo Do While

Este ciclo funciona casi de la misma manera que el ciclo While tradicional. La diferencia fundamental es que la primera vuelta se ejecutará siempre, y recién para la segunda es que la condición será evaluada para determinar si continuar ciclando o concluir el bloque. Un ejemplo codificado sería:

```
int n = 0;
do{
   n++;
}while(n<10)
```