

TEMA 3: Estructuras de Control: Iterativas

Estructuras Iterativas

En la programación existen las estructuras básicas y fundamentales de un algoritmo, ahora mediante las estructuras condicionales somos capaces de definir la expresión que es la más adecuada según se presente el problema, pero cuando se necesita repetir varias veces un proceso no son suficientes, para ello son las estructuras iterativas capaces de enunciar procesos más complicados y definidos para redundar mientras cumpla o no una condición.

La palabra iteración significa repetición, reiteración; por tanto, en este apartado vamos a estudiar las estructuras que implican repetir o iterar un fragmento de código y las sentencias que se definen en un lenguaje. Aunque aún no hayamos hablado de este tipo de estructuras, su necesidad resulta casi evidente en múltiples algoritmos. De hecho son las estructuras más importantes en los lenguajes imperativos, como el lenguaje C++ que estamos estudiando.

Imaginemos por ejemplo que queremos saber si un número natural n es primo. La solución de esta pregunta, según la definición de número primo, implica comprobar si la división de n por los números menores que él, es exacta o no (es decir, si el resto del cociente es cero o no). Por tanto, el código que resolvería esta pregunta implica la necesidad de una estructura iterativa o reiterativa que vaya repitiendo la división desde 2 hasta $n-1$. Otros casos que nos exigen disponer de estructuras iterativas son, por ejemplo aquellos donde un problema matemático se resuelve repitiendo una misma operación. En general, tal es el caso de las expresiones matemáticas que se escriben con un sumatorio.

Cada instrucción o conjunto de instrucciones de un bucle se define como iteración. Una forma de controlar un bucle es mediante una variable llamada contador cuyo valor se incrementa o decrementa en una cantidad constante en cada repetición que se produzca. También, los bucles suelen utilizar otro tipo de variables llamadas acumulador, cuya misión es almacenar una cantidad variable resultante de operaciones sucesivas y repetidas. Es como un contador, con la diferencia que el Estructuras Selectivas incremento/decremento es variable. Las estructuras de control se dividen en:

- Repetir
- Mientras
- Para

Repetir

Es una estructura de control que permite repetir varias veces un bloque de instrucciones hasta que alguna condición lógica sea verdadera. La condición se verifica al final del ciclo por lo que el bloque de instrucciones es ejecutado al menos una vez. Usaremos la siguiente notación:

repeat

...

until <expresión lógica>

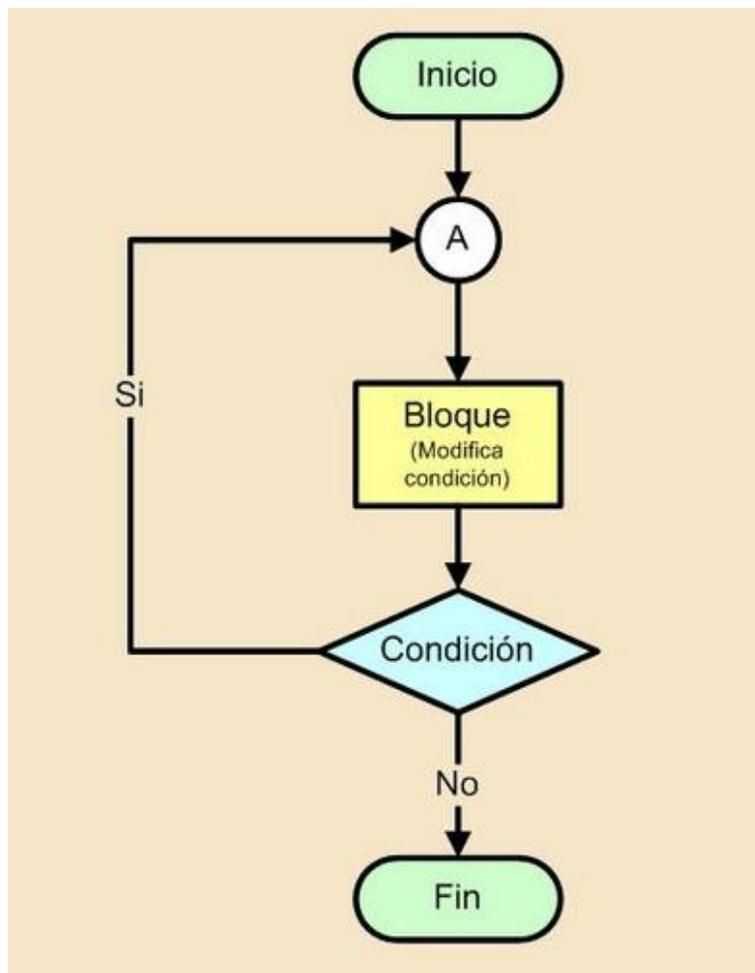
Esto lo leeremos como:

repetir

...

hasta <expresión lógica>

La representación gráfica a través de un diagrama de flujo es la siguiente:



Estructura Repetir

Ejemplo: Determinar el promedio de los números de una serie terminada en 0:

```
Procedure Main()  
Integer X, Ac,C;  
Real Promedio;  
Ac = 0;  
C = 0;  
Repeat  
Read(X);  
Ac = Ac + X;  
C = C + 1;  
Until X = 0  
Promedio = Ac / C;  
Write(Promedio);  
EndProcedure
```

Mientras

Es una estructura de control iterativa que permite ejecutar un bloque de instrucciones 0 o más veces, mientras que cierta condición lógica sea verdadera. Usaremos la notación:

```
While <condición> do  
...  
EndWhile
```

Nótese que a diferencia del repetir, la condición es evaluada al comienzo del ciclo y la ejecución del ciclo se detiene cuando la condición deja de ser verdadera. Es posible convertir un repetir en mientras y viceversa.

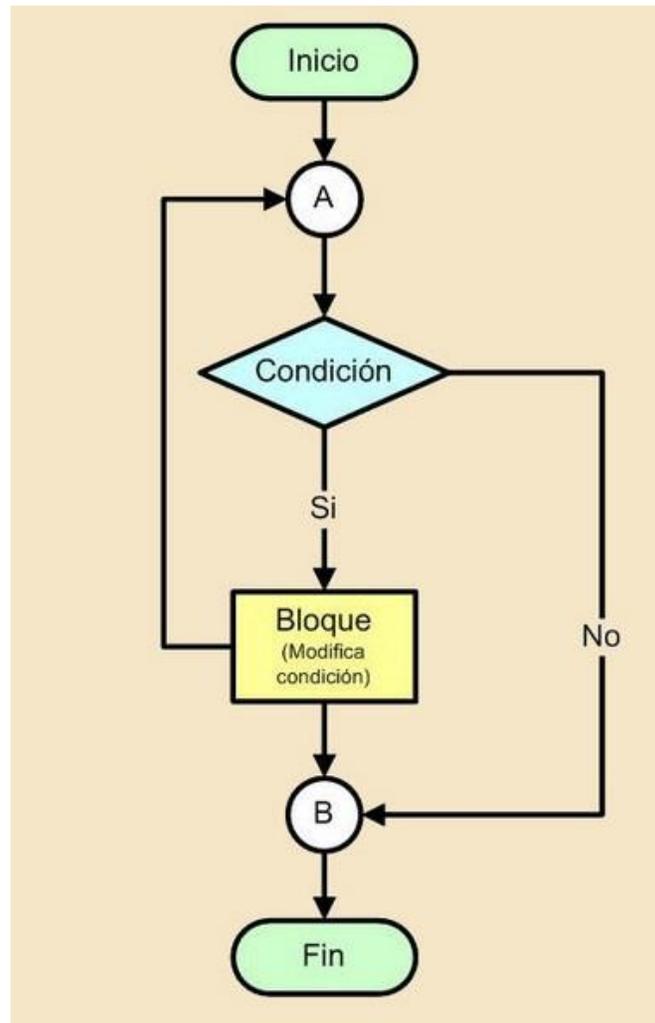
Considérese el siguiente bloque repetir:

```
Repeat  
A;  
...  
Z;  
Until <condición>
```

Esto se convierte en:

```
A;  
...  
Z;  
While NOT <condición> do  
A;  
...  
Z;  
EndWhile
```

La representación gráfica a través de un diagrama de flujo es la siguiente:



Estructura Mientras

Ejemplo: Escribir los números de 2 en 2 desde el 2 hasta el 100

```
Procedure Main()  
Integer X;  
X=2;  
While (X<=100) do  
Write(X)  
X = X + 2;  
EndWhile  
EndProcedure
```

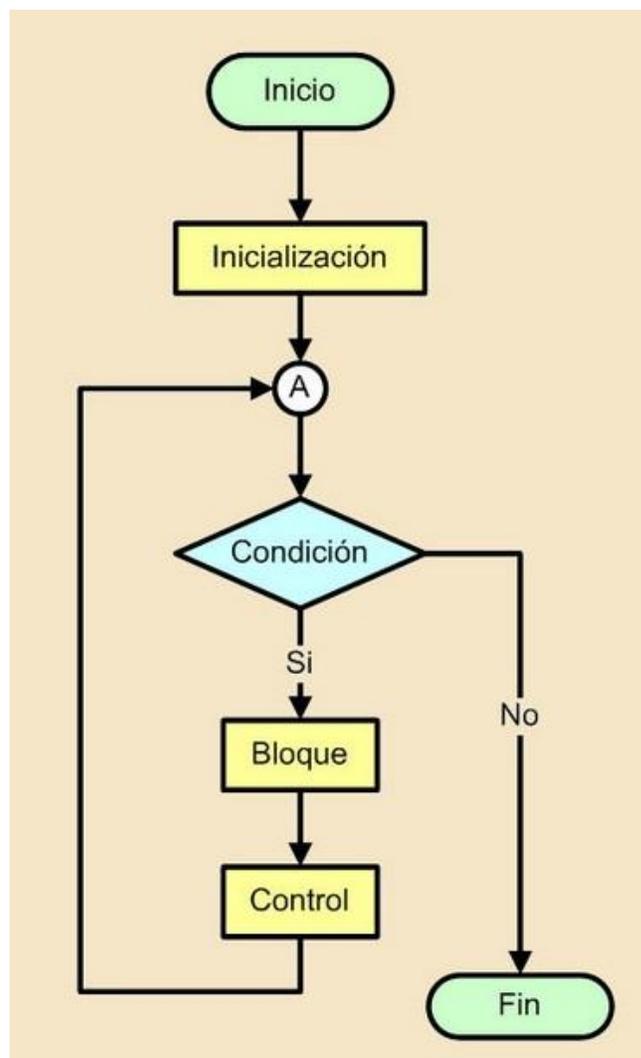
Para

Es una estructura de control iterativa controlada por una variable que se incrementa automáticamente en cada iteración. La parada del ciclo está controlada por el valor de la variable. Usaremos la notación:

```
For <Inicialización Variable> to <expresión aritmética> [step <Expresión aritmética>] do  
    ...  
EndFor
```

Nota: Si se omite el parámetro "Step" se asume que el incremento de la estructura es de 1 en 1.

La representación gráfica a través de un diagrama de flujo es la siguiente:



Estructura Para

Ejemplo: Escribir la tabla de multiplicar de un número x.

```
Procedure Main()  
Integer i, x,t;  
Read(x);  
For i = 1 to 11 do  
t = i * x;  
Write(t);  
EndFor  
EndProcedure
```

Licenciado Wilson Manyoma, Caracas, Noviembre 2012