

¿Qué es un ciclo?

Período de tiempo que, acabado, se vuelve a contar de nuevo.

También podemos decir que es una serie de pasos (procesos) repetitivos que se conocen como ciclos.

Existen 3 ciclos que son los más comunes:

- **For (para):** Se dice que es un ciclo predeterminado debido a que para poderse utilizar se debe conocer de antemano la **cantidad de veces que se va a repetir** el grupo de instrucciones. Su sintaxis es:

```
PARA I = LI; LF; K
```

```
// Instrucciones (se repiten la cantidad de veces asignada)
```

```
FIN PARA
```

DEFINICIONES:

I → Es el índice del ciclo, es un campo variable numérico entero, que toma como primer valor LI y que se incrementa en K cada vez que se ejecutan las instrucciones incluidas dentro del ciclo.

LI → Se le denomina valor inicial, es un campo numérico entero, no necesariamente variable. Es el primer valor que toma el índice.

LF → Se le denomina límite final, es un campo numérico entero, no necesariamente variable, cuando el índice supera este valor se da por terminado el proceso repetitivo y el control es transferido a la instrucción que se encuentre inmediatamente después del ciclo.

K → Incremento, es el valor que se le suma al índice cada vez que se ejecuta el grupo de instrucciones que hay dentro del ciclo.

Ejemplo 1: Escribir un algoritmo que muestre en la pantalla los múltiplos de 7 comprendidos entre 1 y 70 inclusive.

ANÁLISIS: Como podemos observar, se necesita una variable que **comience en 7** y que cada vez, después de imprimirla **se le sume 7**, debe detenerse **cuando llegue a 70**. Lógicamente podemos implementar un ciclo PARA cuyo **valor inicial sea el 7**, el **valor final sea el 70** y el **incremento sea de 7** así:

```
Inicio
```

```
Para I = 7; 70; 7
```

```
Imprima I
```

```
Fin para
```

```
Termine
```

Video de profundización:

<https://youtu.be/vExr9lgFCZ4>

- **While (mientras):** La estructura repetitiva mientras es aquella en que el cuerpo del bucle se repite mientras se cumple una determinada **condición**. cuando se ejecuta la instrucción mientras, **la primera cosa que sucede es que se evalúa la condición**. si se evalúa falsa, no se toma ninguna acción y el programa prosigue en la siguiente instrucción del bucle. si la **expresión booleana** es verdadera, entonces se ejecuta el cuerpo del bucle, después de lo cual se evalúa de nuevo la expresión booleana. este proceso se repite una y otra vez mientras la expresión booleana sea verdadera. La sintaxis es la siguiente:

Mientras CONDICIÓN haga

// Instrucciones (se repiten la cantidad de veces asignada)

Fin mientras

NOTA: Se suele usar una cantidad entera (contador) que suma de 1 en 1 para llegar a la condición y el ciclo termine.

Ejemplo 1: Realizar un algoritmo que lea 3 números y determine la suma entre ellos.

ANÁLISIS: Debemos considerar **la condición** de un límite de números (**3**) con los que debemos **sumar uno a uno (los números que se ingresan)** para mostrar su **totalidad (suma)**, pero se debe considerar que **el contador** debe sumar cada vez que terminan las instrucciones **para cumplir los 3 números**.

INICIO

ENTERO: contador, numero, suma

contador=1

suma=0

MIENTRAS (contador <= 3) HAGA

 MOSTRAR ('DIGITE UN NUMERO')

 LEER (numero)

 suma = suma + numero

 contador = contador + 1

FIN MIENTRAS

 MOSTRAR ('LA SUMA ES:', suma)

FIN

Video de profundización:

<https://youtu.be/F2MSmbrnyws>

- **Repita (do while):** Es un tipo de estructura repetitiva **eficiente**. Lo que lo diferencia con el while es que en la estructura do-while la condición **se evalúa al finalizar el ciclo**, esto hace que las instrucciones se **ejecuten cuando menos una vez**. La sintaxis es la siguiente:

```
Haga
// Instrucciones (se repiten dependiendo la respuesta)
mientras (CONDICIÓN)
```

Ejemplo 1: Ingresar el nombre del alumno, la nota, luego preguntar si desea continuar, al final presentar el número de aprobados. Se aprueba con 3.5

ANÁLISIS: Lo primero que debemos tener en cuenta es **la respuesta del usuario** (continuar o no) para seguir realizando el ciclo, porque como ya sabemos **se realiza mínimo una vez**, pero luego **evalúa la condición**, entonces, necesitamos pedir los datos y preguntar si la nota es mayor o igual a 3.5 para contar los aprobados, pero **también preguntar si quiere continuar o no**.

```
INICIO
STRING continua, nombre
DECIMAL nota
ENTERO numeroAprobados
numeroAprobados = 0
REPITA
    MOSTRAR ("Digite nombre del alumno: ")
    LEER nombre
    MOSTRAR ("Digite la nota del alumno: ")
    LEER nota
    Si nota >= 3.5 entonces
        numeroAprobados = numeroAprobados + 1
    FIN SI
    MOSTRAR (" Desea continuar (si/no): ")
    LEER continua
    HASTA (continua == "no")
    MOSTRAR ("Cantidad aprobados:", numeroAprobados)
FIN
```

Video de profundización:

<https://youtu.be/plGrHtZbgzs>

Repositorio: <https://github.com/jfelipeq14/ciclos>