



CICLOS CUALITATIVOS

Se debe recordar que los ciclos son estructuras repetitivas y tiene dos esquemas: Cuantitativos y Cualitativos y que, de manera general, todo ciclo debe tener tres aspectos fundamentales:

1. Dónde Inicia (Límite Inicial).
2. Dónde termina (Límite Final).
3. Una instrucción que modifica el inicio y hace que llegue a su terminación.

En esta clase, se tratará el esquema de Ciclos Cualitativos y de estos existen dos tipos: Ciclo Mientras y Haga Mientras.

- ✦ Se utilizan cuando no se conoce el número de veces que debe repetirse la secuencia de un determinado ciclo. En lugar de usarse un “N” se usa una “**cualidad**”.
- ✦ El usuario decide cuando terminar las repeticiones sin importar el número, lo que lo diferencia de un Ciclo cuantitativo, en el cual el número de iteraciones es conocido desde el principio.

CICLO MIENTRAS

ESTRUCTURA:

Escriba: “Ingrese el dato”

Lea: dato // **Dónde inicia**

Esta expresión es con el dato que se lee antes **Dónde termina.**

Mq (**expresión lógica**) haga

//secuencia de instrucciones

Escriba: “Ingrese el dato”

Lea: dato

FMq

El dato que se captura antes de finalizar el ciclo, es el que **modifica la expresión lógica** que controla el ciclo

EJEMPLO 1:

Haga un algoritmo que encuentre el promedio de un grupo de datos numéricos positivos.

En este ejemplo, no se sabe cuántos datos son (*el N que se definía en los cuantitativos*), pero si se sabe una **cualidad o una característica** y es que deben ser números positivos. Por lo tanto, esta cualidad sirve para controlar el ciclo.



Inicio

Real: promedio

promedio $\leftarrow$ Calculo ()

Escriba: "El promedio de los datos ingresados es =", promedio

Fin

Funcion Calculo ()

Entero: suma $\leftarrow$ 0, n $\leftarrow$ 0, num

Escriba: "Ingrese un número positivo (0 o negativo para salir)"

Lea: num // Límite Inicial

Mq (num>0) haga // Límite Final

suma $\leftarrow$ suma + num

n $\leftarrow$ n+1

Escriba: "Ingrese un número positivo (0 o negativo para salir)"

Lea: num //instrucción que modifica la expresión

FMq

Retorne suma/n

FinFuncion

EJEMPLO 2:

Será el mismo ejemplo anterior, pero con Haga Mq.

Funcion Calculo ()

Entero: suma $\leftarrow$ n $\leftarrow$ 0, num

Escriba: "Ingrese un número positivo (0 o negativo para salir)"

Lea: num // Límite Inicial

Haga

suma $\leftarrow$ suma + num

n $\leftarrow$ n+1

Escriba: "Ingrese un número positivo (0 o negativo para salir)"

Lea: num //instrucción que modifica la expresión

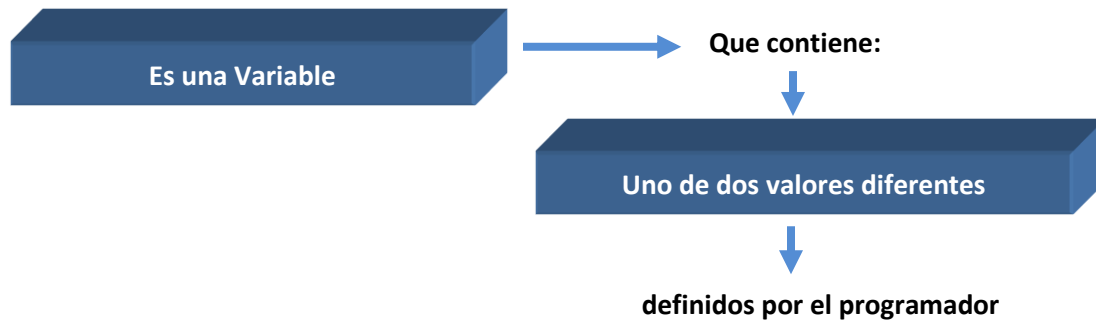
Mq (num>0) // Límite Final

Retorne suma/n

FinFuncion



BANDERAS



- ✦ Es un seleccionador de uno de dos valores.
- ✦ Previamente (antes de usarse como seleccionador), debe tener uno de los dos valores asignados.
- ✦ Es muy usado como “cualidad” en ciclos cualitativos.
- ✦ El valor de la bandera, puede ser de cualquier tipo de dato: Ejemplo: Numérico: 1 o 2, Caracter: 's' o 'n'.

EJEMPLO 1:

Hacer un algoritmo que encuentre la suma de los valores de un conjunto de números enteros, la cantidad de valores negativos, positivos, iguales a 0 y el total de números del conjunto.

Inicio

Real: sum

sum $\leftarrow$ Cantidad()

Escriba: “Y el resultado de la suma de los valores es = ”, sum

Fin

Funcion Cantidad()

Entero: totalneg $\leftarrow$ 0, totalpos $\leftarrow$ 0, total0s $\leftarrow$ 0, suma $\leftarrow$ 0, totaln, num

Caracter: ban $\leftarrow$'s'



```
Mq (ban = 's') haga
    Escriba: "Digite un número"
    Lea: num
    suma ← suma + num

    Si (num < 0) ent
        totalneg ← totalneg + 1
    sino
        Si (num > 0) ent
            totalpos ← totalpos + 1
        sino
            total0s ← total0s + 1
        Fsi
    Fsi

    Escriba: "Desea continuar?"
    Lea: ban
    Fmq
```

totaln ← totalneg + totalpos + total0s

Escriba: "La cantidad de números negativos es =", totalneg
Escriba: "la cantidad de números positivos es =", totalpos
Escriba: "la cantidad de números iguales a cero es =", total0s
Escriba: "el número total del conjunto es =", totaln

Retorne suma
Fin_Funcion

EJEMPLO 2: Las banderas también son muy útiles, en lo referente a cálculo.

Elaborar un algoritmo que genere los primeros N términos de la sucesión: 17, 15, 18, 16, 19, 17, 20, 18, 21,...

Inicio

Entero: N

Escriba: "Digite cuántos términos desea generar"
Lea: N

Sucesion(N)

Fin



Procedimiento Sucesion (Entero: n)

Entero: con, ter $\leftarrow$ 17, ban $\leftarrow$ 1 // la bandera tomará valores de 1 ó 2

```
Para (con=1, n, 1) haga
  Escriba: ter
  Si (ban = 1) ent
    ter  $\leftarrow$  ter - 2
    ban  $\leftarrow$  2
  sino
    ter  $\leftarrow$  ter + 3
    ban  $\leftarrow$  1
  Fsi
Fpara
```

Fin_Procedimiento

- ⊕ Las banderas también son usadas, en rompimiento de ciclos, es decir para detener un ciclo forzosamente, porque ya se cumplió lo que se necesitaba (es una terminación anormal de un ciclo).
- ⊕ Esta ruptura se hace cambiando el sentido de la expresión lógica que controla el ciclo, para que se vuelva falsa y no se continúe ejecutando la secuencia de instrucciones del ciclo.

EJEMPLO 3:

Se tiene una serie de N números enteros. Se quiere saber si dentro de ellos existe por lo menos uno, cuyo valor sea cero.

Inicio

Entero: N

Caracter: resultado

Escriba: "Digite la cantidad de números de la serie"

Lea: N

resultado $\leftarrow$ BuscarCero (N)

```
Si (resultado = 'V' v resultado = 'V') ent
  Escriba: "Hay un cero en la serie"
sino
  Escriba: "No hay ceros en la serie"
Fsi
```

Fin



Funcion BuscarCero (Entero: n)

Entero: con $\leftarrow$ 1, num

Caracter: ban $\leftarrow$ 'F'

Mq (con $\leq$ n $\wedge$ ban = 'F') haga

 Escriba: "Digite un número"

 Lea: num

 Si (num = 0) ent

 ban $\leftarrow$ 'V'

 sino

 con $\leftarrow$ con + 1

 Fsi

Emq

Retorne ban

Fin_Funcion