



Clase 06: Diagramas de Flujo Estructura Iterativa

Introducción Lógica Computacional



Agenda



Desafiase a crear, diseñar y elaborar algoritmos
que resuelvan problemas iterativos

- Desarrollo de Algoritmos: Iterativo
- Condición Lógica / Operadores Relacionales / Operadores Lógicos / Tablas de Verdad
- Asignacion y Actualizacion de variables / Inicializar – Incrementar y Decrementar variables : Contadores y Acumuladores
- Ciclos exactos – Ciclos condicionales





Estructuras repetitivas

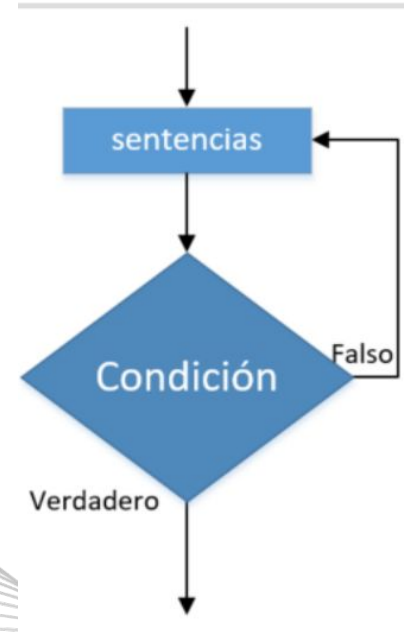
Las instrucciones iterativas o de repetición permiten ejecutar una secuencia de instrucciones más de una vez. Un bucle es una sección de código que se repite. Es decir que cuando se termina de ejecutar la última instrucción del conjunto, el flujo de control retorna a la primera sentencia y comienza una nueva repetición. Se denomina iteración al hecho de repetir la ejecución de una secuencia de acciones.





Estructura repetir-hasta

Esta estructura repetitiva se utiliza cuando se desea que se ejecute una iteración al menos una vez antes de comprobar la condición de repetición. Es decir, primero se ejecuta el bucle y luego se comprueba la condición para reproducir una nueva iteración o no del bucle. Esta estructura ejecuta el bucle de repetición mientras el resultado de la condición que se evalúa sea falso.





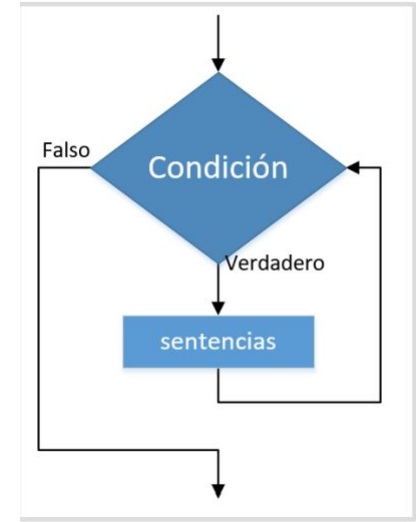
Estructura repetir/ hasta

```
Algoritmo repetir_num  
  
    Definir num Como Entero  
  
    Repetir  
        Mostrar "Ingrese un número:"  
        Leer num  
    Hasta Que num == 7  
  
FinAlgoritmo
```



Estructura mientras

En la estructura mientras, se evalúa primero la condición y si esta es verdadera, entonces se ejecutan las sentencias definidas en el bucle. Al finalizar la iteración, se vuelve a evaluar la condición para volver a ejecutar una nueva iteración hasta que la condición dé como resultado un valor falso.





Estructura mientras

```
1  Algoritmo repetir_num
2
3      Definir num Como Entero
4
5      num = 0
6
7      Mientras num < 3
8          num ← num + 1
9          Escribir "El número es:", num
10
11 FinMientras
12
13 FinAlgoritmo
```





BUCLAS CONTROLADAS POR CONTADOR

```
1 Algoritmo repetir_num
2
3   Definir num Como Entero
4
5   num = 0
6
7   Mientras num < 3
8     num ← num + 1
9     Escribir "El número es:", num
10
11 FinMientras
12
13 FinAlgoritmo
```

1. Se utiliza una variable de contador para controlar el número de iteraciones que se realizarán en el bucle.
2. En cada iteración del bucle, el contador se actualiza y se verifica si se ha alcanzado el número de iteraciones deseadas. Si no se ha alcanzado, se repite el bucle.
3. Estos bucles son útiles cuando se conoce de antemano el número de veces que se desea ejecutar un bloque de código.





BUCLAS CONTROLADAS POR CENTINELA

```
1  Algoritmo repetir_num
2
3      Definir num Como Entero
4      Escribir "Ingrese número"
5      Leer num
6      mientras num ≠ 0 hacer
7          Escribir "Ingrese otro número"
8          Leer num
9      fin mientras
0  FinAlgoritmo
1
```

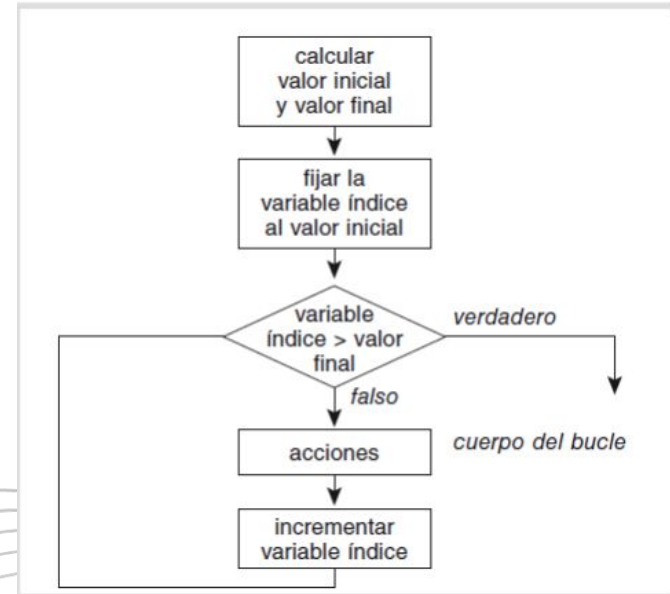
Se utiliza una condición de salida para determinar cuándo se debe salir del bucle. En cada iteración del bucle, se verifica si se cumple la condición de salida. Si se cumple, se sale del bucle. Estos bucles son útiles cuando no se conoce de antemano el número de iteraciones que se necesitan y se necesita salir del bucle en función de una condición determinada.



Estructura para/desde

En muchas ocasiones se conoce el número de veces que se desea que un bucle se repita.

Para ejecutar el bucle, se evalúa una condición que debe ser verdadera. La condición más comúnmente usada es que el valor del índice sea menor o igual a un valor final. Si esto se cumple, entonces se ejecuta el bucle. Al finalizar, se incrementa o decrementa el índice y se vuelve a evaluar la condición. Cuando la condición es falsa, se ignora el bucle y se continúa con el resto del algoritmo.





Estructura para/desde

```
} Algoritmo estructura_Para_Desde
|   acumulado = 0
|   para i = 1 hasta 10 hacer
|       acumulado = acumulado + i
|   fin para
|   Escribir("La suma de los números del 1 al 10 es: "), acumulado
} FinAlgoritmo
}
```





EJERCICIOS CON ESTRUCTURAS REPETITIVAS

1. Realizar un programa que solicite al usuario que ingrese una clave, y que continúe pidiéndole la clave hasta que ingrese la clave correcta (por ejemplo, "1234").
2. Hacer un algoritmo en Pseint para calcular la suma de los primeros cien números con un ciclo mientras.
3. Realizar un programa que pida al usuario que ingrese un número entero positivo, y que muestre por pantalla la suma de los números enteros pares entre 1 y el número ingresado por el usuario.





EJERCICIO EN EQUIPOS

1. Crea una aplicación que permita adivinar un número. La aplicación genera un número aleatorio del 1 al 100. A continuación va pidiendo números y va respondiendo si el número a adivinar es mayor o menor que el introducido, además de los intentos que te quedan (tienes 10 intentos para acertarlo). El programa termina cuando se acierta el número (además te dice en cuantos intentos lo has acertado), si se llega al límite de intentos te muestra el número que había generado.





¿QUE SON LOS ARREGLOS -ARRAY-?

Un arreglo es una estructura de datos estática y representa un conjunto finito y ordenado de elementos del mismo tipo (homogéneos). Los arreglos pueden ser unidimensionales, también llamados vectores (o arreglos lineales), bidimensionales (matrices) o multidimensionales.

Scalar

24

Vector

$\begin{bmatrix} 2 & -8 & 7 \end{bmatrix}$

row

or
column $\begin{bmatrix} 2 \\ -8 \\ 7 \end{bmatrix}$

Matrix

$\begin{bmatrix} 6 & 4 & 24 \\ 1 & -9 & 8 \end{bmatrix}$

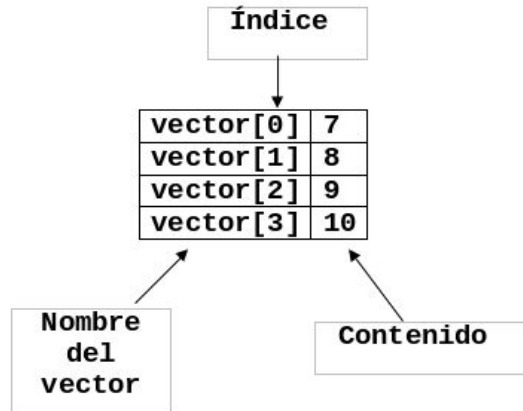
row(s) x column(s)





¿QUÉ SON LOS VECTORES?

Un arreglo es una estructura de datos estática y representa un conjunto finito y ordenado de elementos del mismo tipo (homogéneos). Los arreglos pueden ser unidimensionales, también llamados vectores (o arreglos lineales), bidimensionales (matrices) o multidimensionales.



```
1  Proceso sin_titulo
2      Dimension datos(3);
3  FinProceso
4
```





¿QUÉ SON LAS MATRICES?

Un arreglo es una estructura de datos estática y representa un conjunto finito y ordenado de elementos del mismo tipo (homogéneos). Los arreglos pueden ser unidimensionales, también llamados vectores (o arreglos lineales), bidimensionales (matrices) o multidimensionales.

