

ALGORITMOS Y LENGUAJES DE PROGRAMACION

ESTRUCTURAS NO SECUENCIALES
Sentencias Repetitivas

PSEUDOCÓDIGO

SENTENCIAS REPETITIVAS

Muchos problemas requieren una capacidad de repetición en la cual el mismo cálculo o secuencia de sentencias se **repita**, una y otra vez, usando diferentes conjuntos de datos.

PSEUDOCÓDIGO

SENTENCIAS REPETITIVAS

Los casos de repetición incluyen la **verificación** continua de las **entradas de datos** del usuario hasta que se ingrese, una contraseña válida; contar y acumular totales; y la aceptación constante de datos de entrada y el recálculo de valores de salida que sólo se detiene al ingresar un valor concreto.

PSEUDOCÓDIGO

SENTENCIAS REPETITIVAS

La secuencia de sentencias que se repiten en la solución de un problema se conoce como **ciclo o bucle**, porque después de ejecutarse la última sentencia del ciclo, se regresa, a la primera sentencia de la sección y comienza otra repetición a través de la sección repetitiva.

PSEUDOCÓDIGO

Para las estructuras repetitivas se utilizan dos tipos especiales de variables:

Contadores:

Son variables que incrementan o disminuyen su valor en una **cantidad constante**, la misma que debe inicializarse en un valor constante.

```
contador    ←    1;  
contador    ←    contador +1;
```

PSEUDOCÓDIGO

Acumuladores:

Son variables que se usan para incrementar o disminuyen su valor en una **cantidad variable**, deben ser inicializados su valor.

suma \leftarrow **0 ;**

suma \leftarrow **suma + n ;**

PSEUDOCÓDIGO

SENTENCIAS REPETITIVAS

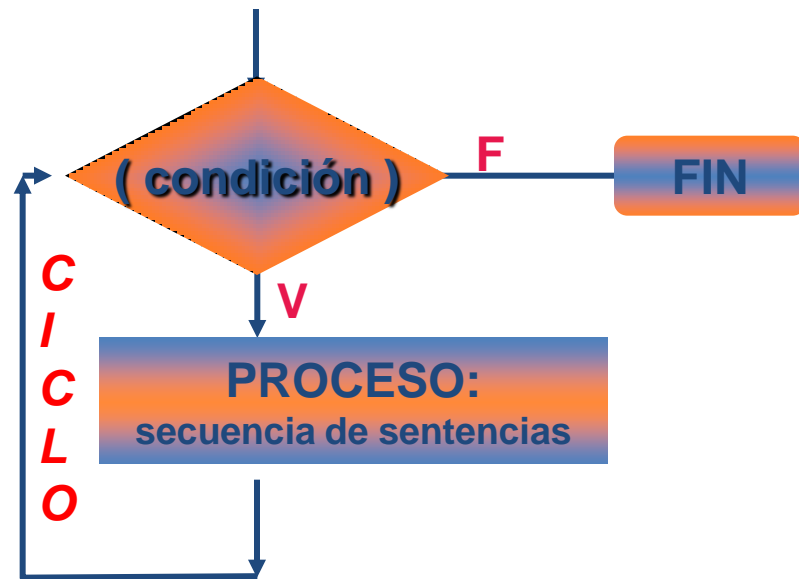
En general, existen tres formas de representar las sentencias repetitivas:

- Mientras Hacer
- Repetir Hasta Que
- Para.

PSEUDOCODIGO

a) SENTENCIA REPETIR : MIENTRAS HACER

CICLOS CONTROLADOS
EN LA ENTRADA



Mientras (Condición) Hacer
Sentencia (s)

FinMientras

PSEUDOCODIGO

a) SENTENCIA REPETIR : MIENTRAS HACER

Una sentencia **Mientras** contiene una condición que controla la ejecución de un proceso formado por una o más sentencias, dicho proceso se ejecutará repetidamente si la **condición es verdad**.

La **condición** contenida dentro del paréntesis es la **condición evaluada** para determinar si se ejecuta la **sentencia que sigue a la condición evaluada**.

PSEUDOCODIGO

a) SENTENCIA REPETIR : MIENTRAS HACER

i). La repetición condicional de cero ciclos

Si la primera evaluación de la condición es falsa, entonces la repetición nunca se ejecutará.

ii) Repetición indefinida de la secuencia de sentencias. (El bucle infinito)

La repetición indefinida que nunca se termina se denomina Bucle infinito o sin fin para evitar dichos ciclos indefinidos se debe estipularse una SENTENCIA que permita alterar el valor de la expresión probada en la sección repetitivo.

iii) Finalización de bucles con datos de entrada.

El método más correcto para terminar un bucle que lee una lista de valores es un centinela. Un valor centinela es un valor especial usado para indicar el final de una lista de datos.

CONSTRUCCION DE ALGORITMOS

PROBLEMA N°8

Escribir un algoritmo que lea un número enteros positivos “n” y luego imprima su factorial.

CONSTRUCCION DE ALGORITMOS

Programa Problema8

Leer n;

minumero <- n;

fact <- 1;

MIENTRAS n > 0 Hacer

fact <- fact * n;

n <- n - 1;

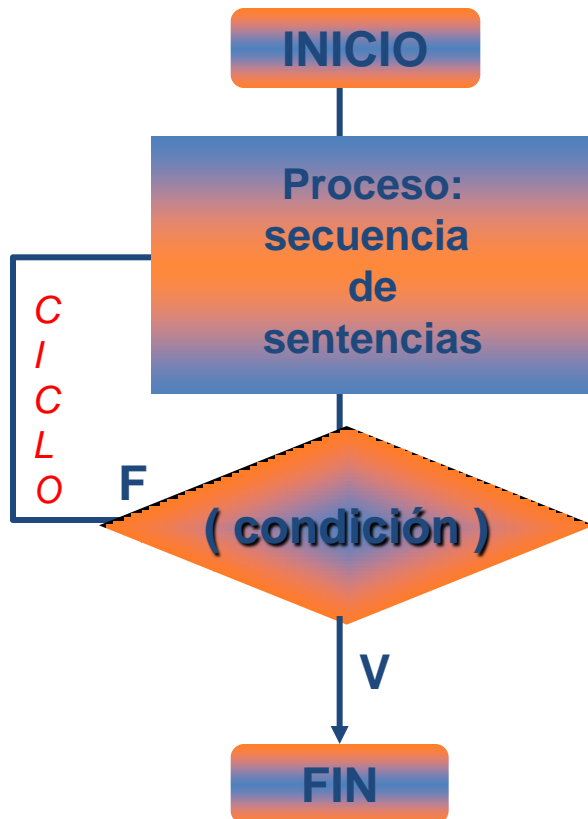
FinMientras

Escribir "El factorial de n es: ", fact;

FinPrograma

PSEUDOCODIGO

b) SENTENCIA REPETIR : HASTA QUE



CICLO CONTROLADO A LA SALIDA

Repetir
Secuencia de sentencias
HASTAQUE (condición) sea verdad

La sentencia **HASTA QUE**
Ejecuta una secuencia de instrucciones
hasta que la condición sea verdad.

PSEUDOCODIGO

b) SENTENCIA REPETIR HASTA QUE

La condición es una expresión, el valor que representa sólo puede ser verdadero o falso.

Se repite una secuencia de sentencias **HASTA QUE** **la condición tome un valor de verdad.**

El ciclo se ejecuta por lo menos una vez porque la condición se evalúa después de la ejecución de cada iteración.

CONSTRUCCION DE ALGORITMOS

PROBLEMA N°9

Diseñar un algoritmo que permita calcular la suma acumulada de la serie :

$$1+2+3+4+....+n$$

El usuario ingresa el numero de términos a sumar.

CONSTRUCCION DE ALGORITMOS

Programa Problema9

Leer n;

cont <- 1;

suma <- 0;

Repetir

suma <- suma + cont;

cont <- cont + 1;

Hasta Que ($n < cont$)

Escribir " La suma es:", suma ;

FinPrograma

PSEUDOCODIGO

c) SENTENCIA REPETITIVA: PARA

La sentencia **PARA** se utiliza para implementar un ciclo que se repite un número definido de veces.

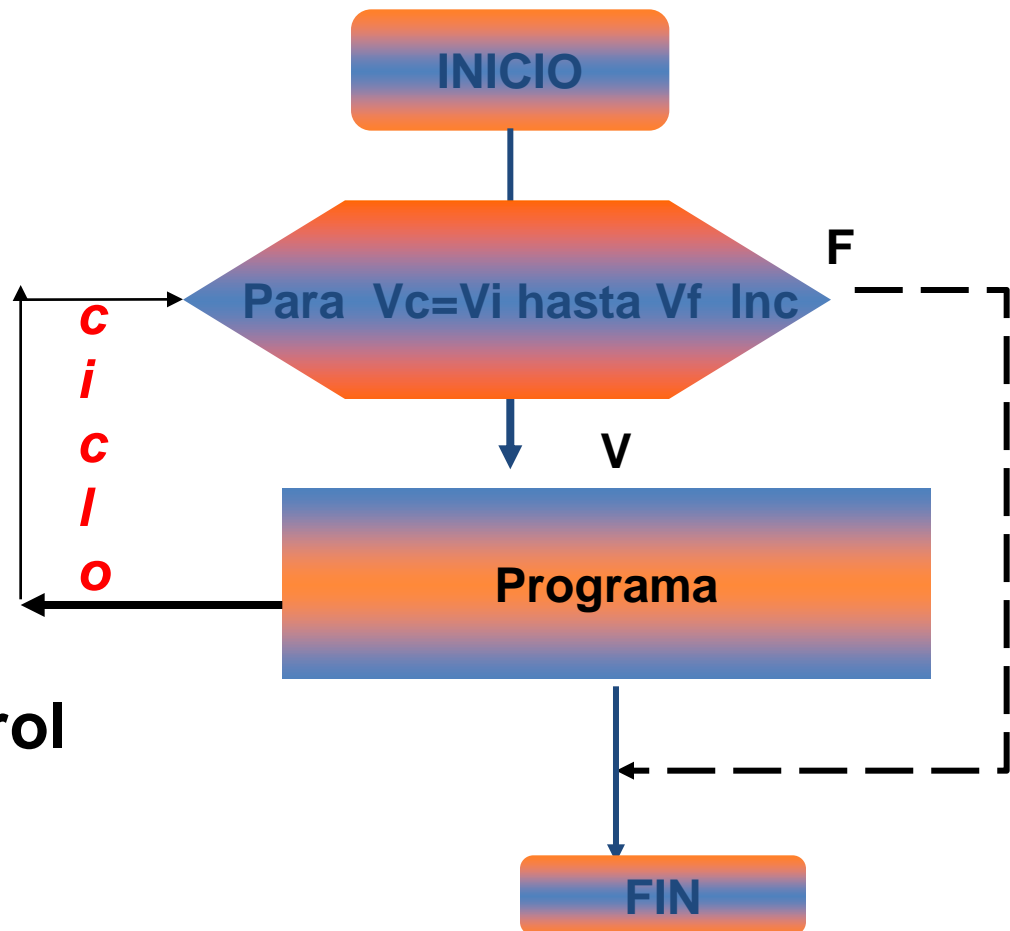
Donde:

Vc: variable de control

Vi: valor inicial

Vf: valor final

Inc: incremento



PSEUDOCODIGO

c) SENTENCIA REPETITIVA : PARA

Para $V_c = V_i$ hasta V_f (con Inc) **Hacer**

c
/
c
/
o

sentencia1

sentencia2

sentencia3

FinPara

PSEUDOCODIGO

c) SENTENCIA REPETIR - PARA

La sentencia **Para** tiene los siguientes elementos:

Vc: es la variable de control de la estructura repetitiva y actúa como un contador.

Vi: es el valor inicial que inicializa la variable de control (número de repeticiones) de la estructura repetitiva.

Inc: es el elemento de incremento que aumenta o disminuye la variable de control (contador) cada vez que se ejecuta el ciclo.

CONSTRUCCION DE ALGORITMOS

PROBLEMA N° 10

**Escribir un algoritmo que lea un
número enteros positivos “n” y
luego
imprima su factorial.**

CONSTRUCCION DE ALGORITMOS

Programa Problema10

Leer n;

factorial <- 1;

Para i <- 1 hasta n Hacer

factorial <- factorial * i;

FinPara

Escribir " El Factorial de N es = ", factorial ;

FinPrograma

CONSTRUCCION DE ALGORITMOS

***** Ejecución Iniciada. ******

5

El Factorial de N es = 120

***** Ejecución Finalizada. ******

PSEUDOCODIGO

c) SENTENCIA REPETIR PARA

Los **CICLOS ANIDADOS** son utilizados cuando un **ciclo Para interior** esta contenido dentro de otro **ciclo exterior**.

```
Para i ← 1 hasta 5      // inicio del ciclo exterior | F
    Escribir "i toma el valor de", i;
    Para j ← 1 hasta 4  // inicio del ciclo interior* j
        Escribir " j = ", j ;
        FinPara
    Finpara              // fin del ciclo exterior *i
```

PSEUDOCODIGO

c) SENTENCIA REPETIR PARA

La salida del algoritmo sería la siguiente:

i toma el valor de 1 // ciclo exterior

j = 1, j = 2, j = 3, j = 4 // ciclo interior j

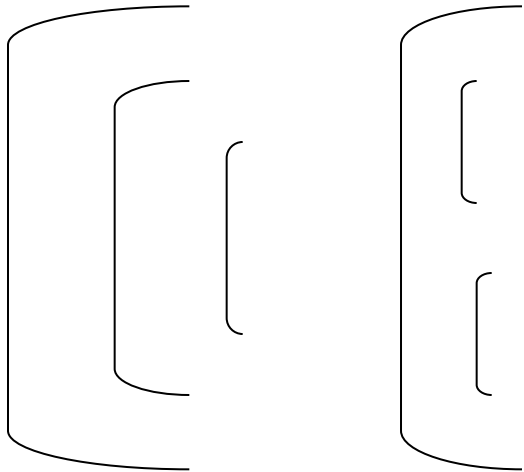
i toma el valor de 2

j = 1, j = 2, j = 3, j = 4 // ciclo interior j

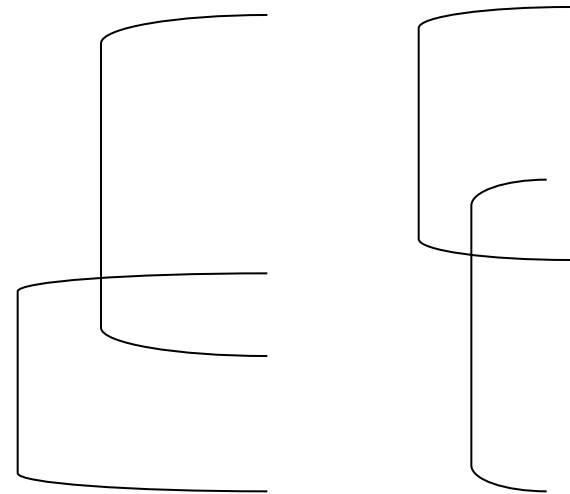
PSEUDOCÓDIGO

ESTRUCTURAS REPETITIVAS ANIDADAS

Es posible insertar un bucle dentro de otro. La estructura interna debe estar incluida totalmente dentro de la externa y no puede existir solapamiento.



**BUCLE ANIDADO
CORRECTAMENTE**



**BUCLE ANIDADO
INCORRECTAMENTE**

CONSTRUCCION DE ALGORITMOS

PROBLEMA N° 11

Diseñar un programa que permita determinar el promedio de tres prácticas después de eliminar la menor nota.