



Pseudocódigo

Ing. Sonia Alexandra Pinzón Nuñez

Tecnología en Sistematización de Datos
Facultad Tecnológica - Universidad Distrital

PSEUDOCODICO: Definición

Esta técnica permite representar el algoritmo mediante un lenguaje más estructurado, facilitando su posterior codificación.

```
Inicio
Instrucción 1
Instrucción 2
Si condición entonces
    Instrucción 3
:
Instrucción n
Fin
```

PSEUDOCÓDIGO: Cómo se Hace?

- 1 Cada instrucción que se va a realizar debe comenzar por un **verbo**, ejemplo: Muestre, Haga, Lea, etc.
- 2 Se debe mantener una **identación** o sangría sobre el margen izquierdo para identificar fácilmente el comienzo y final de las estructuras
- 3 La representación de las **estructuras** son similares u homónimas de los lenguajes de programación, ejemplo: inicio, fin, mientras que, repita_hasta, si_entonces_sino, etc.

PSEUDOCÓDIGO: Cómo se Hace?

Inicio : Denota el punto de inicio del algoritmo.

Leer : Denota la acción de introducir datos o variables desde un dispositivo estándar de entrada.

Imprimir : Representa la acción de enviar datos desde variables a un dispositivo estándar de salida.

Calcular : Denota la realización de cualquier operación aritmética que genere valores para ser almacenados en una variable.

Fin: Denota el punto de finalización del algoritmo.

Pseudocódigo: Diseño

Pseudocódigo

Definición del problema

Análisis del problema

Selección de la mejor alternativa

Algoritmo

Diagramación

Pseudicódigo: Definición del Problema

Está dada por el enunciado del problema, el cuál debe ser claro y completo

Es importante que conozcamos exactamente que se desea.

Mientras que esto no se comprenda, no tiene caso pasar a la siguiente etapa.



Pseudicódigo: Definición del Problema

Ejemplo

Capturar un grupo de números hasta que se digite 0, e indicar cuales de ellos son punto silla.



Un número punto silla es aquel que dado tres valores, el valor central es, es mayor que el de su izquierda y menor que el de su derecha.

Pseudicódigo: Definición del Problema

Ejemplo

Si los números digitados por el usuario son:

5,4,12,15,6,20,32



Los números punto silla serán:

12 y 20

Pseudocódigo: Análisis del Problema

Los datos de **entrada** que nos suministran

El **proceso** que debe realizarse

Los datos de salida o **resultados** que se esperan

El programa deberá mostrar los números punto silla, por lo tanto es mejor ir evaluando cada número.

ALGORITMO: Definición de Constantes y Variables

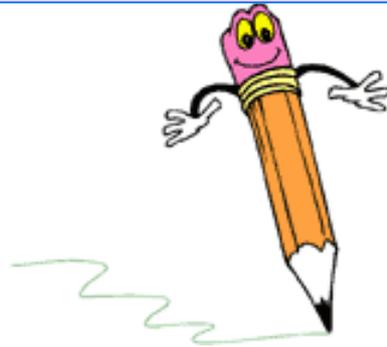
Constantes

No existe un valor dentro del programa que debe permanecer constante

Variables



num,
num_ant,
num_sig, num_m
nroT



PSEUDOCÓDIGO: Ejemplo

Inicio

Haga num=0, num_ant=0, num_sig=0, nroT=0

Mostrar "Digite número" }

Capture num

Mientras que (num != 0)

Haga nroT =nroT+1

caso(nroT)

caso 1: Haga num_ant=num

caso 2: Haga num_m=num

caso 3:

Haga num_sig=num

Si ((num_m>num_ant) y (num_m<num_sig))

Muestre "Es punto silla", num_m

Otro caso

Haga num_ant=num_m;

num_m=num_sig;

num_sig=num;

Si ((num_m>num_ant) y (num_m<num_sig))

Muestre "Es punto silla", num_m

Fin Caso

Muestre "Escriba un Numero para salir 0"

Capture num

Fin Mientras

Si (nroT<3)

Muestre "Debe digitar por lo menos tres numeros"

Fin

Para evitar que el programa ejecute instrucciones con valores no requeridos, se captura antes de comenzar el ciclo

PSEUDOCÓDIGO: Ejemplo

Inicio

Haga num=0, num_ant=0, num_sig=0, nroT=0

Mostrar "Digite número"

Capture num

Mientras que (num != 0)

Haga nroT =nroT+1 }

caso(nroT)

caso 1: Haga num_ant=num

caso 2: Haga num_m=num

caso 3:

Haga num_sig=num

Si ((num_m>num_ant) y (num_m<num_sig))

Muestre "Es punto silla", num_m

Otro caso

Haga num_ant=num_m;

num_m=num_sig;

num_sig=num;

Si ((num_m>num_ant) y (num_m<num_sig))

Muestre "Es punto silla", num_m

Fin Caso

Muestre "Escriba un Numero para salir 0"

Capture num

Fin Mientras

Si (nroT<3)

Muestre "Debe digitar por lo menos tres numeros"

Fin

Es necesario ir contando cuantos números va digitando el usuario, ya que para hallar el punto silla se debe tener al menos tres valores.

PSEUDOCÓDIGO: Ejemplo

Inicio

```
Haga num=0, num_ant=0, num_sig=0, nroT=0
Mostrar "Digite número"
Capture num
Mientras que (num != 0)
  Haga      nroT =nroT+1
  caso(nroT)
  caso 1:   Haga num_ant=num }
  caso 2:   Haga num_m=num
  caso 3:
    Haga num_sig=num
    Si ((num_m>num_ant) y (num_m<num_sig))
      Muestre "Es punto silla", num_m
  Otro caso
    Haga num_ant=num_m;
      num_m=num_sig;
      num_sig=num;
    Si ((num_m>num_ant) y (num_m<num_sig))
      Muestre "Es punto silla", num_m
  Fin Caso
  Muestre "Escriba un Numero para salir 0"
  Capture num
Fin Mientras
Si (nroT<3)
  Muestre "Debe digitar por lo menos tres numeros"
Fin
```

Si el usuario solo ha digitado un valor, ese inicialmente será el numero anterior
num_ant

PSEUDOCÓDIGO: Ejemplo

Inicio

Haga num=0, num_ant=0, num_sig=0, nroT=0

Mostrar "Digite número"

Capture num

Mientras que (num != 0)

Haga nroT = nroT + 1

caso(nroT)

caso 1: Haga num_ant=num

caso 2: Haga num_m=num }

caso 3:

Haga num_sig=num

Si ((num_m > num_ant) y (num_m < num_sig))

Muestre "Es punto silla", num_m

Otro caso

Haga num_ant=num_m;

num_m=num_sig;

num_sig=num;

Si ((num_m > num_ant) y (num_m < num_sig))

Muestre "Es punto silla", num_m

Fin Caso

Muestre "Escriba un Numero para salir 0"

Capture num

Fin Mientras

Si (nroT < 3)

Muestre "Debe digitar por lo menos tres numeros"

Fin

Al digitar un segundo
valor este se
almacenará en
num_m

PSEUDOCÓDIGO: Ejemplo

Inicio

```
Haga num=0, num_ant=0, num_sig=0, nroT=0
Mostrar "Digite número"
Capture num
Mientras que (num != 0)
  Haga      nroT =nroT+1
  caso(nroT)
  caso 1:   Haga num_ant=num
  caso 2:   Haga num_m=num
  caso 3:
    Haga num_sig=num
    Si ((num_m>num_ant) y (num_m<num_sig))
      Muestre "Es punto silla", num_m
  Otro caso
    Haga num_ant=num_m;
      num_m=num_sig;
      num_sig=num;
    Si ((num_m>num_ant) y (num_m<num_sig))
      Muestre "Es punto silla", num_m
  Fin Caso
  Muestre "Escriba un Numero para salir 0"
  Capture num
Fin Mientras
Si (nroT<3)
  Muestre "Debe digitar por lo menos tres numeros"
Fin
```

Al digitar el tercer valor, este será almacenado en **num_sig** y se podrá hacer la correspondiente comparación, para mostrar el número punto silla.

PSEUDOCÓDIGO: Ejemplo

Inicio

```
Haga num=0, num_ant=0, num_sig=0, nroT=0
Mostrar "Digite número"
Capture num
Mientras que (num != 0)
  Haga      nroT =nroT+1
  caso(nroT)
  caso 1:   Haga num_ant=num
  caso 2:   Haga num_m=num
  caso 3:
    Haga num_sig=num
    Si ((num_m>num_ant) y (num_m<num_sig))
      Muestre "Es punto silla", num_m
  Otro caso
    Haga num_ant=num_m;
      num_m=num_sig;
      num_sig=num;
    Si ((num_m>num_ant) y (num_m<num_sig))
      Muestre "Es punto silla", num_m
  Fin Caso
  Muestre "Escriba un Numero para salir 0"
  Capture num
Fin Mientras
Si (nroT<3)
  Muestre "Debe digitar por lo menos tres numeros"
Fin
```

Si el usuario digita más números, se deberán modificar los contenidos de **num_ant**, **num_m** y **num_sig**, usando la opción **otro caso** de la estructura **caso**.

PSEUDOCÓDIGO: Ejemplo

Inicio

```
Haga num=0, num_ant=0, num_sig=0, nroT=0
Mostrar "Digite número"
Capture num
Mientras que (num != 0)
  Haga      nroT =nroT+1
  caso(nroT)
  caso 1:   Haga num_ant=num
  caso 2:   Haga num_m=num
  caso 3:
    Haga num_sig=num
    Si ((num_m>num_ant) y (num_m<num_sig))
      Muestre "Es punto silla", num_m
  Otro caso
    Haga num_ant=num_m;
      num_m=num_sig;
      num_sig=num;
    Si ((num_m>num_ant) y (num_m<num_sig))
      Muestre "Es punto silla", num_m
  Fin Caso
  Muestre "Escriba un Numero para salir 0" }
  Capture num
Fin Mientras
Si (nroT<3)
  Muestre "Debe digitar por lo menos tres numeros"
Fin
```

Antes de volver a evaluar la condición del ciclo, es necesario volver a capturar otro valor, de lo contrario, el programa quedaría en un ciclo indefinido, ya que el número nunca sería igual a 0.

PSEUDOCÓDIGO: Ejemplo

Inicio

Haga num=0, num_ant=0, num_sig=0, nroT=0

Mostrar "Digite número"

Capture num

Mientras que (num != 0)

 Haga nroT =nroT+1

caso(nroT)

caso 1: Haga num_ant=num

caso 2: Haga num_m=num

caso 3:

 Haga num_sig=num

Si ((num_m>num_ant) y (num_m<num_sig))

Muestre "Es punto silla", num_m

 Otro caso

 Haga num_ant=num_m;

 num_m=num_sig;

 num_sig=num;

Si ((num_m>num_ant) y (num_m<num_sig))

Muestre "Es punto silla", num_m

Fin Caso

Muestre "Escriba un Numero para salir 0"

 Capture num

Fin Mientras

Si (nroT<3)

Muestre "Debe digitar por lo menos tres numeros"

Fin

Des pues de haber evaluado el ciclo, es necesario confirmar si el número de valores digitados en realidad fue de al menos tres, de lo contrario se le debe indicar al usuario que sucedió.

PSEUDOCÓDIGO: Código en Lenguaje C

```
#include <stdio.h>
#include <conio.h>
void main()
{   int num,num_m,num_ant,num_sig,nroT=0;
    clrscr();
    printf("Escriba un Numero para salir 0"); scanf("%d",&num);
    while(num!=0)
    {   nroT++;
        switch(nroT)
        { case 1:   { num_ant=num; break;}
          case 2:   { num_m=num; break;}
          case 3:
            { num_sig=num;
              if ((num_m>num_ant) &&(num_m<num_sig))
                printf ("%d es punto silla", num_m);
              break;}
          default:
            { num_ant=num_m;  num_m=num_sig;
              num_sig=num;
              if ((num_m>num_ant) &&(num_m<num_sig))
                printf ("%d es punto silla", num_m);  break; }
        }
        printf("Escriba un Numero para salir 0");  scanf("%d",&num);
    }
    if (nroT<3)
    { printf ("Debe digitar por lo menos tres numeros");  }
    getch();
}
```