

DÍA 6 (5) CLASE JUEVES-06-MAYO-2021

Repaso de algoritmos, entrada, proceso, salida. Repaso de las actividades realizadas (crear algoritmo para cambiar la rueda de un vehículo, cambiar un pañal, cambiar la clave de un correo).

Datos e información (nucleótidos y adn)

Recordamos los **datos primitivos** (byte, short, int, long, float, double, boolean, char)

String: cadena de caracteres

Variables: espacio de memoria del tipo de dato que declaremos, varía a lo largo de la ejecución.

Constante: espacios de memoria que permanecen constante durante la ejecución del programa.

Pseudocódigo: pseudolenguaje de programación.

Estructura secuencial: Una acción sigue a otra en secuencia (como cuando se baila de manera fluida sin intervenir el movimiento, sin pararlo, sin darle otra dirección, dejando que fluya). En la vida sería algo así como sólo seguir el río.

Asignación: paso de valores o resultados a una zona de memoria. Dicha zona será reconocida con el nombre de la variable que recibe el valor.

Asignación simple: consiste en pasar un valor a una variable, $A=2$

Contador: consiste en usarla como un verificador del número de veces que se realiza un proceso, $a=a+1$

Acumuladores: consiste en usarla como un sumador en un proceso, $a=a+b$

Operadores: +, -, *, /, >, <, >=, <=, <>, == (cuando queremos comparar dos valores), %

Lógica proposicional: una expresión que se puede comprobar si es verdadera o falsa.

Operador lógico Y: comparar proposiciones simples, transformándose a compuesta. **Será verdadera siempre cuando todas las proposiciones involucradas sean verdaderas.**

Operador lógico O: permite comparar proposiciones simples, transformándose en una compuesta. **Será verdadera siempre cuando alguna de las proposiciones sea verdadera.**

Lectura: Consiste en leer los valores que nos entregará el usuario (cuando el usuario ingresa los datos)

Escritura (print): enviar por pantalla un resultado o mensaje (lo que programamos)

Estructura de selección SI: en esta estructura de selección, **las instrucciones dependen de una condición para ejecutarse.** El algoritmo analiza el resultado de dicha condición booleana (se cumple o no) y realiza la acción correspondiente a ese resultado. Si no se cumple la condición, no hace nada.

Estructura de selección SI... SINO (selección doble): esta estructura funciona de la misma manera que la selección tradicional, sólo que **en caso de no cumplirse la condición realiza otra acción**, correspondiente a la instrucción2.

Estructura de selección SI... SINO ANIDADA: es **anidable dentro de si** misma.

Estructura de selección SEGÚN (selección múltiple): se evalúa el valor de una variable (valor) y dependiendo del mismo, **se corresponde a uno de los casos de nuestro algoritmo y ejecuta la acción correspondiente al mismo**. En caso de no verificarse ningún caso, la estructura no hace nada, a menos que se agregue un caso para cuando esto ocurra.

Estructuras iterativas: ciclos repetitivos, nos permite **realizar repeticiones de nuestras instrucciones**, de manera que cada acción se realiza varias veces.

Estructura iterativa MIENTRAS: es un bucle que **realizará iteraciones de las instrucciones hasta que deje de cumplirse cierta condición** (puede que nunca se cumpla la condición y la instrucción no se ejecute)

Estructura iterativa HACER MIENTRAS: cumple la misma función que mientras, la diferencia está en que la **instrucción se ejecuta al menos una vez**.

Estructura iterativa PARA: bucle que realizará un **determinado número de iteraciones, bien definidos en un intervalo (desde valor inicial hasta valor final)** y se indica como es el incremento de dicho valor.

¿?

1) Realizar un algoritmo en pseudocódigo que resuelva: Mostrar por pantalla los números de 1 al 100 y del 100 al 1

Inicio
A=1
Print "A"
Si 100>A=>1 HACER
A=A+1
FIN SI

Inicio
A=100
Print "A"
Si 100>=A>1 HACER
A=A-1
FIN SI

Está bien
¿?
Creo que
no porque
debería
usar alguna
estructura
iterativa....

```
sin_titulo>* X
1 Algoritmo sin_titulo
2   variables= n
3   contador=0
4   Escribir "Escribiré los números del 1 al 100"
5   Para contador=contador+1 Hasta 100
6     Escribir contador
7   Fin Para
8 FinAlgoritmo
```

PSeInt - Ejecutando proceso SIN_TITULO

2
3
4
5
6
7
8
9
10
11
12
13
14
15

AHORA SIII!

The screenshot shows the PSeInt IDE with a pseudocode algorithm and its execution output.

```

1  Algoritmo sin_titulo
2      definir contador Como Entero
3      contador=101
4      Para contador=contador-1 hasta 1 Hacer
5          Escribir contador
6      FinPara
7  FinAlgoritmo
8

```

Execution Output (PSeInt - Ejecutando proceso SIN_TITULO):

```

100
99
98
97
96
95
94
93
92
91
90
89
88

```

Buttons: ☐ No cerrar esta ventana ☐ Siempre visible

No me salió de manera fluida, me equivoqué, me costó, tuve que mirar cómo había hecho el ejercicio anterior.

2) Realizar un algoritmo en pseudocódigo que resuelva: Sumar sólo los números pares ingresados por un usuario (la cantidad de números que se ingresan deben ser determinadas por el usuario)

The screenshot shows the PSeInt IDE with a pseudocode algorithm for summing even numbers and its execution output.

```

1  Algoritmo Sumar_sólo_numeros_pares
2      Definir n Como Entero
3      Definir acumulador como entero
4      acumulador=0
5      Escribir "ingrese cantidad de números a sumar", cn
6      Leer cn
7      Repetir
8          Escribir "ingrese el número a sumar", n
9          Leer n
10         Hasta Que n==cn
11         Si n%2=0
12             Escribir "el total es", acumulador= acumulador+n
13         Fin si
14  FinAlgoritmo

```

Execution Output (PSeInt - Ejecutando proceso SUMAR_SÓLO_NUMEROS_PARES):

```

*** Ejecución Iniciada. ***
ingrese cantidad de números a sumar0
> 4
ingrese el número a sumar0
> 1
ingrese el número a sumar1
> 2
ingrese el número a sumar2
> 3
ingrese el número a sumar3
> 4
el total esFALSO
*** Ejecución Finalizada. ***

```

Realmente no pude con esta. Igual me parece extraño que mi variable en número a ingresar sea n, porque y la variable de los demás números va a ser la misma ¿? Y no me resulta sumar...

```

1  Algoritmo sumar_sólo_nos_pares
2  definir suma Como Entero
3  suma=0
4  definir contador Como Entero
5  contador=0
6  definir var Como Entero
7  var=n
8  Escribir "ingrese cantidad de números que desea sumar", cn
9  Leer cn
10 Para contador=contador+cn Hasta 1
11     Escribir "ingrese número a sumar", n
12     Leer n
13     Si n%2==0 Entonces
14         Escribir, n=suma+n
15     Sino n=n
16     FinSi
17 FinPara
18 FinAlgoritmo
19

```

PSelnt - Ejecutando proceso SUMAR_SÓLO_NOS_PARES

```

ingrese cantidad de números que desea sumar0
> 4
ingrese número a sumar0
> 4
VERDADERO
ingrese número a sumar4
> 3
ingrese número a sumar3
> 2
VERDADERO
ingrese número a sumar2
> 1
*** Ejecución Finalizada. ***

```

☐ No cerrar esta ventana ☐ Siempre visible

Sigue sin resultarme pero siento que está mejor planteada que la primera, no sé, estoy frustrada.

```

1  Algoritmo sumar_sólo_nos_pares
2  definir suma Como Entero
3  acumulador=0
4  definir contador Como Entero
5  contador=0
6  definir var Como Entero
7  var=n
8  Escribir "ingrese cantidad de números que desea sumar", cantidad_numeros
9  Leer cantidad_numeros
10 Para contador=contador+cantidad_numeros Hasta 1
11     Escribir "ingrese número a sumar", numero_a_sumar
12     Leer numero_a_sumar
13     Si numero_a_sumar%2==0 Entonces
14         acumulador=acumulador+numero_a_sumar
15     Sino numero_a_sumar=numero_a_sumar
16     FinSi
17 FinPara
18 Escribir "la suma de sus números pares es", acumulador
19 FinAlgoritmo
20

```

PSelnt - Ejecutando proceso SUMAR_SÓLO_NOS_PARES

```

ingrese cantidad de números que desea sumar0
> 5
ingrese número a sumar0
> 1
ingrese número a sumar1
> 3
ingrese número a sumar3
> 2
ingrese número a sumar2
> 8
ingrese número a sumar8
> 0
la suma de sus números pares es10

```

☐ No cerrar esta ventana ☐ Siempre visible

PUDE
PORFINNN
! Ahora
sólo
necesito
arreglar el
número
que me
sale luego
de sumar.

3) Realizar un algoritmo en pseudocódigo que resuelva: Sumar los números pares e impares ingresados por un usuario y contar la cantidad de números pares e impares (la cantidad de números que se ingresan debe ser determinada por el usuario)

```
4  Algoritmo contarsumar_números_pares_impares
5      contador=0
6      acumulador=0
7      Escribir "ingrese la cantidad de números que desea sumar", cantidad_numeros
8      Leer cantidad_numeros
9      Para contador=cantidad_numeros hasta 1 Hacer
10         Escribir "ingrese un número", numero_n
11         Leer numero_n
12         Si numero_n%2==0 Entonces
13             acumulador=acumulador+numero_n
14             contador=contador+numero_n
15         FinSi
16     FinPara
17     Escribir "la suma de los números pares es" acumulador
18     Escribir "y son" contador
19 FinAlgoritmo
```

Se me hizo más fácil escribir el código pero quedé en un bucle infinito. Lo veré más tarde porque estoy tired.

```
Algoritmo contarsumar_números_pares_impares
    Definir contador como entero
    contador=0
    definir acumulador como entero
    acumulador=0
    definir cantidad_numeros Como Entero
    Escribir "ingrese la cantidad de números que desea sumar", cantidad_numeros
    Leer cantidad_numeros
    Para contador=contador+cantidad_numeros hasta 1 Hacer
        Escribir "ingrese un número", numero_n
        Leer numero_n
        Si numero_n%2==0 Entonces
            acumulador=acumulador+numero_n
            contador=contador+1
        FinSi
    FinPara
    Escribir "la suma de los números pares es" acumulador
    Escribir "y son" contador
FinAlgoritmo
```

Tengo problemas con el contador, estoy segura que es eso, pero no sé bien qué en específico 😞

```

3 //y contar la cantidad de números pares e impares (la cantidad de números que se ing
4 Algoritmo contarsumar_números_pares_impares
5     Definir contadora como entero
6     contadora=0
7     Definir contadorb Como Entero
8     contadorb=0
9     definir contadorc Como Entero
10    contadorc=0
11    definir acumuladorb como entero
12    acumuladorb=0
13    definir acumuladorc Como Entero
14    acumuladorc=0
15    definir cantidad_numeros Como Entero
16    Escribir "ingrese la cantidad de números que desea sumar", cantidad_numeros
17    Leer cantidad_numeros
18    Para contadora=contadora+1 hasta cantidad_numeros Hacer
19        Escribir "ingrese un número", numero_n
20        Leer numero_n
21        Si numero_n%2==0 Entonces
22            acumuladorb=acumuladorb+numero_n
23            contadorb=contadorb+1
24        SiNo
25            acumuladorc=acumuladorc+numero_n
26            contadorc=contadorc+1
27        FinSi
28    FinPara
29    Escribir "la suma de los números pares es" acumuladorb
30    Escribir "y son" contadorb
31    Escribir "la suma de los números impares es" acumuladorc
32    Escribir "el total de números impares son", contadorc
33 FinAlgoritmo

```

```

*** Ejecución Iniciada. ***
ingrese la cantidad de números que desea sumar0
> 5
ingrese un número0
> 1
ingrese un número1
> 8
ingrese un número8
> 2
ingrese un número2
> 1
ingrese un número1
> 1
la suma de los números pares es10
y son2
la suma de los números impares es3
el total de números impares son3
*** Ejecución Finalizada. ***
☐ No cerrar esta ventana ☐ Siempre visible 

```

El profe me hizo notar el error (necesitaba 3 contadores diferentes, 1 para el proceso si entrara en bucle tro para los pares y otro para los impares)

4) Realizar un algoritmo en pseudocódigo que resuelva: Se ingresa una cantidad de números, calcular la suma de todos los números pares ingresados y la multiplicación de todos los impares ingresados, además determinar si de los números ingresados fueron más pares o impares (la cantidad de números que se ingresan deben ser determinada por el usuario.

```

1 //Se ingresa una cantidad de números
2 //calcular la suma de todos los números pares ingresados y la
3 //multiplicación de todos los impares ingresados, además determinar si d
4 // fueron más pares o impares
5 Algoritmo completo
6     contador=0
7     acumuladorsuma=0
8     acumuladormultiplicacion=1
9     Escribir "ingrese cantidad de números deseada", cantidad_numeros
10    Leer cantidad_numeros
11    Para contador=contador+1 hasta cantidad_numeros
12        Escribir "ingrese número", numero_ingresado
13        Leer numero_ingresado
14        Si numero_ingresado%2==0 Entonces
15            acumuladorsuma=acumuladorsuma+numero_ingresado
16        Sino
17            acumuladormultiplicacion=acumuladormultiplicacion*numero_ingresado
18        FinSi
19    FinPara
20    escribir "la suma de los números pares ingresados es", acumuladorsuma
21    Escribir "la multiplicación de los números impares ingresados es", acumuladormultiplicacion
22 FinAlgoritmo

```

```

ingrese cantidad de números deseada0
> 4
ingrese número0
> 3
ingrese número3
> 28
ingrese número28
> 2
ingrese número2
> 3
la suma de los números pares ingresados es30
la multiplicación de los números impares ingresados es9
*** Ejecución Finalizada. ***
☐ No cerrar esta ventana ☐ Siempre visible

```

Me sorprende lo rápido que resolví este

5) Realizar un algoritmo en pseudocódigo que resuelva: determine el mayor de N números ingresados por el usuario.





```
Algoritmo determinarMayorDeNNumerosIngresadosPorUsuario
  Definir cantidad_numeros Como Entero
  definir contador como entero
  contador=0
  numeroactual=0
  Escribir "ingrese la cantidad de números a escribir", cantidad_numeros
  Leer cantidad_numeros
  Para contador=contador+1 hasta cantidad_numeros hacer
    Escribir "ingrese número", numeroingresado
    Leer numeroingresado
    Si numeroingresado>numeroactual
      numeroactual=numeroingresado
    FinSi
  FinPara
  Escribir "el mayor de los números ingresados es", numeroactual
FinAlgoritmo
```

```
PSeInt - Ejecutando proceso DETERMINARELMAYORD
*** Ejecución Iniciada. ***
ingrese la cantidad de números a escribir0
> 3
ingrese número0
> 6
ingrese número6
> 3
ingrese número3
> 1
el mayor de los números ingresados es6
*** Ejecución Finalizada. ***
☐ No cerrar esta ventana ☐ Siempre visible
```

El truco estaba en igualar el número actual al número ingresado (: aunque creo que es trampa colocar el número actual en 0 xd

DIAGRAMA DE FLUJO: Es la **representación gráfica de un algoritmo**. También se puede decir que es la representación detallada en forma gráfica de cómo deben realizarse los pasos en la computadora para producir resultados. Esa representación gráfica se da cuando varios símbolos (que indican diferentes procesos en la computadora), se relacionan entre sí mediante líneas que indican el orden en que se deben ejecutar los procesos.

Simbología

Símbolo	Significado
	Terminal /Inicio.
	Entrada de datos.
	Proceso.
	Decisión.

Simbología

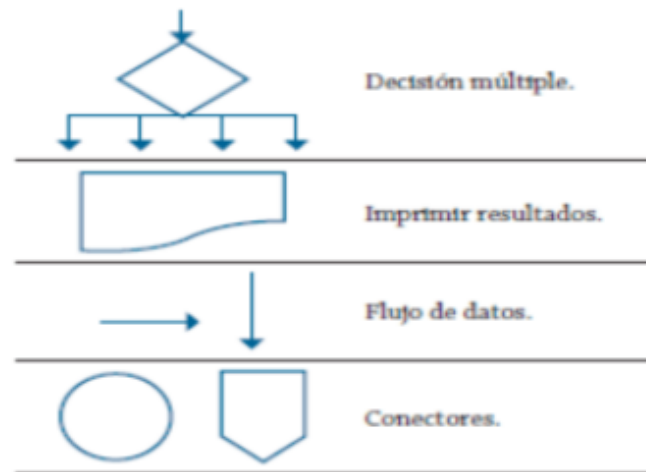
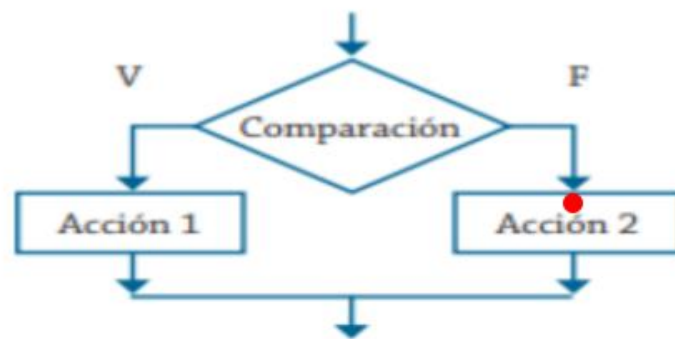


Diagrama de flujo – Estructura selectiva



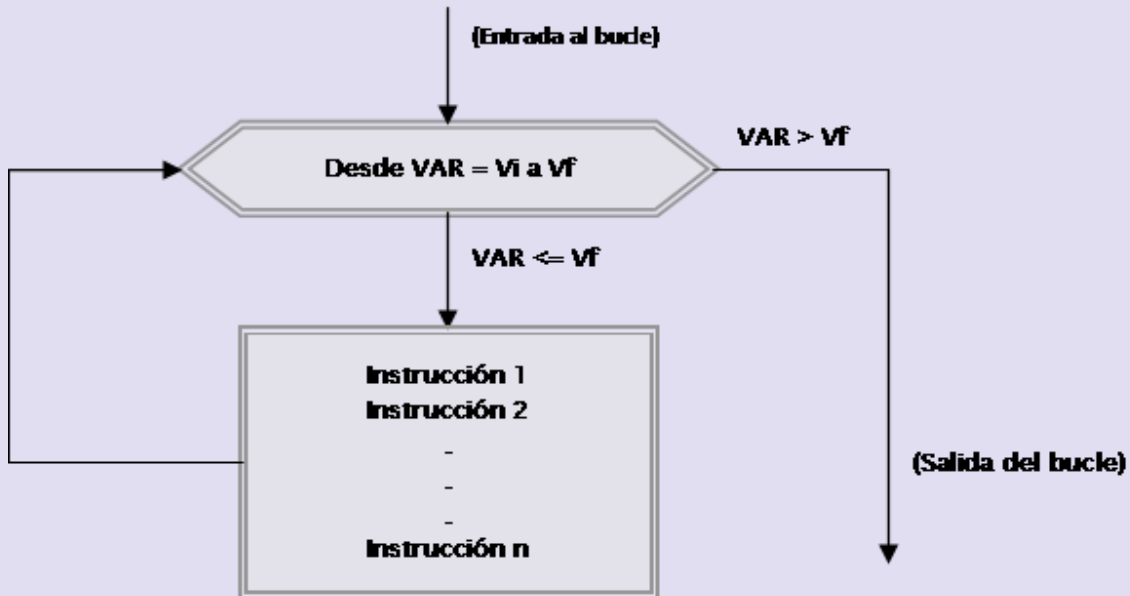
Simbología - Ejemplo



FIN DE CLASE-

POST CLASE-

ESTRUCTURAS DE REPETICIÓN (BUCLES)



CONTADOR:

Es una **variable cuyo valor se incrementa o decrementa** en una cantidad constante cada vez que se produce un determinado suceso o acción. Los contadores se utilizan con la finalidad de contar sucesos o acciones internas de un bucle.

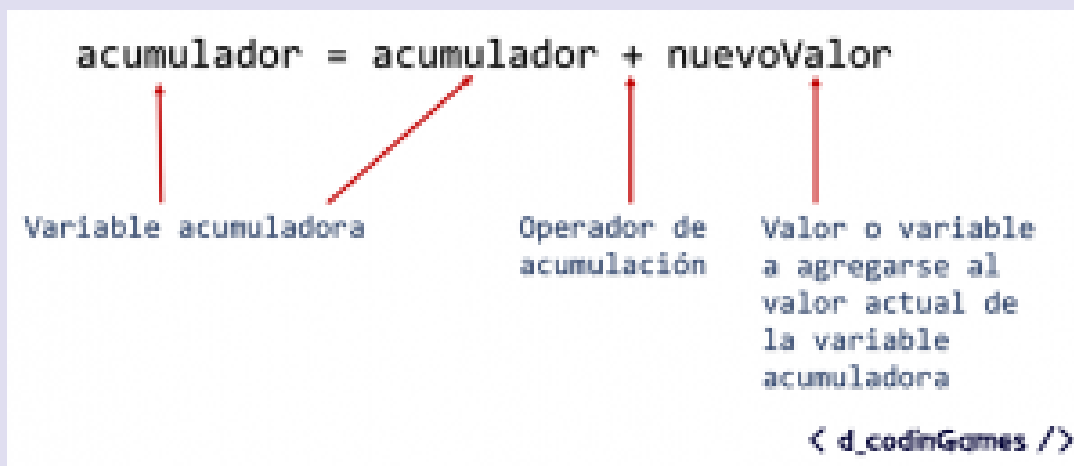
Es importante **realizar una operación de inicialización** y posteriormente las sucesivas de **incremento o decremento** del mismo. La inicialización consiste en **asignarle al contador un valor con el que arrancará el proceso de conteo**. Este proceso **deberá situarse "antes y fuera del bucle"**.

Introducir 5 número y contar los números pares.

```
Proceso ContarPares
  Definir var,cont,num como Entero;
  cont<-0;
  Para var<-1 Hasta 5 Hacer
    Escribir Sin Saltar "Dime un número:";
    Leer num;
    Si num % 2 = 0 Entonces
      cont<-cont+1;
    FinSi
  FinPara
  Escribir "Has introducido ",cont," números pares.";
FinProceso
```

ACUMULADOR:

Variable que se utiliza para sumar valores. Se utiliza normalmente dentro de un ciclo pero cambiamos su valor sumándole una variable, es decir, no siempre se le suma la misma cantidad.



Introducir 5 número y sumar los números pares.

```
Proceso SumarPares
    Definir var,suma,num como Entero;
    suma<-0;
    Para var<-1 Hasta 5 Hacer
        Escribir Sin Saltar "Dime un número:";
        Leer num;
        Si num % 2 = 0 Entonces
            suma<-suma+num;
        FinSi
    FinPara
    Escribir "La suma de los números pares es ",suma;
FinProceso
```

En el caso de un programa requiera el uso de contadores o acumuladores, es importante inicializar los mismos. Normalmente cuando se desea efectuar sumatorias, el valor se lo debe iniciar en 0 (*cero*). Si se desea totalizar multiplicaciones, el valor se lo debe iniciar en 1 (*uno*).

Entendí! (porfin) el contador y el acumulador, aunque tengo que practicarlos porque es donde más cometo errores.

Utilizando PSeint me está quedando más claro ya que puedo verificar en tiempo real cómo se ejecuta las instrucciones que doy, y si es que tengo algún error, maravilloso (: Aunque he hecho cosas muy sencillas, me falta practicar un montón en esto. Sigo luchando con los ejercicios de esta clase. Desde ahora comenzaré a trabajar en JAVA, iré dejando Pseint de a poco.

Fue una clase difícil, luché mucho con los primeros ejercicios, realmente horas, hasta soñé que resolvía uno jaja, luego, ya los últimos, me sorprendió la rapidez con que los pude resolver, aún no sé cómo los hice tan rápido xd

