

Pseudocódigo Español (PSeInt):

Contenido

- Introducción 2
- Estructuras de Control: 2
 - Entrada y salida de información 2
 - Estructuras selectivas 2
 - Selectiva Simple (if) 2
 - Selectiva Doble (if-else)..... 2
 - Selectiva Múltiple (switch) 3
 - Estructuras Iterativas 3
 - Bucle con condición (while) 3
 - Bucle con condición de entrada (do-while) 3
 - Bucle for 4
 - Bucle for-each 4
- Funciones: 5
 - Creación y Llamada: 5
 - Arrays: 5

Introducción

Antes de empezar a programar, es ideal comenzar con una pequeña idea de qué se puede hacer, en estos casos se puede aplicar el pseudocódigo, que es, dicho de una forma muy sencilla, un medio para programar usando palabras de nuestro idioma. La aplicación más conocida para pseudocódigo es PSeInt.

En pseudocódigo no es necesario crear y declarar variables antes de usarse como es el caso de otros lenguajes, y puede ir variando según las necesidades de cada persona.

```
Algoritmo NotasAlumno  
[...]  
FinAlgoritmo
```

Estructuras de Control:

Todas las estructuras de control (salvo las de entrada y salida), al igual que los algoritmos y funciones, empiezan donde se encuentra su nombre y acaban en la marca de Fin (en el caso de los algoritmos, sería FinAlgoritmo)

Entrada y salida de información

Son dos simples, “Escribir” y “Leer”:

- Con “Escribir”, se imprime una variable o un dato dado en pantalla
- Con “Leer”, se guarda lo que se escriba por el teclado en una variable

```
Escribir "Escribe un número"  
Leer num
```

Estructuras selectivas

Selectiva Simple (if)

La estructura en Pseudocódigo es: Si + condición + [Entonces] + (instrucciones) + FinSi

```
Escribir "Escribe un número"  
Leer num  
Si num>0 Entonces  
    Escribir num  
FinSi
```

En este caso, escribiría solo los números positivos.

Selectiva Doble (if-else)

Es igual que la estructura selectiva simple, añadiendo un SiNo al final:

```
Escribir "Escribe nota del alumno"  
Leer nota  
si (nota<5) Entonces  
    Escribir "El alumno ha suspendido con un " nota  
SiNo  
    Escribir "El alumno ha aprobado con un " nota  
FinSi
```

Se pueden anidar estructuras selectivas (Si – SiNo Si SiNo FinSi FinSi)

Selectiva Múltiple (switch)

La estructura es Según + condición o variable + Hacer + casos. Se pueden anidar casos.

Un ejemplo sería el siguiente, con los meses:

```
Algoritmo Dias_Del_Mes
  Escribir "Escribe el número del mes que quieras"
  Leer num_mes
  si ((1<=num_mes) y (num_mes<=12)) Entonces
    Segun num_mes Hacer
      1: mes = "Enero"
      2: mes = "Febrero"
      3: mes = "Marzo"
      4: mes = "Abril"
      5: mes = "Mayo"
      6: mes = "Junio"
      7: mes = "Julio"
      8: mes = "Agosto"
      9: mes = "Septiembre"
      10: mes = "Octubre"
      11: mes = "Noviembre"
      12: mes = "Diciembre"
    FinSegun
    Segun num_mes Hacer
      1, 3, 5, 7, 8, 10, 12: dias = 31
      2: dias = 28
      4, 6, 9, 11: dias = 30
    FinSegun
    Escribir "Has elegido el número " num_mes ", el mes es " mes " y
tiene " dias " días."
  Sino
    escribir "Número inválido"
  FinSi
  Escribir "Fin del algoritmo"
FinAlgoritmo
```

Estructuras Iterativas

Bucle con condición (while)

La estructura principal es Mientras + condición + Hacer, seguido de un FinMientras al final.

Hasta que no se incumpla la condición, no se saldrá del bucle

```
Escribir "Escriba la nota del alumno."
Leer nota
Mientras (0>nota) o (nota>10)
  Escribir "Escriba una nota válida"
  Leer nota
FinMientras
```

Bucle con condición de entrada (do-while)

La estructura principal es Repetir + instrucciones + Hasta Que + condición. La diferencia con el bucle de condición es que este entra al menos una primera vez al código, y luego ya lee la condición

```
Hacer
  Escribir "Escriba una nota válida para el alumno"
  Leer nota
Hasta Que (nota>0 y nota<10)
```

Bucle for

A diferencia de Java, el bucle for en Pseudocódigo tiene la siguiente estructura: Para número Hasta número Con Paso número + instrucciones. Sigue más o menos el mismo concepto, salvo que en el for de java no es estrictamente necesario que llegue a x número, en este es lo único viable.

```
Para i=1 hasta 100 Con Paso 1
    si i%2=0 o i%3=0
        Escribir i ", " Sin Saltar
    FinSi
FinPara
```

Bucle for-each

Su estructura es Para Cada elemento de {Objeto o Conjunto} Hacer + instrucciones + Fin Para Cada.

Es un for que recorre todo el conjunto que se le dé.

```
Para Cada elemento de array Hacer
    Escribir elemento ", " Sin Saltar
Fin Para Cada
Escribir ""
```

Funciones:

Las funciones son aquellos algoritmos que se encuentran fuera del principal, y que solo se activan por medio de **llamadas** a estos. Hay dos tipos de funciones: Las funciones que no devuelven valor y las que sí lo devuelven.

Creación y Llamada:

Para las funciones normales, se crean simplemente con Funcion + nombre + instrucciones + FinFuncion, y se llaman poniendo su nombre.

```
Funcion numespar(num)
    si num%2=0
        Escribir num " es par"
    SiNo
        Escribir num " es impar"
    FinSi
FinFuncion

Algoritmo ParOImpar
    Escribir "Escribe un número entero"
    leer num
    numespar(num)
FinAlgoritmo
```

Para las funciones que devuelven valores, se crea dándole un valor a la función, y para llamarla no sirve simplemente con poner su nombre, sino un una variable tomará un valor:

```
Funcion absol<-sacarabsoluto(num)
    absol=-num
FinFuncion

Algoritmo numeroabsoluto
    Escribir "Escribe un número"
    Leer num
    Si num<0
        absol= sacarabsoluto(num)
    SiNo
        absol=num
    FinSi
    Escribir "El valor absoluto de " num " es " absol
FinAlgoritmo
```

Arrays:

Los arrays se crean por medio del comando Dimension nombre[numero], donde "numero" son los espacios que tiene. Para hacer referencia a un hueco exacto, se llama al array y se especifica el hueco a observar:

```
Dimension array[2]
array[1]=10
array[2]=40
Escribir array[1]
```