

Pseudocódigo

El pseudocódigo es la representación narrativa (no hay reglas sintácticas estrictas) de un algoritmo, escrita en lenguaje natural utilizando las estructuras de control típicas de algún Lenguaje de Programación y algunos símbolos algebraicos.

Ejemplo:

Reglas:

En ALGORITMO indicamos el nombre del programa, y justo después debemos declarar las variables que serán usadas en el programa (con VARIABLE) indicando su tipo de dato (un número de tipo ENTERO, en nuestro ejemplo).

Entre INICIO y FIN escribiremos las diferentes acciones que irá realizando el programa: usamos ESCRIBIR() para indicar que se mostrará un mensaje por pantalla (el texto que se deba mostrar 'tal cual' debe ir entre paréntesis), y LEER() para pedir un dato al usuario.

Fíjate también en que cada línea termina en punto y coma (menos INICIO y FIN).

Ejercicio: **Pedir dos números y mostrar la multiplicación de ambos.**

1. ALGORITMO Sumar;
2. VARIABLE
3. ENTERO Numero1, Numero2, Resultado;
4. INICIO
5. ESCRIBIR("Dime dos números para multiplicar: ");
6. LEER(Numero1, Numero2);
- 7.
8. Resultado <- Numero1 * Numero2;
- 9.
10. ESCRIBIR("La respuesta es: ", Resultado);
11. FIN

En este caso observa cómo declaramos varias variables de tipo **ENTERO** (también se podrían haber puesto en una línea cada una, seguida de **VARIABLE**) y cómo hemos hecho para solicitar varios valores a la vez con **LEER()**.

También hemos realizado una operación aritmética (sumar los números) y que usamos '<-' para asignar el resultado a la variable 'Resultado'.

Ejercicios de clase:

- 1) Pedir un número y mostrarlo por pantalla.
- 2) Mostrar por pantalla el triple de un número entero introducido por el usuario.
- 3) Pedir el lado del cuadrado y mostrar el perímetro.

Respuestas:

1) Pedir un número y mostrarlo por pantalla.

```

1. ALGORITMO Mostrar;
2. VARIABLE
3.   ENTERO entrada;
4. INICIO
5.   ESCRIBIR("Dame un número");
6.   LEER( entrada );
7.   ESCRIBIR( entrada );
8. FIN

```

2) Mostrar por pantalla el triple de un número entero introducido por el usuario.

```

1. ALGORITMO Multiplicar;
2. VARIABLE
3.   ENTERO a, y;
3. INICIO
4.   ESCRIBIR("Dame un número");
5.   LEER( a );
6.   y <- a * 3;
7.   ESCRIBIR(y);
8. FIN

```

3) Pedir el lado del cuadrado y mostrar el perímetro.

```

1. ALGORITMO Triple;
2. VARIABLE
3.   ENTERO Número Resultado;
3. INICIO
4.   ESCRIBIR("Dame un número");
5.   LEER( Número );
6.   Resultado <- Número * 4;
7.   ESCRIBIR(Resultado);
8. FIN

```

Estructuras de control

Las estructuras de control deciden qué camino hay que seguir en función de una condición. Aquí veremos algunas de ellas:

Estructura secuencial: consiste en colocar una instrucción tras otra, de manera que se van ejecutando de arriba abajo (Ejercicios anteriores).

Estructura selectiva o condicional: permiten ejecutar un conjunto de instrucciones u otras en función de si se cumple o no una condición.

```

si (se cumple la condición)
inicio
  conjunto de acciones;
fin
sino
inicio
  conjunto de acciones;
fin

```

Estructura iterativa o de repetición: permite repetir una instrucción o grupo de ellas un número fijo de veces o mientras (o hasta que) una condición sea cierta.

Mientras (se cumpla la condición)

```
inicio
conjunto de acciones;
fin
```

```
N=1;           <- base a incrementar (desde)
Repetir
inicio
  conjunto de acciones
  n=n+1;       <- acumulador
fin
Mientras (n<=cantidad de iteraciones)  <- condición de fin de iteración (hasta)
Fin
```

Ejercicio: Pedir dos números enteros y mostrar 'Verdadero' si el primero es mayor.

1. ALGORITMO Verdadero;
2. VARIABLE
3. ENTERO num1, num2;
4. INICIO
5. ESCRIBIR("Dime dos números");
6. LEER(num1, num2);
7. SI(num1 > num2)
8. ESCRIBIR("Verdadero");
9. FIN SI
10. FIN

Uso del mientras:

1. ALGORITMO Mientras;
2. VARIABLE
3. ENTERO Contador;
4. INICIO
5. Contador <- 1;
6. MIENTRAS(Contador <= 100) HACER
7. ESCRIBIR(Contador);
8. Contador <- Contador + 1;
9. FIN MIENTRAS
10. FIN

Ejercicios de clase:

- 1) Pedir tres números y deducir si se han introducido en orden creciente.
- 2) Pedir dos valores y en caso de que no sean iguales indicar cuál es el mayor.
- 3) Pedir un número al usuario y mostrar el nombre del día al corresponde (1=lunes).

Respuestas:

1) Pedir tres números y deducir si se han introducido en orden creciente

1. ALGORITMO Orden;
2. VARIABLE
3. ENTERO a, b, c ;
4. INICIO
5. ESCRIBIR("Dame tres números");
6. LEER(a, b, c);
7. SI (a < b) AND (b < c)
8. ESCRIBIR("En orden creciente");
9. SINO
10. ESCRIBIR("En orden no creciente");
11. FIN SI
12. FIN

2) Pedir dos valores y en caso de que no sean iguales indicar cuál es el mayor.

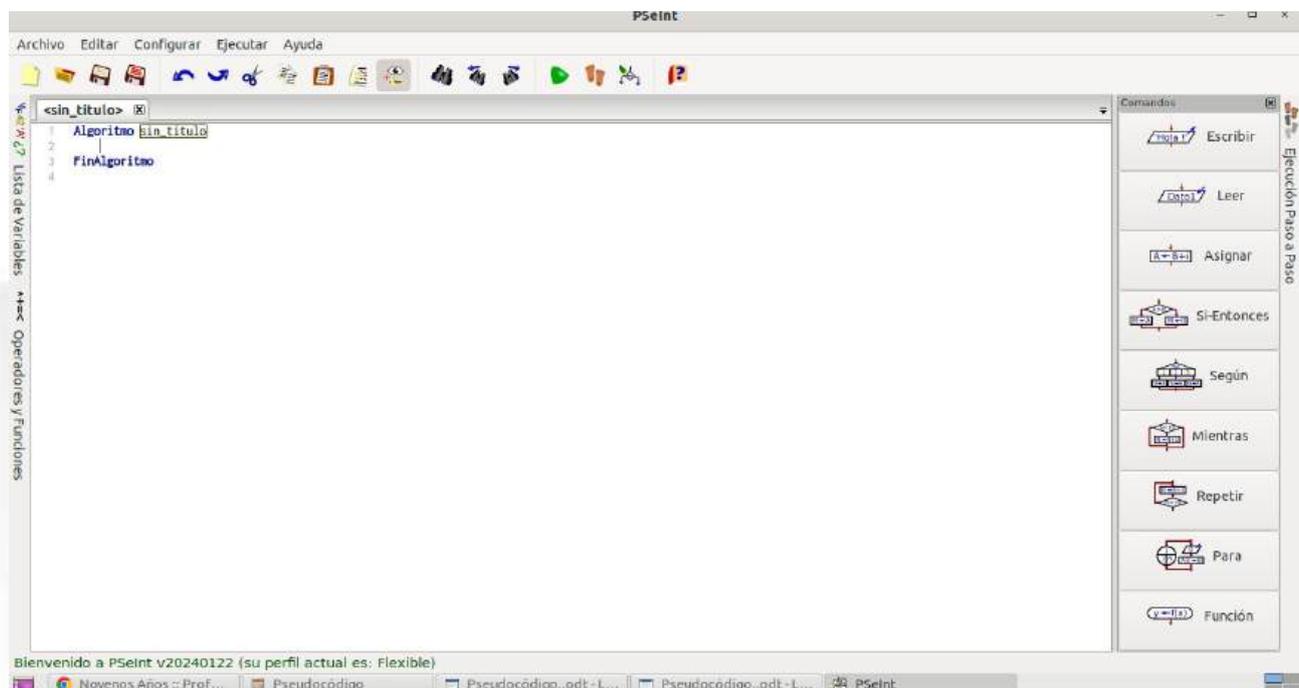
1. ALGORITMO Iguales;
2. VARIABLE
3. ENTERO x, y;
4. INICIO
5. ESCRIBIR("Dame dos números");
6. LEER(x, y);
7. SI(x == y)
8. ESCRIBIR("Son iguales");
9. SINO
10. SI(x > y)
11. ESCRIBIR("x es mayor");
12. SINO
13. ESCRIBIR("y es mayor");
14. FIN SI
15. FIN_SI
16. FIN

3) Pedir un número al usuario y mostrar el nombre del día al corresponde (1=Lunes).

1. ALGORITMO DIA_CASO;
2. VAR
3. ENTERO d;
4. INICIO

```
5.  ESCRIBIR("Introduce un número del 1 al 7");
6.  LEER( d );
7.  SI (d >=1 ) AND (d <= 7)
8.      EN_CASO_DE d HACER
9.          1: ESCRIBIR("Lunes");
10.         2: ESCRIBIR("Martes");
11.         3: ESCRIBIR("Miércoles");
12.         4: ESCRIBIR("Jueves");
13.         5: ESCRIBIR("Viernes");
14.         6: ESCRIBIR("Sábado");
15.         7: ESCRIBIR("Domingo");
16.     FIN CASO
17. SINO
18.     ESCRIBIR("El valor no es válido");
19. FIN SI
20. FIN
```

Uso de PseInt



Operadores utilizados

Matemáticos: + suma, - resta, * multiplicación, / división.

De comparación: > mayor, < menor, <= menor o igual, >= mayor o igual, % módulo.

Algunas sentencias y estructuras

Escribir “ texto”: permite pedir por teclado el ingreso de datos.

Leer variable: permite guardar el dato ingresado por teclado en una variable.

Mostrar “Texto”: permite visualizar en pantalla el texto o contenido de variable (Solo el texto debe ir entre comillas).

Limpiar pantalla: borra de pantalla lo indicado hasta esa línea.

// permite poner comentarios (Solo lo ve el programador)

Si condición

 acción si la condición es verdadera

Sino

 acción si la condición es verdadera

FinSi

Mientras condición **hacer**

 acción si la condición es verdadera

FinMientras

Para variableinicio ← número inicio **hasta** variablefin o número

 acción si la no se llevo al final

FinPara

Segun variable **hacer**

1: acción si la opción elegida es 1

2: acción si la opción elegida es 2

3:

FinSegun

Ejemplo en Pseudocódigo

Pedir un número y decir si es par o impar.

ALGORITMO Pares e impares

 Escribir "Ingrese un número"

 Leer num

 Si $\text{num mod } 2 = 0$

 Mostrar "Par"

 Sino

 Mostrar "Impar"

 FinSi

FinALGORITMO