

# Funciones / Subprocesos en pseint

Las funciones también llamadas Subproceso o SubAlgoritmo, es una instrucción que permite agrupar variables y sentencias, cuya finalidad es la de ejecutar una tarea específica, se pueden añadir una o varias funciones, en un algoritmo, las funciones son subprogramas dentro de un programa, que se pueden invocar (ejecutar) desde cualquier parte del programa, es decir, desde otra función, desde la misma función o desde el programa principal, cuantas veces sea necesario.

## Sintaxis:

```
SubAlgoritmo variable_de_retorno <- Nombre ( Argumentos )
```

```
Fin SubAlgoritmo
```

```
Algoritmo ejemplo
```

```
FinAlgoritmo
```

- **variable\_de\_retorno:** si el subproceso o función, calcula y devuelve un valor, se debe colocar el nombre de la variable que se utilizará para almacenar ese valor; si el subproceso o función no devuelve nada se puede eliminar la variable con su flecha de asignación.
- **nombre:** es el nombre que recibirá el subproceso o función.
- **Argumentos:** los argumentos son variables que requiere la función, utilizando comas para separarlos, si el subproceso o función no requiere argumentos puede dejarse en blanco, y de manera opcional omitir los paréntesis.

Ejemplos de Declaración de una función, subproceso y subalgoritmo :

Funcion correr()	SubProceso ejercicio()
FinFuncion	FinSubProceso
Algoritmo atletas	Algoritmo Ejercicios
FinAlgoritmo	FinAlgoritmo
SubAlgoritmo ejemplo()	
FinSubAlgoritmo	
Algoritmo Ejemplos	
FinAlgoritmo	

Una función, desde el punto de vista de la programación, se define como un proceso en el cual se establece acciones a realizar, siendo invocado en el algoritmo principal, si la funcion o subprocesso no recibe ningún valor pueden colocarse los paréntesis vacíos u omitirse. Las reglas para los nombres de subprocessos, variables de retorno y argumentos son las mismas que para cualquier identificador en pseudocódigo. **ejemplos;**

### **Ejemplo 01;**

```

Funcion datos()
Escribir "Ingrese su nombre"
Leer nombre
Escribir "Ingrese su edad "
Leer edad
FinFuncion
Algoritmo usuario
datos
FinAlgoritmo

```

### **Ejemplo 02;**

```

SubProceso s<-sumar()
s<-10+5
FinSubProceso

```

Algoritmo suma

Escribir "La suma es : " sumar

FinAlgoritmo

ADVERTISEMENT

[REPORT THIS AD](#)

Existen variantes para la estructura de una función. Si la función no retorna ningún valor, como el ejemplo 2, pueden omitirse la variable de retorno y el signo de asignación. Es decir, se puede colocar directamente el nombre de la función, de esta manera solo ejecuta las acciones programadas.

## Parametros

Las funciones o procesos también pueden recibir valores de entrada (llamados parámetros) y el cual retorna un valor resultado. Además, de manera opcional se le puede agregar las palabras claves Por Valor o Por Referencia para indicar el tipo de parámetro en cada argumento.

### Parámetro por valor:

mientras que el pasaje por valor implica que la función opera con una copia de la variable (o el resultado de la expresión) que se utilizó en la llamada, por lo que las modificaciones que aplique la función no se verán reflejadas fuera de la misma.

### Ejemplo 3;

SubProceso d<-doble(numero)

Definir d como Real

d<-numero\*2

FinSubProceso

Algoritmo Ejemplo

Escribir "Ingresa un numero a evaluar"

Leer n

Escribir doble(n)

### FinAlgoritmo

Cuando se crea una función con parámetros por valor; seguida de la variable de retorno, es opcional el uso de su tipo identificador que es la palabra reservada **por valor**, después de ser invocado o llamado el subproceso, se le tiene que asignar el valor de la variable.

#### Ejemplo 04;

```
funcion variable( num por valor)
```

```
Escribir num
```

```
finfuncion
```

```
Algoritmo mostrar_valor
```

```
variable(55)
```

```
FinAlgoritmo
```

Si no se indica, los arreglos se pasan por referencia, las demás expresiones por valor.

#### Parámetro por referencia:

Las variables por referencia declaradas en una función, deben ser llamadas tal como se ha declarado.

El parámetro por referencia implica que si la función modifica el argumento, se modificará en realidad la variable que se utilizó en la llamada

#### Ejemplo 05;

```
funcion variable( num Por referencia)
```

```
Escribir num
```

```
finfuncion
```

```
Algoritmo mostrar_valor
```

```
num<-33
```

```
variable(num)
```

```
FinAlgoritmo
```

Para invocar a la función se debe utilizar su nombre y entre paréntesis los parámetros, que podrán ser expresiones sólo si el tipo de parámetro es por referencia. Una llamada puede ser en sí una instrucción, pero si la función retorna algún valor, también puede utilizarse como operando dentro de una expresión.