

SUBPROGRAMAS EN PSEINT





Objetivos de la Guía

En esta guía aprenderemos a:

- Separar el Algoritmo principal de las Funciones y SubProgramas.
- Diferenciar una función de un subprograma.
- Comprender qué debe ejecutarse en una función o subprograma.
- Lograr enviar información a las funciones o subprogramas a través de parámetros por valor o por referencia.
- Diferenciar pasaje por valor y por referencia.
- Llamar funciones o subprogramas desde el Algoritmo Principal.
- Definir variables de retorno y operar con ellas.
- Utilizar estructuras de control en Funciones y Subprogramas.

PROCEDIMIENTOS

Ahora que sabemos que las **funciones devuelven un valor** para que lo usemos a través de la variable de retorno, vamos a ver un caso distinto. Los **procedimientos** son similares, pero **no devuelven ningún valor**, sólo realizan una tarea.

¿QUÉ ES UN PROCEDIMIENTO?

Un **procedimiento** es un subprograma que ejecuta un proceso específico. En PseInt lo llamaremos **SubProceso**. Ningún valor está asociado con el nombre del procedimiento; por consiguiente, no puede ocurrir en una expresión. Un procedimiento se llama escribiendo su nombre. Cuando se invoca el procedimiento, los pasos que lo definen se ejecutan y a continuación se devuelve el control al programa que le llamó.

Sintaxis

SubProceso Nombre (*parámetros*)

<acciones>

FinSubProceso

Los parámetros tienen el mismo significado que en las funciones.

¿CÓMO INVOCAMOS A UN SUBPROGRAMA?

Un procedimiento puede ser llamado de la siguiente forma:

nombre(argumentos)

- nombre_procedimiento: procedimiento que se va a llamar.
- argumentos: constantes, variables, expresiones.



Pueden encontrar ejemplos para descargar de Procedimientos en Aula Virtual.

RECURSIÓN

Una función o procedimiento que se puede llamar a sí mismo se llama recursivo. La recursión (recursividad) es una herramienta muy potente en algunas aplicaciones, sobre todo de cálculo. La recursión puede ser utilizada como una alternativa a la repetición o estructura repetitiva. El uso de la recursión es particularmente idóneo para la solución de aquellos problemas que pueden definirse de modo natural en términos recursivos.

La escritura de un procedimiento o función recursiva es similar a sus homónimos no recursivos; sin embargo, para evitar que la recursión continúe indefinidamente es preciso incluir una condición de terminación, que suele hacerse con una estructura condicional.



Revisemos lo aprendido hasta aquí

- Declarar subprocesos
- Invocar a subprocesos desde el algoritmo principal
- Enviar argumentos al invocar.
- Qué es la recursión y para qué sirve