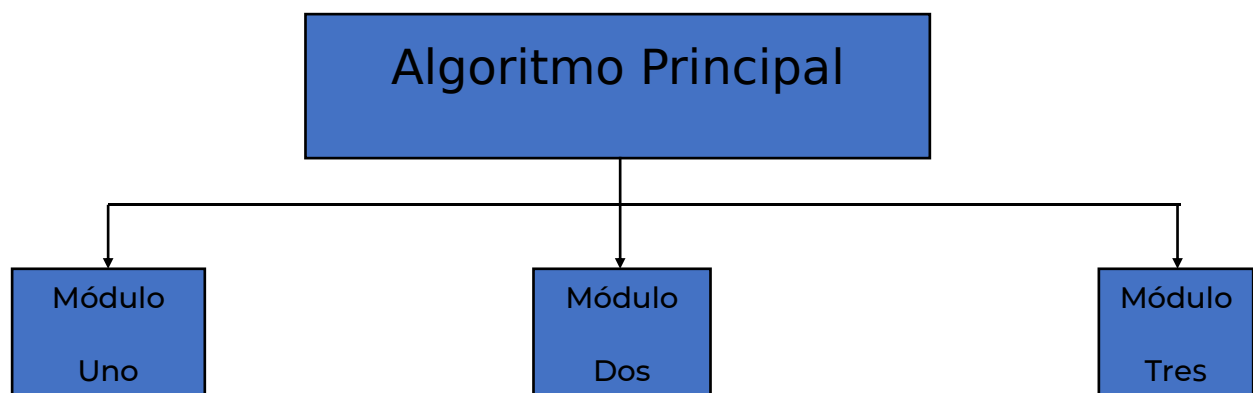


Subprogramas

Es una técnica que permite diseñar la solución de un problema con base en la modularización o segmentación, dado un enfoque de arriba hacia abajo. Esta solución se divide en módulos que se estructuran jerárquicamente, como si fuera el organigrama de una empresa.

Ejemplo:



Un módulo es una rutina, subrutina, subalgoritmo, procedimiento o función que puede definirse dentro de un algoritmo con el fin de ejecutar una tarea específica y puede ser llamado o invocado desde el algoritmo principal cuando sea necesario.

¿Cuándo utilizar modularización?

Este enfoque es útil en dos casos:

1. Cuando existe un grupo de instrucciones o una tarea específica que deba ejecutarse en más de una ocasión.
2. Cuando el problema es complejo o extenso, la solución se divide en módulos, "partes o subtarear más simples o específicas". Dicha solución se organiza en forma jerárquica. Cada una de las partes ejecutará una tarea específica. El módulo principal dirige o coordina el funcionamiento de los módulos subordinados.

Procedimientos:

Un procedimiento es un subprograma que realiza una tarea específica y que puede ser definido mediante 0, 1 o más parámetros. Tanto en la entrada de información al procedimiento como la devolución de resultados desde el subprograma se realiza mediante parámetros, el cual nos sirve para introducir o modificar información del programa principal.

Sintaxis

Procedimiento nombre_del_procedimiento [(parámetros)]

[variables locales]

inicio

instrucciones

fin

Ejemplo :

Elaborar un procedimiento que presente 5 asteriscos en una línea horizontal.

cadena[25] nombre

procedimiento asteriscos

entero i

inicio

para i <- 1 hasta 5 haga

escriba "*"

fin para

fin

inicio

escriba "Ingresar el nombre ..: "

lea nombre

llamar asteriscos

llamar nueva_linea

escriba nombre

llamar nueva_linea

llamar asteriscos

fin

Funciones

Las funciones son subprogramas que hacen una o más instrucciones pero que siempre nos devuelven un solo valor .

Sintaxis

funcion nombre_funcion [(parámetros)]: tipo_de_retorno

[variables locales]

inicio

instrucciones

Retorne valor

fin

Ejemplo :

calculo de la potencia Usaremos el mismo ejercicio que usamos para los procedimientos, para hacer una demostración de cómo cambiaría el programa usando una función para el cálculo de la potencia.

entero base,expo,pot

funcion Potencia(entero base , entero expo): entero

entero i, resp

inicio

resp<-1

para l <- 1 hasta expo haga

resp <- resp * base

fin para

retorne resp

fin

procedimiento ingreso(var entero base,var entero expo)

inicio

escriba "Ingresar la base "

lea base

escriba " Ingresar el exponente "

lea expo

fin

inicio

llamar ingreso(base,expo)

pot<-potencia(base,expo)

escriba "Potencia es ", pot

fin