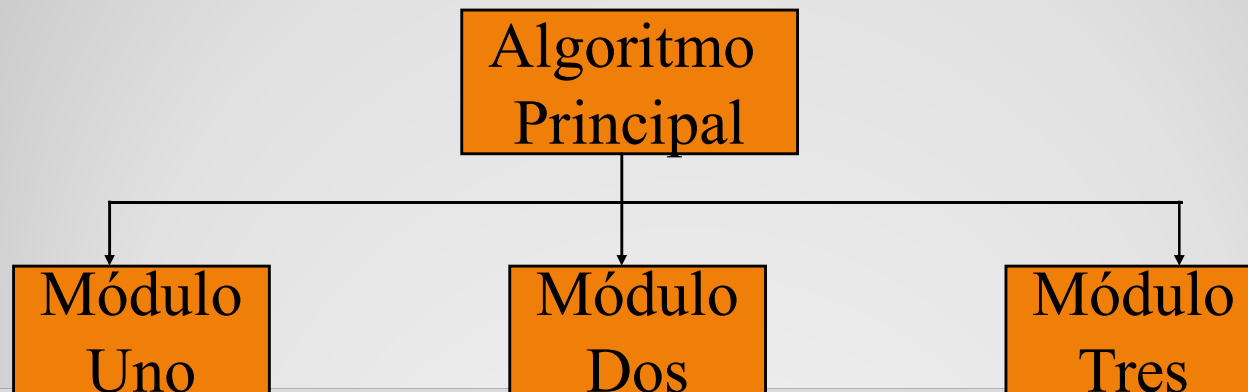


Programacion Estructurada - Funciones y Procedimientos(Pseu docódigo)

CFGs.ASIR - Pseudocódigo

Programación Modular

- 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 (Top-Down). Esta solución se divide en módulos que se estructuran jerárquicamente, como si fuera un organigrama. Ejemplo:



Definición de Módulo

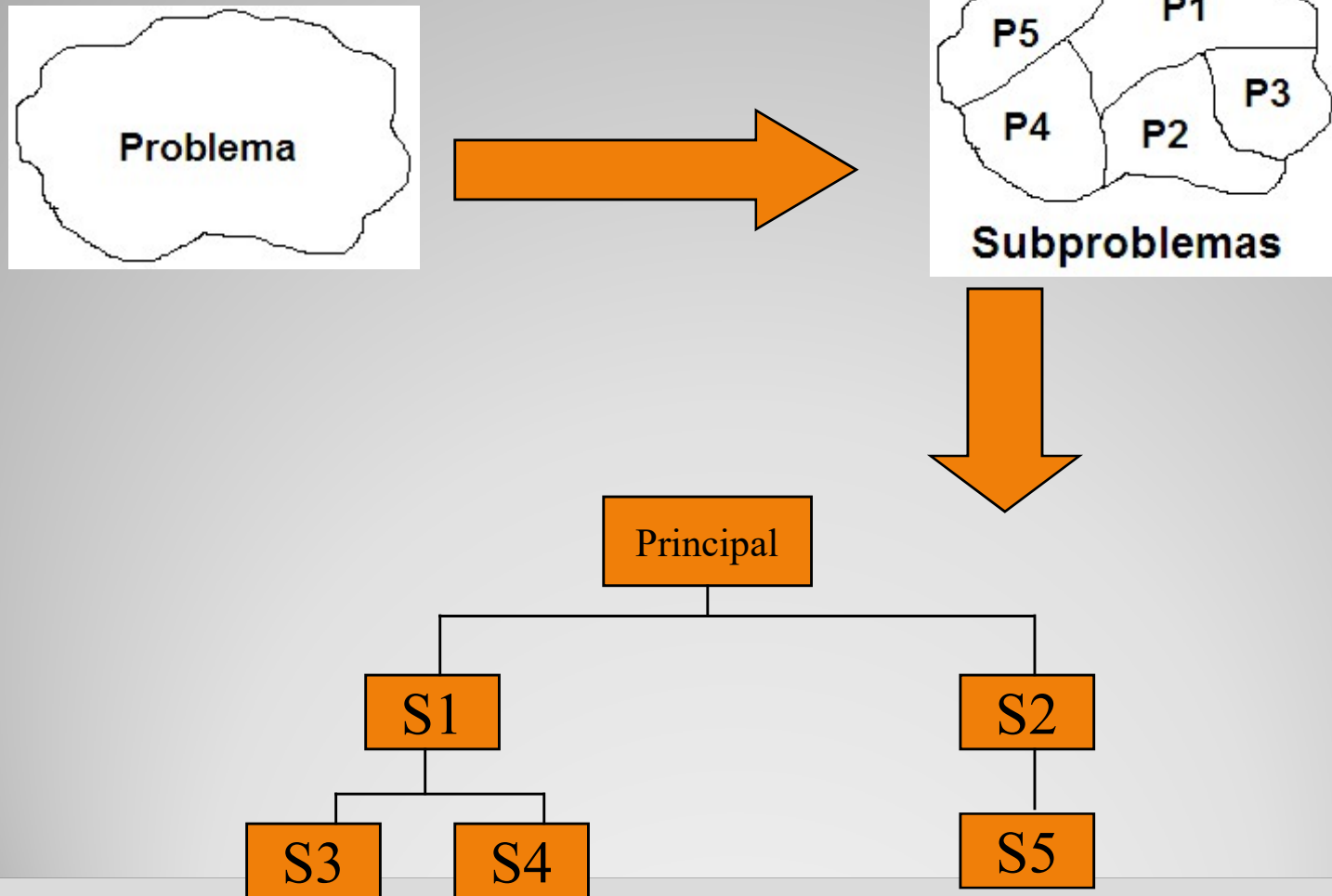
- Un **módulo** es una rutina, subrutina, subalgoritmo, **procedimiento** o **función** que puede definirse dentro de un algoritmo con el fin de ejecutar una tarea específica y puede ser llamado o invocado desde el algoritmo principal cuando sea necesario.

¿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 subtareas 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.

Esquema de Programación Modular



Suprocesos en Pseint

En Pseint las funciones/procedimientos se denominan subprocesos

SubProceso variable_de_retorno <- **Nombre** (
Argumentos)

FinSubProceso

Proceso P1

FinProceso

- Para invocar a la función se debe utilizar su **nombre** y dentro de () los parámetros o argumentos.

Las funciones deben ser declaradas antes de su llamada desde el módulo principal

// funcion que no recibe ni devuelve nada

SubProceso Saludar

 Escribir "Hola mundo!";

FinSubProceso

// funcion que recibe un argumento por valor, y devuelve su doble

SubProceso res <- CalcularDoble(num)

 Definir res como real;

 res <- num*2; // retorna el doble

FinSubProceso

Ejemplos de Funciones

Llamada a las Funciones desde el módulo principal

Proceso PruebaFunciones

Definir x Como Real;

Escribir "Llamada a la funcion Saludar:";

Saludar; // como no recibe argumentos pueden omitirse los paréntesis vacíos

Escribir "Ingrese un valor numérico para x:";

Leer x;

Escribir "El doble de ",x," es ", CalcularDoble(x);

Escribir "El valor original de x es ",x;

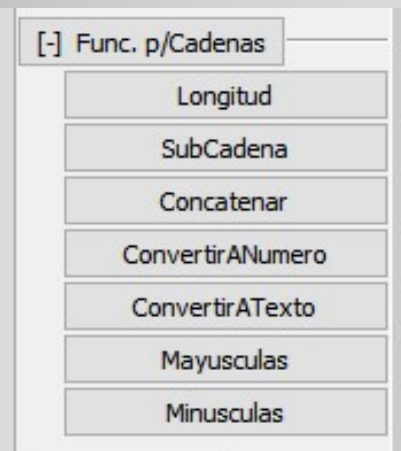
FinProceso

Invocar a una funcion/Subproceso

- Una llamada puede ser en sí una instrucción (A), pero si la función retorna algún valor, también puede utilizarse como operando dentro de una expresión(B).
- Ejemplo A) resultado<-CalcularDoble(4);
- Ejemplo A) Saludar;
- Ejemplo B) resultado<- CalcularDoble(4)+20;

Funciones de cadena predefinidas

- Existen funciones predefinidas para tratar cadenas de texto que puedes encontrar en la ayuda de Pseint. Son:



```
1  Algoritmo sin titulo
2
3      Longitud(cadena)
4      SubCadena(cadena, desde, hasta)
5      Concatenar(cadena, cadena)
6      ConvertirANumero(cadena)
7      ConvertirATexto(numero)
8      Mayusculas(cadena)
9      Minusculas(cadena)
10 FinAlgoritmo
11
```

EJERCICIOS

- Escribe un programa usando funciones de cadena para convertir un texto alternando una letra mayúscula con otra minúscula.
- Escribe un programa que cuente el número de vocales que se encuentran en el texto introducido por teclado.