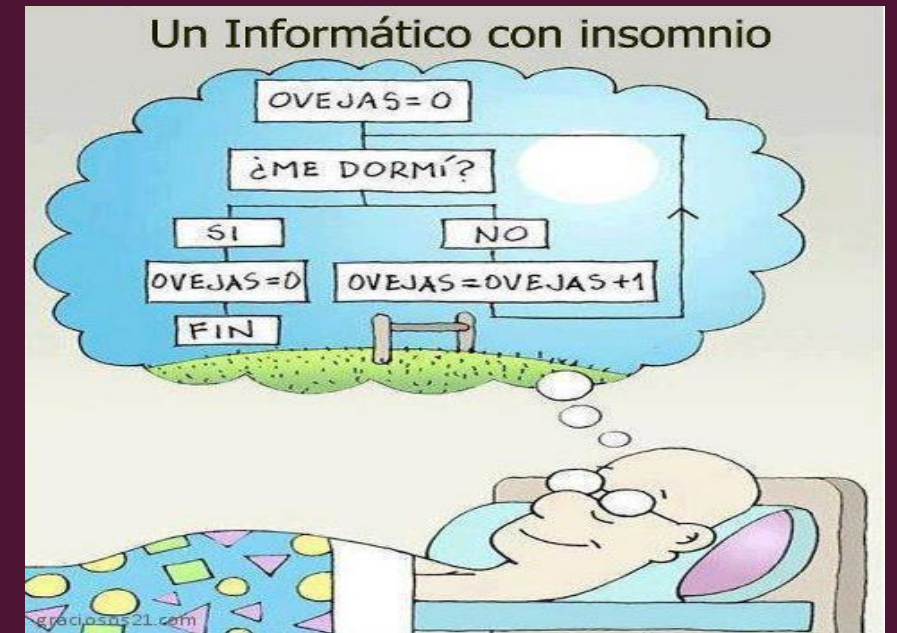


PSEUDOCÓDIGO

PROCEDIMIENTOS Y FUNCIONES



¿QUÉ SON?

- **Tipos de instrucciones:**

- Primitivas
- De control
- Llamadas a subprogramas

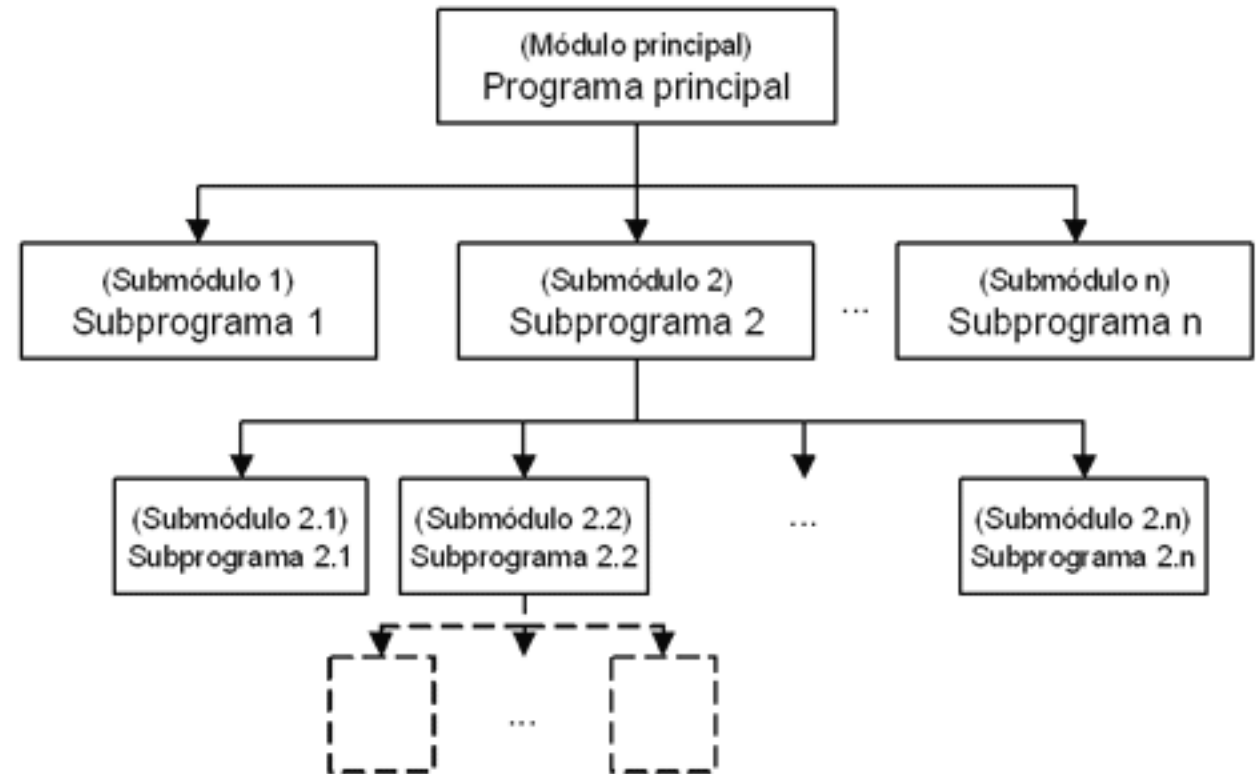
- **Subprograma**

- Procedimientos
- Funciones

- **Divide y vencerás**

- **Criterios**

- Tamaño
- Simplicidad
- Legibilidad
- Reutilización de código



PROCEDIMIENTOS

`/* Cabecera */`

```
procedimiento <nombre_del_procedimiento>(
[ <lista_de_parámetros_formales> ] )
```

`/* Declaraciones */`

```
[ constantes
  <declaraciones_de_constantes> ]
[ tipos_de_datos
  <declaraciones_de_tipos_de_datos> ]
[ variables
  <declaraciones_de_variables> ]
```

`/* Cuerpo */`

```
inicio
  <bloque_de_instrucciones>
fin
```

- **Sintaxis** muy similar a la de un programa
- **Diferencias:**
 - Procedimiento ↔ Programa
 - Paréntesis después del nombre. Dentro de los paréntesis, los parámetros formales
 - Declaraciones de ámbito local ↔ Declaraciones de ámbito global
 - Los subprogramas también se declaran:
 - de ámbito global (en el programa principal)
 - de ámbito local (en los subprogramas)
- **Reglas:**
 - Realizar las declaraciones lo más localmente posible
 - Utilizar nombres de procesos distintos
 - Tener en cuenta cuando se libera el espacio ocupado por una variable

PROCEDIMIENTOS

PROCEDIMIENTOS SIN PARÁMETROS:

- **Declaración:** En sección de declaraciones:

[subprogramas

[<nombre_del_procedimiento>()*]

- **Uso:** Instrucción dentro del programa o procedimiento

<nombre_del_procedimiento>()

PROCEDIMIENTOS CON PARÁMETROS (argumentos):

- **Ventajas de disponer de parámetros:**

- Proporcionar uno o más datos de entrada al llamado.
- Devolver uno o más datos de salida al llamante.

- **Tipos de parámetros**

- Entrada
- Salida
- Entrada y salida

- **Declaración**

[subprogramas

[<nombre_del_procedimiento>(lista_de_parámetros_formales)]*]

- **Uso**

<nombre_del_procedimiento>([lista_de_parámetros_actuales])



Modifica el programa de los menús y
benefíciate de usar procedimientos

MODIFICACIÓN I AL PROGRAMA DE LOS MENÚS

En el programa de los menús:

- Creamos un procedimiento sin parámetros que imprima el menú de opciones 1. Una vez creado sustituiremos el trozo de código que imprime este menú por la llamada al procedimiento creado.
- Crearemos un procedimiento sin parámetros que imprima el menú de opciones 2. Una vez creado sustituiremos el trozo de código que imprime este menú por la llamada al procedimiento creado.

Las funciones podrían ser, por ejemplo:

- ImprimirMenu1()
- ImprimirMenu2()

PARÁMETROS

PARÁMETROS FORMALES (ficticios):

- Variables locales declaradas en cabecera de procedimiento
- Almacenan:
 - Datos de entrada (proporcionados en la llamada)
 - Datos de salida (devueltas)
 - Datos de entrada y salida
- No son lo mismo que las variables locales (*declaradas en la sección de declaración, no se usan para transferir datos desde/hacia el subproceso*)

LISTA DE PARÁMETROS FORMALES:

[[E <nombre_del_tipo_de_dato> : <lista_de_variables> ;]*

[S <nombre_del_tipo_de_dato> : <lista_de_variables> ;]*

[E/S <nombre_del_tipo_de_dato> : <lista_de_variables> ;]*

PARÁMETROS ACTUALES:

- Son los que aparecen entre paréntesis en la llamada al procedimiento.
- En la llamada al procedimiento: se copian los datos de entrada en las variables correspondientes. ➡ **Paso por valor**
- A los parámetros formales de salida se le asigna un valor dentro del procedimiento, pero lo que realmente ocurre es que se asigna el valor al parámetro actual correspondiente ➡ **Paso por referencia**



Modifica el programa de los menús y benefíciate de usar procedimientos con parámetros

MODIFICACIÓN 2 AL PROGRAMA DE LOS MENÚS

En el programa de los menús resultado de la modificación anterior:

- Creamos un procedimiento con parámetros que recibiendo como entrada la opción recogida del primer menú (el de pide la operación: suma, producto, opuesto o salir), y devolviendo si se debe salir, la llamada a dicho procedimiento sustituya al trozo de código que realiza dicha operativa.

El procedimiento podría ser:

- Procedimiento RealizarOperacion(E Entero: opc; S Booleano: finalizar)

FUNCIONES

/* Cabecera */

```
<tipo de dato> funcion <nombre_del_procedimiento>([  
<lista_de_parámetros_formales> ] )
```

/* Declaraciones */

```
[ constantes  
  <declaraciones_de_constantes> ]  
[ tipos_de_datos  
  <declaraciones_de_tipos_de_datos> ]  
[ variables  
  <declaraciones_de_variables> ]
```

/* Cuerpo */

```
inicio  
  <bloque_de_instrucciones>  
  retornar expresion  
fin
```



Modifica el programa de los menús y utiliza una función para hacer las operaciones

- **Sintaxis** muy similar a la de un procedimiento

- **Diferencias:**

- Procedimiento ↔ Función.
- Posee un tipo de dato correspondiente a la función que aparece en la cabecera.

- **Declaración**

```
[ subprogramas  
  [tipo _de _dato <nombre_del_procedimiento>(  
    lista_de_parámetros_formales)]]*
```

- **Uso**

```
vble = <nombre_del_procedimiento>(  
  [lista_de_parámetros_actuales])
```


MODIFICACIÓN 3 AL PROGRAMA DE LOS MENÚS

En el programa de los menús resultado de las dos modificaciones anteriores:

- Creamos una función que recibiendo como entrada los número a operar y la operación a realizar, devuelva el resultado. El código del programa que realiza las distintas operaciones debe sustituirse por la llamada a esta función.

La función podría ser:

- Funcion Entero Opera(E Entero: ope1, ope2, operacion)

PROCEDIMIENTOS Y FUNCIONES

