



# Sesión 5 – Semana 3

---

## Subalgoritmos - Funciones

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701  
Medellín (Antioquia), Colombia



# Contenido

- Mapa conceptual
- Introducción
- Definición subalgoritmo
- Funciones

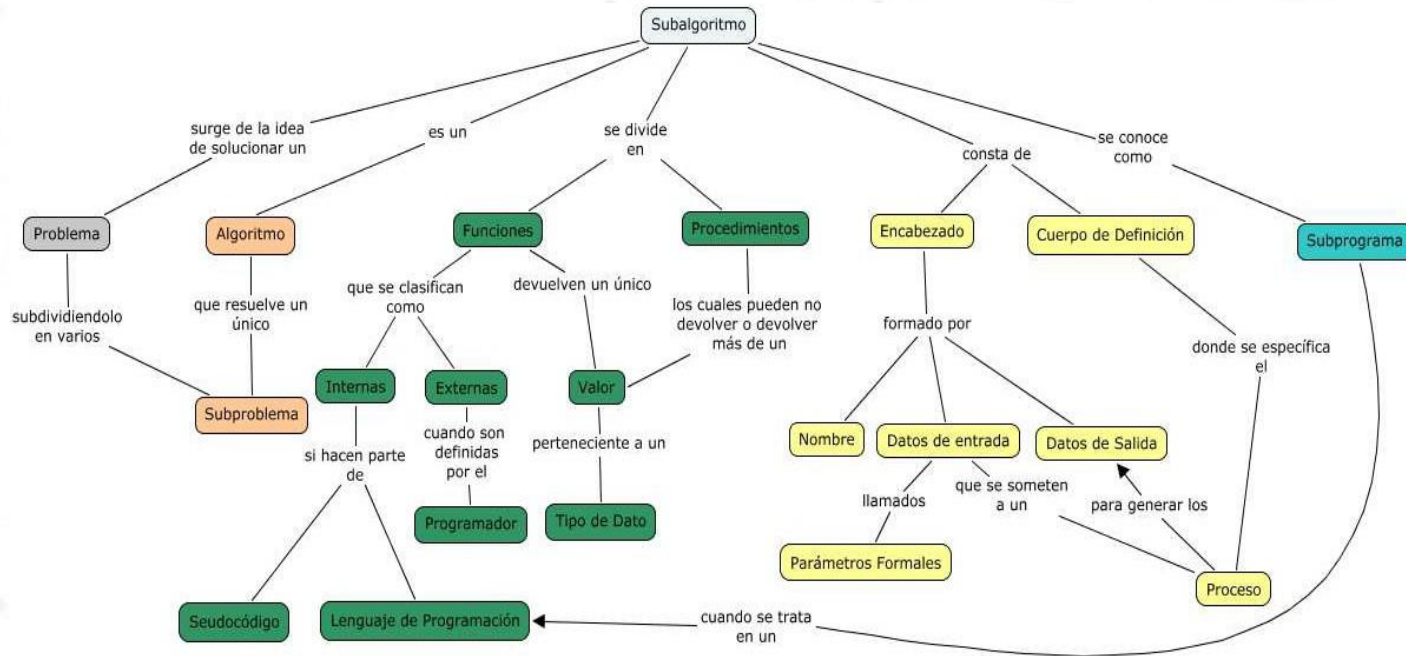
WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701  
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG

# Mapa Conceptual de Subalgoritmo



WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701  
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



# Introducción

- La solución de problemas complejos se facilita si se dividen en problemas mas sencillos (subproblemas).
- La solución de estos subproblemas se realiza con subalgoritmos.
- El algoritmo principal soluciona el problema principal mediante la invocación apropiada de subalgoritmos.
- La técnica de dividir un problema principal en subproblemas se suele denominar "divide y vencerás".
- Solucionar un problema principal a partir de soluciones de subproblemas se conoce como diseño descendente, ya que se inicia con un problema general y finalizamos en el diseño específico de las soluciones

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701  
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G



# Ejemplo

Consideremos el problema de calcular el área de un triángulo. Este problema se puede dividir en subproblemas:

Subproblema 1: Entrada de datos (altura y base).

Subproblema 2: Calcular el área.

Subproblema 3: Salida de resultados.

La parte del algoritmo principal que resuelve los tres subproblemas es:

```
⋮  
lea(altura, base)  
area ← base * altura / 2  
muestre(area)  
⋮
```

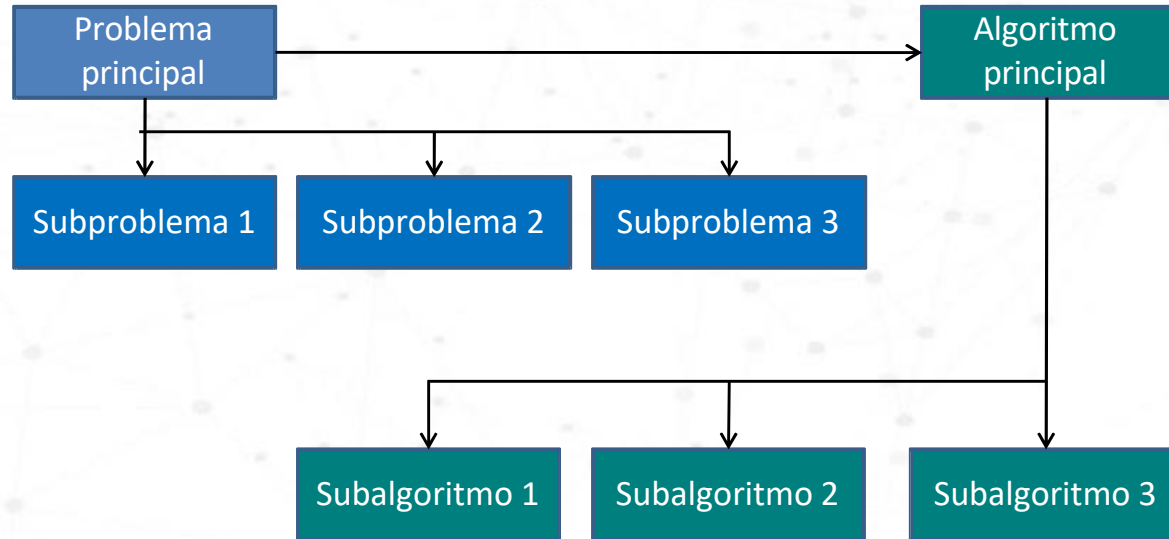
WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701  
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG

# Diseño Descendente



WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701  
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



# Subalgoritmo

- Secuencia de declaraciones e instrucciones que llevan a cabo **una** tarea específica.
- Son de dos tipos: funciones y procedimientos.
- Se escriben una vez, pero pueden ser invocados (utilizados) en diferentes partes del algoritmo principal u otro subalgoritmo.
- Cuando se trata en un lenguaje de programación, se denomina subprograma.
- Constan de un encabezado y un cuerpo de definición.

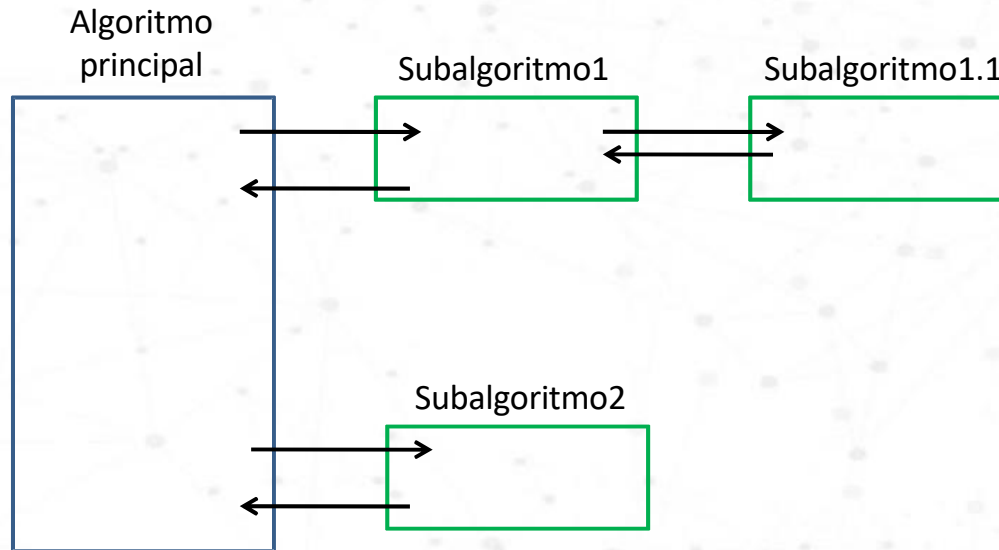
WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701  
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG

# Algoritmo con diferentes niveles de subalgoritmos



WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701  
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG





# Funciones

- Matemáticamente, una función es una operación que toma uno o más valores llamados argumentos y produce un valor denominado resultado –valor de la función para los argumentos dados.
- Todos los lenguajes de programación tienen funciones incorporadas (intrínsecas o internas) y permiten crear funciones definidas por el usuario (funciones externas). Las funciones devuelven un valor de algún tipo de dato, sea simple o compuesto.

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701  
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G



# Estructura de las funciones (externas) en pseudocódigo

<TipoDeResultado> **funcion**

<NombreFuncion>(<ListaParametrosFormales>) [Declaración de  
variables y/o constantes locales]

**inicio**

<Instruccio

nes>

**devolver**

<Expresión>

**fin\_funcion**

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701  
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



# Elementos del encabezado de las funciones

- **<TipoDeResultado>** : Indica el tipo del valor devuelto por la función. Puede ser cualquiera de los tipos de datos simples o compuestos.
- **funcion**: Indica que se esta definiendo una función.
- **<NombreFuncion>**: Nombre asociado con la función, debe ser un nombre de identificador válido seguido de paréntesis.
- **<ListaParametrosFormales>**: Conjunto de declaración de variables sobre las cuales operará la función para producir el resultado. Estas variables son locales a la función donde se declaran.



# Elementos del cuerpo de las funciones

- **[Declaración de variables locales]**: Se especifica el nombre de las variables y su tipo en caso que se requieran para la solución del problema. Las variables declaradas dentro de la función pertenecen exclusivamente a ésta.
- **<Instrucciones>**: Instrucciones que constituyen la definición de la función.
- **devolver**: Se utiliza para regresar el resultado de la función. Cuando se ejecuta esta instrucción termina inmediatamente la función y el control del programa se transfiere al llamador de la función.
- **<Expresion>**: Sólo existe si la función se ha declarado con valor de retorno e indica el valor devuelto por la función.



# Ejemplo 1: Definición de una función mediante pseudocódigo

*Diseñe una función que calcule y devuelva el área de un triángulo.*

real funcion calcularAreaTriangulo(real: base, altura)

variables

real: area

inicio

area  $\leftarrow$  base \* altura / 2

devolver area

fin\_funcion

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701  
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



# Invocación de funciones

- Una función puede ser llamada de la forma siguiente: NombreFunción (Lista de argumentos)
- NombreFunción: Función que se invoca o llama.
- Lista de argumentos: Constantes, variables, expresiones o valores retornados por funciones.

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701  
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



## Ejemplo 2: Invocar o llamar funciones

algoritmo LlamarFuncion

variables

real: b, h, a

inicio

muestre('Ingresa la base del triangulo: ')

lea(b)

muestre('Ingresa la altura del triangulo: ')

lea(h)

$a \leftarrow \text{calcularAreaTriangulo}(b, h)$

muestre('El área es: ', a)

fin

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701  
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG

# Ejemplo 3: Suma de enteros consecutivos

Diseñe una función que reciba un entero n y retorne la suma de los números enteros entre uno y n.

entero funcion sumarEnterosConsecutivos(entero: n)

variables

entero: i, sumatoria

inicio

sumatoria  $\leftarrow$  0

para i  $\leftarrow$  1 hasta n

sumatoria  $\leftarrow$  sumatoria + i

fin\_para

devolver sumatoria

fin\_funcion

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701  
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



# Ejemplo 4: Función que determina si un número es primo

logico funcion esPrimo(entero: n)

variables

entero: divisores, i

inicio

divisores  $\leftarrow$  0

para i  $\leftarrow$  1 hasta n

si n mod i = 0 entonces

divisores  $\leftarrow$  divisores + 1

fin\_si

fin\_para

si divisores = 2 entonces

devolver verdadero

si\_no

devolver falso

fin\_si

fin\_funcion

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701  
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG

# Ejemplo 5: Función que determina si un número es primo (Mejorado)

```
logico funcion esPrimo2(entero: n)
variables
    entero: i
inicio
    si n < 2 entonces
        devolver falso
    si_no
        para i ← 2 hasta n - 1
            si (n mod i) = 0 entonces
                devolver falso
            fin_si
        fin_para
    fin_si
    devolver verdadero
fin_funcion
```

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701  
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG

## Ejemplo 6: Función que retorna la cantidad de primos que hay entre uno y n

entero funcion contarPrimos(entero: n)

variables

entero: contador, i

inicio

contador  $\leftarrow$  0

para i  $\leftarrow$  1 hasta n

si esPrimo2(i) = verdadero entonces

contador  $\leftarrow$  contador + 1

fin\_si

fin\_para

devolver contador

fin\_funcion

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701  
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG

# Ejemplo 7: función que determina si un año es bisiesto

logico funcion bisiesto(entero: anio)

inicio

si anio mod 4 = 0 && (anio mod 100  $\neq$  0 o anio mod 400 = 0) entonces

devolver verdadero

si\_no

devolver falso

fin\_si

fin\_funcion

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701  
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG

# Algunas funciones internas de PSeint

Función interna (Built-in)	
[-] Func. Matemáticas	[-] Func. p/Cadenas
abs (valor absoluto)	Longitud
trunc (valor truncado)	SubCadena
redon (valor redondeado)	Concatenar
raiz (raíz cuadrada)	ConvertirANumero
sen (seno)	ConvertirATexto
cos (coseno)	Mayusculas
tan (tangente)	Minusculas
asen (arcoseno)	
acos (arcocoseno)	
atan (arcotangente)	
ln (logaritmo natural)	
exp (func. exponencial)	
azar (numero aleatorio)	

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701  
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



# Ejercicios

- Solucionar nuevamente los ejemplos propuestos en esta presentación.
- Solucionar el taller

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701  
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



# Referencias

- Joyanes Aguilar, Luis. Fundamentos de Programación: Algoritmos, estructuras de datos y objetos, 4ª edición, Madrid: McGraw-Hill, 2008.

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701  
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



■ [WWW.MAKAIA.ORG](http://WWW.MAKAIA.ORG)

Info: [comunicaciones@makaia.org](mailto:comunicaciones@makaia.org)

Corporación MAKAlA

Medellín, Colombia

Carrera 43A – 34-155.

Almacentro

Torre Norte, Oficina 701

Teléfono: (+574) 448 03 74

Móvil: (+57) 320 761 01 76



@makaiaorg