

Algoritmos y Resolución de Problemas

Eje N° 2



fcefn

Facultad de Ciencias Exactas, Físicas y Naturales
Universidad Nacional de San Juan

Di DEPARTAMENTO
DE INFORMÁTICA

2020

-
- Estrategia de resolución de problemas con una computadora

- ☐ Construcción de Algoritmos simples

- ☐ Construcción de algoritmos con Subprogramas

Un **Algoritmo** es un **conjunto ordenado y finito** de pasos que permite **solucionar un problema**

- **Precisión**
- **Finitud**
- **Efectividad**
- **Robustez**
- **Correctitud**
- **Compleitud**
- **Eficacia**
- **Eficiencia.**

PROBLEMA :

Calcular la superficie y el perímetro de un rectángulo , cuyas dimensiones se conocen

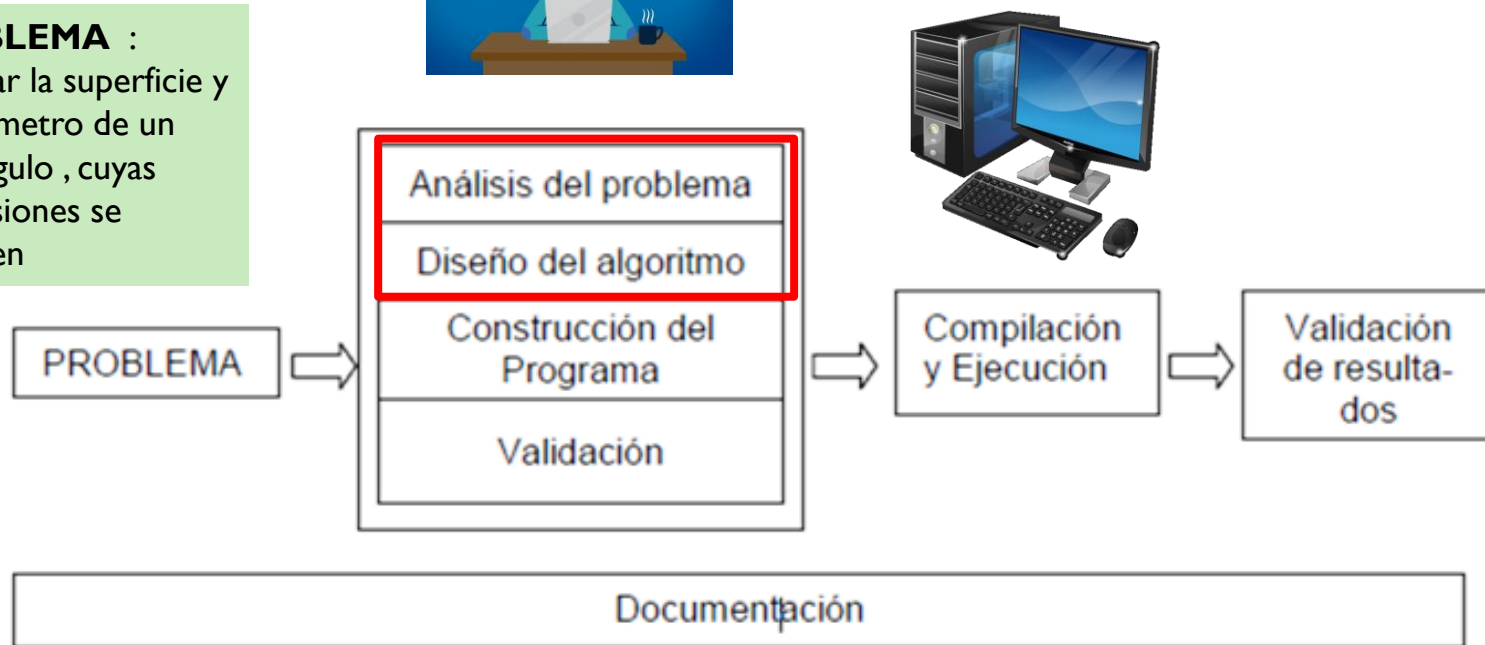


Figura 1.6 Temporalidad de las etapas en la resolución de problemas con la computadora

Etapas resolución de problemas utilizando computadoras	Objetivo
1° Análisis del problema	Ayuda a la comprensión del problema, a comprender ¿qué se debe hacer? . Se lo denomina también especificación del problema.
2° Diseño del Algoritmo	Permiten explorar soluciones y seleccionar la que se considere más adecuada Se determina ¿cómo se debe hacer? Permite indicar cómo el algoritmo realizará la tarea
3° Codificación. Construcción del programa	Se traduce el algoritmo a un lenguaje de programación para que pueda ser interpretado y ejecutado por una computadora.
4° Compilación y ejecución	
5° Validación	Permite comprobar la solución obtenida
6° Documentación	Se registra lo que se realiza durante todo el proceso, y en cada una de las distintas etapas.



ESTRATEGIA DE RESOLUCIÓN DE PROBLEMAS (4-pasos)





- ESTRATEGIA DE RESOLUCIÓN DE PROBLEMAS -



- CONSTRUCCIÓN DE ALGORITMOS

1° PASO

Análisis del problema

Resolver un problema requiere transformar mediante un proceso, los datos de entrada en Datos de salida

Ayuda a la comprensión del problema

Responder :

1° Análisis del problema.

¿qué hacer?

¿Qué se debe obtener? DATOS DE SALIDA

¿Con que datos se cuenta? DATOS DE ENTRADA

¿Existen Restricciones? Condiciones de los datos

Especificaciones de un problema

Especificación de Salidas

Especificación de Entradas

ALGORITMO

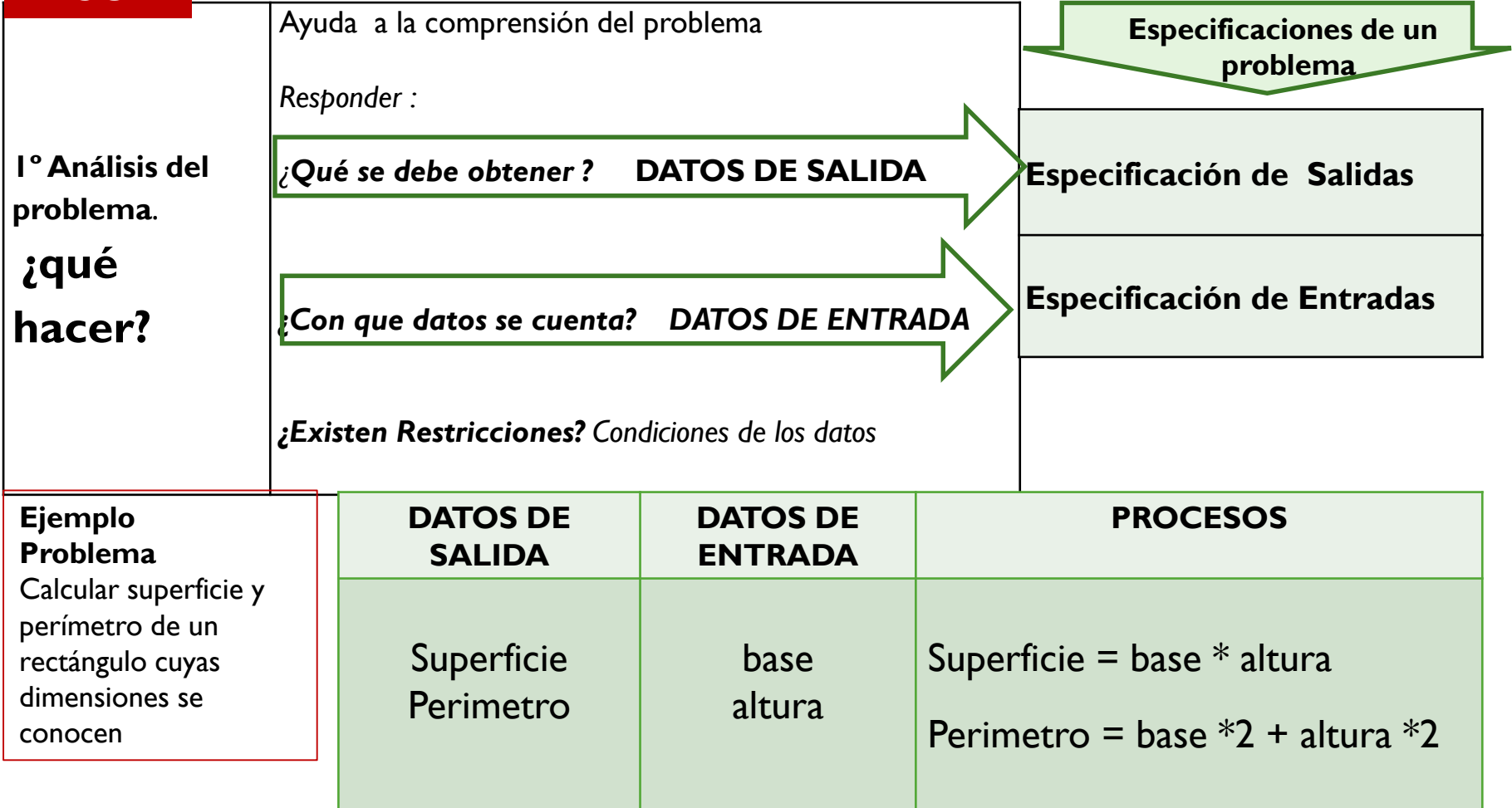
Entrada

PROCESO

Salida

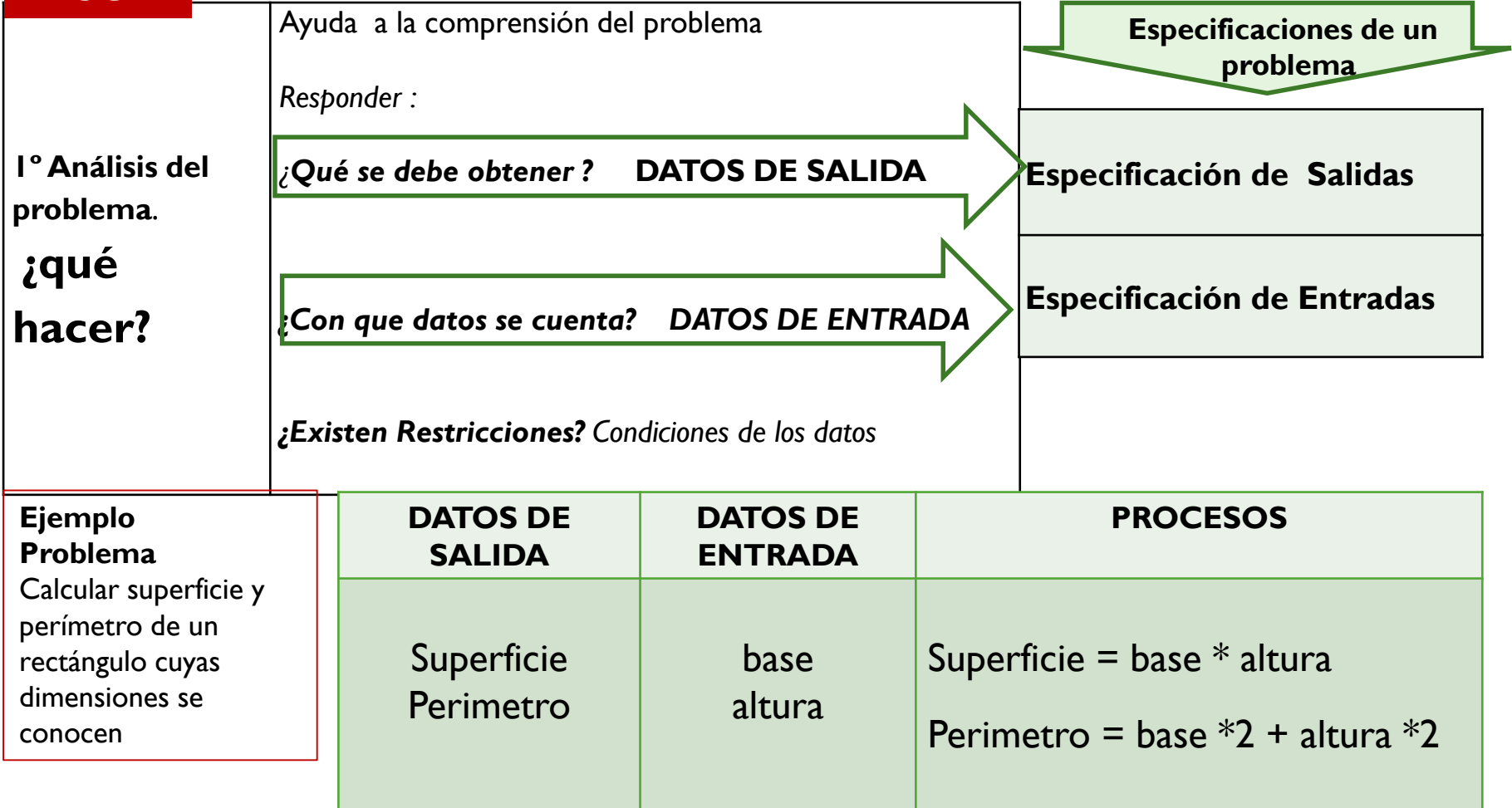
1° PASO

Análisis del problema



1° PASO

Análisis del problema



**2°
PASO**

Identificar las acciones del proceso

DATOS DE SALIDA	DATOS DE ENTRADA	PROCESOS
Superficie Perimetro	base altura	Superficie = base * altura Perimetro = base *2 + altura *2

ALGORITMO

Entrada

PROCESO

Salida

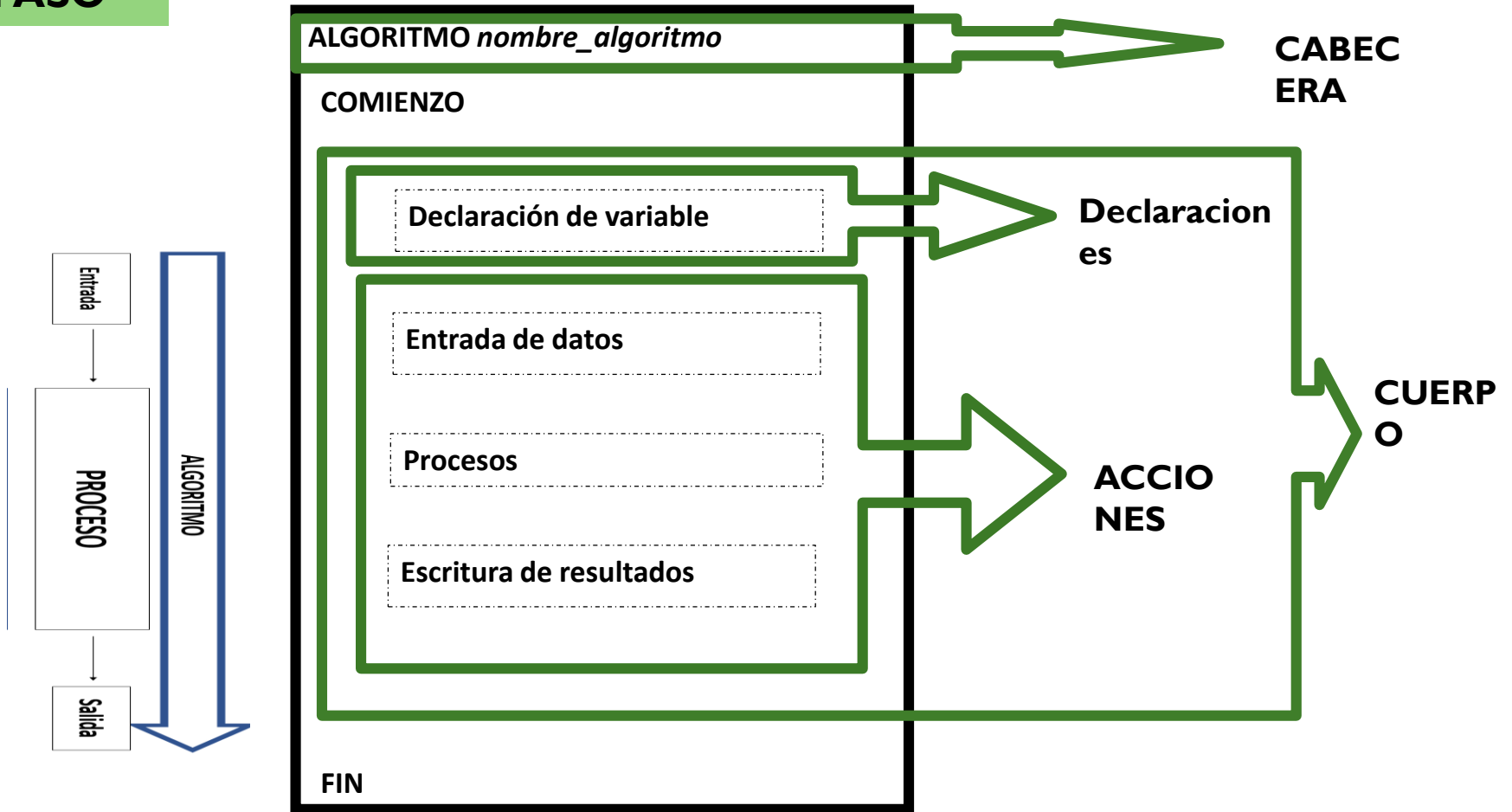
Leer base
Leer altura

Superficie = base * altura
Perimetro = base *2 + altura *2

Escribir Superficie
Escribir Perimetro

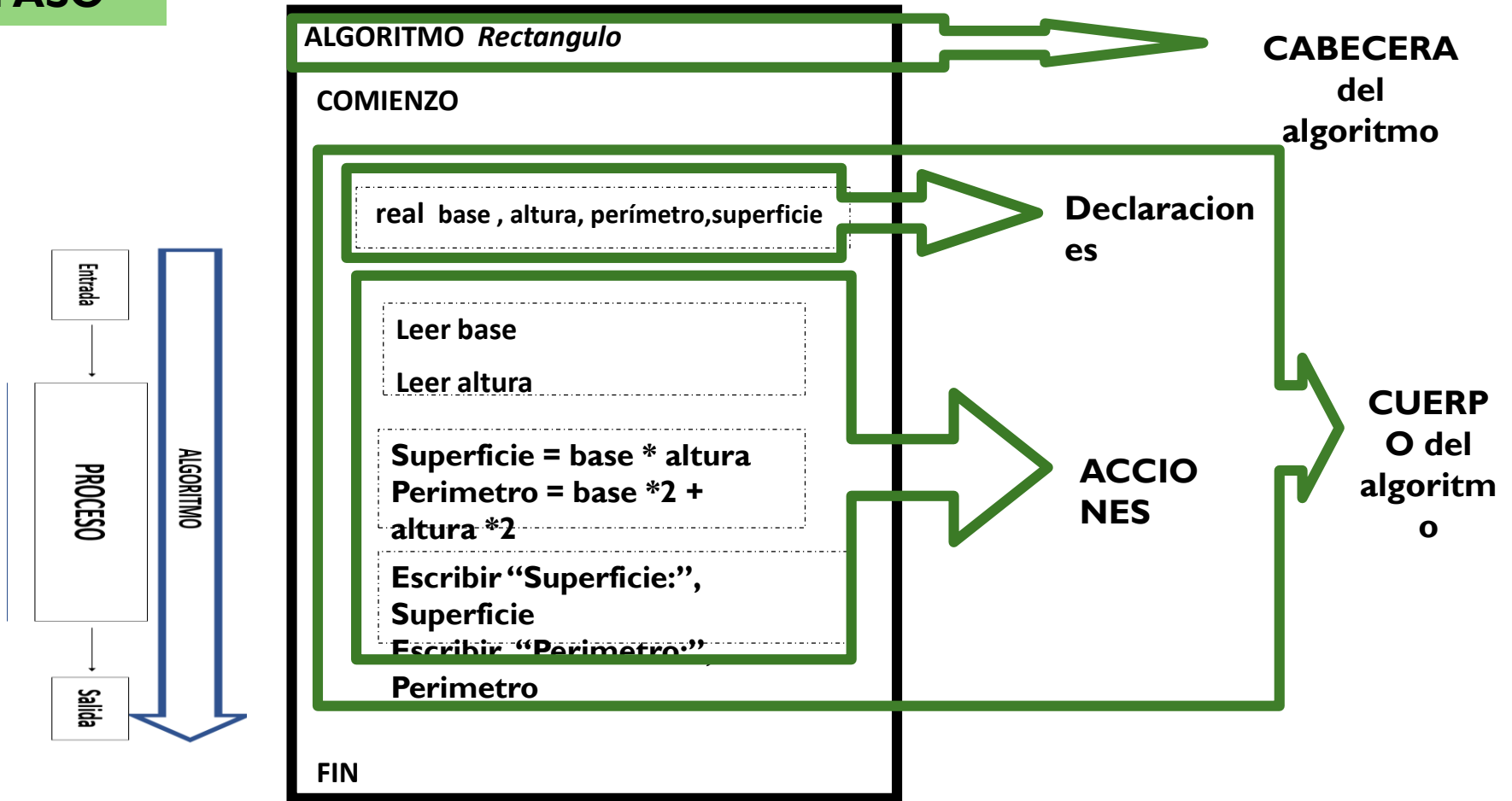
3° PASO

Elaboración del algoritmo




3° PASO


Elaboración del algoritmo - Ejemplo



SEGUIMIENTO DEL ALGORITMO

MEMORIA				PANTALLA	
 VARIABLES	base	altura	Superficie	Perimetro	SALIDA

SEGUIMIENTO DEL ALGORITMO

MEMORIA				PANTALLA	
 VARIABLES	base	altura	Superficie	Perimetro	SALIDA
	10	5	50	30	Superficie: 50 Perímetro: 30

ESTRATEGIA DE RESOLUCIÓN DE PROBLEMAS

-

**CONSTRUCCIÓN DE
ALGORITMOS CON
SUBPROGRAMAS**

1º PASO

Análisis del problema

Ayuda a la comprensión del problema

Responder :

¿Qué se debe obtener? DATOS DE SALIDA

¿Con que datos se cuenta? DATOS DE ENTRADA

¿Existen Restricciones? Condiciones de los datos

Especificaciones de un problema

Especificación de Salidas

Especificación de Entradas

1º Análisis del problema.

¿qué hacer?

DATOS DE SALIDA

DATOS DE ENTRADA

PROCESOS

**1°
PASO**

Análisis del problema

Identificar los Subproblemas (Subprocesos – Subprogramas)

ALGORITMO

Entrada

PROCESO

Salida

Aplica Técnica divide y vencerás (DYV)

Aplica la
técnica
divide y
vencerás
DYV

Subproceso
SUPERFICIE

SubprogramaI

Subproceso N
PERIMETRO

SubprogramaN

El problema se divide en subproblema

El algoritmo se divide en subprograma

I° PASO

Análisis del problema Identificar los Subproblemas (Subprocesos – Subprogramas)

Ayuda a la comprensión del problema

Responder :

I° Análisis del problema.

¿qué hacer?

¿Qué se debe obtener? **DATOS DE SALIDA**

¿Con que datos se cuenta? **DATOS DE ENTRADA**

¿Existen Restricciones? Condiciones de los datos

Especificaciones de un problema

Especificación de Salidas

Especificación de Entradas

Ejemplo Problema

Calcular superficie y perímetro de un rectángulo cuyas dimensiones se conocen

DATOS DE SALIDA

Superficie
Perímetro

DATOS DE ENTRADA

base
altura

PROCESOS

Leer datos
Calcular superficie
Calcular perímetro
Mostrar resultados

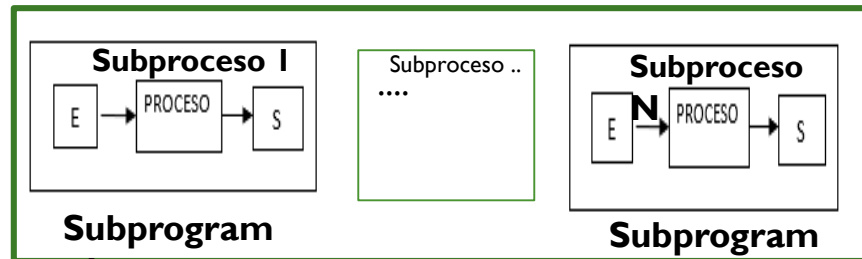
**2°
PASO**

**Identificar los Subproblemas (Subprocesos – Subprogramas) –
Descripción de cada uno**

ALGORITMO



Aplica Técnica divide y vencerás (DYV)

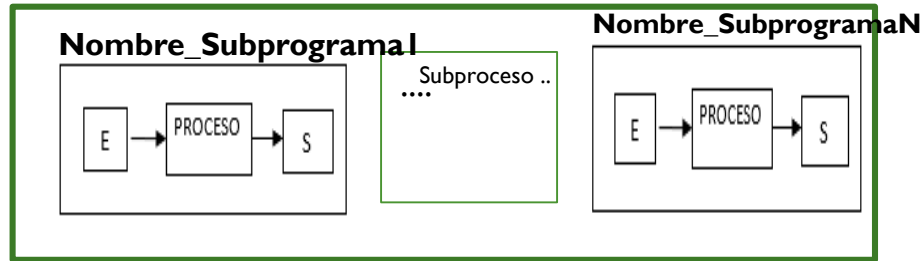


El problema se divide en subproblema

El algoritmo se divide en subprograma

2° PASO

Identificar los Subproblemas (Subprocesos – Subprogramas) –
Descripción de cada uno



Leer base, altura

Sup_Rect : Calcula superficie del rectángulo. **Entrada** base y altura

Salida: Superficie

Per_Rect : Calcula perímetro del Rectángulo. **Entrada** : base, altura.

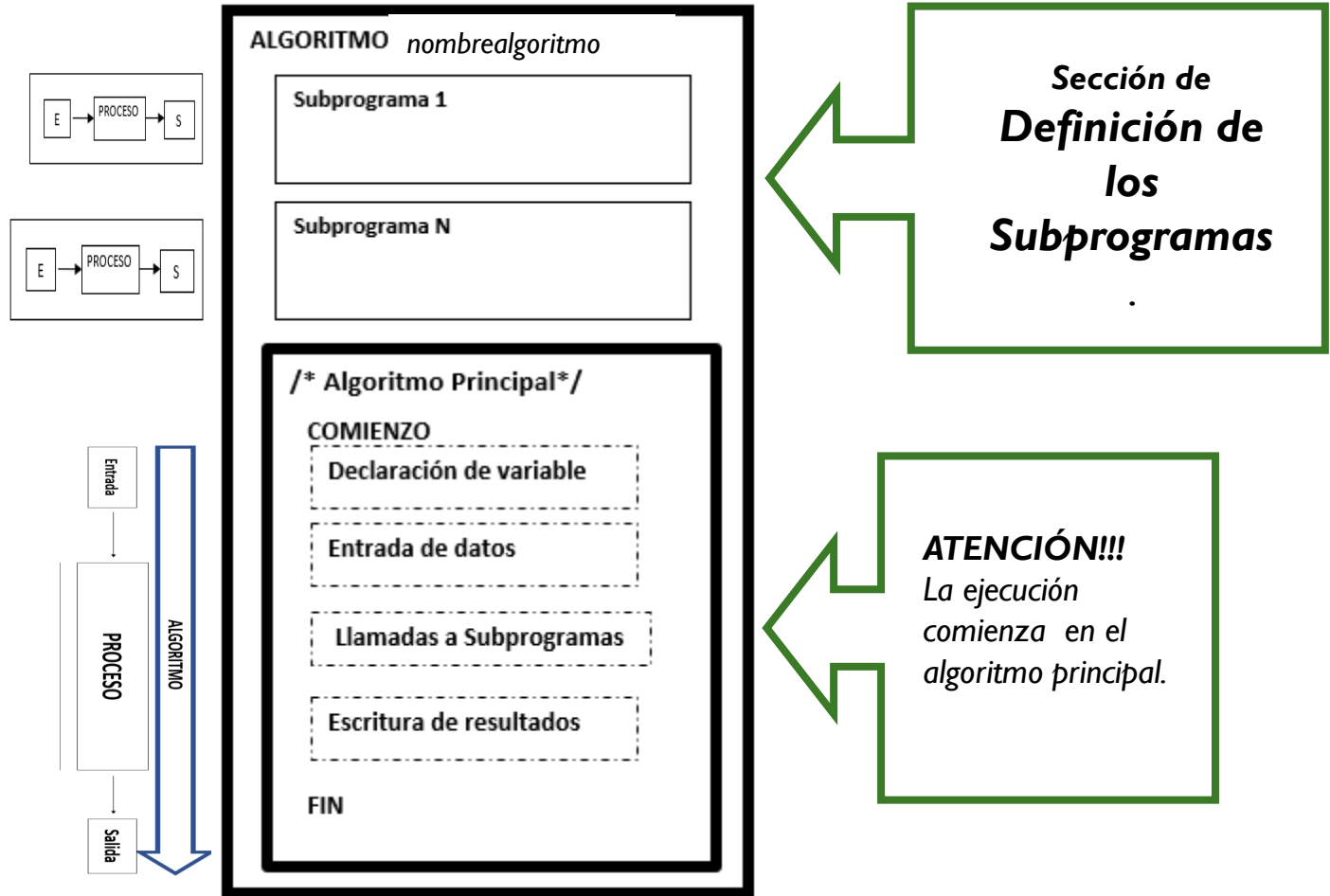
Salida: Perímetro

Mostrar: Informa resultados. **Entrada** : Superficie y Perímetro-

Salida: Mensaje

3° PASO

Elaboración del algoritmo principal



3° PASO

Elaboración del algoritmo principal

ALGORITMO nombre_algoritmo

SUBPROGRAMA

/-----Algoritmo principal -----*/*

Comienzo

declaración de variables locales

... Acciones -----

R = *nombre_subprograma (parámetros actuales)*

nombre_subprograma (parámetros actuales)

Escribir (nombre_subprograma (parámetros actuales))

Fin.

Se escribe las
llamadas a los
subprogramas

Parámetro
actuales o
reales
transfieren
los datos al
subprograma

3° PASO

Elaboración del algoritmo principal

En el ejemplo

ALGORITMO nombre_algoritmo

SUBPROGRAMA

/-----Algoritmo principal -----*/*

Comienzo

Real base, altura, superficie, perímetro

leer base, altura

superficie = **Sup_Rect** (*base, altura*)

perímetro = **Per_Rect** (*base, altura*)

Mostrar (*superficie, perimetro*)

Fin.

Se escribe las
llamadas a los
subprogramas

Parámetro
actuales o
reales
transfieren
los datos al
subprograma

4°

PASO

Definición de subprogramas

Caso I:

Retorna un sólo resultado

llamadas a los subprogramas

ALGORITMO nombre_algoritmo

/*-----definición de subprograma1-----*/

tipo de resultado nombre_subprograma1 (**parámetros formales**)

/*Encabezado*/

comienzo

declaraciones de variables locales

acciones

retorna(**resultado**)

fin

/*Cuerpo del subprograma*/

/*-----Algoritmo principal -----*/

Comienzo

R = nombre_subprograma (**parámetros actuales**)

En una asignación

Escribir (nombre_subprograma (**parámetros actuales**))

En una escritura

H= 4+ nombre_subprograma (**parámetros actuales**) - 9

En una expresión

Fin.

4°

Definición de subprograma

Caso 2:

**Retorna
0 (cero)
resultado**

**llamada al
subprogramas**

ALGORITMO nombre_algoritmo

/*-----definición de subprograma1-----*/

Void nombre_subprograma1(**parámetros formales**)

comienzo

declaraciones de variables locales

acciones

Retorna ()

fin

/*Encabezado*/

**/*Cuerpo del
subprograma*/**

/*-----Algoritmo principal -----*/

Comienzo

declaración de variables locales

...

nombre_subprograma (parámetros actuales)

.....

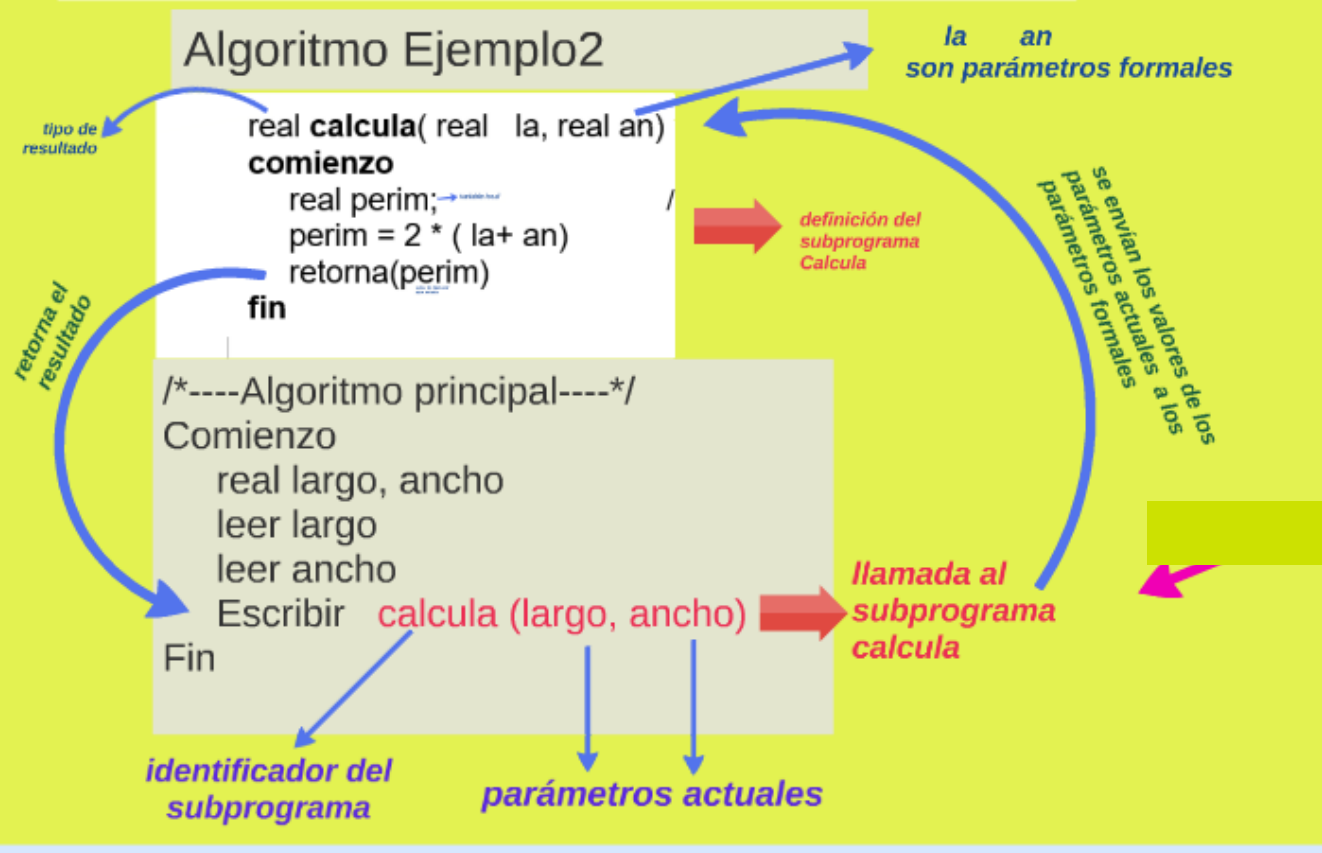
Fin.

¿qué es un subprograma?

Es un algoritmo que permite resolver una **tarea específica** y que retorna como resultado **un valor o ninguno.**

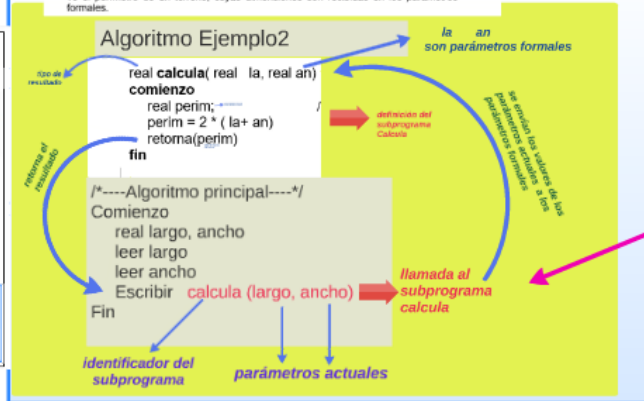
Ejemplo 2

El siguiente ejemplo corresponde a la definición del subprograma *calcula*, que devuelve el perímetro de un terreno, cuyas dimensiones son recibidas en los parámetros formales.



Ejemplo 2

El siguiente ejemplo corresponde a la definición del subprograma calcula, que devuelve el perímetro de un terreno, cuyas dimensiones son recibidas en los parámetros formales.



Si el subprograma retorna un resultado la llamada debe formar parte de :

como parte de una acción de escribir

en una asignación

como parte de una expresión

```
/*----Algoritmo principal----*/
Comienzo
  real largo, ancho, p
  leer largo
  leer ancho
  p = calcula (largo, ancho)
  escribir " el perímetro es ", p
Fin
```

```
/*----Algoritmo principal----*/
Comienzo
  real largo, ancho, p
  leer largo
  leer ancho
  H= calcula (largo, ancho) * 2
  escribir H
Fin
```

```
/*----Algoritmo principal----*/
Comienzo
  real largo, ancho, p
  leer largo
  leer ancho
  Si calcula (largo, ancho) > 200
    entonces escribir "el perímetro es mayor a 200"
Fin
```

PARA RECORDAR

Cuando se usan subprogramas es importante tener en cuenta que:

- los parámetros reales deben coincidir en tipo, orden y cantidad con los parámetros formales.
- el cuerpo del subprograma debe incluir al menos una acción de retorno que denominaremos **retorna**, para retornar el control al punto de llamada o invocación
- La acción **retorna** permite que se devuelva un resultado, el cual colocaremos entre paréntesis: **retorna(resultado)**
- Si el subprograma no retorna un resultado, la acción **retorna** la escribiremos **retorna ()**

en el encabezado de un subprograma:

Si el subprograma **no incluye parámetros**, el identificador del subprograma puede ir seguido de un **par de paréntesis vacíos** o con la palabra reservada **void**.

```
tipo nombresubp ( )  
tipo nombresubp (void)
```

Si el subprograma **no devuelve resultado**, se coloca la palabra reservada **void** (void significa carente de resultado), en caso de devolver un resultado debe especificarse su tipo.

```
void nombresubp (parametros formales )
```



Seguimientos de algoritmos con subprogramas

Para realizar el seguimiento de un algoritmo que utilice subprogramas, se debe representar las variables y los parámetros en cuadros diferentes según sean del algoritmo principal o de los subprogramas. Esto es debido a que cuando se ejecute el algoritmo, el programa principal y cada subprograma se almacenan en distintos lugares de memoria.

Ejemplo 25

Realizar el seguimiento del siguiente algoritmo para el lote de prueba: 10, 5

Algoritmo rectangulo1

real **calcula** (real la, real an) /*la y an son parámetros formales*/

Comienzo

real perim; /* perim es una variable local del subprograma perímetro */
perim = 2 * (la + an)
retorna(perim)

Fin

Comienzo /*----Algoritmo principal----*/

real largo, ancho
Leer largo
Leer ancho
Escribir "El perímetro del rectángulo es", calcula (largo, ancho)

Fin

Algoritmo principal		
Largo	ancho	Salida
10	5	El perímetro del rectángulo es 30

Subprograma Calcula			
la	an	perim	Salida
10	5	30	