ITESO
Universidad Jesuita de Guadalajara
50 AÑOS

Algoritmos y solución de problemas

Fundamentos de Programación

Otoño 2008

Mtro. Luis Eduardo Pérez Bernal

Introducción

- En las ciencias de la computación se ocupa de los **problemas computables**.
- Se le llama **problema computable** a aquella abstracción de la realidad que tiene representación algorítmica.
- Los **algoritmos** permiten encontrar la solución a problemas computables.
- Intuitivamente las personas efectuamos cotidianamente una serie de pasos, procedimientos o acciones que nos permitan alcanzar algún resultado o resolver un problema (al bañarnos, al desayunar, al ir a la universidad). En realidad todo el tiempo estamos aplicando algoritmos para resolver problemas.

Algoritmo

- Es un método para la resolución de problemas.
- Es un conjunto de pasos a seguir para la solución a un problema.
- Es una serie finita de instrucciones para realizar una tarea.

- Formalmente:

  Es un conjunto de pasos, procedimientos o acciones que nos permiten alcanzar un resultado o resolver un problema.

Características de los algoritmos

- Las características que debe cumplir un algoritmo son:
  - Un algoritmo debe ser ***Preciso*** e indicar el órden de realización de cada paso.
  - Un algoritmo debe ser ***Definido***, es decir, si se sigue un algoritmo dos veces, se debe obtener el mismo resultado.
  - Un algoritmo debe ser ***Finito***, es decir, si se sigue el algoritmo se *debe terminar el algún momento*.

Otras características de los algoritmos

Debe cumplir con:

- Una secuencia de instrucciones claras y finitas
- Debe ser correcto y debe resolver el problema planteado en todas sus facetas
- Debe ser legible

Resolver problemas

- ¿Qué tipo de problemas se pueden resolver?
  - Computables

- ¿Qué métodos hay para resolver problemas computables?
  - Metodología de la programación (centrado en los algoritmos)

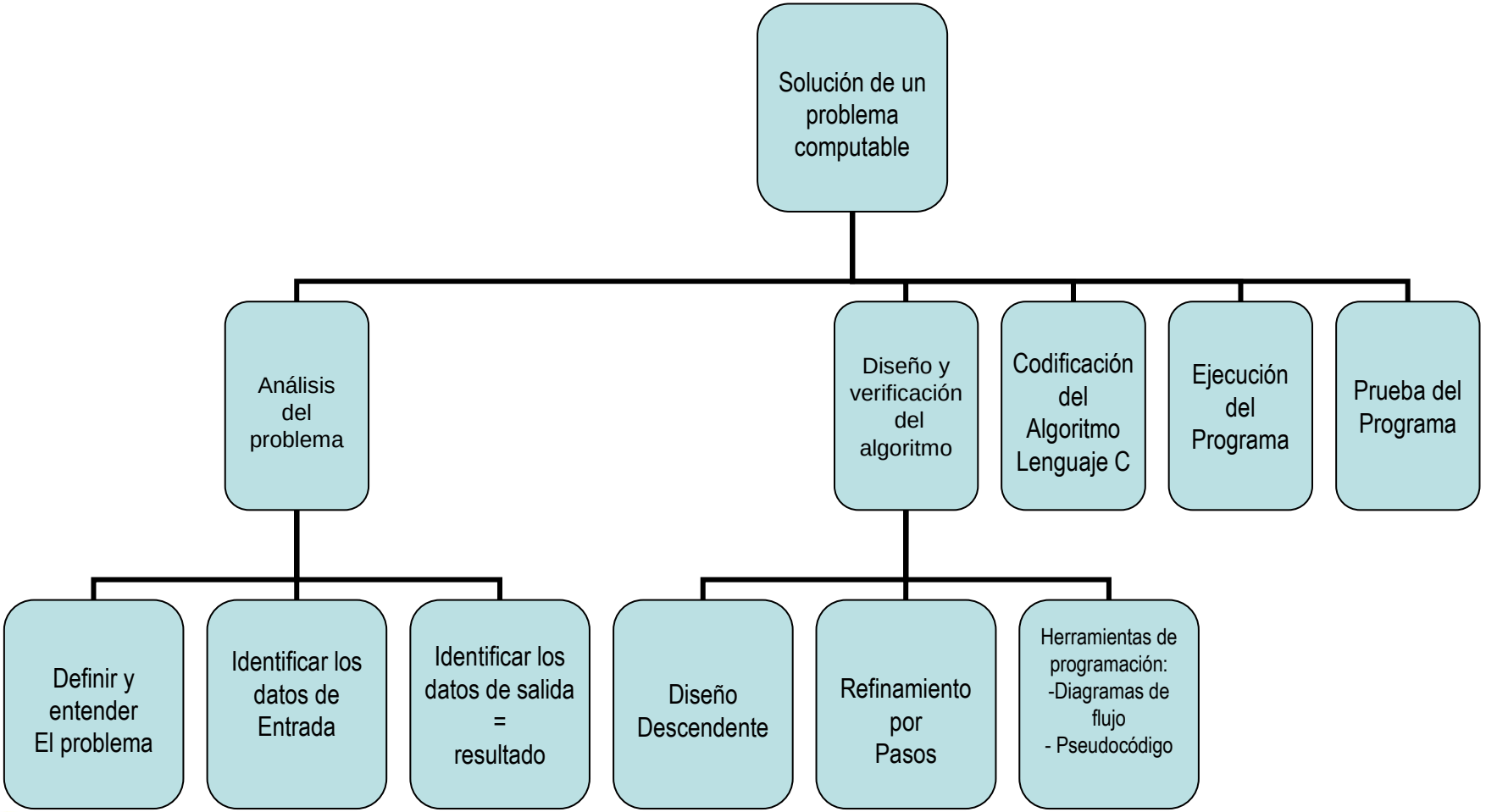
Fases para resolver un problema computable

- Diseño de programas
  - **Análisis** del problema
  - **Diseño** del algoritmo
  - Verificación manual del algoritmo
- En la computadora
  - **Codificación** del algoritmo
  - **Ejecución** del programa
  - **Verificación** del programa
  - **Mantenimiento** (documentación)

- Análisis
- Diseño (descendente, refinamiento paso a paso)

- Codificación
- Ejecución
- Prueba
- Mantenimiento

Fases para la solución de un problema computable

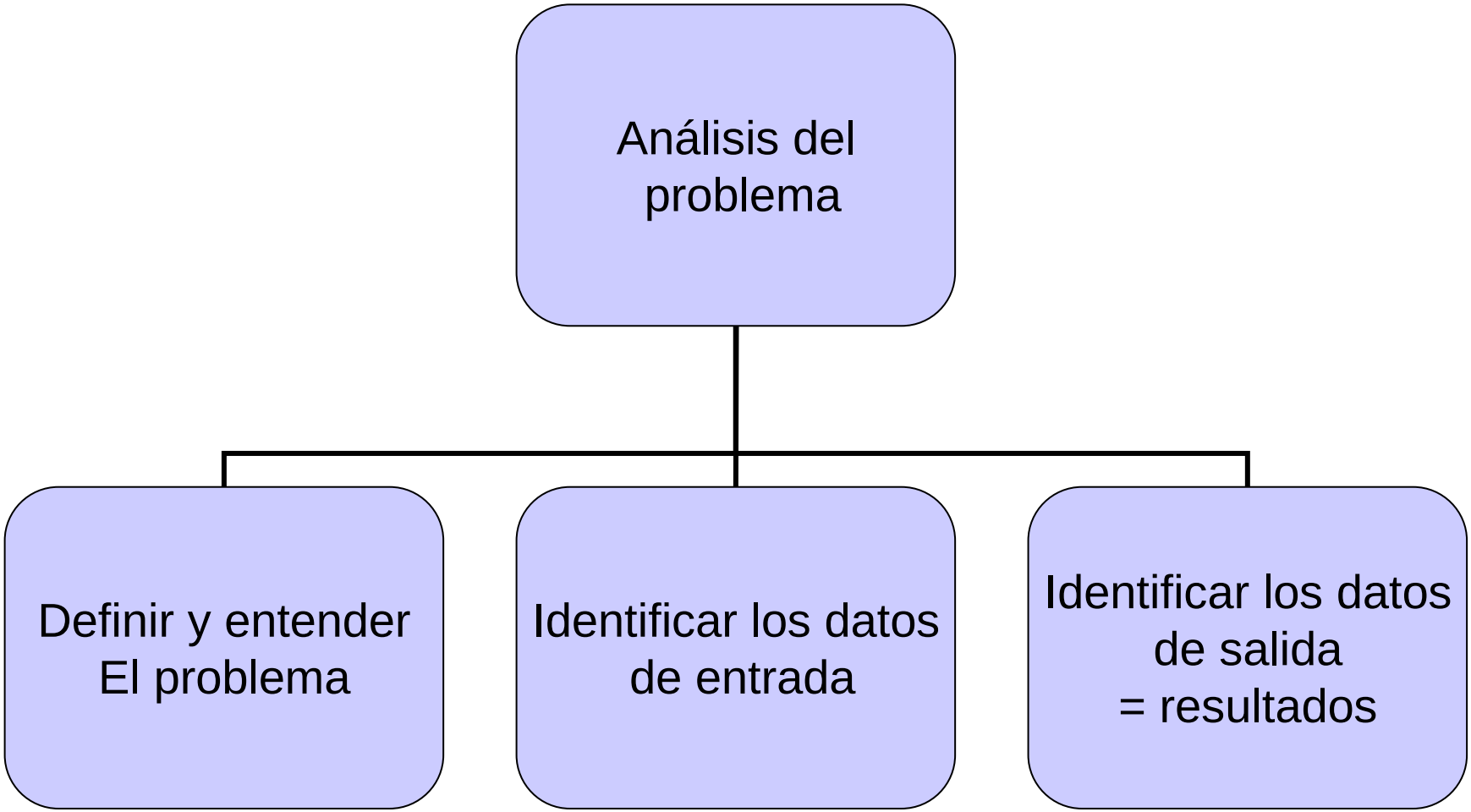


ITESO

Análisis del problema

- Es el primer paso a seguir para encontrar la solución a un problema computable es el *análisis del problema*.
- En el *análisis del problema* se requiere del máximo de creatividad e imaginación.
- Debido a que se busca una solución se debe examinar cuidadosamente el problema a fin de **identificar** que tipo de información es necesaria producir. En seguida se deben identificar aquellos elementos de información ofrecidos por el problema y que resulten útiles para obtener la solución al problema.
- Finalmente, un procedimiento para producir los resultados deseados a partir de los datos, es decir, el **algoritmo**.

Análisis del problema

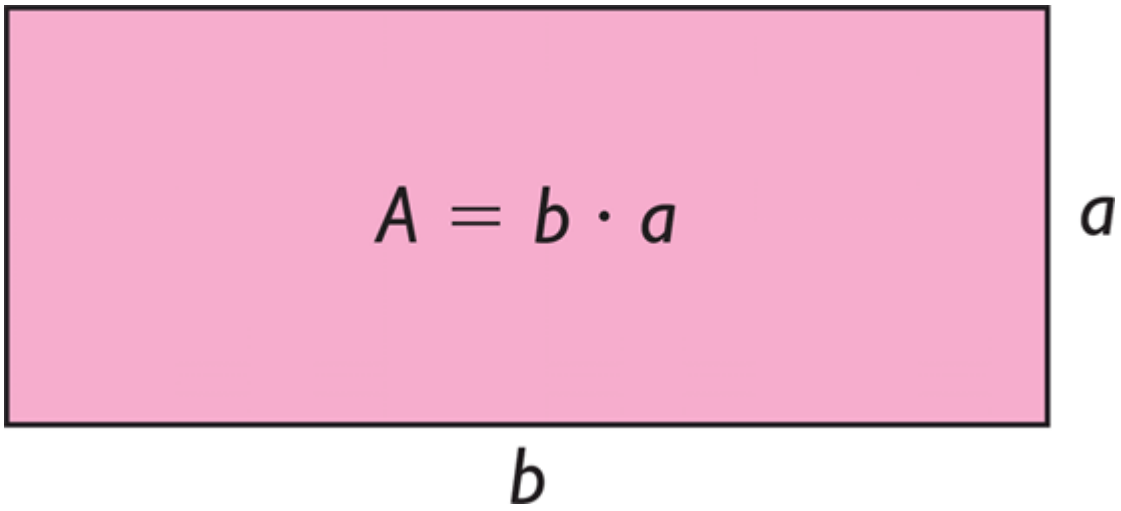


Elementos que conforman un algoritmo

- **Entrada.** Los datos iniciales que posee el algoritmo antes de ejecutarse.
- **Proceso.** Acciones que lleva a cabo el algoritmo.
- **Salida**. Datos que obtiene finalmente el algoritmo.

Ejemplo: calcular el área de un rectángulo

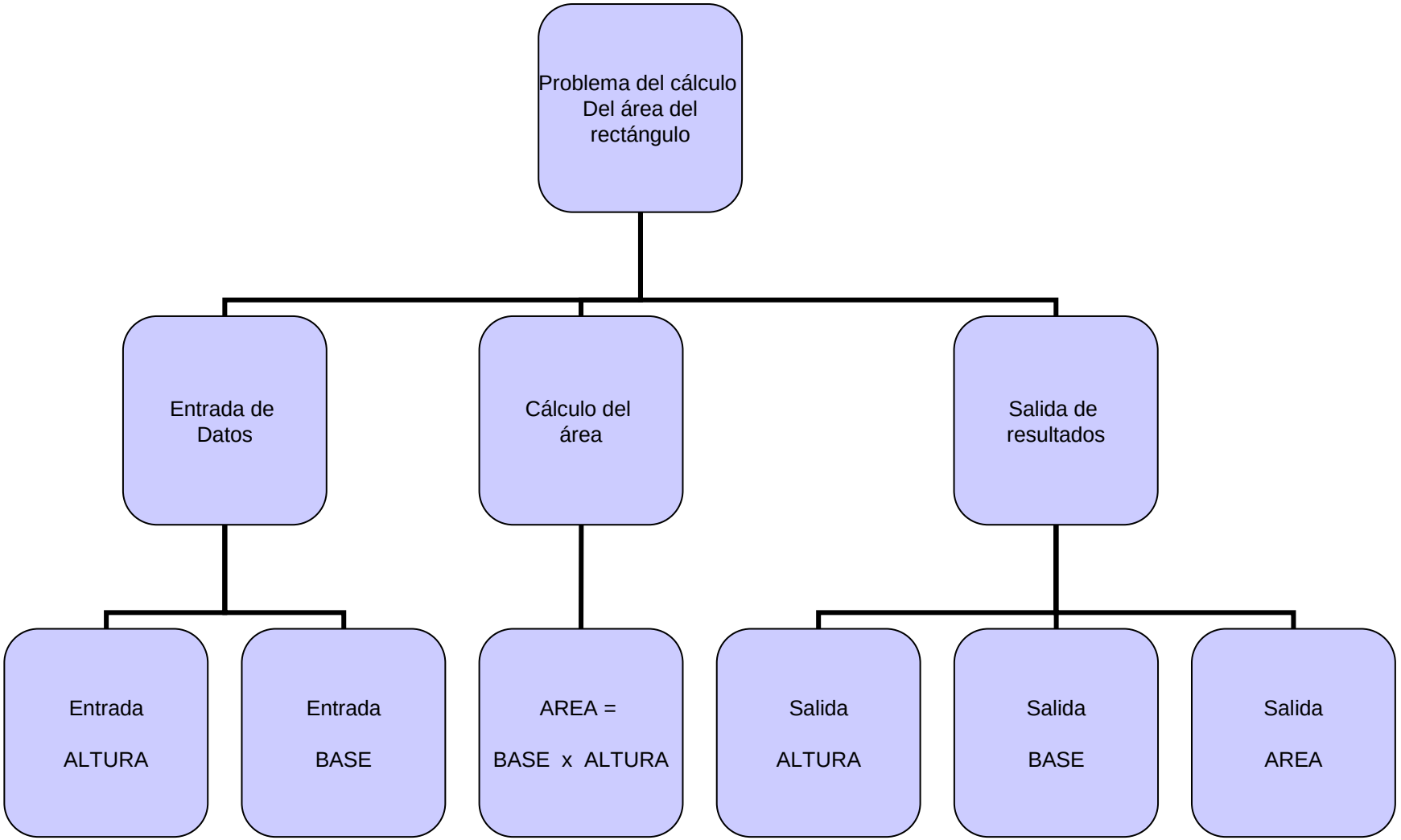
- Análisis del problema
  - El cálculo del área del rectángulo se puede dividir en:
    - **Entrada** de datos (altura, base)
    - **Proceso:** Cálculo del área (= base x altura)
    - **Salida** de datos (base, altura, área)



Diseño del algoritmo

- La solución de un problema complejo puede requerir muchos pasos, es necesario dividir el problema en subproblemas más sencillos de resolver.
- Este método se denomina *divide y vencerás* y es aplicable a la resolución y escritura de algoritmos y programas para computadora.
- Este método de división de un problema en otros subproblemas más sencillos se puede expresar para conseguir su solución en una computadora, mediante el método denominado **diseño descendente**.
- El proceso de la rotura de un problema principal en etapas o subproblemas más sencillos se denomina **refinamiento paso a paso o sucesivos.**

Diseño descendente y refinamiento paso a paso



Herramientas de programación

- Las herramientas de programación utilizadas como lenguajes algorítmicos son:
  - **Pseudocódigo**: es un lenguaje algorítmico, muy parecido al español pero más conciso que permite la redacción rápida del algoritmo.
  - **Diagramas de flujo**: ha sido la herramienta de programación por excelencia, y aún hoy sigue siendo muy utilizada. Es fácil de diseñar pues el flujo lógico del algoritmo se muestra en un diagrama en lugar de palabras.

Pseudocódigo

- Es un lenguaje de *pseudoprogramación*, es decir, muy parecido a un lenguaje de programación.
- El **pseudocódigo** es muy fácil de utilizar, ya que es muy similar al español.
- Algunas palabras utilizadas en el pseudocódigo:
  - Inicio
  - Fin
  - Leer
  - Escribir
  - Asignar (x← y+z)

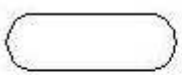
Ejemplo de pseudocódigo

```
Programa CalculoAreaRectangulo
Inicio
    leer; base, altura
    area←base x altura
    escribir; base, altura, area
Fin
```

Diagramas de flujo

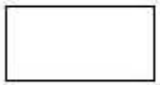
- Un diagrama de flujo utiliza símbolos estándar en el que cada paso del algoritmo se visualiza dentro del símbolo y en el orden en que estos pasos se ejecutan, se indica conectándolos con flechas llamadas *líneas de flujo*, ya que indican el flujo lógico del algoritmo.
- Los símbolos utilizados en los diagramas de flujo han sido estandarizados por la ANSI (American National Institute) y por la ISO (International Standard Organization)

Símbolos de diagramas de flujo



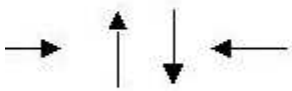
Terminal (representa el Inicio y el Final, de un programa, puede representar también una parada o interrupción programada que sea necesario realizar en un programa.

Entrada/Salida (cualquier tipo de introducción de datos)



Proceso (cualquier tipo de operación que pueda originar cambio de valor, formato o posición de la información almacenada en memoria, operaciones aritméticas).

Conector (Sirve para enlazae dos partes cualesquiera de un organigrama a través de un conector en la salida y otro conector en la salida)

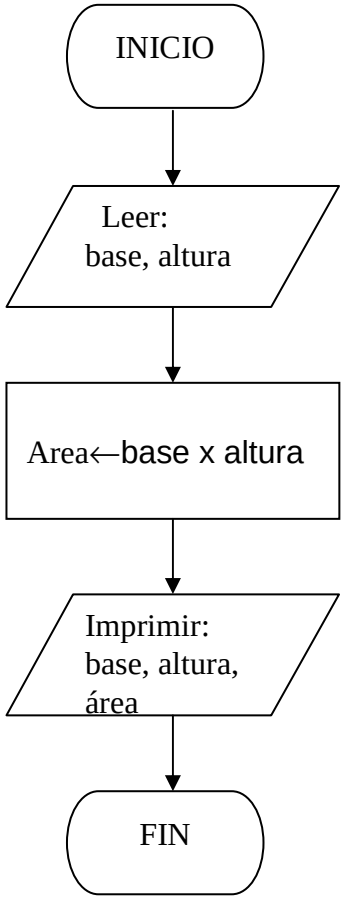


Indicador de dirección o línea de flujo (indica el sentido de ejecución de las operaciones)

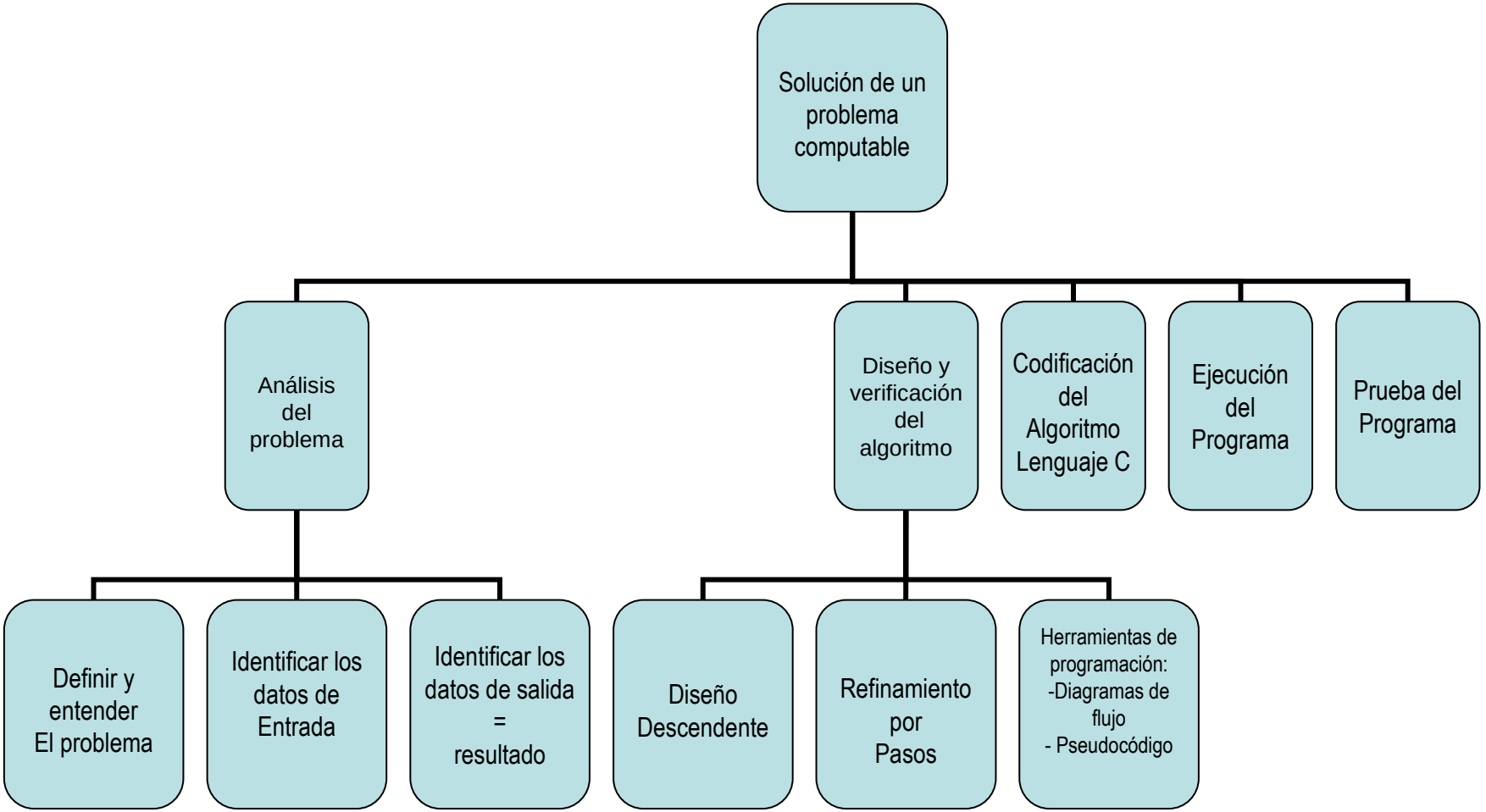
Línea conectora (sirve de unión entre dos símbolos).

ITESO

Ejemplo: Diagrama de flujo



Fases para la solución de un problema computable



ITESO