RESOLUCIÓN DE PROBLEMAS COMPUTACIONALES I
PROBLEMAS – ALGORITMOS - PROGRAMAS

PROBLEMAS

- La principal razón por la que las personas aprenden a programar es para utilizar el ordenador como una herramienta de resolución de problemas.
- Ejemplo de problemas:
  - Determinar el producto de dos números a y b
  - A partir del peso de un producto y de su precio por kilogramo calcular el precio final.
  - Calcular a partir de la cantidad de dinero entregada y del precio de un producto producto, la cantidad de dinero que debe ser devuelta.
  - Calcular la media de 10 números dados.
  - Calcular el número de palabras repetidas en una lista

ALGORITMOS

- ¿Qué necesitamos para resolver un problema?
  - Una secuencia ordenada de pasos que conduzcan a la solución de un problema concreto, sin ambigüedad y en un tiempo finito.
  - Ejemplo
    1. Leer precioKg
    2. Leer peso
    3. precioFinal ← peso x precioKg
    4. Escribir precioFinal

MÁS SOBRE ALGORITMOS

- La palabra algoritmo se deriva de la traducción al latín de la palabra Alkhô-warîzmi
- Nombre de un matemático y astrónomo árabe que escribió un tratado sobre manipulación de números y ecuaciones en el siglo IX .

MÁS SOBRE ALGORITMOS

- Debe cumplir las siguientes condiciones:
  - Precisión:
    - Debe indicar el orden de realización de cada acción, de forma clara y sin ambigüedades.
    - Sólo el número de pasos precisos para llegar a la solución (no deben darse pasos de más).
  - Repetitividad:
    - Si se sigue el algoritmo dos veces con los mismos datos de entrada, se obtendrán los mismos datos de salida independientemente del momento de ejecución
  - Finitud:
    - El algoritmo debe terminar en algún momento.
    - Si el algoritmo nunca acaba no obtendremos ninguna solución y, como se ha señalado, el objetivo principal de un algoritmo es obtener la solución de un problema.

MÁS SOBRE ALGORITMOS

- Otro ejemplo:
  - Algoritmo para subir una escalera:
    - Levantar un pie hasta la altura del primer escalón
    - Adelantarlo hasta apoyarlo en dicho escalón
    - Levantar el otro pie hasta la altura del escalón siguiente.
    - Repetir los dos pasos anteriores mientras queden escaleras

MÁS SOBRE ALGORITMOS

- Medición de la calidad de un algoritmo
  - Validez:
    - El algoritmo construido hace exactamente lo que se pretende hacer.
  - Eficiencia:
    - El algoritmo debe dar una solución en un tiempo razonable.
    - Por ejemplo, para sumar 20 a un número dado podemos dar un algoritmo que sume uno veinte veces, pero esto no es muy eficiente. Sería mejor dar un algoritmo que lo haga de un modo más directo.
  - Optimización:
    - se trata de dar respuesta a la cuestión de si el algoritmo diseñado para resolver el problema es el mejor.

PROGRAMAS

Para que una computadora resuelva un problema según un algoritmo → éste tiene que ser convertido a un programa traduciéndolo a algún **lenguaje de programación.**

```
#Primer programa en python
#Leer los datos
n1 = int(input('Primer numero: '))
n2 = int(input('Segundo numero: '))
#Realizar los calculos
suma=n1+n2
#Obtener el resultado
print('La suma de los numeros es')
print suma
```

FASES EN LA RESOLUCIÓN DE PROBLEMAS

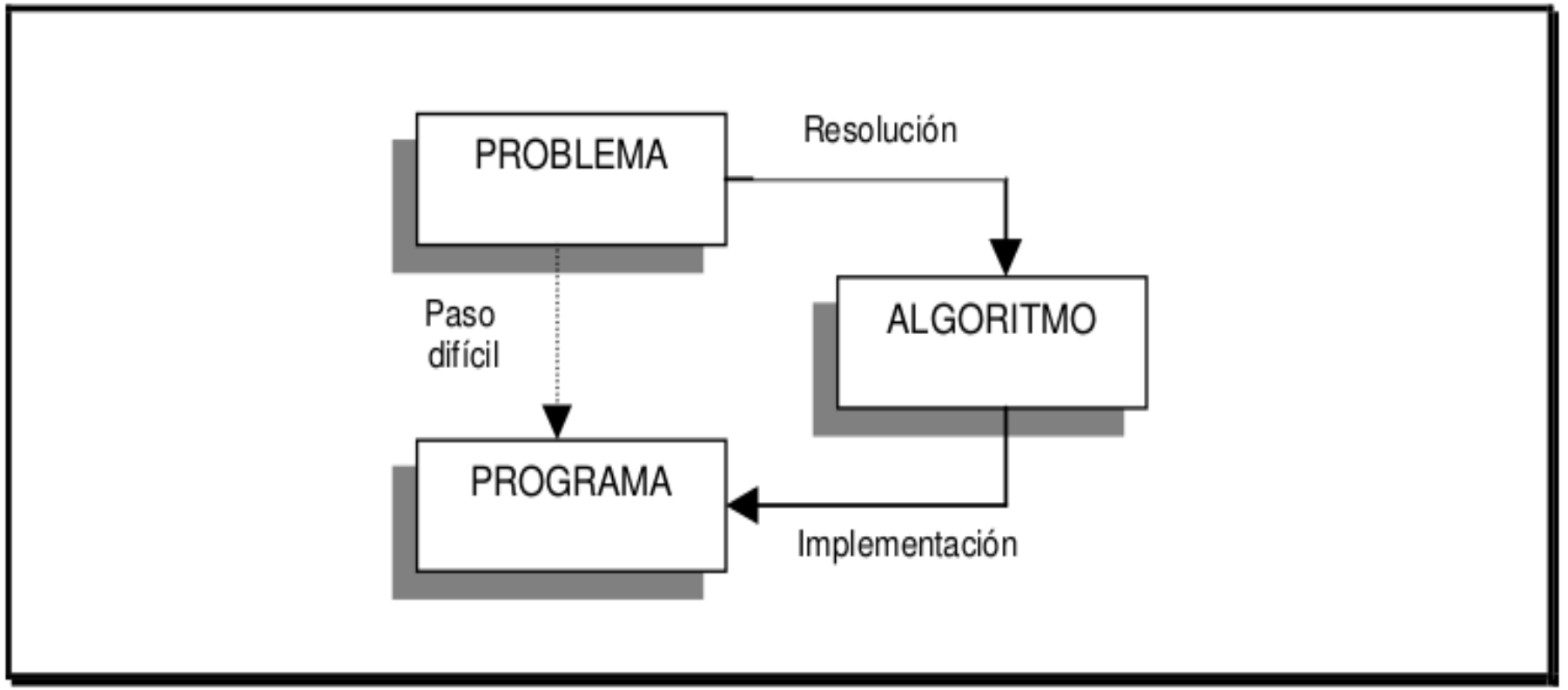

Figura 1. Fases de resolución e implementación

ALGORITMOS

- En el algoritmo se plasman las tres partes fundamentales de una solución informática:
  - Entrada
  - Proceso
  - Salida

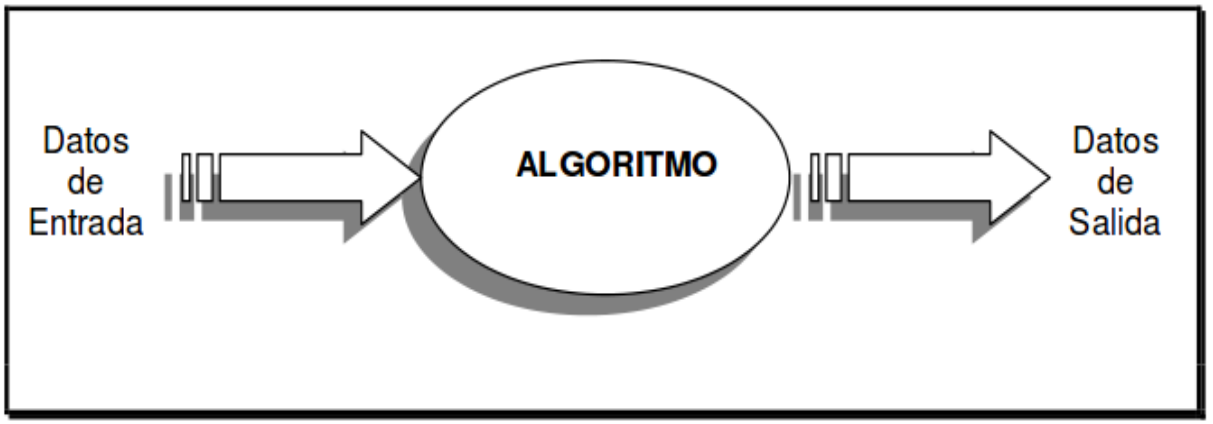

Figura 2. Entrada, algoritmo y salida

- Luego tendremos:
  - Instrucciones de entrada – LEER (Nombres variables)
  - Instrucciones de proceso
  - Instrucciones de salida – ESCRIBIR (mensajes, Nombres_Variables)

DATOS

- Para poder operar, un programa necesita datos.
- Un dato es un símbolo físico que se utiliza para representar la información.
- **Datos simples**
  - Números: 4.5, 6, 7.988
  - Caracteres: 'a', 'b', '@'
  - Valores lógicos: *true*, *false*
- **Colecciones**
  - Arrays: [3,4,5,6,7], [a,b,c,d,e]
  - Cadenas de caractéres: “hola”, “ qué tal”
  - Tablas: 3 4 5 6

    5 5 6 7
  - Listas, pilas, colas, árboles....