

# RESOLUCIÓN DE PROBLEMAS COMPUTACIONALES II FASES

# FASES EN LA RESOLUCIÓN DE PROBLEMAS

1. Análisis del problema.
2. Diseño del algoritmo.
3. Verificación del algoritmo.

# ANÁLISIS DEL PROBLEMA

- El primer objetivo que nos debemos plantear es obtener una correcta comprensión de la naturaleza del problema.
  - 1) ¿Qué información debe proporcionar la resolución del problema? → Datos de salida
  - 2) ¿Qué datos se necesitan para resolver el problema? → Datos de entrada

# ANÁLISIS DEL PROBLEMA. EJEMPLO

- Ejemplo para el problema de la báscula
  - La información de entrada para el problema es el peso y precio por kilogramo del producto
  - Las salidas del programa el precio final del producto y la cantidad devuelta

# DISEÑO DE ALGORITMOS

- Una vez que el problema ha quedado bien definido -> debemos plantearnos buscar una secuencia de pasos que lo resuelvan e indiquen al ordenador las instrucciones a ejecutar.
- Ejemplo:  
$$\text{precio} \leftarrow \text{peso} \times \text{precio por kilogramo}$$

# DISEÑO DE ALGORITMOS

- Sin embargo, las soluciones a problemas más complejos pueden requerir muchos más pasos.
- Estrategias más usuales:
  - **Partición o divide y vencerás:**  
Consiste en dividir un problema grande en unidades más pequeñas que puedan ser resueltas individualmente.
  - **Resolución por analogía:**  
Dado un problema, se trata de recordar algún problema similar que ya esté resuelto.
- Evidentemente la conjunción de ambas técnicas hace más efectiva la labor de programar: dividir un problema grande en trozos más pequeños ya resueltos.

# DISEÑO DE ALGORITMOS

- Diseño descendente o diseño por refinamientos sucesivos:
  - Método de resolución de problemas no triviales
  - 1) Se comienza el proceso con un enunciado muy general o abstracto de la solución del problema, de donde se identifican las tareas más importantes a ser realizadas, y el orden en el que se ejecutarán.
  - 2) A continuación se procede repetidamente refinando por niveles, de manera que con cada descomposición sucesiva se obtiene una descripción más detallada incluyendo nuevas acciones a realizar.
  - 3) El proceso finaliza cuando el algoritmo esté lo suficientemente detallado y completo para ser traducido a un lenguaje de programación.

# DISEÑO DE ALGORITMOS

- Ejemplo: Vamos a diseñar un algoritmo para cambiar una bombilla fundida.



# VERIFICACIÓN DEL ALGORITMO

- Recorrer todos los caminos posibles del algoritmo comprobando en cada caso que se obtienen los resultados esperados.
- Realizaremos una ejecución manual del algoritmo con datos significativos que abarquen todo el posible rango de valores y comprobaremos que la salida coincide con la esperada en cada caso.