

Metodología de la programación

Indice

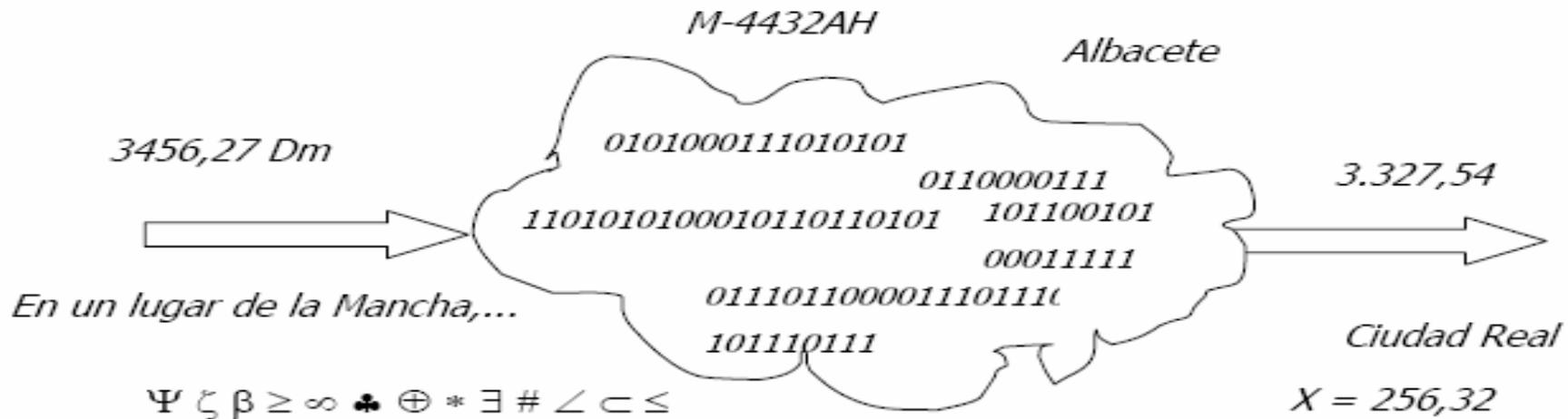
- Qué es un Computador
 - Estructura funcional
- Resolución de problemas con el computador
 - Ciclo de vida de los programas

¿Qué es un computador?

Máquinas que:

- **Leen** datos desde un dispositivo externo y los almacena para su uso posterior.
- **Manipulan** los datos almacenados mediante la ejecución de instrucciones que constituyen un programa.
- **Imprimen** los resultados en un dispositivo externo.

Ordenadores o Computadores



Estructura funcional de un ordenador

– Unidad de memoria

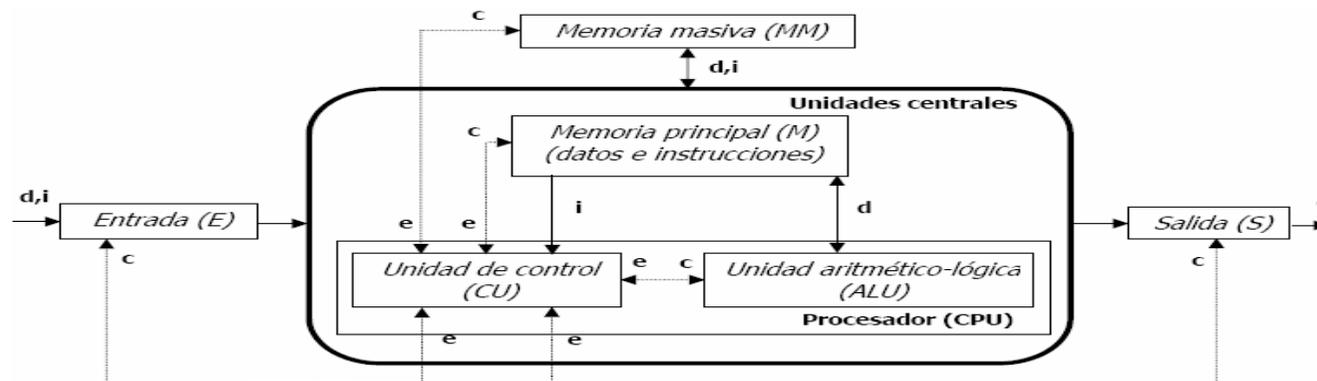
- Almacena datos e instrucciones de un programa.

– Unidad de procesamiento:

- Controla el comportamiento del computador por completo (registros).
- Realiza operaciones sobre los datos (ALU).

– Unidades de entrada y salida

- Ingresa o presenta la información hacia y desde la memoria.



*d: datos; i: instrucciones
e: señales de estado; c: señales de control*

Resolución de problemas con el computador

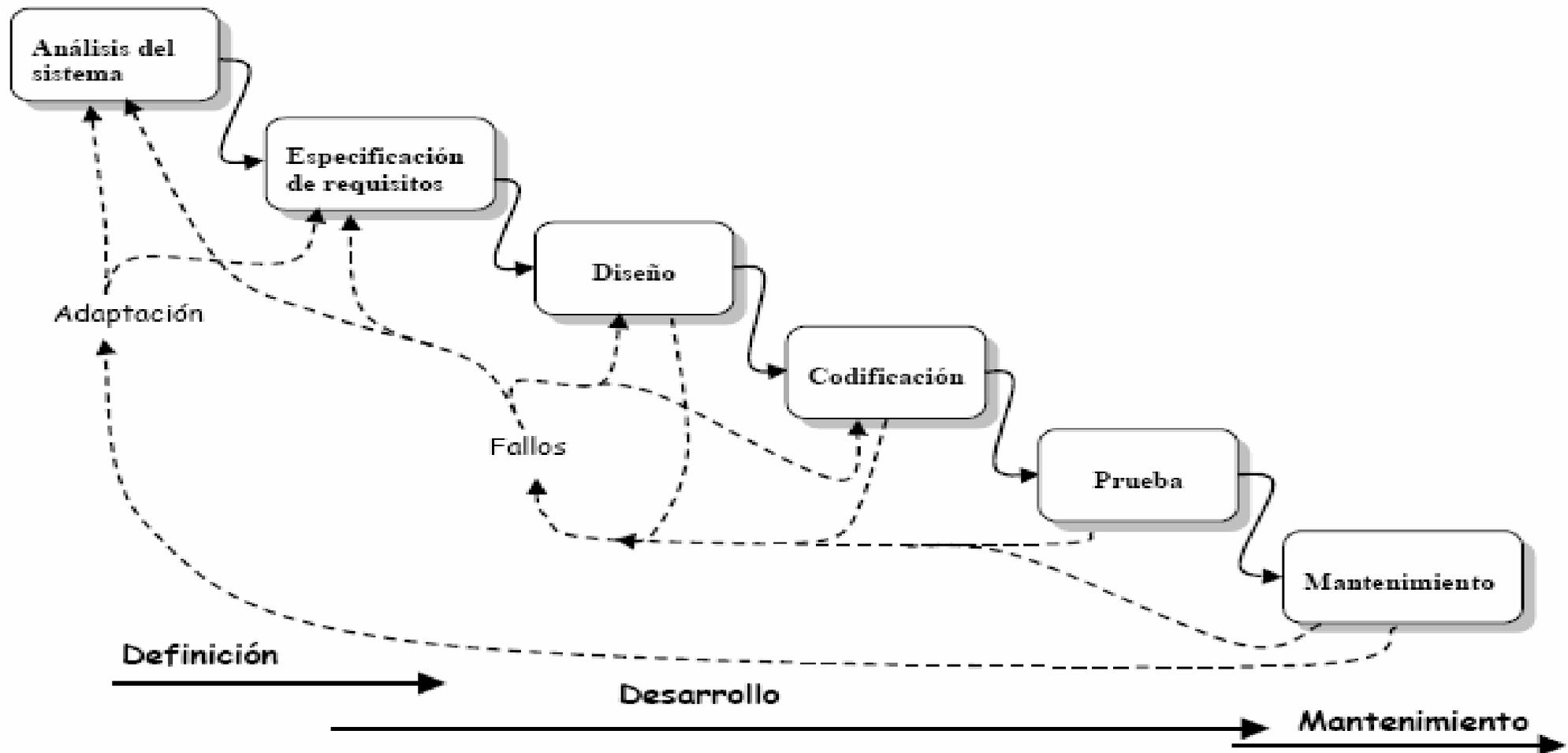
- Problema + computador \Rightarrow **algoritmo**.
- **Algoritmo** = secuencia de acciones precisas para obtener la solución de un problema.
- Ejemplos de algoritmos son:
 - Indicaciones para conseguir algo.
 - Una receta de cocina.
- Niveles de detalle de los algoritmos:
 - Alto nivel: no se dan detalles.
 - Bajo nivel: muchos detalles.

Resolución de problemas con el computador

- **Programas**
Conjunto de instrucciones especificadas en un lenguaje de programación concreto, que pueden ejecutarse en un ordenador y que resuelven un problema. Puede incluir uno o más algoritmos.
- **Lenguaje de programación**
Lenguaje formal utilizado para comunicarnos con un ordenador e imponerle la realización concreta de un conjunto de órdenes.
Se usa para *implementar* un algoritmo en un programa de ordenador. Hay lenguajes de alto nivel (C, Fortran, Lisp) y de bajo nivel (ensamblador).
- **Programación**
Actividad de resolver problemas mediante el computador.

Ciclo de vida de los programas (software)

- **Clásico**



Ciclo de vida clásico

- **Definición del problema:** Especificación completa del problema, incluyendo definición precisa de los datos de entrada y del resultado.
- **Diseño de la solución:** Definición del esquema de la solución.
 - Arquitectura de la solución
 - Funciones de cada módulo
 - Relaciones y comunicación entre módulos
 - Representación de los datos

Ciclo de vida clásico

- **Refinamiento de la solución:** Detallar la secuencia de pasos del algoritmo. Se usa para su desarrollo:
 - *pseudocódigo:*
 - mezcla de lenguaje natural y notación matemática.
 - independiente de los lenguajes de programación.
 - Refinamiento sucesivo: proceso de detallar las instrucciones hasta un nivel preciso y útil.
 - Es deseable verificar los datos de entrada. Se comprueba si los datos son aceptables.

Ciclo de vida clásico

- **Desarrollo de la estrategia de prueba:**
consiste en seleccionar los casos de prueba y los resultados esperados para *verificar* los algoritmos y programas. Se puede seguir un método formal.
- **Codificación y prueba del programa:**
Consiste en pasar el pseudocódigo a un lenguaje de programación específico y probarlo. Si hay errores se analiza el lugar donde se encuentra el error (*depuración*).

Ciclo de vida clásico

- Causas de error comunes:
 - Codificación.
 - Algoritmo.
 - Diseño del programa.
 - El cálculo del resultado del caso de prueba.
- **Documentación.** Se realiza durante todos los pasos. Debe incluir lo siguiente:
 - Explicación del método.
 - Decisiones del diseño.
 - Problemas encontrados en la escritura y prueba del programa.
 - Listado comentado del programa.
 - *Manual de Usuario*: describe el uso del programa.

Ciclo de vida clásico

- **Mantenimiento.** Incluye todas las actividades posteriores al uso del programa:
 - Eliminación de errores (mantenimiento correctivo).
 - Mejorar el programa (más velocidad, menos recursos, etc).
 - Adición de nuevas características o especificaciones (mantenimiento perfectivo).
 - Actualización de la documentación.

Ciclo de vida de los programas (software)

- **Proceso unificado**

