

Programa de teoría

Parte I. Estructuras de Datos.

1. Abstracciones y especificaciones.
2. Conjuntos y diccionarios.
3. Representación de conjuntos mediante árboles.
4. Grafos.

Parte II. Algorítmica.

1. Análisis de algoritmos.

2. Divide y vencerás.
3. Algoritmos voraces.
4. Programación dinámica.
5. Backtracking.
6. Ramificación y poda.

A.E.D.
Tema 0-2. Algorítmica

Algoritmos y Estructuras de Datos

PARTE II: ALGORÍTMICA (o ALGORITMIA)

Tema 0. Introducción

- 0.1. Definición y propiedades.
- 0.2. Análisis de algoritmos.
- 0.3. Diseño de algoritmos.

A.E.D.
Tema 0-2. Algorítmica

0.1. Definición y propiedades.

- **Algoritmo:**

Conjunto de reglas para resolver un problema.

- **Propiedades**

- **Definibilidad:** El conjunto debe estar bien definido, sin dejar dudas en su interpretación.
- **Finitud:** Debe tener un número finito de pasos que se ejecuten en un tiempo finito.



A.E.D.
Tema 0-2. Algoritmica

0.1. Definición y propiedades.

- **Algoritmos deterministas:** Para los mismos datos de entrada se producen los mismos datos de salida.
- **Algoritmos no deterministas:** Para los mismos datos de entrada pueden producirse diferentes de salida.
- **ALGORITMIA:** Ciencia que estudia técnicas para construir algoritmos eficientes y técnicas para medir la eficacia de los algoritmos.
- **Objetivo:** Dado un problema concreto encontrar la mejor forma de resolverlo.

A.E.D.
Tema 0-2. Algoritmica

0.1. Definición y propiedades.

Recordamos:

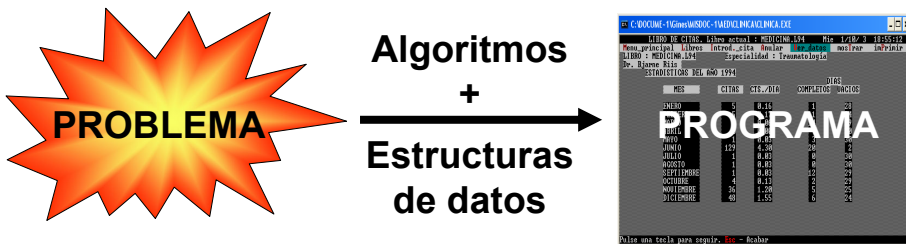
Objetivo de la asignatura

Ser capaz de **analizar, comprender y resolver** una amplia variedad de **problemas** de programación, diseñando soluciones **eficientes** y de **calidad**.

Pero **ojo**, los algoritmos no son el único componente en la resolución de un problema de programación.

A.E.D.
Tema 0-2. Algoritmica

0.1. Definición y propiedades.



Algoritmos + Estructuras de Datos = Programas

- **Estructura de datos:** Parte estática, almacenada.
- **Algoritmo:** Parte dinámica, manipulador.

A.E.D.
Tema 0-2. Algoritmica

0.1. Definición y propiedades.

MÉTODO CIENTÍFICO

INFORMÁTICO

1.Observación.	↔	1. Análisis del problema
2.Hipótesis.	↔	2. Diseño del programa (alg. y estr.)
3.Experimentación.	↔	3. Implementación (programación)
4.Verificación.	↔	4. Verificación y pruebas

A.E.D.
Tema 0-2. Algorítmica

0.1. Definición y propiedades.

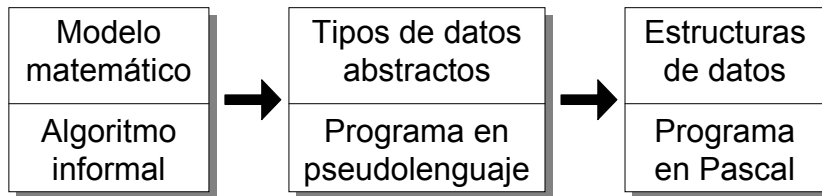
Otras ideas...

- **Refinamiento por pasos sucesivos.**
 - Escribir la estructura de la solución en pseudocódigo, de manera muy genérica.
 - Especificar los pasos de forma cada vez más detallada, y precisa.
 - Repetimos el refinamiento hasta llegar a una implementación.

A.E.D.
Tema 0-2. Algorítmica

0.1. Definición y propiedades.

- **Proceso de resolución propuesto por Aho.**



- Más en las asignaturas de Ingeniería del Software...

A.E.D.
Tema 0-2. Algoritmica

0.2. Análisis de algoritmos.

ALGORITMIA = ANÁLISIS + DISEÑO

- **Análisis de algoritmos:** Estudio de los recursos que necesita la ejecución de un algoritmo.
- No confundir con análisis de un problema.
- **Diseño de algoritmos:** Técnicas generales para la construcción de algoritmos.
- Por ejemplo, divide y vencerás: dado un problema, divídelo, resuelve los subproblemas y luego junta las soluciones.

A.E.D.
Tema 0-2. Algoritmica

0.2. Análisis de algoritmos.

- **Análisis de algoritmos.** Normalmente estamos interesados en el estudio del tiempo de ejecución.
- Dado un algoritmo, usaremos las siguientes notaciones:
 - $t(..)$: Tiempo de ejecución del algoritmo.
 - $O(..)$: Orden de complejidad.
 - $o(..)$: O pequeña del tiempo de ejecución.
 - $\Omega(..)$: Cota inferior de complejidad.
 - $\Theta(..)$: Orden exacto de complejidad.

A.E.D.
Tema 0-2. Algoritmica

0.2. Análisis de algoritmos.

- **Ejemplo.** Analizar el tiempo de ejecución y el orden de complejidad del siguiente algoritmo.

Hanoi (N, A, B, C: integer)

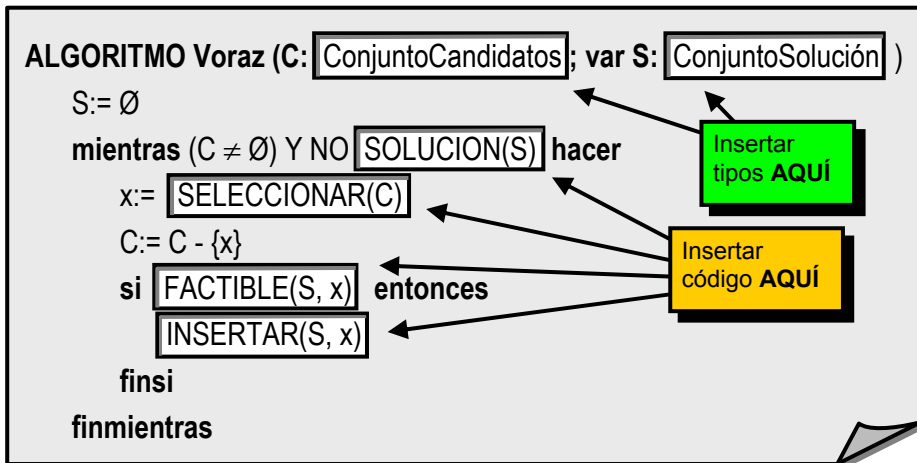
```
if N=1 then
  Mover (A, C)
else begin
  Hanoi (N-1, A, C, B)
  Mover (A, C)
  Hanoi (N-1, B, A, C)
end
```

- **Mecanismos:**
 - Conteo de instrucciones.
 - Uso de ecuaciones de recurrencia.
 - Medida del trabajo total realizado.

A.E.D.
Tema 0-2. Algoritmica

0.3. Diseño de algoritmos.

- **Diseño de Algoritmos.** Técnicas generales, aplicables a muchas situaciones.
- **Esquemas algorítmicos.** Ejemplo:



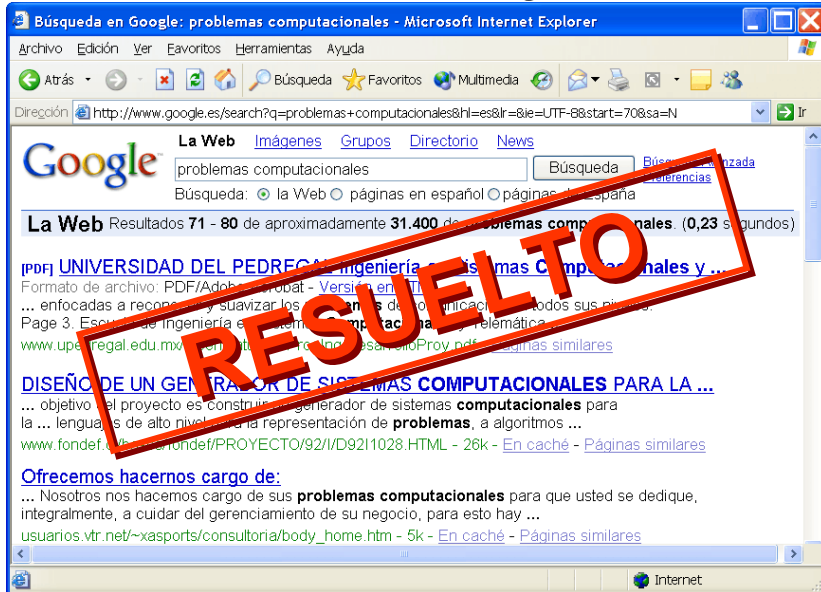
A.E.D.
Tema 0-2. Algorítmica

0.3. Diseño de algoritmos.

¿Qué clase de problemas?

A.E.D.
Tema 0-2. Algorítmica

0.3. Diseño de algoritmos.



Tema 0-2. Algoritmica

0.3. Diseño de algoritmos. Planificador de rutas



Tema 0-2. Algoritmica

0.3. Diseño de algoritmos.

- **EL JUEGO DE LAS CIFRAS.**

Dado un conjunto de 6 enteros, encontrar la forma de conseguir otro entero, utilizando las operaciones de suma, resta, producto y división entera (y sin usar cada número más de una vez).

A.E.D.
Tema 0-2. Algoritmica

0.3. Diseño de algoritmos.



A.E.D.
Tema 0-2. Algoritmica

0.3. Diseño de algoritmos.



A.E.D.
Tema 0-2. Algoritmica

cifras.exe

0.3. Diseño de algoritmos.

- **RETO.** Implementar un programa para resolver el problema, más rápido que la versión del profesor, y que no pierda ninguna solución.
- **RECOMPENSA.**
 - Un 10 en la tercera práctica de la asignatura (diseño de algoritmos).
 - Un 10 en el ejercicio correspondiente del examen.
 - Más 1 punto de notas de clase.

A.E.D.
Tema 0-2. Algoritmica

0.3. Diseño de algoritmos.

Jugador de Ajedrez

Situación Inicial



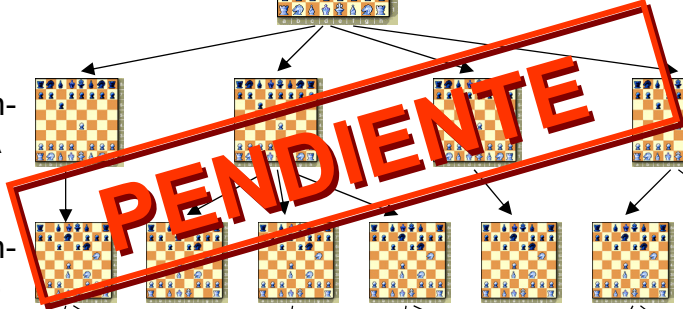
Movimientos de A



Movimientos de B



Movimientos de A



A.E.D.
Tema 0-2. Algoritmica

0.3. Diseño de algoritmos.

Problema del viajante



A.E.D.
Tema 0-2. Algoritmica

0.3. Diseño de algoritmos.

- **Técnicas de diseño** de algoritmos:
 1. Divide y vencerás.
 2. Algoritmos voraces.
 3. Programación dinámica.
 4. Backtracking.
 5. Ramificación y poda.
- **Dado un problema:** seleccionar la técnica, seguir el proceso/esquema algorítmico, obtener el algoritmo y comprobarlo.
- **Recordar:** No empezar tecleando código como locos.