

Algoritmos y Estructuras de Datos II

Tema 0. Introducción

0.1. Definición y propiedades

0.2. Análisis y diseño de algoritmos

0.3. Heurísticas para una buena programación

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.



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 eficiencia de los algoritmos.
- **Objetivo:** dado un problema concreto encontrar la mejor forma de resolverlo.

0.1. Definición y propiedades

Recordamos:

Objetivo de la asignatura

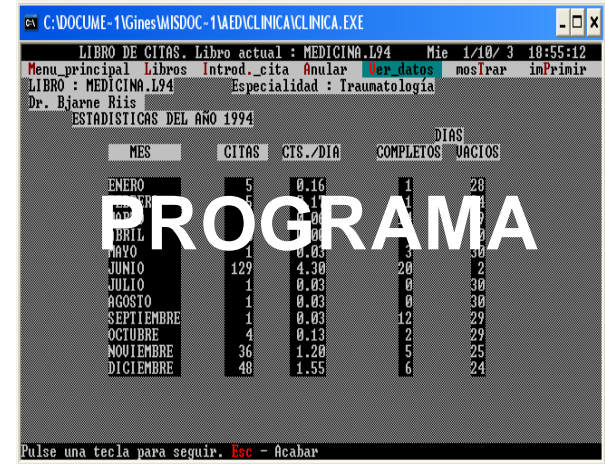
Ser capaces de **analizar, comprender y resolver** una amplia variedad de **problemas** de programación, diseñando soluciones **eficientes** y de **calidad**.

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

0.1. Definición y propiedades



Algoritmos
+
Estructuras
de datos



LIBRO DE CITAS. Libro actual : MEDICINA.L94 Mie 1/10/ 3 18:55:12
Menu_principal Libros Introd_cita Anular Ver_datos mostrar imprimir
LIBRO : MEDICINA.L94 Especialidad : Traumatologia
Dr. Bjarne Riis
ESTADISTICAS DEL AÑO 1994

MES	CITAS	CTS./DIA	DIAS	
			COMPLETOS	VACIOS
ENERO	5	0.16	1	28
FEBRERO	5	0.16	1	28
MARZO	5	0.16	1	28
ABRIL	5	0.16	1	28
MAYO	1	0.03	3	26
JUNIO	129	4.30	20	2
JULIO	1	0.03	0	30
AGOSTO	1	0.03	0	30
SEPTIEMBRE	1	0.03	12	29
OCTUBRE	4	0.13	2	29
NOVIEMBRE	36	1.20	5	25
DICIEMBRE	48	1.55	6	24

Pulse una tecla para seguir. Esc - Acabar

Algoritmos + Estructuras de Datos = Programas

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

0.1. Definición y propiedades

Resolver problemas

¿Cómo se resuelve un problema?

¿Cuándo se dice que la solución es eficiente y de calidad?

¿Qué clase de problemas?

0.1. Definición y propiedades

ARQUITECTO

1. Estudio de viabilidad, análisis del terreno, requisitos pedidos, etc.
2. Diseñar los planos del puente y asignar los materiales.
3. Poner los ladrillos de acuerdo con los planos.
4. Supervisión técnica del puente.

INFORMÁTICO

1. **Análisis** del problema
2. **Diseño** del programa (alg. y estr.)
3. **Implementación** (programación)
4. **Verificación** y pruebas

A.E.D.

Tema 0-2. Algorítmica

0.1. Definición y propiedades

MÉTODO CIENTÍFICO

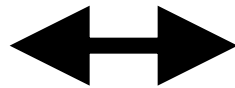
1. Observación.



2. Hipótesis.



3. Experimentación.



4. Verificación.



INFORMÁTICO

1. **Análisis** del problema

2. **Diseño** del programa
(alg. y estr.)

3. **Implementación**
(programación)

4. **Verificación** y pruebas

A.E.D.

Tema 0-2. Algorítmica

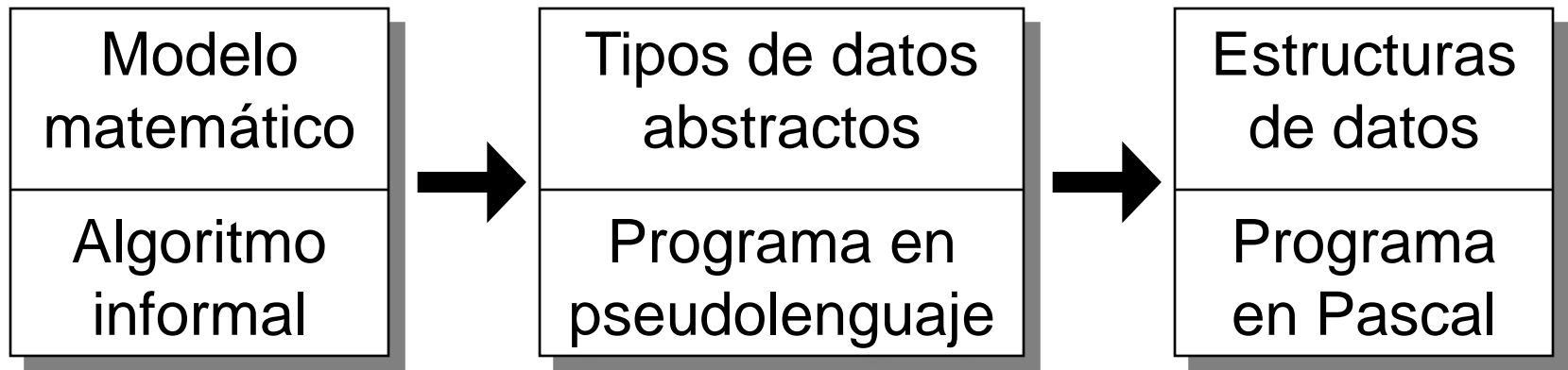
0.1. Definición y propiedades

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.

0.1. Definición y propiedades

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



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

0.2. Análisis y diseño 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.

0.2. Análisis y diseño 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.

0.2. Análisis y diseño 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.

0.2. Análisis y diseño de algoritmos

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

```
ALGORITMO Voraz (C: ConjuntoCandidatos; var S: ConjuntoSolución)
S := ∅
mientras (C ≠ ∅) Y NO SOLUCION(S) hacer
    x := SELECCIONAR(C)
    C := C - {x}
    si FACTIBLE(S, x) entonces
        INSERTAR(S, x)
    fin si
finmientras
```

Insertar tipos AQUÍ

Insertar código AQUÍ

0.2. Análisis y diseño de algoritmos

¿Qué clase de problemas?

0.2. Análisis y diseño de algoritmos

Búsqueda en Google: cualquier cosa - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

← Atrás → Búsqueda Favoritos Multimedia

Dirección <http://www.google.es/search?q=cualquier+cosa&ie=UTF-8&oe=UTF-8&hl=es&meta=>

Google [Búsqueda Avanzada](#) [Preferencias](#) [Herramientas del idioma](#) [Sugerencias de Búsqueda](#)

cualquier cosa Búsqueda en Google

Búsqueda: la Web páginas en español Páginas de España

[La Web](#) [Imágenes](#) [Grupos](#) [Directorio](#) [News](#) [¡Nuevo!](#)

Se buscó **cualquier cosa** en la Web. Resultados 1 - 10 aproximadamente de 4,00 La búsqueda tardó 0.27 segundos

[Cualquier cosa](#)
Cualquier Cosa. ... que finalmente apareció en mi mundo. ... viernes 26 de septiembre de 2009. ... bueno, sí. ... la ... los ...
www.cualquier.blogspot.com - 1k - En caché - Páginas similares

[Aprenda](#)
Aprenda **Cualquier cosa**...? [Esto está disponible solamente en inglés].
... usted ha leído tan a través del libro entero y ahora ...
library.thinkquest.com/29901/learn_sp.htm - 3k - En caché - Páginas similares

[EL TIEMPO.COM - Fútbol -> Iván Ramiro Córdoba: "Ronaldo ...](#)
... ELTIEMPO.COM Iván Ramiro Córdoba: "Ronaldo puede inventar **cualquier cosa**". ... De todas maneras, Ronaldo puede inventar **cualquier cosa** en el terreno de juego. ...
eltiempo.terra.com.co/.../RUMU/ELIM_R/NOTI_E/previasprimerafecha/ARTICULO-WEB-NOTA_INTERIOR-1239473.html - 37k - En caché - Páginas similares

Internet

0.2. Análisis y diseño de algoritmos

Planificador de rutas

The image shows a screenshot of a route planning application titled "Guía Campsa España 2000, INTERACTIVA". The interface includes a map of Spain with a network of roads, a sidebar with navigation tools, and a panel for location details. A large, 3D orange stamp with the word "RESUELTO" is overlaid diagonally across the map. At the bottom, there is an "Itinerario" section with a list of points: "Cagitan" (green dot) and "CORUÑA, LA / A CORUÑA" (red dot). A "Calcular ruta" button is visible, with a mouse cursor clicking on it. To the right of the button, there are three options: "Más Rápida", "Más Corta", and "Preferencias". A small inset map of Spain is located in the bottom right corner.

RESUELTO

Guía Campsa España 2000, *INTERACTIVA*

Población: Cagitan

Cagitan

Superficie: 45097 m²

Provincia: MURCIA

Itinerario

- Cagitan
- CORUÑA, LA / A CORUÑA

Calcular ruta

Más Rápida
Más Corta
Preferencias

0.2. Análisis y 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).

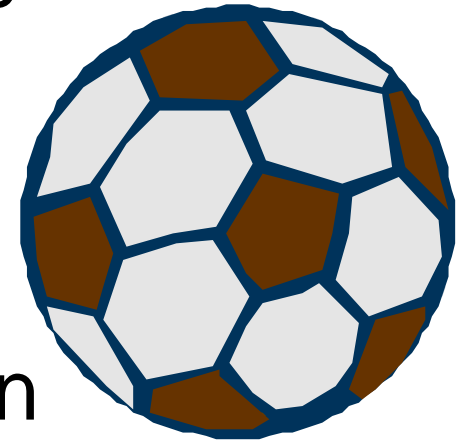
0.2. Análisis y diseño de algoritmos

- **Caso 1.** 1 1 5 2 10 7
990
- **Caso 2.** 6 8 10 9 4 75
835
- Implementación muy rápida.

0.2. Análisis y diseño de algoritmos

- **PROBLEMA DEL BALÓN DE FÚTBOL.** Se pinta una región (conexa) del balón colorando de verde algunos hexágonos o pentágonos del balón.

La región se describe moviéndose por el contorno de la misma, poniendo 1 ó 2 según el vértice tenga al lado 1 ó 2 trozos de verde.



A partir de la descripción, calcular el número de trozos verdes, blancos y negros.

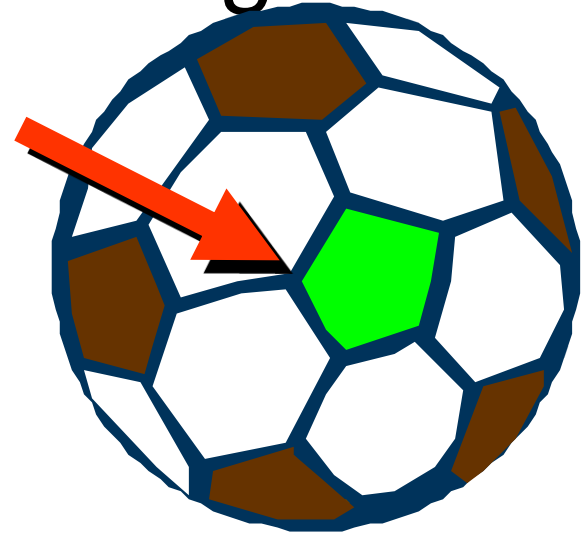
A.E.D.

0.2. Análisis y diseño de algoritmos

- **Caso 1.** 1, 1, 1, 1, 1

Resultado: 1 verde

11 negros, 20 blancos



- **Caso 2.** 2, 1, 1, 2, 1, 1,
1, 2, 1, 1, 1

Resultado: 3 verdes

11 negros, 18 blancos

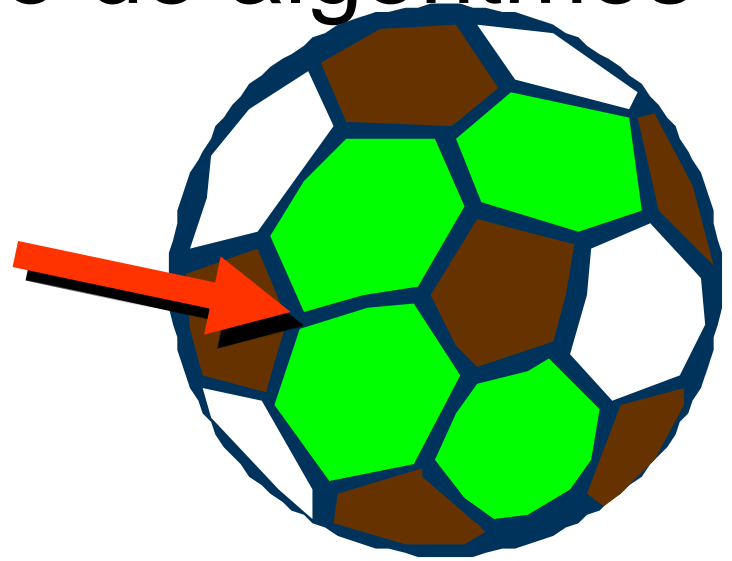


0.2. Análisis y diseño de algoritmos

- **Caso 3.** 2, 1, 1, 2, 1, 1,
1, 1, 2, 2, 2, 1, 1, 1, 1,
2, 1, 1

Resultado: 4 verdes

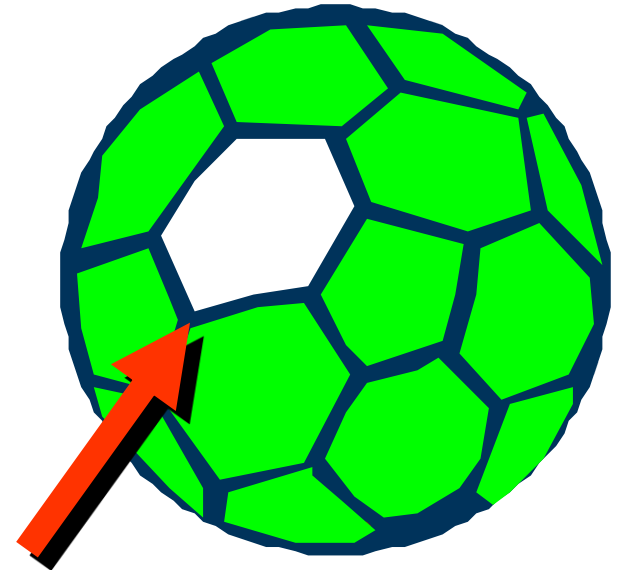
12 negros, 16 blancos



- **Caso 4.** 2, 2, 2, 2, 2, 2

Resultado: 31 verdes

0 negros, 1 blanco



0.3. Consejos para una buena programación

1. Proceso de análisis/diseño. No empezar tecleando código como locos.

2. Usar abstracciones, respetando los dos principios básicos:

- **Encapsulación:** las funciones relacionadas deben ir juntas (clases, módulos, paquetes, etc.).
- **Ocultación de la implementación:** Los aspectos de implementación no son visibles fuera del módulo, clase, etc.

0.3. Consejos para una buena programación

- 3. Reutilizar programas, librerías, tipos, etc. existentes.** Y programar pensando en la posible reutilización futura. Un nuevo programa no debe partir desde cero.
- 4. No resolver casos concretos, sino el problema en general.** Si no se requiere un esfuerzo adicional, el algoritmo debe resolver un caso genérico.
- 5. Repartir bien la funcionalidad.** Repartir la complejidad del problema de forma uniforme. No crear procedimientos muy largos: usar subrutinas. De esta forma se mejora la **legibilidad** del código.