



UNIVERSIDAD  
DE GRANADA



# Administración de Bases de Datos

## Grado en Ingeniería Informática

### Tema 1 – El Nivel Interno

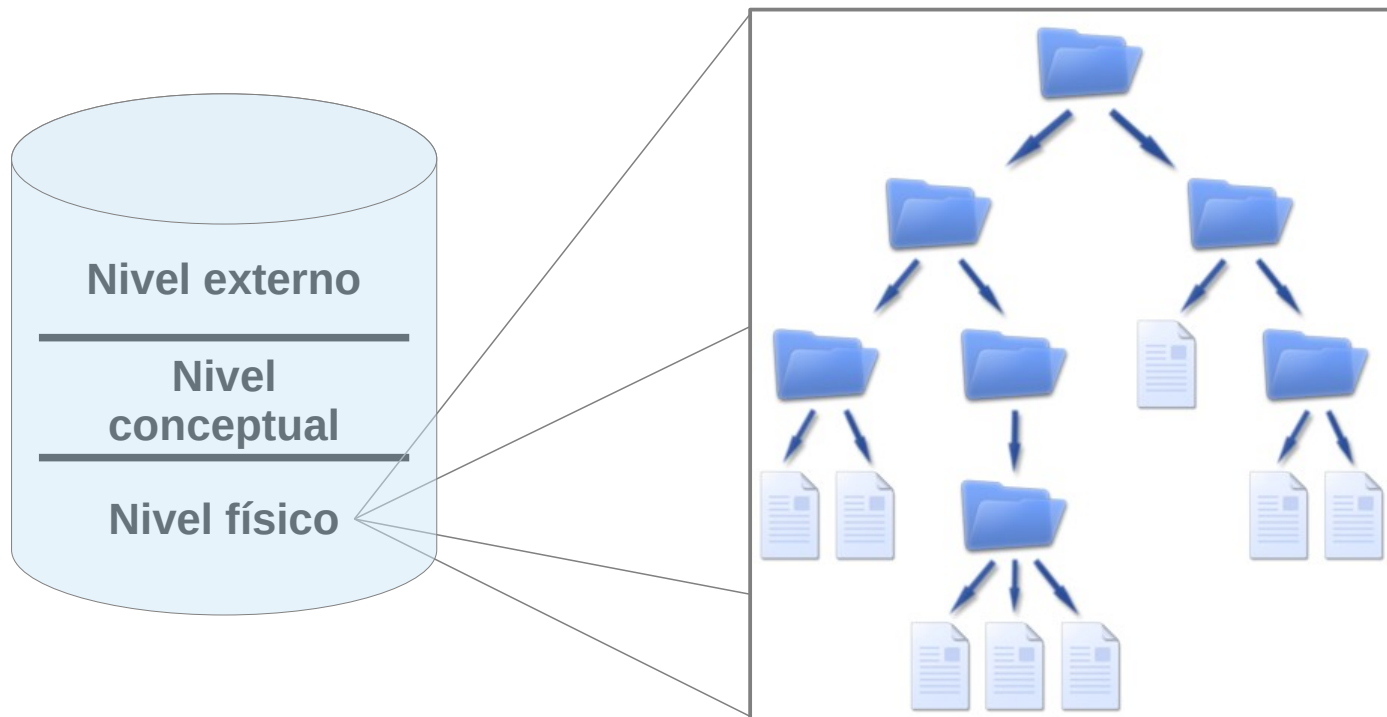


I. J. Blanco, A. G. López Herrera

Departamento de Ciencias de la  
Computación e Inteligencia Artificial  
<http://decsai.ugr.es>

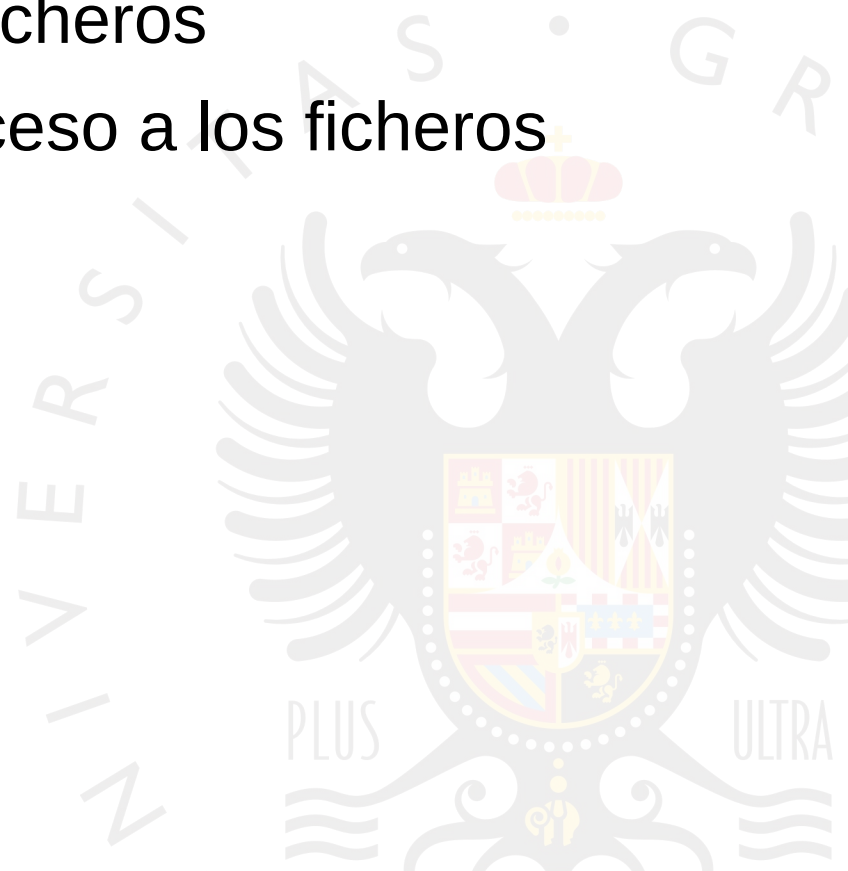
- Introducción al tema
- Medidas para evaluar un sistema de archivos
- Registros y bloques
- Organización de archivos y métodos de acceso
- Evaluación del sistema

- Eficiencia en grandes cantidades de datos:
  - forma de almacenamiento de los datos
  - forma de acceso rápido a los datos almacenados
  - arquitecturas para relacionar los datos



<http://commons.wikimedia.org/wiki/File:FilesAndFolders.png>

- Será necesario:
  - Comparar sistemas de ficheros
  - Comparar modos de acceso a los ficheros



Parámetro	Mide...
$R$	la memoria necesaria para almacenar un registro
$T$	el tiempo para encontrar un registro arbitrario
$T_F$	el tiempo para encontrar un registro por clave
$T_W$	el tiempo para escribir un registro cuando ya se tiene su posición
$T_N$	el tiempo para encontrar el siguiente registro a uno dado
$T_I$	el tiempo necesario para insertar un registro
$T_U$	el tiempo necesario para actualizar un registro
$T_X$	el tiempo necesario para leer el archivo
$T_Y$	el tiempo necesario para reorganizar el archivo

- recuperar un registro por valor de clave
- obtener el siguiente registro
- insertar registro (ampliar el fichero)
- actualización de registro
- leer todo el archivo
- reorganizar el archivo



- Un SGBD almacena la información en tablas
- La estructura de una tabla la determinan las columnas
- La información de una tabla se almacena en filas

¿Y cómo se almacena todo a nivel físico?

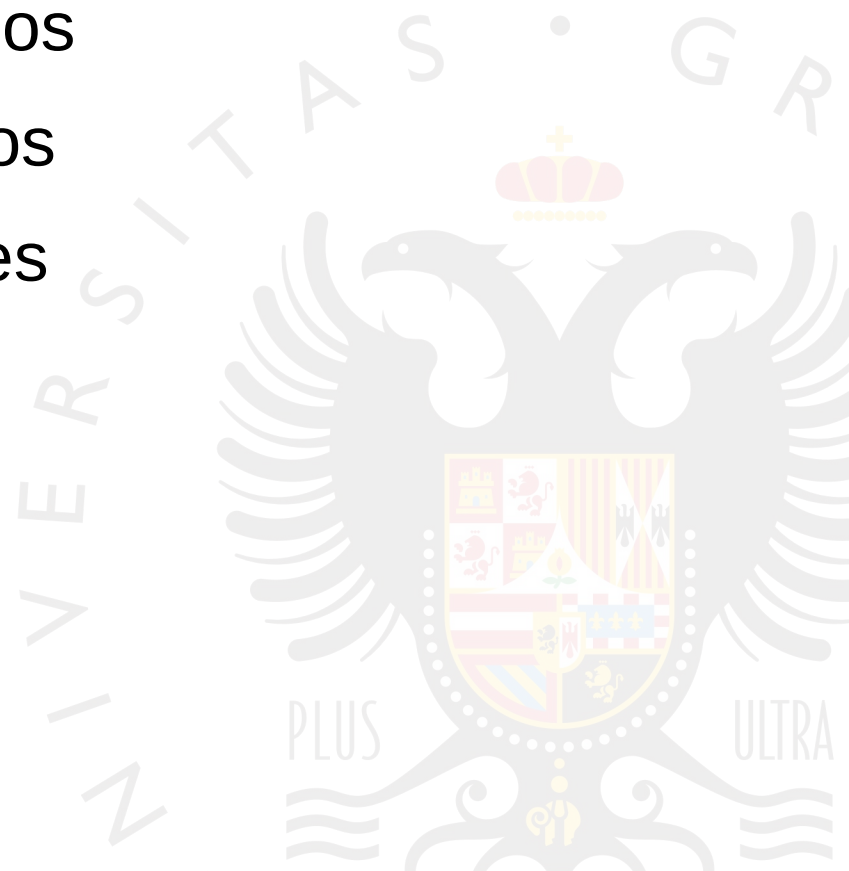


- Un SGBD provee de una serie de tipos de datos para las columnas de una tabla:
  - numéricos: enteros y reales,
  - cadenas de caracteres: CHAR, VARCHAR, VARCHAR2
  - fecha: DATE, TIME, TIMESTAMP
  - otros tipos: BLOBs, CLOBs, ...

Tamaño de un registro:

Tipo	Tamaño
CHAR(x)	x Bytes
VARCHAR2(X)	de 1 a x+1 Bytes
FLOAT	6 Bytes
BINARY_INTEGER	2 Bytes
...	...

- Campo: almacena un valor
- Registro: conjunto de campos
- Bloque: conjunto de registros
- Fichero: conjunto de bloques



- Registro de longitud fija

NRP	Nombre	Coddep	Salario
3477A	María Pérez	5	1527
5 B	30 B	2 B	6 B

- Registro de longitud variable

Factura	Linea	Concepto	Cant	Precio
325	1	Análisis#	1	300
2 B	2 B	9 B	2 B	6 B

- Estructura homogénea, distintos tamaños:

325	1	Análisis#	1	300
325	2	Tratamiento#	1	250
325	3	Chequeo#	1	300

- Estructura heterogénea, distintos tamaños:

NRP=3477A	Nombre=María	Coddep=	Salario=1527;
Factura=325;	Línea=1;	Concepto=Análisis;	Cant=1; Precio=300;

Registro de longitud fija:

- $V_i$ : longitud del valor del campo  $i$ -ésimo

$$R = \sum_i V_i$$

## Registro de longitud variable:

- A: longitud media de los nombres de atributo
- V: longitud media de los valores de atributo
- a': número medio de atributos
- s: número de separadores por atributo

$$R = a' (A + V + s)$$

## Registro de longitud variable:

Factura=325;	Linea=1;	Concepto= Análisis;	Cant=1;	Precio=300;
--------------	----------	---------------------	---------	-------------

$$A = (7 + 5 + 8 + 4 + 6) / 5 = 6$$

$$V = (2 + 2 + 8 + 2 + 6) / 5 = 4$$

$$s = 2$$

$$R = a' (6 + 4 + 2) = 12a'$$



- Unidad de información:
  - transferida por un dispositivo de almacenamiento masivo
  - almacenada en el área de trabajo de la memoria y denominada *buffer*
- Características:
  - Tamaño fijo para toda la DB
  - Múltiplo del bloque físico del S. O.

- La forma en la que se ajustan los registros a un bloque
- Un registro en disco tiene que pasar a un bloque de S. O. y después a un bloque de DB para ser tratado.
- El *factor de bloqueo*,  $Bfr$ , es el número de registros que caben en un bloque y depende del tamaño del mismo  $B$  y del tamaño de los registros  $R$ .

- Puede fijarse *a priori* por el administrador.
- Incluye una cabecera  $C$  con información útil al sistema (referencias, fecha de actualización, número de accesos simultáneos, etc.) que se resta a  $B$ .
- Hay dos métodos básicos:
  - bloqueo fijo o entero
  - bloqueo partido o encadenado

- Se rellena el bloque con tantos registros como sea posible.
  - Bloqueo entero con registros de longitud fija

$$B_{fr} = \left\lfloor \frac{B - C}{R} \right\rfloor$$

- Bloqueo entero con registros de longitud variable:
  - El siguiente registro cabe en el bloque si hay espacio.
  - Las marcas de separación de registros ocupan espacio:
    - Caracteres especiales
    - Distancia al comienzo del siguiente registro
    - Tabla de posiciones de los campos (con inicial y final)

$$Bfr = \left\lfloor \frac{B - C}{R + M} \right\rfloor$$

- Se escriben registros en un bloque hasta que no quede espacio.
- El último registro puede caber entero o partirse en dos partes en dos bloques distintos.
- Es necesaria una referencia del primer bloque al segundo.

- Bloqueo entero con registros de longitud variable:
  - El siguiente registro cabe en el bloque si hay espacio.
  - Las marcas de separación de registros ocupan espacio:
    - Caracteres especiales
    - Distancia al comienzo del siguiente registro
    - Tabla de posiciones de los campos (con inicial y final)

$$B_{fr} = \left\lfloor \frac{B - C}{R + M} \right\rfloor$$

- Problemas:
  - Búsqueda difícil de registros partidos
  - Actualización de ficheros completos complicada
- Ventajas:
  - No desperdicia espacio en los bloques.
  - Única solución cuando el tamaño del registro es mayor que el de un bloque.



- Bloqueo partido con registros de longitud variable:
  - Las marcas de separación de registros ocupan espacio:
    - Caracteres especiales
    - Distancia al comienzo del siguiente registro
    - Tabla de posiciones de los campos (con inicial y final)

$$B_{fr} = \left\lfloor \frac{B - P - C}{R + M} \right\rfloor$$

- El espacio que se pierde en marcas, referencias, espacio en el que no cabe un registro.
  - Un ejemplo para bloqueo partido con registros de longitud variable:

$$W = \frac{(P + Bfr \cdot M)}{Bfr} = \frac{P}{Bfr} + M$$

- El bloqueo fijo es más eficiente para registros pequeños.
- El bloqueo partido es más eficiente para registros grandes.

