UNIVERSIDAD DE GRANADA

DECSAI

Administración de Bases de Datos

Grado en Ingeniería Informática

Tema 1 – El Nivel Interno

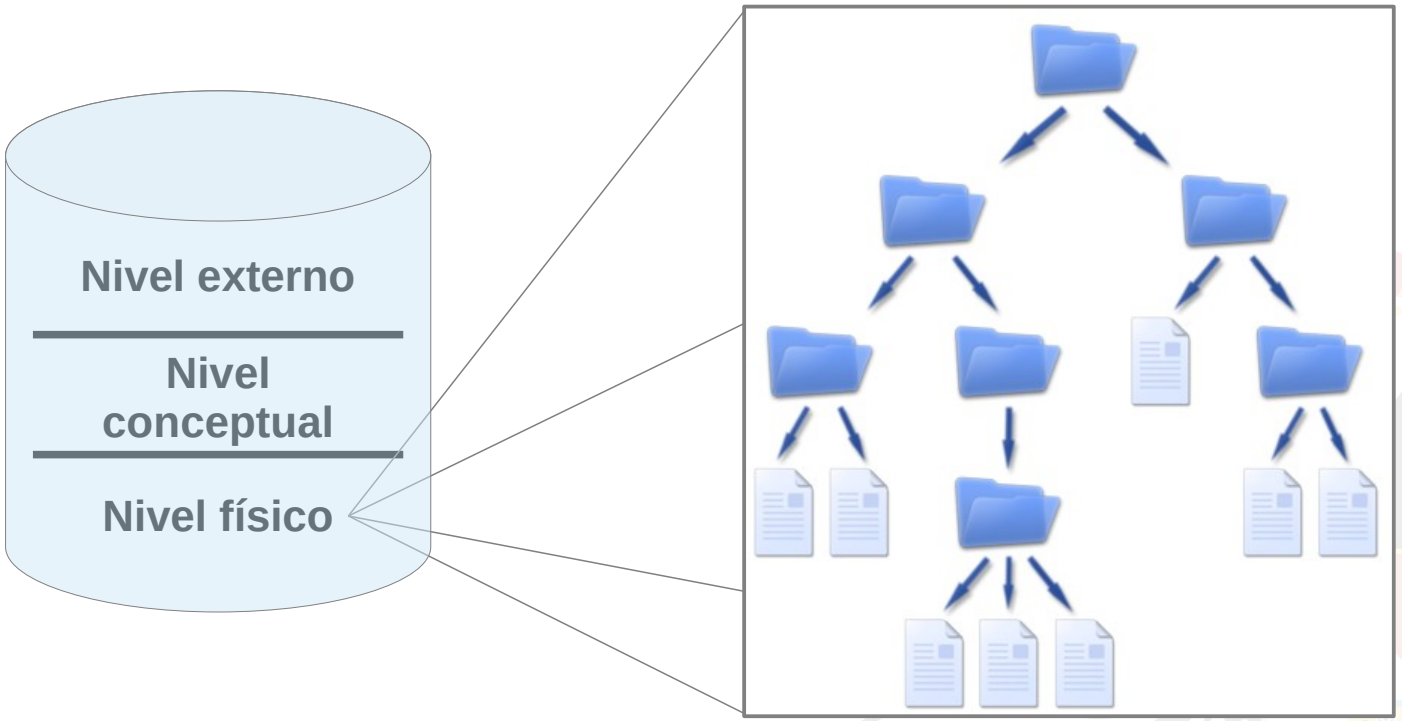
I. J. Blanco, A. G. López Herrera

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
...	...

I. J. Blanco, A. G. López Herrera

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

Registro de longitud fija:

$$R = \sum_i V_i$$

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

I. J. Blanco, A. G. López Herrera

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

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

I. J. Blanco, A. G. López Herrera

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

Registro de longitud variable:

$$A=(7+5+8+4+6)/5=6$$
$$V=(2+2+8+2+6)/5=4$$
$$s=2$$
$$R=a'(6+4+2)=12\text{a}'$$

- 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

I. J. Blanco, A. G. López Herrera

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

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

- 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.

I. J. Blanco, A. G. López Herrera

- 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)

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

I. J. Blanco, A. G. López Herrera

- 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)

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

I. J. Blanco, A. G. López Herrera

- 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.

- Principalmente, cuatro:
  - Archivo Secuencial Físico (ASF)
  - Archivo Secuencial Lógico (ASL)
  - Archivo Secuencial Indexado (ASI)
  - Archivo de Acceso Directo (AAD)
- Los demás, son combinaciones de estos.

I. J. Blanco, A. G. López Herrera

- Hipótesis:
  - Registros de estructura variable (y longitud variable)
  - Campo: (Attr_id, valor)
  - Dos separadores: entre id y valor, y entre campos

NRP=3477A;	Nombre=María Pérez;	Coddep=5;	Salario=1527;

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

$$R = a' \cdot (A + V + 2)$$

I. J. Blanco, A. G. López Herrera

$$T_F = \sum_{i=1}^{n} \frac{i}{n} \cdot T = \frac{n+1}{2} \cdot T \approx \frac{n}{2} \cdot T$$

I. J. Blanco, A. G. López Herrera

$$T_N = T_F$$

I. J. Blanco, A. G. López Herrera

$$T_I = T_W$$

I. J. Blanco, A. G. López Herrera

- Si el tamaño de registro no cambia:

$$T_A = T_F + T_W$$

- Si el tamaño del registro cambia:

$$T_A = T_F + T_W + T_I$$

I. J. Blanco, A. G. López Herrera

- Independientemente del contenido de registro:

$$T_X = n \cdot T$$

- Lectura ordenada según el valor de un atributo:

$$T_X = n \cdot T_F$$

- *O*: número de registros añadidos
- *d*: registros marcados para borrar

$$T_Y = (n+O)\cdot T + (n+O-d)\cdot T_W$$

I. J. Blanco, A. G. López Herrera

- Registros ordenados según una *clave física:* uno o varios campos
- Registros de longitud fíja (mismos campos y en el mismo orden)
- Estructura de registro en la cabecera de fichero
- Muy útiles para hacer *merge*

- Abrir hueco para nuevos registros es costoso
- Usan una zona de desbordamiento (no ordenada) de tipo ASF
- Es necesario reconstruir el fichero cuando la zona de desbordamiento crece

$$R = \sum_i V_i$$

- La cabecera no supone mucho.
- Si hay muchos valores nulos, el fichero se vuelve no denso.

I. J. Blanco, A. G. López Herrera

- El valor de búsqueda no es clave:

$$T_F = \frac{n}{2} \cdot T$$

- y hay registros en la zona de desbordamiento:

$$T_F = \frac{n}{2} \cdot T + \frac{O}{2} \cdot T$$

- El siguiente valor de clave está en el propio fichero:

$$T_N = T$$

- El siguiente valor de clave está en el fichero de desbordamiento:

$$T_N = T + O \cdot T$$

$$T_F \approx \log_2(n) \cdot T$$

- El valor de búsqueda es clave:

I. J. Blanco, A. G. López Herrera

- ¡Orden!
- Procedimiento:
  - localiza el punto de inserción
  - escribir el registro
  - leer y escribir los restantes

$$T_I = T_F + \frac{n}{2} \cdot T + \frac{n}{2} \cdot T_W [+ T_W]$$

I. J. Blanco, A. G. López Herrera

- Si hay varios registros para insertar y hay fichero de desbordamiento, podemos insertar todos los nuevos registros en él.

$$T_I = T_W$$

- pero después hay que insertarlos en el fichero *maestro*:

$$T_I = \frac{T_Y}{O}$$

I. J. Blanco, A. G. López Herrera

- Si no se cambia el valor de la clave:

$$T_U = T_F + T_W$$

- Si se cambia el valor de la clave:

$$T_U = T_F + T_W + T_I$$

I. J. Blanco, A. G. López Herrera

- Sólo se usa archivo maestro:

$$T_X = n \cdot T$$

- Se usa desbordamiento (hay que ordenarlo) y leer en *merge*

$$T_X = T_C(O) + (n+O) \cdot T$$

I. J. Blanco, A. G. López Herrera

- Sólo si hay desbordamiento.
- Procedimiento:
  - re-ordenar el fichero de desbordamiento
  - mezclar ambos ficheros (maestro y desbordamiento) en un nuevo fichero

$$T_Y = T_C(O) + (n+O)\cdot T + (n+O-d)\cdot T_W$$

I. J. Blanco, A. G. López Herrera

- Acelera el acceso por clave
- Se busca el valor de clave en el *índice* y éste irá acompañado de una posición en el fichero.
- Si la posición es de registro, se habla de *índice denso*.
- Si la posición es de bloque, se habla de *índice no denso*.

- Estructura:
  - Un fichero de datos: ASL con una posible área de desbordamiento (que no se gestiona igual que en el ASL)
  - Un fichero de índice: ASL con registros de longitud fija (y estructura uniforme) que contiene valor de clave y dirección del fichero de datos

- El número de registros del fichero de *índice* coincide con el número de registros del fichero *maestro*.
- Puede haber otros índices sobre el mismo fichero *maestro*, llamados *secundarios*.

$$R = \sum_i V_i + (V_k + P)$$

$$T_F = \log_2(n) \cdot T + T$$

$$T_N = T + T = 2 \cdot T$$

I. J. Blanco, A. G. López Herrera

- Si el fichero maestro no tiene área de desbordamiento:
  - Insertar registro de datos en fichero maestro T_{I1}
  - Insertar registro de índice en fichero de índice T_{I2} $T_{I1} = T_{I2} = T_F + \frac{n}{2} \cdot T + \frac{n}{2} \cdot T_W [+T_W]$

$$TI = T_{I1} + T_{I2} = 2 \cdot \left(T_F + \frac{n}{2} \cdot T + \frac{n}{2} \cdot T_W [+T_W] \right)$$

I. J. Blanco, A. G. López Herrera

- Si el fichero maestro tiene área de desbordamiento:
  - Insertar registro de datos en el área de desbordamiento del fichero maestro
  - Insertar registro de índice en fichero de índice T_{I2} $T_{I2} = T_F + \frac{n}{2} \cdot T + \frac{n}{2} \cdot T_W [+ T_W]$

$$TI = T_W + T_{I2}$$

I. J. Blanco, A. G. López Herrera

- Si no se cambia el valor de la clave:

$$T_U = T_F + T_W$$

- Si se cambia el valor de la clave:

$$T_U = 2 \cdot (T_F + T_W) + T_{I1} + T_{I2}$$

I. J. Blanco, A. G. López Herrera

- Por índice principal:

$$T_X = n \cdot T$$

- Por índice secundario:

$$T_X = n \cdot T + n \cdot T = 2 \cdot n \cdot T$$

- Por índice secundario y desbordamiento:

$$T_X = (n+O) \cdot T + n \cdot T + O \cdot T = 2 \cdot (n+O) \cdot T$$

I. J. Blanco, A. G. López Herrera

- Después de muchos borrados.
- Procedimiento:
  - re-ordenar el fichero de desbordamiento
  - mezclar ambos ficheros (maestro y desbordamiento) en un nuevo fichero
  - crear el índice de nuevo

$$T_Y = T_C(O) + (n+O)\cdot T + (n+O-d)\cdot T_W + (n+O-d)\cdot T_W$$

I. J. Blanco, A. G. López Herrera

- Índices densos demasiado grandes para memoria.
- Posición apunta a *página.*
- El fichero de índice contiene tantas entradas como páginas contiene el fichero de datos.
- Una clave dada está en una página (es mayor o igual que la clave de esa página y menor que la de la siguiente)

$$\frac{n}{Bfr}$$

I. J. Blanco, A. G. López Herrera

- Procedimiento de búsqueda en no denso:
  - Se selecciona en el índice la clave inmediatamente inferior a la buscada
  - Se carga el bloque del maestro
  - Se busca secuencialmente
- Si no está, hay que acceder al maestro. En un denso no es necesario.

- En cada página del maestro se suele dejar un espacio para nuevos registros.
- En maestros con fichero de desbordamiento, cada registro tiene un enlace al siguiente en el fichero de desbordamiento. En una página, hay un enlace a registro de desbordamiento (si la página está completa). En el fichero de desbordamiento, un registro puede tener enlace al siguiente.

I. J. Blanco, A. G. López Herrera

- Efectivos cuando los archivos son grandes y/o los registros son grandes.
- Si el índice es grande, puede indexarse, a su vez. Si el índice del índice es grande... A esto se le llama *índice multinivel*.
- Tamaño de *página del índice*

$$y = \left\lfloor \frac{B - C}{V + P} \right\rfloor$$

- Si es de *m* niveles, todos los niveles son no densos.
- La estructura es de árbol. Al primer nivel se le llama *raíz*.
- El nivel 1 es el más cercano al fichero maestro.

- En general, el fichero maestro tiene desbordamiento:

$$r = P + \sum_i V_i$$

- El índice de primer nivel tiene una entrada por bloque del maestro, luego el número de entradas es:

$$i_1 = \left\lceil \frac{n}{Bfr} \right\rceil$$

I. J. Blanco, A. G. López Herrera

- Los niveles superiores tienen su propio *B* y su propio *y, con lo que el número de entradas será*:

$$i_k = \left\lceil \frac{i_{k-1}}{y} \right\rceil$$

- y el número de bloques:

$$b_k = \left\lceil \frac{i_k}{y} \right\rceil = i_{k+1}$$

- Espacio necesario para índices:

$$I = (b_1 + b_2 + \ldots + b_m) \cdot B$$

- y el espacio medio necesario por registro:

$$R = r + \frac{O}{n} \cdot r + \frac{I}{n}$$

I. J. Blanco, A. G. López Herrera

$$T_F = T_M + (m-1) \cdot T_{F_i} + T'_F$$

$$T_F = T_M + (m-1) \cdot T_{F_i} + T'_F + T_{F\,Cadena}$$

$$T_N \approx T$$

- Se tiene que insertar en el fichero maestro y queda espacio en el bloque:

$$T_I \approx T_F + \frac{1}{2} \cdot Bfr \cdot \left(T + T_W\right) + T_W$$

- Se tiene que insertar en el fichero maestro y no queda espacio en el bloque:

$$T_I \approx T_F + \frac{1}{2} \cdot Bfr \cdot \left(T + T_W\right) + 2 \cdot T_W$$

- Se tiene que insertar en el fichero de desbordamiento:

$$T_I \approx T_F + 2 \cdot T_W$$

- Si no cambia el valor de la clave:

$$T_U = T_F + T_W$$

- Si cambia el valor de la clave:

$$T_U = T_F + T_W + T_I$$

$$T_X \approx (n+O) \cdot T$$

$$T_Y = T_X + (n + O - d) \cdot T_W + k \cdot I$$

I. J. Blanco, A. G. López Herrera

- Sólo son útiles si hay consultas que los utilicen.
- Los atributos candidatos son los involucrados en la cláusula WHERE (cuando se ven afectados por la igualdad o el uso de rangos) o aquellos atributos de reunión.
- Si el número de tuplas con valores repetidos para un campo supera el 5%, no debería crearse un índice sobre ese campo.

- Los campos de un índice multiatributo deben ordenarse según las consultas más frecuentes.
- No se deben indexar campos actualizados frecuentemente.

- Aprovecha el acceso directo de disco
- Mediante la clave se obtiene la posición.
- Prevé nuevas entradas dejando espacio libre.
- Hay varios métodos de transformación.

- Supón un campo clave de tamaño V (si es numérico, hay 10^V valores posibles; si es alfabético, hay 26^V valores posibles).
- El fichero tendrán *m* celdas para *n* datos (incluyendo espacios libres) luego $m>=n$.
- Si una clave tiene *b* símbolos distintos, entonces hay b^V posibles claves, que cumplirá $b^V>>m$ y el algoritmo transforma cada clave en un valor entre *0* y *m-1*

- Puesto que $b^v >> m$:
  - Se dan **colisiones**: dos claves con la misma posición. Soluciones:
    - Desbordamiento
    - Direcciones de página
  - Hay **huecos**: ninguna clave da una posicion. Solución:
    - Usar para resolver colisiones

I. J. Blanco, A. G. López Herrera

- Direccionamiento cerrado: el espacio de posiciones en un único archivo.
  - **Busqueda lineal**: una colisión se almacena en una posición libre del mismo bloque (secuencial en búsqueda de registro)
  - **Realeatorización**: una colisión se almacena en otra posición mediante otra transformación de clave, y sucesivamente.

- Direccionamiento abierto: el espacio de posiciones en más de un fichero (principal y desbordamiento).
  - **Listas enlazadas**: colisiones en fichero de desbordamiento (estructura ASI)
  - **Bloques de desbordamiento**: colisiones en bloques del fichero de desbordamiento

- Hashing dinámico: el espacio de posiciones y la transformación se adaptan dinámicamente.

$$r = P + \sum_i V_i$$

$$S_F = m \cdot r + c \cdot r = (m + c) \cdot r$$

$$R = \frac{S_F}{n} = \frac{(m + c)}{n} \cdot r$$

$$T_F = C + T + p \cdot T_{F\_Cadena}$$

I. J. Blanco, A. G. López Herrera

$$T_I = C + 2 \cdot T_W$$

$$T_U = T_F + T_W$$

$$T_X \simeq (m+c) \cdot T$$

I. J. Blanco, A. G. López Herrera

$$T_Y = T_X + T_{Carga}$$

$$T_{Carga} = \sum_{i=1}^{n} T_I(i)$$

- Muchos de estos parámetros pueden estimarse, pero mejor estimar que ir a ciegas.

- Carga por demandas de almacenamiento:
  - Se ve afectada por: número de registros, número total de atributos, número medio de atributos por registro, longitud de campos, longitud de identificadores de atributos.
  - Carga por recuperación de información: se calcula teniendo en cuenta el número de solicitudes a un archivo en un conjunto de transacciones en un tiempo dado.

- Carga por demandas de almacenamiento:
  - Carga por actualización: se calcula teniendo en cuenta el número de actualizaciones a la base de datos en base a la frecuencia de creación de registro, de actualización de registro, de eliminación y de ampliación.

I. J. Blanco, A. G. López Herrera

- Basado en la probabilidad de las operaciones.
- Estimación de beneficios:
  - Generalmente basados en factores económicos.
  - Se tienen en cuenta los gastos de personal.
  - Tiempos de respuesta bajos centran al personal. Se calculan en base a:
    - tiempo de solicitud
    - tiempo de computación
    - tiempo de presentación de resultados

- En términos de uso del sistema:
  - En general:
    - tiempo medio para plantear una consulta
    - tiempo medio para describir una salida de datos
    - tiempo medio de procesamiento de operación
    - retraso medio en presentación de resultados
  - Hay que tener en cuenta el tipo de usuarios, porque incluye un costo adicional en el manejo

- Mantenimiento de los datos
  - El mayor coste está asociado a la actualización.
  - Eso conlleva un coste de detección y corrección de errores:
    - antes de actualización: difícil y costoso
    - durante la actualización: puede llevar a acciones no deseadas e incrementos de costes en el sistema
    - después de actualización: hay que rastrear el alcance y es mejor no cambiarlo

- Ante cambios frecuentes, será necesario prever más espacio reservado.
- La densidad mide la proporción de espacio ocupado (y depende del tipo de archivo)

$$D = \sum_{F_i} n_i \cdot \frac{R+W}{\mu_{D_i}}$$

I. J. Blanco, A. G. López Herrera

- Componentes:
  - discos,
  - controladoras,
  - canales, …
  - En general, mejor una sola unidad mayor que varias más pequeñas

I. J. Blanco, A. G. López Herrera

- Distribución:
  - varios procesadores y un dispositivo de almacenamiento

- Para parámetros conocidos de carga y efectividad de archivos, podemos escoger.
- Primer enfoque: modelar cada diseño físico y seleccionar.
  - Es costoso

- Segundo enfoque: reducir posibilidades. Primero, el hardware adecuado. Después el diseño físico para el uso estimado dados los recursos existentes.
  - estimar la demanda
  - estimar recursos
  - comparar para encontrar una combinación satisfactoria
  - si no acoplan, ambas estimaciones se ajustan de forma más precisa
  - si la discordancia es importante, cambiar el equipo o el diseño físico y empezar de nuevo

- Procesos compartidos: existe competencia
  - A nivel de procesador: afecta al tiempo de respuesta
  - A nivel de disco: provoca recolocación constante del cabezal, lo que afecta al registro siguiente y a la lectura exhaustiva
  - La *programación concurrente* permite bloquear un dispositivo hasta que una tarea concluya o ésta lo libere.

- Beneficios:
  - cantidad de datos disponibles
  - intensidad de uso de la BD
- Costes:
  - almacenamiento (inicial + ampliaciones)
  - procesamiento (escalabilidad de discos, controladoras o canales)