

Tema 2: Organización básica

1. **Introducción:** Objetivos de diseño y factores que influyen en la organización
2. **Organización básica:** Organización Serial
3. **Organización ordenada:** Secuencial
4. **Mantenimiento:** *reorganización y gestión de huecos*



@LABDA. Univ. Carlos III

Tema 2.1: Introducción

Objetivos en el Diseño Físico de Ficheros

- Tiempo de acceso
- Volumen y Ocupación

Factores que influyen en la organización del fichero

- Tipo de los procesos y Actividad (reg. procesados) y
- Volatilidad (cambios por unidad de tpo.)
- Crecimiento



@LABDA. Univ. Carlos III

Tema 2.1: Objetivos de diseño

Cálculo del tiempo de acceso a registro físico (bloque):

$$t_{\text{bloque}} = t_{\text{localización}} + t_{\text{transferencia}}$$

- El tiempo de localización será el número de localizaciones por el tiempo empleado en cada una
- El tiempo de transferencia será el número de bloques transferidos por el tiempo empleado leer un bloque

Cálculo del tiempo de acceso a registro aleatorio:

$$t_{\text{aleatorio}} = n^{\circ} \text{ accesos} \cdot t_{\text{bloque}}$$



@LABDA, Univ. Carlos III

Tema 2.1: Objetivos de Diseño - Coste Global

Frecuencia Relativa de un Proceso:

Todo fichero estará sometido a un cjto. de procesos $P \equiv \{P_1..P_n\}$

Cada proceso tendrá una frecuencia f_i asociada, referida a...

- a una unidad de tiempo (por ejemplo, segundos u horas)
- frecuencia relativa: al conjunto P de todos los procesos, tal que $\sum_{i=1..n} f_i = 1$

Coste Global (referido a una organización del fichero):

- Una organización física de un fichero (O_k) define su organización base y las organizaciones auxiliares con las que cuenta.
- Cada proceso P_i tendrá en O_k un coste C_i asociado (y expresado en nº accesos o en tiempo), que dependerá directamente de la organización.
- El coste global de la organización se define como: $C(O_k, P) = \sum_{i=1..n} C_i \cdot f_i$
(puede venir referida a una unidad de tiempo)



@LABDA, Univ. Carlos III

Tema 2.1: Objetivos de diseño

Volumen y Ocupación de un fichero

- **volumen**: número de bytes que contiene
- **ocupación útil**: bytes útiles del fichero

Se cumple que: $\text{volumen} \geq \text{ocupación útil}$

$$\text{densidad (\%)} = \frac{\text{ocupación útil}}{\text{volumen real}} \cdot 100$$

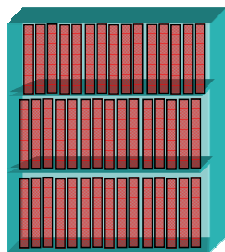
$$\text{mejora (\%)} = \frac{\text{densidad}_{\text{final}}}{\text{densidad}_{\text{inicial}}} \cdot 100$$



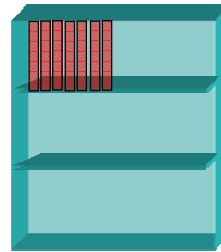
@LABDA, Univ. Carlos III

Tema 2.1: Objetivos de diseño

$$\text{densidad (\%)} = \frac{\text{ocupación útil}}{\text{volumen real}} \cdot 100$$



15 libros / estantería



¿cuál es la densidad?



@LABDA, Univ. Carlos III

Tema 2.1: Objetivos de diseño

Factores referidos a un dispositivo

Volumen y ocupación referidos a un fichero sobre un dispositivo cambian sensiblemente porque hay que tener en cuenta el bloque.

volumen fichero = bloques ocupados · tamaño bloque
siendo n° registros totales \leq bloques ocupados · factor de bloqueo

$$\text{densidad (\%)} = \frac{\text{Tamaño medio reg.} \cdot n^{\circ} \text{ registros lógicos}}{\text{Tamaño reg. físico} \cdot n^{\circ} \text{ reg. físicos}} \cdot 100$$

soporte ideal: el volumen del reg. físico coincide con el del reg. lógico y por lo tanto el número de registros es el mismo.



@LABDA. Univ. Carlos III

Tema 2.1: Tipos de diseño

Diseño Físico:

“Determinación de la organización física de un fichero”

Organización física: disposición de los registros en el soporte, relativa a su implementación, orden, direccionamiento, apuntamientos, etc.

Diseño Lógico:

“Descripción de la estructura lógica de los registros de un fichero”

Estructura lógica (de un registro): descripción y disposición de los elementos de un registro, que en conjunto definen un individuo.

Diseño Físico del registro Lógico:

“Implementación de un registro lógico en secuencias de bytes”

Diseño Físico-Lógico (registro): descripción de las cadenas de bytes utilizadas para almacenar registros, y de los convenios necesarios para su interpretación.



@LABDA. Univ. Carlos III

Tema 2.1: Ejemplo

A) Calcular la densidad ideal y la densidad de fichero para un fichero con RLF con 1.000 registros en un soporte cuyo tamaño de bloque es 1Kb.

Campo	Compuesto	Tipo	Frecuencia %	Veces max.	Veces. media	Tamaño max.	Tamaño medio
Título		C	100			42	30
Año		N	100			4	4
Productora		C	100			20	9
Director			100				
	Nombre	C	100			20	12,5
	Apellido	C	100			20	12,5
Actores		C	75	5	2,8		
	Nombre	C	100			20	12,5
	Apellido	C	100			20	12,5
BS			50				
	Nombre	C	100			20	12,5
	Apellido	C	100			20	12,5
	Formato	C	100	3	3	3	3



@LABDA. Univ. Carlos III

Tema 2.1: Ejemplo

Ejemplo: densidad del fichero de películas en DVD

1º Partimos del diseño lógico

Título C(42)

Año N(4)

Productora C(20)

Género C(9)

Director (nombre C(20); apellido C(20))

Actores (nombre C(20); apellido C(20))*

Banda Sonora

[Autor B.S. (nombre C(20); apellido C(20));

Formato C(3)+]



@LABDA. Univ. Carlos III

Tema 2.1: Ejemplo

Ejemplo: densidad del fichero de películas en DVD

2º Realizamos el diseño físico – lógico para registros de longitud fija

Título	b(42)
Año	b(4)
Productora	b(20)
Género	b(9)
Director	(nombre b(20), apellido b(20))
Actor/iz	(nombre b(20), apellido b(20)) ⁵
BS	((nombre b(20), apellido b(20))
Formato	(b(3)) ³

¿cómo mejorar la densidad?

$$\text{densidad (\%)} = \frac{\text{ocupación útil}}{\text{volumen real}} \cdot 100$$

iReduciendo el volumen!



@LABDA, Univ. Carlos III

Ejercicio de diseño lógico

Ejemplo 2: Diseño lógico y físico-lógico del registro “estudiante ITIG” para RLF y RLV. Densidad ideal y de fichero sabiendo que tenemos 1500 registros y el $T_{bq} = 1\text{KB}$ (fichero no-consecutivo).

- Nombre completo (20 nombre + 20 apellidos)
- Teléfonos (+34 y número) si los tiene, como máximo 3
- Dirección (50 como máximo)
- Asignaturas aprobadas. Nombre de la asignatura (50 carac. como máximo), curso (1..3) y convocatoria (1..6). En total se sabe que un alumno como máximo tendrá 40 asignaturas aprobadas
- Si ha cursado las 40 asignaturas, también necesitamos saber si ha realizado el proyecto y si es así el título (70 carac.) y el año en que lo presentó.



@LABDA, Univ. Carlos III

Ejercicio de diseño lógico

Campo	Compuesto	Tipo	Frecuencia %	Veces max.	Veces. media	Tamaño max.	Tamaño medio
Nombre-C			100				
	Nombre	C				20	10
	Apellidos	C				20	15
Telefonos			80	3	2		
	Prefijo	C				3	3
	Número	N				9	9
Dirección		C	100			50	40
Asig-Apro			90	40	20		
	Nombre	C				50	30
	Curso	N				1	1
	Convocatoria	N				1	1
Proyecto			20				
	Título	C				70	50
	Año	N				4	4



@LABDA. Univ. Carlos III

Ejercicio de diseño lógico

• Diseño lógico

Nombre-C (nombre C(20), apellidos C(20))
 Telefonos (prefijo C(3), número N(9))*
 Dirección C(50)
 Asig-Apro (nombre C(50),
 curso N(1),
 convocatoria N(1))*
 Proyecto [(título C(70); año N(4))]

• Diseño físico-lógico para RLF

Nombre-C (nombre B(20), apellidos B(20))
 Telefonos (prefijo "+34", número B(9))³
 Dirección B(50)
 Asig-Apro (nombre B(50),
 curso B(1),
 convocatoria B(1))⁴⁰
 Proyecto (título B(70); año B(4))

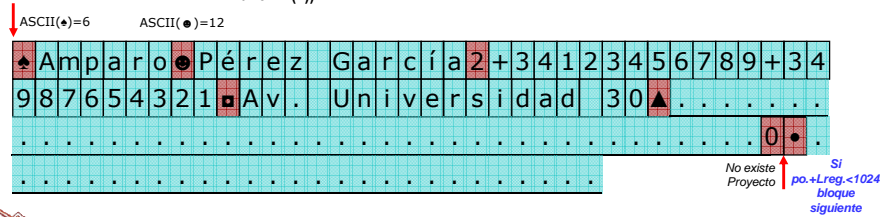


@LABDA. Univ. Carlos III

Ejercicio de diseño lógico

Diseño físico-lógico para RLV

Nombre-C	(longitud.nombre B(1), nombre B(longitud.nombre), longitud.apellidos B(1), apellidos B(longitud.apellidos))
Existe-Telefonos	B(1)
Telefonos	(prefijo "+34", número B(9)) ^{Existe-Telefonos}
Longitud.dirección	B(1)
Dirección	B(longitud.dirección)
N-asig-apro	B(1)
Asig-Apro	(longitud.nombre B(1), nombre B(longitud.nombre), curso B(1), convocatoria B(1)) ^{N-asig-apro}
Existe-proy	B(1)
Proyecto	(longitud.titulo B(1), título B(longitud.titulo), año B(4))



@ LABDA. Univ. Carlos III

Ejercicio de diseño lógico

Densidad ideal para RLF

$$D. \text{ Ideal (\%)} = \frac{\text{Ocupación Reg.}}{\text{Volumen Reg.}} = \frac{671}{2280} = 29,4\%$$

Campo	Compuesto	Veces max.	Tamaño max.	Vol. campo
Nombre-C				
	Nombre		20	20
	Apellidos		20	20
Telefonos		3		
	Prefijo		3	9
	Número		9	27
Dirección			50	50
Asig-Apro		40		
	Nombre		50	2000
	Curso		1	40
	Convocatoria		1	40
Proyecto				
	Título		70	70
	Año		4	4
Volumen Registro				2280

Campo	Compuesto	Frecuencia %	Veces media	Tamaño medio	Ocup. campo
Nombre-C		100			
	Nombre			10	10
	Apellidos			15	15
Telefonos		80	2		
	Prefijo			3	4,8
	Número			9	14,4
Dirección		100		40	40
Asig-Apro		90	20		
	Nombre			30	540
	Curso			1	18
	Convocatoria			1	18
Proyecto		20			
	Título			50	10
	Año			4	0,8
Ocupación Registro					671

@ LABDA. Univ. Carlos III

Ejercicio de diseño lógico

- Densidad del Fichero

$T_{bq} = 1 \text{ KB} = 1024 \text{ B}$
Número de Reg. = 1500

Tratamos con registros no-consecutivos, es decir, *factor de bloqueo* es entero.

$f = (1024/\text{volumen Reg.}) = (1024/2280) = 0,45$
= 3 (cada reg. lógico cabe 3 físicos) registros expandidos

Densidad = $(671*1500)/(1500*3*1024) = 21,8\%$



@LABDA. Univ. Carlos III

Tema 2.1: Optimización

¿Cómo podemos mejorar el volumen de un fichero?

Utilizando registros de longitud variable con **dos mecanismos**:

- Campos de control**: mejoran el manejo (ahorrando espacio)
- Codificación de campos**: codifica el valor de un campo.



@LABDA. Univ. Carlos III

Tema 2.1: Optimización

Elemento de Datos

Campos de control: mejoran el manejo (ahorrando espacio)

- *Existencia*: en campos opcionales, indica si aparece o no
- *Longitud*: en campos variables, indica el número de caracteres
- *Reiteración*: en grupos repetitivos, indica el número de ocurrencias
- *Fin de Campo*: en campos indefinidos, indican que acaba

Registro

- *Fin (inicio) de Registro*: permite registros consecutivos en el soporte
- *Tipo*: indica el tipo de registro a continuación (f. heterogéneos)
- *Mapa*: indica los registros que se aplican (f. heterogéneos)



@LABDA, Univ. Carlos III

Tema 2.1: Optimización

Codificación de campos: codificar el contenido del campo

- Si se hace, debe ser completamente transparente a usuario.
- *La compresión disminuye el tamaño, pero la densidad no es alterada significativamente*

Algunos Tipos de Codificación:

1. *Codificación numérica*: no almacenar dígitos, representar números
Ejemplo: DNI C(9) → N(4)
2. *Enumerados*: equivalen a una codificación numérica natural
(se puede establecer una codificación de ese tipo)

Ejemplo: color ∈ {blanco, amarillo, naranja, rojo, verde, azul, negro}
color C(8) → N(1)



@LABDA, Univ. Carlos III

Tema 2.1: Optimización

Codificación numérica:

- para un rango de números naturales [0..m]

¿cuántos bytes son necesarios para codificar el número?

- se halla la longitud (nº de dígitos) del número codificado en binario, $l = \log_2(m)$ donde:
 - m es el elemento mayor del dominio
- se halla el número de bytes $c = l / 8$

m	RESIDUO(m;2)	Peso	Decimal
2007	1	2 ⁰	1
1003	1	2 ¹	2
501	1	2 ²	4
250	0	2 ³	0
125	1	2 ⁴	16
62	0	2 ⁵	0
31	1	2 ⁶	64
15	1	2 ⁷	128
7	1	2 ⁸	256
3	1	2 ⁹	512
1	1	2 ¹⁰	1024
SUM			2007

- para un rango de números enteros [n..m] (m positivo)

representación en exceso a n: para almacenar x, tomamos x-n
el nuevo rango es [0..(m-n)] y procedemos como en el 1º caso

- para un rango de números enteros negativo [n..m] (n y m negativos)

dado que el signo es un factor k constante no se almacena (tomamos x / k)
el nuevo rango es positivo, y se procede como en los casos anteriores



@LABDA. Univ. Carlos III

Tema 2.1: Problema

B) Diseño físico-lógico con compresión de datos.

Calcular la densidad ideal y la densidad de fichero para un fichero con RLV con 1.000 regs. en un soporte cuyo tamaño de bloque es 1Kb. *
Incluir Género C(9) con tres valores posibles {terror, drama, comedia}

Campo	Compuesto	Tipo	Frecuencia %	Veces max.	Veces. media	Tamaño max.	Tamaño medio
Título		C	100			42	30
Año		N	100			4	4
Productora		C	100			20	9
Género		C	100			9	6
Director			100				
	Nombre	C	100			20	12,5
	Apellido	C	100			20	12,5
Actores		C	75	5	2,8		
	Nombre	C	100			20	12,5
	Apellido	C	100			20	12,5
BS			50				
	Nombre	C	100			20	12,5
	Apellido	C	100			20	12,5
	Formato	C	100	3	2	3	3



@LABDA. Univ. Carlos III

Tema 2.1: Ejemplo

Ejemplo: densidad del fichero de películas en DVD

1º Partimos del diseño lógico

Título	C(42)
Año	N(4)
Productora	C(20)
Género	C(9)
Director	(nombre C(20); apellido C(20))
Actores	(nombre C(20); apellido C(20))*
Banda Sonora	
	[Autor B.S. (nombre C(20); apellido C(20));
Formato	C(3)+]



@LABDA, Univ. Carlos III

Tema 2.1: Ejemplo

Ejemplo: densidad del fichero de películas en DVD

2º Realizamos el diseño físico – lógico para registros de longitud variable.

longitud_titulo	B(1)	
Título	B(longitud_titulo)	
Año	B(4)	
longitud_productora	B(1)	
Productora	B(longitud_productora)	
Género	B(9)	
Director	(long_nombre B(1), nombre B(long_nombre), long_apellido B(1), apellido B(long_apellido))	
número_actores	B(1)	
Actores	(long_nombre B(1), nombre B(long_nombre), long_apellido B(1), apellido B(long_apellido)) ^{n_actores}	
Existe_B.S.	B(1)	
B.S.		
(Autor B.S.		
(long_nombre B(1), nombre B(long_nombre), long_apellido B(1), apellido B(long_apellido))		
n_formato	B(1)	
Formato	B(3) ^{n_formato} existe_B.S.	



@LABDA, Univ. Carlos III

Tema 2.1: Ejemplo

Ejemplo: densidad del fichero de películas en DVD

2º Realizamos el diseño físico – lógico para registros de longitud variable.

longitud_titulo	B(1)
Título	B(longitud_titulo)
Año	B(2)
longitud_productora	B(1)
Productora	B(longitud_productora)
Género	B(1)
Director	(long_nombre B(1), nombre B(long_nombre), long_apellido B(1), apellido B(long_apellido))
n_actores	B(1)
Actores	(long_nombre B(1), nombre B(long_nombre), long_apellido B(1), apellido B(long_apellido)) ^{n_actores}
Existe_B.S	B(1)
B.S	(Autor B.S. (long_nombre B(1), nombre B(long_nombre), long_apellido B(1), apellido B(long_apellido))
n_formato	B(1)
Formato	B(1) _{formato existe_B.S.}

Codificación Numérica
(¿cuántos bytes?)

Codificación Enumerados
(¿cuántos bytes?)



@ L.ABDA, Univ. Carlos III

Tema 2.1: Ejemplo

Ejemplo: densidad del fichero de películas en DVD

2º Realizamos el diseño físico – lógico para registros de longitud variable.

longitud_titulo	B(1)
Título	B(longitud_titulo)
Año	B(2)
longitud_productora	B(1)
Productora	B(longitud_productora)
Género	B(1)
Director	(long_nombre B(1), nombre B(long_nombre), long_apellido B(1), apellido B(long_apellido))
n_actores	B(1)
Actores	(long_nombre B(1), nombre B(long_nombre), long_apellido B(1), apellido B(long_apellido)) ^{n_actores}
Existe_B.S	B(1)
B.S	(Autor B.S. (long_nombre B(1), nombre B(long_nombre), long_apellido B(1), apellido B(long_apellido))
Formato	B(1) _{formato existe_B.S.}

Codificación Mapa
(¿cuántos bytes?)



@ L.ABDA, Univ. Carlos III

Tema 2.1: Problema

B) Diseño físico-lógico con compresión de datos.

Calcular la densidad ideal y la densidad de fichero para un fichero con RLV con 1.000 regs. en un soporte cuyo tamaño de bloque es 1Kb. *
Incluir Género C(9) con tres valores posibles {terror, drama, comedia}

Campo	Compuesto	Tipo	Frecuencia %	Veces max.	Veces. media	Tamaño max.	Tamaño medio
Título		C	100			42	30
Año		N	100			2	2
Productora		C	100			20	9
Género		C	100			1	1
Director			100				
	Nombre	C	100			20	12,5
	Apellido	C	100			20	12,5
Actores		C	75	5	2,8		
	Nombre	C	100			20	12,5
	Apellido	C	100			20	12,5
BS			50				
	Nombre	C	100			20	12,5
	Apellido	C	100			20	12,5
	Formato	C	100	1	1	1	1



@LABDA, Univ. Carlos III

Ejercicio de diseño físico-lógico

Ejemplo 2: Diseño físico-lógico del registro “estudiante ITIG” optimizado.

- Nombre completo (20 nombre + 20 apellido)
- Teléfonos (+34 y número) si los tiene, como máximo 3
- Dirección (50 como máximo)
- Asignaturas aprobadas. Nombre de la asignatura (50 carac. como máximo), curso (1..3) y convocatoria (1..6). En total se sabe que un alumno como máximo tendrá 40 asignaturas aprobadas
- Si ha cursado las 40 asignaturas, también necesitamos saber si ha realizado el proyecto y si es así el título (70 carac.) y el año en que lo presentó.



@LABDA, Univ. Carlos III

Ejercicio de diseño físico-lógico

Campo	Compuesto	Tipo	Frecuencia %	Veces max.	Veces. media	Tamaño max.	Tamaño medio
Nombre-C			100				
	Nombre	C				20	10
	Apellidos	C				20	15
Telefonos			80	3	2		
	Prefijo	C				3	3
	Número	N				9	9
Dirección			100			50	40
Asig-Apro			90	40	20		
	Nombre	C				50	30
	Curso	N				1	1
	Convocatoria	N				1	1
Proyecto			20				
	Título	C				70	50
	Año	N				4	4



@LABDA. Univ. Carlos III

Tema 2.1: Factores en un Fichero

Tasa de crecimiento: indica en qué proporción aumentan los registros del fichero por unidad de tiempo

$$T_C = T_I - T_B$$

- la *tasa de crecimiento esperada* permite reservar espacio vacío (para permitir la llegada de nueva información sin reorganizar)
- el espacio físico necesario para ese crecimiento dependerá de la **organización del fichero**.



@LABDA. Univ. Carlos III

Tema 2.1: Factores en un Fichero

- Tasa de Actividad: porcentaje de reg. tratados en un proceso

$$T_A = \frac{\text{registros procesados}}{\text{registros totales}} \cdot 100$$

} respecto de un proceso

- Volatilidad: porcentaje de cambios por unidad de tiempo
(según cada operación, consideraremos tasas de inserción, borrado, y modificación)

$$T_I = \frac{\text{registros insertados}}{\text{registros totales}} \cdot 100$$
$$T_B = \frac{\text{registros borrados}}{\text{registros totales}} \cdot 100$$
$$T_M = \frac{\text{registros modificados}}{\text{registros totales}} \cdot 100$$

} referido a unidad de tiempo



@LABDA. Univ. Carlos III

Tema 2.2: Organización Serial

Surge gracias a los soportes seriales:

Soporte Serial: proporciona registros físicos en serie, esto es, que se registran uno detrás de otro, siempre en el mismo orden

Ejemplo de soporte serial: la cinta magnética

Funcionamiento

Instrucciones: leer (bloque) y reset

- procesos selectivos: tiene que buscar uno o varios registros
 - antes de buscar, se apunta al principio del fichero (reset)
 - se van leyendo todos los registros hasta identificar el buscado
 - si se quiere localizar varios, se lee todo el fichero
- procesos a la totalidad: no precisan localización ni orden → óptimos



@LABDA. Univ. Carlos III

Tema 2.2: Organización Serial

Interacción con Ficheros Seriales:

Recuperación:

- consulta: se recupera el contenido

Actualización:

- inserción: se añaden registros al final del fichero
- borrado:
 - borrado físico: se *vacía* el registro → se desplaza el resto
 - borrado lógico: se *marca* el registro → se genera un *hueco*
- modificación:
 - registros fijos: se altera el contenido
 - registros variables: se borra el antiguo y se reinserta modificado



@LABDA, Univ. Carlos III

Tema 2.2: Organización Serial

Cálculo del tiempo de acceso:

- *máximo*: peor caso → leer todos

$$t_{max} = n^{\circ} \text{registros} \cdot t_{acceso_registro}$$

- *medio*: se calcula como el acceso a la mediana (el de en medio)

$$t_{medio} = \frac{n^{\circ} \text{registros} + 1}{2} \cdot t_{acceso_registro}$$

¿De qué tipo de registro hablamos: físico o lógico?



@LABDA, Univ. Carlos III

Tema 2.3: Organización Secuencial

*Surge a partir de la serial, introduciendo un orden de registros
Se propicia gracias a una clave de ordenación física*

Instrucciones: **las seriales más desplazar (avanzar y retroceder)**

- Se acceden los bloques aleatoriamente, y por ello se precisa un mecanismo para poder localizar el comienzo del primer registro
 - a nivel físico (por bloques): comienzo de bloque
 - a nivel físico-lógico (registros consecutivos): marca de inicio/fin

Procesos:

- procesos selectivos: puede aprovechar el que vayan ordenados
 - localizar un registro: búsqueda dicotómica
 - localizar varios registros: dicotómica + acceso serial
 - claves alternativas: búsqueda serial (leer todo el fichero)
- procesos a la totalidad: gran eficiencia en procesos ordenados



@LABDA. Univ. Carlos III

Tema 2.3: Organización Secuencial

Interacción con Ficheros Secuenciales

Consecutivos: El primer carácter de un bloque puede no corresponder al primer registro → se necesita una marca de inicio (o fin) de registro

Recuperación (consulta): recuperar el contenido de un registro o varios.

Actualización:

- inserción: se añaden registros al final del fichero → altera el orden (*)
- borrado: igual que en serial, pero es más difícil reutilizar huecos.
- modificación:
 - registros fijos: se altera el contenido si no altera el orden (si no se modifica la clave de ordenación)
 - para modificar clave de ordenación, y en registros variables: se borra el antiguo y se reinserta modificado → altera el orden (*)

Nota(*): observar que se genera un *área desordenada* al final del fichero

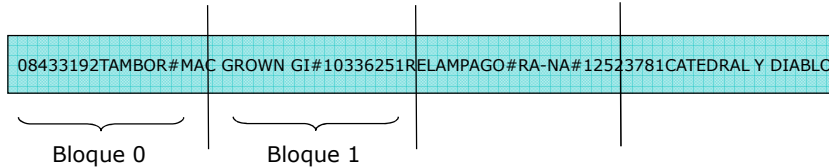
↪ Actualización en no consecutivo puede que exista hueco en el bloque



@LABDA. Univ. Carlos III

Tema 2.3: Organización Secuencial

Consecutivos: El primer carácter de un bloque puede no corresponder al primer registro → se necesita una marca de inicio (o fin) de registro



¿cómo podemos interpretar la información contenida en el bloque 1 sin haber leído previamente el bloque 0?

El diseño lógico de registro se corresponde con

ISBN C(8),
Título C(72),
Editorial C(35)



@L.A.B.D.A. Univ. Carlos III

Tema 2.3: Organización Secuencial

Localización en Ficheros Secuenciales:

Búsqueda dicotómica:

- mirar el de en medio: si coincide, fin
- si no coincide, escoger la mitad que contiene el elemento buscado
- volver a empezar (sobre la mitad escogida)

Para claves **no unívocas**, se hará **Búsqueda Dicotómica Extendida:**

- Buscar primer elemento (por *búsqueda dicotómica*)
- Buscar hacia arriba hasta encontrar uno distinto (fallo)
- Buscar hacia abajo hasta encontrar uno distinto (fallo)
- Las entradas que coinciden, se incluyen en un '*conjunto resultado*'



@L.A.B.D.A. Univ. Carlos III

Tema 2.3: Organización Secuencial

Tiempo de acceso Búsqueda Dicotómica:

- *consideramos sólo el máximo*:

peor caso → encontrarlo en la última vuelta

media → similar al tpo. máximo

$$n^{\circ}accesos_{max} = \lceil \log_2(n+1) \rceil$$

$$t_{max} = n^{\circ}accesos_{max} \cdot t_{acceso_registro}$$

- El número de elementos en la búsqueda (n) depende de la relación físico-lógica:
 - en ficheros con registros expandidos, se trata del número de registros log.
 - en el resto (caso habitual) se trata del número de bloques del fichero

- En ficheros secuenciales, se usa búsqueda dicotómica con una salvedad:

- búsqueda dicotómica sobre área ordenada
- si no se encuentra, **búsqueda serial sobre área desordenada**

$$n^{\circ}accesos_{sec} = \lceil \log_2(n+1) \rceil + n^{\circ}accesos_{serial}$$



@LABDA, Univ. Carlos III


Tema 2.3: Organización Secuencial

Tiempo de acceso Búsqueda Dicotómica Extendida:

- *Se busca a través de una clave que presenta k coincidencias (de media), y que tiene v valores distintos ($k * v =$ número total de registros).*

- *Para hallar el número de accesos, consideramos sólo el máximo:*

- peor caso →
- encontrar el primer elemento en la última vuelta
 - el bloque anterior se leería sólo para encontrar un fallo
 - se recuperarán $k+1$ registros (k coincidencias y un fallo)

no consecutiva 

$$n^{\circ}accesos_{max} = \lceil \log_2(n+1) \rceil + \lceil \frac{(k+1)}{f_b} \rceil$$

- El número de elementos en la búsqueda (n) depende de la relación físico-lógica:
 - si los k registros caben en un bloque, se utilizará n =número de bloques
 - en el resto de casos, se debe utilizar el número de valores distintos (v)
- Se debe seguir considerando el área desordenada (siempre que exista).



@LABDA, Univ. Carlos III

Tema 2.4: Mantenimiento serial

Reutilización de huecos

Borrado Físico:

- en organización serial y registros fijos:
 - para borrar se lee el último registro y se escribe sobre el que se desea eliminar; el último se vacía.
- en organización serial y registros variables:
 - tamaños diversos → para borrar se desplazan todos los registros posteriores al eliminado → altamente **ineficiente**

Borrado Lógico:

- en organización serial (en general):
 - para borrar se introduce una marca de borrado lógico (1 acceso escritura), y al insertar se recorre el fichero buscando un hueco de tamaño adecuado. Si no hubiera un hueco suficientemente grande, se inserta al final.
 - *Lista de Huecos*: (tamaño+posición)

Reorganización = Compactación: se van desplazando registros para eliminar huecos



@LABDA, Univ. Carlos III

Tema 2.4: Mantenimiento secuencial

Reutilización de huecos

1.- Lista ordenada de huecos: (clave ant, clave post, tamaño)

- al borrar actualiza la lista de huecos;
- al insertar, comprueba si existe un hueco apropiado, lo usa y actualiza la lista; si no existe hueco, inserta el nuevo registro al final

2.- Espacio libre distribuido: (org. secuencial no consecutiva)

- el fichero se divide en 'cubos' (bloque físico-lógico de registros)
- cada cubo puede contener un bloque (o más; habitualmente uno)
- cada cubo reserva un porcentaje de su espacio para registros que crecen, y un porcentaje para nuevos registros → (casi) todos los cubos tienen hueco
- para insertar, se mira si cabe en su cubo; si no, se inserta al final



@LABDA, Univ. Carlos III

Tema 2.4: Mantenimiento secuencial

Reorganización:

reescribir todos los registros ordenados

- *Problema: algoritmos sobre el propio fichero muy pesados (coste elevado)*
- *Habitualmente se usan algoritmos basados en almacenamiento auxiliar*

¿Cómo afecta a la **tasa de crecimiento** la organización del fichero?

- en un fichero serial con reutilización de huecos coincidiría con la tasa de crecimiento
- fichero sin reutilizar huecos coincidiría con la tasa de inserción



@LABDA, Univ. Carlos III