

Tema 5c



El Nivel Interno

Apartado 3

Organización y métodos de
acceso: Acceso Directo

Tema 5c Introducción al Nivel Interno

Acceso directo

☞ Otra forma de acceder a un registro almacenado:

- No hay una estructura adicional.
- Se usa un *algoritmo* que nos indique *directamente* la *posición* del *registro* deseado.

☞ Acceso directo:

- *Calcular directamente la dirección de un registro mediante la aplicación, a un campo determinado del mismo, de algún algoritmo o función.*
 - El campo debe identificar unívocamente al registro.

Tema 5c Introducción al Nivel Interno

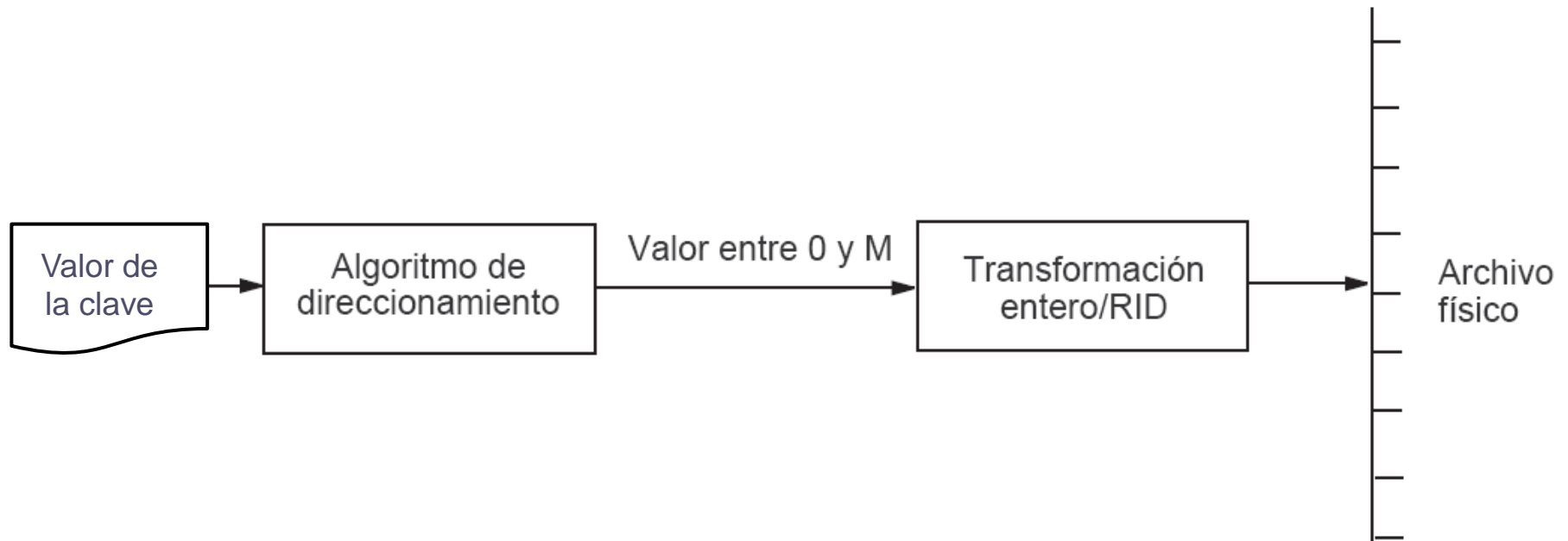
Acceso directo

Funcionamiento:

- Normalmente no es posible establecer una clave física que sea totalmente correlativa y única para cada registro.
- Hay que buscar un algoritmo que transforme los valores de un cierto campo en una dirección.
 - Entrada: campo clave
 - Salida: valor entero positivo fácilmente transformable en RID.

Tema 5c Introducción al Nivel Interno

Acceso directo



Tema 5c Introducción al Nivel Interno

Acceso directo

- ☛ Los **algoritmos** de direccionamiento **no** suelen **mantener** el **orden** de la **clave**.
 - Los **registros no** están **almacenados** según el **orden de su clave física**.
 - Problemas con la **recuperación por intervalos**.

Tema 5c Introducción al Nivel Interno

Acceso directo: algoritmos

- Hay una gran **variedad** de **algoritmos**:
 - Dependen del tipo de clave:
 - Si la **clave** es **alfanumérica**, hay que **transformarla** a un valor **numérico**.
 - Suelen estar basados en un mecanismo de **generación de números pseudoaleatorios**:
 - Cuadrados centrales*:
 - Se eleva la **clave** al **cuadrado** y se eligen **tantos dígitos centrales** como sea **necesario**.
<http://research.cs.vt.edu/AVresearch/hashing/midsquare.php>
 - Congruencias*:
 - Se **divide** la clave **por M** y se **toma el resto**. (M suele ser primo).
<http://research.cs.vt.edu/AVresearch/hashing/modfunc.php>
 - Desplazamiento*:
 - Se **superponen** adecuadamente los **dígitos binarios** de la **clave** y luego **se suman**.
 - Conversión de base*:
 - Se **cambia base** de numeración y **se suprimen** algunos dígitos resultantes.
 - [Comparación del rendimiento de diversas funciones Hash](#)

Tema 5c Introducción al Nivel Interno

Acceso directo: problemas

- Salvo que el campo clave se diseñe para ello, es **prácticamente imposible** encontrar una **transformación** que dé un valor entero positivo en un rango de valores limitado **tal que**:
 - No haya dos valores distintos de clave que den lugar al mismo número.
 - *Colisiones*.
- Los **algoritmos producen huecos**:
 - Zonas vacías del rango de salida, **no asignadas** por el algoritmo.
 - Se traducen en **huecos** en el **fichero de datos**.

Tema 5c Introducción al Nivel Interno

Acceso directo: problemas

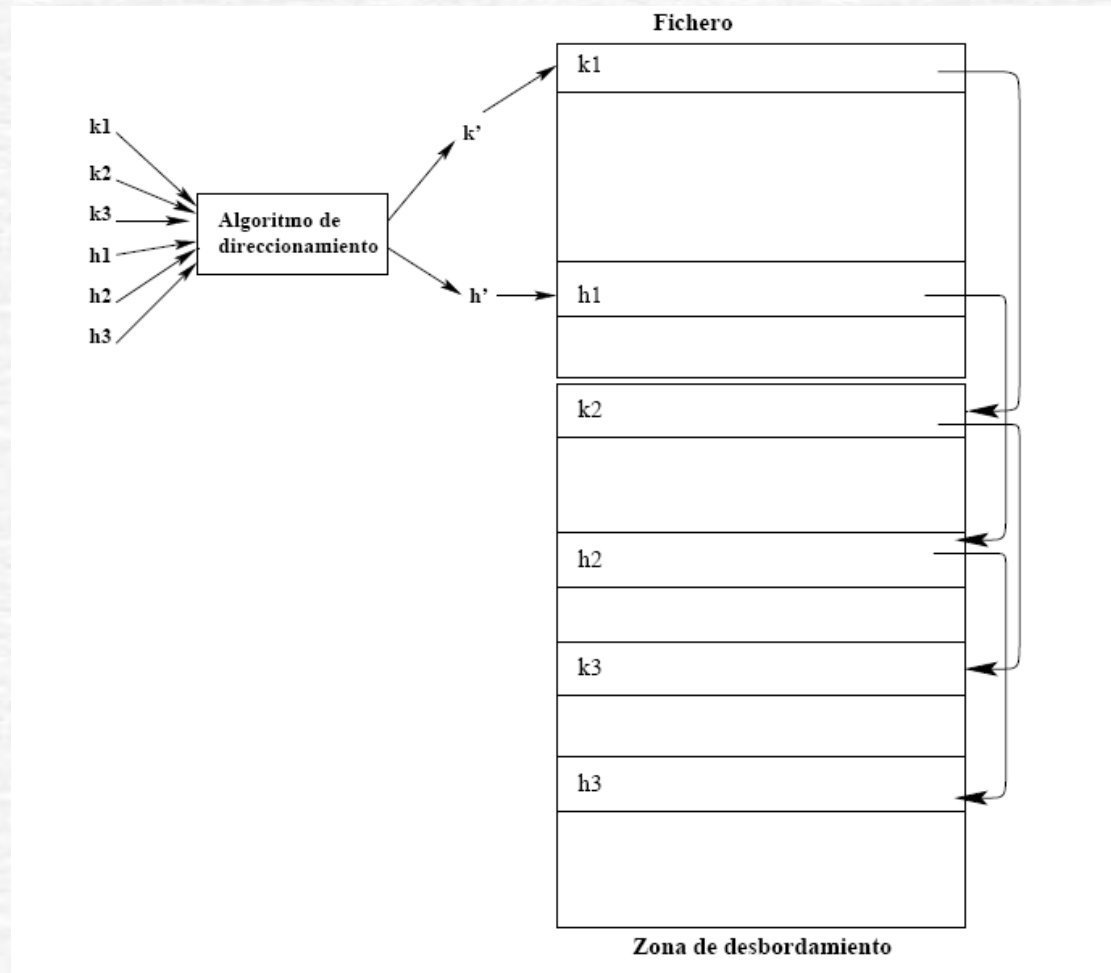
Para gestionar **colisiones y huecos**:

- **Combinar el acceso directo con una gestión mediante listas de colisión:**
 - Zona de desbordamiento.
 - Colisión:
 - El registro problemático se almacena en la zona de desbordamiento.
 - Los **sinónimos** (registros con claves que producen colisión) **se conectan mediante una lista**.

Tema 5c Introducción al Nivel Interno

Acceso directo: problemas

Selecciona en el Applet: Chaining W separate overflow .
Selecciona “setup” y establece 1 en: “Slots per bucket”.



Tema 5c Introducción al Nivel Interno

Acceso directo: problemas

- ☞ Si **crecen las listas** de sinónimos:
 - El acceso **directo** puro **no** resulta **adecuado**.
 - Mantener listas.
 - Zona de **desbordamiento casi** como el **fichero original**.
- ☞ Han aparecido técnicas más sofisticadas:
 - **Hashing**

Tema 5c Introducción al Nivel Interno

Acceso directo: Hashing básico

- Si el problema principal es que los valores de las claves no están uniformemente distribuidos en el intervalo $[0, M]$:
 - Se acumulan en una parte de este intervalo.
 - Solución:
 - Asignar más espacio a esa parte del intervalo.

Tema 5c Introducción al Nivel Interno

Acceso directo: Hashing básico

👉 Técnica:

- Se divide el espacio del fichero en "cubos" (buckets).
- El algoritmo de direccionamiento asigna cubos, no direcciones concretas.
- En cada "cubo" puede haber más de un registro.
- Ciertos rangos de valores tienen asignados más cubos que otros.
- Se complementa con el uso de "cubos de desbordamiento".

Tema 5c Introducción al Nivel Interno

Acceso directo: Hashing básico

Parámetros:

- Número de cubos.
- Tamaño de los cubos (relación con bloques físicos). ("slots")
- La transformada clave/dirección, que debe tener en cuenta la distribución de la clave según rangos para que unos cubos no se llenen mucho y otros se queden muy vacíos.
- Selecciona en el Applet: Chaining W separate overflow
 - Selecciona "Setup":
 - Slots per bucket: 2; Buckets: 7; Size overflow: 10; Data Size: 20
 - Pincha "Execute".

Tema 5c Introducción al Nivel Interno

Acceso directo: Hashing básico

☞ Para insertar un registro:

- Transformar la clave.
- Localizar el cubo correspondiente.
- Si hay sitio se inserta el registro y hemos terminado.
- Si no hay sitio, se sitúa el registro en un cubo de desbordamiento conectándolo con el cubo que realmente le corresponde mediante punteros.
- Sobre el enlace transparencia anterior:
 - Pulsa "Step" sucesivamente para ver como se van insertando los registros de uno en uno.

Tema 5c Introducción al Nivel Interno

Acceso directo: Hashing básico

El proceso de búsqueda:

- Transformar la clave.
- Localizar el cubo correspondiente.
- Realizar una búsqueda secuencial dentro del cubo.
 - Si hemos encontrado el registro, el proceso termina.
 - En caso contrario, se impone “un barrido por punteros” a través de los cubos de desbordamiento.
- Describid el proceso para encontrar un determinado valor de la clave sobre el ejemplo de la transparencia anterior.

Tema 5c Introducción al Nivel Interno

Acceso directo: Hashing básico

Fichero

250	
350	
1150	
1900	
2010	
2150	
3250	
2200	
5100	
4200	
6700	
6200	
7650	

Zona de desbordamiento

1230	
0034	
0035	
3254	
2450	

Tema 5c Introducción al Nivel Interno

Acceso directo: Hashing dinámico

- El hashing básico sigue teniendo **problemas**:
 - Es necesario **conocer** la **distribución previa** de las **claves** para **asignar adecuadamente** los **cubos**.
 - En otro caso siguen apareciendo **huecos/colisiones**.
 - Al **aumentar** el número de **registros**, **aumentan** los registros en **páginas de desbordamiento**.
 - Se hacen necesarias las **reorganizaciones**.

Tema 5c Introducción al Nivel Interno

Acceso directo: Hashing dinámico

☞ Solución:

- Trabajar de forma **dinámica**.
- ☞ Se **parte** de una configuración uniforme y de **pocos cubos**.
- ☞ Los **restantes, se van generando** conforme se necesitan.
 - Se asignan a los rangos conforme la afluencia de registros lo demanda.
 - *Hashing dinámico o extensible*

Tema 5c Introducción al Nivel Interno

Acceso directo: Hashing dinámico

👉 Técnica:

- El valor transformado del campo clave nos lleva a la entrada de una tabla índice que se almacena en memoria.
- Allí está la dirección del cubo donde se encuentran los registros que tienen asociado este valor transformado.
- Puede ocurrir que varias entradas de la tabla conduzcan al mismo cubo.
- Proceso:
 - Inicialmente, todas las entradas apuntan al mismo cubo.
 - A medida que vamos insertando registros, se van generando nuevos cubos y cambiando las salidas de la tabla índice.

Tema 5c Introducción al Nivel Interno

Acceso directo: Hashing dinámico

Algoritmo de Hashing Dinámico

Datos de partida:

- k = clave física para direccionar
- $k' = h(k)$ valor entero entre 0 y M
- n = número de bits que tiene k' en binario
- $d \leq n$, los d primeros dígitos de k' seleccionan el cubo donde está el registro y se llaman pseudollave.
- $b < d \leq n$, inicialmente el archivo tiene 2^b cubos distintos, como máximo tendrá 2^d

Tema 5c Introducción al Nivel Interno

Acceso directo: Hashing dinámico

Algoritmo

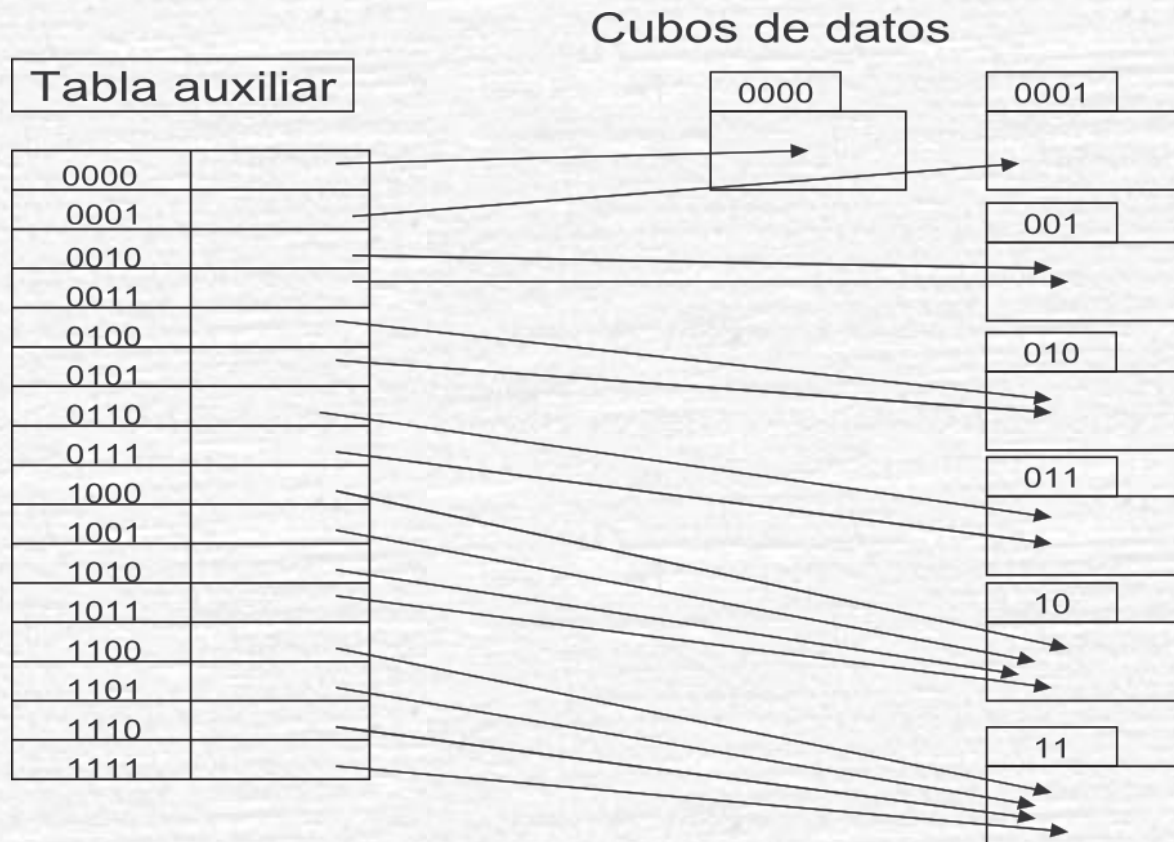
- Se considera una **tabla índice** en memoria con 2^d filas.
- En la **primera columna** de esta tabla (**valores de campo clave**) se sitúan todas las **posibles sucesiones** de d dígitos binarios:
 - d es la “**profundidad global**” de la **tabla**.
- En principio, todas las **entradas** cuyos b primeros dígitos son **iguales** apuntan al **mismo cubo**.
 - Allí se almacenan los **registros** cuyo valor de k' tiene esos b primeros dígitos.
- Todos los **cubos** tienen en principio “**profundidad local**” igual a b .
- Cuando **se llena** un **cubo** se divide en 2, poniendo en uno de ellos los **registros** con el **dígito $b+1$ de k' a 0** y en el otro los que lo tienen igual a 1. La **profundidad local** de estos cubos **aumenta una unidad**.

Tema 5c Introducción al Nivel Interno

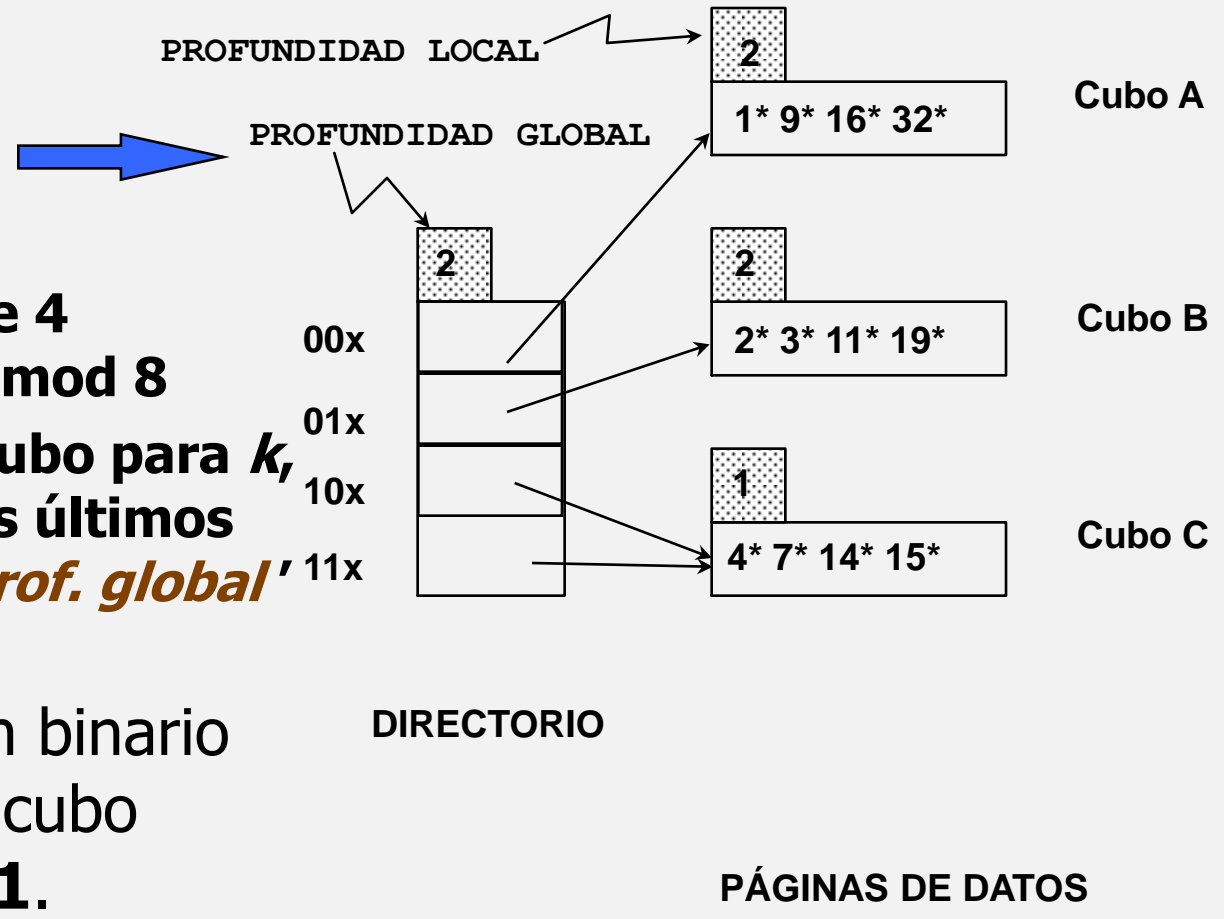
Acceso directo: Hashing dinámico

Videoanimación de Hashing Extensible

Nota: En esta videoanimación el **desdoble** se realiza **partiendo** del **dígito menos significativo** del valor devuelto por $h(k)$, en vez de partir **del dígito mas significativo** como hemos **descrito** en este tema.



Ejemplo



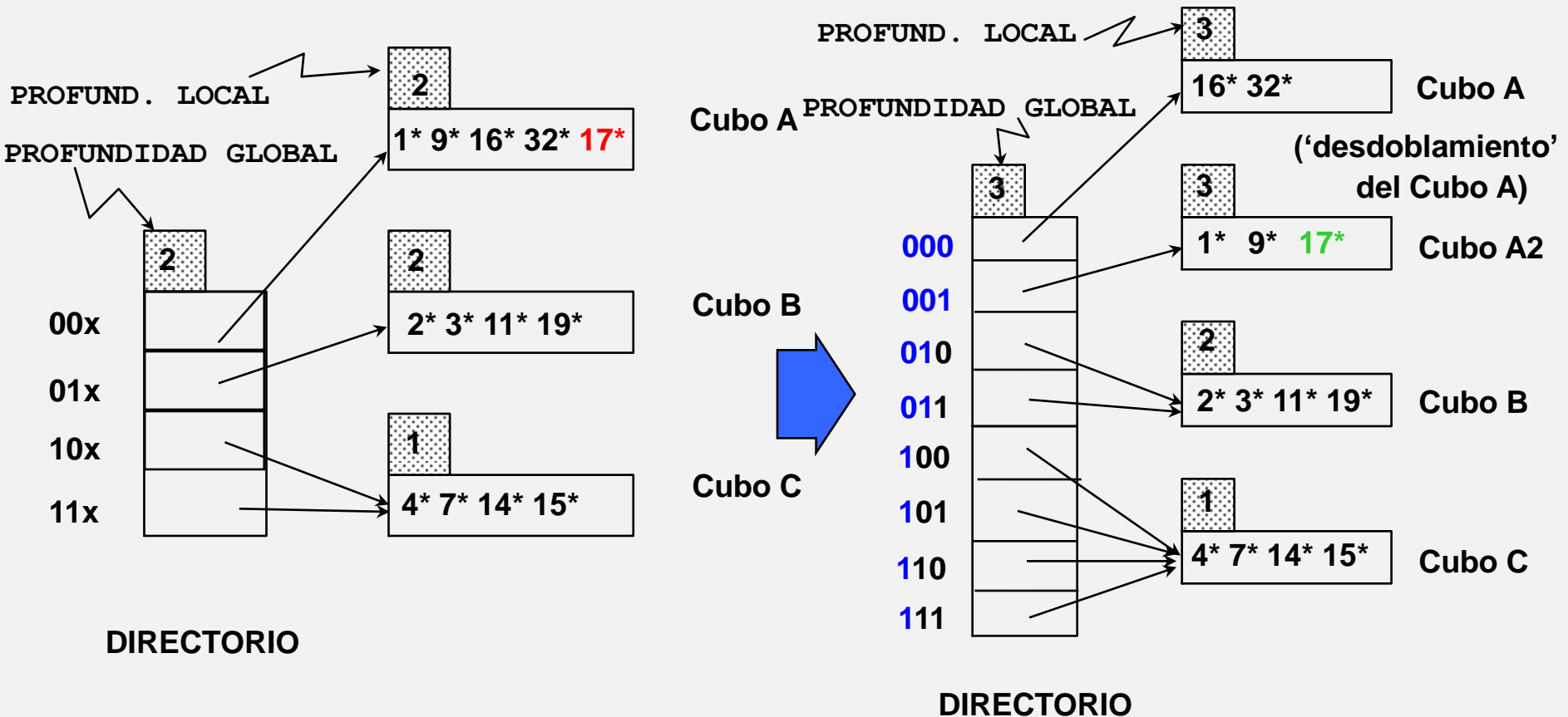
- **Directorio:** vector de 4 elementos, $h(k) = k \bmod 8$
- Para encontrar un cubo para k , se toman de $h(k)$ los últimos bits marcados en '*prof. global*' (2).

- Si $h(k) = 3 =$ en binario **011**, cae en el cubo apuntado por **01**.

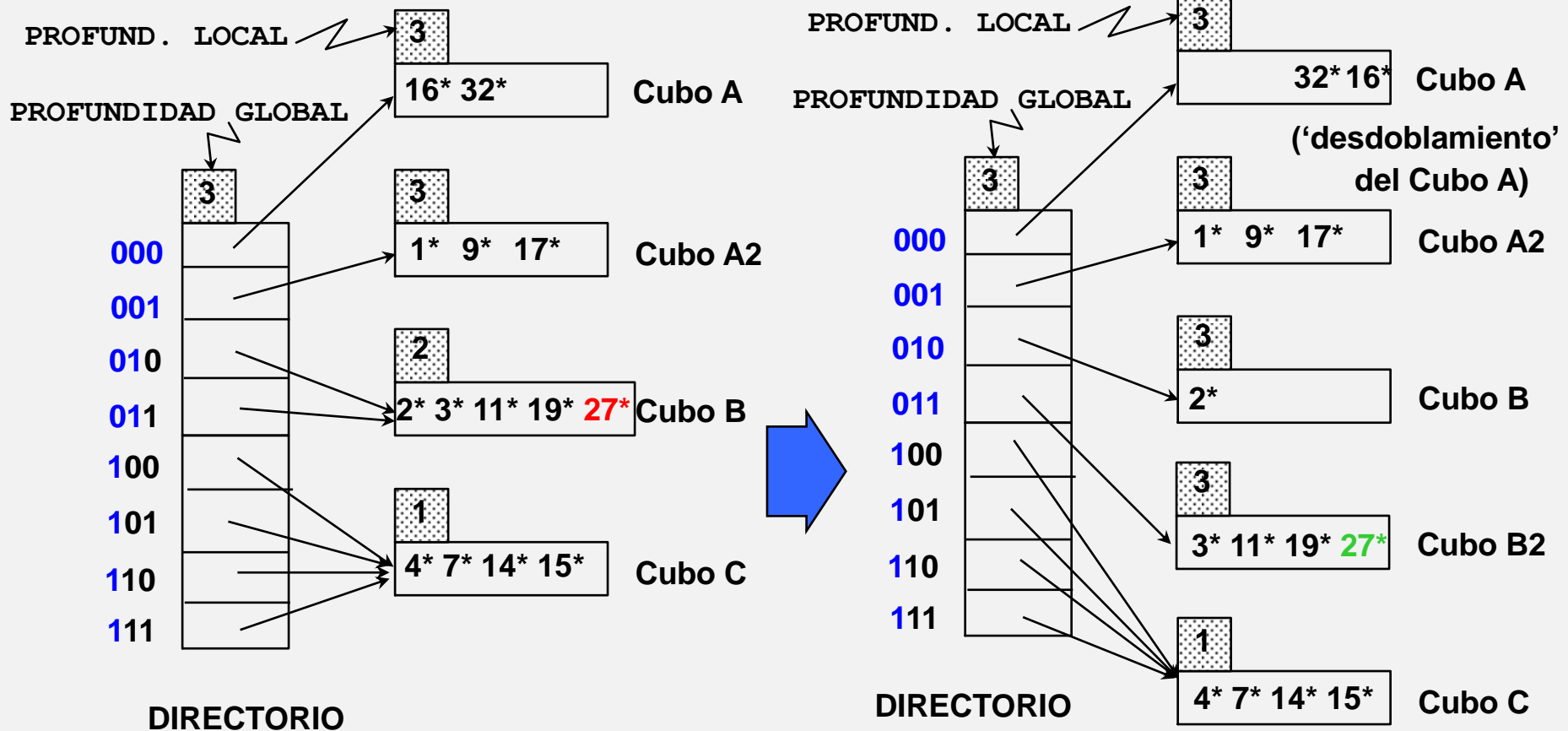
❖ **Inserción:** Si el cubo está lleno, se divide y se re-distribuyen entre los dos cubos.

❖ Si es necesario, se desdobra el directorio (no siempre es necesario) siempre que la profundidad local sea menor o igual que la global.

Insertamos 17, $h(17)=1=001$ (Causa desdoblamiento del cubo A y del directorio)



Insertamos 27, $h(27)=3 = \mathbf{011}$ (Sólo causa desdoblamiento del Cubo B)



Estructura Hash. Din. para: { 2, 3, 4, 7, 11, 19, 23, 29, 31 } con Func. De Dispersión: Mod 8 y 3 registros por cubo.

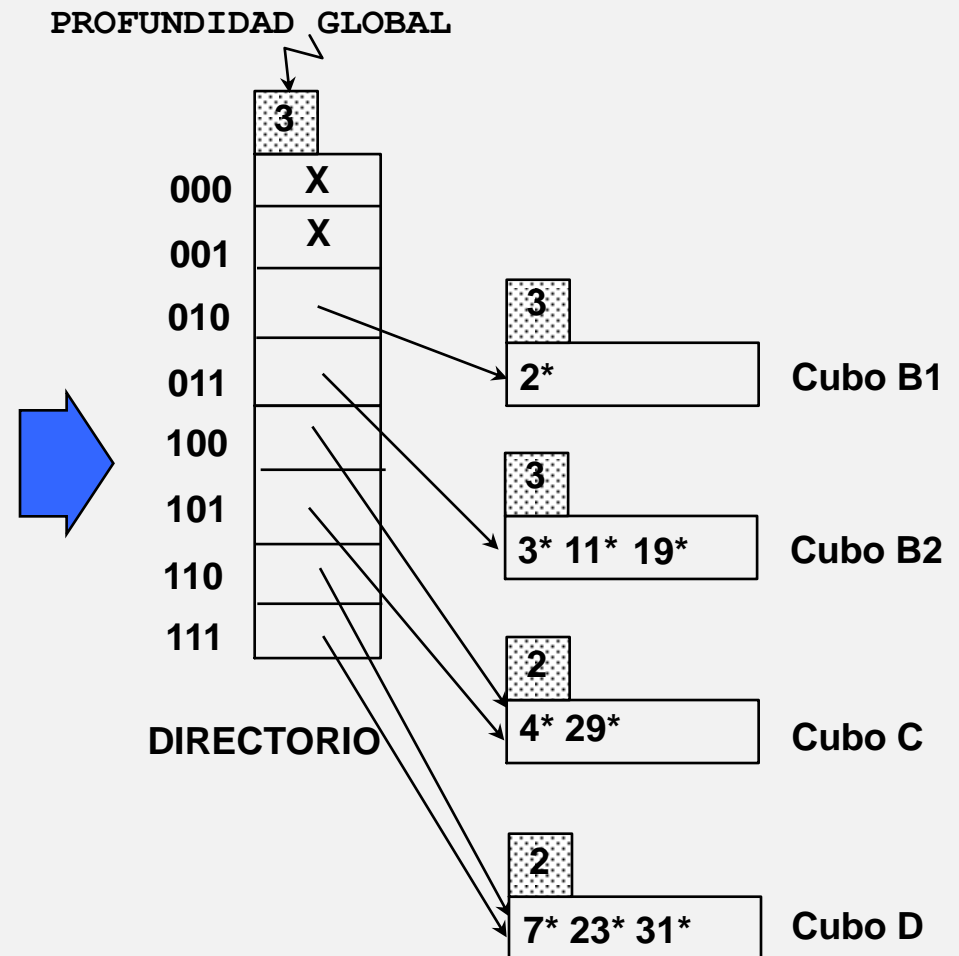
Mod 8 (bin)	Nº	Prof. Local	Puntero
000			
001			
010	2		
011	3,11,19		
100	4		
101	29		
110			
111	7,23,31		



Mod 8 (bin)	Nº	Prof. Local	Puntero
000		NA	
001			
010	2	3	
011	3,11,19	3	
100	4,29	2	
101			
110	7,23,31	2	
111			

Estructura Hash. Din. para: { 2, 3, 4, 7, 11, 19, 23, 29, 31 } con Func. De Dispersión: Mod 8 y 3 registros por cubo.

Mod 8 (bin)	Nº	Prof. Local	Punte ro
000		NA	X
001			X
010	2	3	CuboB1
011	3,11,19	3	CuboB2
100	4,29	2	CuboC
101			
110	7,23,31	2	CuboD
111			



Eliminación de 11 y 31

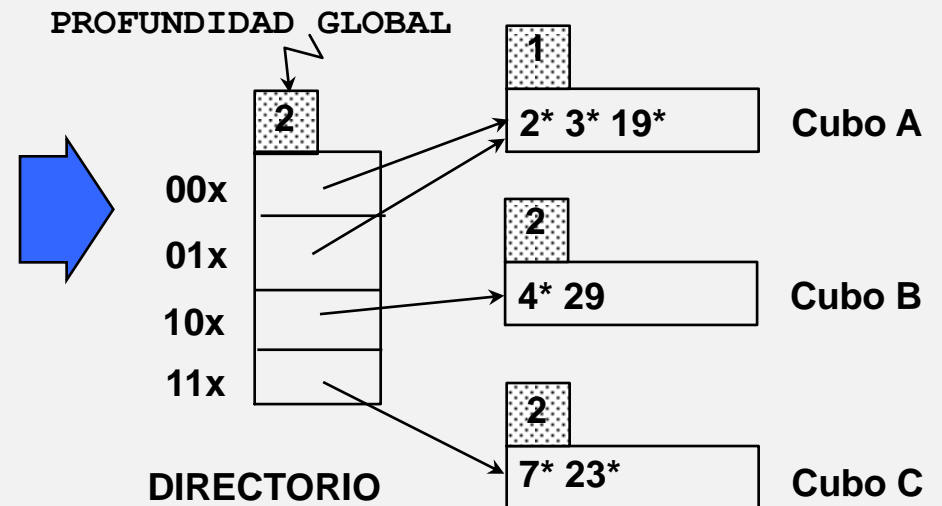
Mod 8 (bin)	Nº	Prof. Local	Punte ro
000		NA	X
001			X
010	2	3	
011	3, 11 , 19	3	
100	4, 29	2	
101			
110	7, 23, 31	2	
111			



Mod 8 (bin)	Nº	Prof. Local	Punte ro
000	2, 3, 19	1	
001			
010			
011			
100	4, 29	2	
101			
110	7, 23	2	
111			

Eliminación de 11 y 31

Mod 8 (bin)	Nº	Prof. Local	Punte ro
0 00	2,3,19	1	CuboA
0 01			
0 10			
0 11			
1 00	4,29	2	CuboB
1 01			
1 10	7,23	2	CuboC
1 11			



Inserción de: 1, 15, 36, 40, 46 y 25

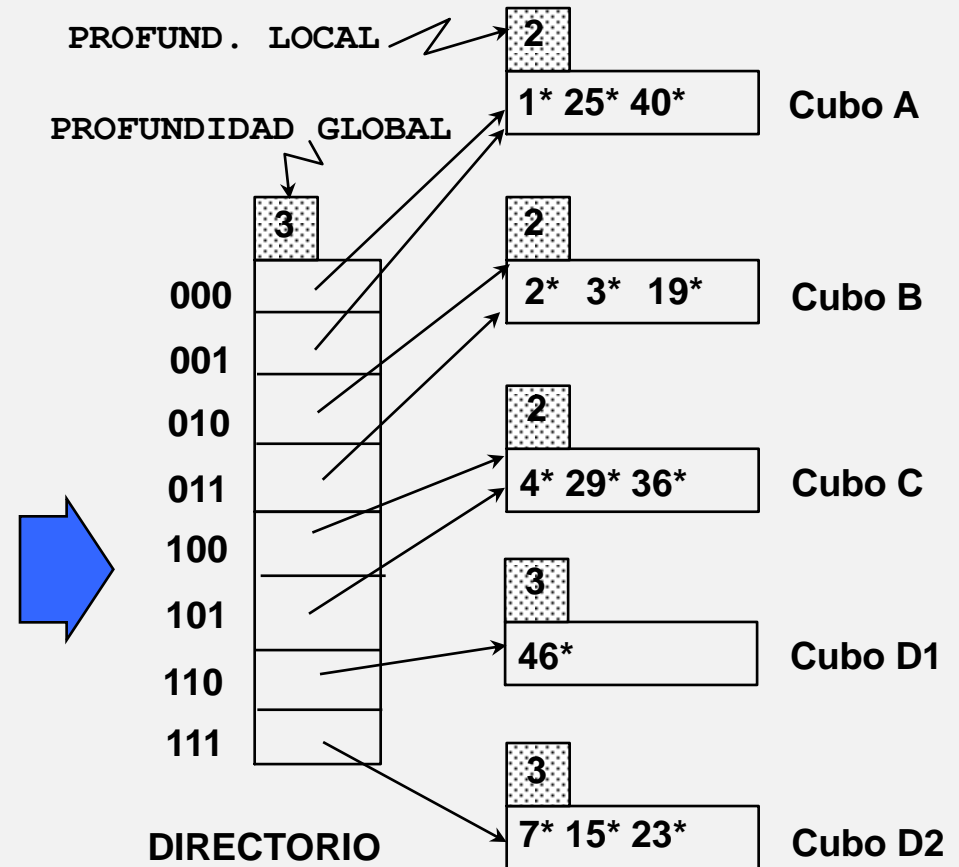
Mod 8 (bin)	Nº	Prof. Local	Punter o
000	1,2,3, 19,25, 40	1	Cubo A
001			
010			
011			
100	4,29,36	2	Cubo B
101			
110	7,15, 23,46	2	Cubo C
111			



Mod 8 (bin)	Nº	Prof. Local	Puntero
000	1,25,40	2	
001			
010	2,3,19	2	
011			
100	4,29,36	2	
101			
110	46	3	
111	7,15,23	3	

Inserción de: 1, 15, 36, 40, 46 y 25

Mod 8 (bin)	Nº	Prof. Local	Punte ro
000	1,25,40	2	Cubo A
001			
010	2,3,19	2	Cubo B
011			
100	4,29,36	2	Cubo C
101			
110	46	3	CuboD1
111	7,15,23	3	CuboD2



Tema 5c Introducción al Nivel Interno

Acceso directo: Hashing dinámico

- El hashing **dinámico** supera los problemas clásicos del **acceso directo**.
- También tiene sus **inconvenientes**:
 - Utilizar una **tabla índice adicional** (nuevos accesos a disco si no cabe en memoria).
 - El **tamaño máximo** de la tabla **depende de "n"** y, por tanto, de la **función de dispersión** que se elija (cuantos **valores distintos M** pueda devolver).

Tema 5c Introducción al Nivel Interno

Otras técnicas de acceso: multilistas

☞ Ideas básicas de la organización encadenada:

- La organización **indexada** permite el **acceso** a los **registros** según el **valor** de un **campo** que **no** es la **clave física**.
- La **organización encadenada** o con estructuras **multilista** se orientó inicialmente a **resolver este mismo problema**.
- La **idea** básica es:
 - Se construyen **listas** lineales **enlazando** aquellos **registros** que tienen el **mismo valor en el campo clave**.
 - Se usa un **fichero adicional** donde se recogen los **valores** de la **clave** convenientemente **ordenados** y **conectados** con el **principio** y el **final** de su lista.

Tema 5c Introducción al Nivel Interno

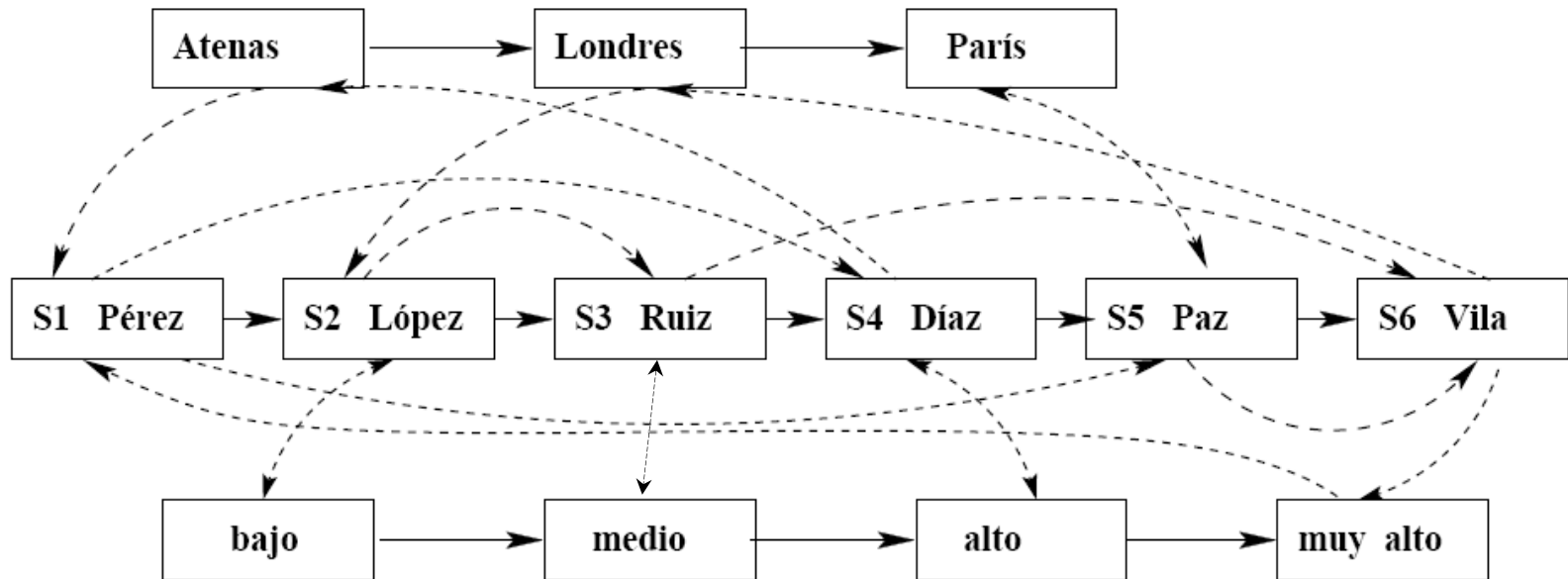
Otras técnicas de acceso: multilistas

➤ Ejemplo:

Código	Nombre	Nivel	Ciudad
S1	Pérez	Muy alto	Atenas
S2	López	Bajo	Londres
S3	Ruiz	Medio	Londres
S4	Díaz	Alto	Atenas
S5	Paz	Muy alto	París
S6	Vila	Muy alto	Londres

Tema 5c Introducción al Nivel Interno

Otras técnicas de acceso: multilistas



Tema 5c Introducción al Nivel Interno

Otras técnicas de acceso: multilistas

✓ Ventajas:

- Es muy eficaz para acceder según los valores del campo clave.
- Se elimina redundancia y se facilitan las operaciones de actualización.

Tema 5c Introducción al Nivel Interno

Otras técnicas de acceso: multilistas

☞ Inconvenientes:

- Cuando se desea **obtener** información del fichero “**padre**” a partir del fichero “**hijo**”, el acceso puede ser muy **costoso**.
- Una estructura tan compleja, con tantas cadenas de punteros implicadas, suele tener **problemas de mantenimiento**.
- Siempre hay que diseñar un **sistema de acceso alternativo**, ya que el proceso siempre **empieza** con un acceso al fichero “**padre**” o al fichero “**hijo**”.

Tema 5c Introducción al Nivel Interno

Otras técnicas de acceso: multilistas

- Se han propuesto algunas **variantes**:
 - Utilizar cadenas de **punteros dobles**.
 - Reduce el proceso de búsqueda a través de los registros "hijos" a la mitad.
 - Mantener los **registros completos** en el fichero "**hijo**", de forma que los accesos "hijo/padre" sean innecesarios.

Tema 5c Introducción al Nivel Interno

Otras técnicas de acceso: multilistas

- ✓ Aunque la idea básica es poder recuperar registros a partir de los valores de un campo que no es su clave física:
 - Los índices ya lo permiten y son más sencillos.
- ✓ La importancia de las multilistas radica en su utilidad para conectar ficheros a través de un campo común.

Tema 5c Introducción al Nivel Interno

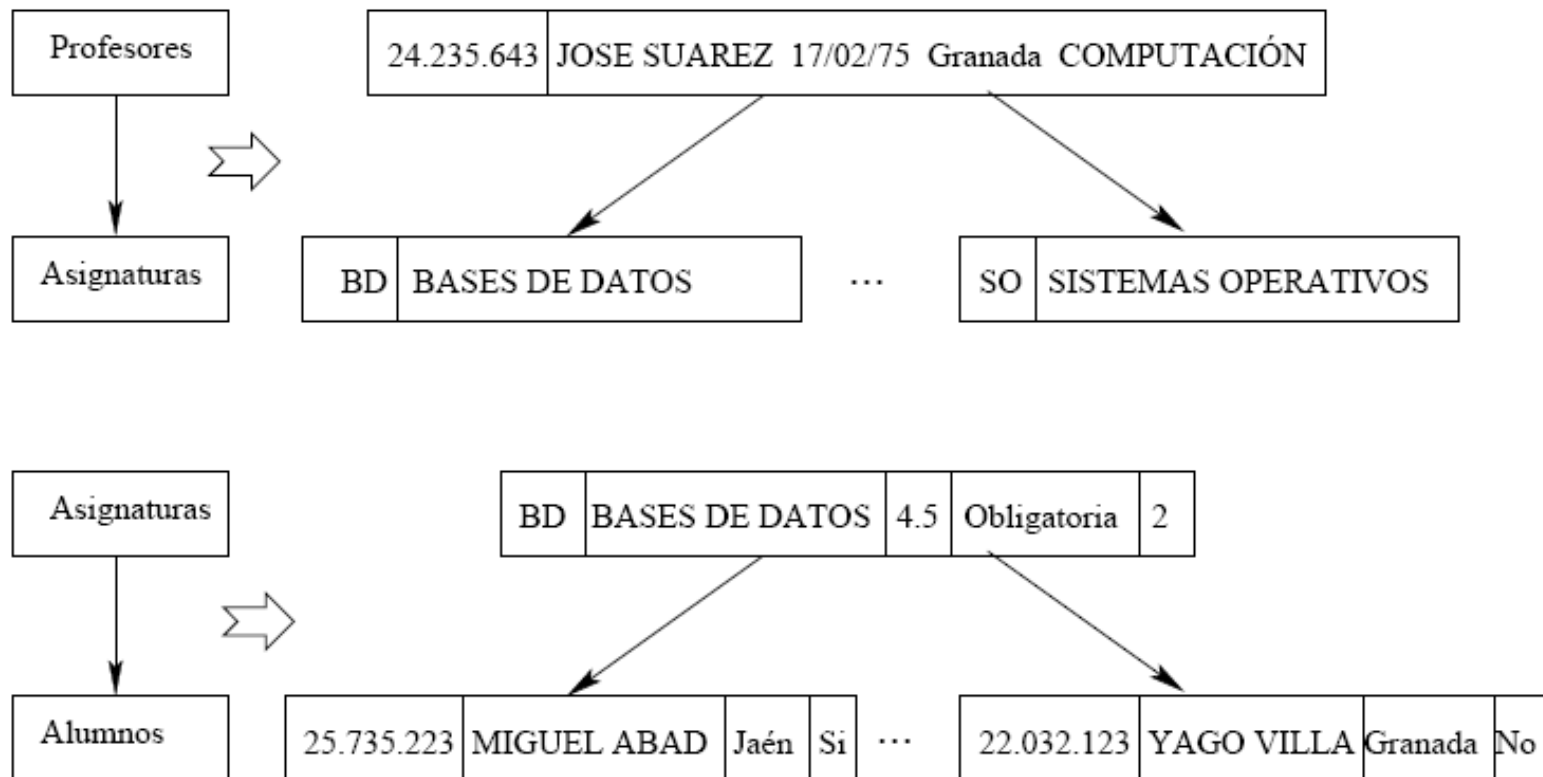
Otras técnicas de acceso: multilistas

● Ejemplo:

- Supongamos que el archivo de ciudades contiene información adicional acerca de cada ciudad:
 - País
 - número de habitantes
 - ...
- Organizando en dos ficheros con listas encadenadas uno de ciudades (padre) y otro de vendedores (hijo), podríamos resolver consultas:
 - "Obtener todos los vendedores que están en ciudades de más de 100.000 habitantes."
- El proceso de consulta sería:
 - Realizar un barrido secuencial del fichero de ciudades, obteniendo las ciudades con más de 100.000 habitantes.
 - Para cada una de estas ciudades, recorrer la lista asociada para obtener los vendedores que están en dicha ciudad.

Tema 5c Introducción al Nivel Interno

Otras técnicas de acceso: multilistas



Tema 5c Introducción al Nivel Interno

Otras técnicas de acceso: multilistas

Hay que hacer notar que una organización de este tipo nos **permitiría** resolver **consultas** del tipo:

- Encontrar los **nombres, créditos y cursos** de las asignaturas que imparten los **profesores** del departamento **"COMPUTACION"**.
- Encontrar los DNIs y nombres de los alumnos matriculados en la asignatura de nombre **"BASES DE DATOS"**.
- Encontrar los nombres de los alumnos a los que da clase el profesor de nombre **JOSE SUAREZ"**.

Tema 5c Introducción al Nivel Interno

Otras técnicas de acceso: multilistas

- Esta organización nos permite reflejar algunas de las características que corresponden al concepto de base de datos:
 - Conectar distintos ficheros de forma que la información se almacene sin redundancia y se pueda recuperar moviéndonos a través de los mismos.
- El uso sistemático de este tipo de organizaciones y el desarrollo de sistemas de acceso y definición para mantenerlas fue el origen de los primeros modelos de bases de datos.
 - “Modelos de datos basados en grafos”