

# ***Ficheros.***

Medios de almacenamiento:

Almacenamiento primario:

Caché, memoria principal

Almacenamiento secundario:

Discos magnéticos, memoria flash, cintas,  
almacenamiento óptico.

# ***Ficheros.***



# ***Ficheros.***

¿Por qué usamos almacenamiento secundario?

Memoria principal es volátil/Memoria secundaria es permanente

Tamaño de la memoria principal.

Coste almacenamiento.

Inconveniente: Velocidad.

# ***Ficheros.***

**El objetivo es minimizar el número de accesos a disco.**

# ***Ficheros.***

**El objetivo es minimizar el número de accesos a disco.**

# ***Ficheros.***

## **Conceptos básicos:**

### **Discos magnéticos:**

Controladora de disco: Unidad física en que está contenido el medio de grabación del disco.

Volumen: Paquete de discos gestionados por una controladora.

Superficies: Superficies grabables (discos) montados sobre un eje que pose el volumen.

Pistas: Corona circular en una superficie.

Cilindros: Conjunto de pistas con el mismo diámetro en las diferentes superficies.

# ***Ficheros.***

## **Conceptos básicos:**

### **Discos magnéticos:**

Sectores: Divisiones físicas de las pistas. Son las unidades más pequeñas de espacio que se pueden direccionar.

Clusters: Es una cantidad fija de sectores contiguos, y constituyen la unidad de espacio más pequeña que se puede asignar a un fichero. El gestor de ficheros ve un fichero como un conjunto de clusters, información que se encuentra almacenada en la tabla de particiones (fact, ntfs,...). Una vez se localiza un cluster, no hay que mover las cabezas para leer todos sus sectores.

Extent: Trozo de fichero formado por varios clusters contiguos. Mediante ellos se pretende enfatizar más la contigüidad física de los sectores. Cuántos más haya, más desperdigado estará el fichero y, por tanto, más tiempo se dedicará a mover las cabezas de lectura/escritura.

# ***Ficheros.***

## **Conceptos básicos:**

### **Discos magnéticos:**

Bloques: Son divisiones lógicas, y por lo tanto, su tamaño es variable. Este tamaño puede ser distinto para cada aplicación, y se puede fijar como un múltiplo del tamaño de registro, por lo que no hay problemas de fragmentación interna.

Direccionamiento: Superficie, Pista, Bloque

Búffer: Área contigua reservada en memoria principal, en la que cabe un bloque. Se usa para realizar operaciones de lectura/escritura entre la memoria principal y el disco.



# ***Ficheros.***

## **Conceptos básicos:**

### **Discos magnéticos:**

Coste de acceso al disco: En general hay tres factores que afectan directamente a la velocidad con que los datos se transfieren entre disco y memoria principal. Cada uno de estos factores consume un tiempo y conlleva una operación física:

Tiempo de búsqueda (seek). Tiempo en mover el brazo con las cabezas de l/e desde su posición actual hasta el cilindro diseccionado.

Tiempo de rotación. El disco debe girar hasta que la cabeza esté situada sobre el sector a leer o escribir..

Tiempo de transferencia: Tiempo empleado en leer o escribir los datos.

# ***Ficheros.***

## **Conceptos básicos:**

### **Discos magnéticos:**

El disco, un cuello de botella: Las prestaciones de los discos están muy lejos de las velocidades de las redes. La CPU tiene que esperar a los datos debido a la lentitud de los discos. Algunas técnicas para resolver estos problemas:

- Multiprogramación. La CPU hace otras tareas mientras espera.

- Stripping: El fichero se divide en trozos y se reparte entre varios discos.

- Discos RAM: Se tiene RAM configurada para actuar como un disco.

- Caché de discos: Se tiene RAM configurada para tener páginas de datos (sectores o bloques) del disco.

# ***Ficheros.***

## **Conceptos básicos:**

### **Registros.**

Colección de valores que corresponden a cada uno de los campos.

### **Ficheros.**

Colección de registros.

Los registros suelen ser del mismo tipo.

### **Bloqueo.**

Bloque (p.e 80) > Registro (p.e 20)  $\Rightarrow fb = B \div R \rightarrow 4$

Pérdida =  $B - (fb * R) = 80 - (4 * 20)$

Bloque = Registro.

Bloque < Registro  $\Rightarrow$  Registro se reparte en más de un bloque.

Organización extendida

# ***Ficheros.***

## **Ficheros**

**Colocación de los bloques en disco.**

**Continua: bloques del fichero en bloques consecutivos del disco.**

**Enlazada: cada bloque tiene un puntero al siguiente.**

**Mixta: grupos de bloques consecutivos enlazados.**

**Indexada: Bloques de índices con punteros a los bloques del fichero.**

# ***Ficheros.***

## **Ficheros**

**Cabecera de un fichero.**

**Descriptor de la estructura del fichero que necesitan los programas que lo manipulan: direcciones en el disco de los bloques del fichero; formato del registro: longitudes de los campos y orden, etc.**

# ***Ficheros.***

## **Ficheros**

### **Operaciones sobre ficheros:**

**Recuperación.**

**Actualización.**

### **Organización de ficheros.**

**Forma de colocar los datos de un fichero en registros, bloques y estructuras de acceso.**

### **Métodos de acceso a un fichero:**

**Diferentes formas de realizar las operaciones de recuperación y actualización sobre los ficheros.**

# ***Ficheros.***

## **Organización básica de ficheros.**

### **Ficheros desordenados.**

- ◆ **Forma más simple de organización**
- ◆ **Los registros se guardan en el orden de inserción.**
- ◆ **Se suele usar con caminos de acceso adicionales (índices)**
- ◆ **Operaciones:**
  - ◆ **Inserción:**
    - ◆ **Colocar el último bloque en el búfer.**
    - ◆ **Añadir detrás del último registro el nuevo.**
    - ◆ **Escribir en disco.**
  - ◆ **Búsqueda: secuencial.**
  - ◆ **Borrado:**
    - ◆ **Copiar el bloque en memoria.**
    - ◆ **Eliminar el registro o marcado como borrado.**
    - ◆ **Escribir en disco.**

# ***Ficheros.***

## **Organización básica de ficheros.**

### **Ficheros desordenados.**

- ◆ **Forma más simple de organización**
- ◆ **Los registros se guardan en el orden de inserción.**
- ◆ **Se suele usar con caminos de acceso adicionales (índices)**
- ◆ **Operaciones:**
  - ◆ **Inserción:**
    - ◆ **Colocar el último bloque en el búfer.**
    - ◆ **Añadir detrás del último registro el nuevo.**
    - ◆ **Escribir en disco.**
  - ◆ **Búsqueda: secuencial.**
  - ◆ **Borrado:**
    - ◆ **Copiar el bloque en memoria.**
    - ◆ **Eliminar el registro o marcado como borrado.**
    - ◆ **Escribir en disco.**



# ***Ficheros.***

## **Organización básica de ficheros.**

### **Ficheros ordenados.**

- ◆ Los registros del fichero se ordenan según el valor de alguno de sus campos.
- ◆ Lectura de los registros en el orden del campo de ordenación es muy eficiente.
- ◆ La búsqueda de un registro por el campo de ordenación es muy rápida.
- ◆ La búsqueda de registros por campos que no sean el de ordenación es muy ineficiente.
- ◆ Inserción y borrado son costosas al ser necesario una expansión o contracción del fichero.

# ***Ficheros.***

## **Organización básica de ficheros.**

### **Ficheros dispersos.**

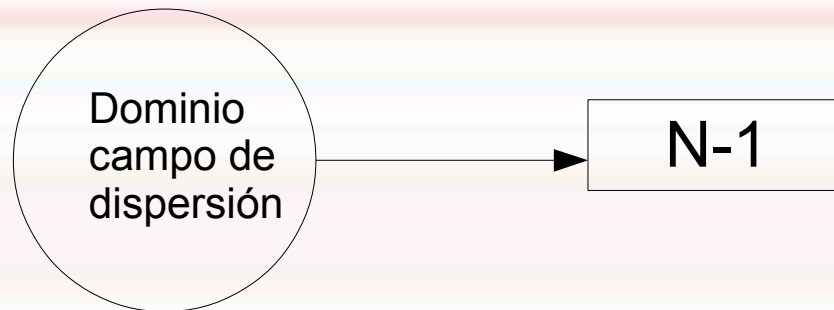
- ◆ Proporciona un acceso rápido para búsquedas por un campo clave.
- ◆ El campo clave se denomina campo de dispersión.
- ◆ Dispersión:
  - ◆ Elegido un campo del fichero  $c$  (usualmente la clave)
  - ◆ Se proporciona una función de dispersión  $d$ .
  - ◆  $d(c)$  = dirección del cubo (un bloque o conjunto de bloques continuos) en el que se va a guardar el registro..

# ***Ficheros.***

## **Organización básica de ficheros.**

### **Ficheros dispersos.**

**d:**



**Función de dispersión típica:  $d(k) = K \bmod N$**

### **Operaciones:**

- ◆ **Inserciones: Problemas de las colisiones.**
- ◆ **Borrados.**
- ◆ **Modificaciones..**

# ***Ficheros.***

## **Indexación y asociación.**

### **Conceptos básicos:**

- ◆ Un índice para un archivo del sistema funciona como el índice de un libro.
- ◆ Las palabras de índice están ordenadas, lo que hace fácil la búsqueda del término que se esté buscando.
- ◆ Además, el índice es mucho más pequeño que el libro, con lo que se reduce aún más el esfuerzo necesario para encontrar las palabras en cuestión.

# ***Ficheros.***

## **Indexación y asociación.**

### **Tipos de índices:**

- ◆ Índices ordenados. Estos índices están basados en una disposición ordenada de los valores.
- ◆ Índices asociativos (*hash indices*). Estos índices están basados en una distribución uniforme de los valores a través de una serie de cajones (*buckets*). El valor asignado a cada cajón está determinado por una función, llamada *función de asociación* (*hash function*).

# ***Ficheros.***

## **Indexación y asociación.**

### **Tipos de índices:**

- ◆ Índices ordenados. Estos índices están basados en una disposición ordenada de los valores.
- ◆ Índices asociativos (*hash indices*). Estos índices están basados en una distribución uniforme de los valores a través de una serie de cajones (*buckets*). El valor asignado a cada cajón está determinado por una función, llamada *función de asociación* (*hash function*).

# ***Ficheros.***

## **Indexación y asociación.**

### **Criterios para elegir una técnica de indexación:**

Se considerarán varias técnicas de indexación y asociación. Ninguna de ellas es la mejor. Sin embargo, cada técnica es la más apropiada para una aplicación específica de bases de datos. Cada técnica debe ser valorada según los siguientes criterios:

# ***Ficheros.***

## **Indexación y asociación.**

### **Criterios para elegir una técnica de indexación:**

Se considerarán varias técnicas de indexación y asociación. Ninguna de ellas es la mejor. Sin embargo, cada técnica es la más apropiada para una aplicación específica de bases de datos. Cada técnica debe ser valorada según los siguientes criterios:

- ◆ Tipos de acceso.  
Los tipos de acceso que se soportan eficazmente. Estos tipos podrían incluir la búsqueda de registros con un valor concreto en un atributo, o buscar los registros cuyos atributos contengan valores en un rango especificado.



# ***Ficheros.***

## **Indexación y asociación.**

### **Criterios para elegir una técnica de indexación:**

- ◆ **Tiempo de acceso.**  
El tiempo que se tarda en buscar un determinado elemento de datos, o conjunto de elementos, usando la técnica en cuestión.
- ◆ **Tiempo de inserción.**  
El tiempo empleado en insertar un nuevo elemento de datos. Este valor incluye el tiempo utilizado en buscar el lugar apropiado donde insertar el nuevo elemento de datos, así como el tiempo empleado en actualizar la estructura del índice.

# ***Ficheros.***

## **Indexación y asociación.**

### **Criterios para elegir una técnica de indexación:**

- ◆ **Tiempo de borrado.**  
Tiempo empleado en borrar un elemento. Este valor incluye el tiempo utilizado en buscar el elemento a borrar, así como el tiempo empleado en actualizar la estructura del índice.
- ◆ **Espacio adicional requerido.**  
El espacio adicional ocupado por la estructura del índice. Como normalmente la cantidad necesaria de espacio adicional suele ser moderada, es razonable sacrificar el espacio para alcanzar un rendimiento mejor.

# ***Ficheros.***

## **Indexación y asociación.**

### **Ejemplo: “Gestión biblioteca”**

A menudo se desea tener más de un índice por archivo.

La mayoría de las bibliotecas mantienen varios catálogos de fichas: por autor, por materia y por título.

Claves de búsqueda: Atributos o conjunto de atributos usados para buscar en un archivo. Hay que observar que esta definición de *clave* difiere de la usada en *clave primaria*, *clave candidata utilizadas para la identificación de un registro*. Este doble significado de *clave* está (por desgracia) muy extendido en la práctica. Usando nuestro concepto de clave de búsqueda vemos que, si hay varios índices en un archivo, existirán varias claves de búsqueda.

# ***Ficheros.***

## **Indexación y asociación.**

Una estructura de índice permite un acceso directo rápido a los registros de un archivo.

Cada estructura de índice está asociada con una clave de búsqueda concreta.

En el ejemplo de la biblioteca un índice almacena de manera ordenada los valores de las claves de búsqueda, y asocia a cada clave los registros que contienen esa clave de búsqueda.

Los registros en el archivo indexado pueden estar a su vez almacenados siguiendo un orden, semejante a como los libros están ordenados en una biblioteca por algún atributo como el número decimal Dewey (también llamado CDD, es un sistema de clasificación de bibliotecas).

Un archivo puede tener varios índices según diferentes claves de búsqueda.

# ***Ficheros.***

## **Indexación y asociación.**

### **Índice Primario.**

Es el índice cuya clave de búsqueda especifica el orden secuencial del archivo que contiene los registros.

Se emplea algunas veces para hacer alusión a un índice según una clave primaria. Sin embargo, tal uso no es normal y debería evitarse.)  
Los índices primarios también se llaman índices de agrupación (clustering índices).

La clave de búsqueda de un índice primario es normalmente la clave primaria, aunque no es así necesariamente.

**Los índices cuyas claves de búsqueda**

especifican un orden diferente del orden secuencial del archivo se llaman **índices secundarios** o **índices sin**

# ***Ficheros.***

## **Indexación y asociación.**

### **Indices secundarios.**

Son índices cuyas claves de búsqueda especifican un orden diferente del orden secuencial del archivo.

Se llaman también índices sin agrupación o non clustering indices

# ***Ficheros.***

## **Indexación y asociación.**

### **Ejemplos.**

Índice primario

los archivos están ordenados secuencialmente según alguna clave de búsqueda.

Estos archivos con índice primario según una clave de búsqueda se llaman **archivos secuenciales indexados**.

Representan uno de los esquemas de índices más antiguos usados por los sistemas de bases de datos.

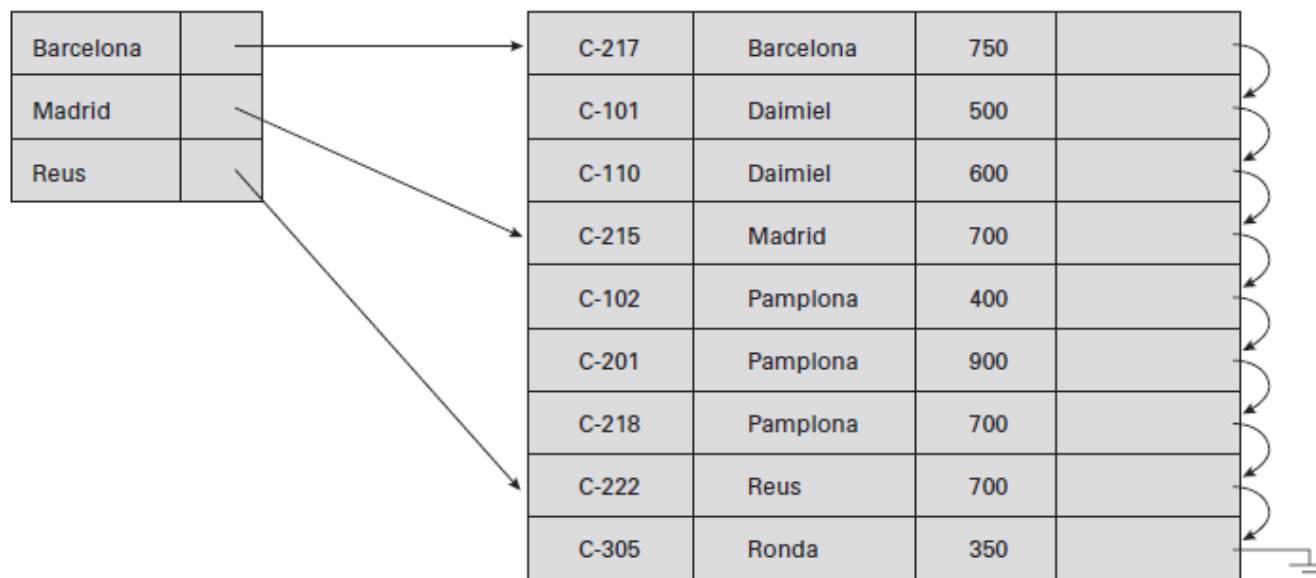
Se emplean en aquellas aplicaciones que demandan un procesamiento secuencial del archivo completo así como un acceso directo a sus registros.

# Ficheros.

## Indexación y asociación.

Ejemplo gestión bancaria.

Los registros están almacenados según el orden de la clave de búsqueda, siendo esta clave *nombre-sucursal*.





# ***Ficheros.***

## **Indexación y asociación.**

### **Índices densos y dispersos.**

Un **registro índice** o **entrada del índice** consiste en un valor de la clave de búsqueda y punteros a uno o más registros con ese valor de la clave de búsqueda. El puntero a un registro consiste en el identificador de un bloque de disco y un desplazamiento en el bloque de disco para identificar el registro dentro del bloque.

Hay dos clases de índices ordenados:

- ◆ Índice denso.
- ◆ Índice disperso.

# ***Ficheros.***

## **Indexación y asociación.**

- **Índice denso.**

Aparece un registro índice por cada valor de la clave de búsqueda en el archivo.

El registro índice contiene el valor de la clave y un puntero al primer registro con ese valor de la clave de búsqueda. El resto de registros con el mismo valor de la clave de búsqueda se almacenan consecutivamente después del primer registro, dado que, ya que el índice es primario, los registros se ordenan sobre la misma clave de búsqueda.

Las implementaciones de índices densos pueden almacenar una lista de punteros a todos los registros con el mismo valor de la clave de búsqueda; esto no es esencial para los índices primarios.

# ***Ficheros.***

## **Indexación y asociación.**

- **Índice disperso.**

Sólo se crea un registro índice para algunos de los valores.

Al igual que en los índices densos, cada registro índice contiene un valor de la clave de búsqueda y un puntero al primer registro con ese valor de la clave.

Para localizar un registro se busca la entrada del índice con el valor más grande que sea menor o igual que el valor que se está buscando. Se empieza por el registro apuntado por esa entrada del índice y se continúa con los punteros del archivo hasta encontrar el registro deseado..

# Ficheros.

## Indexación y asociación.

### Ejemplos

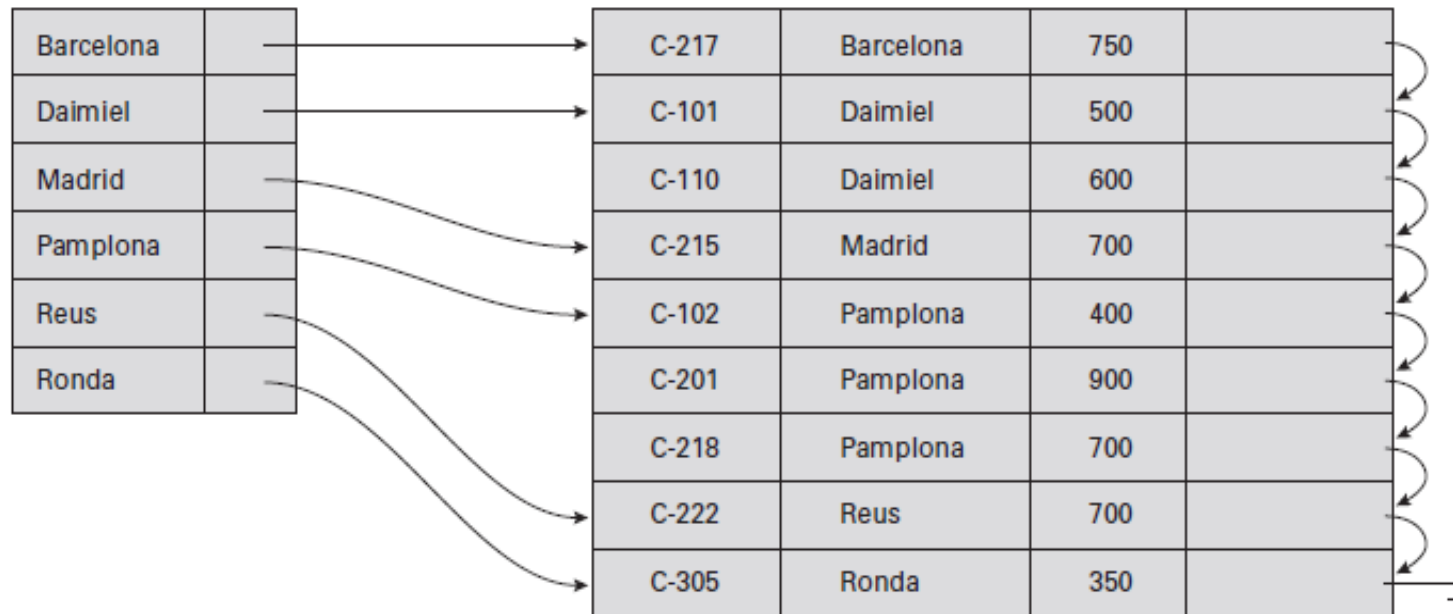


FIGURA 12.2. Índice denso.

# *Ficheros.*

## Indexación y asociación.

### Ejemplos

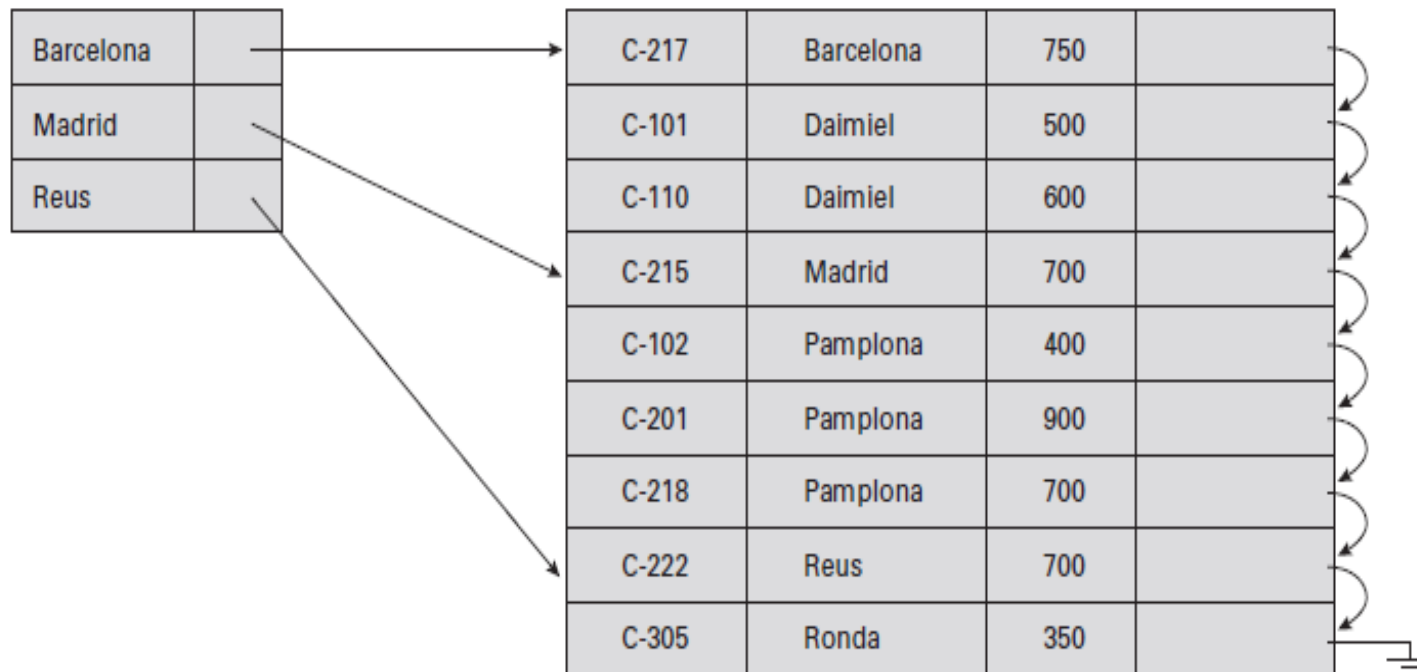


FIGURA 12.3. Índice disperso.

# ***Ficheros.***

## **Indexación y asociación.**

Índices multinivel.

Son índices con dos o más niveles de direccionamiento.

La búsqueda de registros usando un índice multinivel necesita claramente menos operaciones de E/S que las que se emplean en la búsqueda de registros con la búsqueda binaria.

Cada nivel de índice se podría corresponder con una unidad del almacenamiento físico. Así, podríamos tener índices a nivel de pista, cilindro o disco.

Un diccionario normal es un ejemplo de un índice multinivel en el mundo ajeno a las bases de datos. La cabecera de cada página lista la primera palabra en el orden alfabético en esa página. Este índice es multinivel: las palabras en la parte superior de la página del índice del libro forman un índice disperso sobre los contenidos de las páginas del diccionario.

# *Ficheros.*

## Indexación y asociación.

Ejemplo.

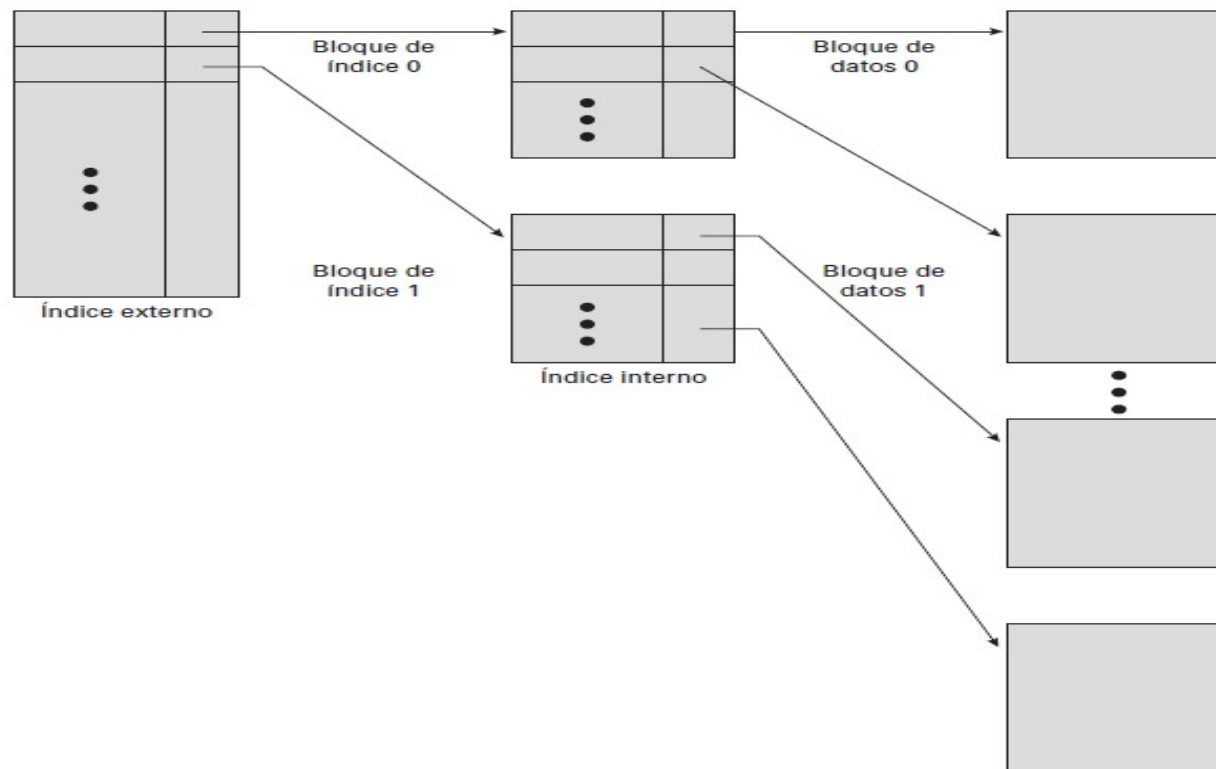


FIGURA 12.4. Índice disperso de dos niveles.

# *Ficheros.*

## Indexación y asociación.

Ejemplo.

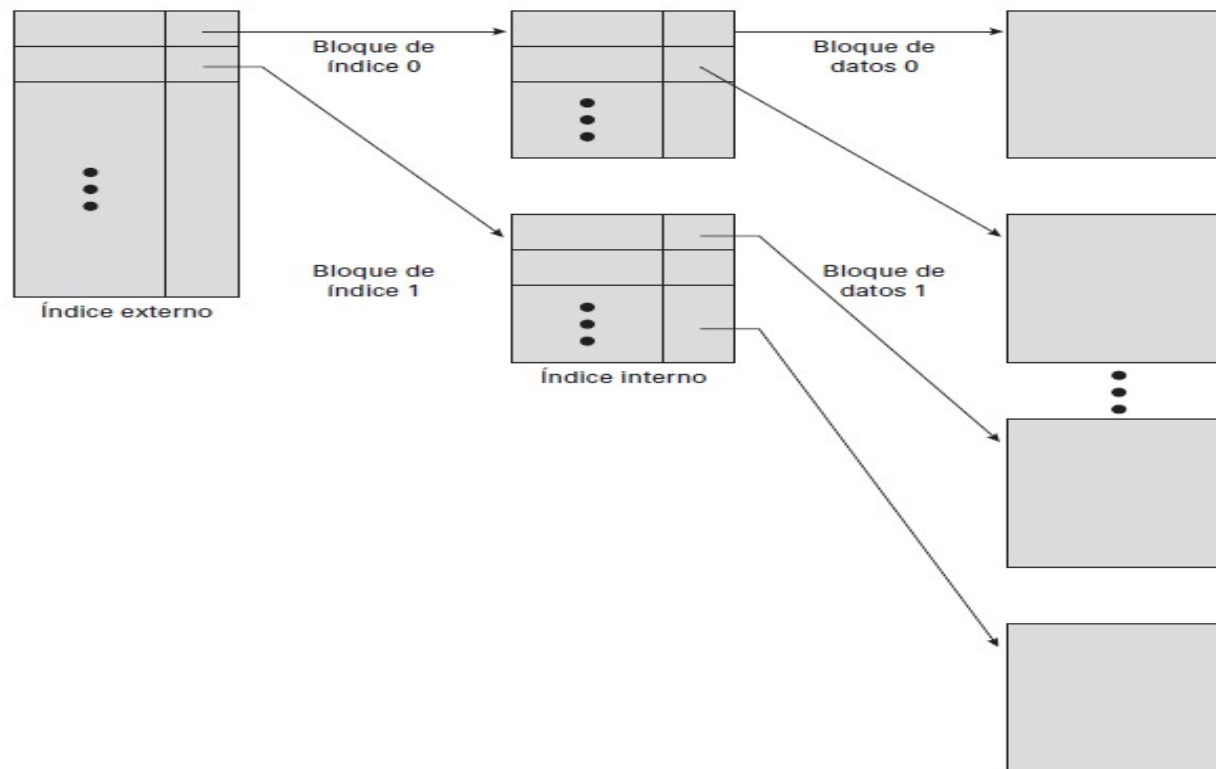


FIGURA 12.4. Índice disperso de dos niveles.



# Ficheros.

## Indexación y asociación.

Ejemplo.

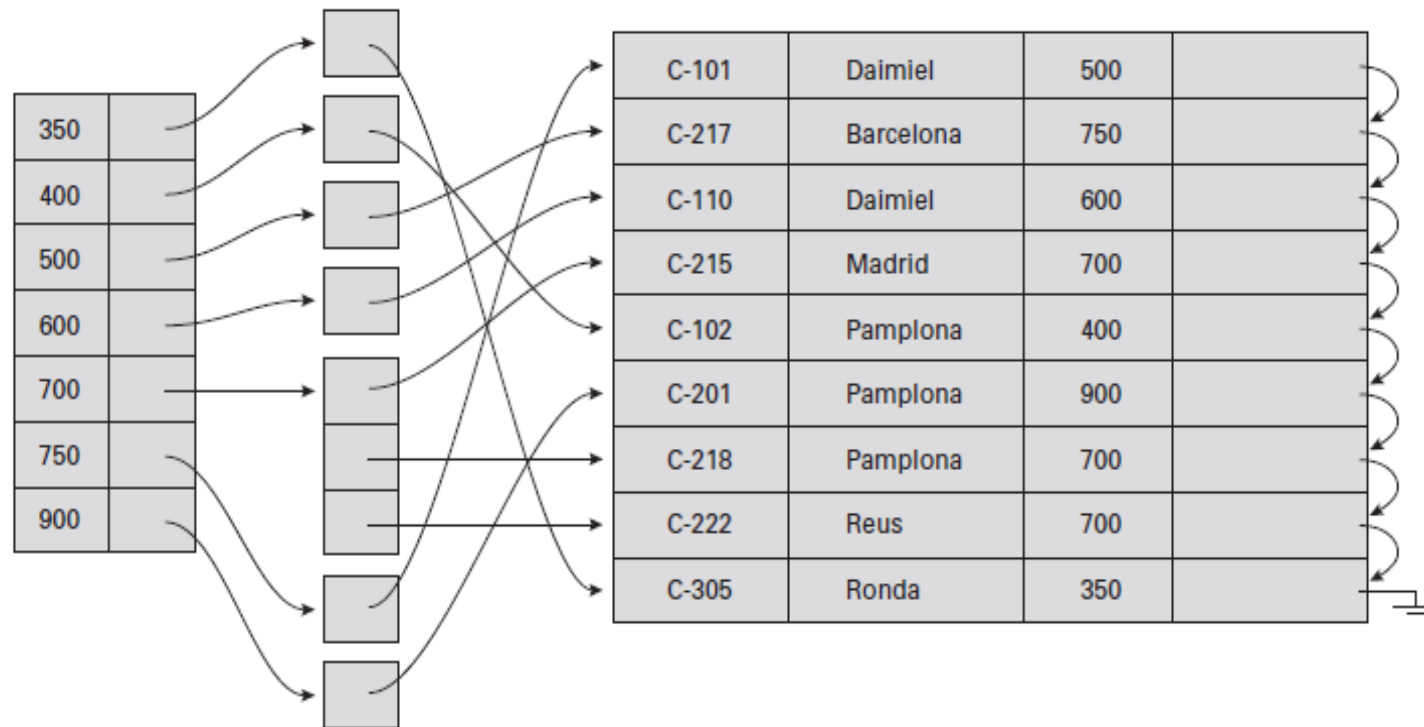


FIGURA 12.5. Índice secundario del archivo *cuenta*, con la clave no candidata *saldo*.

# ***Ficheros.***

## **Indexación y asociación.**

### Árboles B,B+

El inconveniente principal de la organización de un archivo secuencial indexado reside en que el rendimiento, tanto para buscar en el índice como para buscar secuencialmente a través de los datos, se degrada según crece el archivo. Aunque esta degradación se puede remediar reorganizando el archivo, el rendimiento de tales reorganizaciones no es deseable.

Un índice de árbol B+ es un índice multinivel pero con una estructura que difiere del índice multinivel de un archivo secuencial

# Ficheros.

## Indexación y asociación.

### Árboles B,B+

La estructura típica de un árbol B+ puede contener hasta  $n - 1$  claves de búsqueda  $K_1, K_2, \dots, K_{n-1}$  y  $n$  punteros  $P_1, P_2, \dots, P_n$ . Los valores de la clave de búsqueda de un nodo se mantienen ordenados; así, si  $i < j$ , entonces  $K_i < K_j$ .



FIGURA 12.6. Nodo típico de un árbol B+.

# Ficheros.

## Indexación y asociación.

### Árboles B,B+

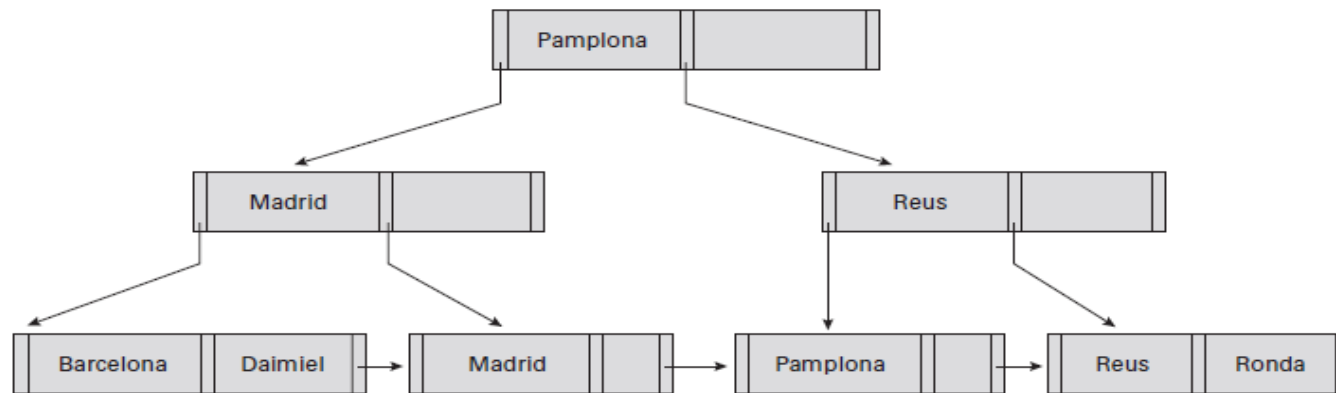


FIGURA 12.8. Árbol B<sup>+</sup> para el archivo *cuenta* ( $n = 3$ ).

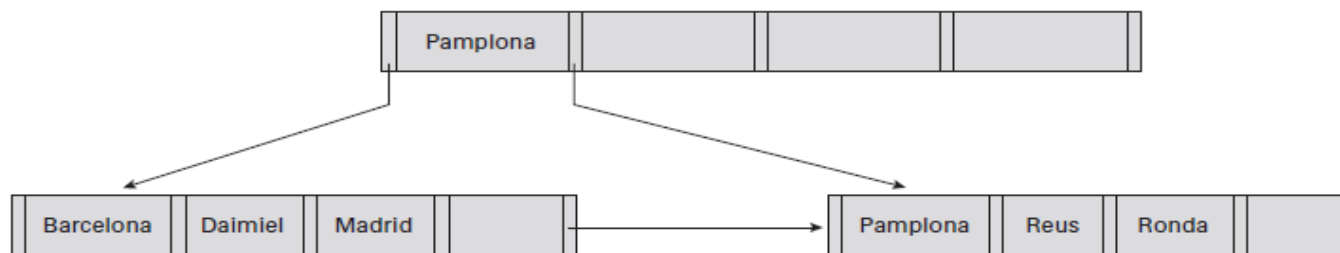


FIGURA 12.9. Árbol B<sup>+</sup> para el archivo *cuenta* ( $n = 5$ ).

# Ficheros.

## Indexación y asociación.

### Árboles B,B+

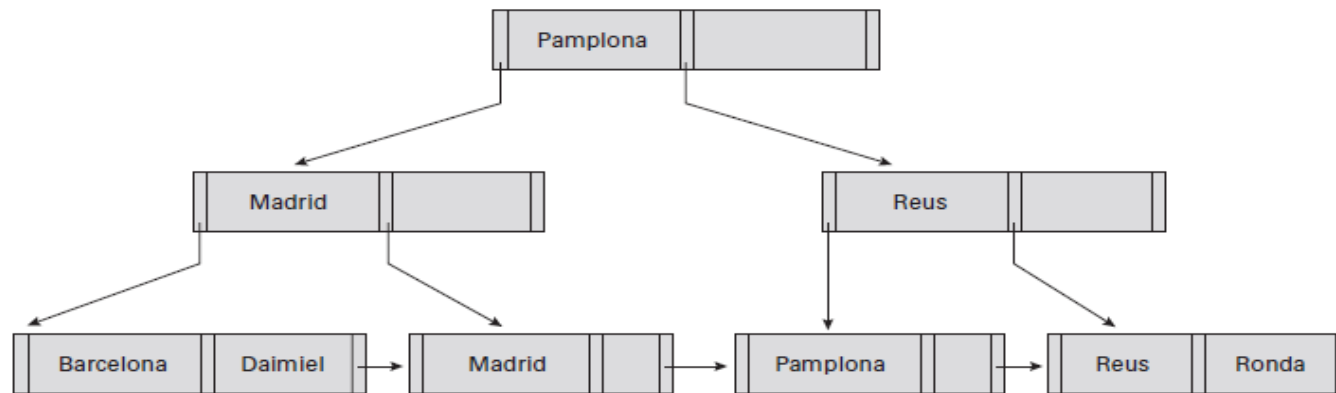


FIGURA 12.8. Árbol B<sup>+</sup> para el archivo *cuenta* ( $n = 3$ ).

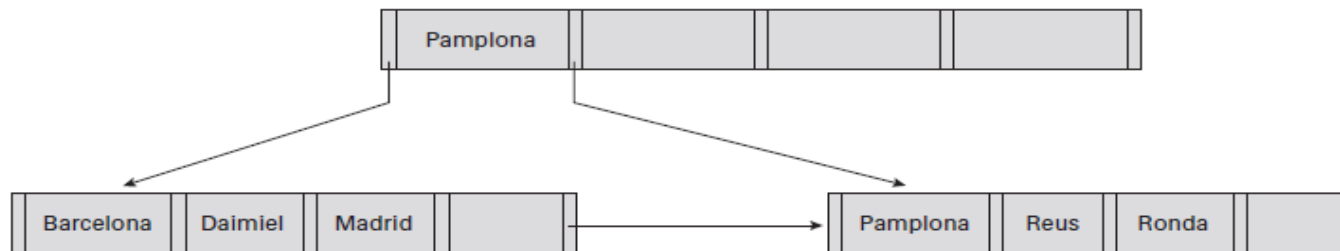


FIGURA 12.9. Árbol B<sup>+</sup> para el archivo *cuenta* ( $n = 5$ ).