

# 1.- Sistemas de almacenamiento

## 1.1- Ficheros

### 1.1.1- ¿Qué es un fichero?

En la década de los setenta, los procesos básicos que se llevaban a cabo en una empresa se centraban en cuestiones relacionadas con contabilidad y facturación. Las necesidades de almacenamiento y gestión de información podían satisfacerse utilizando un número relativamente reducido de archivos en papel agrupados y ordenados, los típicos ficheros clásicos.

Al llevar a cabo una primera informatización, se pasó de tener los datos en formato papel a poder acceder a ellos de manera mucho más rápida a través del ordenador. En ese momento, la informática adaptó sus herramientas para que los elementos que el usuario maneja en el ordenador se parezcan a los que utilizaba manualmente. Así en informática se sigue hablando de ficheros, formularios, carpetas, directorios,...

La información debía ser trasladada desde el papel al formato digital y por lo general, era necesario almacenarla para su posterior recuperación, consulta y procesamiento. De este modo, para llevar a cabo un tratamiento eficiente de ésta era necesario establecer métodos adecuados para su almacenamiento. El elemento que permitió llevar a cabo el almacenamiento de datos de forma permanente en dispositivos de memoria masiva fue **el fichero o archivo**.

**Fichero o archivo:** unidad lógica de almacenamiento, formada por un conjunto de información relacionada y organizada de forma estructurada. Es una secuencia de dígitos binarios que organiza información relacionada con un mismo aspecto.

Los ficheros están formados por **registros lógicos** que contienen datos relativos a un mismo elemento u objeto (por ejemplo, los datos de usuarios de una plataforma educativa). A su vez, los registros están divididos en campos que contienen cada una de las informaciones elementales que forman un registro (por ejemplo, el nombre del usuario o su dirección de correo electrónico).

Registro lógico:

id1	cust1	prod1	store1	price1	date1	qty1
-----	-------	-------	--------	--------	-------	------

Fichero:

id1	cust1	prod1	store1	price1	date1	qty1
-----	-------	-------	--------	--------	-------	------

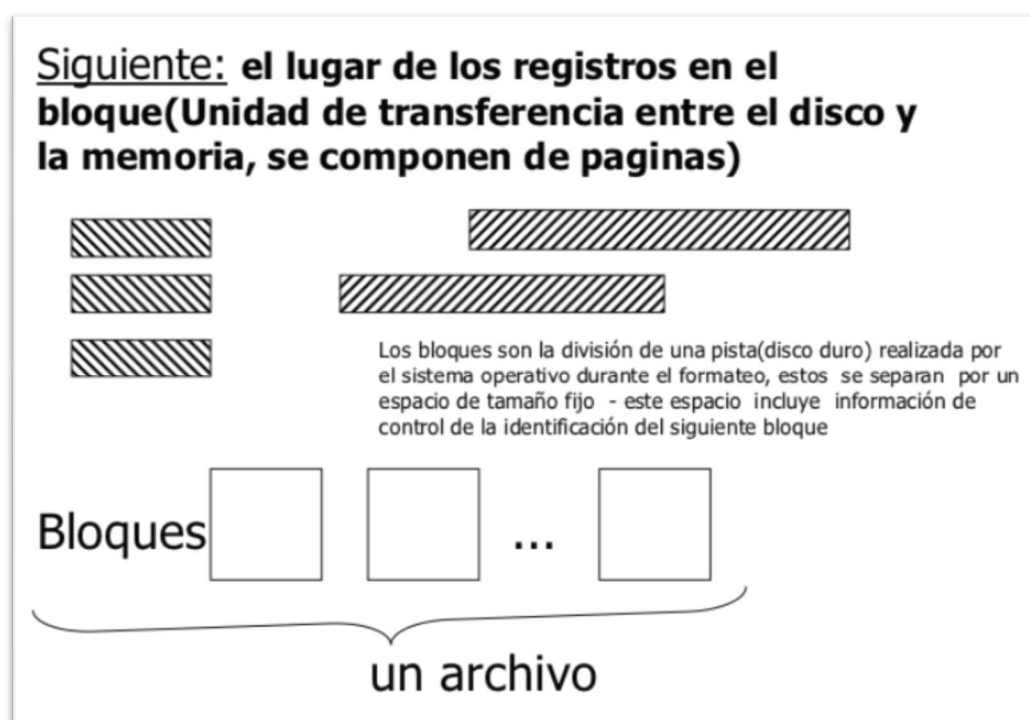
id2	cust2	prod2	store2	price2	date2	qty2
-----	-------	-------	--------	--------	-------	------

id3	cust3	prod3	store3	price3	date3	qty3
-----	-------	-------	--------	--------	-------	------

Los ficheros o archivos se identifican por su nombre y extensión. Según el sistema operativo que los utiliza tienen unas características u otras. Ej. Msdos 8 caracteres como max y luego reemplaza por ~1. Linux cualquier tamaño de nombre. Windows no es case\_sensitive mientras Linux sí.

Hemos de resaltar que los datos están almacenados de tal forma que se puedan añadir, suprimir, actualizar o consultar individualmente en cualquier momento.

Como los ficheros suelen ser muy voluminosos, solo se pueden llevar a la memoria principal partes de ellos para poder procesarlos. La cantidad de información que es transferida entre el soporte en el que se almacena el fichero, y la memoria principal del ordenador, en una sola operación de lectura/grabación, recibe el nombre de **registro físico** o **bloque**.



Normalmente en cada operación de lectura/grabación se transfieren varios registros del fichero, es decir un bloque suele contener varios registros lógicos. Al número de registros que entran en un bloque se le conoce con el nombre de **factor de blocaje**, y a esta operación de agrupar varios registros en un bloque se le llama **bloqueo de registros**.

## (1) Separacion de registros

Bloque



(a) No necesitan ser separados – registros tamaño fijo.

(b) Marcado especial

(c) Dar la longitud de registro (or offsets)

- con cada registro

- dentro del encabezado del bloque

### 1.1.2.- Diseño de registros.

Como hemos visto, un fichero es un conjunto de registros, y un registro está compuesto, a su vez, por un conjunto de campos. Teniendo en cuenta esto en el diseño de un fichero tiene una gran importancia el diseño de los registros que lo forman, además de la selección del soporte de datos más adecuado para almacenarlo y la forma en que se organizan los datos dentro del soporte.

Si observarnos las fichas del archivador de empleados, podemos ver que:

- ❑ La información referida a cada empleado está toda en una ficha (registro).
- ❑ Sobre cada ficha las informaciones (campos) aparecen en un orden, que es el mismo para todas (estructura del registro).
- ❑ Las fichas están ordenadas por algún criterio, para facilitar la búsqueda de una ficha determinada. Por ejemplo, por los apellidos, o por el DNI (campo clave).

Teniendo en cuenta todo esto, el **diseño de un registro** consiste en seleccionar, ordenar y definir las características de los campos que lo forman.

id1	cust1	prod1	store1	price1	date1	qty1
-----	-------	-------	--------	--------	-------	------

Normalmente para representar el diseño de los utilizan unas hojas de diseño, en las que para cada campo se definen:

1. Un **nombre** que nos permite identificar dicho campo en los programas.
2. La **posición** del campo dentro del registro.
3. El **tipo de datos** que va a contener (numérico, alfabético, lógico, ...)
4. El **tamaño** del campo en caracteres (sí el campo es numérico con decimales, debe indicarse el número de decimales que tiene).
5. La **descripción del contenido** del campo, si fuera necesario.

Una vez que tenemos claro los campos que van a formar el registro, el mayor problema que se nos plantea es el seleccionar el tamaño de cada campo, debido a que la información que se va a guardar en un campo no es la misma para todos los registros. Por ejemplo, el campo APELLIDOS del ejemplo guarda los apellidos de los empleados y el tamaño de estos no son iguales para todos los empleados, podemos tener desde un DIAZ DIAZ, hasta un RODRIGUEZ DE LA VILLAJOSYOSA GARCIA-DOMEQ.

Para solucionar este problema pueden adoptarse varias soluciones:

- Definir el campo con el mayor tamaño posible, lo que puede suponer un despilfarro de soporte si un registro valor muy largo al resto.
- Definir el campo con un tamaño intermedio y abreviar los valores que sobrepasen ese tamaño. Esta solución es la más aplicada, y se utilizará siempre que se pueda abreviar la información que almacena el campo.
- Utilizar campos con longitud variable para cada registro. Esta solución sería la que mejora aprovecha el soporte, pero su manejo es complicado para el programador que tendría que utilizar técnicas que le permitan delimitar el comienzo y el final de cada campo de un registro.
- Emplear un sistema de codificación de la información.

## 1.1.3- Tipos de ficheros.

Según la función que vaya a desempeñar los ficheros, éstos pueden ser clasificados de varias maneras. En la siguiente imagen puedes observar una posible clasificación.

- a. **Ficheros permanentes:** contienen información relevante para una aplicación. Es decir, los datos necesarios para el funcionamiento de ésta. Tienen un periodo de permanencia en el sistema amplio. Estos se subdividen en:
  - o **Ficheros maestros:** contienen el estado actual de los datos que pueden modificarse desde la aplicación. Es la parte central de la aplicación, su núcleo. Podría ser un archivo con los datos de los usuarios de una plataforma educativa.
  - o **Ficheros constantes:** son aquellos que incluyen datos fijos para la aplicación. No suelen ser modificados y se accede a ellos para realización de consultas. Podría ser un archivo con códigos postales.
  - o **Ficheros históricos:** contienen datos que fueron considerados como actuales en un periodo o situación anterior. Se utilizan para la reconstrucción de situaciones. Podría ser un archivo con los usuarios que han sido dados de baja en la plataforma educativa.
  - o **Ficheros de parametrización:** Contienen información que al ser leída por los programas influirá en su funcionamiento. Estos ficheros se utilizan para poder adaptar programas o aplicaciones estándar a las necesidades de cada usuario.

Ejemplo: Fichero que almacena el nombre de la empresa, formato de los informes, etc

- b. **Ficheros temporales:** Se utilizan para almacenar información útil para una parte de la aplicación, no para toda ella. Son generados a partir de datos de ficheros permanentes. Tienen un corto periodo de existencia. Estos se subdividen en:
- **Ficheros de movimientos:** Estos ficheros también se les llama ficheros de transacciones porque se utilizan para actualizar los ficheros permanentes, almacenando en cada registro una operación que supone el cambio de alguno de los datos que contiene el fichero permanente. Sus registros tienen que tener al menos un campo en común con el fichero que van a actualizar que es el que va permitir conocer cuál es el registro que hay que modificar.

La vida de estos ficheros suele ser corta ya que cuando se produce la actualización del fichero permanente, se destruyen, o se guardan como un histórico

- **Ficheros intermedios:** almacenan resultados de una aplicación que serán utilizados por otra.
- **Ficheros de maniobras:** almacenan datos de una aplicación que no pueden ser mantenidos en memoria principal por falta de espacio.
- **Ficheros de resultados:** almacenan datos que van a ser transferidos a un dispositivo de salida.

## 1.1.4- Operaciones con ficheros

Las operaciones que se pueden hacer con los ficheros pueden utilizar todos los registros del fichero o solo una parte de ellos. De entre las operaciones que utilizan todos los registros de los ficheros podemos citar las operaciones de:

- **Creación** - Consiste en la grabación, por primera vez, sobre un soporte de los registros de un fichero.
- **Apertura y cierre** - Para poder hacer cualquier operación con los registros de un fichero tiene que estar abierto. En el tiempo que no se utilizan los datos que almacena el fichero debe permanecer cerrado para evitar que se deteriore la información que almacena. Para empezar a trabajar con los datos de un fichero la primera operación que tenemos que hacer es abrirlo, y cuando terminemos de trabajar con él tenemos que cerrarlo.
- **Borrado** - Consiste en la eliminación de todo el fichero. Esta operación puede realizarse de dos formas: prohibiendo de un modo definitivo el acceso al fichero, de modo que no se pueda leer o escribir en él, o bien, destruyendo la información referente al fichero que está grabada en el soporte que lo contiene. En el primer caso, el fichero puede ser recuperado.
- **Ordenación o clasificación** - Consiste en cambiar el orden en que están grabados los registros del fichero en el soporte. Los registros se ordenan según el contenido de uno o más campos en forma ascendente o descendente. En el primer caso el primer registro es el que tiene el menor valor en el campo utilizado para clasificar los registros, y en el segundo caso, el primer registro contendrá, en el campo utilizado para clasificar los registros, el valor mas alto de todos los registros del fichero.

- ❑ **Duplicado o copiado** - Esta operación consiste en crear un nuevo fichero idéntico a uno ya existente. Esta operación es muy recomendable realizarla como medida de seguridad, por si algún día se nos estropea un fichero. Si esto ocurriera siempre tendríamos la copia para poder recuperarlo.
- ❑ **Fusión o intercalación** - Se denomina también mezcla. Consiste en obtener, de dos ficheros ordenados por un mismo campo y con la misma estructura, otro fichero que contenga todos los registros de ambos y que se mantenga ordenado por el mismo campo.
- ❑ **Partición** - Consiste en dividir un fichero en dos o más, de acuerdo con alguna condición que han de cumplir los registros.

## 1.1.5.- Medidas de utilización de ficheros

Para diseñar un fichero es necesario, en primer lugar, seleccionar y definir los datos que va a contener y, a continuación seleccionar el soporte en que va a estar grabado y la forma de organizar los datos del fichero en él. El primer paso consiste en hacer el diseño del registro.

En este apartado veremos algunas medidas que nos pueden servir de ayuda para decidir que soporte tendremos que utilizar y cual es la organización más adecuada, teniendo en cuenta las operaciones que se van a realizar con mas frecuencia.

**Volumen:** Es la cantidad de memoria necesaria para almacenar los datos del fichero, es decir, el espacio que ocupa el fichero en el soporte, medido en caracteres u octetos (bytes). Se puede calcular conociendo el número de registros que va a tener el fichero y el tamaño de cada registro, por término medio.

$$V = NRT \times LR$$

V - Volumen del fichero

NRT - N° total de registros del fichero

LR - Longitud media, en bytes, del registro

**Crecimiento:** Es una medida del aumento del volumen del fichero. Se puede medir conociendo el n° de registros que se aumenta en cada tratamiento. Se suele expresar en tanto por ciento:

$$Tasa\ de\ crecimiento\ (\%) = (NRA / NRT) \times 100$$

NRA – N° de registros que se aumentan

**Actividad:** Es el número de registros consultados o modificados, expresado en tanto por ciento, con respecto al número total de registros del fichero. Si este porcentaje se determina por cada tratamiento del fichero, se denomina *tasa de consulta o modificación*. Si el porcentaje se calcula para un período de tiempo determinado, se llama *frecuencia de consulta o modificación*.

$$Tasa\ de\ actividad\ (\%) = (NRC / NRT) \times 100$$

NRC – N° de registros consultados o modificados

Esta medida suele emplearse como guía para determinar si el fichero debe explotarse de una forma secuencial o directa, en un soporte direccionable.

**Volatilidad:** Es el número de registros dados de alta o de baja respecto al número total de registros del fichero, expresado en tanto por ciento. Si el porcentaje se calcula por cada tratamiento del fichero, se denomina *tasa de renovación*, y si se calcula para un período de tiempo determinado, se llama *frecuencia de renovación*.

$$Tasa\ de\ renovación\ (\%) = NRAB / NRT \times 100$$

NRAB = N° de registros dados de alta o de baja

Si la tasa de renovación es alta se dice que el fichero es volátil y en caso contrario se dice que el fichero es estable o estático.

**Densidad ideal del registro:** Es la ocupación real de un registro sobre el total de espacio reservado para dicho registro.

$$dR\ (\%) = (Caracteres\_ocupados\_reg / Caracteres\_reservados\_reg) \times 100$$

## 1.1.6.- Organización de los Ficheros

La organización de los ficheros es la forma de estructurar y almacenar los datos es una estructura lógica de almacenamiento para su posterior almacenamiento sobre un soporte físico.

### 1.1.6.1 - Ficheros secuenciales.

a) Introducción:

Un fichero con organización secuencial se caracteriza porque sus **registros están almacenados de forma contigua**, de manera, que la única forma de acceder a él, es leyendo un registro tras otro desde el principio hasta el final.

Existe una **correspondencia total entre el orden lógico y el orden físico**, si entendemos por orden lógico el orden en que son dados de alta y recuperados los registros y por orden físico el orden en que están grabados los registros en el soporte.

En los ficheros secuenciales suele haber una marca indicativa del fin del fichero, que suele denominarse **EOF** (End of File). Para detectar el final del fichero sólo es necesario encontrar la marca EOF.

Este tipo de ficheros pueden utilizar dispositivos o soportes no direccionables o de acceso secuencial, como son las cintas magnéticas de almacenamiento de datos. También se utiliza en los CD de audio y los DVD de vídeo, en los que la música o las imágenes se almacenan a lo largo de una espiral continua.

Los registros almacenados se identifican por medio de una información ubicada en uno de sus campos, a este campo se le denomina **clave o llave**. Si se ordena un archivo secuencial por su clave, es más rápido realizar cualquier operación de lectura o escritura.

b) Características relevantes de los ficheros secuenciales son:

- La lectura siempre se realiza hacia delante.
- Son ficheros monousuario, no permiten el acceso simultáneo de varios usuarios.
- Tienen una estructura rígida de campos. Todos los registros deben aparecer en orden, es decir, la posición de los campos de cada registro siempre ha de ser la misma.
- El modo de apertura del fichero, condiciona la lectura o escritura.
- Aprovechan al máximo el soporte de almacenamiento, al no dejar huecos vacíos.
- Se pueden grabar en cualquier tipo de soporte, tanto en secuenciales como direccionables.
- Todos los lenguajes de programación disponen de instrucciones para trabajar con este tipo de ficheros.
- No se pueden insertar registros entre los que ya están grabados.

En el siguiente gráfico se observa la estructura de un fichero secuencial.





#### C) Ventajas de los ficheros secuenciales

- Aprovecha al máximo el soporte, al no dejar huecos entre los registros.
- Rápido acceso al registro siguiente. Por lo que se hace ideal cuando en cada operación de actualización o consulta se van a procesar la mayoría de los registros.
- Se pueden utilizar cualquier tipo de registros: de longitud fija, variable o indefinida.
- Se pueden grabar en cualquier tipo de soporte, tanto en secuenciales como direccionables.
- Todos los lenguajes de programación disponen de instrucciones para trabajar con este tipo de ficheros.

#### D) Inconvenientes de los ficheros secuenciales.

- El único modo de acceso es el acceso secuencial, por lo que para leer el registro que ocupe la posición  $n^{\circ}$  es necesario leer los  $n-1$  registros anteriores. Este tipo de organización no es adecuado para ficheros en los que se necesita procesar registros aislados, es decir que tengan un índice de utilización bajo.
- No se pueden insertar/intercalar registros entre los que ya están grabados. Si tenemos el fichero ordenado por el contenido de un campo y queremos dar de alta un registro, que según ese orden debería ir entre dos registros que ya existen en el fichero, es necesario copiar todo el fichero en uno nuevo, grabando en el nuevo fichero todos los registros, insertando el registro que se quiere dar de alta en la posición que le corresponda. Otra posibilidad es ir dando de alta los registros al final del fichero y a continuación realizar una operación de reordenación de todo el fichero.
- Si el fichero está grabado en un soporte secuencial por ejemplo una cinta magnética, para poder hacer modificaciones o borrado de registros es necesario hacer una copia del fichero en un fichero nuevo. Si el soporte no es secuencial las modificaciones se pueden hacer sobre el mismo registro, y el borrado se puede hacer de forma lógica, es decir, mediante la grabación de una marca en el registro.

## 1.1.6.2 - Ficheros de acceso directo.

### A) Introducción.

La organización directa está basada en la independencia entre el orden en que se dan de alta los registros y la posición en la que se graban en el soporte. La posición en la que se graban los registros está en función de la información que tenga el campo clave del registro.

**Campo clave:** campo que permite identificar y localizar un registro de manera ágil y organizada.

Cada uno de los registros se guarda en una posición física, que dependerá del espacio disponible en memoria masiva, de ahí que la distribución de los registros sea aleatoria dentro del soporte de almacenamiento. Para acceder a la posición física de un registro se utiliza una dirección o índice, no siendo necesario recorrer todo el fichero para encontrar un determinado registro.

A través de una transformación específica aplicada a la clave, se obtendrá la dirección física en la que se encuentra el registro. Según la forma de realizar esta transformación, existen diferentes modos de acceso:



En el acceso directo la clave coincide con la dirección, debiendo ser numérica y comprendida dentro del rango de valores de las direcciones. Es el método más rápido.

La medida básica de posicionamiento del puntero en el fichero es el byte, dependiendo del tipo de codificación de caracteres que empleemos ([Unicode](#), [ANSI](#)) se utilizarán 1 o 2 bytes por carácter respectivamente. Teniendo esto en cuenta, el puntero avanzará de uno en uno o de dos en dos bytes para poder leer o escribir cada carácter.

B) **características** de los ficheros de acceso directo o aleatorio son:

- Posicionamiento inmediato.
- Registros de longitud fija.
- Apertura del fichero en modo mixto, para lectura y escritura.
- Permiten múltiples usuarios utilizándolos.
- Los registros se borran colocando un cero en la posición que ocupan.
- Permiten la utilización de algoritmos de compactación de huecos.
- Los archivos se crean con un tamaño definido, es decir, con un máximo de registros establecido durante la creación.
- Esta organización sólo es posible en soportes direccionables.

- Se usan cuando el acceso a los datos de un registro se hace siempre empleando la misma clave y la velocidad de acceso a un registro es lo que más nos importa.
- Permiten la actualización de los registros en el mismo fichero, sin necesidad de copiar el fichero.
- Permiten realizar procesos de actualización en tiempo real.

### C) Técnicas de direccionamiento

El elegir una buena técnica de direccionamiento es fundamental en este tipo de organización. A la hora de elegir una técnica de direccionamiento debemos pedirle:

- 1) Que sea **fácil de aplicar**. Tenemos que tener en cuenta que en esta organización es el usuario el que se encarga de calcular la posición en la que se van a guardar los registros y no como en los ficheros indexados que era el sistema operativo el que se encargaba de esto.
- 2) Que deje el **menor número de huecos** posible. Se debe procurar que el método seleccionado nos proporcione valores entre 1 y  $n^\circ$ , siendo  $n^\circ$  el número de celdas que tiene el fichero, sin dejar celdas vacías.
- 3) Que las claves de registros diferentes nos den direcciones diferentes. Cuando dos o más claves diferentes proporcionan, al aplicar la técnica de direccionamiento, la misma dirección se dice que produce *sinónimos*. También se dice que esos registros *colisionan*. Cuando las claves de dos registros dieron lugar a un sinónimo (indican la misma celda), sólo uno de ellos puede ser almacenado en esa celda y tenemos que tener previsto algún procedimiento para calcular la posición en la que se tiene que grabar el otro registro. Los métodos más empleados para el tratamiento de sinónimos son los siguientes:
  - a) Almacenar el registro que colisiona en el **primer hueco libre** que se encuentre a partir de la dirección que le corresponde, y que está ya ocupada. A este método se le conoce como búsqueda lineal o direccionamiento abierto.
  - b) Creando una zona especial del fichero en la que se grabarán todos los registros que dieron lugar a un sinónimo. A esta zona del fichero se le suele llamar **área de desbordamiento (overflow)** o de excedentes. Este es el sistema más empleado.

**D) *Ventajas de la organización:***

1. Al emplear un modo de acceso directo son los ficheros que permiten acceder de la forma más rápida posible a un registro cualquiera del fichero. Ideal para los procesos en los que el tiempo de acceso a los registros es lo más importante, siempre y cuando el acceso se haga siempre por el campo clave.  
El caso más favorable para utilizar la organización directa es cuando las claves son numéricas y permiten utilizar el direccionamiento directo.
2. Permite la actualización de los registros en el mismo fichero, sin necesidad de copiar el fichero.
3. Permite realizar procesos de actualización en tiempo real.

**E) *Inconvenientes de esta organización:***

1. La relación entre clave y dirección (Nº de registro) la define el programador y sólo él es responsable de manejarla correctamente. El acceso a un registro se hace indicándole al sistema su dirección relativa dentro del fichero (nº de registro) y la comprobación de si el registro leído es o no el que buscamos la tiene que hacer el programador, el sistema operativo no nos avisa de posibles errores que se produzcan el cálculo de la dirección.
2. Cuando el rango de claves posibles es muy superior al de claves que realmente están grabadas en el fichero, habrá mucho espacio desaprovechado puesto que tenemos que reservar espacio a todos y cada uno de los registros teóricamente posibles. Pueden existir celdas vacías.
3. El algoritmo de direccionamiento debe garantizar que no se nos producirán dos direcciones iguales para dos valores de clave distintos. En el caso de no elegir un buen método de direccionamiento puede que la aparición de sinónimos haga complicado el acceso a algunos registros debido a que tendríamos que emplear un método de tratamiento de sinónimos que debe controlar el programador.
4. No permite el acceso secuencial a los registros por otro orden que no sea el orden en que están grabados en el soporte.
5. Sólo se puede utilizar un campo clave para acceder a los registros.

## 1.1.6.3 - Ficheros indexados.

### A) Introducción

Esta organización se basa en la utilización de **índices**, que permiten el acceso a un registro del fichero de forma directa, sin tener que leer todos los anteriores (similar a los de los libros. Si nos interesa leer un capítulo concreto podemos recurrir al índice que nos dice en que página comienza, y abrimos el libro por esa página, sin tener que mirar en todas las páginas anteriores para localizarlo).

Por tanto, existirá una **zona de registros** en la que se encuentran los datos del archivo y una **zona de índices (tabla de índices)**, que contiene una tabla con las claves de los registros y las posiciones donde se encuentran los mismos. La tabla de índices estará ordenada por el campo clave.



### B) Los índices

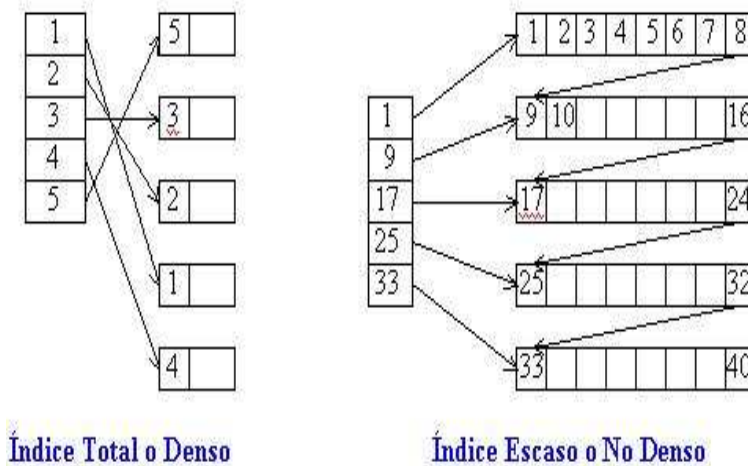
El índice debe estar organizado en función de alguno de los campos de los registros de datos. Se pueden tener tantos índices como se quiera variando la clave (o campo) que se emplee. El índice está formado por registros (entradas) que contienen:

- Clave de organización.
- Puntero(s) al fichero de datos, en concreto al registro que corresponda.

La tabla de índices será cargada en memoria principal para realizar en ella la búsqueda de la fila correspondiente a la clave del registro a encontrar, una vez localizada la dirección, sólo hay que acceder a la zona de registros en el soporte de almacenamiento y posicionarnos en la dirección indicada. Puesto que la tabla debe prever la inclusión de todas las direcciones posibles del archivo, su principal inconveniente resulta determinar su tamaño y mantenerla ordenada por los valores de la clave.

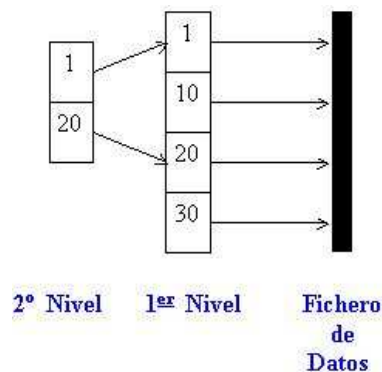
Los índices se pueden clasificar en dos tipos:

- Total o denso: cuando el índice señale a la dirección de un registro del fichero de datos (índice total o denso). El fichero puede estar desordenado.
- Escaso o no denso: cuando el índice apunte a un grupo de registros del fichero de datos que debe estar ordenado.



Tres formas principales de organizar los índices:

- **Secuencial**  
En esta forma de organización se usan cadenas de punteros. Dentro de ella podemos distinguir dos tipos:
  - *Simple:* Casi no se utiliza, las inserciones y supresiones son realizadas por copia.
  - *Encadenada:* Se emplea más que la anterior, aun así tiene el problema de que si crece mucho el fichero de datos, crece también el número de entradas, provocando que la búsqueda, al ser secuencial no se óptima para un acceso directo.
- **Multinivel o Jerarquizada:**  
Consiste en varios índices secuenciales encadenados. Tendremos un índice a los registros de datos y otros índices que apuntarán a un índice de nivel menor.



Este método mejorara la organización secuencial encadenada. Los índices de nivel alto suelen ser escasos y los de primer nivel densos. Al igual que en la organización secuencial, si aumenta el número de registros sigue aumentando el número de entradas.

- **Árbol**  
Viene a mejorar el problema del crecimiento de entradas en un nivel. Se pretende que el número de entradas en cada nivel sea fijo, y lo que crezca sea el número de niveles.

Se usan diferentes tipos de árboles, binarios (de búsqueda y AVL), multirrama y B<sup>+</sup>

### C) Características más relevantes de esta organización:

- El diseño del registro tiene que tener un campo, o combinación de campos, que permita identificar cada registro de forma única, es decir, que no pueda haber dos registros que tengan la misma información en él. A este campo se le llama **campo clave** y es el que va a servir de índice. Un mismo fichero puede tener más de un campo clave, pero al menos uno de ellos no admitirá valores duplicados y se le llama clave primaria. A las restantes se les llama claves alternativas.
- Permiten utilizar el modo de **acceso secuencial** y el modo de **acceso directo** para leer la información guardada en sus registros.
- Para acceder a este tipo de ficheros utilizando el modo de acceso directo se hace conociendo el contenido del campo clave del registro que queremos localizar. Con esa información el sistema operativo puede consultar el índice y conocer la posición del registro dentro del fichero.
- Para acceder a este tipo de ficheros utilizando el modo de acceso secuencial los registros son leídos ordenados por el contenido del campo clave, independientemente del orden en que se fueron grabando (el orden lógico no es igual al orden físico), debido a que el acceso a los datos se hace a través del índice, que para hacer más fácil la búsqueda de los registros, permanece siempre ordenado por el campo clave.
- Solamente se puede grabar en un soporte direccionable. Por ejemplo, un disco magnético. Si esto no fuera así, no podría emplear el acceso directo.

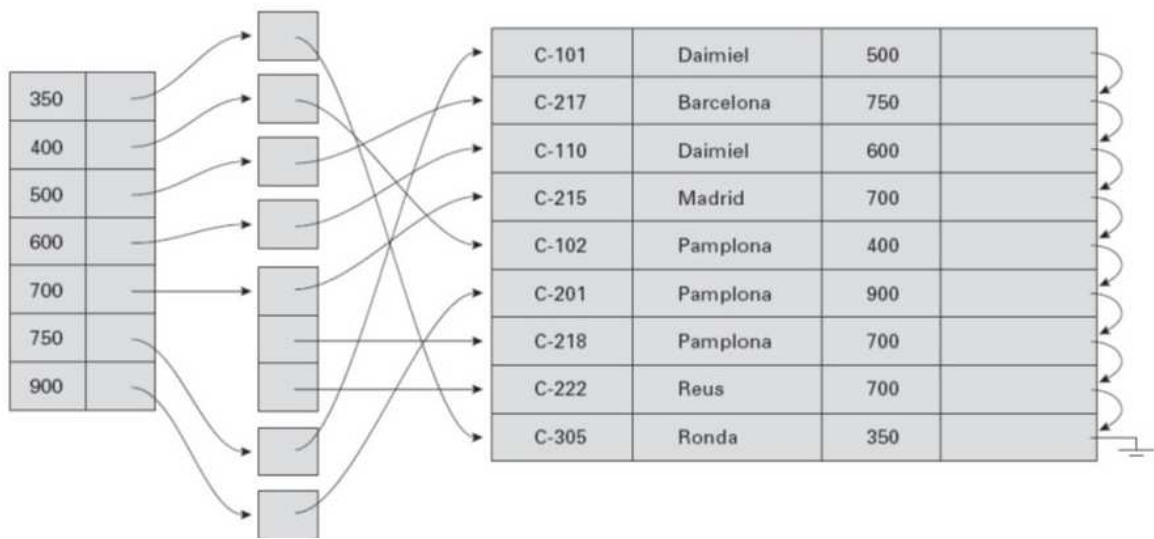
D)

E)

### F) Índices a varios niveles.

Existen diferentes formas de manejar los índices, dependiendo del sistema operativo o el lenguaje de programación con el que se trabaja. Las organizaciones de índices más utilizadas en la práctica son la **ISAM** (Indexed Sequential Access Method) y la **VSAM** (Virtual Storage Access Method).

Cuando el fichero de datos tiene muchos registros puede que el fichero de índices se haga demasiado largo y las búsquedas en él se vayan haciendo lentas, ya que cuantos más registros tenga, más crece el tiempo necesario para localizar un registro. Para evitar este problema se recurre a la utilización de índices a varios niveles, de forma que se sustituye un área de índices única, pero de un tamaño grande, por varias áreas de índices de menor tamaño donde la búsqueda es más rápida.



## 2.9.- Otros (secuenciales indexados, hash.).

Existen otros tipos de organización de ficheros, ficheros secuenciales indexados y ficheros de acceso calculado, a continuación se detallan las características de cada uno de ellos.

### a. Ficheros Secuenciales Indexados:

También llamados parcialmente indexados, al igual que en los ficheros indexados existe una **zona de índices** y otra **zona de registros de datos**, pero esta última se encuentra dividida en **segmentos** (bloques de registros) ordenados.

En la tabla de índices, cada fila hace referencia a cada uno de los segmentos. La clave corresponde al último registro y el índice apunta al registro inicial. Una vez que se accede al primer registro del segmento, dentro de él se localiza (de forma secuencial) el registro buscado.

Esta organización es muy utilizada, tanto para procesos en los que intervienen pocos registros como para aquellos en los que se maneja el fichero completo.

Las principales características son:

- Permite el acceso secuencial. Esto es muy interesante cuando la tasa de actividad es alta. En el acceso secuencial, además, los registros se leen ordenados por el campo clave.
- Permite el acceso directo a los registros. Realmente emula el acceso directo, empleando para ello las tablas de índices. Primero busca la clave en el área de índices y luego va a leer al área de datos en la dirección que le indica la tabla.



- Se pueden actualizar los registros en el mismo fichero, sin necesidad de crear un fichero nuevo de copia en el proceso de actualización.
- Ocupa mas espacio en el disco que los ficheros secuenciales, debido al uso del área de índices.
- Solo se puede utilizar soportes direccionables.
- Obliga a una inversión económica mayor, por la necesidad de programas y, a veces, hardware mas sofisticado.

#### b. Ficheros de Acceso Calculado o Hash:

Cuando utilizamos ficheros indexados es necesario siempre tener que consultar una tabla para obtener la dirección de almacenamiento a partir de la clave. La técnica del acceso calculado o **hash**, permite accesos más rápidos, ya que en lugar de consultar una tabla, se utiliza una transformación o función matemática (función de hashing) conocida, que a partir de la clave genera la dirección de cada registro del archivo. Si la clave es alfanumérica, deberá previamente ser transformada en un número.

El mayor problema que presenta este tipo de ficheros es que a partir de diferentes claves se obtenga la misma dirección al aplicar la función matemática o transformación. A este problema se le denomina **colisión**, y las claves que generan la misma dirección se conocen por **sinónimos**. Para resolver este problema se aplican diferentes métodos, como tener un bloque de excedentes o zona de sinónimos, o crear un archivo de sinónimos, etc.

Para llevar a cabo la transformación existen multitud de métodos, siendo algunos:

- **Módulo:** La dirección será igual al resto de la división entera entre la clave y el número de registros.
- **Extracción:** La dirección será igual a una parte de las cifras que se extraen de la clave.

Una buena transformación o función de hash, será aquella que produzca el menor número de colisiones. En este caso hay que buscar una función, a ser posible [biunívoca](#), que relacione los posibles valores de la clave con el conjunto de números correlativos de dirección. Esta función consistirá en realizar una serie de cálculos matemáticos con el valor de la clave hasta obtener un número entre 1 y n, siendo n el número de direcciones que tiene el fichero.

#### Autoevaluación

##### En un fichero con acceso calculado:



Se utiliza la dirección como clave.



Hay una tabla en la que está cada clave con la dirección del registro correspondiente.



La dirección se obtiene a partir de la clave mediante un algoritmo.

## **1.2- Bases de Datos**

## **1.3- Big Data**