

Bases de datos Unidad 1

**Universidad de Los Andes
Escuela de Ingeniería de Sistemas
Departamento de Computación**

Tema 1. Los sistemas de gestión de archivos

Tema 1. Los Sistemas de gestión de archivos

▶ **Contenidos**

- ▶ Estructuras de datos para la organización de índices de acceso:
 - ▶ Monodimensionales
 - ▶ Multidimensionales/Espaciales
- ▶ Objetivos
- ▶ Funciones
- ▶ Organización y métodos de acceso:
 - ▶ Relativos o directos
 - ▶ Aleatorios
 - ▶ Indizados

▶ **Objetivo**

- ▶ Introducir las estructuras de datos avanzadas y su relación con los métodos de acceso para los archivos almacenados en memoria secundaria
- ▶ **Actividad:**
 - ▶ Leer: Elmasri y Navathe cap. 13 y 14

Los sistemas de gestión de archivos

- ▶ Encargados de administrar la memoria secundaria (MS) permitiendo el uso de ciertas primitivas de acceso (abrir, cerrar, leer, escribir) a los archivos almacenados
- ▶ Archivo lógico vs. Archivo físico
- ▶ Memoria principal y memoria escondida (almacenamiento volátil) vs. Discos magnéticos y ópticos (almacenamiento no volátil)
- ▶ Discos magnéticos o dispositivos en línea vs. Cintas magnéticas o dispositivos fuera de línea





Archivos

- ▶ **Aplicaciones de bases de datos solo requieren de una pequeña porción de la BD en un momento dado, por ello hay que:**

- ▶ localizarla en disco,
- ▶ copiarla en la memoria principal para procesarla y
- ▶ luego escribirla en disco, si es que se modificaron los datos



- ▶ **Los datos en disco están almacenados en archivos de registros, indexados o no**

- ▶ Capacidad de los discos se mide en números de bytes (B) que puede almacenar (KB, MB, GB, TB y PB)



Discos

▶ **Magnéticos:**

- ▶ 1952: RAMAC (5 mega caracteres de 7 bits) 4 discos grandes de aluminio
- ▶ 1962: discos removibles de 2MB
- ▶ 1973: 3340 winchester drive de 60MB
- ▶ 1993: 1GB, 1" de alto y densidad 354 MB/pul². duro (40GB, 60GB, etc.), zip (100MB, 250MB, 750MB), jaz (1GB), syquest (44MB)

▶ **Magneto-ópticos**

▶ **Estado solido (SSD): 2 TB, lectura/escritura 520MB/seg**

▶ **Ópticos: Láser que calienta un punto en la cubierta de óxido de metal, un imán alinea el óxido para un 1 ó 0.**

- ▶ Tecnología regrabable. CD-ROM (700MB), CD-WROM. DVD-vídeo y DVD-ROM (4.7GB/cara y capa)

▶ **Flash disk: lentas en escritura, para celulares y cámaras y ahora también para computadoras (pen drive), se conectan por los puertos USB**



Arreglos de discos (RAID, 1987)

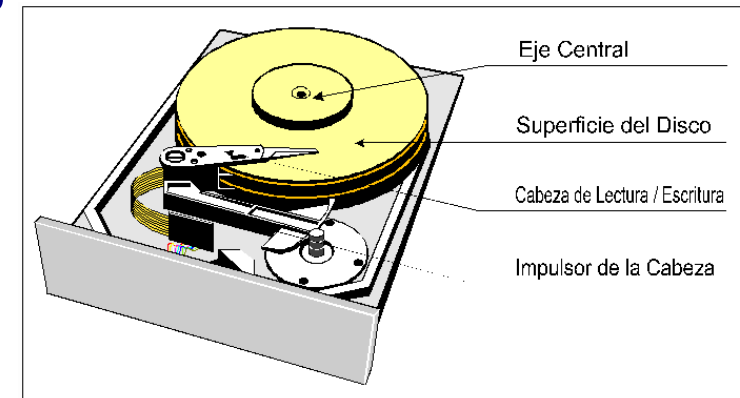
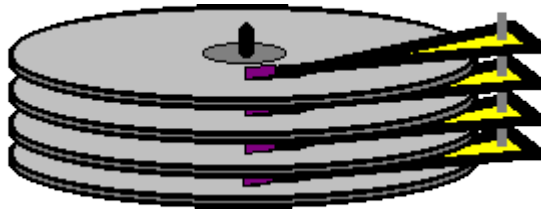
- ▶ **Nivel 0 (striping):** accede a todos los discos en paralelo, colocando los bytes de cada archivo en varios discos. Si falla un disco, falla todo el sistema
- ▶ **Nivel 1 (espejos):** L/E datos en dos o más discos, si hay una falla se pasa al espejo instantáneamente. Mejor rendimiento, menor capacidad. Tolerancia a fallas
- ▶ **Nivel 2:** no disponible comercialmente, discos no estándares
- ▶ **Nivel 3 (paridad):** coloca los datos en varios discos, pero usa un disco para tener los datos de paridad. Si hay una falla, puede reconstruir los datos sin tener la copia física (espejo)
- ▶ **Nivel 4 (paridad):** como el nivel 3 pero coloca bloques en vez de bytes, pero el disco de paridad es el cuello de botella
- ▶ **Nivel 5 (paridad):** usa bloques de datos y paridad colocados en los diferentes discos, eliminando el cuello de botella del 4
- ▶ **Nivel 6 (paridad dual):** como el nivel 5, pero calcula la paridad dos veces y la escribe en los diferentes discos. Soporta varias fallas simultáneas, pero baja el rendimiento del arreglo
- ▶ **Nivel 10:** varios pares de drivers con sus espejos, no hay pérdida de datos a menos que ambos drivers del par fallen al mismo tiempo





Discos

- ▶ **Volumen o paquete de discos: varios discos apilados con un eje común**
- ▶ **Pistas: concéntricas**
- ▶ **Número de pistas de un disco: hasta 800**
- ▶ **Capacidad de cada pista: entre 4 y 50 KB**
- ▶ **Tamaño de los bloques (B) se fija durante la iniciación y no se puede modificar dinámicamente, normalmente entre 512 y 4096 B**





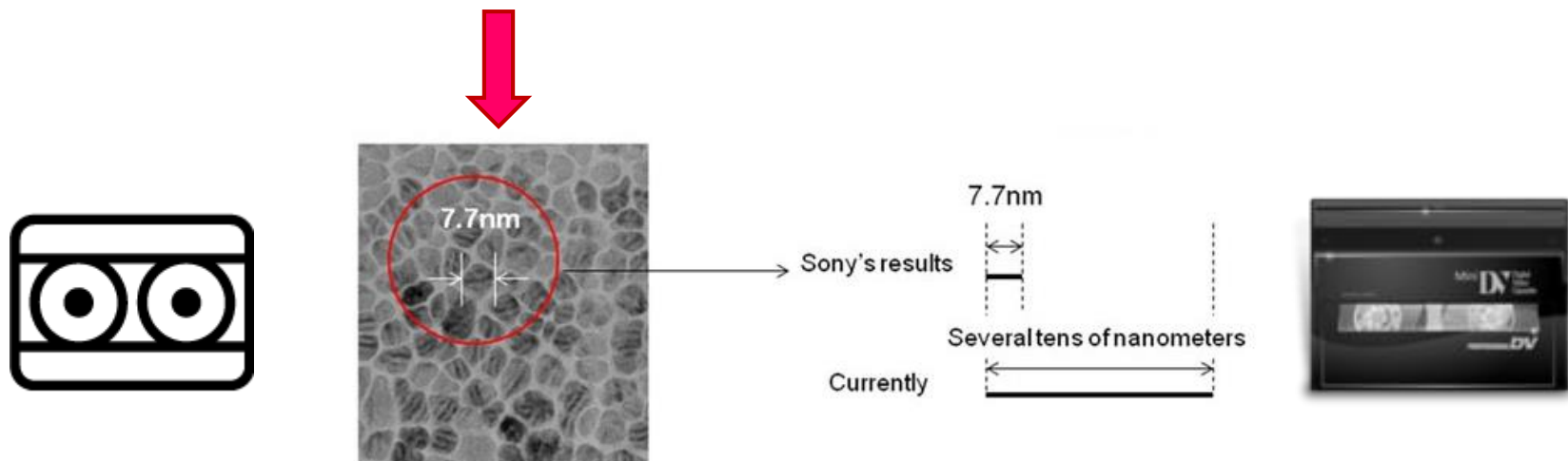
Discos

- ▶ **La dirección de hardware de un bloque es una combinación de: número de superficie, número de pista en la superficie y número de bloque en la pista**
- ▶ **Buffer: área contigua reservada en la memoria principal donde se almacena un bloque.**
 - ▶ En caso de tener varios bloques el tamaño del buffer se adecua a dicha capacidad
- ▶ **Discos de cabeza fija vs. Discos de cabeza móvil**
- ▶ **Tiempo total de localización y transferencia de bloque = tiempo de búsqueda + retardo rotacional (rr) + tiempo de transferencia de bloque (ttb), generalmente entre 15 y 60 mseg.**
- ▶ **Concurrencia intercalada vs. Concurrencia simultánea (paralelo)**



▶ **Principalmente para respaldo**

- ▶ 1952: carretes de cinta de 12" y 1,4 GB
- ▶ 2002: cartuchos de 1GB-50GB. Prototipo IBM de 1TB
- ▶ 2011: Prototipo de Fujifilm e IBM de 35TB basada en nanotecnología
- ▶ 2014: Sony de 185TB usa nano partículas magnéticas de 7,7 nanómetros de promedio, unos 0,0000077 mm



- ▶ **Un registro se compone de campos que tienen asociado un tipo de dato, lo cual constituye el tipo de registro**
- ▶ **Tipos de datos: Entero (*int*), entero largo (*long int*), caracter (*char*), real (*float*), cadena de caracteres (*string*), lógico o booleano, fecha (*date*), hora (*time*) y BLOB (*binary large object*)**
- ▶ **Tamaño de un registro (*R*): número de *B* que contiene**
- ▶ **Registros de longitud fija vs. Registros de longitud variable**
- ▶ **Campos de longitud variable y grupos repetitivos**
- ▶ **Factor de bloqueo (*fbl*) = $\lfloor B / R \rfloor$ registros por bloque**
- ▶ **Espacio desperdiciado en un bloque = $B - (fbl * R)$ bytes**

- ▶ **Número de bloques que necesita un archivo $NB = \lfloor r/fbl \rfloor$ bloques, donde r es el número total de registros del archivo y fbl es el factor de bloqueo**

- ▶ **Asignación:**

- ▶ continua
- ▶ enlazada
- ▶ segmentada
- ▶ indexada



Archivo fbl=4

r1	r2	r3	r4	
r5				

Archivo fbl=4

r1	r2	r3	r4	
r5				

Organización de archivos

- ▶ **Forma en que los registros y bloques se colocan e interconectan en el almacenamiento**
- ▶ **Método de acceso: grupo de programas que permiten la aplicación de operaciones a un archivo**
- ▶ **Operaciones:**
 - ▶ Registro por registro
 - ▶ Buscar (*find*) o localizar (*locate*), buscarSiguiente (*findNext*)
 - ▶ leer (*read*) u obtener (*get*)
 - ▶ eliminar (*delete*)
 - ▶ modificar (*modify*)
 - ▶ insertar (*insert*)
 - ▶ Para un conjunto de registros
 - ▶ buscarTodo (*findAll*) y buscarOrdenado (*findOrdered*)
 - ▶ reorganizar (*reorganize*)
 - ▶ abrir (*open*) y cerrar (*close*)
- ▶ **Estáticos vs. Dinámicos**

Archivos de registros desordenados

- ▶ **Inserción: el registro nuevo se coloca al final, haciendo la inserción muy eficiente**
 - ▶ Se carga el último bloque en el buffer, se añade el nuevo registro y se reescribe el bloque en disco
- ▶ **Búsqueda lineal: (bloque por bloque) es más costosa**
 - ▶ En caso de un archivo con NB bloques, se requiere buscar en promedio $NB/2$ bloques
- ▶ **Eliminación: no hay compactación de espacio, se carga el bloque en el buffer, se elimina el registro y se reescribe el bloque en disco**
 - ▶ Otra técnica es utilizar un campo especial como marcador de eliminación
 - ▶ Ambas formas de eliminación necesitan reorganizaciones periódicas

Archivos de registros desordenados

- ▶ **Para leer todos los registros ordenados por alguno de sus campos, se le aplica un método de ordenamiento externo como el de ordenamiento-mezcla (*merge-sort*)**
- ▶ **En caso de registros de longitud fija con bloques no extendidos y asignación contigua (archivo relativo) se accede al registro por su posición**
 - ▶ si los registros se numeran $0, 1, 2, \dots, r-1$ en cada bloque y
 - ▶ los bloques se numeran $0, 1, 2, \dots, fbl-1$
 - ▶ entonces el i -ésimo registro del archivo se encuentra en el bloque $\lfloor i / fbl \rfloor$ y será el $(i \bmod fbl)$ -ésimo registro de ese bloque

Archivos de registros ordenados

- ▶ **Orden según un campo clave con la ventaja de encontrar más rápido los registros según la clave usando búsqueda binaria, por lo que en promedio se requieren $\lg NB$ accesos a bloques**
 - ▶ Los accesos por otra clave diferente a la de ordenamiento implica una búsqueda lineal
- ▶ **Inserción y eliminación más costosas pues se debe mantener el orden**
- ▶ **Inserción implica encontrar el punto de inserción según el valor de la clave del nuevo registro y abrir un espacio en esa posición**
 - ▶ Implica correr en promedio la mitad de los registros una posición hacia abajo, lo que se traduce en leer y reescribir la mitad de los bloques del archivo

Archivos de registros ordenados

- ▶ **Opciones para inserción: mantener espacio libre dentro de los bloques o tener un archivo de desborde no ordenado donde se insertarán los nuevos registros y periódicamente se fusiona éste con el archivo ordenado o maestro.**
- ▶ **Eliminación menos problemática si se usa el marcador de eliminación**
 - ▶ Durante la reorganización del archivo se eliminan realmente los registros marcados
- ▶ **Normalmente se usan en BD con caminos de acceso adicionales como los índices**

Archivo disperso o aleatorio (*hashing*)

- ▶ Se le aplica una función de dispersión a la clave de dispersión arrojando la dirección del bloque donde debe ir o estar el registro
- ▶ Los métodos de transformación clave-dirección externos son los de dispersión:

- ▶ dinámica
- ▶ extensible
- ▶ lineal

- ▶ Ellos dividen el espacio de direcciones en cubetas y usan una tabla que asocia el número de la cubeta con la dirección del bloque que contiene al registro

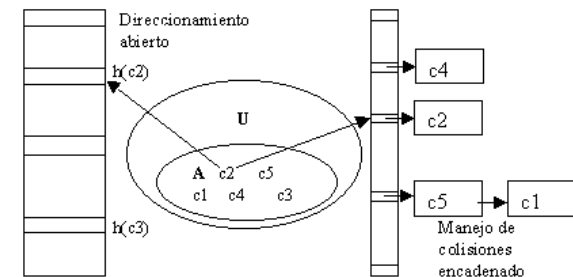
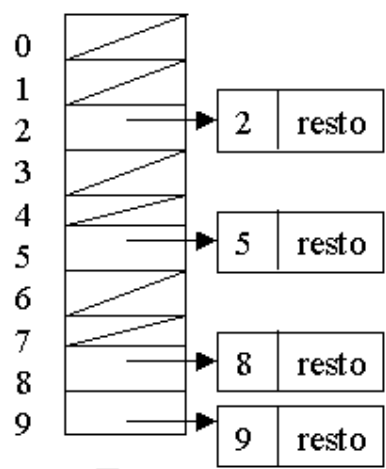
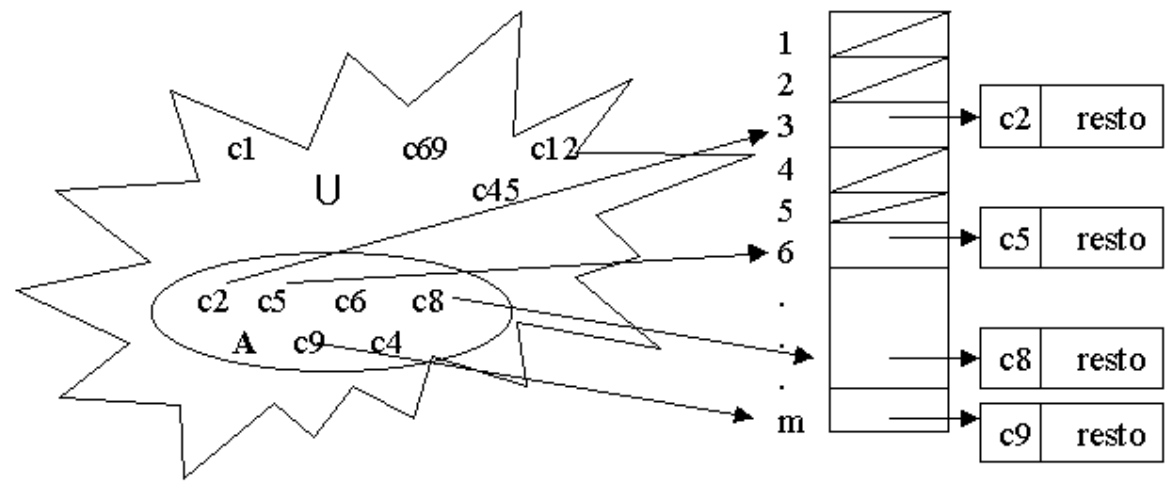
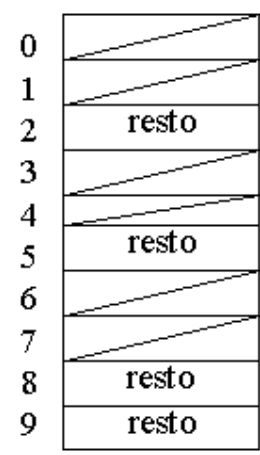


Tabla *hash*



T



T

buscarTabla(TipoClave: cl)
 regrese T(cl)
insTabla(TipoEleTabla: e)
 T(e.Clave()) = e
eliTabla(TipoClave: cl)
 T(cl) = Nulo

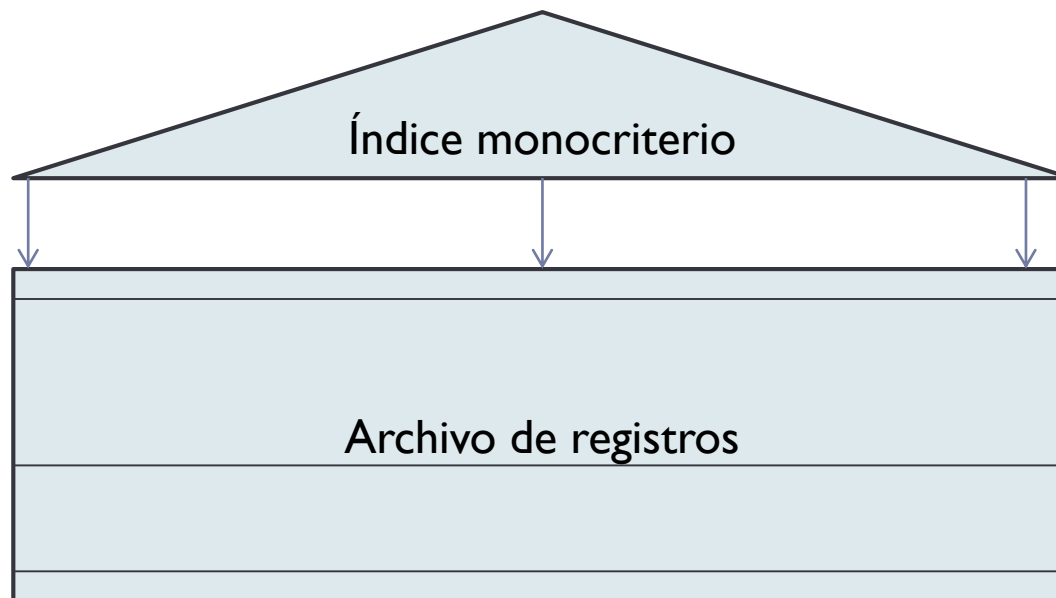
Todas las operaciones en $O(1)$

Organización indizada

- ▶ **Índices: agilizan los procesos de búsqueda y recuperación de los datos almacenados en MS (discos)**
- ▶ **Indexación de archivos: permite tener varios caminos de acceso a los datos almacenados en los archivos**
- ▶ **Índices:**
 - ▶ monocriterio o monodimensionales y
 - ▶ multicriterio o multidimensionales

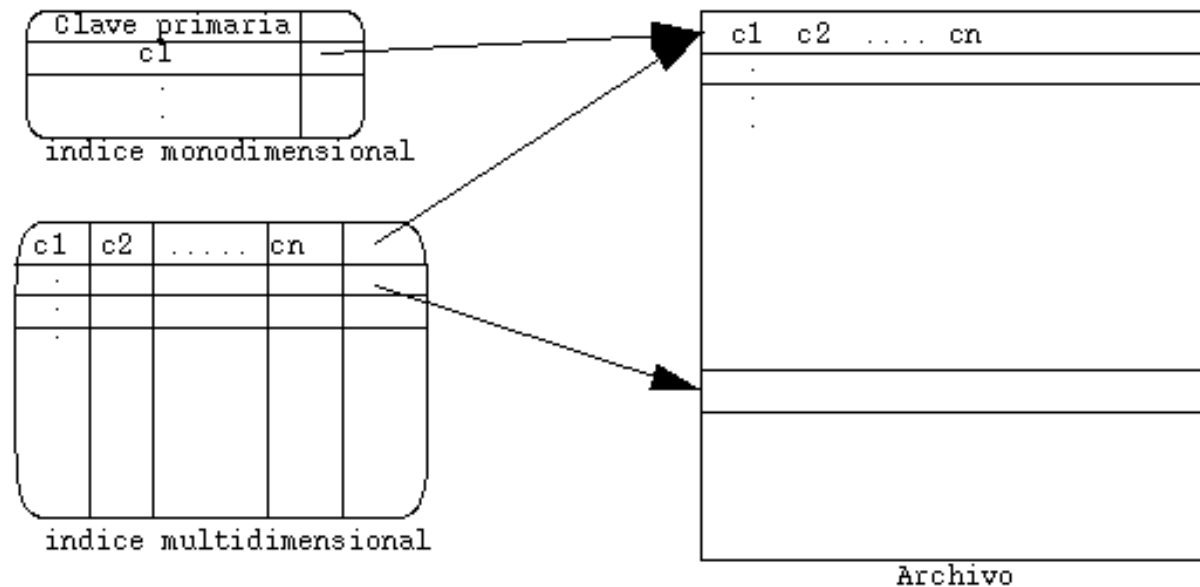
Índices monocriterio

- ▶ **Sus entradas se construyen según una única clave que normalmente es una clave primaria (uno o varios campos concatenados de un registro que identifican unívocamente cada registro del archivo de datos)**



Índices multicriterios

- ▶ **Sus entradas están construidas con varios campos no concatenados (claves secundarias) de un registro del archivo de datos**



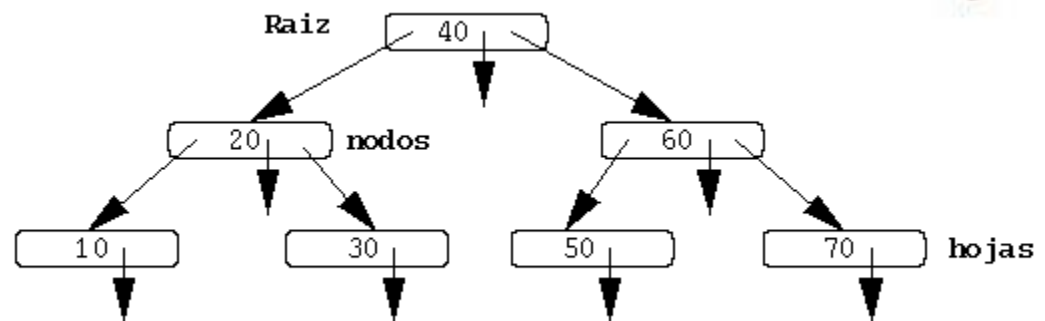
Árbol binario

- ▶ **Un árbol binario de tipo T es una estructura que puede ser vacía o que puede estar formada por:**
 - ▶ un dato de tipo T denominado raíz del árbol binario
 - ▶ un árbol binario de tipo T denominado subárbol izquierdo
 - ▶ un árbol binario de tipo T denominado subárbol derecho
- ▶ **Altura del árbol binario (h)**
- ▶ **Número máximo de nodos = $2^{h+1} - 1$**



Árbol binario de búsqueda

- inserción
- eliminación
- consulta



▶ **Ventajas:**

- ▶ Facilidad y simplicidad en los algoritmos de tratamiento
- ▶ Para encontrar la información buscada no es siempre necesario llegar al nivel de las hojas
- ▶ Los costos de tratamiento son logarítmicos

▶ **Desventajas:**

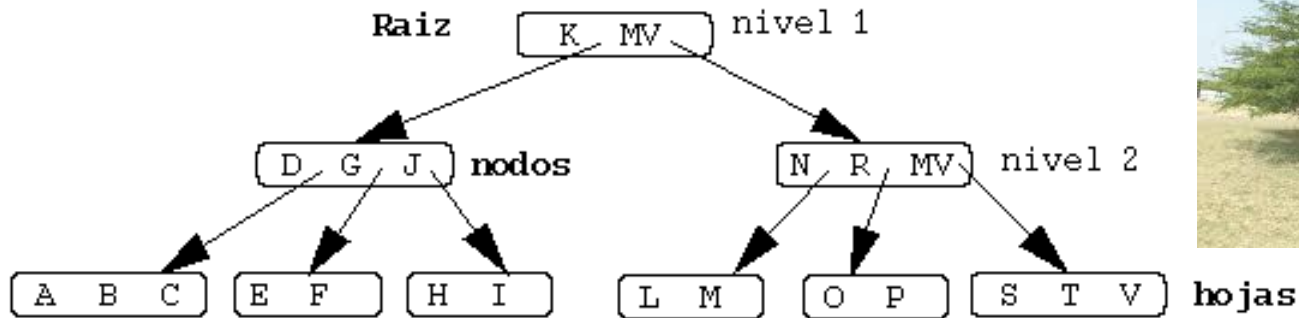
- ▶ Depende del orden de inserción de los datos, por lo que es muy frecuente obtener árboles desequilibrados o desbalanceados (una ramas muy largas en comparación con otras). Este desequilibrio obliga a la aplicación de algoritmos para equilibrar, que por lo general no son simples
- ▶ Se hace difícil realizar un tratamiento secuencial de las claves, dependiendo de la política de inserción

- ▶ **Propuestos por vez primera por R. Bayer en 1970**

Definición:

"Sean los enteros $h \geq 0$ y k que pertenece a los Naturales, donde h es la altura del árbol y k es llamado el orden del árbol_B. Un árbol T pertenece a la clase $t(k,h)$ de los árboles_B, si T no está vacío ($h > 0$) o si tiene las propiedades siguientes:

- ▶ Cada rama, desde la raíz hasta cualquier hoja tiene una longitud h denominada altura de T
- ▶ Cada nodo tiene como mínimo $k+1$ hijos excepto la raíz y las hojas. La raíz es una hoja o ella tiene como mínimo 2 hijos
- ▶ Cada nodo tiene como máximo $2k+1$ hijos"



► Número de nodos:

- Nmin es el número mínimo de nodos en el árbol:
 - $N_{min} = 1 + 2 [(k+1)^0 + (k+1)^1 + \dots + (k+1)^{h-2}]$
 - $N_{min} = 1 + 2/k [(k+1)^{h-1} - 1]$ para $h \geq 2$
- Nmax es el número máximo de nodos en el árbol

$$N_{max} = \sum_{i=0}^{h-1} (2k+1)^i = \frac{1}{2k} (2k+1)^h - 1 \text{ para } h \geq 1$$

- ▶ **Altura del árbol (h) de páginas se calcula en función del número de claves en el archivo (N)**
- ▶ **Operaciones:**
 - ▶ Búsqueda
 - ▶ Inserción
 - ▶ Eliminación

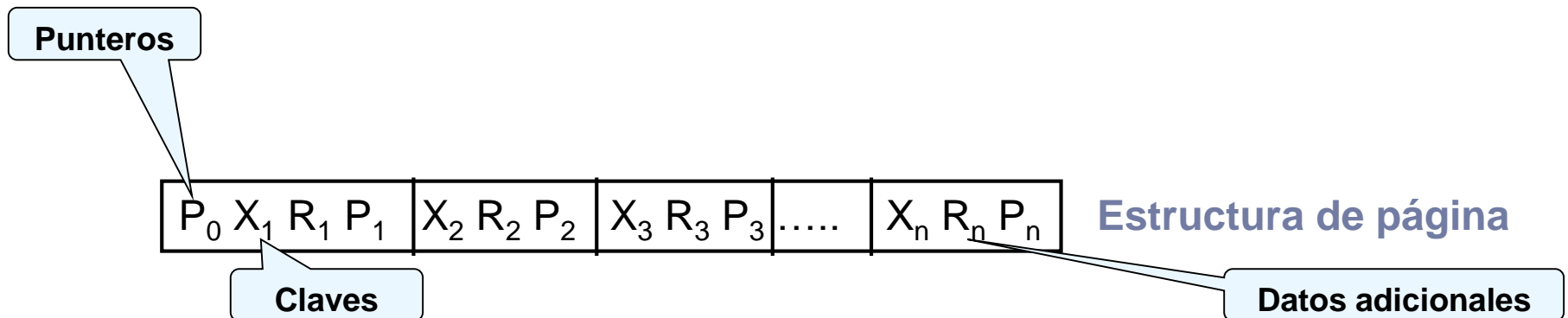


$$N + 1 \geq 2(k + 1)^{h-1}$$

$$h \leq 1 + \log_{k+1} \frac{N + 1}{2}$$

Implementacion del árbol_B

- ▶ **Cada nodo es una página y las propiedades de dicha estructura son:**
 - ▶ Cada página tiene entre k y $2k$ claves, excepto la raíz que tiene entre 1 y $2k$ claves.
 - ▶ Sea L el número de claves en una página P y P no es una hoja, entonces P tiene $L+1$ hijas.
 - ▶ Dentro de cada página las claves X_1, X_2, \dots, X_L están almacenadas siguiendo el orden creciente o ascendente de su alfabeto



▶ **Ventajas:**

- ▶ Facilidad para la construcción y mantenimiento del árbol
- ▶ Obtención de un árbol siempre equilibrado
- ▶ No es necesario llegar siempre hasta el último nivel (hojas) para encontrar una clave solicitada
- ▶ La altura del árbol es menor que la de un árbol binario
- ▶ Los costos de tratamiento son logarítmicos

▶ **Desventajas:**

- ▶ El tiempo y la dificultad para obtener un tratamiento secuencial de las claves en el árbol
- ▶ El tamaño de las entradas es bastante grande por la inclusión de la información asociada a la clave R, lo que implica menos entradas por página, es decir más accesos a memoria secundaria

Autoevaluación

1. **¿Qué es un sistema de gestión de archivos?**
2. **¿Cuál es la principal diferencia entre cinta y disco magnético?**
3. **¿Qué es un registro y cómo se almacenan en MS?**
4. **¿Cuáles son las principales operaciones de los archivos?**
5. **¿Cuáles son las diferencias entre los archivos de registros ordenados y desordenados?**
6. **¿Cuáles son las diferencias entre la organización indizada y aleatoria?**
7. **¿Qué es un índice monocriterio y uno multicriterio?**
8. **¿Cuáles son las ventajas y desventajas de un árbol_B?**
9. **¿Cuáles son las diferencias entre los árboles binarios y los B?**

Dibuje lo solicitado a continuación para las claves:

**2, 34, 56, 1, 45, 44, 89, 78, 64, 77, 92, 43, 55, 12, 9, 8, 10, 32,
16, 4 y 18**

- 1. Un árbol binario**
- 2. La implementación de un árbol_B con $k=2$**