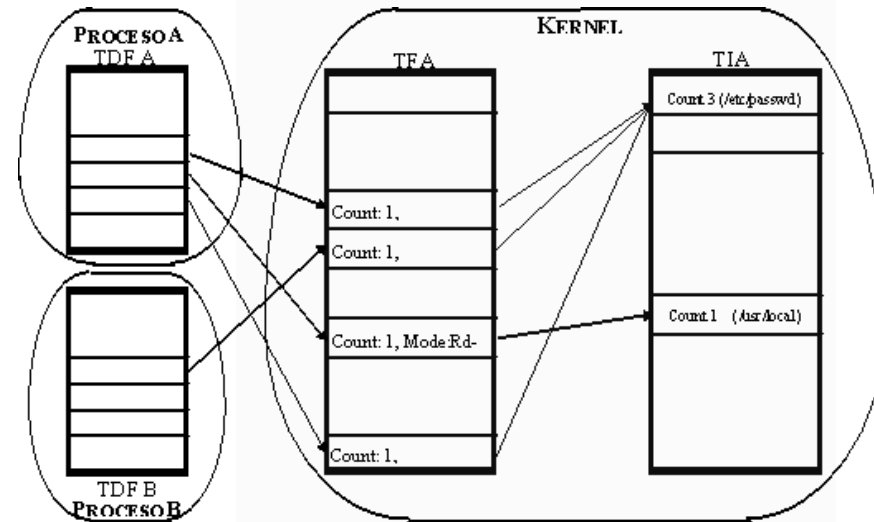


Manejo e Implementación de Archivos

Guatemala 16 de
septiembre de 2021

Ing. David Luna



Universidad
Rafael Landívar
Tradición Jesuita en Guatemala

Agenda

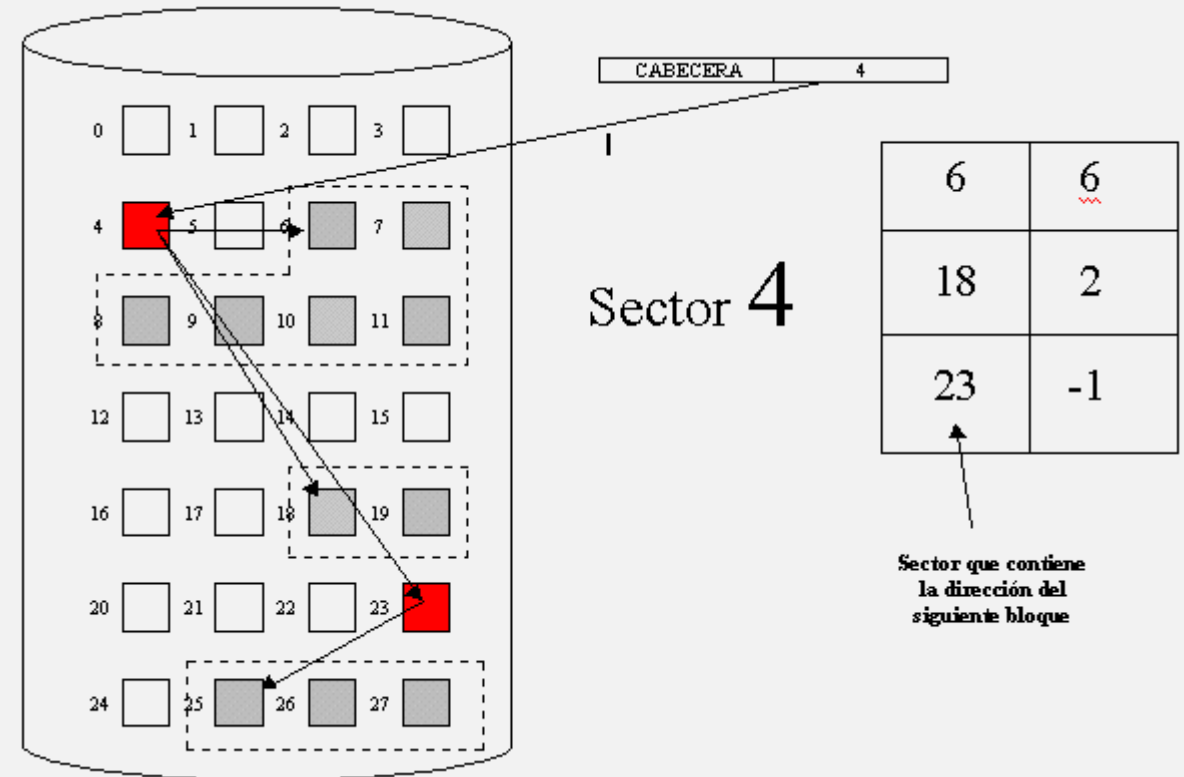
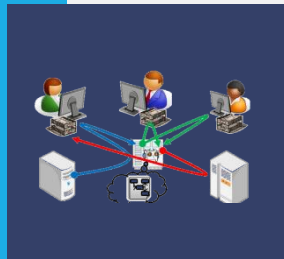


**Asignación de
espacio**



Cierre

Asignación de espacio



Asignación / Liberación de espacio

Aspectos a considerar:

- Forma de almacenamiento de archivos y directorios.
- Administración del espacio en disco.
- Forma de hacerlo de la mejor manera (Eficiencia y confiabilidad)

Asignación / Liberación de espacio

Se deben tener presentes problemas tales como la “fragmentación” creciente del espacio en disco:

- Ocasiona problemas de rendimiento al hacer que los archivos se desperdigen a través de bloques muy dispersos.
- Técnicas para aliviar el problema de la “fragmentación” consisten en realizar periódicamente:
 - “Condensación”: se pueden “reorganizar” los archivos expresamente o automáticamente según algún criterio predefinido.

http://upload.wikimedia.org/wikipedia/commons/4/40/Defragmenting_disk.gif

- “Recolección de basura o residuos” (garbage collector): se puede hacer fuera de línea o en línea, con el sistema activo, según la implementación.

- Reorganización del espacio

Condensación



- Problemas de rendimiento

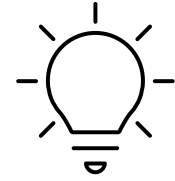
Fragmentación



- Recolección de desechos

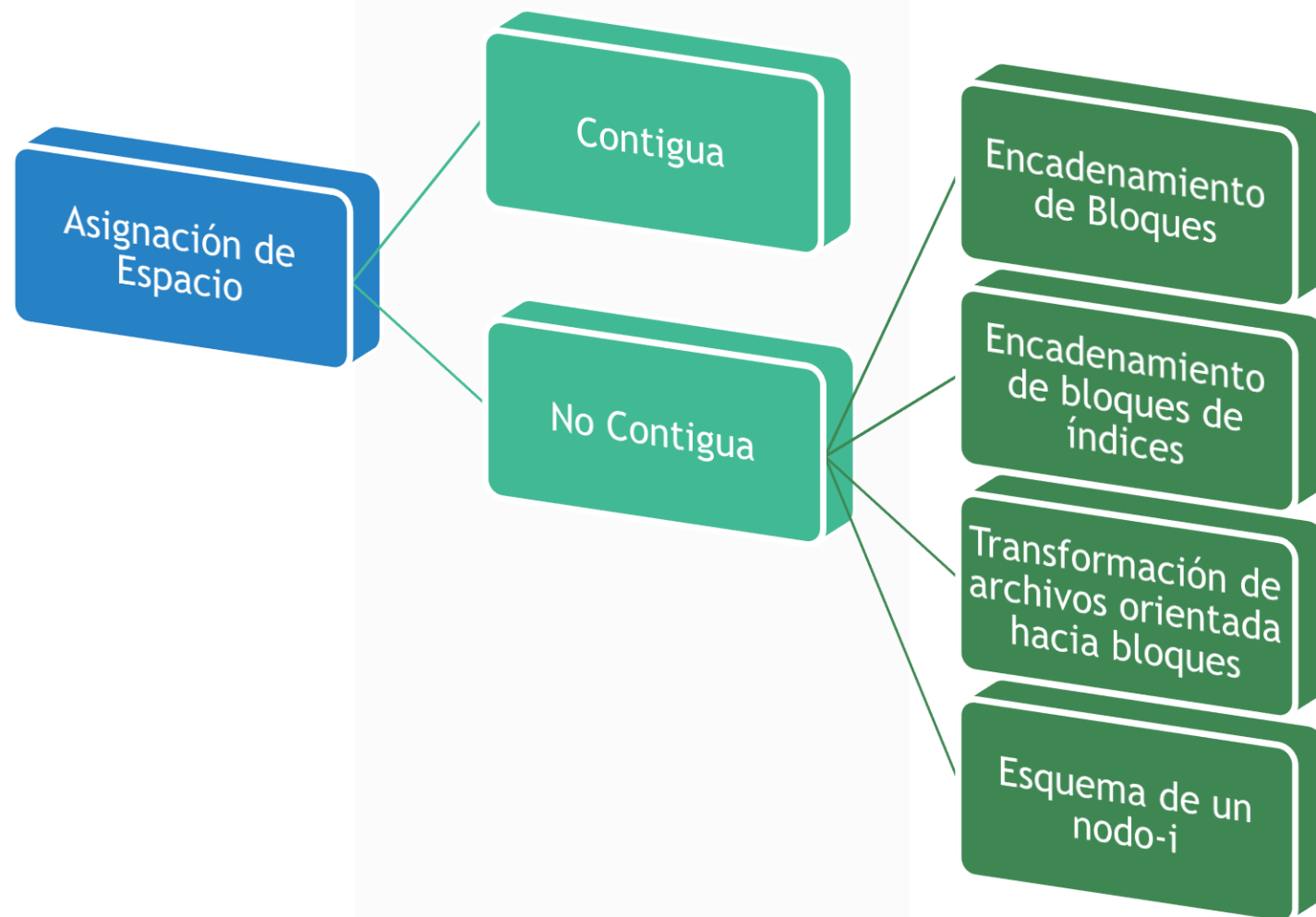
Garbage Collector





El aspecto clave de la implantación del almacenamiento de archivos es *el registro de los bloques asociados a cada archivo.*

Implementación de Archivos



Asignación Contigua o Adyacente.

Los archivos son asignados a áreas contiguas de almacenamiento secundario.

La principal **ventajas** es:

- Facilidad de implantación, ya que solo se precisa el número del bloque de inicio para localizar un archivo.

Los principales **defectos** son:

- Se debe conocer el tamaño máximo del archivo al crearlo.
- Produce una gran fragmentación de los discos.

Asignación No Contigua.

Son esquemas de almacenamiento más dinámicos.

El sistema trata de asignar nuevos bloques a un archivo eligiendo bloques libres lo más próximos posible a los bloques del archivo existentes.

Se destacan los siguientes:

1. Encadenamiento de bloques
2. Encadenamiento de bloques de índices
3. Transformación de archivos orientada hacia bloques
4. Esquema de un nodo-i

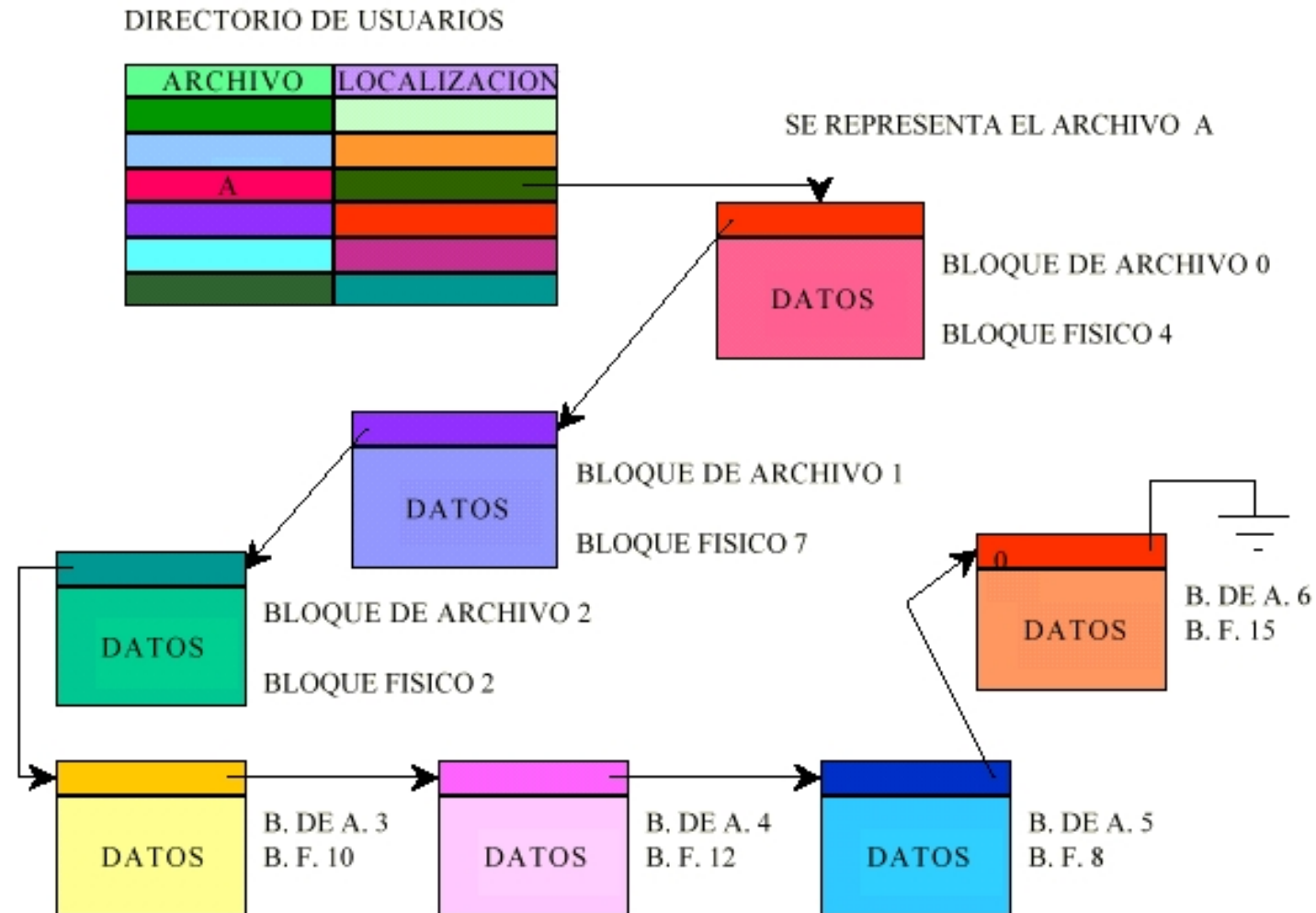
Asignación No Contigua.

1. *Encadenamiento de bloques o lista ligada:*

- Las entradas en el directorio apuntan al primer bloque de cada archivo.
- Cada uno de los bloques de longitud fija que forman un archivo contiene dos partes:
 - Un bloque de datos.
 - Un apuntador al bloque siguiente.
- Cada bloque contiene varios sectores.

Asignación No Contigua.

1 Encadenamiento de bloques o lista ligada:



Figura~4.4: Encadenamiento de bloques o lista ligada de bloques.

Asignación No Contigua.

1 *Encadenamiento de bloques o lista ligada:*

- Localizar un registro determinado requiere:
 - Buscar en la cadena de bloques hasta encontrar el bloque apropiado.
 - Buscar en el bloque hasta encontrar el registro.
- El examen de la cadena desde el principio puede ser lento ya que debe realizarse de bloque en bloque, y pueden estar dispersos por todo el disco.
- Se pueden usar “*listas de encadenamiento doble*”, hacia adelante y hacia atrás, con lo que se facilita la búsqueda



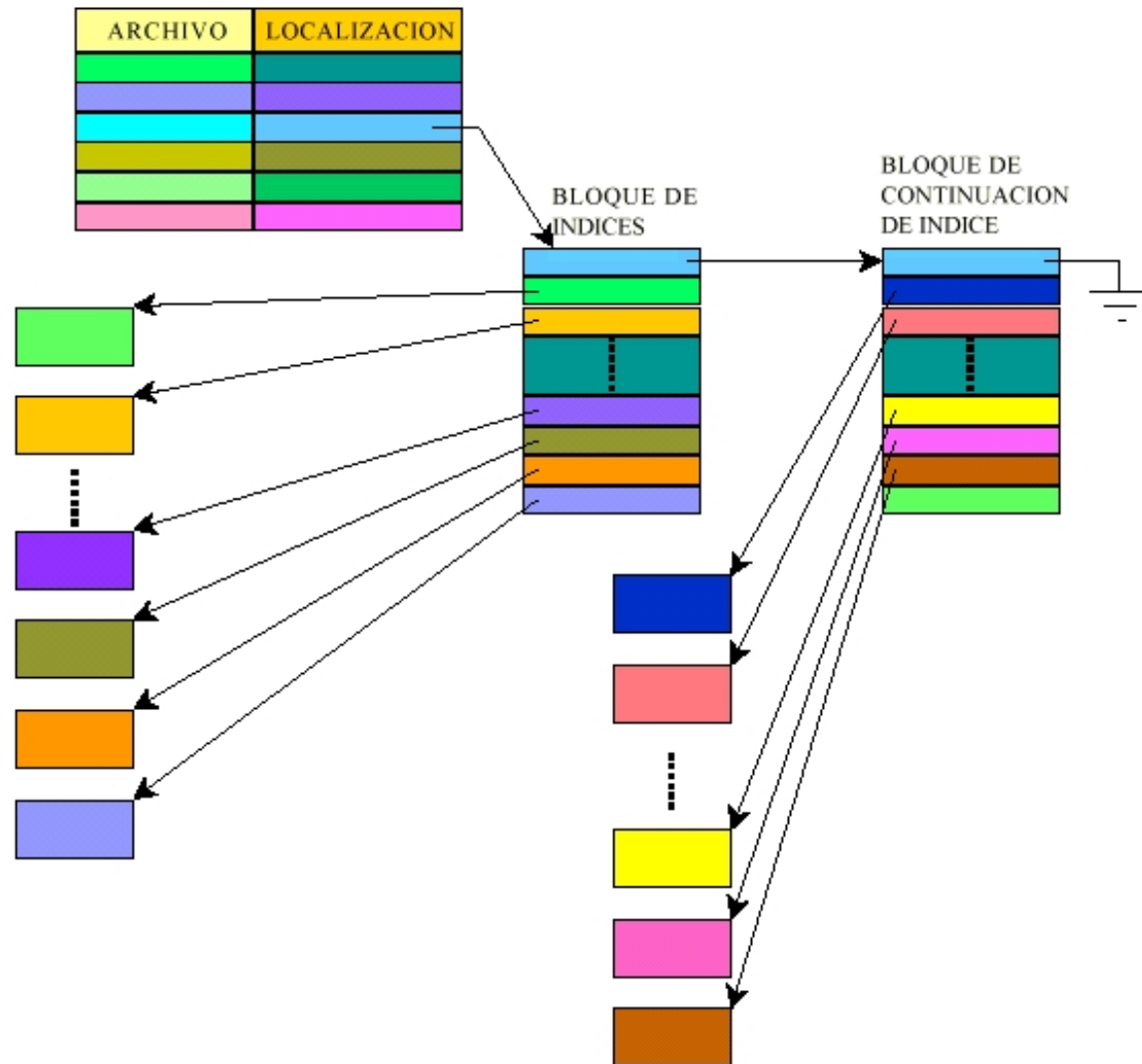
Asignación No Contigua.

2. Encadenamiento de bloques de índice:

- Los apuntadores son colocados en varios bloques de índices separados:
 - Cada bloque de índices contiene un número fijo de elementos.
 - Si es necesario utilizar más de un bloque de índices para describir un archivo, se encadena una serie de bloques de índices.

Asignación No Contigua.

2. Encadenamiento de bloques de índice:



Asignación No Contigua.

2. Encadenamiento de bloques de índice:

- La gran **ventaja** es que la búsqueda puede realizarse en los propios bloques de índices.
- Los bloques de índices pueden mantenerse juntos en el almacenamiento secundario para acortar la búsqueda.
- La principal **desventaja** es que las inserciones pueden requerir la reconstrucción completa de los bloques de índices:
 - Una posibilidad es dejar vacía una parte de los bloques de índices para facilitar inserciones futuras y retardar las reconstrucciones.
- Es suficiente que el dato del directorio contenga el número de bloque inicial para localizar todos los bloques restantes, sin importar el tamaño del archivo.

Asignación No Contigua.

3. Transformación de archivos orientada hacia bloques :

- Se utilizan números de bloques en vez de apuntadores.
- Se conserva un mapa del archivo, conteniendo una entrada para cada bloque del disco.
- Las entradas en el directorio del usuario apuntan a la primera entrada al *mapa del archivo* para cada archivo.
- Cada entrada al mapa del archivo contiene el número del bloque siguiente de ese archivo.
- La entrada al mapa del archivo correspondiente a la última entrada de un archivo determinado se ajusta a algún valor "*centinela*" ("*nil*") para indicar que se alcanzó el último bloque de un archivo.
- El sistema puede mantener una *lista de bloques libres*.
- La principal **ventaja** es que las cercanías físicas del disco se reflejan en el mapa del archivo

DIRECTORIO DE USUARIOS

ARCHIVO	LOCALIZACION
A	8
B	6
C	2

MAPA DEL ARCHIVO

0	22
1	NIL
2	5
3	26
4	9
5	20
6	10
7	LIBRE
8	17
9	1
10	14
11	LIBRE
12	3
13	4
14	0
15	LIBRE
16	LIBRE
17	12
18	13
19	NIL
20	23
21	LIBRE
22	18
23	19
24	LIBRE
25	LIBRE
26	NIL
27	LIBRE

orientada hacia

BLOQUES FISICOS EN EL ALMACENAMIENTO SECUNDARIO

BLOQUE 0 B(4)	BLOQUE 1 B(10)	BLOQUE 2 C(1)	BLOQUE 3 A(4)	BLOQUE 4 B(8)	BLOQUE 5 C(2)	BLOQUE 6 B(1)
BLOQUE 7 LIBRE	BLOQUE 8 A(1)	BLOQUE 9 B(9)	BLOQUE 10 B(2)	BLOQUE 11 LIBRE	BLOQUE 12 A(3)	BLOQUE 13 B(7)
BLOQUE 14 B(3)	BLOQUE 15 LIBRE	BLOQUE 16 LIBRE	BLOQUE 17 A(2)	BLOQUE 18 B(6)	BLOQUE 19 C(5)	BLOQUE 20 C(3)
BLOQUE 21 LIBRE	BLOQUE 22 B(5)	BLOQUE 23 C(4)	BLOQUE 24 LIBRE	BLOQUE 25 LIBRE	BLOQUE 26 A(5)	BLOQUE 27 LIBRE

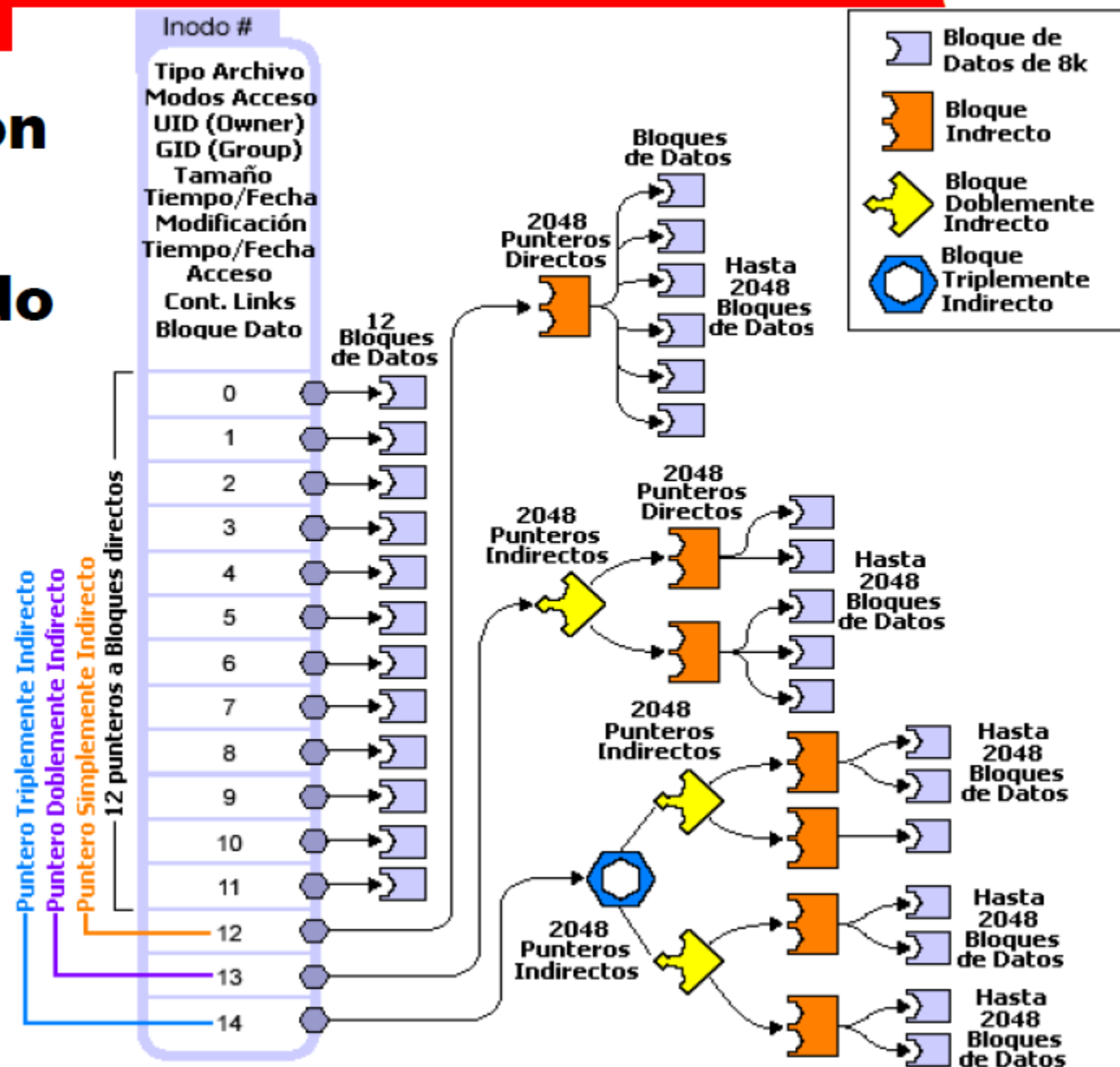
Figura 4.6: Transformación de archivos orientada hacia bloques.

Asignación No Contigua.

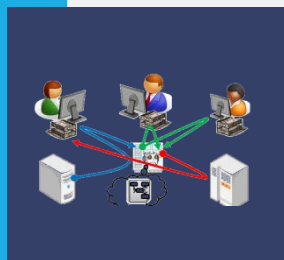
4. *Nodos-i (nodos índices)* :

- Se asocia a cada archivo una pequeña **tabla**, llamada nodo-i (nodo índice):
 - Contiene los atributos y direcciones en disco de los bloques del archivo.
 - Se traslada del disco a la memoria principal al abrir el archivo.
 - Si el archivo es pequeño, toda la información está en el nodo-i.
 - Si el archivo es grande, una de las direcciones en el nodo-i es la dirección de un bloque en el disco llamado *bloque simplemente indirecto*:
 - Contiene las direcciones en disco adicionales.
 - Si resulta insuficiente, otra dirección en el nodo-i, el *bloque doblemente indirecto*, contiene la dirección de un bloque que presenta una lista de los bloques simplemente indirectos:
 - Cada bloque simplemente indirecto apunta a un grupo de bloques de datos.
- De ser necesario se pueden utilizar *bloques triplemente indirectos*

Información contenida en un i-nodo



Seguridad y Respaldo





Seguridad del Sistema de Archivos

Muchos esquemas de protección se basan en la hipótesis de que
el sistema conoce la identidad de cada usuario:

- La identificación de los usuarios se conoce como la *autenticación de los usuarios*.

Seguridad del Sistema de Archivos

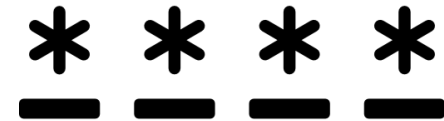
- Muchos métodos de autenticación se basan en:



• Algo que es *el usuario*.

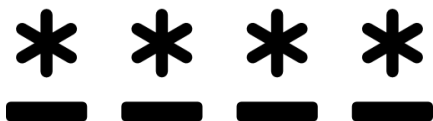


• Algo que *posee el usuario*.



• La identificación de algo *conocido por el usuario*.

Seguridad del Sistema de Archivos



- **Contraseñas**
 - Son la forma de autenticación más utilizada.
 - Son de fácil comprensión e implementación.
 - Deben *almacenarse cifradas (encriptadas)*.
 - Se deben prever intentos de ataques, consistentes en *pruebas de combinaciones* de nombres y contraseñas.

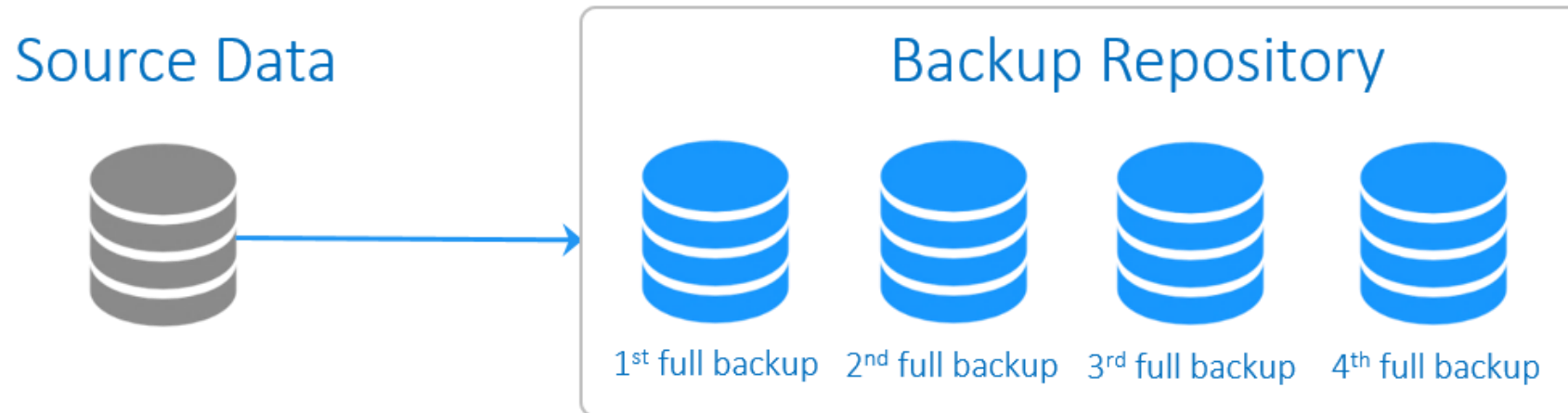
Respaldo y recuperación

La destrucción de la información, ya sea accidental o intencional, es una realidad y tiene distintas causas :

- Fallas de hardware y de software.
- Fenómenos meteorológicos atmosféricos.
- Fallas en el suministro de energía.
- Incendios e inundaciones.
- Robos, vandalismo (incluso terrorismo).
- Esta posible destrucción de la información debe ser tomada en cuenta por:
 - Los sistemas operativos en general.
 - Los sistemas de archivos en particular.

Respaldo y recuperación

Full Backup



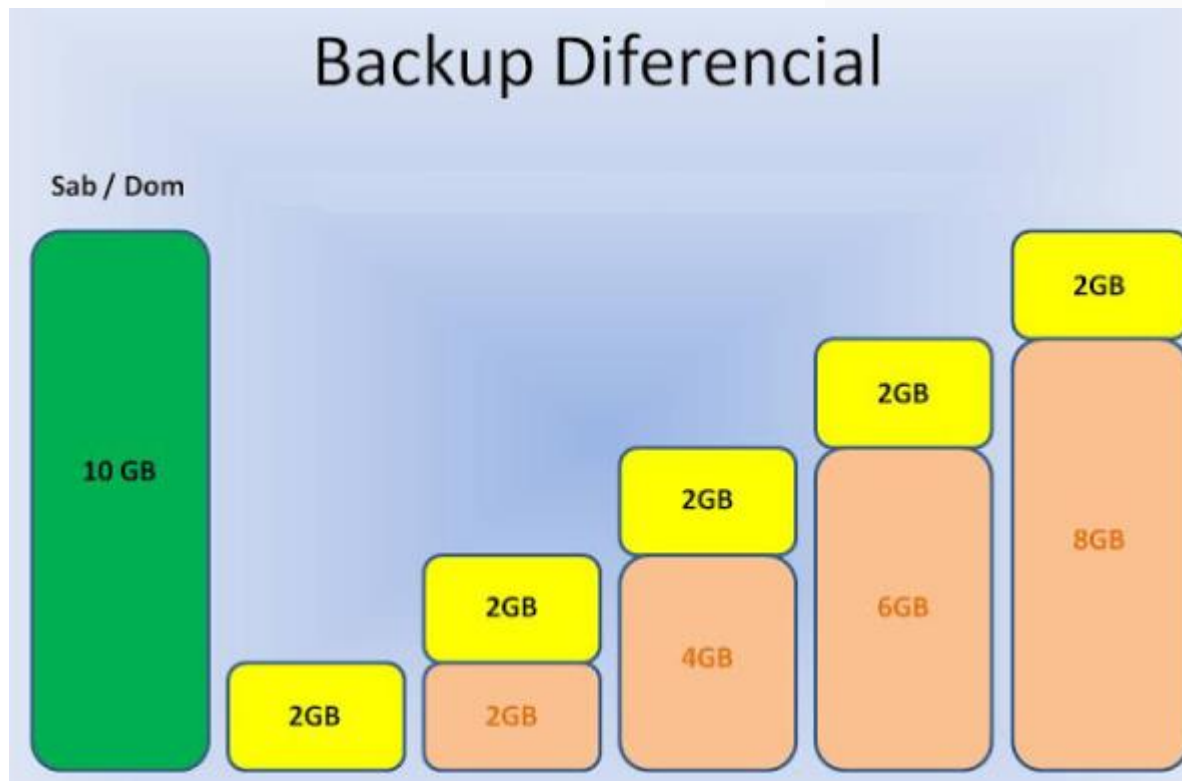
Copia completa: Copia todos los archivos cada vez que se lanza el proceso.

El proceso es tardado. Requiere más espacio.

La restauración es más rápida y simple.

Respaldo y recuperación

2. Copia diferencial:

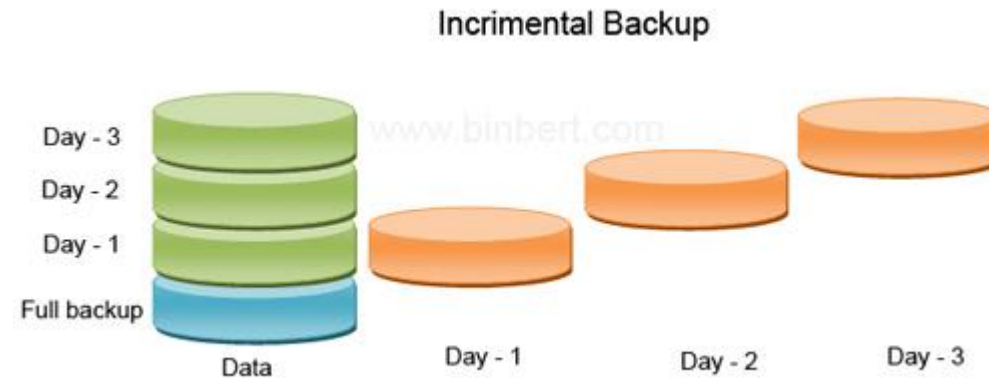


- Copia únicamente los archivos que han sido creados o modificados desde la última copia completa.
- Proceso de hacer la copia es más rápida.
- Las copias diferenciales se van haciendo más grandes ya que reflejan los cambios desde una marca fija en el tiempo.

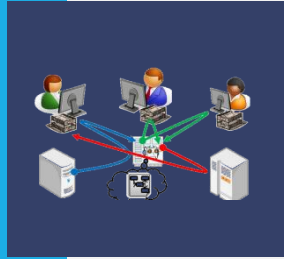
Respaldo y recuperación

Copia Incremental: (o diferencial incremental).

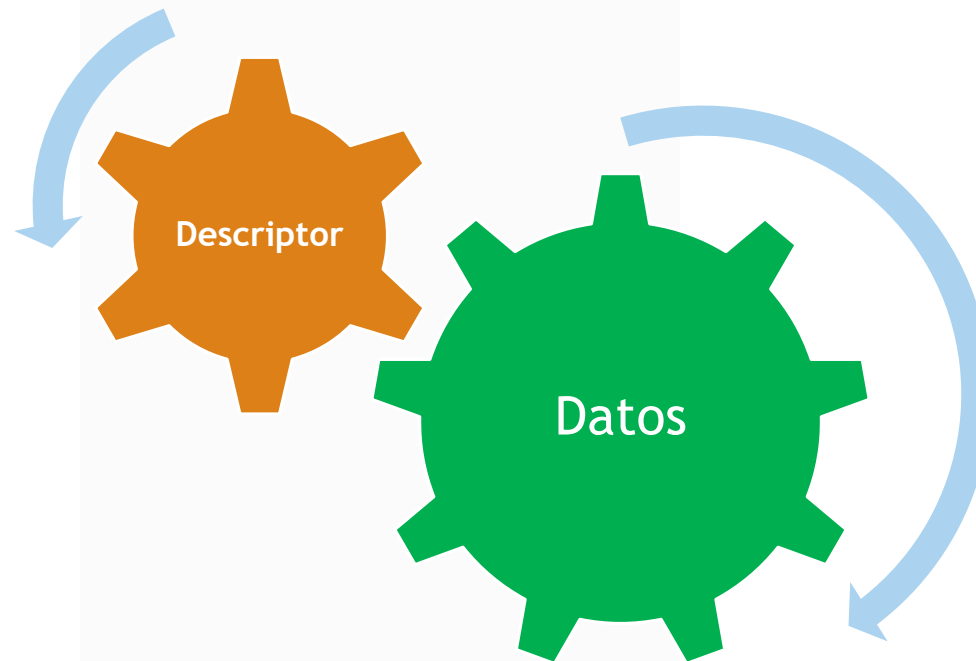
Es más avanzada (o “inteligente”) ya que sólo copia los ficheros creados o modificados desde el último backup, ya sea completo o incremental, reduciendo la cantidad de información a copiar en cada proceso.



Componentes de un archivo



Componentes de un Archivo



Descriptor:

El descriptor de archivos o bloque de control de archivos, son los datos que el sistema necesita para administrar dicho archivo.

- ⦿ Nombre simbólico del archivo
- ⦿ Localización del archivo en el almacenamiento secundario
- ⦿ Organización del archivo (método de organización y acceso)
- ⦿ Datos de control de acceso
- ⦿ Tipo (archivo de datos, programa objeto, programa fuente, etc.)
- ⦿ Disposición (permanente, temporal)
- ⦿ Fecha y hora de creación
- ⦿ Fecha de la última modificación
- ⦿ Usuario que modificó
- ⦿ Estadísticas de acceso (número de lecturas, escrituras, tiempos de acceso, etc.)
- ⦿ Registros activos e inactivos.
- ⦿ Metadatos
- ⦿ Y más...

Contenido del Archivo:

Datos, que pueden estar estructurados en la forma de:

➤ Registros,

- Están compuestos por campos. Estos campos se identifican por lo que se conoce como "metadatos" (información acerca de los datos), en el caso de los campos el metadato es el nombre de dicho campo.

➤ Formación abstracta,

- El archivo tiene al inicio (a veces al final) una serie de bytes que representan los metadatos del archivo, es decir, si por ejemplo el archivo es una imagen al inicio tendremos bytes que nos digan el largo, el ancho, el número de colores, etc, estos bytes se encuentran especificados en el "formato" del archivo. Son formatos estándar de aplicaciones específicas que se emplean en todas las plataformas

Gracias

¿ALGUNA PREGUNTA?