

Gestión de archivos

Un **archivo** es una colección de datos permanente asociada a un nombre, los archivos proporcionan a los procesos entrada/salida permanente.

El **objetivo principal** de la gestión de archivos es proporcionar al usuario o a la aplicación acceso transparente a los archivos.

Los **objetivos particulares** son:

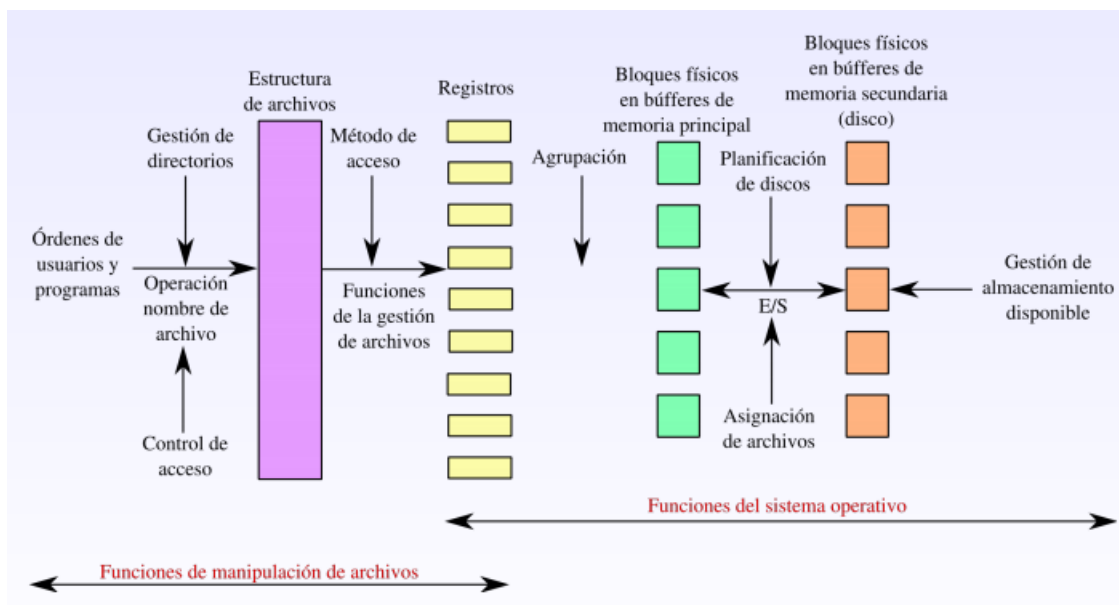
- Cumplir con las necesidades de gestión de datos.
- Garantizar que los datos de los archivos sean válidos.
- Optimizar el rendimiento.
- Ofrecer soporte de E/S.
- Minimizar o eliminar la posibilidad de pérdida o destrucción de datos.
- Proporcionar soporte de E/S para múltiples usuarios.

Cada usuario debe ser capaz de:

- Crear, borrar y modificar sus archivos.
- Acceder (si tiene permiso) a los archivos de otros usuarios.
- Controlar qué tipos de accesos estarán permitidos a sus archivos.
- Reestructurar sus archivos de manera adecuada al problema.
- Mover datos entre archivos.
- Guardar una copia de reserva y recuperar sus archivos en caso de que hayan sufrido algún daño.
- Acceder a sus archivos mediante un nombre simbólico.

La **arquitectura** de la gestión de archivos es la siguiente:

1. Solicitud de acceso a archivos, por el nombre de estos.
2. Solicitud de acceso a bloques lógicos que pertenecen a un archivo.
3. Responsable del comienzo y final de toda E/S. Mantiene las estructuras de control.
4. Instrucciones de alto nivel para acceder a bloques de disco.
5. Instrucciones de bajo nivel (al hardware) para acceder a bloques de disco, comenzar y terminar operaciones de E/S.



El **sistema de archivos** esta compuesto de una colección de archivos organizados de manera lógica en directorios divididos lógicamente o físicamente en particiones independientes.

El **MBR (registro maestro de arranque)**, carga el S.O. contenido en su partición y en el caso de ordenadores con múltiples sistemas operativos en ese bloque se encuentra el sistema de arranque de sistemas. Por uniformidad, cada partición inicia con el bloque de arranque, aunque no contenga un S.O. arrancable. Este es el **bloque de arranque**.

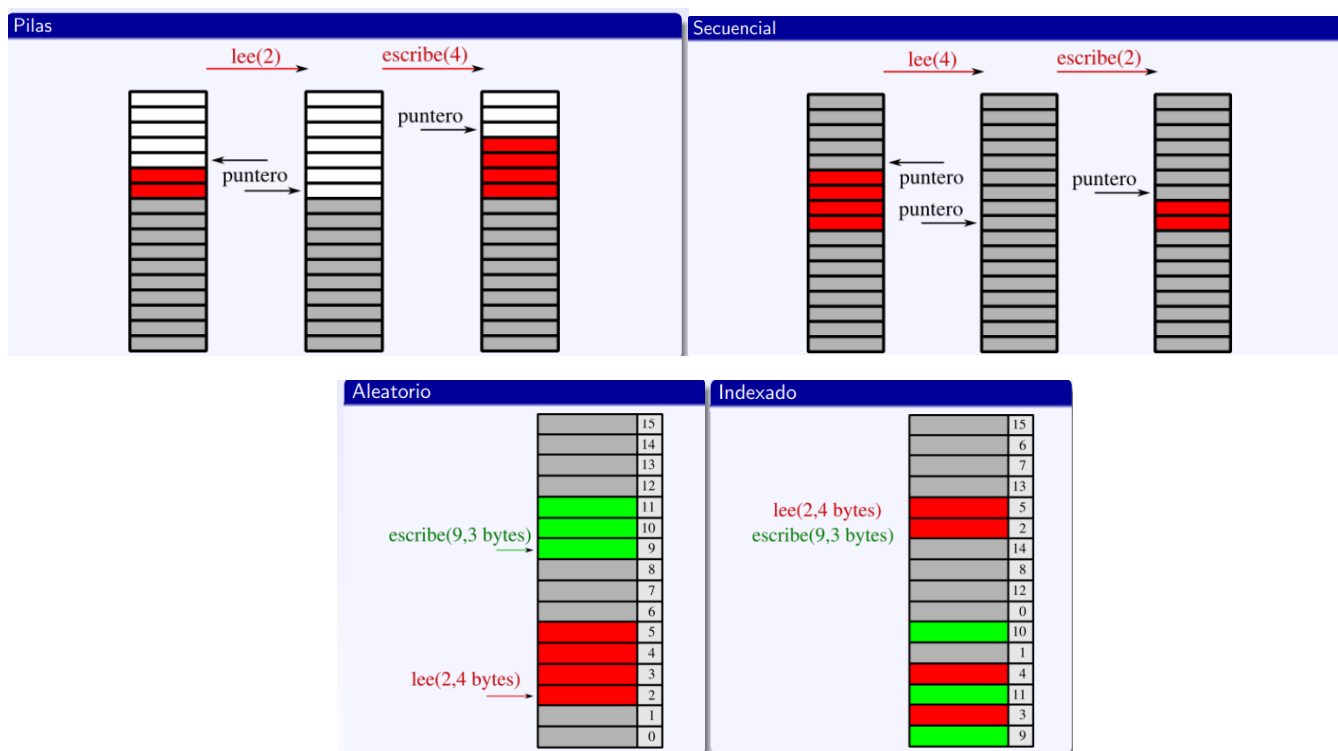
El **superbloque**, contiene parámetros claves: Tipo de sistemas de archivos (FAT, NTFS, ...), número de bloques, tamaño de bloques e información administrativa. Es el bloque siguiente al de arranque y se transfiere del disco a memoria al arrancar el sistema.

Hay varias clases de archivos:

- Normales: Contienen información de usuario. Texto y binarios.
- Directorios: Mantienen estructura del sistema de archivos.
- Especiales: Caracteres (para modelos dispositivos de E/S), bloques (para modelar discos) de dispositivo.
- ...

El tipo de fichero (o la aplicación que puede “entender” el fichero) está codificado en la cabecera.

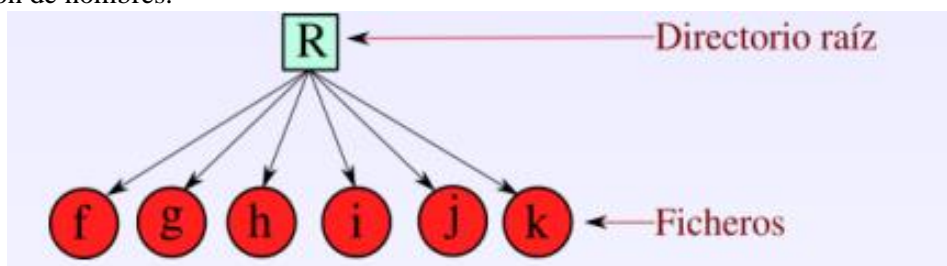
Existen varios métodos de acceso a los archivos:



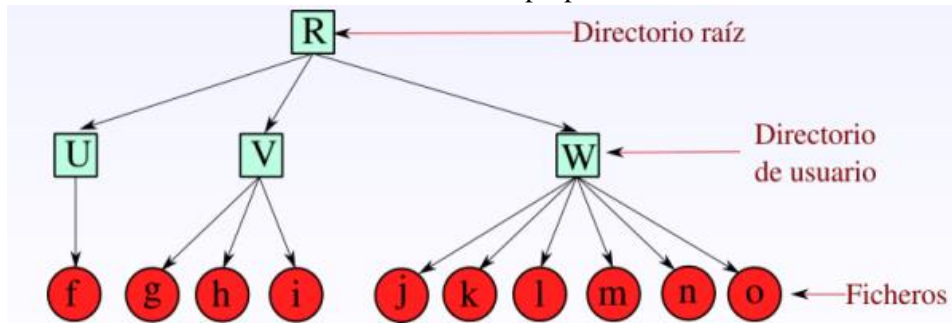
Los directorios contienen información sobre los archivos (atributos, ubicación y propietario). El directorio es propiamente un archivo, poseído por el sistema operativo. Ofrece una traducción entre los nombres de archivo y los archivos propiamente dichos.

La estructura lógica de los directorios se divide en:

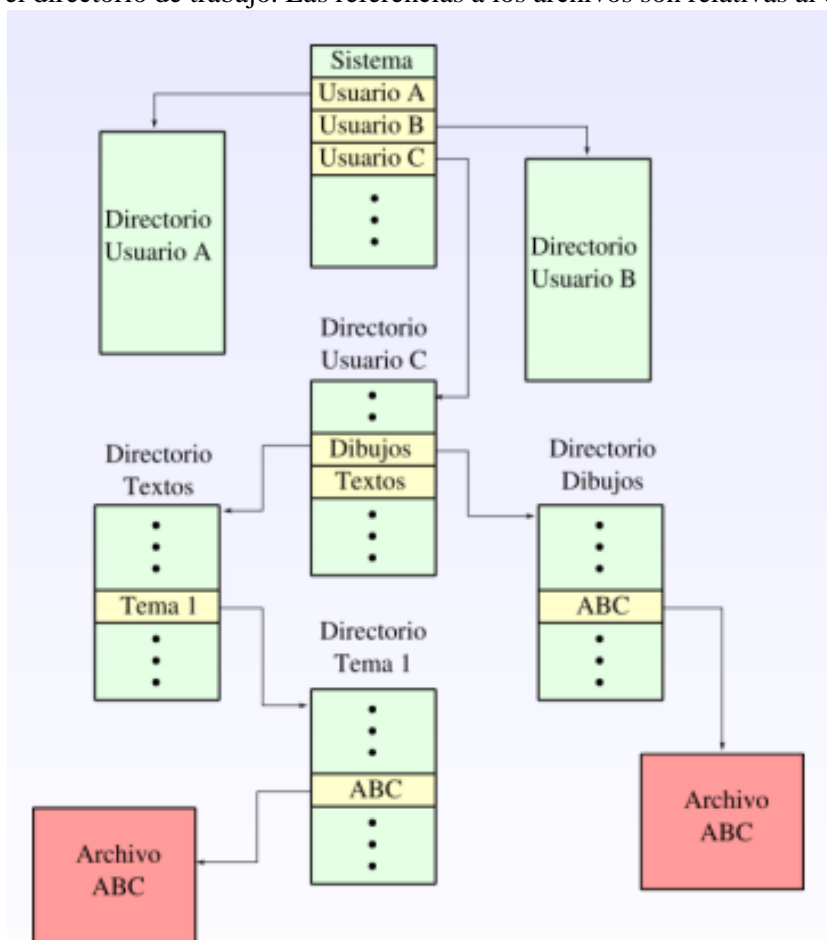
- **Directorio de un solo nivel:** Todos los ficheros comparten el mismo directorio. El problema es que puede haber confusión de nombres.



- **Directorio de dos niveles:** Cada usuario tiene un directorio propio.

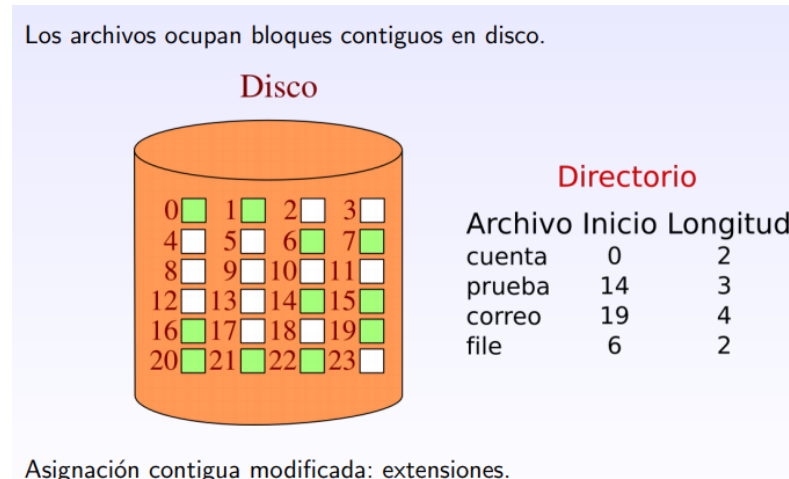


- **Directorio con estructura de árbol:** Generalización de los directorios de dos niveles.
 - Cada archivo tiene un nombre y una ruta de acceso absoluta, que es el camino desde el directorio raíz hasta el archivo.
 - Se usan separadores (/ , \ , > , dependiendo del sistema).
 - La ruta de acceso relativo indica el camino hasta un archivo a partir del directorio de trabajo o directorio activo.
 - Identificadores especiales, “.” directorio de trabajo, “..” directorio padre.
- **Estructura mononivel:** Es una lista de entradas, una para cada archivo. Puede representarse con un simple archivo secuencial, con el nombre del archivo haciendo las veces de clave. No ofrece ayuda a la organización de archivos, pero obliga al usuario a tener cuidado de no usar el mismo nombre para dos tipos diferentes de archivo. Las búsquedas son lentas y se complica el uso de comodines.
- **Estructura de 2 niveles:** Un directorio para cada usuario y un directorio maestro. El directorio maestro contiene una entrada para cada directorio de usuario. Cada directorio de usuario es una simple lista de los archivos del usuario. Todavía no ofrece a los usuarios ayuda alguna para estructurar sus conjuntos de archivos.
- **Estructura jerárquica:** Existe un directorio maestro que contiene un número determinado de directorios de usuario. Cada uno de estos directorios puede tener a su vez subdirectorios y archivos como entradas. Cualquier archivo puede ser localizado siguiendo un camino desde el directorio raíz o maestro, descendiendo por varias ramas. Se pueden tener varios archivos con el mismo nombre mientras tengan caminos únicos. El directorio actual es el directorio de trabajo. Las referencias a los archivos son relativas al directorio de trabajo.

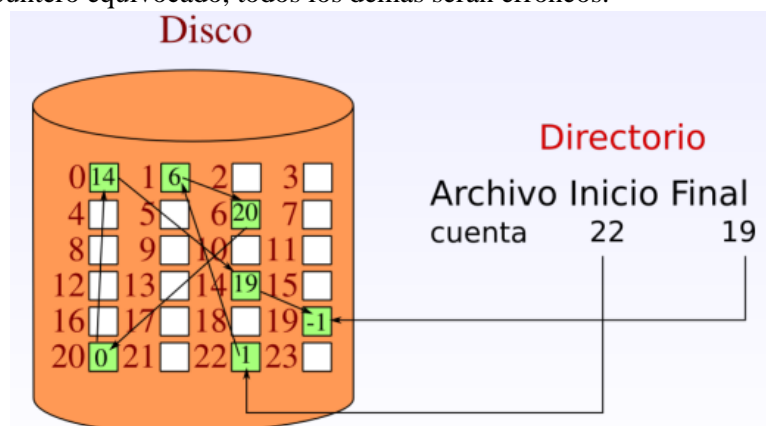


Implantación:

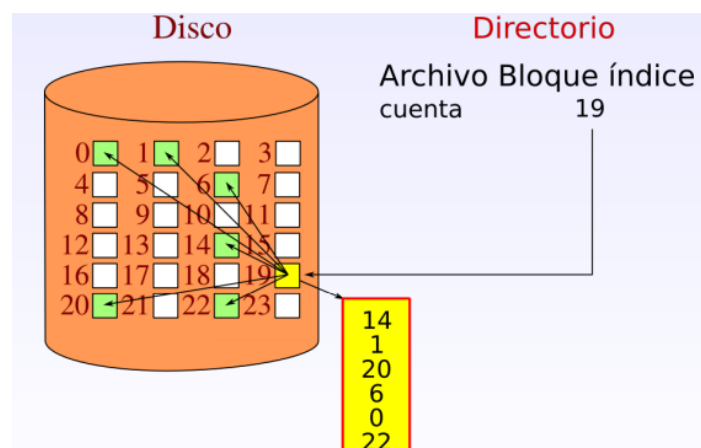
- **Asignación contigua de bloques:** Es fácil de implementar ya que la localización de un archivo se realiza a través de un único número. Tiene un alto rendimiento, los accesos requieren una única búsqueda y luego como máximo movimientos de un sector. Lo malo es que la asignación dinámica de espacio se complica. Normalmente se usan los métodos del primer ajuste o del mejor ajuste, se produce fragmentación externa. Se asigna espacio a un archivo sin saber su tamaño final, esto se soluciona sobre asignando espacio y/o reubicando los archivos cuando ocupen todo el hueco existente.



- **Asignación enlazada con listas:** Se evita la fragmentación externa e interna. Lo malo es que el acceso directo es muy lento, se requiere espacio para almacenar los punteros y no es del todo fiable ya que si se produce un fallo y se obtiene un puntero equivocado, todos los demas serán erróneos.

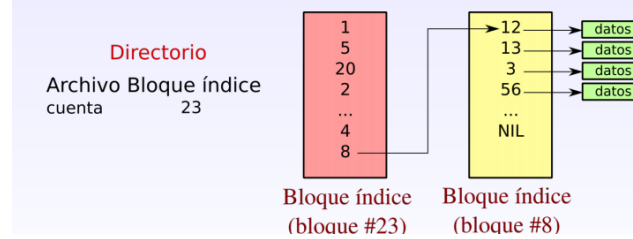


- **Asignación enlazada con índices:** Se resuelven los problemas de la asignación enlazada, sacando los apuntadores de los bloques de disco y almacenarlos en una tabla-imagen del disco llamada **tabla de asignación de archivos** o **FAT**. La tabla contiene todos los punteros en una zona de disco al principio de cada partición, y está indexada por el número de bloque.
- **Asignación indexada:** Agrupa todos los punteros de un archivo en un bloque índice. Facilita el acceso directo, aunque se produce fragmentación interna.



Hay varias posibilidades para la asignación indexada:

Esquema enlazado



Índice Multinivel

