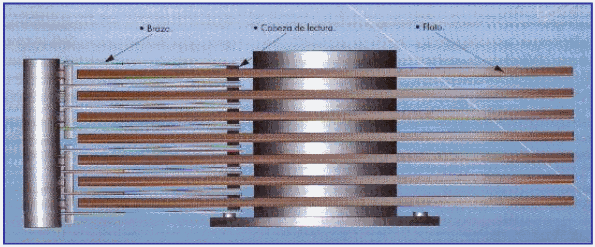
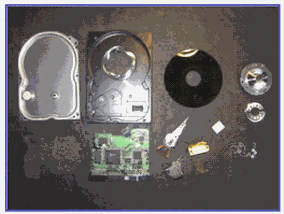
Cuestionario 3 Discos

1. **¿Cuál es la estructura física y lógica de un disco?**

El disco duro esta compuesto por varios discos o platos apilados distantes de una carcasa impermeable al aire y al polvo.



* **Piezas de un disco duro:**

****

Como se puede apreciar en la figura un disco duro esta contenido de diferentes piezas que se van a mencionar a continuación:

* Platos o discos donde se graban los datos.
* El cabezal de [lectura](http://www.monografias.com/trabajos14/textos-escrit/textos-escrit.shtml)/[escritura](http://www.monografias.com/trabajos16/metodo-lecto-escritura/metodo-lecto-escritura.shtml).
* El impulsor de cabezal ([motor](http://www.monografias.com/trabajos10/motore/motore.shtml)).
* Electroimán que es el que mueve el cabezal.
* Un circuito electrónico de control lo cual contiene, la interfaz con el ordenador, [memoria](http://www.monografias.com/trabajos13/memor/memor.shtml) caché.
* Una caja que protege al disco duro de la suciedad o polvo del medio.
* Una bolsita desecante con lo cual se evita la humedad.
* Tornillos que son especiales.

## ESTRUCTURA [FÍSICA](http://www.monografias.com/Fisica/index.shtml) DE UN DISCO DURO:

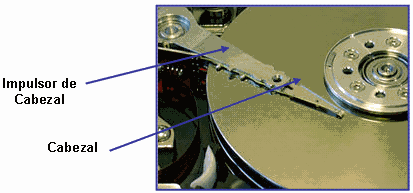
El disco duro esta compuesto por las siguientes [estructuras](http://www.monografias.com/trabajos15/todorov/todorov.shtml#INTRO):

**Platos:**

También llamados discos. Estos discos están elaborados de aluminio o [vidrio](http://www.monografias.com/trabajos11/vidrio/vidrio.shtml) recubiertos en su superficie por un material ferromagnético apilados alrededor de un eje que gira gracias a un motor, a una [velocidad](http://www.monografias.com/trabajos13/cinemat/cinemat2.shtml#TEORICO) muy rápida. El diámetro de los platos oscila entre los 5cm y 13 cm.

Cabezal de lectura/escritura:

Es la parte del disco duro que lee y escribe los datos del disco. La mayoría de los discos duros incluyen una cabeza de lectura/escritura a cada lado del plato o disco, pero hay algunos discos de alto [desempeño](http://www.monografias.com/trabajos15/indicad-evaluacion/indicad-evaluacion.shtml) tienen dos o mas cabezas sobre cada que tienen dos o más cabezas sobre cada superficie esto de manera que cada cabeza atienda la mitad del disco reduciendo la distancia del desplazamiento radial.

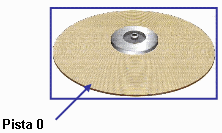
****

**Impulsor de Cabezal:**

Es un motor que mueve los cabezales sobre el disco hasta llegar a la pista adecuada, donde esperan que los sectores correspondientes giren bajo ellos para ejecutar de manera efectiva la lectura/escritura.

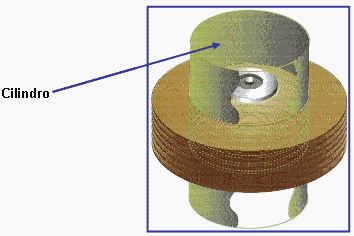
**Pistas:**

La superficie de un disco esta dividida en unos elementos llamadas pistas concéntricas, donde se almacena la [información](http://www.monografias.com/trabajos7/sisinf/sisinf.shtml). Las pistas están numeradas desde la parte exterior comenzando por el 0. Las cabezas se mueven entre la pista 0 a la pista más interna.

****

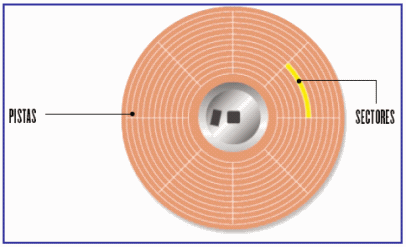
**Cilindro:**

Es el conjunto de pistas concéntricas de cada cara de cada plato, los cuales están situadas unas encima de las otras. Lo que se logra con esto es que la cabeza no tiene que moverse para [poder](http://www.monografias.com/trabajos35/el-poder/el-poder.shtml) acceder a las diferentes pistas de un mismo cilindro. Dado que las cabezas de lectura/escritura están alineadas unas con otras, la controladora de disco duro puede escribir en todas las pistas del cilindro sin mover el rotor. Cada pista esta formada por uno o más cluster.

****

**Sector:**

Las pistas están divididas en sectores, el número de sectores es variable. Un sector es la unidad básica de almacenamiento de datos sobre los discos duros. Los discos duros almacenan los datos en pedazos gruesos llamados sectores, la mayoría de los discos duros usan sectores de 512 bytes cada uno. Comúnmente es la controladora del disco duro quien determina el tamaño de un sector en el momento en que el disco es formateado, en cambio en algunos [modelos](http://www.monografias.com/trabajos/adolmodin/adolmodin.shtml) de disco duro se permite especificar el tamaño de un sector.



**Cluster:**

Es un [grupo](http://www.monografias.com/trabajos14/dinamica-grupos/dinamica-grupos.shtml) de sectores, cuyo tamaño depende de la capacidad del disco.

A continuación se [muestra](http://www.monografias.com/trabajos11/tebas/tebas.shtml) una tabla que representa esta relación:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tamaño del Driver** | **Tipo de FAT**  **(bits)** | **Sectores por cluster** | **Tamaño del Cluster (kb)** |
| **0-15** | 12 | 8 | 4 |
| **16-127** | 16 | 4 | 2 |
| **128-255** | 16 | 8 | 4 |
| **256-511** | 16 | 16 | 8 |
| **512-1023** | 16 | 32 | 16 |
| **1024-2048** | 16 | 64 | 32 |

## GEOMETRÍA DEL DISCO DURO:

Ahora vamos a ver [la organización](http://www.monografias.com/trabajos6/napro/napro.shtml) [electrónica](http://www.monografias.com/trabajos5/electro/electro.shtml) de cualquier disco duro según el número físico real de platos, cabezas, pistas y sectores:

* Se sabe que el disco duro tiene una cabeza de lectura/escritura para cada cara de un plato, entonces si se sabe el numero de cabezas que hay en un disco duro automáticamente se sabe el numero de platos que contiene y viceversa.
* El número de pistas varia según el tipo de disco duro, para los discos duros antiguos el numero de pista era de 305 en cambio los discos duros más nuevos pueden tener 16000 pistas o más.
* El número de pistas por superficie es igual al número de cilindros. Al multiplicar el número de cabezas con el número de cilindros se sabe el número de pistas del disco.
* El número de sectores varía según el tipo de disco duro, para los discos duros antiguos el número de sectores era de 8 en cambio para los discos duros más modernos es de 60 sectores o más.

## ESTRUCTURA [LÓGICA](http://www.monografias.com/trabajos15/logica-metodologia/logica-metodologia.shtml) DE UN DISCO DURO:

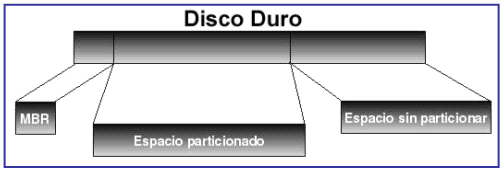
La [estructura](http://www.monografias.com/trabajos15/todorov/todorov.shtml#INTRO) lógica de un disco duro esta formado por:

* Sector de arranque.
* Espacio particionado.
* Espacio sin particionar.

**Sector de arranque:** Es el primer sector de un disco duro en él se almacena la tabla de particiones y un [programa](http://www.monografias.com/Computacion/Programacion/) pequeño llamado Master Boot. Este programa se encarga de leer la tabla de particiones y ceder el control al sector de arranque de la partición activa, en caso de que no existiese partición activa mostraría un mensaje de error.

**Espacio particionado:** Es el espacio del disco que ha sido asignado a alguna partición.

**Espacio sin particionar:** Es el espacio del disco que no ha sido asignado a ninguna partición.



* A su vez la estructura lógica de los discos duros internamente se pueden dividir en varios volúmenes homogéneos dentro de cada [volumen](http://www.monografias.com/trabajos5/volfi/volfi.shtml) se encuentran una estructura que bajo el sistema operativo [MS-DOS](http://www.monografias.com/trabajos/manualdos/manualdos.shtml) es el siguiente:

|  |
| --- |
| **Sector de arranque (BOOT).** |
| **Tabla de asignación de ficheros (FAT)** |
| **Una o más copias de la FAT** |
| **Directorio raíz.** |
| **Zona de datos para**[**archivos**](http://www.monografias.com/trabajos7/arch/arch.shtml)**y subdirectorios.** |

Cada zona del volumen acoge estructuras de datos del sistema de archivos y también los diferentes archivos y subdirectorios. No es posible decir el tamaño de las diferentes estructuras ya que se adaptan al tamaño del volumen correspondiente.

A continuación vamos a definir cada una de las estructuras mostrada en el cuadro.

**1.-Sector de arranque (BOOT):** En el sector de arranque se encuentra la información hacerca de la estructura de volumen y sobre todo del BOOTSTRAP-LOADER, mediante el cual se puede arrancar el PC desde el DOS. Al formatear un volumen el BOOT se crea siempre como primer sector del volumen para que sea fácil su localización por el DOS.

**2.-Tabla de asignación de ficheros (FAT):** La FAT se encarga de informar al DOS que sectores del volumen quedan libres, esto es por si el DOS quiere crear nuevos archivos o ampliar archivos que ya existen. Cada entrada a la tabla se corresponde con un número determinado de sectores que son adyacentes lógicamente en el volumen.

**3.-Uno o más copias de la FAT:** El DOS permite a los [programas](http://www.monografias.com/Computacion/Programacion/) que hacen el formateo crear una o varias copias idénticas de la FAT, esto va a ofrecer la ventaja de que se pueda sustituir la FAT primaria en caso de que una de sus copias este defectuosa y así poder evitar la perdida de datos.

**4.-Directorio Raíz:** El directorio raíz representa una estructura de datos [estática](http://www.monografias.com/trabajos5/estat/estat.shtml), es decir, no crece aún si se guardan más archivos o subdirectorios. El tamaño del directorio raíz esta en relación al volumen, es por eso que la cantidad máxima de entradas se limita por el tamaño del directorio raíz que se fija en el sector de arranque.

**5.-Zona de datos para archivos y subdirectorios:** Es la parte del disco duro donde se almacenan los datos de un [archivo](http://www.monografias.com/trabajos7/arch/arch.shtml). Esta zona depende casi en su totalidad de las interrelaciones entre las estructuras de datos que forman el sistema de archivos del DOS y del camino que se lleva desde la FAT hacia los diferentes sectores de un archivo.

1. **¿Qué es la FAT y cómo está organizada?**

Tabla de asignación de archivos, comúnmente conocido como FAT (del inglés file allocation table), es un sistema de archivos desarrollado para MS-DOS, así como el sistema de archivos principal de las ediciones no empresariales de Microsoft Windows hasta Windows Me.

FAT es relativamente sencillo. A causa de ello, es un formato popular para disquetes admitido prácticamente por todos los sistemas operativos existentes para computadora personal. Se utiliza como mecanismo de intercambio de datos entre sistemas operativos distintos que coexisten en la misma computadora, lo que se conoce como entorno multiarranque. También se utiliza en tarjetas de memoria y dispositivos similares.

## Estructura, el sistema de archivos FAT se compone de cuatro secciones:

**El sector de arranque:** Siempre es el primer sector de la partición (volumen) e incluye información básica, punteros a las demás secciones, y la dirección de la rutina de arranque del sistema operativo.

**La región FAT:** Contiene dos copias de la tabla de asignación de archivos (por motivos de seguridad). Estos son mapas de la partición, indicando qué clusters están ocupados por los archivos.

**La región del directorio raíz**: Es el índice principal de carpetas y archivos.

**La región de datos:** Es el lugar donde se almacena el contenido de archivos y carpetas. Por tanto, ocupa casi toda la partición. El tamaño de cualquier archivo o carpeta puede ser ampliado siempre que queden suficientes clusters libres. Cada cluster está enlazado con el siguiente mediante un puntero. Si un determinado cluster no se ocupa por completo, su espacio remanente se desperdicia.

Una partición se divide en un conjunto de clusters de idéntico tamaño. Son pequeños bloques discontinuos. El tamaño del clúster depende de la variante de FAT utilizada. Varía entre 2 y 32 kilobytes. Cada archivo ocupa uno o más clusters en función de su tamaño. De manera que un archivo queda representado por una cadena secuencial de clusters (una lista enlazada). Cada clúster de la cadena no tiene por qué ser adyacente al anterior. Esto es lo que provoca la fragmentación.

1. **¿Cuál es la cantidad de copias de la FAT que tiene su disco de trabajo, en todos los discos es igual?**

Dos copias. Es lo mismos para todos los discos particionados en FAT.

1. **¿Qué significa una FAT de 12, 16 o 32 bits? Explicar**

FAT12: La versión inicial se de FAT se conoce ahora como FAT12. Es un sistema de archivos para disquete, por lo que tiene varias limitaciones:

Las direcciones de bloque solamente contienen 12 bits. Esto complica la implementación.

El tamaño del disco se almacena como una cuenta de 16 bits expresada en sectores, lo que limita el espacio manejable a 32 megabytes.

FAT16: En 1987 apareció lo que hoy se conoce como el formato FAT16. Se eliminó el contador de sectores de 16 bits. El tamaño de la partición ahora estaba limitado por la cuenta de sectores por clúster, que era de 8 bits. Esto obligaba a usar clusters de 32 KiB con los usuales 512 bytes por sector. Así que el límite definitivo de FAT16 se situó en los 90 GiB.

Esta mejora estuvo disponible en 1988 gracias a MS-DOS 4.0. Mucho más tarde, Windows XP aumentó el tamaño máximo del cluster a 64 kilobytes gracias al "truco" de considerar la cuenta de clusters como un entero sin signo. No obstante, el formato resultante no era compatible con otras implementaciones de la época, y además, generaba más fragmentación interna (se ocupaban clusters enteros aunque solamente se precisaran unos pocos bytes).

FAT32:  fue la respuesta para superar el límite de tamaño de FAT16 al mismo tiempo que se mantenía la compatibilidad con MS-DOS en modo real. Microsoft decidió implementar una nueva generación de FATutilizando direcciones de cluster de 32 bits (aunque sólo 28 de esos bits se utilizaban realmente).

En teoría, esto debería permitir aproximadamente 268.435.538 clusters, arrojando tamaños de almacenamiento cercanos a los 8 [TiB](https://es.wikipedia.org/wiki/Tebibyte" \o "Tebibyte). Sin embargo, debido a limitaciones en la utilidad ScanDisk de Microsoft, no se permite que FAT32 crezca más allá de 4.177.920 clusters por partición (es decir, unos 124 [GiB](https://es.wikipedia.org/wiki/Gibibyte" \o "Gibibyte)). Posteriormente, Windows 2000 y XP situaron el límite de FAT32 en los 32 GiB. Microsoft afirma que es una decisión de diseño, sin embargo, es capaz de leer particiones mayores creadas por otros medios.

FAT32 apareció por primera vez en Windows 95 OSR2. Era necesario reformatear para usar las ventajas de FAT32. Curiosamente, [DriveSpace](https://es.wikipedia.org/wiki/DriveSpace" \o "DriveSpace) 3 (incluido con Windows 95 y 98) no lo soportaba. Windows 98 incorporó una herramienta para convertir de FAT16 a FAT32 sin pérdida de los datos. Este soporte no estuvo disponible en la línea empresarial hasta [Windows 2000](https://es.wikipedia.org/wiki/Windows_2000).

El tamaño máximo de un archivo en FAT32 es 4 GiB (232−1 bytes), lo que resulta engorroso para aplicaciones de captura y edición de video, ya que los archivos generados por éstas superan fácilmente ese límite.

1. **- ¿Cuáles son las opciones para particionar un disco, que proporciona el comando FDISK del sistema operativo? ¿Qué limitaciones tiene? ¿Cuál es la diferencia con el comando FORMAT?**

Fdisk es un software que esta disponible para varios sistemas operativos, el cual permite dividir en forma lógica un disco duro, siendo denominado este nuevo espacio como partición.

La descripción de las particiones se guarda en la tabla de particiones que se localiza en el sector 0 de cada disco.

DOS

IBM introduce fdisk, llamado (“Fixed Disk Setup Program version 1.00”) con el lanzamiento del IBM PC/XT, en marzo de 1983, el primer PC para almacenar los datos en un disco duro , y el IBM Personal Computer DOS versión 2.0 IBM PC DOS

La versión 1.0 podía ser usada para crear una partición FAT12 DOS, borrar, cambiar la partición activa o mostrar datos de la partición. El registro de inicio maestro soporta hasta cuatro particiones, y los otros tres estaban destinados para otros sistemas operativos tales como CP/M-86 y Xenix, que se espera que sus utilidades de particionamiento propios como fdisk funcionase pero no los soportaba. En agosto de 1984, PC DOS 3.0 añade particiones FAT16 para soportar discos duros más grandes con mayor eficiencia.

En abril de 1987, PC DOS/fdisk 3.30 añade soporte para particiones extendidas, que podría albergar hasta 23 particiones "lógicas" o unidades de volumen.

Soporte para FAT16B se añadió con Compaq MS-DOS 3.31, y más tarde se convirtió disponible con MS-DOS/PC DOS 4.0.

La mayoría de los programas fdisk de DOS, incluyendo el programa fdisk que viene con el original de Windows 95, sólo son capaces de crear particiones FAT de tipos de FAT12, FAT16 y FAT16B.

Windows

Un derivado del fdisk de MS-DOS se suministra con Windows 95, Windows 98, Windows Me y más tarde. Sólo las versiones de fdisk envío con Windows 95B o temprano son capaces de manipular las particiones FAT32. Windows 2000 y versiones posteriores no usan fdisk, tienen la función de Administrador de discos lógicos, así como DiskPart. La opción número 1 sirve para crear una partición primaria de DOS, por ejemplo para instalar

Opciones en sistemas DOS/WIN95/98/ME/2000 Y XP FAT,

La opción número 1 sirve para crear una partición primaria de DOS, por ejemplo para instalar Windows 95, 98 o ME (también para 2000 y XP pero es mejor formatear en NTFS para instalar estos sistemas operativos).

La opción número 2 es establecer una partición activa, la cual nos sirve para indicarle a la bios en que partición debe buscar primero el sistema operativo, es decir, daremos de alta la partición en la que pensemos instalar el sistema operativo.

La opción número 3 nos sirve cuando por algún motivo queremos eliminar de una partición lógica de DOS.

La opción número 4 nos muestra un informe detallado de todas las particiones del disco duro como la etiqueta de volumen, sistema de archivos, tamaño del disco (siempre expresado en MB).

La opción número 5 solo aparece cuando tenemos en la PC otra unidad de Disco Duro instalada y esta opción nos permite alternar entre un Disco Duro u otro para trabajar sobre ellos.

Fdisk no reconoce particiones NTFS ya que este sistema de archivos salió a la luz con Windows NT y a partir de ahí se implantó en los sistemas operativos Windows 2000 y Windows XP.

Sin embargo, aunque Fdisk no trabaje con el sistema de archivos NTFS nos puede mostrar particiones (en este caso) NTFS pero identificándolas como Particiones o Sistemas de Archivos Non-DOS.

1. **Defina el tiempo de acceso y ancho de banda de un disco.**

El tiempo de acceso a disco está determinado por la suma del tiempo de posicionamiento y el retraso rotacional.

El ancho de banda del disco es el número total de bytes transferidos, dividido entre el tiempo total transcurrido entre la primera solicitud de [servicio](http://www.monografias.com/trabajos14/verific-servicios/verific-servicios.shtml) y la finalización de la última transferencia.

1. **¿Cuál es el objetivo del Sistema Operativo en cuanto a la planificación de discos?**

Es claro que una política de planificación debe intentar maximizar la capacidad de ejecución, el número de peticiones servidas por unidad de tiempo. Debido a la planificación se reduce el tiempo desperdiciado en las esperas de las búsquedas, con lo que se puede mejorar la media de los tiempos de respuesta. Si una política de planeación no intenta más que maximizar la capacidad de ejecución sin minimizar al mismo tiempo la varianza, podría procesar peticiones. (Cuanto menor es la varianza, más predecible es el sistema).

El objetivo es reducir los tiempos de acceso en la lectura o escritura de los datos. Además del tiempo de acceso y del tiempo de transferencia, existen algunos retrasos en las colas que normalmente tienen asociada una operación de E/S a disco. Cuando un proceso emite una solicitud de E/S, primero debe esperar en una cola a que el dispositivo esté disponible. En ese momento, el dispositivo queda asignado al proceso. Si el dispositivo comparte un único canal de E/S o un conjunto de canales con otras unidades del disco, puede producirse una espera adicional hasta que el canal esté disponible. En ese punto se realizará la búsqueda con que comienza el acceso al disco.

1. **¿cuáles son los distintos algoritmos de planificación de disco? Explique cada uno de ellos y analice las ventajas y desventajas de uno con respecto a los otros.**

## Planificación FCFS

La forma más sencilla de planificación de disco es, desde luego, el ***servicio por orden de llegada (FCFS, first come, first served).*** No proporciona el servicio más rápido.

La planificación FCFS es justa en el sentido de que una vez que llega una petición, se fija su lugar dentro de la cola de espera. Una petición, se fija su lugar dentro de la cola de espera. Una petición no puede ser desplazada por la llegada de otra con prioridad más alta.

La FCFS es aceptable cuando la carga en un disco es ligera. Pero a medida que crece la carga, la FCFS tiende a saturar el dispositivo y los tiempos de respuesta se incrementan. La FCFS ofrece una varianza pequeña, pero tiene tiempos de espera muy grandes.

## Planificación SSTF

Parece razonable atender todas las solicitudes cercanas a la posición actual de la cabeza antes de mover la cabeza a una posición lejana para atender otras solicitudes. Este supuesto es la base del algoritmo de ***tiempo de búsqueda más corto primero (SSTF, shortest-seek-time-first),*** que selecciona la solicitud que tiene el menor tiempo de búsqueda a partir de la posición actual de la cabeza.

En esta política la petición que da por resultado la distancia de búsqueda más corta (y, con esto, el tiempo de búsqueda más corto) es la siguiente en ser servida, aunque esa petición no sea la primera en la cola.

Los patrones de búsqueda SSTF tienden a estar muy relocalizados, dando como resultado que las pistas internas y externas reciban un servicio pobre, en comparación con las pistas del centro. La SSTF es útil en sistemas de procesamiento por lotes, en los cuales la capacidad de ejecución es lo más importante. Pero la alta varianza de los tiempos de respuesta (es decir, su falta de predecibilidad) lo hace inaceptable para los sistemas interactivos.

Este algoritmo mejora sustancialmente el [desempeño](http://www.monografias.com/trabajos15/indicad-evaluacion/indicad-evaluacion.shtml).

La planificación SSTF es en esencia una forma de planificación de [trabajo](http://www.monografias.com/trabajos34/el-trabajo/el-trabajo.shtml) más corto primero (SJF) y, al igual que la planificación SFJ, puede cause inanición de algunas solicitudes.

Aunque el algoritmo SSTF representa una mejora sustancial respecto al algoritmo FCFS, no es óptimo.

## Planificación SCAN

En el algoritmo SCAN, el brazo del disco parte de un extremo del disco y se mueve hacia el otro, atendiendo las solicitudes a medida que llega a cada cilindro, hasta llegar al otro extremo del disco. Ahí, la dirección de movimiento de la cabeza se invierte, y continúa la [atención](http://www.monografias.com/trabajos14/deficitsuperavit/deficitsuperavit.shtml). La cabeza barre continuamente el disco de un lado a otro.

Esta política, desarrollada por Denning, opera como SSTF, excepto que selecciona la petición que da como resultado la distancia de búsqueda más corto en una dirección seleccionada. La SCAN no cambia de dirección hasta que ha alcanzado el cilindro exterior o hasta que ya NO haya peticiones pendientes en la dirección con preferencia.

La SCAN se comporta de manera parecida al SSTF desde el punto de vista de la mejora en la capacidad de ejecución y de la media de los tiempos de respuesta, pero elimina mucha de la [discriminación](http://www.monografias.com/trabajos28/nociones-fundamentales-discriminacion/nociones-fundamentales-discriminacion.shtml) inherente a los esquemas SSTF y ofrece una varianza menor.

El algoritmo SCAN también se conoce como ***algoritmo de elevador***, ya que el brazo del disco se comporta igual que el elevador de un edificio, que atiende primero todas las solicitudes para subir y luego cambia de dirección para atender las solicitudes de abajo.

**Planificacion SCAN de n-pasos**

En esta estrategia, el brazo del disco se mueve de un lado a otro como en SCAN, pero sólo da servicio a aquellas peticiones que se encuentran en espera cuando comienza un recorrido particular. Las peticiones que llegan durante un recorrido son agrupadas y ordenadas para un servicio óptimo durante el recorrido de regreso.

La SCAN de n-pasos ofrece un buen rendimiento de la capacidad de ejecución y de la media de los tiempos de respuesta. Su característica más significativa es una menor varianza de los tiempos de respuesta que las planeaciones SSTF y SCAN convencionales. La SCAN de n-pasos evita la posibilidad de postergación indefinida que tiene lugar si un gran número de peticiones que llegan al cilindro que está siendo servido y guarda estas peticiones para ser servidas durante el recorrido de regreso.

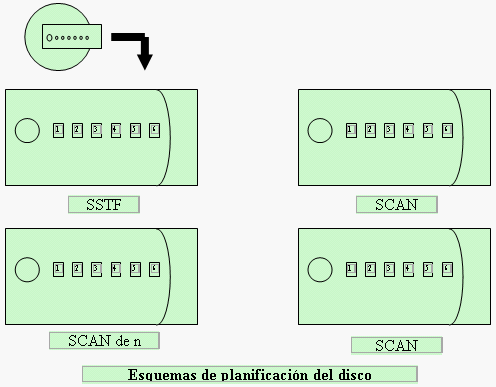
## Planificacion C-SCAN

La planificación SCAN circular (C-SCAN) es una variante de SCAN diseñada para dar un tiempo de espera más uniforme. Al igual que SCAN, C-SCAN mueve la cabeza de un extremo del disco al otro, atendiendo las solicitudes en el camino, sólo que ahora, cuando la cabeza llega al otro extremo, regresa de inmediato al principio del disco sin atender solicitudes.

El algoritmo de planificación C-SCAN básicamente trata los cilindros como una lista circular que continúa del último cilindro al primero.

En la estrategia C-SCAN, el brazo se mueve del cilindro exterior al interior, sirviendo a las peticiones con menor tiempo de búsqueda. Cuando el brazo ha completado su recorrido hacia adentro, salta a la petición más cercana al cilindro exterior y a continuación reanuda su recorrido hacia adentro procesando peticiones.

La C-SCAN puede implementarse de forma que las peticiones que llegan durante un recorrido sean servidos en el siguiente. De esta forma C-SCAN elimina completamente la [discriminación](http://www.monografias.com/trabajos/discriminacion/discriminacion.shtml) contra las peticiones para los cilindros exterior e interior. Tiene una varianza de los tiempos de respuesta muy pequeña.



## Planificación LOOK

En la práctica, ningunos de estos dos algoritmos se implementan así. Por lo regular, el brazo sólo llega hasta la última solicitud en cada dirección y luego cambia de dirección inmediatamente, sin primero ir hasta el extremo del disco. Estas versionas de SCAN y C-SCAN se llaman LOOK y C-LOOK, porque miran si hay una solicitud antes de continuar en una dirección dada.

## Esquema Eschenbach

En esta estrategia el movimiento del brazo del disco es circular como en C-SCAN, pero con varias excepciones importantes. Cada cilindro es servido exactamente por una pista completa de información, haya o no peticiones para este cilindro. Las peticiones se redondean a ser servidas dentro de un cilindro para tomar ventaja de la posición rotacional, pero si dos peticiones traslapan posiciones de sectores dentro de un cilindro, solamente se sirve una en el movimiento actual del brazo del disco.

Con carga baja, la política SCAN es la mejor, mientras que con cargas medias y pesadas, la C-SCAN produce los mejores resultados. La C-SCAN con optimización rotacional maneja de manera efectiva condiciones de carga muy pesadas y ha demostrado ser mejor que el esquema Eschenbach bajo todas las condiciones de carga.

**10- ¿Cuáles serían los criterios para escoger un determinado algoritmo de planificación?**

SSTF es común y tiene un atractivo natural. SCAN y C-SAN funcionan mejor en sistemas en los que la carga sobre el disco es intensa, y no son propensos al problema de inanición.

Con cualquier algoritmo de planificación, es desempeño depende en gran medida del número y los tipos de solicitudes.

El método de asignación de [archivos](http://www.monografias.com/trabajos7/arch/arch.shtml) puede influir mucho sobre las solicitudes de servicio de disco. Un [programa](http://www.monografias.com/Computacion/Programacion/) que lee un [archivo](http://www.monografias.com/trabajos7/arch/arch.shtml) que se asignó contiguamente generará varias solicitudes cercanas entre sí en el disco, y el movimiento de la cabeza será limitado. Un archivo enlazado o indexado, en [cambio](http://www.monografias.com/trabajos2/mercambiario/mercambiario.shtml), podría incluir bloques muy dispersos en el disco, y el movimiento de la cabeza sería mayor.

La colocación de los directorios y los bloques índice en caches de la memoria principal puede ayudar también a reducir el movimiento del brazo del disco, sobre todo para las solicitudes de lectura.

Es conveniente que el algoritmo de planificación de disco se escriba como módulo independiente del sistema operativo, a fin de [poder](http://www.monografias.com/trabajos35/el-poder/el-poder.shtml) sustituirlo por un algoritmo diferente si es necesario. SSTF y LOOK son opciones razonables para el algoritmo por omisión.

Los algoritmos descritos sólo consideran las distancias de búsqueda. En los discos modernos, la latencia rotacional puede ser casi tan grande como el tiempo de búsqueda promedio, pero es difícil para el sistema operativo reducir la latencia rotacional mediante planificación, porque los discos modernos no revelan la ubicación [física](http://www.monografias.com/Fisica/index.shtml) de los bloques lógicos.

**11- Administración del espacio de intercambio: ¿Qué es el sistema de intercambio para qué se utiliza? ¿Dónde se ubica? ¿Cómo se administra?**

La mayoría de los [sistemas operativos](https://es.wikipedia.org/wiki/Sistema_operativo) modernos poseen un mecanismo llamado [memoria virtual](https://es.wikipedia.org/wiki/Memoria_virtual), que permite hacer creer a los programas que tienen más memoria que la disponible realmente. Como en realidad no se tiene físicamente toda esa memoria, algunos [procesos](https://es.wikipedia.org/wiki/Proceso_(inform%C3%A1tica)) no podrán ser ubicados en la [memoria RAM](https://es.wikipedia.org/wiki/Memoria_de_acceso_aleatorio).

En este caso es cuando es útil el espacio de intercambio: el [sistema operativo](https://es.wikipedia.org/wiki/Sistema_operativo) puede buscar un proceso poco activo, y moverlo al área de intercambio (el [disco duro](https://es.wikipedia.org/wiki/Disco_duro)) y de esa forma liberar la memoria principal para cargar otros procesos. Mientras no haga falta, el proceso extraído de memoria puede quedarse en el disco, ya que ahí no utiliza memoria física. Cuando sea necesario, el sistema vuelve a hacer un intercambio, pasándolo del disco a memoria RAM. Es un proceso lento (comparado con usar sólo la memoria RAM), pero permite dar la impresión de que hay más memoria disponible.

**Posibles ubicaciones**

En los [sistemas operativos](https://es.wikipedia.org/wiki/Sistema_operativo) se puede usar como área de intercambio un [fichero](https://es.wikipedia.org/wiki/Fichero) o una [partición](https://es.wikipedia.org/wiki/Partici%C3%B3n_de_disco) (los dos son en realidad parte de un [disco duro](https://es.wikipedia.org/wiki/Disco_duro) o almacenamiento secundario).

**Fichero de intercambio**

Un [fichero](https://es.wikipedia.org/wiki/Fichero) en blanco puede prepararse para ser usado como área de intercambio. Esto tiene una gran ventaja:

es fácil de crear, borrar, ampliar o reducir, según se crea necesario (a diferencia de una partición)

Pero también alguna desventaja:

le afecta la [fragmentación](https://es.wikipedia.org/wiki/Desfragmentaci%C3%B3n), ya que se encuentra dentro de un [sistema de ficheros](https://es.wikipedia.org/wiki/Sistema_de_ficheros)

El problema de la fragmentación no es grave, ya que el espacio de intercambio no siempre es accedido de forma secuencial, sino directa (en oposición a la secuencial).

**Partición de intercambio**

También se puede dedicar una [partición](https://es.wikipedia.org/wiki/Partici%C3%B3n_de_disco) entera del [disco duro](https://es.wikipedia.org/wiki/Disco_duro) (o el disco completo) como área de intercambio. Ventajas:

se puede conseguir mejor rendimiento si se coloca la partición en la zona más rápida del disco, que es al principio (en los [cilindros](https://es.wikipedia.org/wiki/Cilindro) exteriores del disco los datos están más separados y el posicionamiento de la cabeza lectora y la localización de la información es más lenta);

no hay problemas de [fragmentación](https://es.wikipedia.org/wiki/Desfragmentaci%C3%B3n)

no hay que usar ningún [sistema de ficheros](https://es.wikipedia.org/wiki/Sistema_de_ficheros) en concreto

Tiene pocas desventajas:

crear una partición es un proceso algo difícil; de todas formas, si se elige bien el tamaño de la partición, no hará falta ningún cambio en el futuro

la partición siempre ocupará el mismo espacio, aunque no se esté usando al 100%

**12- ¿Qué se entiende por confiabilidad de los discos?**

**13- ¿Cuáles son las tres operaciones que tiene que realizar el hardware de una unidad de disco ante una petición de acceso a un bloque determinado?**

Ante una petición de acceso a un bloque determinado, el hardware del disco debe realizar,

fundamentalmente, tres operaciones:

- Mover el brazo buscando la pista o cilindro correspondiente (búsqueda).

- Esperar a que el bloque se sitúe frente á la cabeza de lectura/escritura (latencia).

- Transmitir el bloque deseado (transmisión).

**14- Concepto de archivo. Atributo de archivo.**

Desde el punto de vista de los usuarios, los archivos son grupos de informaciones

relacionadas entre sí sobre los que puede realizar diversas operaciones (lectura, escritura,

actualización, borrado, etc.). El sistema será el responsable de realizar dichas operaciones,

adecuándolas a las características de los dispositivos físicos utilizados.

**15- Operaciones con archivos.**

Los archivos informáticos son el medio de que disponemos para almacenar información no volátil en un dispositivo de almacenamiento. Los Sistemas de archivos de que disponen los sistemas operativos disponen de mecanismos para que un usuario pueda manipular los archivos (seleccionar, editar, ejecutar, borrar, ...). Desde el punto de vista de un programador un archivo es un medio para poder leer datos de entrada para su programa o donde poder guardar los resultados de su ejecución. Todo lenguaje de programación debe disponer de algún mecanismo para que el programador pueda manipular archivos desde un programa. Estos mecanismos pueden ser más o menos sofisticados o versátiles dependiendo del lenguaje de programación que estemos considerando, pero deben haber unas funciones básicas para poder acceder a un archivo, estas son:

Lectura (consulta).- Esta operación consiste el leer la información contenida en fichero sin alterarla.

Escritura (modificación).- Consiste en actualizar el contenido del fichero bien añadiéndole nuevos datos o borrando parte de los que contenía.

Apertura.- Antes de acceder a un fichero, tanto para consultar como para actualizar su información, es necesario abrirlo. Esta operación se debe realizar previamente a las operaciones de lectura o escritura.

Cierre.- Cuando se ha terminado de consultar o modificar un fichero, por lo general, del mismo modo que se tuvo que abrir para realizar alguna operación de lectura/escritura sobre él, éste deberá ser cerrado.

**16- Tipos de archivos.**

**17- Método de acceso a los archivos.**

**18- Estructura de directorios. Directorio de uno y dos niveles, con estructura de árbol. ¿Qué es un sistema de archivo y cuáles son sus misiones?**

20- ¿Cuáles son las estrategias que utili