

## Apuntes clase filesystem FASTA

---

Cada **partición** de una computadora, celular, etc.. tendrá su propio file system, por ende, dentro de un dispositivo se podrían manejar varios file systems a la vez, y no uno solo.

El SO trata a los archivos como bloques lógicos (**clusters**) de bytes (a cada archivo le asigna 'x' cantidad de clusters de por ejemplo 4KB (<https://support.microsoft.com/es-ar/help/140365/default-cluster-size-for-ntfs-fat-and-exfat>)).

Luego en el disco duro están los bloques físicos (**sectores**) que tienen un tamaño fijo generalmente de **512 Bytes** (no se puede leer menos que eso).

**Nota:** este tamaño viene "de fábrica" con el dispositivo, no se puede cambiar; lo que sí podemos cambiar es el tamaño del cluster, ya que es algo lógico.

---

Un **cluster** es una secuencia contigua de sectores del disco duro ([https://es.wikipedia.org/wiki/Cl%C3%BAster\\_\(sistema\\_de\\_archivos\)](https://es.wikipedia.org/wiki/Cl%C3%BAster_(sistema_de_archivos))).

Cuando el SO le asigna a un archivo 1 cluster de 4KB (4.096 bytes), significa que 8 sectores de 512 bytes del disco duro estarán asignados a ese archivo.

**Nota:** un clúster es lo mínimo que el SO le puede asignar a un archivo, de ahí a medida que se vaya llenando el archivo, le irá asignando otro clúster más y así. Si el archivo pesa menos que los 4k de bloque logico, igualmente el espacio quedará como 'ocupado' por ese archivo. (y creo que cuando quiere leer un archivo, mínimamente debe leer 1 cluster: **preguntar**)

---

Todo lo que quede **grabado en el disco**, quedará ahí a menos que se sobrescriban los datos. Es decir, cuando borramos un archivo de nuestro sistema, los datos seguirán en el disco hasta que por ejemplo otro archivo que requiera espacio y use esos sectores, los sobrescriba ([estamos haciendo una eliminación 'lógica' y no 'física'](#)).

Por ejemplo, en FAT32 se pone en **0x00** toda la cadena de clusters de la *FAT* y en el *directory entry* se cambia el primer caracter del nombre del archivo por **0xE5**.

**PREGUNTAR:** ¿ENTONCES ESE DIRECTORY ENTRY QUEDARÁ "OCUPANDO" ESPACIO POR SIEMPRE NO? o sea, tanto los registros de la FAT como los clusters correspondientes al archivo borrado, ¿podrán ser sobrescritos por otro archivo, pero no su directory entry ??.

**Nota:** solo el área de datos de un filesystem (archivos y directorios) está organizada en 'clusters' (por lo menos así es en FAT32), o sea, solo ahí se emplea el concepto de 'cluster' (a los archivos se le asignan clusters, que son secuencias de sectores). En las regiones más "administrativas" (como el area de boot o el area de fats en FAT32) se suele emplear directamente el término 'sector' para ir moviendonos por esas áreas. **PREGUNTAR BIEN.**

---

**Sistema de archivos en sentido amplio:** especificaciones/restricciones/diseño del manejo de archivos en el sistema (FAT, NTFS, EXT...). Sería como una clase en POO.

**Sistema de archivos en sentido estricto:** es una instancia de un SdA en sentido amplio, es todo el conjunto de archivos/información de nuestra computadora. Sería como un objeto en POO.

Una partición de un dispositivo de almacenamiento puede verse como un **gran archivo** (realmente el dispositivo entero se ve así, y cada partición es una región de él) que está estructurado de cierta manera según su sistema de archivos (FAT tiene una estructura, NTFS tiene otra, etc...).

Por ejemplo, si una partición tiene un file system FAT32, al principio de este gran archivo estará el area de boot, luego la region de las 2 FAT, y luego el área de directorios y datos (acá estarán guardados todos nuestros datos). Todo esto lo podemos ver abriendo una partición de algún dispositivo en un editor hexadecimal (como **WinHex**).

**Encargado de:**

- Organización de la estructura de todo el **dispositivo** -> **MBR** o **GPT**
- Organización de la estructura de una **partición** -> **Filesystem** (FAT32, NTFS, ext4, etc...)

Cuando '**formateamos**' un dispositivo de almacenamiento (o más bien, una partición de él), justamente lo que se hace es darle formato; se lo estructura en base al file system que tendrá. Si se elige que tenga FAT32, durante el proceso de formato se escribirán todos los datos 'burocráticos' del area de boot, las regiones fat, etc... Además, se definirá el tamaño de asignación de archivos, o sea el tamaño de un 'cluster', entre otras cosas.

(preguntar bien a Bruno estos 2 párrafos) **GOOD ;)**

## EN RESUMEN:

Un **filesystem** es la estructura que tiene una partición de un dispositivo de almacenamiento. Es el medio para poder acceder a los archivos y directorios que haya en una partición. Existen diversos *filesystems* (FAT32, NTFS, ext4, etc...) y cada uno de ellos tendrá su forma de organizar/administrar una partición, cada uno habrá sido diseñado diferente, con diferentes normativas, convenciones y estructuras.

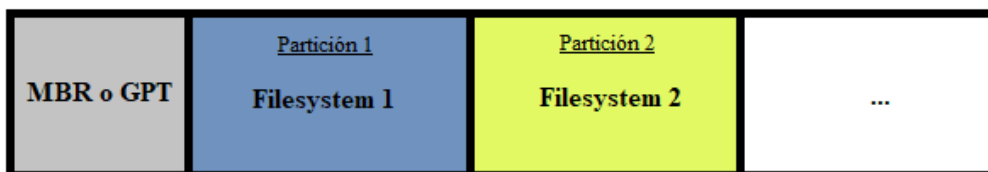
Una **partición** es una división lógica de un dispositivo de almacenamiento; el sistema operativo la considera como si fuera un dispositivo, pero el dispositivo es uno solo, es único. Por ejemplo, el disco duro de una computadora (*hdd*) es un dispositivo de almacenamiento, el cual puede estar particionado de una manera lógica, no física (realmente el disco físico no se “particiona”); esto lo podemos ver en el explorador de Windows con las típicas letras C: y D: (en el caso que tengamos 2 particiones).

Entonces cuando decimos que un *filesystem* se corresponde a una partición, es justamente eso. **Cada partición de un dispositivo de almacenamiento tendrá su correspondiente *filesystem***, los cuales pueden ser distintos entre sí (particion1 -> FAT32 ; particion2 -> NTFS ; etc).

Y por otro lado, también hay un encargado de organizar/estructurar el dispositivo de almacenamiento en su totalidad. Principalmente maneja lo referido a las particiones. Por ejemplo, nos indica en que parte del dispositivo comienza cada una. Esto se hace mediante una tabla de particiones, y hoy en día hay 2 grandes implementaciones de esto: **MBR y GPT**.

Entonces un dispositivo de almacenamiento se podría ver así:

### **Dispositivo de almacenamiento**



**Nota:** El sistema operativo trata a todos los dispositivos de almacenamiento **por igual**, por ende, tanto un *hdd*, como un *ssd*, como un *pendrive*, etc... se podrán particionar, tendrán filesystems, y se usa el concepto de bloque físico: **sector** (generalmente 512 bytes) y bloque lógico: **cluster** (secuencia de sectores) para leer y escribir datos del dispositivo. Es una de sus capas de abstracción.