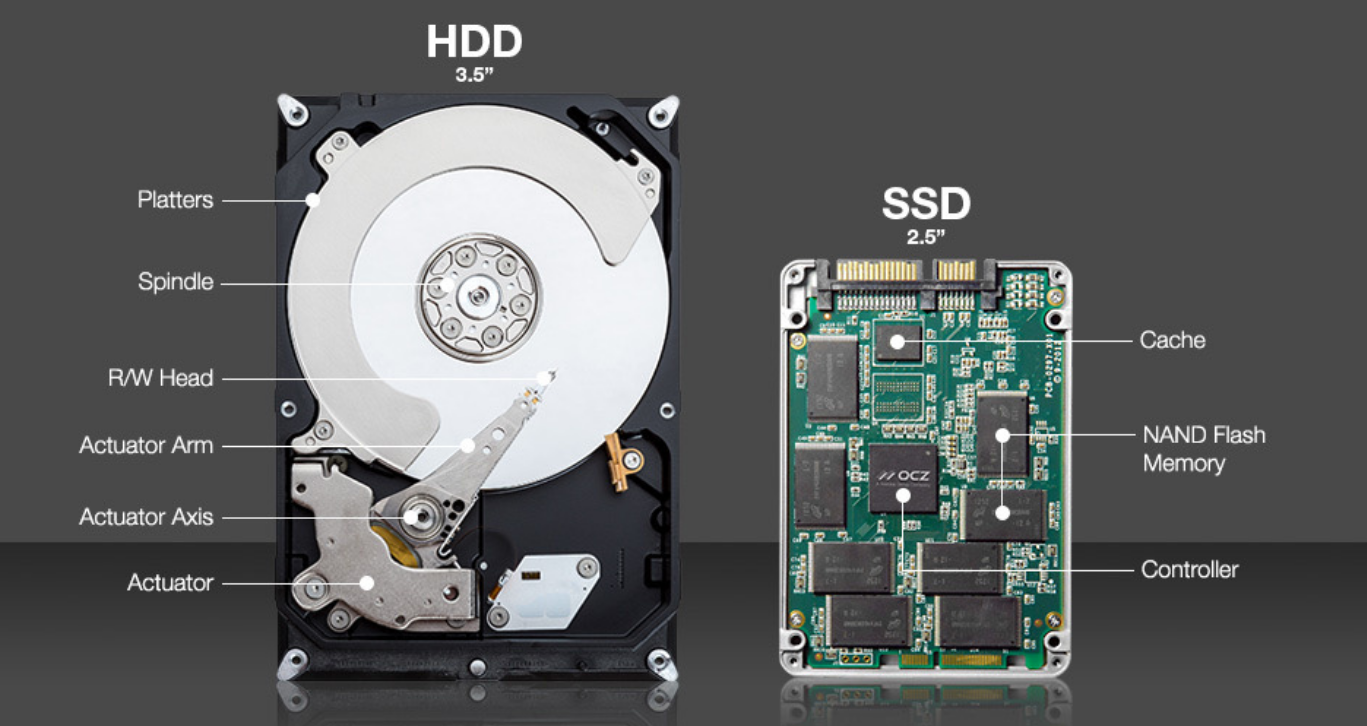
**1.2. TEMA: Sistemas de ficheros**

**1.2.1. Introducción:**

Sistemas de ficheros:

* Actualmente, parte esencial de cualquier sistema informático.
* Todo sysadmin es responsable de asegurar la integridad de los datos que gestiona.

Hard Disk Drive (HDD) vs Solid State Drive (SSD):



Conceptos:

* Disco: Dispositivo físico de almacenamiento.



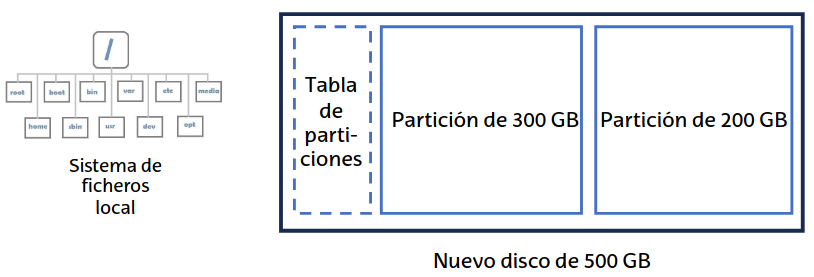
* Partición: Unidad lógica de almacenamiento
  + Organizar disco en 1 partición:



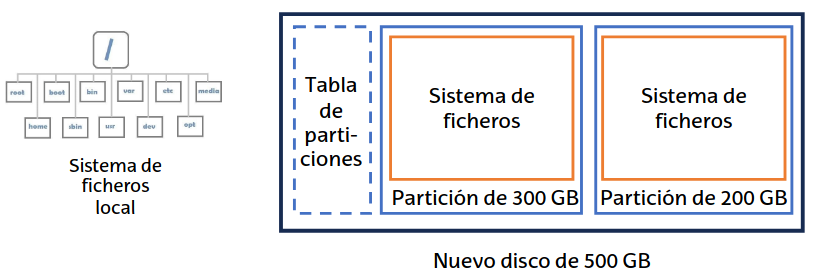
* + Organizar disco en 2 particiones:



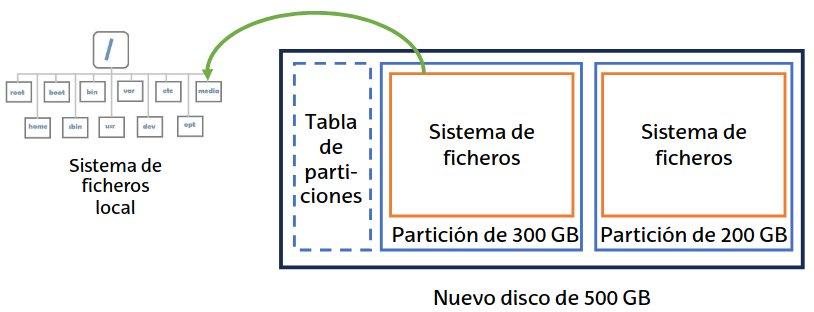
* Tabla de particiones: Esquema que indica cómo se organizan las particiones.
  + Se crea antes de definir las particiones.
  + Generalmente es MBR o GPT.

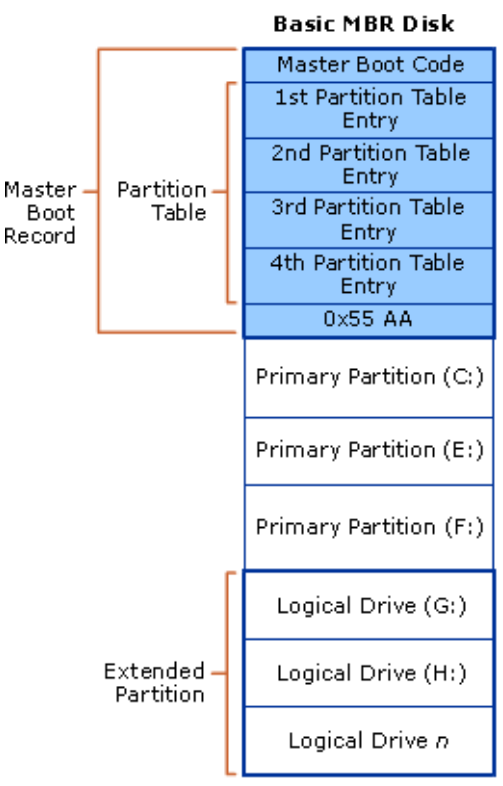


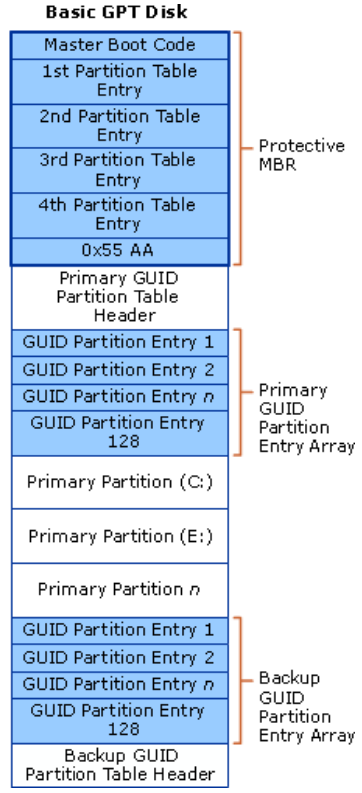
* Sistema de ficheros: Estructura que gestiona el almacenamiento y recuperación de datos.



* Montaje: Enlace de un sistema de ficheros con otro.
  + Ej.: La 1ª partición del nuevo disco está disponible en /media → La carpeta /media sería el punto de montaje.



Master Boot Record (MBR):

* A veces mostrado como DOS (por MS-DOS).
* Introducido en 1983.
* Permite dividir 1 disco en 4 particiones primarias.
* Para crear más particiones:
  + Convertir 1 partición primaria en lógica.
  + Crear particiones extendidas dentro de la lógica.
  + Límite: 23 particiones extendidas.
* Almacena la meta información al comienzo del disco.
* Límite: 2 TB por disco.

GUID Partition Table (GPT):

* Introducido entre 1990 y 2000.
* Permite realizar hasta 128 particiones en 1 disco.
* Almacena la meta-información distribuida por el disco.
* Límite: 9.7 zetabytes por disco.
* No es tan compatible como MBR.
  + Para poder usar un disco GPT para arranque, el

sistema debe ser BIOS UEFI.

**1.2.2. Administración:**

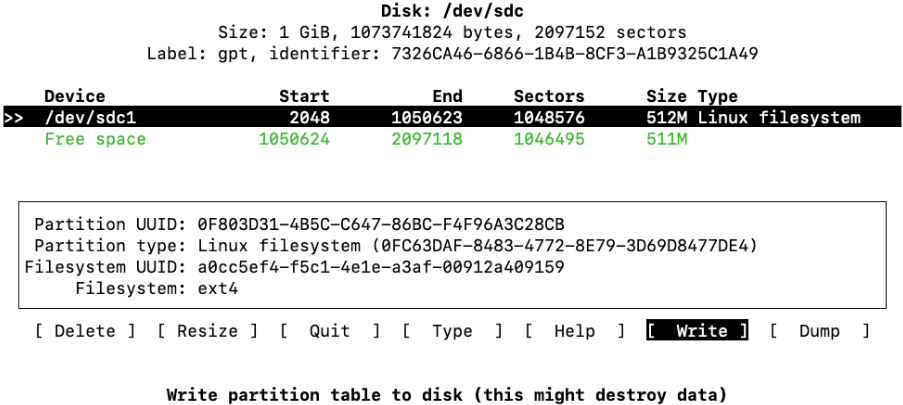
Fichero del dispositivo:

* Fichero del sistema que posibilita que las aplicaciones accedan a un dispositivo.
  + Acceden a través del kernel.
* Localización: /dev
  + Dispositivos SATA: /dev/sdX (X = a, b, …)
    - 1er disco SATA: /dev/sda
    - 2º disco SATA: /dev/sdb
    - …
  + Dispositivos RAID: /dev/mdX
  + Especiales. Ejemplos:
    - /dev/null (nulo)
    - /dev/urandom (números aleatorios)

Particiones:

* Están en /dev, con un número adjunto al nombre del disco.
  + Ej.: El 2º disco SATA con 2 particiones sería:
    - /dev/sdb → Disco.
    - /dev/sdb1 → Partición 1 del disco.
    - /dev/sdb2 → Partición 2 del disco.
* Kernels recientes, el sistema crea un alias para cada partición:
  + Se puede usar cada vez que sea necesario.
  + Evita tener que comprobar nombres después de cada reinicio.
  + Ejemplo: /dev/disk/by-uuid/{UUID} enlaza al /dev/sdXX correspondiente.
    - Listar UUID de cada partición: comando **blkid**.

Manipular particiones:

* Comando **fdisk**:
  + Sintaxis → fdisk <fichero-de-dispositivo>
    - Ej.: fdisk /dev/sda
  + Algunas opciones:
    - p → Ver tabla de particiones de disco.
    - n → Nueva partición.
    - w → Escribir nueva tabla de particiones.
    - q → Salir.
* Comando **cfdisk**:
  + Variante visual de fdisk
  + Más fácil de utilizar
  + No tiene todas las funciones de fdisk.

Formatear una partición:

* Crear un sistema de ficheros en una partición.
* Comando **mkfs**:
  + Crea un sistema de ficheros en una partición.
  + Sintaxis → mkfs.<tipo-de-sistema> <partición>
    - Ej.: mkfs.ext4 /dev/sda3
  + Alternativa: mkfs [-V -t tipo-de-sistema] <partición>
    - Ej.: mkfs –t ext4 /dev/sda3
    - **Se desaconseja el uso de esta forma!**

Montar una partición:

* Por defecto, comando **mount**:
  + Sintaxis → mount <opciones> [fichero-disp] [punto-montaje]
  + Algunas opciones:
    - -t → Tipo sistema.
    - -r → Montar solo en lectura.
  + Ejemplo → mount -t ext4 /dev/sdc1 /home/unai/miDisco
* Generalmente, tras montar un sistema de ficheros, el punto de montaje pertenece a “root”.
* Para hacer que pertenezca a nuestro usuario, se puede usar bindfs:
  1. Instalar bindfs:

sudo apt install bindfs

* 1. Montar con el parámetro user:

sudo mount -o user -t <tipo> <partición> <puntoMontaje>

* 1. Utilizar bindfs para remontar con permisos de usuario:

sudo bindfs -u $(id -u) -g $(id -g) <puntoMontaje> <puntoMontaje>

* Para desmontar, requiere utilizar umount 2 veces.

Desmontar una partición

* Por defecto, comando **umount**:
  + Sintaxis → umount [punto-montaje]
  + Requiere que ningún proceso esté usando la partición.

Obtener información:

* Comando **lsblk**:
  + Muestra discos y particiones.
  + Parámetro “-e7” para ocultar particiones Snap en Ubuntu.
* Comando **df**:
  + Muestra particiones y puntos de montajes.
  + Parámetro ”-h” para mostrar tamaños en formato “humano”.
  + Parámetro “-t” para mostrar los tipos de sistemas de ficheros.
* Comando **mount**:
  + Muestra las particiones montadas.
  + Parámetro “-l” para listar.
  + Parámetro “-t” para indicar tipo de sistema, p.e. “-t ext4”.

Discos en GCP:

* Añadir un disco a una instancia:
  + Editar la configuración de la MV.
  + En la sección “Discos adicionales”.
    - Seleccionar “Agregar nuevo disco”.
  + Configurar el nuevo disco, entre otras:
    - Nombre y tamaño en GB.
    - Origen: “blank disk” para un disco en blanco.
    - Tipo: ver siguiente diapositiva.
  + Crear disco y guardar cambios en la instancia.
* Administrar discos
  + Apartado “Discos” en la sección “Almacenamiento” de Compute Engine.
* Tipos de discos:
  + El coste mensual depende de la región.
  + Precios para la región europe-north2 a fecha 10/09/2025:

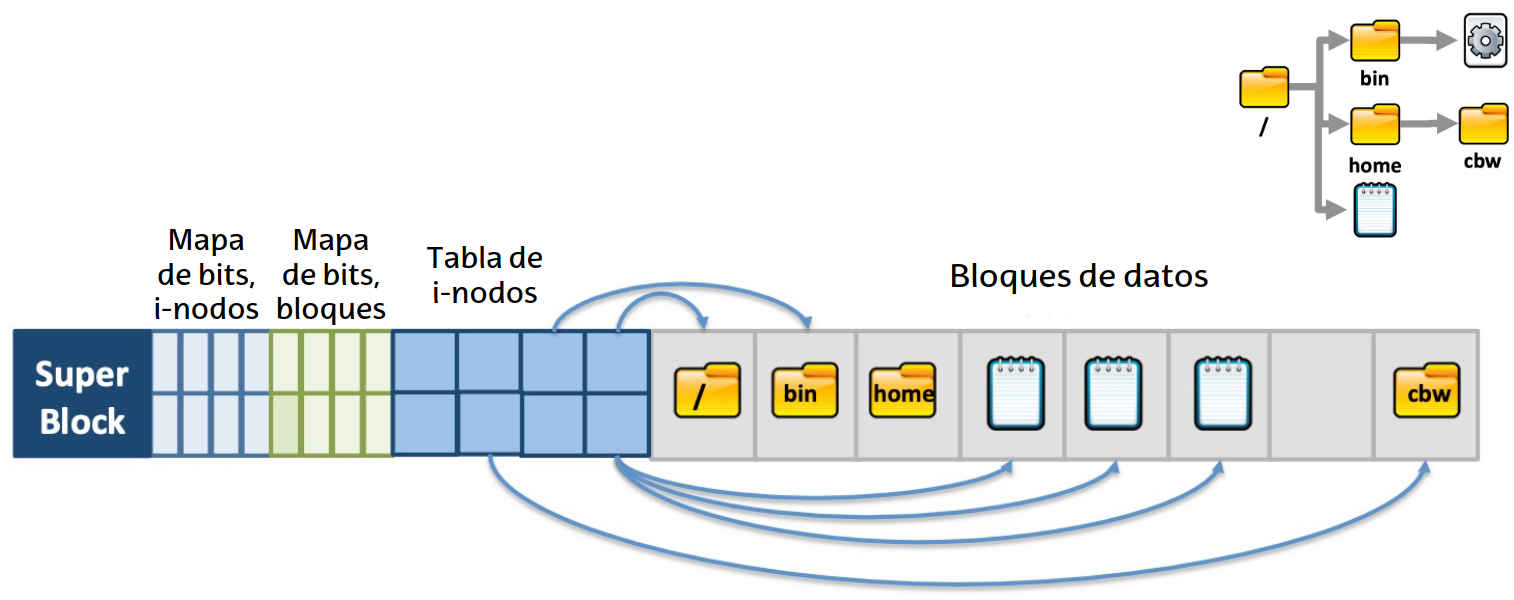
| **Tipo** | **Significado** | **Coste ($/mes)** |
| --- | --- | --- |
| pd-standard | Discos HDD | 0,04 / GB |
| pd-balanced | Discos SSD configurados para ser competitivos en coste | 0,1 / GB |
| pd-ssd | Discos SSD | 0,17 / GB |
| pd-extreme | Discos SSD configurados para máximo rendimiento | 0,12 / GB |

**1. ariketa:**

El sistema de ficheros:

* Parte del Sistema Operativo que administra la memoria de los dispositivos y unidades.
* Características principales:
  + Ficheros identificados por un nombre.
  + Meta-información para cada fichero: Fecha de creación, permisos …
  + Organización como una jerarquía tipo árbol.
* Implementación:
  + Internamente los ficheros se almacenan en bloques secuenciales.
  + La jerarquía no se tiene en cuenta a nivel interno.

Sistema EXT:

* Su nombre viene de **Ext**ended File System:
  + Creado en 1992, el primer sistema de ficheros para Linux.
  + Versión actual: Ext4 (en uso estable desde 2008).
  + Compatibilidad hacia delante y hacia atrás.
    - Un sistema ext3 puede ser montado como ext4 sin cambios.
* Organiza los datos en base a i-nodos:

BtrFS:

* Creado en 2009, parte de Linux desde 2013
* Características relevantes integradas: Gestor de volúmenes integrado, gestor de snapshots, …
* Recomendado para grandes tamaños de datos:
  + El mismo BtrFS puede expandirse a varios discos.
* Posible sucesor de Ext4, ya que no parece que vaya a haber Ext5.
* Algunas características integradas:
  + Compresión de ficheros: Comprime automáticamente cada fichero incluido
  + Sistema integrado de cuotas: Permite establecer límites de uso a los usuarios.
  + Balance automático: Reparte datos entre diferentes discos para evitar que se llenen.

ZFS:

* Creado en 2001 por Sun Microsystems, ahora Oracle.
* Es un sistema de ficheros y gestor de volúmenes.
* Muy estable.
* Licencia privativa: Linus Torvalds ha creado polémica al respecto.
* Variante libre: OpenZFS
  + Creado en 2013.
  + Utilizado en entornos Linux.

Sistema FAT:

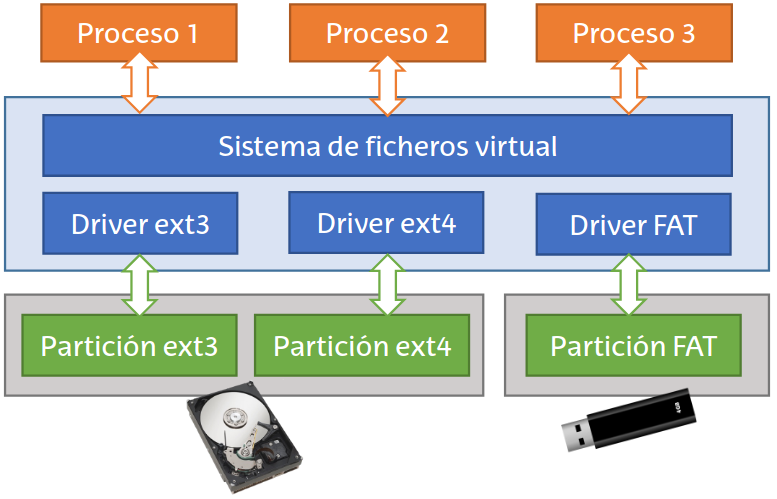
* Creado en 1977 y utilizado por MS-DOS.
* File Allocation Table - Tabla de reserva de ficheros.
  + Actualmente en desuso en sistemas de escritorio.
* Variantes en uso a día de hoy:
  + exFAT: Orientado a tarjetas flash (p.e. SDXC) y memorias USB.
  + FATX: Utilizado en las videoconsolas Xbox.
  + VFAT: Utilizado en el arranque de algunos SSOO.

Sistema NTFS

* Creado en 1993 y utilizado en Microsoft Windows
* New Technology File System.
  + Introduce journaling: sistema de diario.
* Versión actual: v3.1
  + No es tan compatible como FAT
* En proceso de ser sustituido por ReFS.
  + Disponible en Windows Server desde 2012.
  + Windows 11 puede leer discos ReFS, pero no crearlos.

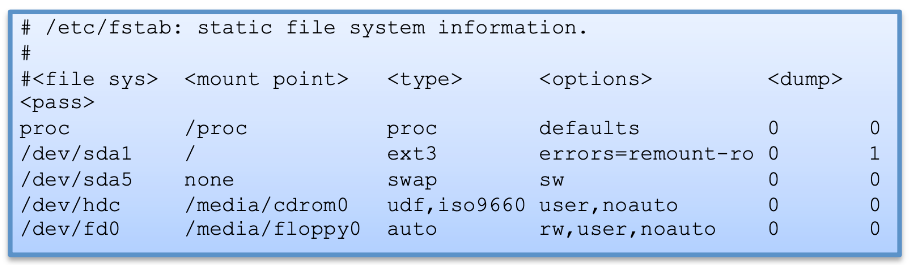
Sistema de ficheros virtual

* Es una interfaz de Linux que:
  + Expone una API POSIX a los procesos.
  + Envía las peticiones concretas al Driver que corresponda.
* Aunque el SSOO monte particiones de diferentes tipos, su uso es transparente a los procesos



Montaje automático:

* El fichero /etc/fstab define los dispositivos a montar automáticamente al arrancar el sistema.
* Columnas:
  + <file sys> → Dispositivo
  + <mount point> → Punto de montaje (directorio)
  + <type> → Tipo de partición: ext3, ext4, swap, …
  + <dump> → Frecuencia de backup, no se usa en la actualidad
  + <pass> → Flag: Ejecutar fsck al siguiente arranque
* Ejemplo:



Monitorizar el uso de ficheros:

* Comando **fuser**:
  + Muestra qué ficheros están abiertos y por qué proceso.
  + Sintaxis → fuser -mv <directorio>
  + La columna Acceso muestra el modo:
    - f: abierto para lectura, F: abierto para escritura, c: directorio de trabajo, …
* Comando **lsof**:
  + Listado de todos los ficheros abiertos.
* Ante el error “Target is busy” al desmontar una partición, utilizarlos para ver ficheros en uso.

Comprobar el sistema de ficheros

* Comando **fsck**:
  + Detección y corrección (a veces) de problemas de corrupción del sistema de ficheros.
  + Compara la lista de bloques libres con las direcciones en los i-nodos.
  + Verifica la lista de i-nodos libres con los i-nodos de los directorios.
* Comando **badblocks**:
  + Detecta y excluye sectores inválidos del disco.
* Funciones SMART de un disco duro:
  + Self Monitoring Analysis and Reporting Technology.
  + Herramientas para acceder a la información de estado del disco.
  + Software y funcionalidades dependen del fabricante.

Redimensionar el sistema de ficheros:

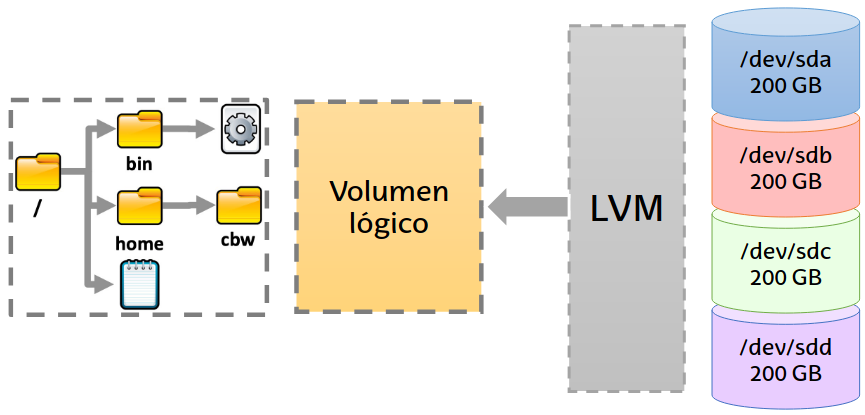
* Comando **resize2fs**:
  + Requiere una versión del kernel **≥** 2.6
  + Tiene que haber espacio suficiente para poder redimensionar.



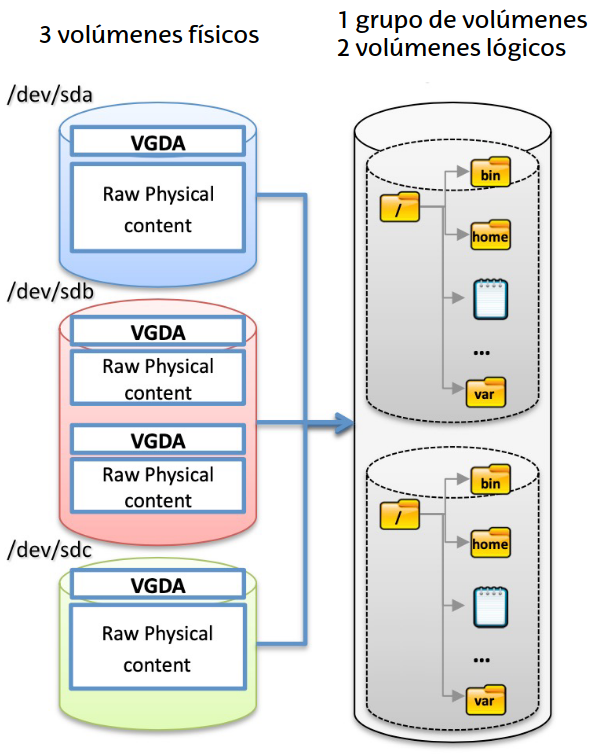
* + Es conveniente hacer una copia de seguridad de la tabla de particiones:
    - Utilizando **dd** → dd if=/dev/sdc of=part.bkp count=1 bs=1.
* Comando **parted**:
  + Sintaxis → parted /dev/sdX
  + Permite copiar, mover, cambiar sistemas de ficheros.

**2. ariketa:**

**1.2.3. LVM:**

Logical Volume Manager (LVM):

* Crea una capa de abstracción sobre el almacenamiento físico.
* Permite crear volúmenes lógicos que “escondan” el hardware real.

Jerarquía de LVM:

* Volúmenes físicos:
  + Partición completa.
  + Contiene el VGDA.
    - Volume Group Descriptor Area.
  + Contiene los datos físicos.
* Grupos de volúmenes:
  + Equivalente a “super-discos”.
* Volúmenes lógicos:
  + Equivalente a “super-particiones”.
  + Albergan los sistemas de ficheros.

Administración de LVM:

* Comando **pvcreate**:
  + Crea un volumen físico.
  + Sintaxis → pvcreate [partición]
  + Antes, hay que crear la partición (cfdisk).
* Comando **vgcreate**:
  + Crea un grupo de volúmenes con varios volúmenes físicos.
  + Sintaxis → vgcreate [nombre-grupo] [vols-físicos]
    - Ejemplo: vgcreate grupovol /dev/sdb1 /dev/sdc1
* Comando **lvcreate**:
  + Crea un volumen lógico.
  + Sintaxis → lvcreate [nombre-grupo] -l [tamaño] -n [nombre-volum-log]
    - Ejemplo: lvcreate grupovol -l 100%FREE -n miVolumen
* Comando **vgextend**:
  + Añade un nuevo volumen físico al grupo de volúmenes.
* Comando **lvextend**:
  + Extiende un volumen lógico a un grupo de volúmenes más grande.
* Comando **resize2fs**:
  + Redimensiona el sistema de ficheros.
* Comandos **vgreduce** (grupo de volúmenes) y **lvreduce** (volumen lógico):
  + Reducen el tamaño de los volúmenes.
* Comando **lvdisplay**:
  + Muestra el estado del volumen.

Ventajas de LVM:

* Gestión flexible del almacenamiento en disco:
  + Elimina los límites del espacio físico.
* Almacenamiento redimensionable:
  + Los volúmenes se pueden agrandar/reducir de forma simple.
  + Algunas operaciones no requieren desmontar el sistema de ficheros.
* Traslado de datos en caliente:
  + Los datos se pueden mover entre discos aunque estén en uso.
  + Se puede reemplazar un disco sin interrumpir el servicio.
* Captura de instantáneas:
  + Simplifica las copias de seguridad.

Desventajas de LVM:

* La recuperación de datos ante fallos no es trivial:
  + Recomendable combinarlo con otras técnicas.
* No es soportado por todos los SSOO.
* Requiere combinar con un sistema de ficheros compatible:
  + Habitualmente con Ext4.

Sistemas de ficheros con gestor de volúmenes integrados:

* BtrFS, ZFS, …
* Eliminan la necesidad de utilizar LVM de forma adicional.

**3. ariketa:**

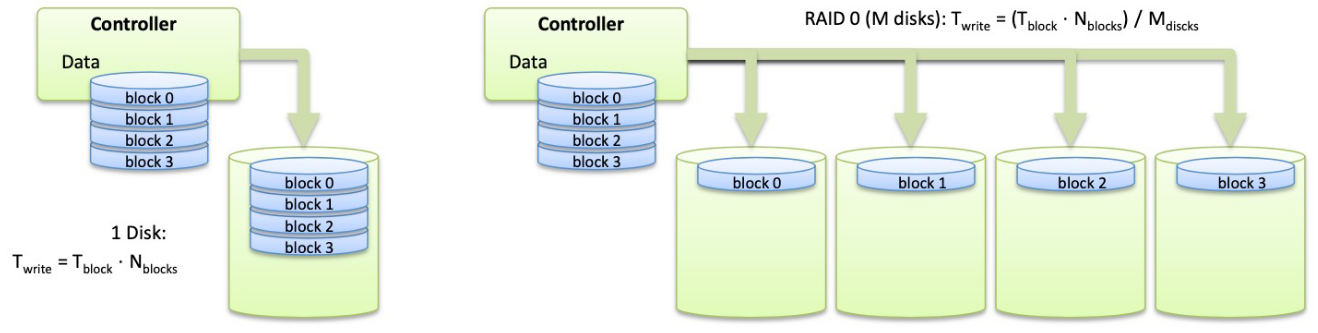
**1.2.4. RAID:**

Redundant Array of Independent Disks (RAID):

* Técnica para distribuir o replicar datos entre varios discos.
* Es transparente para el usuario y para el SSOO.
* Tiene diferentes opciones (niveles) de configuración.
  + Según necesidades de fiabilidad, rendimiento y capacidad.
* Se puede implementar a nivel HW o SW:
  + Hardware: Más eficiente pero más caro.
    - Imagen: Controladora PCI para RAID 0, 1, 5 y 10.
  + Software: Apropiado para RAID 0 y 1.

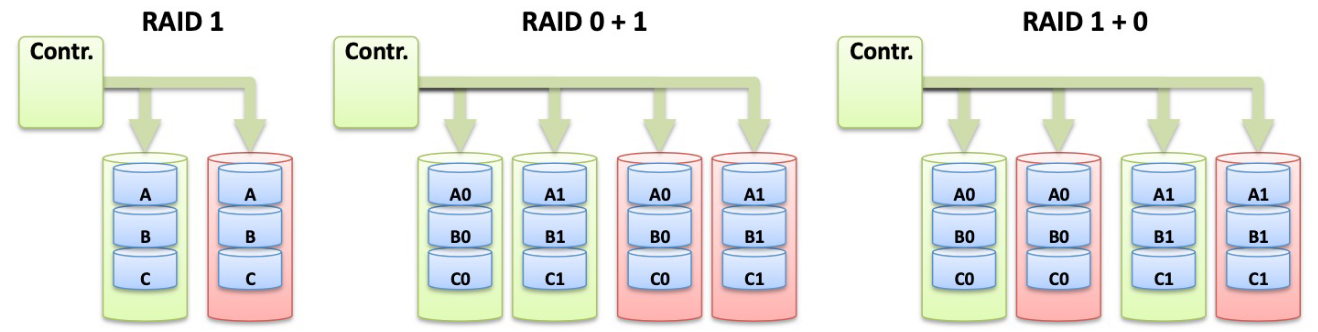
RAID 0 (Striping/Volumen dividido):

* Los datos se dividen en segmentos y se distribuyen entre los discos.
* **Rendimiento**: Bueno, acceso paralelo a los discos. Cuantos más discos, más velocidad.
* **Fiabilidad**: No hay tolerancia a fallos.
* **Capacidad**: 100% de uso (0 redundancia).



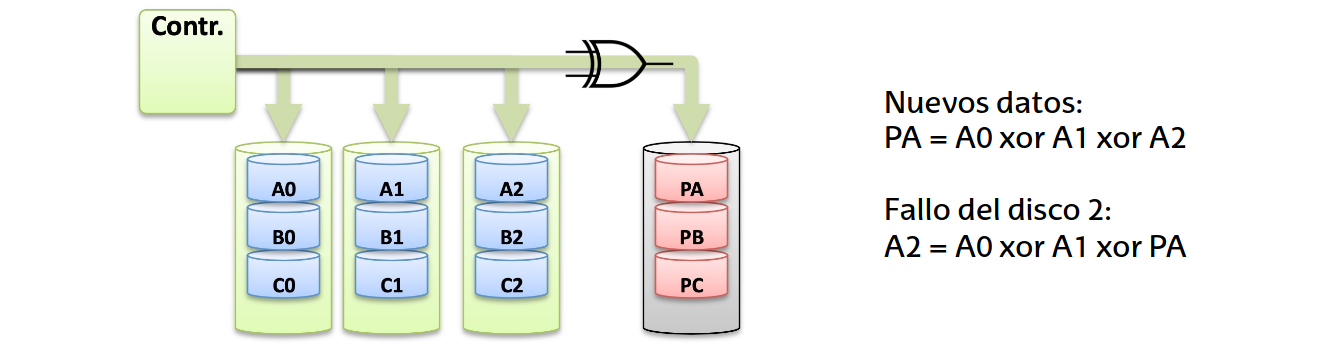
RAID 1 (Espejo):

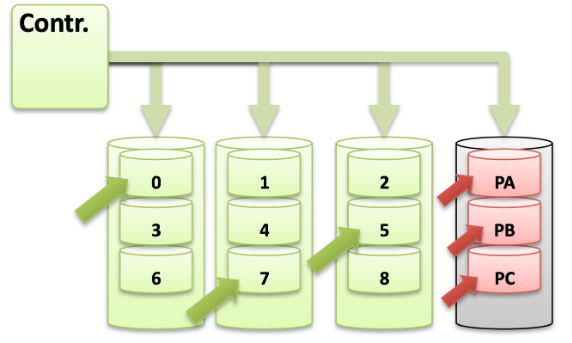
* Utilizar un disco secundario para copiar todos los datos.
* **Rendimiento**: Bajo, debido al exceso de escrituras.
* **Fiabilidad**: Alta, por la alta redundancia.
* **Capacidad**: 50% de la disponible.



RAID 4 (Striping + paridad):

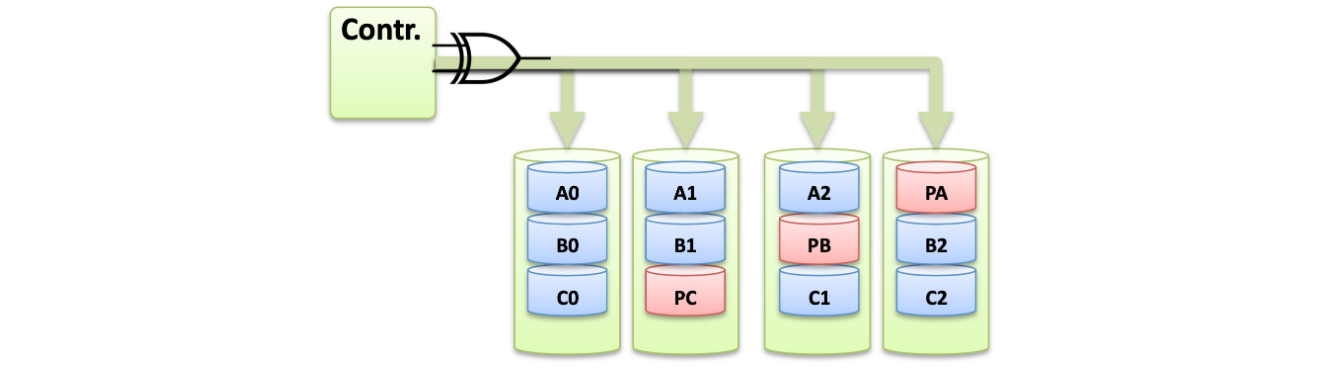
* Un disco almacena información de paridad sobre el resto.
* **Rendimiento**: Bueno en lectura, malo en escritura.
* **Fiabilidad**: Tolerancia al fallo de 1 disco.
* **Capacidad**: 1 disco dedicado exclusivamente a redundancia.



* Su mayor problema son las escrituras serializadas en el mismo disco.
  + Ejemplo: Actualizar las posiciones 0, 5 y 7:
    - Leer bloques 0, 5 y 7 y PA, PB y PC.
    - Calcular el nuevo valor de PA, PB y PC.
    - Escribir los nuevos bloques de datos.
    - Escribir los nuevos bloques de paridad.
      * Esto implica escrituras serializadas.
      * Bajo rendimiento.

RAID 5 (Striping + paridad distribuida):

* La información de paridad se distribuye por todos los discos.
* **Rendimiento**: Mejor que RAID 4, elimina la escritura serializada.
* **Fiabilidad**: Tolerancia al fallo de 1 disco.
* **Capacidad**: Se dedica el equivalente a 1 disco a redundancia.

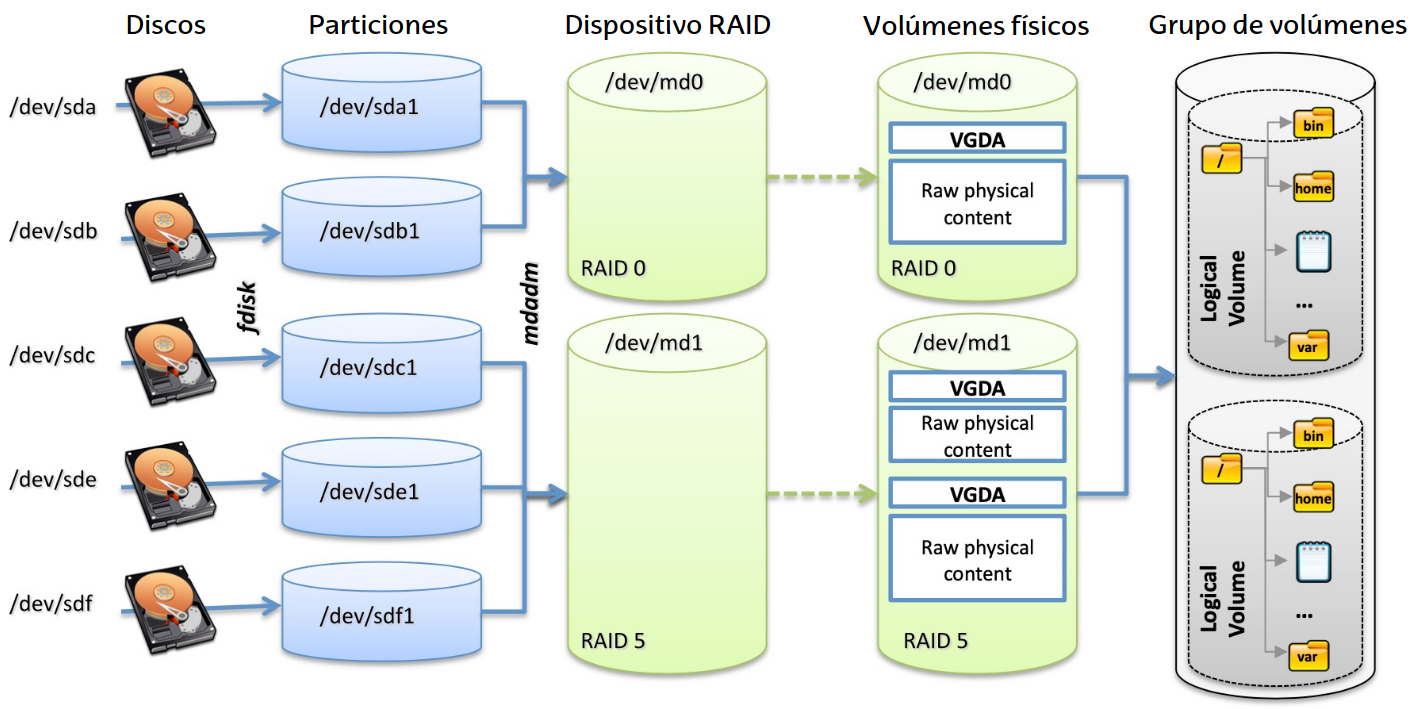


Otros niveles RAID:

* RAID 2, RAID 3:
  + Paridad a nivel de bit (RAID2) o byte (RAID3), en lugar de bloque.
  + No es muy utilizado.
* RAID 6 (Striping + Doble paridad):
  + RAID 4 pero usando el doble de espacio para paridad.
  + Tolerante al fallo de 2 discos.
* RAID anidados (Jerarquías en árbol):
  + P.e, RAID 0+1, RAID 1+ 0 (10), …

Combinando RAID y LVM:

* LVM se debe implementar sobre RAID.



Administración RAID (Comando **mdadm**):

* Crear un dispositivo RAID:
  + Crear disp. md0 en /dev/md0:

mdadm --create /dev/md0 --verbose --level=0 --raid-devices=2 /dev/sdb /dev/sdc2

* + Los discos tienen que haber sido previamente particionados (cfdisk).
  + El proceso de creación se puede monitorizar:
    - cat /proc/mdstat
* Monitorizar el sistema RAID:
  + mdadm --monitor [opciones] /dev/md0
* Eliminar (desactivar) RAID:
  + Parar el dispositivo → mdadm --stop /dev/md0
  + Limpiar información → mdadm --zero-superblock /dev/sdX
    - Limpia la información existente de un dispositivo RAID parado.

En caso de fallo de 1 disco:

* Asumiendo un sistema RAID 5.
* El disco roto se puede recuperar automáticamente:
  + Eliminar el disco roto del RAID → mdadm /dev/md0 –r /dev/sdc1
  + Reemplazar el disco físico por otro (Debe ser idéntico).
  + Crear particiones como en el original → fdisk /dev/sdc
  + Añadir al dispositivo RAID → mdadm /dev/md0 –a /dev/sdc1
  + Monitorizar el proceso de reconstrucción → cat /proc/mdstat
* Se puede simular el fallo de un disco:
  + mdadm/dev/md0 -f/dev/sdc1
  + Toda la información en: /var/log/syslog

**4. ariketa:**

**1.2.5. Backups:**

Backups:

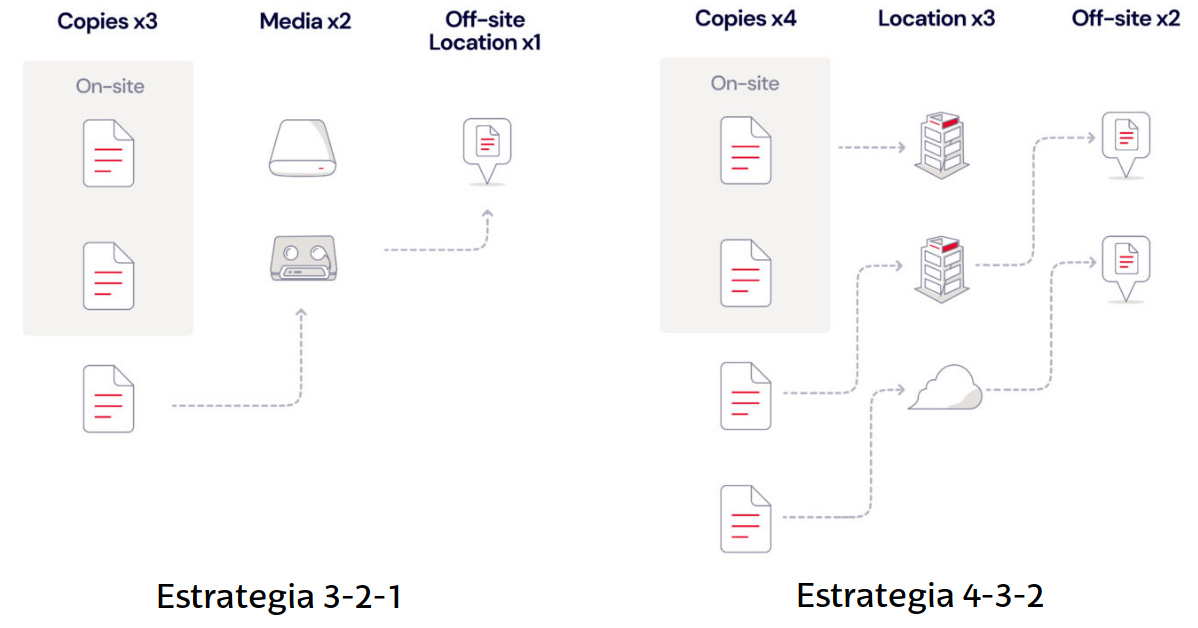
* RAID no es suficiente para tener una disponibilidad del 100%.
* Las copias de seguridad son esenciales:
  + Solución para eventos inesperados, tanto HW como SW.
  + Evita potenciales problemas de los usuarios.
* Requiere dedicar recursos exclusivos:
  + Recursos físicos:
    - Discos dedicados exclusivamente a copias, Servidores SAN, …
    - Cintas: LTO (LinearTape-Open), SAIT, AIT.
  + Almacenamiento en la nube.

Política de copias:

* Debe ir acorde a nuestros requisitos.
* ¿**Qué** es necesario guardar?
  + Datos de usuarios/aplicaciones/sistema.
  + Las partes críticas del sistema.
* ¿**Cuándo** queremos hacer las copias?
  + No recargar el sistema en momentos críticos.
  + Dependerá del nivel de uso y la parte del sistema de ficheros.
  + Automatizar las copias (usando p.e. cron).
* ¿**Dónde** queremos hacer las copias?
  + Balance entre copias locales y en ubicaciones remotas.

Estrategias para el almacenamiento de Backups:

* Ejemplos:



Comando **rsync**:

* Herramienta GNU para backups.
* Web oficial: <https://rsync.samba.org/>
* Forma de uso más simple:
  + rsync [opciones] <origen> <destino>
  + Opciones:
    - -v → Modo verboso.
    - -a → Mantiene usuarios.
    - -z → Comprime antes de copiar.
    - -h → Tasas de transferencia y tamaños en formato MB/s en vez de bytes/s.
  + Ejemplo: rsync -vazh /home /dev/sdc
* Se suele utilizar para copias remotas por red.

Comando **rsnapshot**:

* Herramienta basada en rsync para realizar copias incrementales, gestionando un histórico de las mismas con rotación.
  + Web: <https://rsnapshot.org/>
  + Documentación: <https://wiki.archlinux.org/title/Rsnapshot>
* Instalación → apt install rsnapshot
* Configuración → /etc/rsnapshot.conf
* Uso:
  + rsnapshot configtest → Verifica que un fichero de configuración es correcto.
  + rsnapshot <TAG> → Realiza una copia del tipo <TAG>, p.e. “daily”.
  + rsnapshot-diff → Compara 2 copias hechas en instantes diferentes.

Alternativas más rudimentarias:

* Comando **tar**:
  + Combinándolo con herramientas de compresión (bzip, zip).
* Comando **dd**:
  + dd if=/dev/sda2 of=/dev/tape
* Comando **cp -a**:
  + Para replicar contenido de disco a nivel de fichero.

Alternativas comerciales:

* Dell Backup and Recovery
* IBM Storage Protect
* etc.

Los siguientes datos son de BackBlaze:

* Consultora dedicada al almacenamiento en la nube.
* Cada trimestre publica un informe detallando:
  + Ratio de fallos en sus discos duros.
  + Comparativas de rendimiento.
  + Datos históricos.

