

PROMETEO

Unidad 5: Redes y conectividad en sistemas operativos

Sistemas informáticos

Técnico Superior de DAM / DAW



Sesión 16: Seguridad en redes (WPA2, WPA3, VPN, firewalls básicos).

La seguridad en redes es la **primera línea de defensa en cualquier entorno digital**. Desde una red doméstica hasta una infraestructura empresarial, la información viaja constantemente a través de canales inalámbricos o cableados que pueden ser interceptados, manipulados o atacados. Por eso, entender y aplicar correctamente los estándares y herramientas de seguridad es esencial para proteger la confidencialidad, integridad y disponibilidad de los datos.

Wi-Fi segura: del WPA2 al WPA3

El estándar **WPA2 (Wi-Fi Protected Access 2)** ha sido durante más de una década el pilar de la seguridad Wi-Fi. Utiliza el protocolo de cifrado AES (Advanced Encryption Standard) para proteger la comunicación entre el punto de acceso y los dispositivos conectados. Sin embargo, con el tiempo se han descubierto vulnerabilidades (como el ataque KRACK en 2017) que demostraron que WPA2, aunque robusto, no es infalible.

Su sucesor, **WPA3**, representa un salto cualitativo en protección. Introduce un nuevo protocolo de autenticación llamado SAE (Simultaneous Authentication of Equals), que reemplaza el método de intercambio de claves precompartidas (PSK) de WPA2. SAE utiliza un sistema de intercambio basado en contraseñas resistentes a ataques de fuerza bruta y evita que un atacante pueda probar combinaciones de contraseñas sin estar físicamente cerca de la red.

Además, WPA3 incorpora:

- **Cifrado individual por usuario**, lo que significa que aunque dos dispositivos estén conectados a la misma red, su tráfico no puede ser interceptado entre ellos.
- **Mayor longitud de claves de cifrado (192 bits)** en entornos empresariales.
- **Protección contra ataques de diccionario offline**, un tipo de ataque en el que un atacante intenta adivinar contraseñas sin necesidad de estar conectado.

El resultado es una red inalámbrica mucho más segura y resistente frente a ataques modernos.

VPN: privacidad y anonimato en la red

Una **VPN (Virtual Private Network)** crea un **túnel cifrado** entre el dispositivo del usuario y un servidor remoto. Este túnel protege el tráfico de red de miradas indiscretas, especialmente en **redes Wi-Fi públicas**, donde es común que atacantes intenten interceptar la información que viaja sin cifrar.

La VPN oculta la dirección IP real del usuario y cifra todos los datos, de modo que incluso si alguien logra capturar el tráfico, solo verá datos ilegibles. Existen diferentes protocolos de VPN, como **OpenVPN**, **WireGuard** o **IPSec**, cada uno con sus ventajas en velocidad y seguridad.

El uso de VPN no solo es relevante para usuarios domésticos, sino también en entornos corporativos: permite a los empleados acceder de forma segura a la red interna de la empresa desde cualquier lugar, manteniendo la confidencialidad de los datos.

Firewalls: guardianes de la red

El **firewall (cortafuegos)** es un componente esencial de cualquier sistema de defensa. Actúa como un filtro entre la red interna (confiable) y el exterior (internet), analizando cada conexión entrante o saliente y bloqueando aquellas que no cumplan las reglas establecidas.

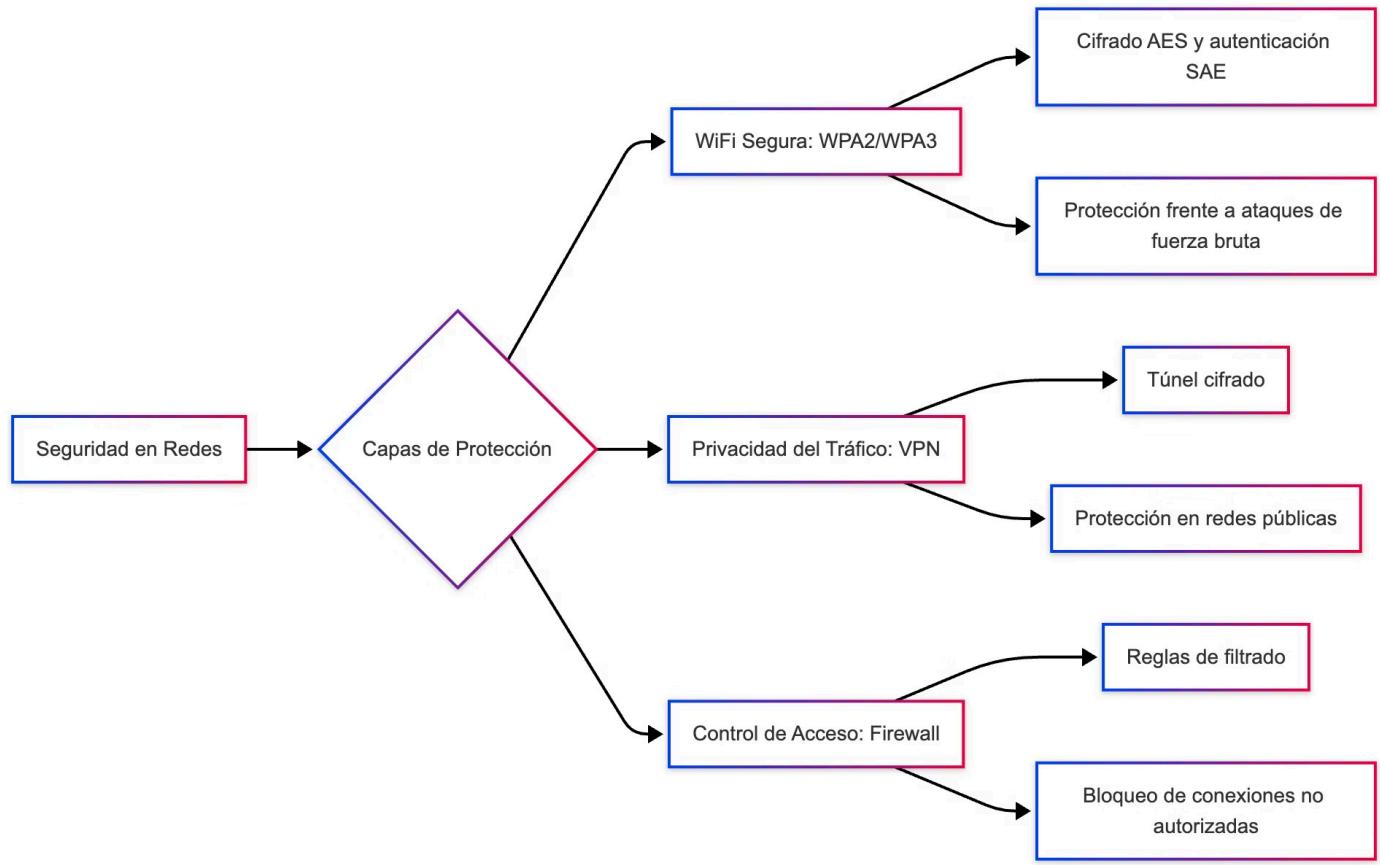
Los firewalls pueden ser:

- **De hardware**, integrados en routers o dispositivos dedicados.
- **De software**, instalados en sistemas operativos (Windows Defender Firewall, UFW en Linux, etc.).

Su función no es solo bloquear, sino también **registrar y alertar** sobre intentos sospechosos de acceso. En una red doméstica, un firewall evita que un programa malicioso se comunique con el exterior sin permiso; en una red empresarial, protege los servidores de intentos de intrusión o escaneo de puertos.

En conjunto, WPA3, las VPN y los firewalls forman una **arquitectura de defensa en capas**, donde cada componente cubre una parte del riesgo: cifrado del canal, privacidad del tráfico y control de accesos.

Esquema Visual: Capas de protección en redes seguras



ⓘ Interpretación del esquema:

- **Capa 1 (WPA3):** protege la red Wi-Fi mediante cifrado y autenticación segura.
- **Capa 2 (VPN):** garantiza la privacidad del tráfico y evita la interceptación en redes inseguras.
- **Capa 3 (Firewall):** controla y bloquea accesos no deseados, actuando como frontera de seguridad.

Cada capa refuerza la anterior, creando una estructura de defensa integral frente a amenazas internas y externas.



Caso de Estudio

pfSense – Seguridad profesional desde el software libre

Contexto

Una pequeña empresa tecnológica con 20 empleados necesitaba proteger su red interna sin invertir en costosos equipos comerciales. Buscaban una solución capaz de ofrecer firewall avanzado, acceso remoto seguro y segmentación de red para invitados.

Estrategia

Optaron por implementar **pfSense**, un sistema operativo de código abierto basado en FreeBSD especializado en funciones de router y firewall. Instalado en un equipo antiguo con dos tarjetas de red, pfSense se convirtió en el centro neurálgico de la seguridad de la empresa.

Las configuraciones clave incluyeron:

- **Firewall avanzado:** se definieron reglas detalladas para permitir únicamente el tráfico necesario (HTTP, HTTPS, VPN y correo).
- **VPN OpenVPN integrada:** los empleados podían conectarse a la red corporativa desde casa de manera cifrada y segura.
- **Red de invitados aislada:** creada mediante VLANs, evitando que los dispositivos de visitantes accedieran a recursos internos.
- **Sistema de detección de intrusos (IDS/IPS) con Snort:** para identificar comportamientos anómalos.

Resultado

- La empresa consiguió **seguridad de nivel empresarial sin coste de licencias**.
- Reducción del 90 % en intentos de acceso no autorizado detectados.
- Conexiones VPN estables para el teletrabajo, cumpliendo políticas de seguridad corporativa.

pfSense demostró que la seguridad avanzada no depende del presupuesto, sino de una arquitectura bien diseñada y herramientas de código abierto gestionadas con conocimiento.

Herramientas y Consejos

El refuerzo de la seguridad en redes no requiere grandes inversiones, sino **buenas prácticas y elección de herramientas adecuadas.**

Herramientas recomendadas

pfSense



Firewall avanzado / Router

Configurar políticas de red y VPN seguras. Ideal para redes domésticas o pymes.

UFW (Uncomplicated Firewall)



Firewall de software en Linux

Simplifica la gestión de iptables con comandos simples. Ejemplo: sudo ufw enable.

OpenVPN / WireGuard



VPN de código abierto

Crear túneles cifrados para proteger conexiones en redes públicas.

NordVPN / ProtonVPN



VPN comerciales confiables

Conexión segura para usuarios domésticos, con políticas "no-logs".

Router WPA3



Hardware

Asegura el cifrado más moderno para redes Wi-Fi domésticas o corporativas.

Consejos prácticos

1 Activa WPA3 siempre que sea posible.

Asegúrate de que tu router y dispositivos lo soportan; la diferencia en seguridad frente a WPA2 es significativa.

2 Evita contraseñas débiles.

Una red Wi-Fi segura empieza por una clave robusta de al menos 12 caracteres combinando letras, números y símbolos.

3 Usa VPN en redes públicas.

En cafeterías, aeropuertos o universidades, una VPN evita que tu tráfico sea interceptado.

4 Configura el firewall para denegar por defecto.

Solo permite los puertos y servicios necesarios. En Linux, sudo ufw default deny incoming establece esta política.

5 Mantén tu router actualizado.

Muchos routers antiguos tienen vulnerabilidades conocidas que se solucionan con actualizaciones de firmware.

6 Segmenta tu red.

Crea una red separada para invitados o dispositivos IoT, reduciendo el riesgo de accesos cruzados.

7 Supervisa los registros del firewall.

Analiza intentos de conexión sospechosos para detectar posibles ataques tempranamente.

Mitos y Realidades

 **MITO:** "Usar una Wi-Fi pública es seguro si la web usa HTTPS."

→ **FALSO.** HTTPS solo protege la comunicación con un sitio web concreto. En una red pública, un atacante podría interceptar tus solicitudes DNS o redirigirte a sitios falsos. Una **VPN cifra todo el tráfico** (no solo el de una página web), protegiendo así cada paquete que sale de tu dispositivo.

 **REALIDAD:**

Las VPN cifran la conexión completa, evitando que cualquiera en la misma red pueda ver qué sitios visitas o capturar tus credenciales.

 **MITO:** "Los firewalls solo sirven para empresas."

→ **FALSO.** Todo dispositivo conectado a internet es un posible objetivo. Los sistemas operativos modernos incluyen firewalls integrados (como UFW en Linux o el firewall de Windows) precisamente porque **también son necesarios en entornos personales.**

 **REALIDAD:**

Un firewall doméstico puede bloquear intentos de conexión saliente de malware o controlar qué programas acceden a internet, aportando una capa esencial de defensa.

Resumen Final

- **WPA3:** estándar Wi-Fi más seguro, con autenticación SAE y cifrado individual.
- **VPN:** crea un túnel cifrado que protege tu tráfico y oculta tu IP.
- **Firewall:** controla el tráfico de red y bloquea conexiones no autorizadas.
- **UFW:** herramienta sencilla en Linux para gestionar reglas de firewall.
- **Defensa en capas:** Wi-Fi segura + VPN + Firewall = red protegida.



Sesión 17: Tipos de almacenamiento: HDD, SSD, NVMe, Cloud Storage

El almacenamiento es uno de los pilares de cualquier sistema informático. Sin un medio confiable donde guardar los datos, ningún ordenador o servidor podría funcionar correctamente. Sin embargo, no todos los tipos de almacenamiento son iguales: difieren en velocidad, durabilidad, consumo, precio y, sobre todo, en el tipo de uso para el que resultan más eficientes.

Durante décadas, los **HDD (Hard Disk Drive)** fueron la opción dominante. Son discos mecánicos que almacenan datos en platos magnéticos giratorios, leídos por un brazo mecánico. Su ventaja principal sigue siendo la **gran capacidad por bajo coste**, lo que los hace ideales para archivado o almacenamiento masivo de datos. Sin embargo, su naturaleza mecánica los hace **lentos, ruidosos y vulnerables a golpes**. Un HDD típico tiene velocidades de lectura/escritura entre **100 y 200 MB/s**.



Con la llegada de la **memoria flash**, aparecieron los **SSD (Solid-State Drive)**. Estos eliminan las partes móviles y almacenan datos en chips de memoria NAND. Esto se traduce en **mayor velocidad, menor consumo energético y una resistencia mecánica muy superior**. Un SSD moderno puede alcanzar fácilmente **500 MB/s en lectura y escritura secuencial**, y los tiempos de acceso son casi instantáneos. Además, los SSD son silenciosos y ligeros, lo que los hace el estándar actual para sistemas operativos y aplicaciones.

Pero el salto más significativo en los últimos años ha sido el de los **SSD NVMe (Non-Volatile Memory Express)**. Este tipo de almacenamiento utiliza el **bus PCI Express (PCIe)** en lugar del tradicional SATA, eliminando los cuellos de botella del pasado. Un SSD NVMe puede superar los **3.000 MB/s**, e incluso los **7.000 MB/s** en modelos de gama alta. Este tipo de unidades se usa ampliamente en entornos profesionales que requieren máxima velocidad, como edición de vídeo 4K, análisis de datos o ejecución de máquinas virtuales. La clave es que NVMe no es una forma diferente de memoria, sino **un protocolo de comunicación** más eficiente que permite a la CPU interactuar con la unidad sin las limitaciones del SATA.

- ⓘ Por último, el **almacenamiento en la nube (cloud storage)** ha transformado la forma en que trabajamos con los datos. En lugar de guardar los archivos físicamente en un dispositivo local, estos se almacenan en servidores distribuidos por todo el mundo, gestionados por empresas como **Google (Drive)**, **Microsoft (OneDrive)**, **Amazon (S3)** o **Dropbox**. El usuario puede acceder a sus archivos desde cualquier dispositivo conectado a Internet. El almacenamiento en la nube no reemplaza necesariamente al almacenamiento local, sino que lo **complementa**. Es ideal para sincronización de archivos, colaboración en tiempo real y respaldo remoto, aunque depende totalmente de la conexión a Internet y la confianza en el proveedor.

En el entorno profesional actual, el equilibrio perfecto suele lograrse combinando varios tipos de almacenamiento:

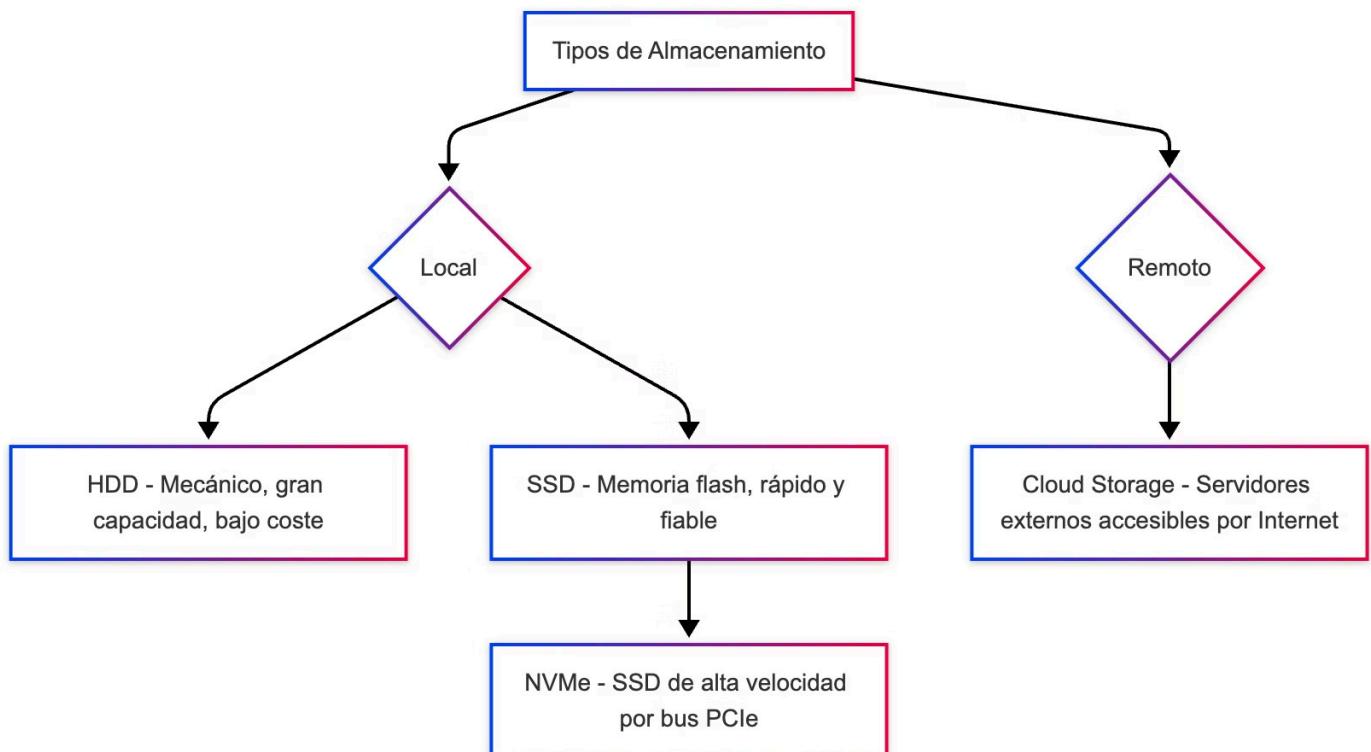
- **SSD o NVMe** para rendimiento del sistema operativo y aplicaciones.
- **HDD** para almacenamiento de datos masivos o copias de seguridad locales.
- **Cloud storage** para colaboración, movilidad y respaldo remoto.



La elección no depende solo de la tecnología, sino del **uso y la estrategia de gestión de la información** que adopte cada empresa o profesional.

Esquema Visual

El diagrama permite visualizar la evolución desde tecnologías locales físicas hacia soluciones distribuidas y remotas, mostrando que hoy en día la estrategia ideal suele combinar ambos enfoques.



ⓘ Descripción del esquema:

- **Nodo A:** representa el concepto general de "Tipos de Almacenamiento".
- **Nodo B (Local):** incluye las unidades físicas dentro del ordenador o conectadas por cable.
- **Nodo D (HDD):** enfatiza capacidad y coste.
- **Nodo E (SSD):** señala velocidad y fiabilidad como ventajas clave.
- **Nodo F (NVMe):** es una subcategoría de SSD que se comunica mediante el bus PCIe.
- **Nodo C (Remoto):** muestra el almacenamiento basado en la nube.
- **Nodo G (Cloud Storage):** simboliza el acceso ubicuo a datos desde cualquier dispositivo.

El diagrama permite visualizar la evolución desde tecnologías locales físicas hacia soluciones distribuidas y remotas, mostrando que hoy en día la estrategia ideal suele combinar ambos enfoques.



Caso de Estudio

Samsung Electronics

Contexto

Samsung se ha consolidado como **Líder mundial en almacenamiento de estado sólido**. A diferencia de otros fabricantes, controla todo el proceso: desde la producción de chips NAND flash hasta el desarrollo del firmware y software de gestión. En 2020 superó los **60 millones de SSD vendidos**, y su tecnología impulsa tanto ordenadores personales como centros de datos de alto rendimiento.

Estrategia

La compañía lanzó dos líneas emblemáticas:

- **Samsung 870 EVO (SATA SSD)**: orientada al usuario doméstico y profesional que busca fiabilidad y velocidad equilibrada.
- **Samsung 980 PRO (NVMe PCIe 4.0)**: enfocada a profesionales creativos y gamers exigentes, con velocidades que superan los **7.000 MB/s**.

Además, Samsung desarrolló el software **Samsung Magician**, que permite:

- Monitorizar el estado del disco mediante atributos **S.M.A.R.T.**
- Actualizar el firmware automáticamente.
- Ejecutar pruebas de rendimiento y optimizar el uso de energía.

En paralelo, la compañía apuesta por la **integración con servicios de cloud storage**, ofreciendo sincronización directa entre dispositivos móviles y ordenadores mediante **Samsung Cloud**, alineando así el ecosistema de almacenamiento local y remoto.

Resultado

Gracias a esta estrategia, Samsung se mantiene en la cima del mercado global con más del **35 % de cuota de SSD en 2024**, y sus unidades son referencia en durabilidad (más de **1.200 TBW**, o terabytes escritos). El caso de Samsung ilustra cómo la innovación continua en almacenamiento impacta directamente en la velocidad y la eficiencia de todos los dispositivos digitales modernos.

Herramientas y Consejos



Combinación inteligente de medios

Utiliza un **SSD o NVMe para el sistema operativo y programas**, y un **HDD de gran capacidad para almacenamiento de datos**. Así obtendrás rapidez en el uso diario sin renunciar a espacio asequible.

Ejemplo: SSD NVMe 500 GB + HDD 2 TB.



Herramientas de diagnóstico y mantenimiento

- **CrystalDiskInfo (Windows)**: muestra temperatura, tiempo de encendido y salud S.M.A.R.T.
- **GSmartControl (Linux/macOS)**: alternativa multiplataforma para verificar el estado de discos.
- **Samsung Magician / Crucial Storage Executive**: software específico de fabricantes para optimización.



Evita desfragmentar SSD

A diferencia de los HDD, los SSD no mejoran su rendimiento con la desfragmentación. Este proceso **reduce su vida útil**, ya que implica escrituras innecesarias. En su lugar, utiliza herramientas de optimización que ejecutan el comando **TRIM**, encargado de mantener el rendimiento del SSD.



Backups en la nube y sincronización

Plataformas como **Google Drive, Dropbox** o **OneDrive** facilitan el acceso y respaldo remoto. Sin embargo, para datos sensibles considera opciones cifradas como **Tresorit** o **pCloud Crypto**, que permiten **encriptación de extremo a extremo**.



Gestión profesional de almacenamiento

- Para empresas: soluciones como **Synology NAS + Synology Drive** permiten combinar almacenamiento local (RAID) con sincronización en la nube híbrida.
- Para desarrolladores o técnicos: herramientas como **rclone** permiten sincronizar y automatizar copias entre distintos servicios de cloud.

Mitos y Realidades

 **Mito:** "Los SSD duran muy poco y se desgastan rápido."

→ **FALSO.** Aunque las celdas NAND tienen un número limitado de ciclos de escritura, la tecnología moderna de **nivelación de desgaste (wear leveling)** y el uso de celdas **3D TLC o QLC** permiten que un SSD tenga una vida útil de **más de 10 años de uso normal**. Además, los fabricantes garantizan durabilidades específicas (medidas en TBW) que exceden con creces las necesidades de un usuario promedio.

 **Mito:** "El almacenamiento en la nube es 100 % seguro y no requiere copias de respaldo."

→ **FALSO.** Aunque los servicios en la nube implementan redundancia y cifrado, los mayores riesgos provienen del **factor humano**: borrados accidentales, errores de sincronización o ataques de ransomware. La regla profesional de protección de datos sigue siendo la **3-2-1: 3 copias de los datos, 2 tipos de soporte diferentes** (por ejemplo, un SSD y un disco externo), y **1 copia fuera de la ubicación física** (como la nube).

Resumen Final

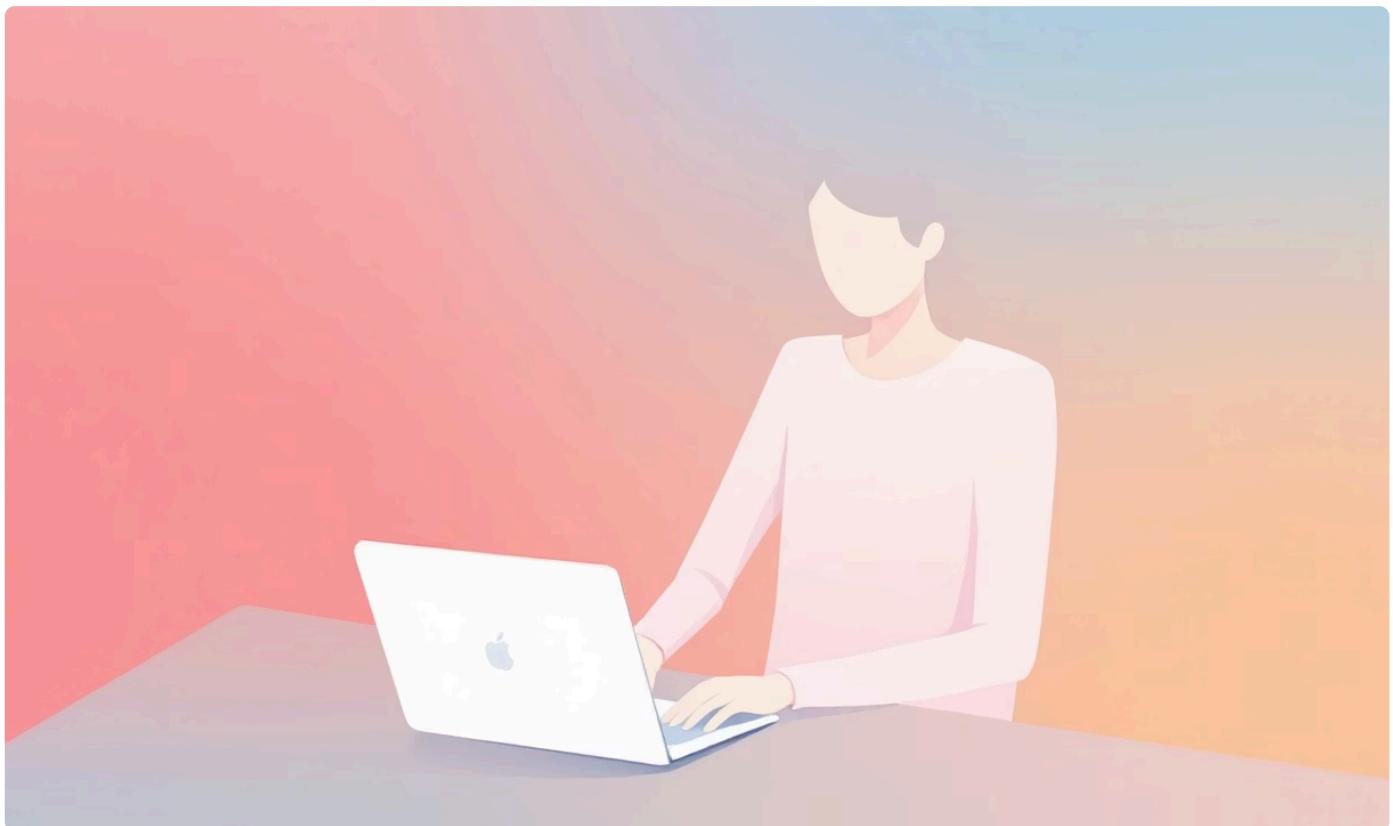
- **HDD:** mecánico, lento, pero con alta capacidad y bajo coste.
- **SSD:** rápido, silencioso y resistente, ideal para sistema operativo.
- **NVMe:** variante ultrarrápida de SSD que usa el bus PCIe.
- **Cloud Storage:** acceso remoto a datos desde cualquier lugar.
- **Regla 3-2-1:** nunca confíes en una sola copia; combina medios locales y remotos.

Sesión 18: Sistemas de archivos (NTFS, EXT4, APFS, ZFS, Btrfs).

Un sistema de archivos es el "alfabeto" con el que un sistema operativo organiza, nombra, guarda y recupera tus datos en discos y memorias. Sin él, el almacenamiento sería un bloque amorfio de bits imposible de utilizar. Aunque todos cumplen la misma misión, no todos lo hacen igual ni con las mismas garantías: cambian la forma en la que guardan metadatos, cómo protegen la integridad, qué límites de tamaño soportan o qué herramientas ofrecen para reparar errores. Entender estas diferencias te ayuda a elegir la opción adecuada para cada escenario profesional.

NTFS (Windows) es el estándar de Microsoft desde Windows NT. Aporta permisos detallados (ACLs), compresión, journaling para evitar corrupción tras caídas, cifrado (EFS), cuotas, enlaces duros/simbólicos y archivos de gran tamaño. Es robusto para uso de estación de trabajo y servidores Windows. Su desventaja habitual fuera de Windows es la compatibilidad: otros sistemas pueden leerlo/escribirlo con drivers de terceros, pero no siempre con todas sus funciones avanzadas.

EXT4 (Linux) es el caballo de batalla del ecosistema GNU/Linux. Hereda la solidez de EXT2/EXT3, añade journaling mejorado, extents (asignación contigua eficiente para archivos grandes), timestamps con mayor precisión, y soporta volúmenes y archivos muy grandes. Es estable, rápido y con herramientas maduras (e2fsck, tune2fs). ¿Su "pero"? No integra de serie funcionalidades modernas como snapshots o comprobación de integridad end-to-end.



APFS (Apple, macOS/iOS) está pensado para SSD/NVMe actuales. Emplea **copy-on-write (CoW)** para evitar sobreescrituras peligrosas, snapshots rápidos, cifrado nativo por volumen o por archivo, y "clones" de archivos instantáneos sin copiar datos físicamente (hasta que cambian). Su diseño reduce latencias en operaciones frecuentes de escritorio (duplicar carpetas, Time Machine, etc.). En contrapartida, su soporte fuera del ecosistema Apple es limitado.



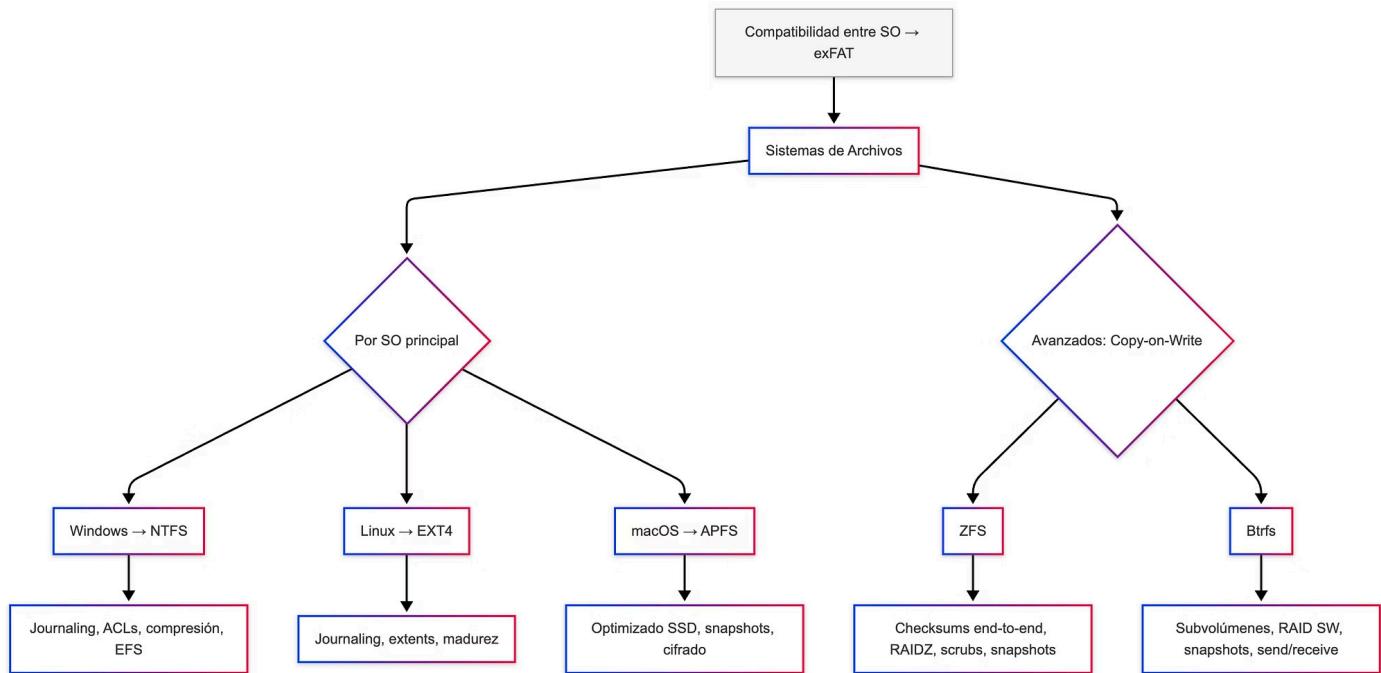
ZFS y Btrfs pertenecen a la "siguiente generación": no solo son sistemas de archivos, también integran conceptos de **gestión de volúmenes, snapshots, RAID por software y protección de integridad**.

Ambos usan CoW y **checksums** para detectar corrupción silenciosa ("bit rot").

- **ZFS** combina "pooles" de almacenamiento (zpool), **RAIDZ**, deduplicación opcional, compresión transparente y **scrubs** periódicos para detectar y reparar errores usando redundancia. Es referencia en integridad y fiabilidad para NAS y servidores.
- **Btrfs** integra subvolúmenes, RAID en capas de software, snapshots y envío/recepción de snapshots (send/receive) para replicación. Es muy flexible para estaciones Linux y servidores, con una administración moderna (balanceo, cuotas por subvolumen).

Como regla práctica: **EXT4** para entornos Linux generales por su madurez; **NTFS** para Windows; **APFS** para macOS y dispositivos Apple; y **ZFS/Btrfs** cuando necesitas **integridad, snapshots y gestión avanzada**. Para discos compartidos entre sistemas, **exFAT** ofrece compatibilidad sin la sobrecarga de NTFS ni las limitaciones de FAT32.

Esquema Visual



ⓘ Cómo leerlo:

- **NTFS/EXT4/APFS** son la elección por defecto cuando trabajas "dentro" de su SO nativo (Windows/Linux/macOS).
- **ZFS/Btrfs (CoW)** añaden **integridad** mediante checksums, **snapshots** casi instantáneos y **gestión de volúmenes/RAID** integrada: clave para servidores, NAS o estaciones críticas.
- **exFAT** actúa como pista rápida de **compatibilidad** para discos externos intercambiables.

Caso de Estudio

ZFS Project

Contexto

ZFS nace en Sun Microsystems con un objetivo ambicioso: **eliminar clases enteras de errores de almacenamiento** combinando en una sola capa lo que antes hacía un mosaico de soluciones (RAID, gestor de volúmenes, sistema de archivos, verificación de datos). Tras su apertura (OpenZFS), la comunidad ha continuado su evolución en múltiples plataformas (BSD, Linux) manteniendo los principios originales de integridad.

Estrategia/Arquitectura

- **Pooles (zpool)**: en lugar de "volúmenes fijos", ZFS crea un **pool** flexible de dispositivos. La capacidad se reparte dinámicamente entre **datasets** y **zvols** sin tener que "reparticionar".
- **Protección end-to-end**: cada bloque de datos y metadatos lleva un **checksum**. Al leer, ZFS **verifica** que lo recuperado coincide con lo escrito. Si hay redundancia (RAIDZ1/2/3, espejos), puede **autorreparar** el bloque defectuoso leyendo una copia buena.
- **Copy-on-Write**: nunca sobrescribe datos en sitio. Escribe en una ubicación nueva, y solo cuando la escritura completa es correcta, "apunta" las estructuras al nuevo bloque. Resultado: **consistencia** incluso tras un apagón.
- **Snapshots y clones**: snapshots instantáneos y eficientes (solo metadatos); clones de escritura diferida que comparten bloques mientras no cambian. Esto habilita **versionado**, entornos de **pruebas**, y **rollbacks** en segundos.
- **Mantenimiento (scrub)**: tarea periódica que lee todo el pool, **valida checksums** y repara corrupción latente antes de que afecte a los usuarios.
- **Compresión/deduplicación**: compresión transparente (lz4 habitual) para ganar rendimiento y espacio; deduplicación selectiva en casos de uso apropiados.

Resultado/Impacto

- En **NAS domésticos y laboratorios** (TrueNAS, por ejemplo), ZFS ha reducido drásticamente la incidencia de corrupción silenciosa y ha simplificado la administración (crecer el pool, snapshots, replicación).
- En **entornos empresariales**, su propuesta de valor es **integridad + disponibilidad**: si un disco empieza a devolver bloques corruptos, ZFS lo detecta en la lectura rutinaria o durante un **scrub**, corrige con la redundancia y marca el dispositivo como sospechoso, evitando "propagar" la corrupción a copias de seguridad.
- Operativamente, ZFS ha cambiado la cultura de almacenamiento: **infra basada en software**, recuperación ante errores silenciosos y **versionado** como práctica estándar.

Herramientas y Consejos



Elección rápida según escenario

- **Windows puro (estaciones/servidores):** NTFS por integración, ACLs y herramientas nativas (BitLocker para cifrado a nivel de volumen si lo necesitas).
- **Linux generalista (aplicaciones, contenedores):** EXT4 por estabilidad y rendimiento constante.
- **macOS/entorno Apple:** APFS, especialmente en SSD/NVMe; aprovecha snapshots y clones en flujos de trabajo creativos.
- **NAS/servidores con integridad y snapshots:** ZFS (TrueNAS, Debian/Ubuntu con OpenZFS) o Btrfs (openSUSE, Fedora) si priorizas flexibilidad y subvolúmenes.



Rendimiento práctico

- En SSD, evita llenar al 100%: deja **overprovisioning** (10–20%) para mantener IOPS estables.
- Activa **compresión** (lz4) en ZFS o Btrfs cuando tus datos sean textuales/logs/VMs: ahorrarás espacio y, a menudo, **ganarás rendimiento** (menos bytes a escribir/leer).

Buenas prácticas con ZFS/Btrfs

- **Snapshots automáticos:** en Btrfs, usa **Snapper** o **Timeshift** para snapshots programados y rollbacks de sistema; en ZFS, programa snapshots periódicos (por hora/día/semana) y **replicación** (send/receive) a otro equipo o disco.
- **Scrubs y balanceos:** en ZFS, agenda **scrubs** mensuales; en Btrfs, ejecuta **balance** cuando cambies el layout o tras borrar grandes volúmenes de datos.
- **Monitoreo:** integra alertas (smartmontools, zpool status, btrfs check) y revisa proactivamente sectores reasignados/errores.



Comandos útiles (Linux)

- **Identificar sistemas de archivos:** df -T (tipo por punto de montaje), lsblk -f (vista consolidada de dispositivos, UUID y FS).
- **Crear sistemas de archivos:** mkfs.ext4 /dev/sdXn, mkfs.btrfs /dev/sdXn.
- **Montaje:** mount /dev/sdXn /mnt; persistencia en /etc/fstab usando **UUID** para evitar sorpresas al reordenar dispositivos.

Compatibilidad entre sistemas

Para discos externos intercambiables entre Windows/macOS/Linux: **exFAT**. Evita FAT32 por sus límites (archivos >4 GB). Si necesitas NTFS en macOS, considera **drivers comerciales** (p. ej., Paragon NTFS) o acceso solo-lectura.

Mitos y Realidades

 **MITO:** "Todos los sistemas de archivos son básicamente iguales."

→ **FALSO.** Las diferencias son profundas: **integridad end-to-end** con checksums (ZFS/Btrfs), **snapshots** eficientes (APFS/ZFS/Btrfs), **ACLs** y cifrado nativo (NTFS/APFS), o **gestión de volúmenes/RAID por software** (ZFS/Btrfs). La elección impacta en **fiabilidad, recuperación ante fallos, rendimiento** y mantenimiento.

 **MITO:** "No puedo leer un disco de Linux en Windows."

→ **FALSO.** Windows no lee **EXT4** de forma nativa, pero puedes usar herramientas de terceros (p. ej., **ext2fsd** en versiones clásicas o drivers actualizados) o montar el sistema a través de **WSL/WSL2** en determinados escenarios. Para intercambiar datos sin complicaciones, usa **exFAT** en discos externos.

Resumen Final

- Un sistema de archivos define **cómo** se guardan y protegen los datos.
- **NTFS/EXT4/APFS** son opciones por defecto en sus SO; **APFS** destaca en SSD con snapshots/clones.
- **ZFS y Btrfs** (CoW) integran **integridad (checksums)**, **snapshots** y **gestión de volúmenes/RAID**: clave para NAS/servidores.
- **exFAT** es la vía rápida de **compatibilidad** entre Windows, macOS y Linux en discos externos.

Sesión 19: Particiones, montaje de volúmenes y LVM

En cualquier sistema operativo, el almacenamiento no se usa directamente "tal cual" viene de fábrica. Antes de guardar archivos, los discos deben **organizarse y dividirse** en zonas llamadas **particiones**, donde se crean y montan los sistemas de archivos. Comprender este proceso es esencial para cualquier profesional que gestione servidores o entornos Linux, ya que de él depende la seguridad y la flexibilidad del almacenamiento.

En Windows, las unidades suelen identificarse con letras (C; D; E:), pero en sistemas **Unix-like** como Linux o macOS, todo el sistema de archivos se estructura como un **árbol único**, donde los dispositivos se integran bajo un mismo punto raíz (/). El proceso de **montaje (mounting)** consiste en **asignar un sistema de archivos físico a un directorio lógico** dentro de ese árbol.

Por ejemplo:

- Montar un disco externo en /mnt/disco permite acceder a sus archivos desde esa carpeta.
- Si se desmonta (umount /mnt/disco), la carpeta sigue existiendo, pero el contenido del disco deja de ser visible.

Cada punto de montaje es una "ventana" hacia un sistema de archivos. Esto otorga enorme **flexibilidad**, ya que puedes montar diferentes discos o particiones en distintas rutas sin que el usuario perciba una ruptura entre ellos.

El papel del particionado

Un disco físico se puede **dividir en particiones** para separar datos del sistema, copias de seguridad, o zonas específicas (por ejemplo, /home, /var, /boot). En Linux, los dispositivos se identifican como /dev/sda, /dev/sdb, etc., y sus particiones como /dev/sda1, /dev/sda2, etc.

Cada partición se puede formatear con un sistema de archivos diferente (EXT4, XFS, Btrfs, etc.), lo que permite adaptar el rendimiento y la seguridad a las necesidades del sistema.

LVM: flexibilidad sobre el particionado tradicional

En sistemas modernos, el esquema de particiones fijo tiene limitaciones. ¿Qué ocurre si el espacio de /home se agota pero /var tiene gigas libres? Con el sistema clásico, habría que reparticionar el disco, lo que implica riesgo y tiempo de inactividad.

Ahí entra **LVM (Logical Volume Manager)**: una capa de abstracción entre los discos físicos y los sistemas de archivos que ofrece una gestión mucho más flexible.

LVM funciona agrupando los discos o particiones físicas en un **grupo de volúmenes (Volume Group, VG)**. De ese grupo se crean **volúmenes lógicos (Logical Volumes, LV)**, que se comportan como si fueran particiones normales, pero que **pueden redimensionarse** fácilmente, incluso en caliente, sin afectar los datos. Por ejemplo:

- Puedes tener tres discos de 1 TB cada uno, combinarlos en un solo grupo de volúmenes de 3 TB y crear dentro volúmenes lógicos para /home, /var y /srv.
- Si /home necesita más espacio, puedes ampliarlo con el comando lvextend sin interrumpir el servicio.

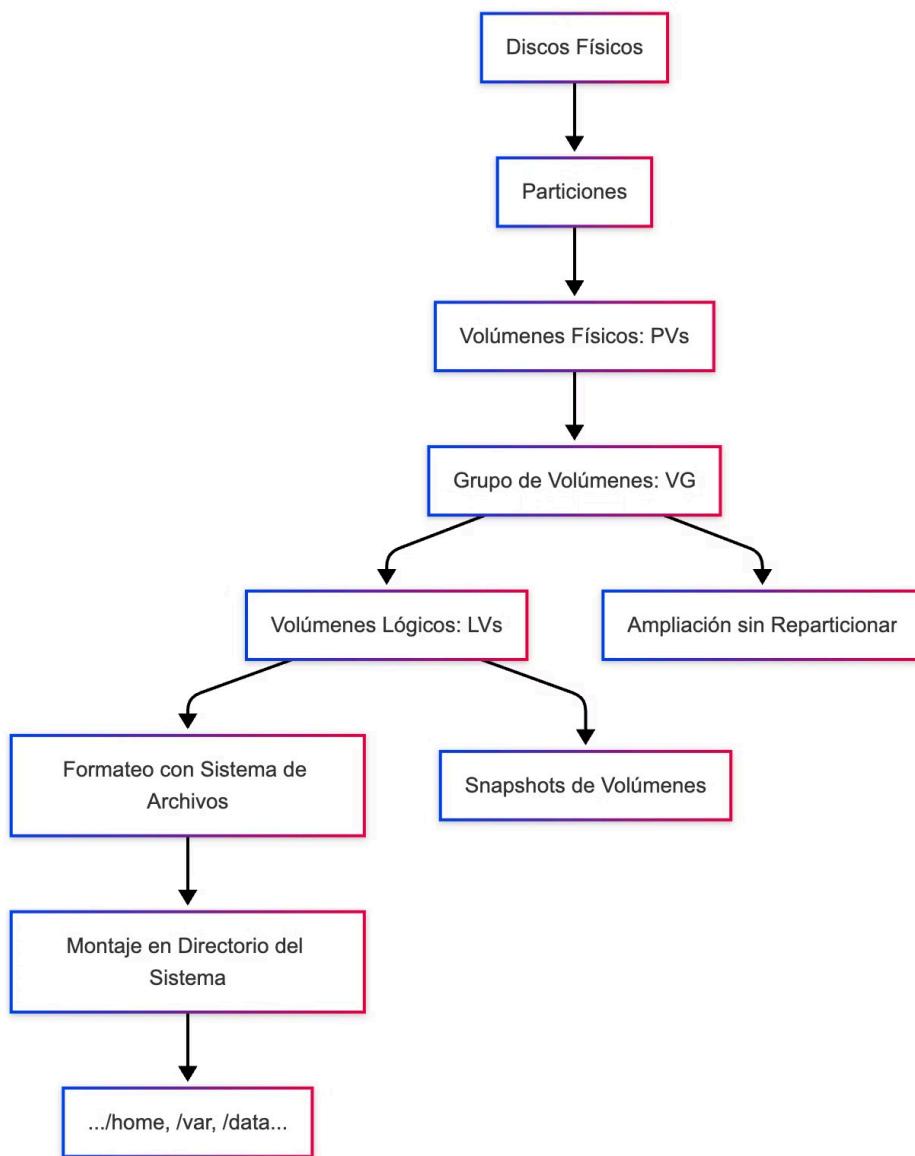
Este enfoque convierte la administración de discos en un proceso **dinámico y escalable**, ideal para servidores, entornos de virtualización o empresas en crecimiento.

Ventajas de LVM

- **Flexibilidad total:** redimensionar particiones sin necesidad de formatear.
- **Gestión avanzada:** crear instantáneas (snapshots) para realizar backups o pruebas sin interrumpir el servicio.
- **Agregación de almacenamiento:** varios discos se comportan como una sola unidad.
- **Extensibilidad:** añadir discos nuevos sin reinstalar el sistema.

En definitiva, LVM ofrece una gestión "modular" del almacenamiento, similar a cómo un arquitecto diseña un edificio con bloques que pueden ampliarse o modificarse sin derrumbar la estructura.

Esquema Visual



ⓘ Descripción del diagrama:

- Los **discos físicos** se dividen en particiones y se inicializan como **volúmenes físicos (PVs)**.
- Varios PV se agrupan en un **Volume Group (VG)**, que actúa como un "pool" de espacio total disponible.
- A partir del VG se crean **Logical Volumes (LVs)**, que se formatean y montan como particiones tradicionales.
- Estos LV pueden **ampliarse, reducirse o duplicarse** mediante snapshots, sin tener que tocar los discos físicos.

Este modelo aporta elasticidad al almacenamiento: puedes modificar el tamaño de los volúmenes "en vivo", algo impensable con el particionado convencional.



Caso de Estudio

Red Hat Enterprise Linux (RHEL)

Contexto

Desde hace más de una década, **Red Hat Enterprise Linux (RHEL)** —y su derivado CentOS— emplean **LVM como esquema predeterminado de particionado** en sus instalaciones de servidor. En entornos corporativos, los sistemas deben crecer, migrarse o reorganizarse sin interrumpir servicios críticos.

Estrategia

Red Hat adoptó LVM para permitir que los administradores:

- **Redimensionen volúmenes en caliente**, por ejemplo, cuando una base de datos o un directorio de logs empieza a llenarse.
- **Agreguen nuevos discos físicos** a un grupo existente sin reinstalar.
- **Crean snapshots temporales** para realizar backups consistentes sin detener aplicaciones.

Los comandos clave (`vgcreate`, `lvcreate`, `lvextend`, `lvremove`) permiten gestionar esta flexibilidad con unas pocas líneas de terminal.

Por ejemplo, si `/var` está al límite, basta con:

```
vgextend vgdata /dev/sdb
lvextend -l +100%FREE /dev/vgdata/var
resize2fs /dev/vgdata/var
```

En segundos, el sistema gana espacio disponible sin necesidad de reiniciar.

Resultado

Este modelo ha permitido a miles de administradores de sistemas en entornos RHEL y CentOS **adaptar el almacenamiento a las necesidades reales del negocio**, reduciendo tiempos de inactividad y simplificando tareas de mantenimiento.

Hoy, incluso distribuciones orientadas al usuario (Ubuntu, Fedora, openSUSE) ofrecen LVM como opción recomendada durante la instalación.



Herramientas y Consejos

Archivo /etc/fstab:

Define qué sistemas de archivos se montan automáticamente al iniciar el sistema.

Ejemplo de entrada típica:

```
/dev/mapper/vgdata-home /home ext4 defaults 0 2
```

Es recomendable usar **UUIDs** o nombres de LVM en lugar de rutas de dispositivo para evitar errores si cambian las letras del disco.

Comandos esenciales de LVM:

- **pvcreate**: inicializa una partición como volumen físico.
- **vgcreate**: crea un grupo de volúmenes a partir de PVs.
- **lvcreate**: genera un volumen lógico.
- **lvextend / lvreduce**: aumenta o reduce su tamaño.
- **lvdisplay, vgdisplay**: muestran información detallada.

Práctica recomendada:

- Mantén una pequeña partición independiente para /boot fuera de LVM. Algunos gestores de arranque no pueden acceder directamente a volúmenes lógicos.
- Documenta tus puntos de montaje y tamaños; los errores en /etc/fstab pueden impedir que el sistema arranque correctamente.

Herramientas visuales:

- **GParted**: particionador gráfico que muestra visualmente las unidades y puntos de montaje.
- **Cockpit (Red Hat/Fedora)**: interfaz web para gestionar LVM, discos y redes de forma intuitiva.
- **Webmin o YaST** (openSUSE) también permiten operaciones sobre volúmenes LVM sin usar terminal.

Montaje manual y automático:

Para montar temporalmente un sistema de archivos:

```
sudo mount /dev/vgdata/home /mnt/home
```

Para hacerlo permanente, edita `/etc/fstab` y añade la entrada correspondiente.

Snapshots en LVM:

Permiten crear una imagen del volumen lógico en un momento determinado. Muy útil antes de aplicar actualizaciones o migraciones.

Ejemplo:

```
lvcreate --size 2G --snapshot --name snap_home /dev/vgdata/home
```

Si algo sale mal, puedes revertir con `lvconvert --merge /dev/vgdata/snap_home`.

Comprende la estructura base:

Discos físicos → PV → VG → LV → sistema de archivos → montaje. Es el flujo de creación que siempre se debe respetar.

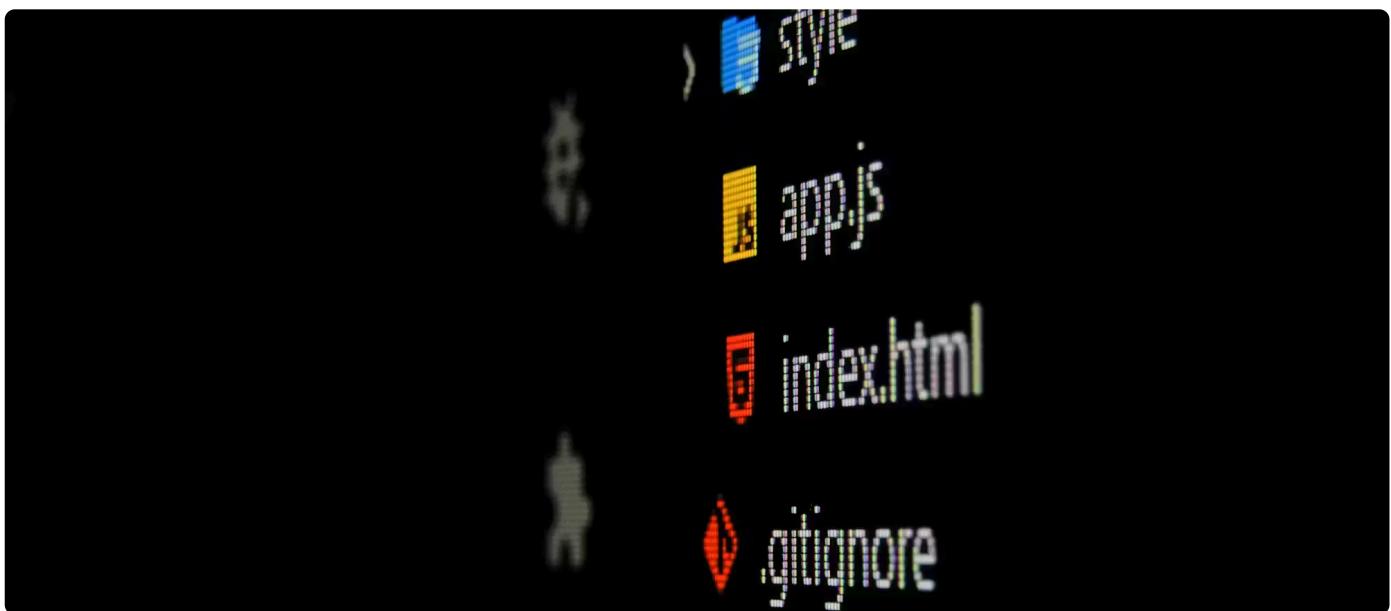
Mitos y Realidades

✗ **MITO:** "LVM ralentiza el rendimiento del disco."

→ **FALSO.** El impacto de LVM en el rendimiento es insignificante en la mayoría de los casos. La capa de abstracción añade una sobrecarga mínima, mientras que las ventajas en flexibilidad y escalabilidad compensan ampliamente. De hecho, en entornos SSD o RAID, el impacto puede ser inferior al 1%.

✗ **MITO:** "Si añado un disco nuevo a LVM, tengo que reinstalar todo."

→ **FALSO.** Esa es precisamente su fortaleza: puedes añadir un disco (vgextend) a un grupo de volúmenes existente y luego expandir un volumen lógico (lvextend) **en caliente**, sin pérdida de datos ni reinstalación del sistema operativo.



❑ Resumen Final

- **Montaje** = proceso de hacer un sistema de archivos accesible desde un directorio del sistema.
- En Linux, **no existen letras de unidad**, sino puntos de montaje dentro del árbol `/`.
- **LVM** crea una capa flexible entre discos físicos y sistemas de archivos.
- Permite **agregar, redimensionar o eliminar volúmenes** sin reinstalar.
- El archivo `/etc/fstab` gestiona los montajes automáticos.
- Ideal para **servidores y entornos dinámicos** que crecen o cambian con frecuencia.