r





**Tabla de contenido**

[1 Clase 12: Compresión y Compilación 3](#_heading=h.ce6gsrpwfocz)

[1.1 Algoritmos de compresión 3](#_heading=h.z2irj8qklkfg)

[1.2 Compilar 4](#_heading=h.az4cjfh13ey4)

[1.2.1 Build Essential 4](#_heading=h.x7i38f5oyr8f)

[1.2.2 Make 4](#_heading=h.30lr8s8xbdsy)

[1.2.3 Configure 5](#_heading=h.43eztki62kf2)

[2 Clase 14: Logs 5](#_heading=h.n8bb95oxctbe)

[2.1 Clasificación 6](#_heading=h.ov95m111mba1)

[2.2 Ventajas – Logs locales 6](#_heading=h.1erc31nie08o)

[2.3 Servidor de Logs 6](#_heading=h.4z0ihv7rkl0)

[2.4 Ventajas – Servidor 6](#_heading=h.ii8pi6xi6ffn)

[2.5 Rotación de Logs 7](#_heading=h.ol3bktocbiql)

[2.5.1 Logrotate 7](#_heading=h.k5thes2e1s6)

[2.6 Logs importantes de GNU/Linux 7](#_heading=h.tmgzzvcizfze)

[2.6.1 Logs “binarios” 8](#_heading=h.spnzbyxnzfmf)

[2.7 dmesg 8](#_heading=h.nupg8s427p1b)

[2.8 Herramientas gráficas de Logs 8](#_heading=h.mzl58ja9z075)

[2.9 Herramientas gráficas de Logs en Windows 9](#_heading=h.9opougxhg968)

[2.10 Prácticas de laboratorio 9](#_heading=h.pbywr2oi87av)

[3 Clase 15: Memorias 9](#_heading=h.fuhl2d4rwjiy)

[3.1 Swap 9](#_heading=h.6ygtvx2kgexz)

[3.2 Tamaño 11](#_heading=h.uo7jg5aiwpo8)

[3.3 Swappiness 11](#_heading=h.92ziwmm40gp5)

[3.4 Como crear una SWAP de tipo archivo 13](#_heading=h.kls2417ndqwp)

[3.5 Prácticas 13](#_heading=h.4g9rm02o68os)

[4 Clase 16: Jerarquía de Memoria, Cache y RAM 14](#_heading=h.xmf6jildrdak)

[4.1 Jerarquía de Memoria 14](#_heading=h.7qn8lluvdth0)

[4.2 Memoria Cache (L1, L2, L3) 15](#_heading=h.ma13bgxjkxfw)

[4.3 Memoria RAM y Conceptos Clave 15](#_heading=h.jga07r1wx5gj)

[4.4 Prácticas de Laboratorio (Linux) 16](#_heading=h.x82wpdq61e70)

[4.5 Prácticas 17](#_heading=h.14nd6aar8pro)

[4.6 Practicas resueltas 17](#_heading=h.z11xy1akqq5)

[5 Clase 17: Directorios 18](#_heading=h.s497vzq5qemb)

[5.1 Filesystem Hierarchy Standard FHS 18](#_heading=h.7xmgi7w3gjzw)

[5.2 Filesystem 19](#_heading=h.e9ee5fyft9ks)

[5.3 Montar FS 19](#_heading=h.psfbwh8ec6sk)

[5.4 Tipos de archivo en GNU/Linux 19](#_heading=h.4z0v66ou31b2)

[5.5 Comandos para manejo de Archivos 20](#_heading=h.bmgfp3igf4f6)

[5.6 Comandos para manejo de sistemas de archivos 22](#_heading=h.opjwqyvl7ojq)

[5.7 Como linux maneja el montado de sistemas de archivos 23](#_heading=h.yr6bovtr8v9k)

[5.8 Practicas 23](#_heading=h.fvvegfko1rxq)

[6 Clase 18: Particiones 24](#_heading=h.z77fcy6qxvtd)

[6.1](#_heading=h.dzslkpu6qjo6) 💾 Introducción a Particiones 24

[6.2](#_heading=h.qhr8mkirwwok) ⚖️ Tablas de Particiones: MBR vs. GPT 24

[∙](#_heading=h.1r4xp6yiqzy) 📏 Discrepancia de Capacidad de Almacenamiento 26

[6.3](#_heading=h.7w4jlkiovcx4) ⚙️ Esquema de Particionado Recomendado 27

[6.3.1 Espacio Swap Recomendado (Según RAM) 27](#_heading=h.4eadchs1nugq)

[6.4](#_heading=h.7vhwd75g9hql) 🛠️ Herramientas de Particionado 27

[6.5 Práctica: 28](#_heading=h.m5nl5brw6349)

[7 Clase 19: RAID 28](#_heading=h.h5fvairmppwt)

[7.1 Concepto de RAID 28](#_heading=h.2jzpyqe74v24)

[7.2 Niveles de RAID 29](#_heading=h.yhwvj05kpgu6)

[7.3 Demostración de Instalación de S.O. en RAID 0 (Debian) 31](#_heading=h.wdrhymalj44i)

[7.4 Falla de RAID 0 32](#_heading=h.i49nqw8ix5xe)

[7.4.1 Espacios de Almacenamiento en Microsoft Windows 32](#_heading=h.snknb2ops9cp)

[8 Clase 20: Raid 2: RAID Avanzados 33](#_heading=h.kacko16zewr8)

[8.1 Análisis de Niveles de RAID Avanzados 33](#_heading=h.gdetm21abw0n)

[8.2 RAID Anidados y Variantes 35](#_heading=h.h93i06s2ymbt)

[8.2.1 RAID 10 (RAID 1+0) 35](#_heading=h.s9645hbyixjs)

[8.2.2 RAID 5E y RAID 6E (Con Disco de Reserva) 35](#_heading=h.jnqlkrb187co)

[En el caso de combinar RAIDs con un 0, siempre el RAID 0 queda por arriba. Por debajo en el Stripping queda el otro tipo de raid. Similar al dibujo del RAID 10. 36](#_heading=h.sgjzxor0vi4w)

[8.3 Práctica de Laboratorio: Instalación de S.O. en RAID 1 (Debian) 36](#_heading=h.itq9y0h9ot39)

[8.4 Falla en RAID 1 37](#_heading=h.5e8j9izbp2ja)

[9 Clase 21: Administración Remota 39](#_heading=h.qsbkpuqtditi)

[10 Exposición: Escritorio Remoto en Windows 44](#_heading=h.p9nwdn7cqut3)

# Clase 12: Compresión y Compilación

## Algoritmos de compresión

**GZIP**: Reemplaza al programa compress de UNIX.

Extensión: .gz

**BZIP2**: Aplicativo libre, comprime y descomprime archivos según algoritmo de mayor porcentaje de comprensión. Basado en la licencia BSD, comprime más y consume más.

Extensión: .bz2

**XZ**: Proporciona compresión potente, a veces mejor q bzip2, pero devora memoria. Tiene una rápida y fácil descompresión.

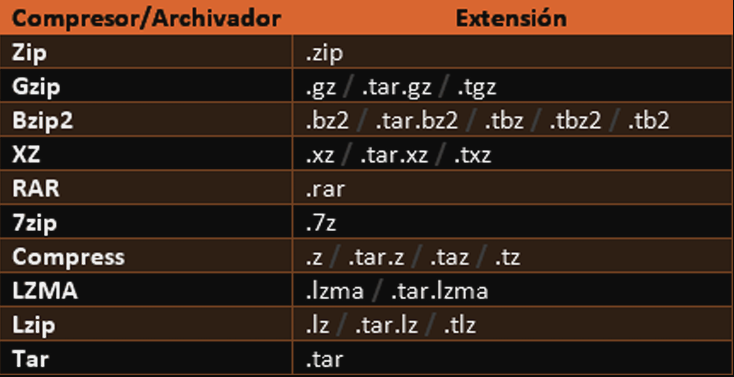
Extensión: .xz

**TAR**: Se usa ampliamente en entornos UNIX. Su principal función es empaquetar archivos.

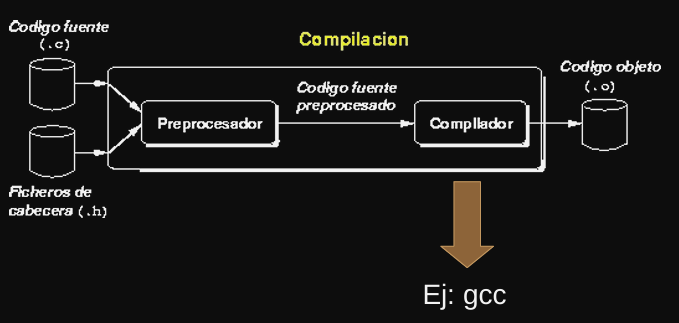
Extensión: .tar

**ZIP no es compatible con GZIP.**

La principal diferencia es que GZIP no empaqueta archivos, solo comprime.



## Compilar



### Build Essential

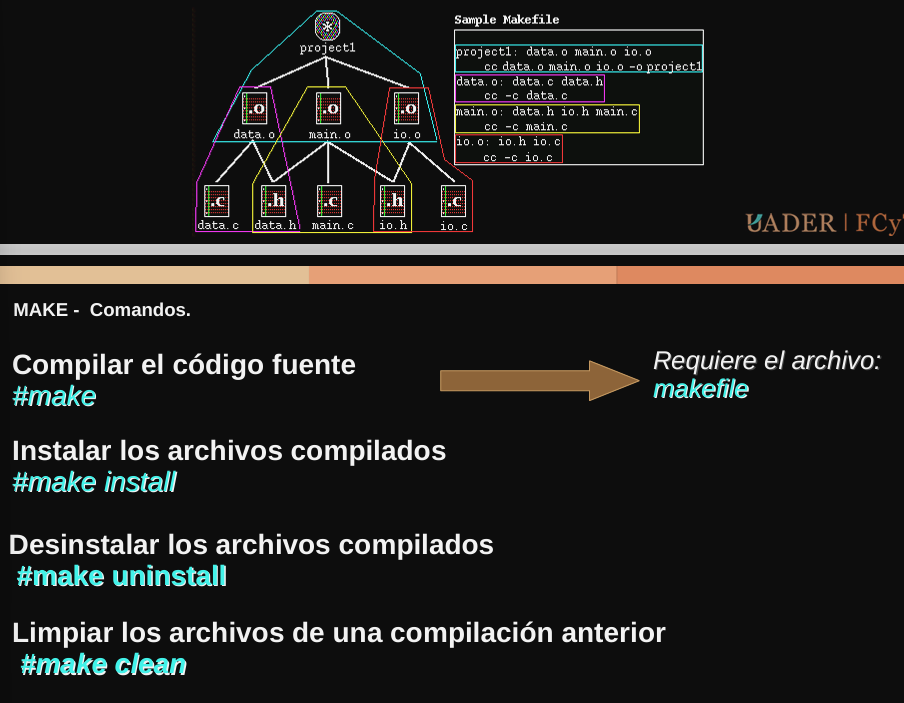
Instala paquetes necesarios para compilar un software.

Incluyen el depurador GNU, la colección de compiladores g++/GNU y algunas herramientas y bibliotecas más que se requieren para compilar un programa. Alguno de estos paquetes son: g++, gcc, libc6-dev, dpkg-dev y make.

### Make

Herramienta de gestión de dependencias.

Para prepararse para usar make, debe escribir un archivo llamado makefile que describa las relaciones entre los archivos en su programa y establezca los comandos para actualizar cada archivo

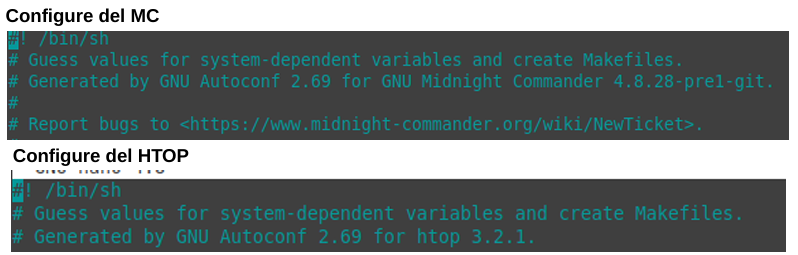


### Configure

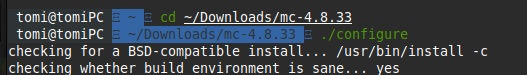
Es un script que se proporciona con el código fuente de la mayoría de paquetes, es quien prepara la construcción del software en su sistema específico.

Asegura tener todas las dependencias para el proceso de compilación e instalación, y descubre/obtiene variables dependientes del sistema que necesita.

Luego es quien crea el archivo Makefile, que necesita el **make**.



La practica es compilar un software.



# Clase 14: Logs

Son registros de eventos generados por sistemas informáticos. Estos archivos contienen información relacionada a un evento específico que ocurrió en un sistema.

Otra definición: Registro oficial de eventos durante un rango de tiempo en particular que se usa para registrar los datos o información sobre quien, qué, cuándo, dónde y porque un evento ocurre.

Who what when where why. La quintuple w.

## Clasificación

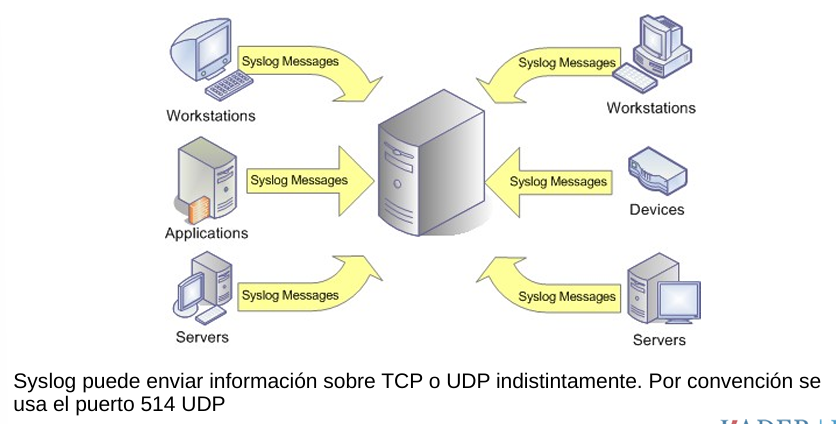
Logs de Sistema: Generados por un demonio. Registran en base al SO. Ej: Servicios, accesos al equipo, mensajes del sistema.

Logs de programa: Registran cronológicamente eventos importantes mientras usamos un programa. Los puede generar la propia app o un demonio.

## Ventajas – Logs locales

* Detección de ataques e intrusos.
* Detección de ataques e intrusos.
* Detección de problemas de hardware/software.
* Análisis forense de sistemas. Preservación de la información.
* Evaluación de vulnerabilidad.
* Cumplimiento de la normativa legal o log de eventos con fines de auditoría.

## Servidor de Logs



## Ventajas – Servidor

* Preservación de la información.
* Respaldo centralizado.
* Almacenamiento centralizado en base de datos.
* Exploración y visualización avanzada: gráficos, tablas, etc.
* Detección de anomalías integrales. Cumplir estándares de seguridad.
* Trazabilidad de los eventos.
* Optimización de los análisis.
* Filtrado avanzado.

Algunos ejemplos son LogAnalyzer, Splunk y Loggly.

## Rotación de Logs

Significa:

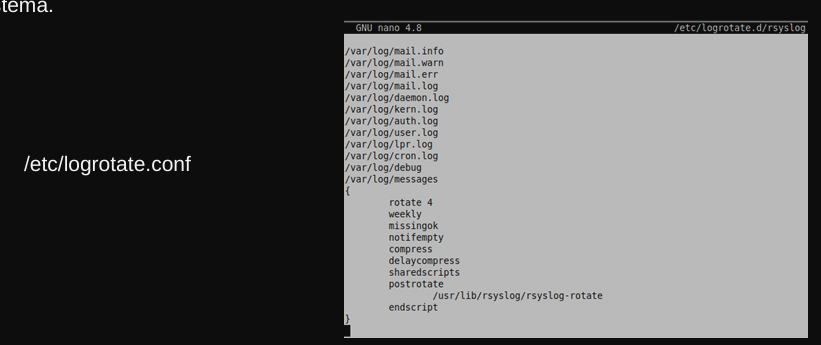
* Determinar el tamaño máximo permitido para un archivo de log.
* Decidir que hacer cuando alcance dicho tamaño (Borrar, renombrar, comprimir) y crear uno nuevo.
* Especificar cuanto tiempo mantendremos dichos registros.

Esto se hace porque sino se te puede ir al carajo el peso de los logs.

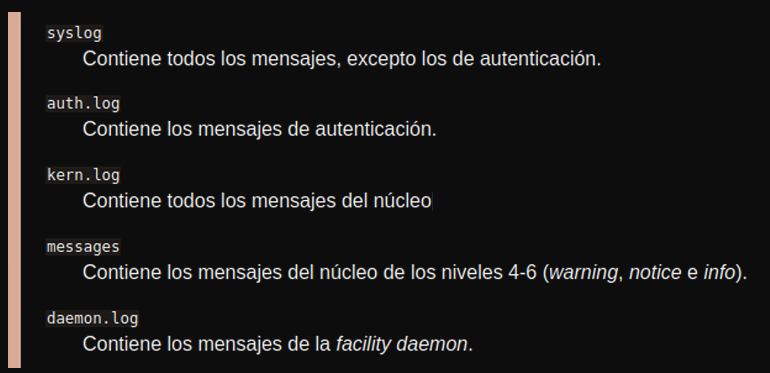


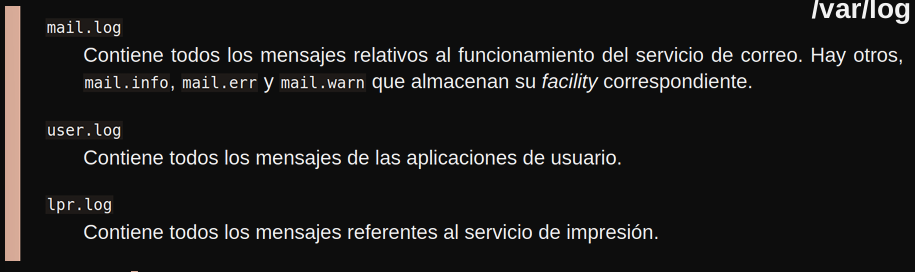
### Logrotate

Administra la compresión y rotación de archivos de logs en sistemas Linux.

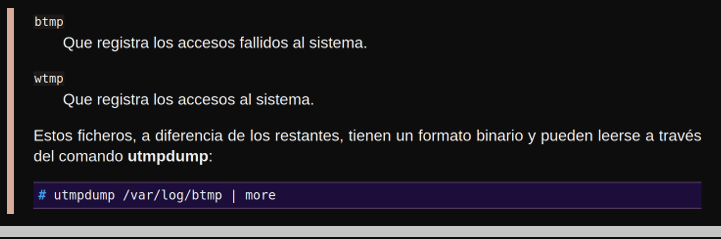


## Logs importantes de GNU/Linux



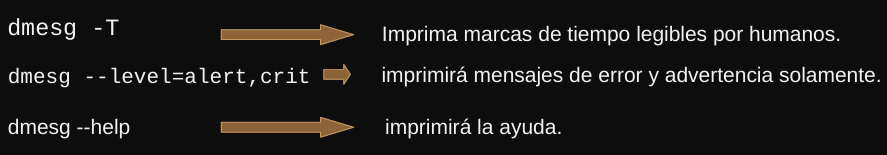


### Logs “binarios”



## dmesg

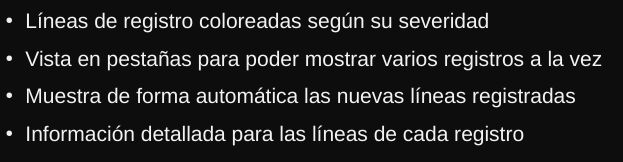
Lista el buffer de mensajes del kernel. Contiene gran variedad de mensajes importantes generados durante el arranque y depuración de aplicaciones. Demonios como el syslog la pueden guardar en el disco duro.



## Herramientas gráficas de Logs

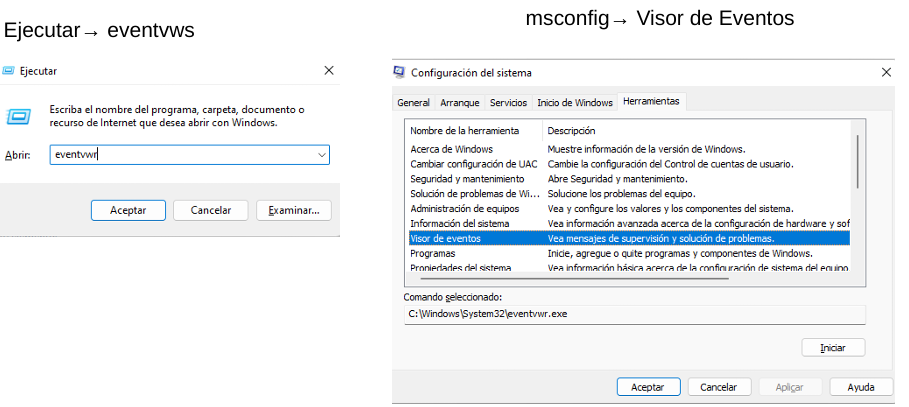
Gnome-Logs, KSystemLog.

KSystemLog muestra los registros del sistema, los agrupa por servicios generales y opcionales. Incluye funcionalidades piolas para leer los archivos de registro de forma agradable.



## Herramientas gráficas de Logs en Windows

Se utiliza el Visor de Eventos.

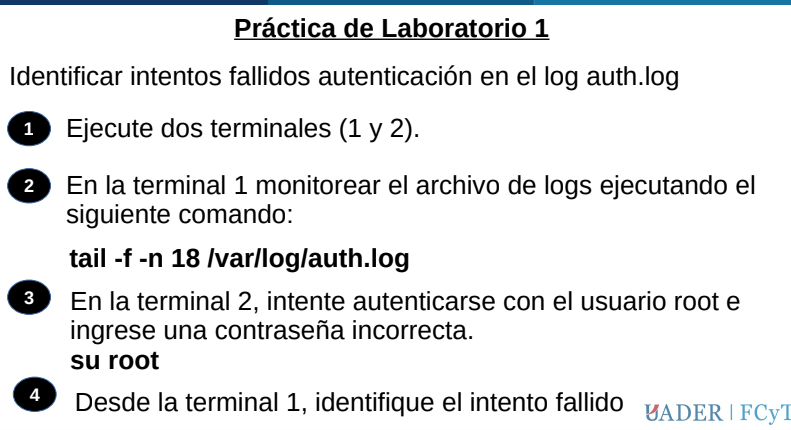


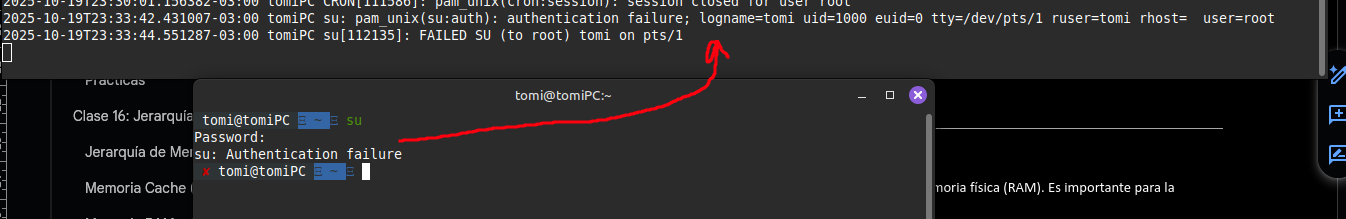
El visor de eventos puede registrar los siguientes eventos (5):

* Eventos reenviados: Vienen de otra máquina.
* Eventos de software: Engloba registros de nuestras aplicaciones. Los errores son los más importantes, y las advertencias le da un grado más bajo de riesgo.
* Eventos de seguridad: Registros de auditorias de inicio de sesión.
* Eventos del sistema.
* Eventos de configuración: Registros en relación a máquinas establecidas como controladores del dominio.

## Prácticas de laboratorio

Si las hago pongo aca los resultados jejejejjeje

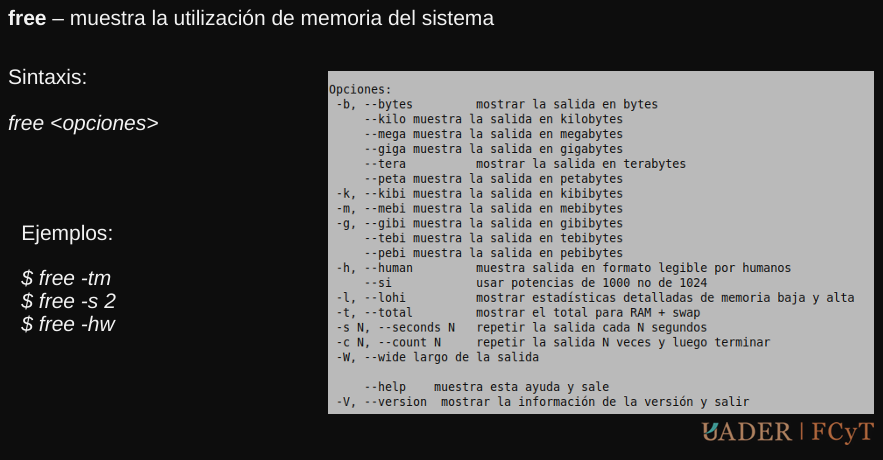
1) 

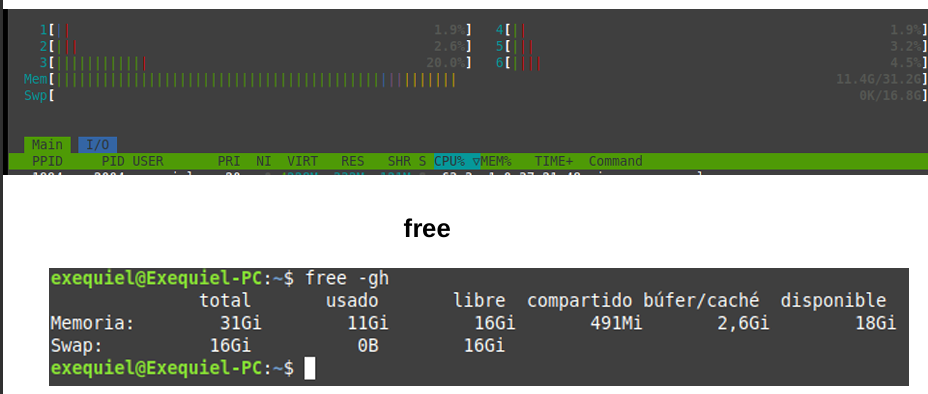


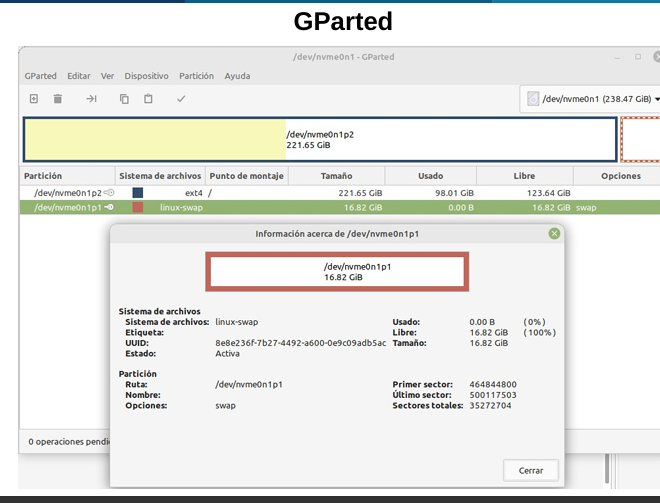
# Clase 15: Memorias

## Swap

Es un espacio que usa el SO si se queda sin memoria física (RAM). Es importante para la suspensión y hibernación también.







## Tamaño

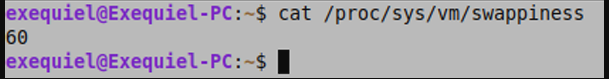
El tamaño del swap recomendado es x1 o x0.5 veces la cantidad de la RAM del sistema, todo depende del uso.

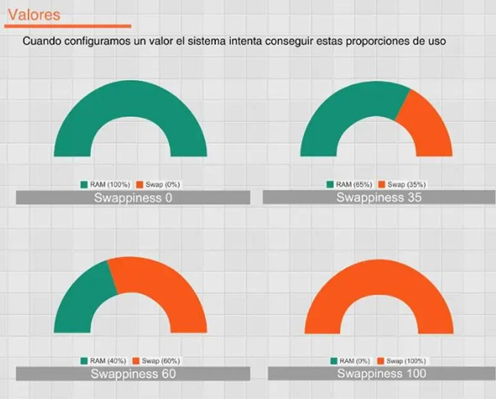
* Si se piensa usar hibernación o suspensión, es recomendable x1.
* Mientras más RAM, no es necesario tener tanto SWAP.
* Depende de la memoria que ocuparán las apps que pretende usar.
* Depende el espacio en el disco que uno tiene disponible.

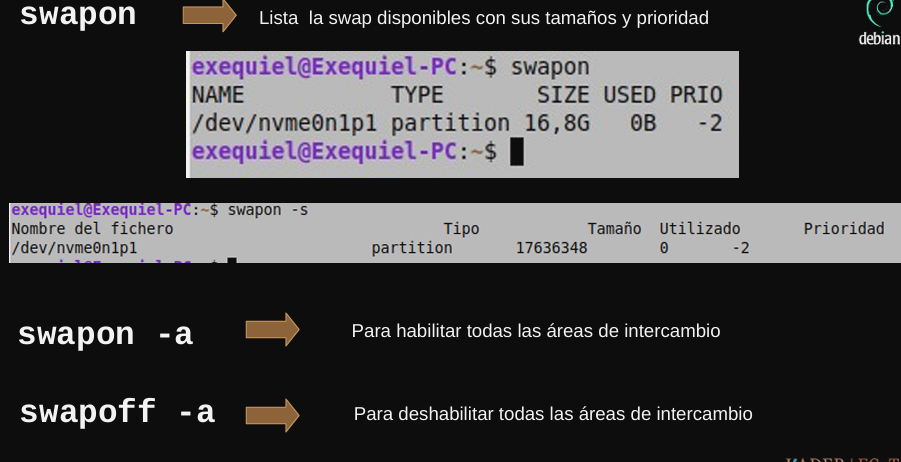
## Swappiness

Propiedad del kernel de Linux, permite ajustar el equilibro entre el uso del SWAP y la memoria RAM. Toma valores de 0 a 100. A mayor valor el swapiness, más agresivo se vuelve el sistema para mandar cosas al SWAP. Valores cercanos a 0 hacen que mantenga todo en la RAM.

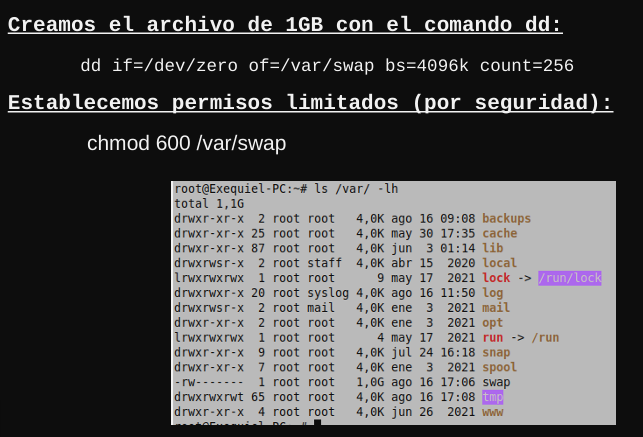
El valor por defecto es 60.

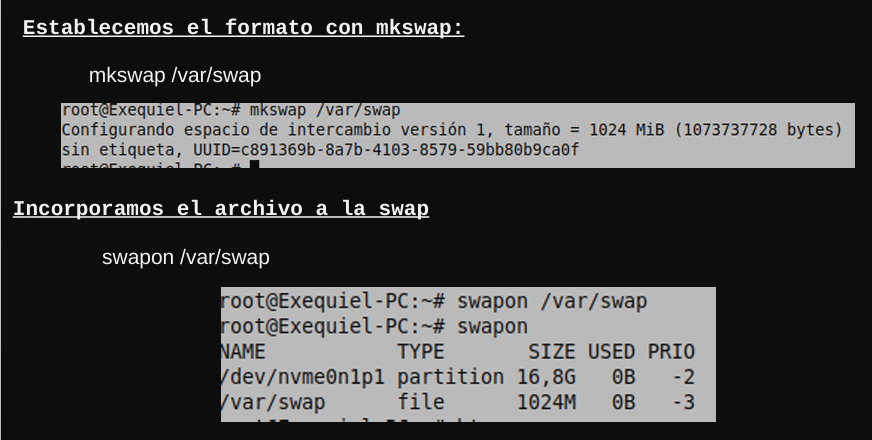




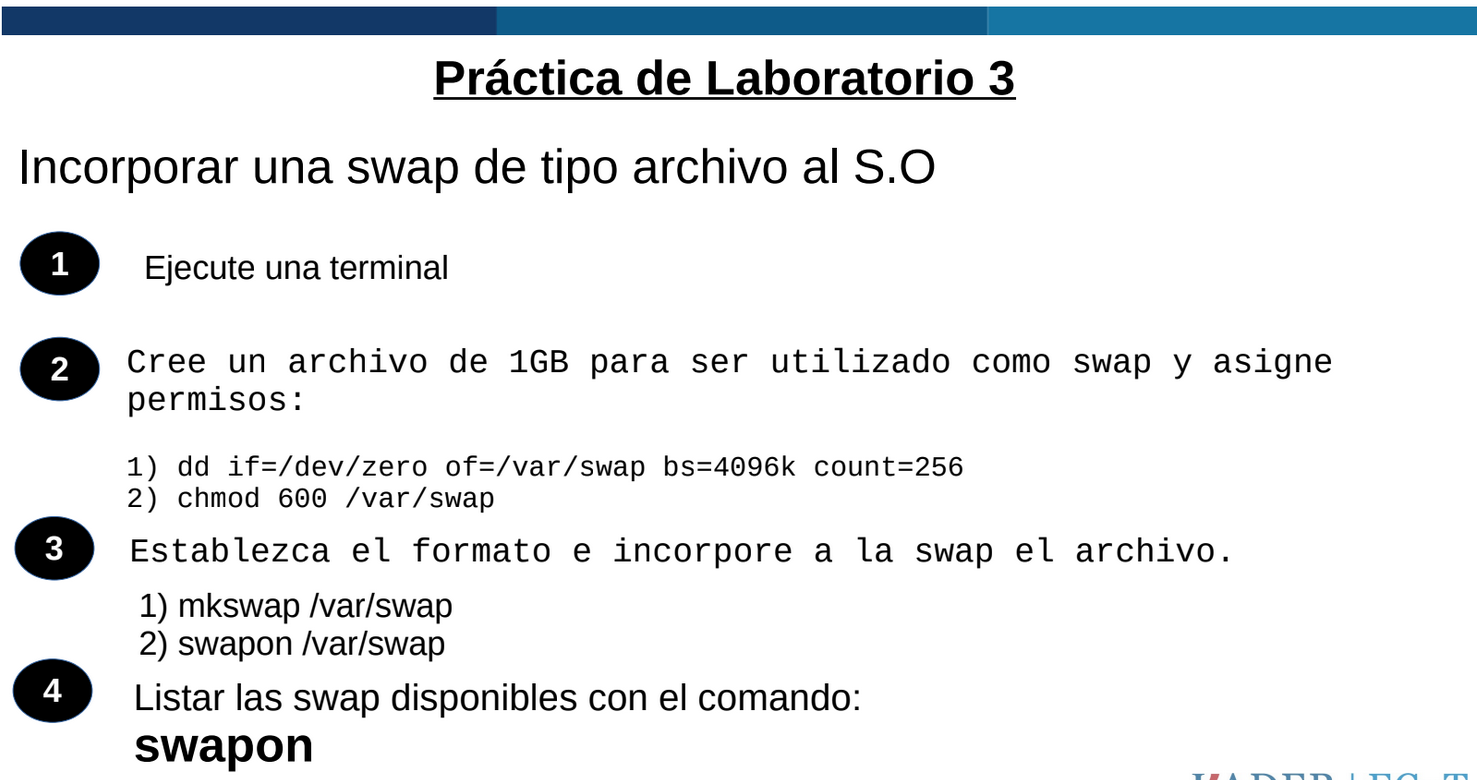


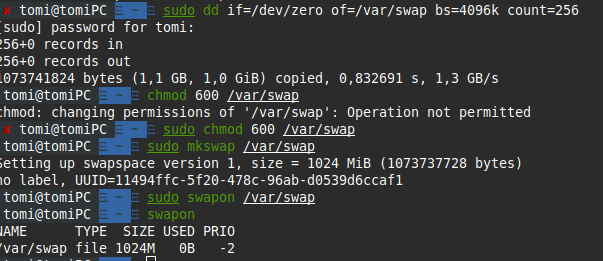
## Como crear una SWAP de tipo archivo





## Prácticas





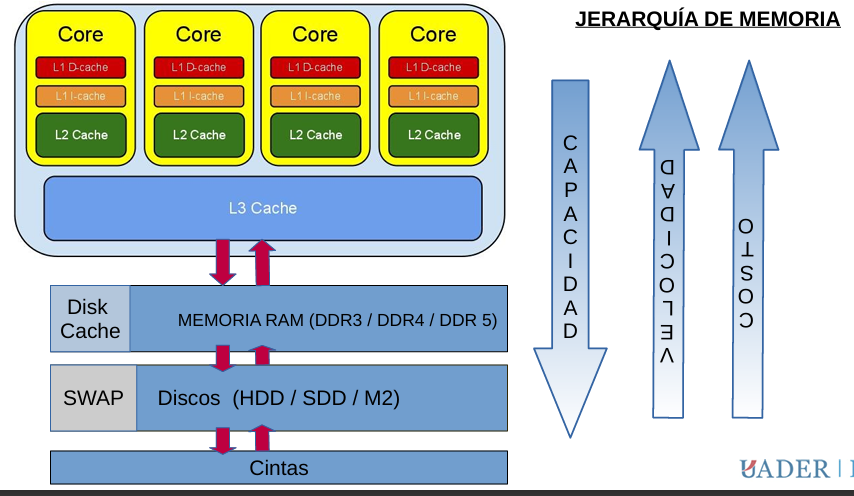
# Clase 16: Jerarquía de Memoria, Caché y RAM

El PDF es una presentación de práctica sobre **Sistemas Operativos** que se centra en la **Jerarquía de Memoria** y el análisis de la memoria **Cache** y **RAM** utilizando comandos de laboratorio, principalmente en sistemas basados en Linux.

Aquí tienes un resumen completo del contenido:

## Jerarquía de Memoria

El sistema de memoria se organiza en capas jerárquicas. Las capas superiores tienen mayor velocidad, menor capacidad y mayor costo por bit que las capas inferiores.

La jerarquía incluye niveles como la caché L1 (I-cache y D-cache), L2 Cache, L3 Cache, Memoria RAM, SWAP, Discos y Cintas.

## Memoria Cache (L1, L2, L3)

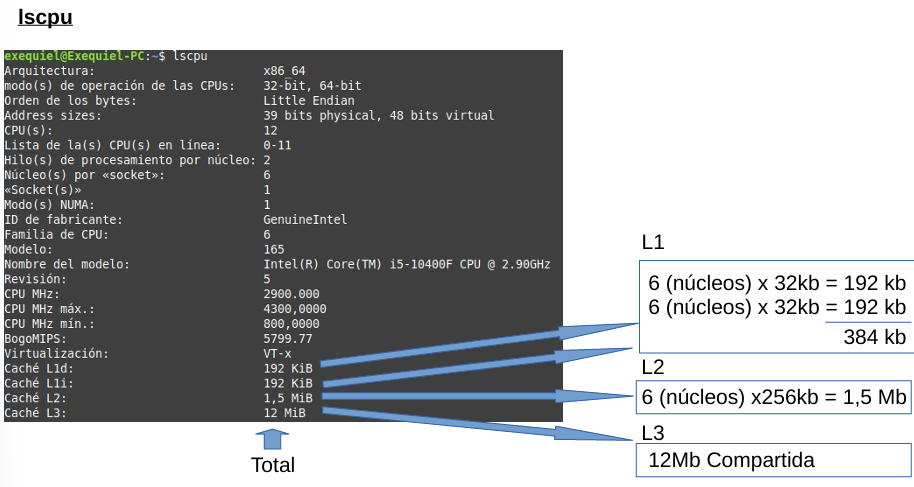
Los niveles de caché (L1, L2, L3) tienen las siguientes características para un ejemplo de CPU (Intel Core i5-10400F):

* **L1 Data (L1D)**: 6 núcleos x 32 KB = **192 KiB**.
* **L1 Instruction (L1I)**: 6 núcleos x 32 KB = **192 KiB**.
* **L2 Cache**: 6 núcleos x 256 KB = **1.5 MiB**.
* **L3 Cache**: **12 MiB** (Compartida).

En términos de rendimiento (ejemplo de AIDA64):

* **L1 Cache** tiene la menor latencia (1.0 ns).
* **L3 Cache** tiene la mayor latencia de las memorias caché (19.0 ns).

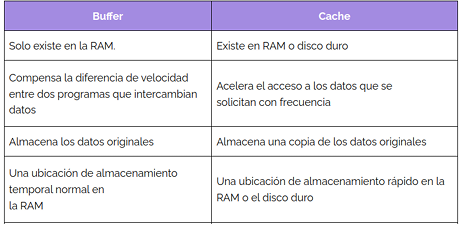
**Esto dijo que lo tomaba, averiguar de cuanto eran las memorias Cache.**

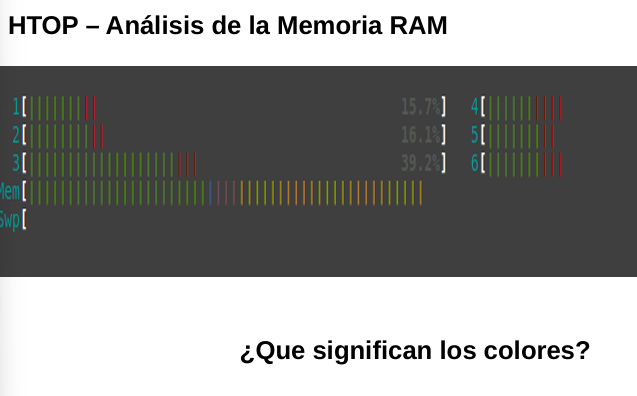


## Memoria RAM y Conceptos Clave

Se introducen y analizan los conceptos de Bufferd, Shared y Cache en el contexto de la memoria RAM:

* **Cache:** Almacena temporalmente una **copia** de un recurso para que las solicitudes posteriores se procesen más rápido. Puede existir en la RAM o en el disco duro.
* **Buffer:** Es un área de almacenamiento temporal en la RAM que almacena la **transferencia de datos** entre dispositivos o aplicaciones. Compensa la diferencia en las velocidades de transferencia, y almacena los datos **originales**. Solo existe en la RAM.
* **Shared (Compartida):** Permite que dos o más procesos compartan una zona común de memoria y es la forma más rápida de Comunicación Interprocesos (IPC).

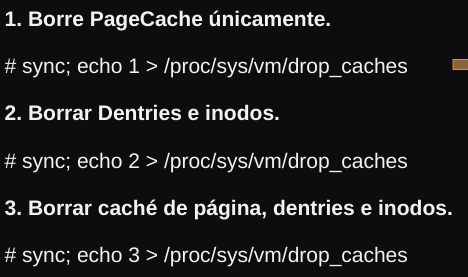


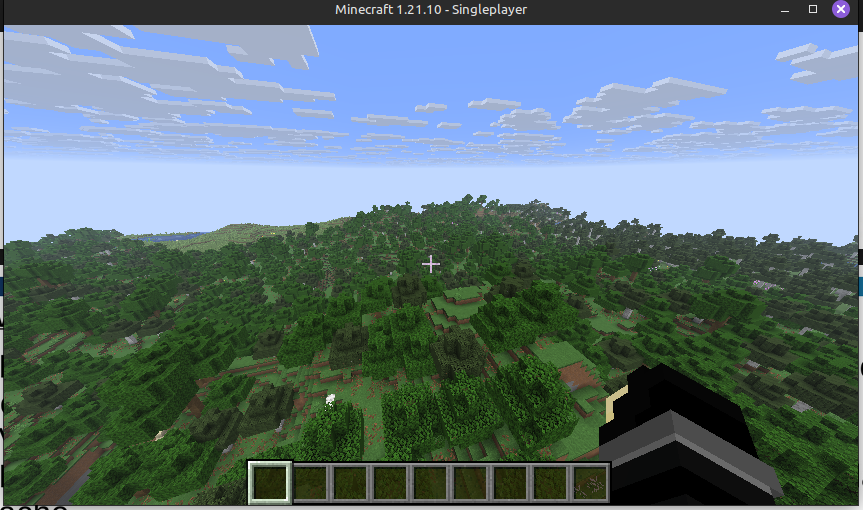


Verde = usado, Azul = buffered, Violeta = Shared, amarillo/naranja cache.

## Prácticas de Laboratorio (Linux)

La agenda incluye varias prácticas de laboratorio utilizando comandos específicos:

* **Comando lscpu (Práctica 1):** Utilizado para identificar el tamaño de la memoria caché del CPU (L1, L2, L3).
* **Comando htop y free (Prácticas 2 y 3):** Utilizados para identificar y monitorear la memoria RAM (Total, Usada, Shared, Buffer, Cache y Disponible).
* **Comando sync:** Fuerza una escritura inmediata de todos los datos almacenados en caché a disco, asegurando que los datos se escriban si el sistema se vuelve inestable.
* **Liberar Memoria Caché (Práctica 3):** Hay tres opciones en Linux para borrar el caché sin interrumpir procesos.
* 
* **RAM Disk con tmpfs (Práctica 4):** tmpfs es un sistema de archivos que guarda todos sus archivos en la memoria virtual (RAM), mostrando las páginas como "Shmem" o "shared" en la información de memoria.
  + Para crear y montar una RAM Disk de 16G, se usan los comandos: # mkdir /tmp/ramdisk y # mount -t tmpfs -o size=16G tmpfs /tmp/ramdisk37373737.
  + **Importante:** Los datos en RAM Disk desaparecen al apagar o reiniciar el equipo, ya que la RAM es volátil.



minecraft en una RAMdisk va re copado mal

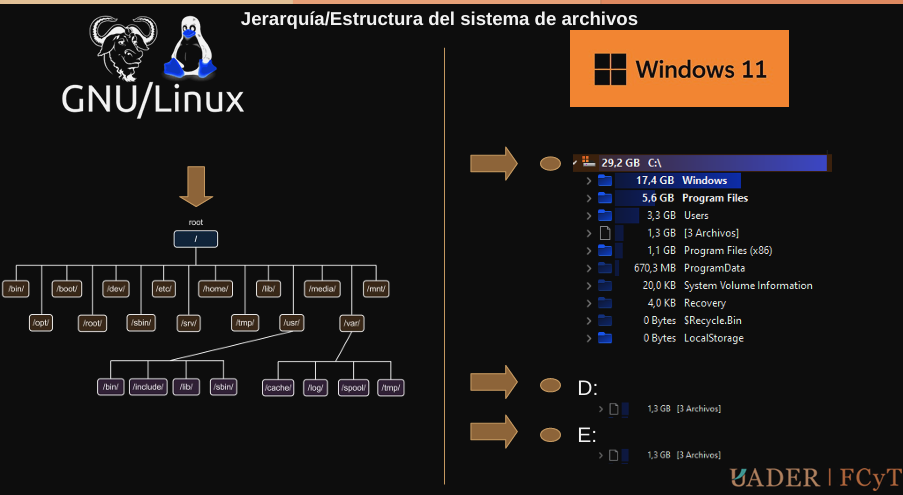
## Prácticas

Se pide investigar y explicar la solapa "Rendimiento -> Memoria" del Administrador de tareas en **Microsoft Windows**. También se solicita visualizar la memoria caché del CPU (L1, L2, L3) en Windows, copiar archivos grandes y explicar cómo esto afecta a la caché de la memoria.

## Practicas resueltas

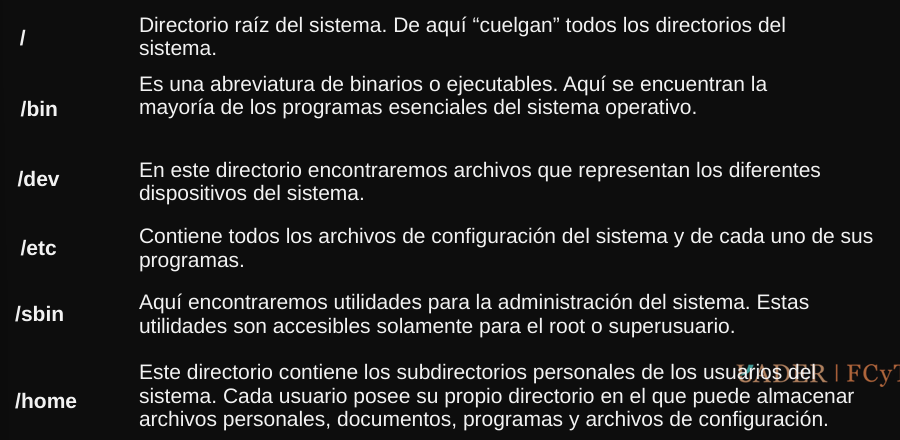
Zzzzzzzz

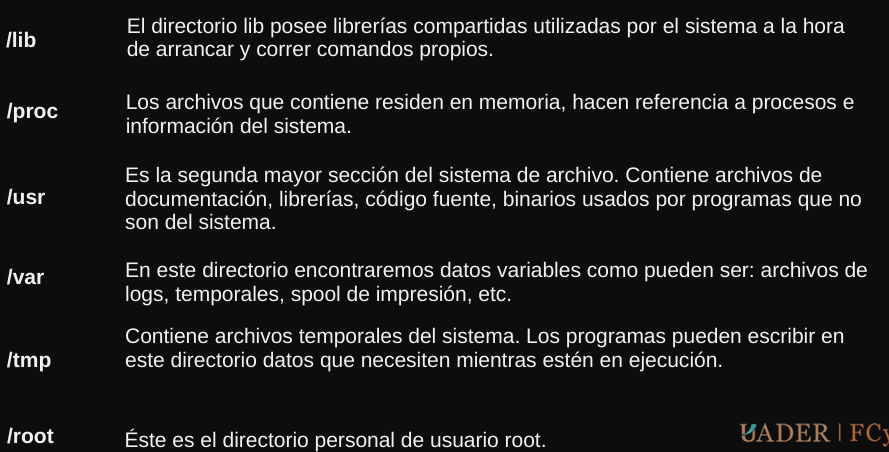
# Clase 17: Directorios



## Filesystem Hierarchy Standard FHS

Es una norma que define los directorios principales y sus contenidos en un sistema operativo GNU/Linux y otros sistemas de la familia Unix.





## Filesystem

**Que es**: Abstracción para manipular datos almacenados en un espacio de disco de forma determinada.

**Que nos permite:** Tratamiento consistente de la información almacenada físicamente en disco. Una jerarquía de archivos y protección de los archivos.

**Que es un FS con Journaling:** Da mayor seguridad a la integridad de los datos que contenemos en nuestros discos rígidos. Recomendado para sistemas de alta disponibilidad.

Sistemas sin Journaling: Ext2, FAT(12,16,32).

Sistemas CON Journaling: ext3, ReiserFS, XFS, JFS, NTFS, ext4.

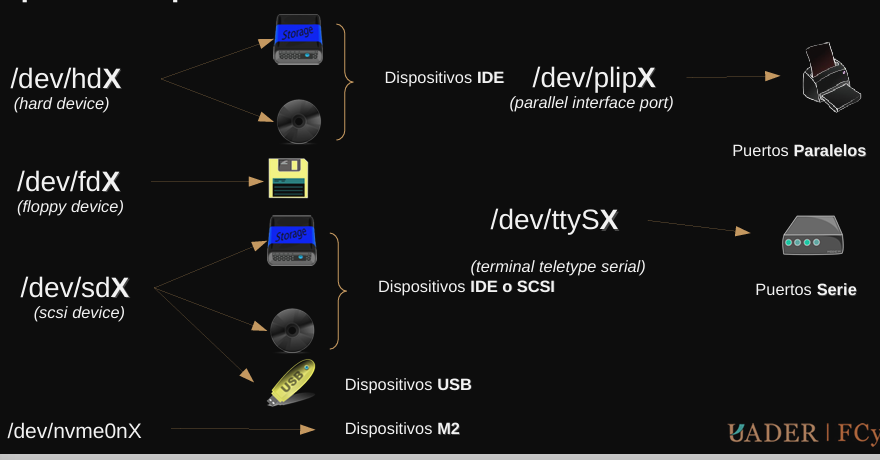
## Montar FS

GNU/Linux no trabaja con unidades como lo hace Windows.

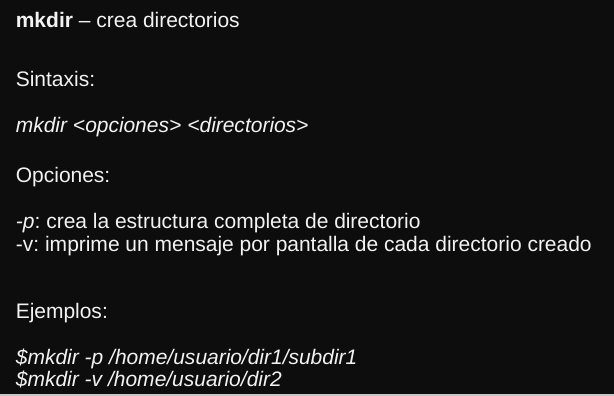
Entonces, para poder acceder a un dispositivo (disco rígido, cdrom, pendrive) necesitamos incorporarlo a nuestro sistema de archivos. Para esto nos valemos de los comandos **mount** y **umount**. Ellos se encargan de poder acceder a un dispositivo, a través de un directorio en nuestro sistema (normalmente ubicados en /media).

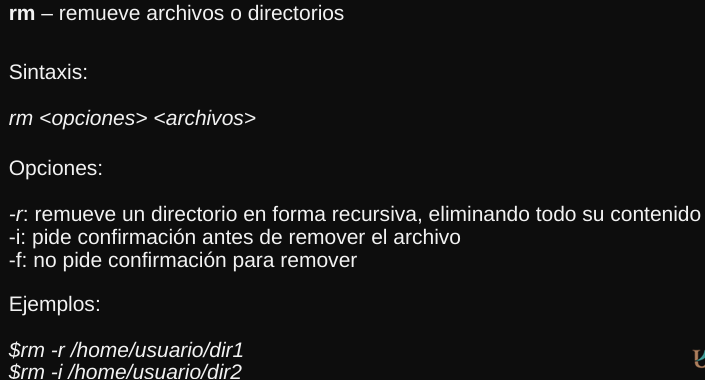
## Tipos de archivo en GNU/Linux

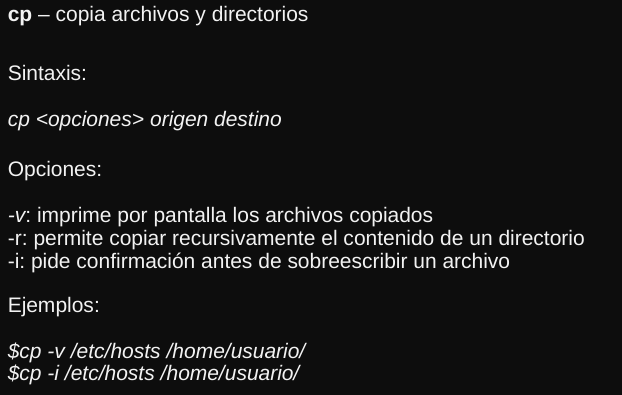
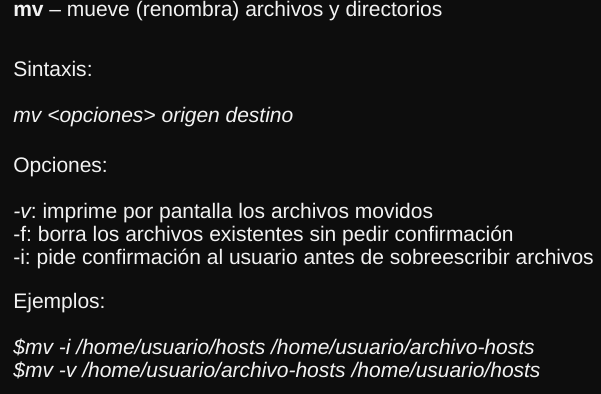
* Archivos o ficheros ordinarios: Archivos de texto, documentos, exes, etc.
* Directorios: También conocidos como carpetas y son tipos de archivos especiales que agrupan otros ficheros de forma estructurada.
* Archivos de Dispositivos: Representan dispositivos conectados al sistema.
  + Seriales: Lee o escribe carácter por carácter (Teclados, mouses, modems)
  + Bloque: Permiten leer y/o escribir datos por bloques de forma aleatoria (Discos, memorias, etc)
  + Especiales: Para pantallas, dispositivos gráficos, etc.

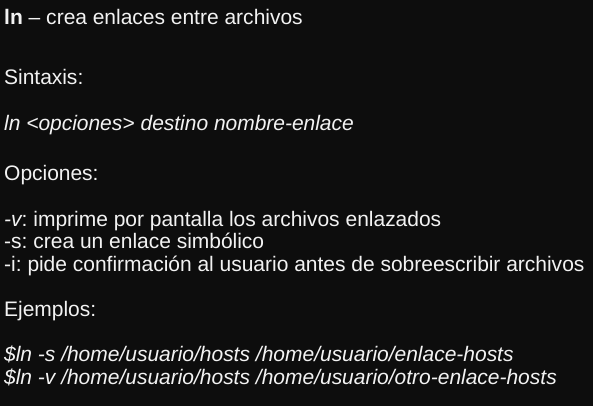


## Comandos para manejo de Archivos

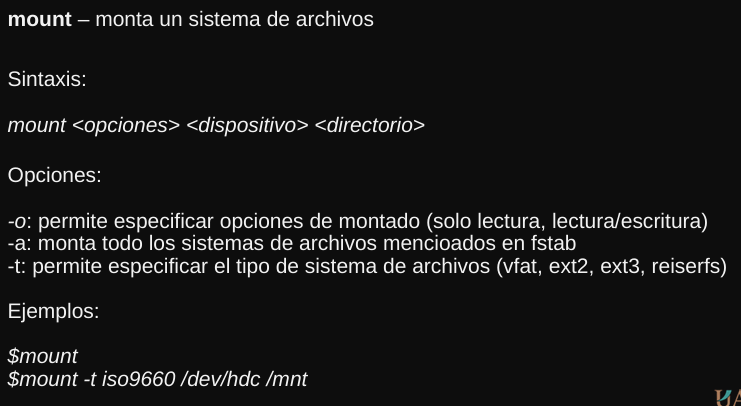


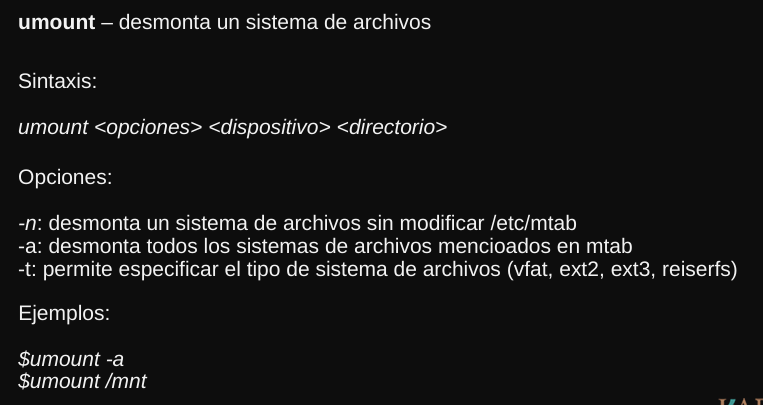


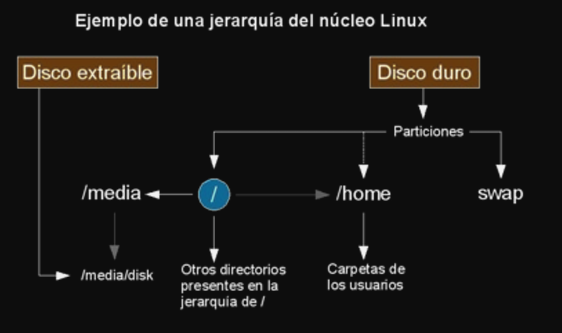




## Comandos para manejo de sistemas de archivos







## Como linux maneja el montado de sistemas de archivos

Utiliza el archivo /etc/fstab (filesystem table).

En él se especifica la ubicación de cada sistema de archivos y sus opciones de montado. Se usa principalmente en la carga del sistema operativo para el montado de particiones en el árbol de directorio de Linux.

Sirve como atajo a la hora de montar dispositivos o sistemas de archivos

## Practicas

Zzzzzzzzzzzz (Quedan las 3 pendientes)

# Clase 18: Particiones

## 💾 Introducción a Particiones

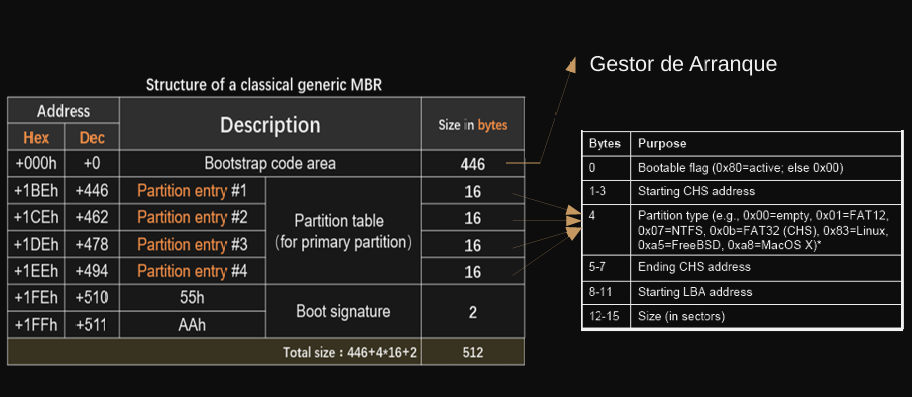
Una **partición de disco** es cada división lógica en una única unidad física de almacenamiento. Los dispositivos pueden dividirse en una o más particiones, cada una con su propio sistema de archivos independiente.

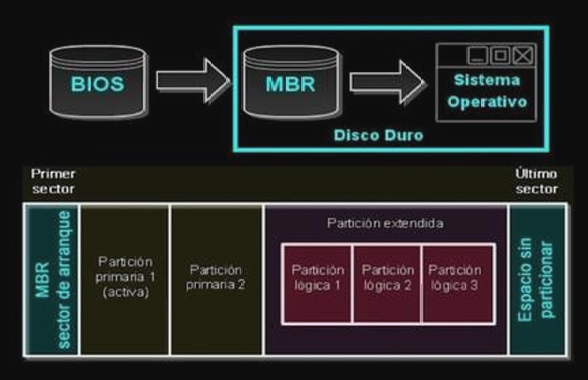
Las particiones se utilizan para:

* Guardar datos sensibles o realizar copias de seguridad de forma aislada.
* Instalar diferentes sistemas operativos en la misma unidad física.

## ⚖️ Tablas de Particiones: MBR vs. GPT

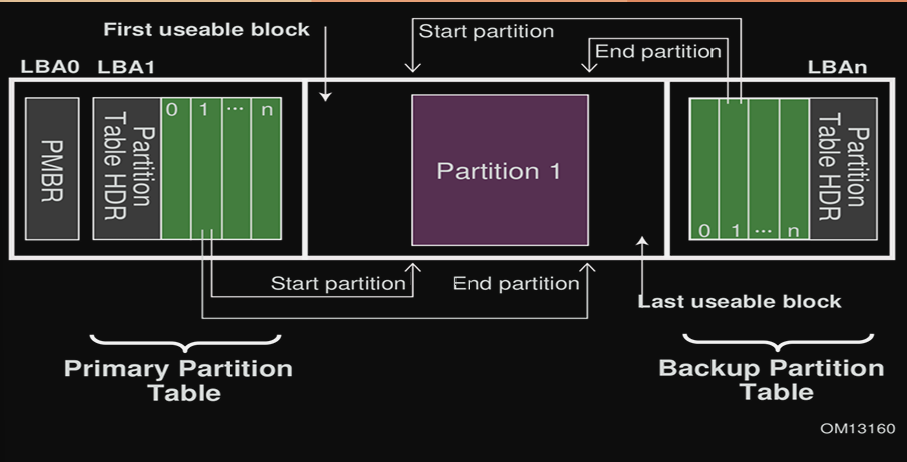
| **Característica** | **MBR (Master Boot Record)** | **GPT (GUID Partition Table)** |
| --- | --- | --- |
| **Arranque con** | **BIOS** (Basic Input Output System) | **UEFI** (Unified Extensible Firmware Interface) |
| **Ubicación** | Primer sector del disco (Sector 0). | Utiliza LBA (Logical Block Addressing). Arranca en el Sector 0 también. |
| **Límite de Tamaño** | No puede direccionar más de **2 TiB** (aproximadamente 2.2 TB). | Límite máximo direccionable de **2.2 ZiB** (casi ilimitado). |
| **Particiones Primarias** | Máximo de **4** (o 3 Primarias y 1 Extendida). | Por defecto, permite hasta **128** particiones primarias. |
| **Estructura** | Incluye el código de *Bootstrap*, la Tabla de Particiones y la Firma de Arranque. | Incluye **MBR Protector** (LBA 0), **Cabecera GPT Primaria** (LBA 1) y **GPT Secundaria** (Tabla de respaldo en el último sector). |





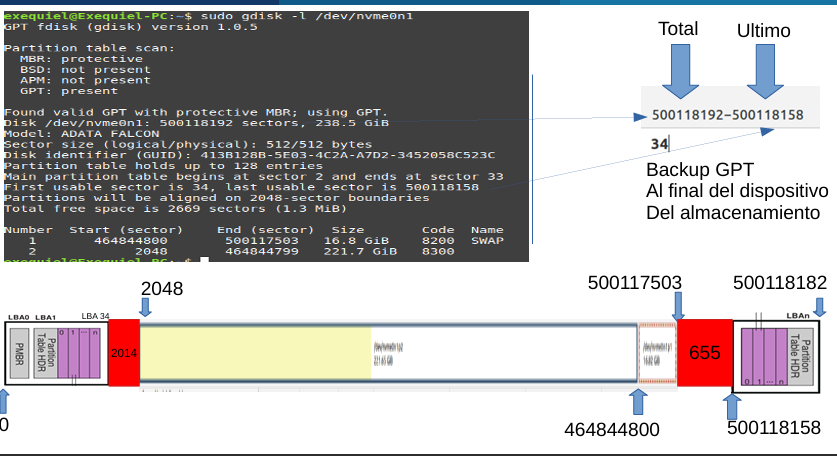
Los discos GPT utilizan LBA (Logical Block Addressing). Se suele organizar asi:

* **LBA 0 “MBR Protector”**: Este primer sector esta reservado para compatibilidad con datos MBR .
* **LBA 1 “Cabecera GPT Primaria”:** Contiene el GUID del disco, ubicación de tabla de particiones primaria, la ubicación de la cabecera GPT Secundaria y sumas de comprobación CRC32 de sí misma y de la tabla de particiones primaria. Especifica el número de entradas de partición en la tabla.
* **LBA 2 hasta LBA 33:** Las 128 entradas de particion por defecto, cada una con su tamaño de entrada 128 bytes. Su GUID de tipo de partición y su GUID de partición única.
* **GPT Secundaria:** Idéntica a la primaria, se usa como tabla de respaldo.
* **Cabecera Secundaria de GPT:** Idéntica a la secundaria, se usa como backup.



Es por esto, que si miramos las particiones del disco con un software del estilo Gparted, podemos ver que aunque sumemos el total de los sectores que usan nuestras particiones, no llegamos a el total de sectores de nuestro disco.

Y es que, la diferencia nos debería dar 34, ya que 34 sectores son los que quedan reservados para el GPT.



## 📏 Discrepancia de Capacidad de Almacenamiento

Existe una diferencia entre la capacidad que anuncia el fabricante (sistema decimal) y la que reporta el sistema operativo (sistema binario):

* **Fabricantes (Sistema Decimal)**: 1 GB = $10^9$ bytes (1,000,000,000 bytes).
* **Sistemas Operativos (Sistema Binario)**: 1 GB = $2^{30}$ bytes (1,073,741,824 bytes).

Esto causa que, por ejemplo, un disco de 500 GB (decimal) sea reportado por el sistema operativo como aproximadamente **465 GB** (binario).

## ⚙️ Esquema de Particionado Recomendado

Para sistemas operativos modernos (x86, AMD64, Intel 64), se recomienda crear las siguientes particiones principales:

* Partición **swap** (área de intercambio).
* Partición **/boot** (contiene archivos necesarios para el arranque).
* Partición **/** (raíz del sistema).
* Partición **home** (datos del usuario).

### Espacio Swap Recomendado (Según RAM)

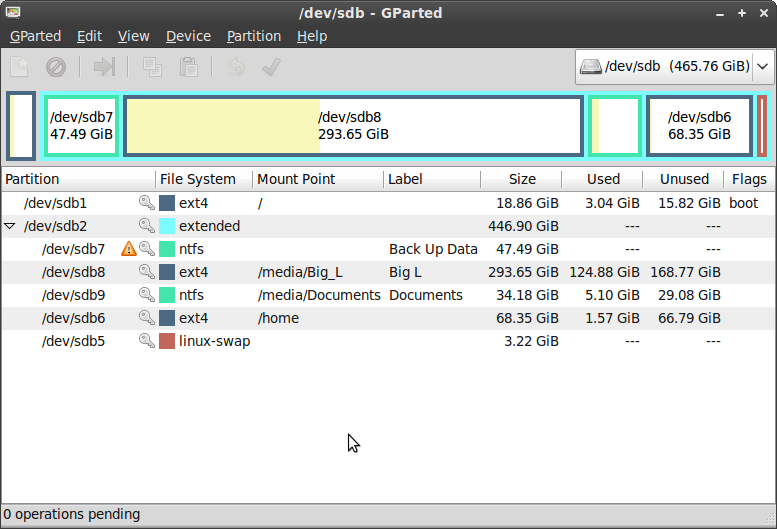
| **RAM Instalada** | **Espacio swap Mínimo** |
| --- | --- |
| 4 GB o menos | 2 GB |
| 4 GB a 16 GB | 4 GB |
| 16 GB a 64 GB | 8 GB |
| 64 GB a 256 GB | 16 GB |
| 256 GB a 512 GB | 32 GB |

## 🛠️ Herramientas de Particionado

| **Tipo de Herramienta** | **Ejemplos** |
| --- | --- |
| **Gráficas** (GUI) | Gparted, gnome-disk-utility, KDE Partition Manager. |
| **Consola** (CLI) | fdisk, cfdisk, **parted**, **lsblk**, blkid, **df**. |

## Práctica:

Analiza esta tablita, identifica los discos y tecnología, tamaño, tipo de tabla (que no lo dice la captura xD pero podemos darnos cuenta cual es), tamaño Swap de partición, y luego por cada partición el tamaño, el Filesystem y donde esta montado.



# Clase 19: RAID

## Concepto de RAID

* **Definición:** RAID (Matriz Redundante de Discos Independientes, del inglés *Redundant Array of Independent Disks*) es un método que busca mejorar el rendimiento de varios discos de almacenamiento conectados a una computadora.
* **Funcionamiento:** Varios discos físicos son presentados como un único disco lógico. Los datos se almacenan de forma distribuida (*stripping*), y se utiliza redundancia para poder reconstruir los datos.
* **Objetivos:** Evitar la pérdida de datos y asegurar su disponibilidad en caso de fallo de un disco duro. Esto se logra almacenando los datos en varios discos físicos con un nivel configurable de redundancia.
* **Beneficios:** Mayor rendimiento y más seguridad.
* **Implementación:** Se puede implementar por *software* (más flexible) o por *hardware* (mayor rendimiento). El material se enfoca en RAID por software.
* **Modo Degradado:** Un sistema RAID con suficiente redundancia puede mantenerse operativo (modo degradado) al fallar un disco, permitiendo que las aplicaciones sigan accediendo a los datos, aunque con un posible impacto en el rendimiento. Se debe reemplazar el disco fallado rápidamente para reconstruir los datos y volver al modo seguro.

## Niveles de RAID

Existen diferentes clasificaciones de niveles de RAID:

* **Estándares:** Desde RAID 0 hasta RAID 6.
* **Anidados:** Como RAID 0+1, 1+0, 30 y 50.
* **Propietarios:** Como Linux MD y RAIDZ.

#### Comparación de niveles de RAID

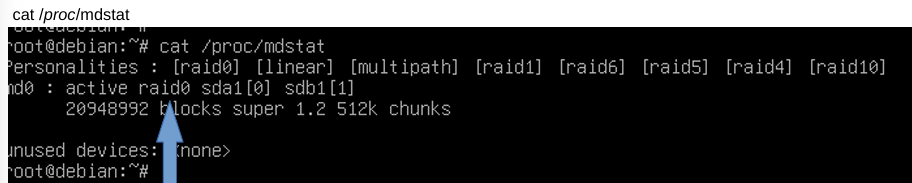
| **Nivel** | **Nombre** | **Mínimo de Discos** | **Características Principales** |
| --- | --- | --- | --- |
| **RAID 0** | Conjunto dividido (*Stripping*) | 2 | Divide los datos en tiras alternadas entre los discos para aumentar la tasa de transferencia y el tiempo de acceso, pero **no provee redundancia** ni tolerancia a fallos. El sistema falla si un disco deja de funcionar. Requiere que los discos tengan el mismo tamaño para no perder rendimiento. |
|  | | | |
| **RAID 1** | Conjunto en espejo (*Mirroring*) | 2 | Almacena los datos de forma idéntica en dos (o más) discos. Si un disco falla, los datos siguen disponibles en el otro. Protege la información en caso de falla y evita interrupciones. |
|  | | | |
| **RAID 4** | Conjunto dividido con disco de paridad dedicado | N+1 | Utiliza N discos para datos y un disco extra dedicado para paridad. Protege contra el fallo de un solo disco, permitiendo la reconstrucción de datos. No es costoso, pero el disco de paridad recibe muchas escrituras, lo que puede reducir su vida útil y reducir la velocidad de escritura. Ideal para ficheros grandes. NO SE USA en la actualidad. |
|  | | | |
| **RAID 5** | Conjunto dividido con paridad distribuida | N+1 | Distribuye los bloques de paridad en todos los discos N+1, evitando el cuello de botella de RAID 4. El sistema continúa funcionando con el fallo de hasta un disco. Presenta alta velocidad de transferencia de datos y es tolerante a fallos con una unidad averiada. |
|  | | | |

## Demostración de Instalación de S.O. en RAID 0 (Debian)

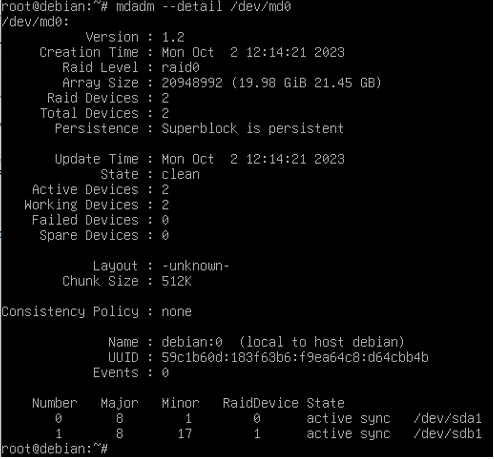
La práctica de laboratorio consistió en instalar Debian GNU/Linux en un array RAID 0 por software utilizando el instalador del S.O.

1. Se creó una Máquina Virtual (VM) con dos discos de 10 GB cada uno (sda y sdb).
2. Durante la instalación, se creó una nueva tabla de particiones en cada disco.
3. Se seleccionó la opción **Configurar RAID por software** en el menú de particionado.
4. Se eligió **Crear un dispositivo MD** y se seleccionó el tipo **RAID0**.
5. Se seleccionaron los espacios libres de /dev/sda y /dev/sdb como dispositivos activos para el array RAID 0.
6. Se confirmaron los cambios para escribir las tablas de particiones y configurar RAID.
7. Se finalizó el particionado, creando un dispositivo RAID 0 de 21.5 GB que incluye particiones primarias tipo raid en ambos discos.
8. Se configuró la partición del Dispositivo RAID 0 (#1) para ser usada como **sistema de ficheros ext4 transaccional** y se le asignó el **punto de montaje /** (sistema de ficheros raíz).
9. Finalmente, se confirma la escritura de los cambios y el formateo de la partición RAID 0 como ext4 para iniciar la instalación del sistema base.

El archivo mdstat nos dice los volúmenes existentes y sus estados.



mdadm --detail /dev/md0



## Falla de RAID 0

* **Simulación de Falla (Desconexión de un disco):** Al desconectar un disco desde VirtualBox, se genera un error de I/O, el *journal* es abortado, y el sistema de ficheros se **remonta como solo lectura** (read-only).
* **Estado de Falla ($mdadm --detail /dev/md0$):** El estado del array cambia a clean, FAILED, y reporta que el dispositivo /dev/sdb1 está missing (perdido), confirmando la falta de tolerancia a fallos de RAID 0.

### Espacios de Almacenamiento en Microsoft Windows

Microsoft Windows ofrece tres tipos de espacios de almacenamiento:

1. **Espacios Simples:**
   * **Objetivo:** Aumentar el rendimiento.
   * **Protección:** No protegen los archivos en caso de fallos de la unidad.
   * **Uso Recomendado:** Datos temporales (ej. archivos de renderizado, archivos temporales de editores de imagen).
   * **Requisito Mínimo:** Al menos dos unidades para ser útiles.
2. **Espacios de Reflejo (*Mirroring*):**
   * **Objetivo:** Aumentar el rendimiento y proteger los archivos (al guardar varias copias).
   * **Tipos:**
     + *Reflejo Doble:* Realiza dos copias, tolera el error de una unidad. Requiere al menos dos unidades (simetría de dos vías).
     + *Reflejo Triple:* Tolera errores en dos unidades. Requiere al menos cinco unidades (simetría de tres vías).
   * **Uso Recomendado:** Almacenar una amplia variedad de datos, desde recursos compartidos hasta bibliotecas VHD. Con el Sistema de archivos resistente (ReFS), Windows mantiene la integridad de los datos.
3. **Espacios de Paridad:**
   * **Objetivo:** Aumentar la eficiencia del almacenamiento y proteger los archivos (al guardar varias copias).
   * **Uso Recomendado:** Archivación de datos y retransmisión por *streaming* de medios (música, vídeos).
   * **Requisito Mínimo:**
     + Al menos tres unidades para proteger contra el error de una unidad.
     + Al menos siete unidades para proteger contra el error de dos unidades.

# Clase 20: Raid 2: RAID Avanzados

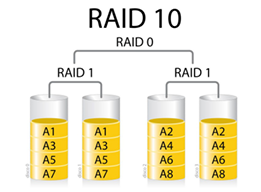
## Análisis de Niveles de RAID Avanzados

| **Nivel** | **Base de Funcionamiento** | **Características Clave** | **Tolerancia a Fallas** |
| --- | --- | --- | --- |
| **RAID 2** | Redundancia por método de Hamming. | Bandas a nivel bit. Acceso paralelo y sincronización de discos. |  |
|  | | | |
| **RAID 3** | Redundancia por paridad. | Bandas a nivel bit. Acceso paralelo y sincronización de discos. |  |
|  | | | |
| **RAID 6** | Extensión de RAID 5. | Cada serie de N bloques tiene dos bloques de redundancia, distribuidos en N+2 discos. Agrega seguridad. Las operaciones de escritura son más lentas (debe escribir un bloque de datos y dos de redundancia). | Pueden fallar hasta dos discos (de N+2) sin comprometer la disponibilidad de los datos. |
|  | | | |

## RAID Anidados y Variantes

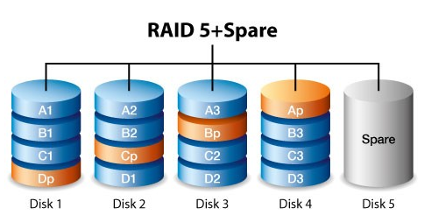
### RAID 10 (RAID 1+0)

* **Definición:** Es una combinación de dos agrupaciones RAID: RAID 1 (espejo) y RAID 0 (bandas).
* **Estructura:** Discos configurados en pares de volúmenes RAID 1, y estos volúmenes son agrupados en una matriz RAID 0.
* **Ventaja:** Puede sobrevivir el fallo de varios discos, siempre que al menos un disco en cada par RAID 1 continúe funcionando.
* **Requisitos Mínimos:** 4 discos.
* **Tolerancia a Falla:** 1 por conjunto de replicación.



### RAID 5E y RAID 6E (Con Disco de Reserva)

* **Definición:** Son variantes de RAID 5 y RAID 6 que incluyen un disco de reserva (*spare*).
* **Tipos de Discos *Spare*:**
  + **Hot spare:** Conectados y preparados.
  + **Standby spare:** En espera.
* **Función del *Spare*:** No es parte del conjunto activo hasta que un disco falla y el conjunto se reconstruye sobre él.
* **Beneficio:** No mejoran el rendimiento, pero minimizan el tiempo de reconstrucción y las labores de administración cuando se producen fallos. Si se te rompe un disco y luego otro, por más que sea un Raid 5, te da tiempo a agregar el 2do.



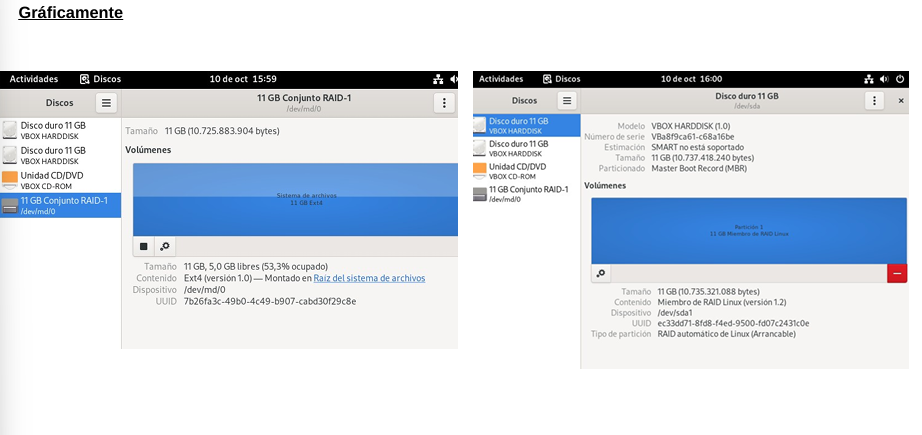
| **Nivel** | **Bandas** | **Replicación** | **Paridad** | **Discos Mínimos** |
| --- | --- | --- | --- | --- |
| **RAID 0** | X |  |  | 2 |
| **RAID 1** |  | X |  | 2 |
| **RAID 5** | X |  | X | 3 |
| **RAID 6** | X |  | X | 4 |
| **RAID 10** | X | X |  | 4 |
| **RAID 50** | X |  | X | 6 |
| **RAID 60** | X |  | X | 8 |

### En el caso de combinar RAIDs con un 0, siempre el RAID 0 queda por arriba. Por debajo en el Stripping queda el otro tipo de raid. Similar al dibujo del RAID 10.

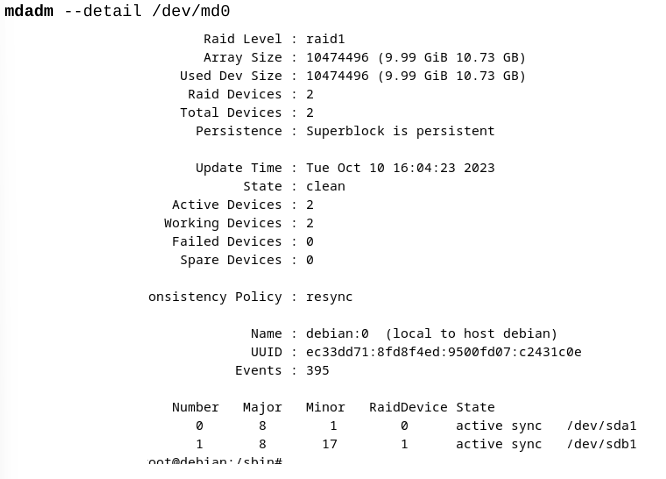
## Práctica de Laboratorio: Instalación de S.O. en RAID 1 (Debian)

La práctica consistió en instalar Debian GNU/Linux en un array RAID 1 por software utilizando dos discos virtuales de 10 GB cada uno (sda y sdb).

1. Se creó una VM en VirtualBox con 2 discos de 10 GB.
2. En el instalador de Debian, se seleccionó el método de particionado **Manual**.
3. Se crearon nuevas tablas de partición en ambos discos (sda y sdb).
4. Se creó una partición primaria en cada disco y se les asignó el tipo **RAID automático de Linux**.
5. Se seleccionó **Configurar RAID por software** y se eligió **Crear un dispositivo MD** de tipo **RAID1** con 2 dispositivos activos ($/dev/sda1$ y $/dev/sdb1$).
6. Se configuró el dispositivo RAID 1 (Dispositivo RAID0 #0, de 10.7 GB) para ser usado como **sistema de ficheros ext4 transaccional** y se le asignó el **punto de montaje /** (raíz del sistema).
7. **Nota Importante:** Es fundamental instalar el Gestor de arranque (ej: GRUB) en ambas particiones.

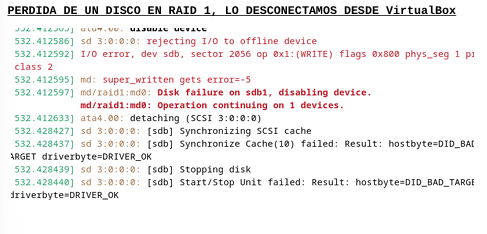


Gráficamente, el SO lo ve como un solo disco. Mientras que si tiramos el comando mdam –detail /dev/md0, nos va a tirar que discos componen dicho raid. Lo pueden ver debajo de la siguiente captura:

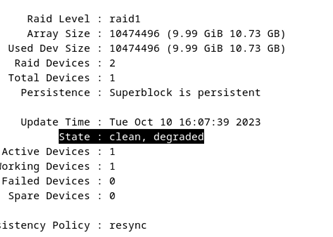


## Falla en RAID 1

* **Simulación de Falla (Desconexión de un disco):**
  + Al desconectar un disco (ej. sdb), el sistema genera errores de I/O y el kernel reporta Disk failure on sdb1, disabling device.
  + El array continúa la operación en 1 dispositivo.



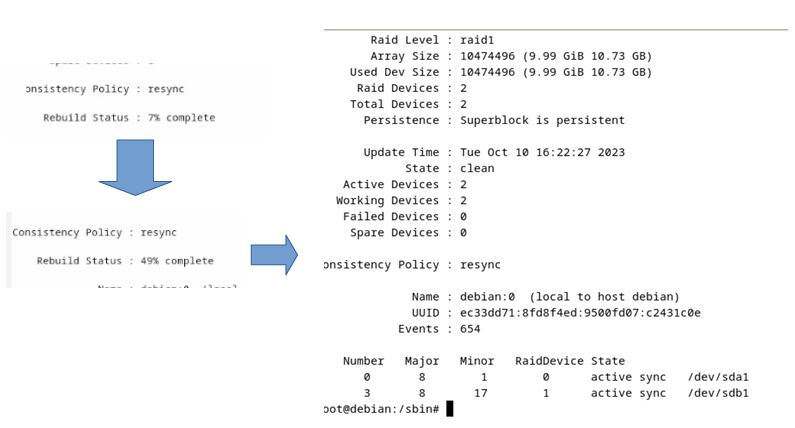
* + El mdadm --detail muestra el **Estado:** clean, degraded (limpio, degradado), con un total de 1 dispositivo y el otro como removed (eliminado).



* **Recuperación del Array RAID 1:**
  + Se conecta un nuevo disco de reemplazo (o el mismo disco tras la reconexión).
  + Se añade el disco al array con el comando: mdadm /dev/md0 --add /dev/sdb1.



* + El sistema inicia la reconstrucción del RAID 1, volviendo al estado clean y active sync en ambos dispositivos una vez completado el proceso.

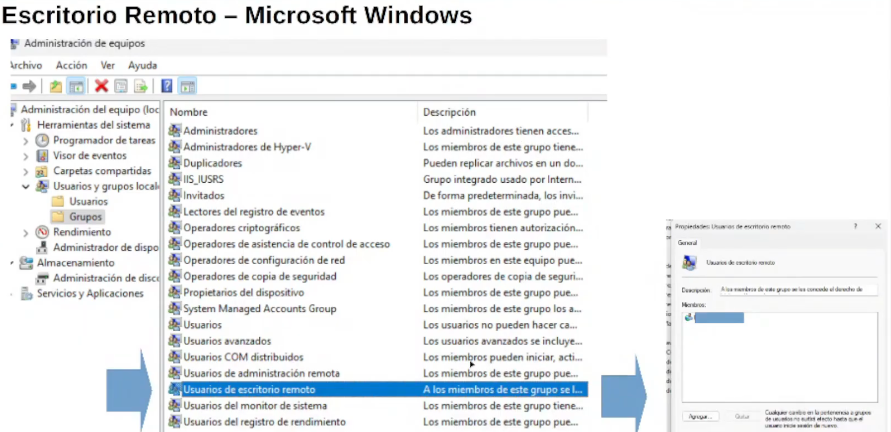


# Clase 21: Administración Remota

## Herramientas para acceder remotamente

SSH, TeamViewer, Escritorio Remoto (En windows), rlogin, lsh, nsh, slush, etc.

## Escritorio Remoto en Windows



Únicamente se puede usar el escritorio remoto con usuarios que pertenezcan al grupo de “Usuarios de Escritorio Remoto”.

En Linux puede haber n usuarios simultáneos, en Windows es multiusuario también. Pero en un Windows hogareño, pese a ser multiusuario, la interfaz gráfica solo la puede usar 1. Mientras que en Windows Server, varios usuarios a la vez pueden tener la GUI abierta.

## SSH vs Telnet

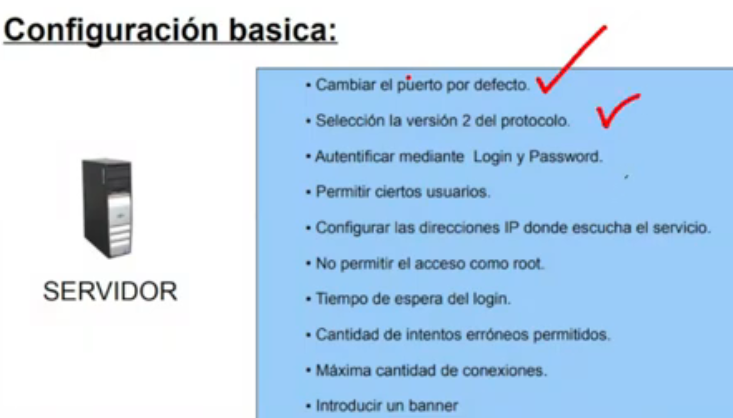
Telnet era la herramienta vieja para administrar remotamente, pero se dejó de usar porque las credenciales viajaban en texto plano, mientras que SSH cifra.

## SSH

Tiene 2 lados, SSH Server y SSH Client.

Existen clientes para todo tipo sistemas.

Por defecto SSH se abre en el puerto 22 (22000), aunque es recomendable moverlo a un puerto más alto, ya que en caso de abrir la conexión a internet, hay bots constantemente escaneando para atacar dicho puerto.

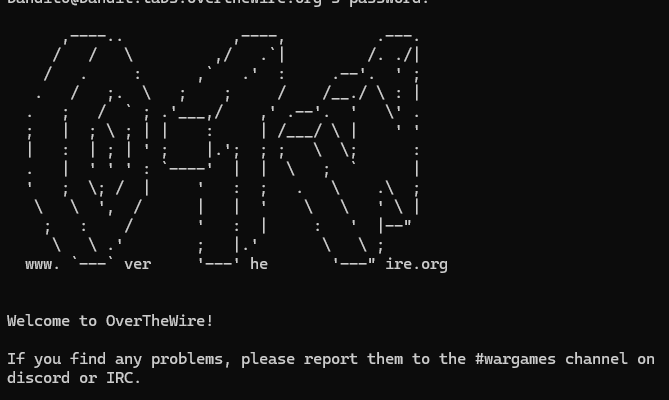


Un banner es el mensajito que te sale al establecer una conexión, como esto:

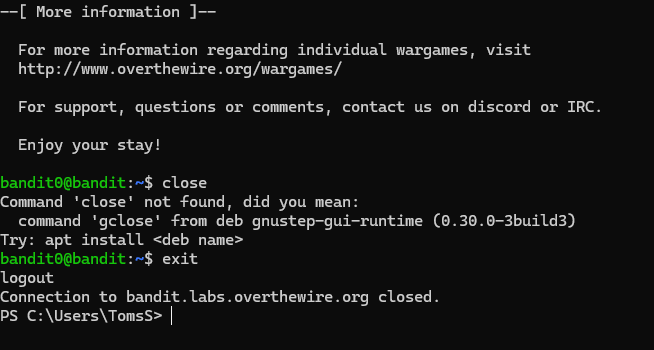
*Banner al iniciar sesion*



*Banner después de iniciar sesión*

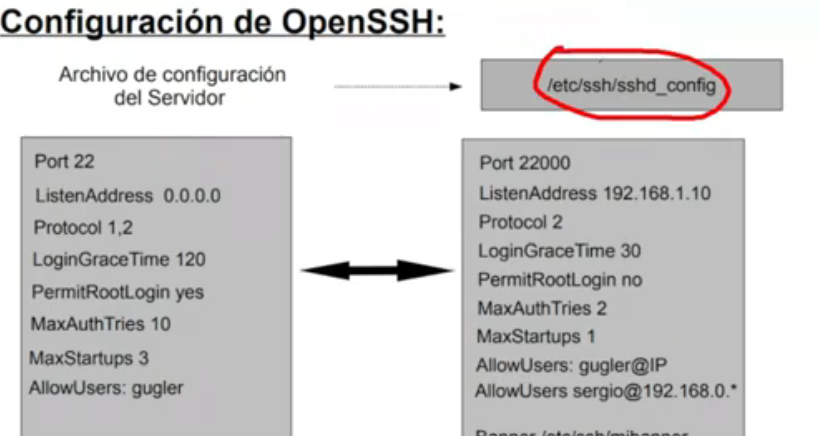


// captura mia logeado en una PC de <https://overthewire.org/wargames/bandit/>



Se puede usar para dar un aviso, tipo decir, lo que hagas se esta monitoreando.

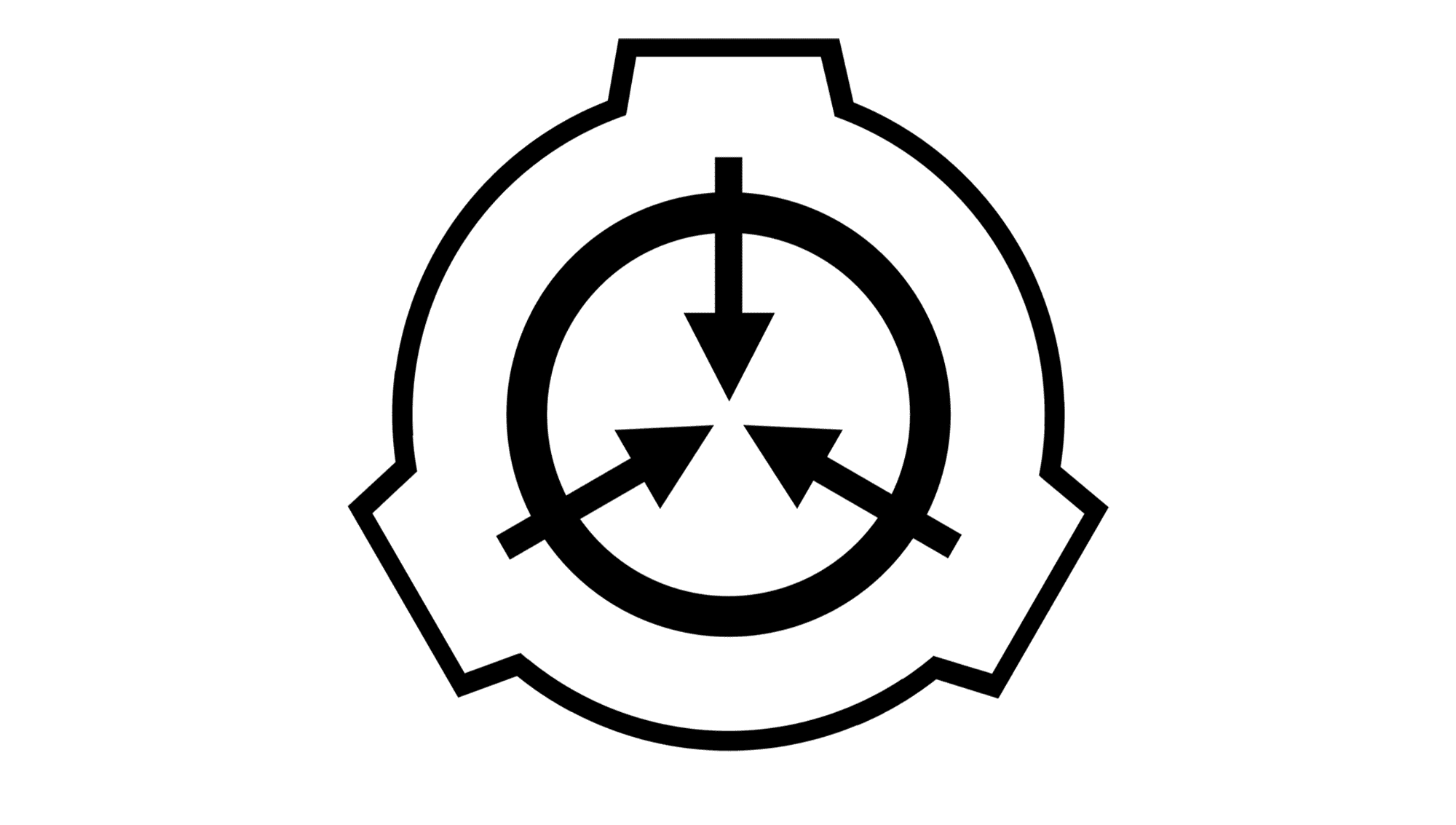
## Configuración de OpenSSH

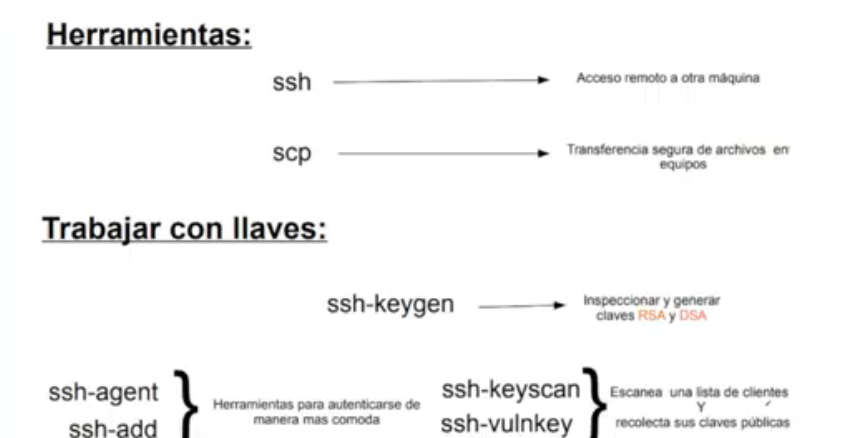


## Tools

SSH = Acceso remoto a la otra máquina

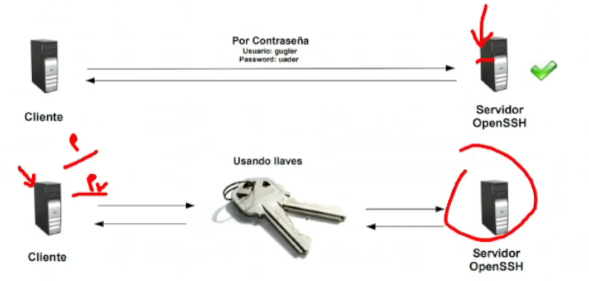
SCP = Transfiere archivos de forma segura a otro Equipo, además de proteger el mundo capturando y conteniendo objetos SCP “Skips”.



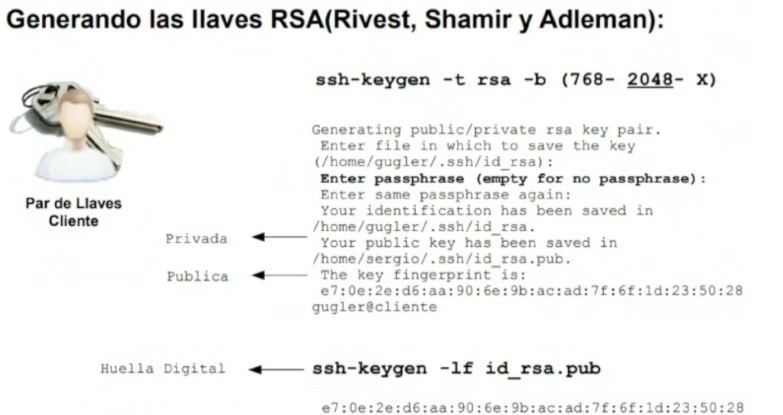


## Llaves

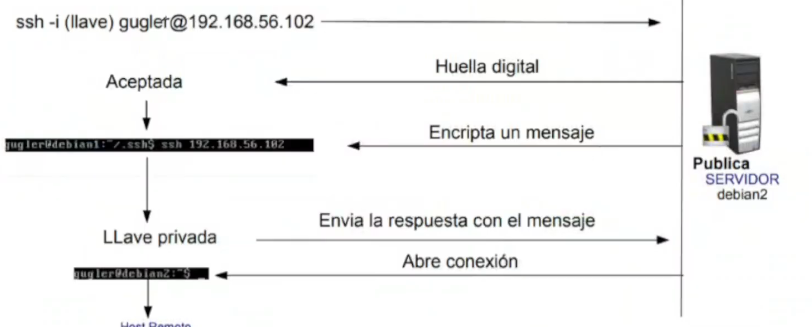
Es otra forma de autenticar usuarios, genera una llave pública y una llave privada, que funciona así: Vos te quedas con una llave pública y una privada, al querer autenticarte le mandas la pública al servidor, el servidor la va a guardar, y te va a pedir la privada. A partir de ahí, siempre que te logees va a requerirte la llave privada, que la saca automáticamente parece ser, y con eso te autenticás. Se pueden combinar los métodos de inicio de sesión.



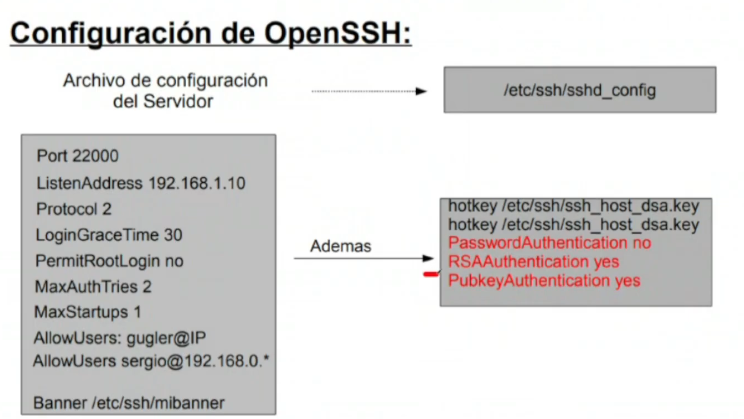
## Crear llaves



Tambien se puede usar el cifrado DSA. Logearse teniendo distintas llaves:



Si tenes una sola no hace falta poner el comando así. Configuración para tener habilitar llaves:



# Exposición: Escritorio Remoto en Windows

Protocolo: RDP (Remote Desktop Protocol) de Microsoft. Utiliza el puerto TCP 3389 por default.

## Arquitectura

Tiene un Modelo cliente – Servidor.

El Servidor escucha peticiones y comparte su escritorio, mientras que el cliente inicia la conexión para controlar al servidor.

El servidor envía imagen de su pantalla, el cliente envía acciones de mouse y teclado.

## Requisitos

Para servidor:

* Windows 10/11 Pro, Enterprise, Education, y Windows Server. (No funca en Home).

Para cliente:

* Cualquier edición de Windows. También disponible para macOS, iOS, Android y Linux.

## Seguridad

* Verifica al usuario ANTES de establecer la sesión completa.
* Hay que pertenecer al grupo mencionado antes.
* Solo los Administradores pueden conectarse por defecto.
* Toda comunicación cliente-servidor viaja cifrada.

## Cliente (mtsc.exe)

Solo requiere la IP o el nombre del equipo para conectarse.

Opciones avanzadas:

* Pantalla: Permite configurar la resolución y usar múltiples monitores.
* Recursos locales: Compartir impresoras, portapapeles, y unidades de disco para transferir archivos.
* Rendimiento: Ajustar la calidad gráfica según la velocidad de la red.

## Limitaciones en sesiones concurrentes.

Windows hogareño:

* Una única sesión activa.
* Al conectarse, la sesión física del PC se bloquea.
* Al iniciar sesión físicamente, la remota se desconecta.

Es decir, un único usuario como hablamos antes.

Windows Server:

* Diseñado para múltiples usuarios simultáneos usando Servicios de Escritorio Remoto (RDS) y licencias CAL.

## Ventajas, Desventajas, Alternativas

* Nativo de Windows
* Centralización de datos y software
* Cliente puede ser un equipo de bajos recursos
* Dependencia total de una conexión de red estable.
* Rendimiento deficiente en apps graficas intensivas.

Alternativas Populares: TeamViewer, AnyDesk, Chrome Remote Desktop, VNC.