Alberto188
www.wuolah.com/student/Alberto188
192

# resument 7-10.pdf

*PAS-ResumenTeoria*

**2° Programación y Administración de Sistemas**

**Grado en Ingeniería Informática**

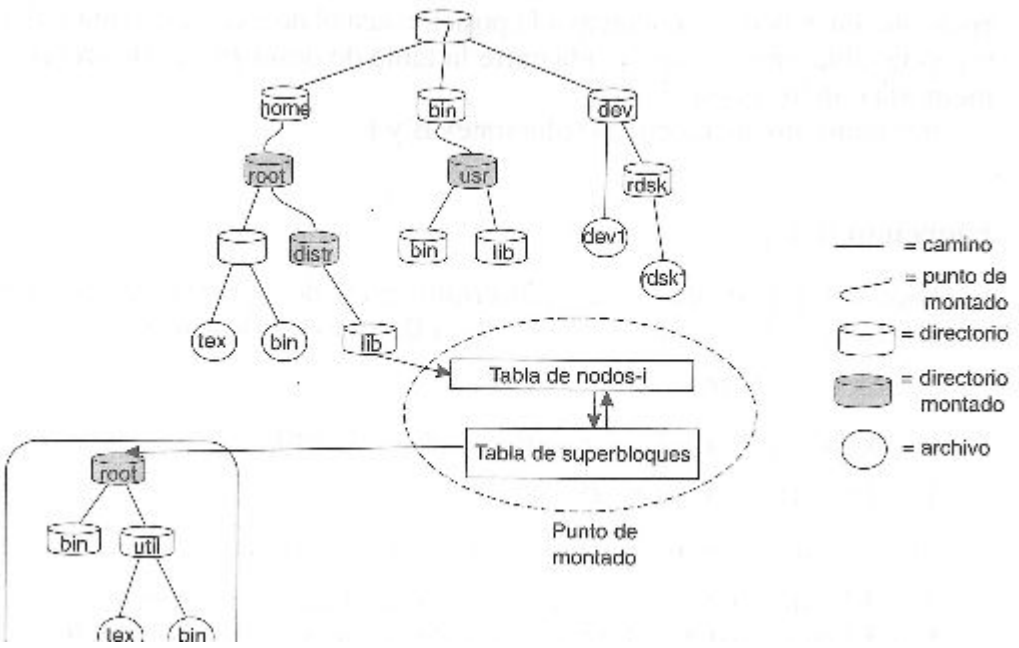
**Escuela Politécnica Superior de Córdoba**
**UCO - Universidad de Córdoba**

**Tema 7: Administración de sistemas de ficheros y discos.**

- Introducción: En GNU/Linux, hay un único sistema de ficheros lógico (o una única jerarquía de ficheros), en ella se organizan todos los dispositivos de almacenamiento disponibles. Cada partición tiene su propio sistema de ficheros, con su directorio raíz y su jerarquía.

1. Montar un sistema de ficheros: añadirlo al sistema de ficheros lógico. Sus datos están disponibles a partir de un punto de montaje.
2. Desmontar un sistema de ficheros: el sistema de ficheros deja de estar disponible dejándolo además consistente.

Los ficheros principales del Sistema Operativo están disponibles desde la raíz del sistema de ficheros lógicos(/). En el arranque del sistema, se monta primero la partición correspondiente a dicha raíz (root) y luego cualquier partición auxiliar. Ejemplo de esquema de sistema de ficheros:



escuela de negocios
CÁMARA DE SEVILLA

# diferénciate

## Con la mejor formación práctica

## www.mastersevilla.com

Titulación de prestigio
en el sector empresarial

## MÁSTER EN DIRECCIÓN Y GESTIÓN DE RECURSOS HUMANOS

-. Comandos para el manejo del sistema de ficheros:

1. mount [opci] <FicheroEspecialBloque> <PtoMontaje>

    a. -t tipo-sf: tipo de sistema de ficheros.
    b. -r: montaje de sólo lectura.
    c. -w: montaje en modo lectura/escritura.
    d. -o opcionesMontaje: Opciones del proceso de montaje (nosuid,exec,remount,etc.).

2. umount <PtoMontaje> (o <FicheroEspecialBloque>): desmontar un sistema de ficheros. Si está siendo utilizado (busy), no se podrá desmontar.

3. fuser: saber qué ficheros se están usando y qué procesos los usan (f: fichero abierto, c: directorio de trabajo, e: ejecutando un fichero ,etc.).

4. lsof: obtener un listado de todos los ficheros abiertos.

```
pedroa@pagutierrezLaptop:~$ fuser -mv / # -m: ficheros montados; -v: verbose
                     USER        PID ACCESS COMMAND
/:                   root     kernel mount /
                     pedroa      2363 Frce. gnome-keyring-d
                     pedroa      2760 Fr.e. icedove-bin
                     pedroa      3206 Fr.e. evince
pedroa@pagutierrezLaptop:~$ lsof
COMMAND  PID  USER    FD TYPE  DEVICE SIZE/OFF     NODE NAME
...
kile     2764 pedroa  mem REG     8,4   499320  5246682 libgcrypt.so.11.6.0
kile     2764 pedroa  mem REG     8,4   659656  5775741 libgnutls.so.26.14.12
kile     2764 pedroa  mem REG     8,4    68416  5775700 libavahi-client.so.3.2.9
kile     2764 pedroa  mem REG     8,4    47448  5775702 libavahi-common.so.3.5.3
kile     2764 pedroa  mem REG     8,4    14480  5775706 libavahi-glib.so.1.0.2
kile     2764 pedroa  mem REG     8,4     4016  5770905 libcanberra.so.0.2.5
kile     2764 pedroa  mem REG     8,4    31304  5771046 libgailutil.so.18.0.1
...
lsof     3271 pedroa  cwd DIR     8,4     4096  6301896 /home/pagutierrez
lsof     3271 pedroa  rtd DIR     8,4     4096        2 /
lsof     3271 pedroa  txt REG     8,4   131312  5767875 lsof
lsof     3271 pedroa  mem REG     8,4    68352  5774521 locale-archive
lsof     3271 pedroa  mem REG     8,4   642216  5248383 libc-2.13.so
lsof     3271 pedroa  mem REG     8,4   141088  5242884 ld-2.13.so
```

- Uso del fichero /etc/fstab: fichero con información sobre todos los sistemas de ficheros a montar y las zonas de intercambios a activar. El fichero tiene el siguiente formato:

fi_especial pto tipo opciones dump_freq pass_num

Significado de los apartados:

1. fi_ especial: fichero especial de bloques (/dev/…).
2. pto: directorio que sirve de punto de montaje.
3. tipo: tipo del sistema de ficheros (ext2,vfat…).
4. Opciones para el proceso de montaje, se separan por "," y sin espacios.
5. dum_freq: "frecuencia del dump" para hacer una copia de seguridad de ese Sistema de Ficheros mediante el comando dump.
6. pass_num: en tiempo de arranque, en qué orden hay que chequear los Sistemas de Ficheros, se ejecuta el comando fsck para comprobar su estado.

-. Opciones del fichero /etc/fstab:

1. rw: Lectura- escritura, es la opción por defecto.
2. ro: Sólo lectura.
3. suid/ nosuid: Permitido (o no) que los bits suid o sgid tenga efecto.
4. auto/ noauto: Montar automáticamente (o no) (ejecutando mount -a se montan todos los automáticos, siempre se ejecuta al arrancar el sistema)
5. exec/ noexec, permitir o no la ejecución de ficheros.
6. usrquota, grquota: activar cuotas.
7. uid=500, gid=100: Propietario y grupo propietarios de los ficheros del Sistema de Ficheros (si el sistema de ficheros no incorpora esta información o si se quiere cambiar).
8. umask=137: Permiso de los ficheros.
9. dev: Interpretar ficheros especiales en el sistema de ficheros.
10. sync: Forzar a que todas las operaciones sean síncronas, esto puede repercutir en la vida de la unidad el disco) .
11. user: permite que los usuarios puedan montar el sistema de ficheros. Solo el mismo usuario podrá desmontarlo. Implica las opciones noexec, nosuid y nodev.
12. users: igual que user pero cualquiera puede desmontarlo.
13. nouser: Solo el root podrá desmontar el SF.
14. owner: permite que un usuario pueda montar el sistema de ficheros, siempre que sea dueño del fichero de dispositivo. Implica las opciones nosuid y nodev.

Contenido del fichero /etc/fstab:

```
1 LABEL=/     /             ext3     defaults,usrquota  0 1
2 /dev/sda3   /windows      vfat     defaults           0 0
3 /dev/dvd    /media/dvd    iso9660  noauto,owner,ro    0 0
4 /dev/fd0    /media/floppy vfat     noauto,uid=500     0 0
5 /dev/sda4   /otrolinux    ext3     defaults           0 2
6 /dev/sda2   swap          swap     defaults           0 0
```

- Comprobación del sistema de ficheros: durante su arranque, los comandos fsck o e2fsk ,chequearon la consistencia o estado del sistema de ficheros, detectando problemas e intentando repararlos. Si se dan problemas en el arranque se obliga al administrador arreglar los errores de manera manual. El SF raíz está montado en modo lectura, y para chequear el SF debe estar montado o desmontado en modo sólo de lectura. Se actúa sobre la estructura:

1. Bloques pertenecientes a varios ficheros
2. Bloques que están marcados como libres, pero no se encuentran en uso y viceversa.
3. Inconsistencias en cuanto al número de enlaces hacia un nodo-i.
4. Nodos-i marcados como libres, pero no están en uso y viceversa.

-. Journaling: para evitar la verificación completa (fsck) de sistemas de ficheros de gran tamaño, se implementa un modelo de control de transaccional basado en logging. Características de este método:

1. Las suboperaciones que modifiquen los metadatos y datos de un archivo se agrupa en la misma transacción.
2. Se el sistema falla, las acciones parcialmente realizadas deshacen o completan, recorriendo el log. Es decir, después de una caída se completan las transacciones commited y se descartan el resto.
3. No se garantiza que el sistema esté actualizado al finalizar la recuperación, sino que es consistente.

Por cada Sub-operación que altera las estructuras de disco se escribe un registro en el log, que incluye las modificaciones en los buffers de i-nodos y de bloques.

- Añadir un sistema de ficheros nuevo o un nuevo disco:

1. Se realiza la conexión física.
2. Crear un fichero especial de dispositivo (si fuese necesario).
3. Crear las particiones: fdisk o parted.
4. Crear sistema de ficheros: mk2fs -t ext2 /dev/sdb3.
5. Etiquetar la partición usando e21abe1: asigna una etiqueta al SF que se puede usar en el fichero /etc/fstab, en el capo fi_especial, mediante LABEL=etiqueta.
6. Crear el directorio que hará el punto de montaje.
7. Montar el nuevo sistema de ficheros.
8. Actualizar /etc/fstab con las opciones necesarias.

-. ¿Qué sistema elegir?:

1. ext2: muy rápido en general, pero no tiene journaling. Se puede usar en un SF en el que se guardaran ficheros temporales.

2. ext3: buen rendimiento en general y journaling.

3. ext4: menor uso del CPU y mayor rapidez en los procesos de lectura y escritura que ext3. Estándar de facto en Linux.

-. Comando para conocer y ajustar parámetros de un SF (ext4/ext3/ext2), es tune2fs, y sus opciones son:

1. -l dispositivo: Listar el contenido del superbloque del SF.
2. -c max-mount-counts dispositivo: establecer el número de montajes máximo sin realizar un fsck.
3. -i numero [d|m|w] dispositivo: Indicar el tiempo máximo entre dos chequeos.
4. -L etiqueta dispositivo: Poner una etiqueta al sistema de ficheros.
5. -m porcentaje dispositivo: Fijar el porcentaje de bloques reservados para procesos especiales de root. Por defecto, 5%.

```
pedroa@pedroa-laptop ~ $ sudo tune2fs -l /dev/sda2
tune2fs 1.42.8 (20-Jun-2017)
Filesystem volume name:   ROOT
Last mounted on:          /
Filesystem UUID:          d73f541a-e887-4885-b42b-98dd433e99de
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype
    needs_recovery extent flex_bg sparse_super large_file huge_file uninit_bg dir_nlink
    extra_isize
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              3145728
Block count:              12582912   Reserved block count:     629142
Free blocks:              7726646
Free inodes:              2545229
First block:              0
Block size:               4096
Fragment size:            4096
Reserved GDT blocks:      1021
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         8192
Inode blocks per group:   512
RAID stride:              32710
Flex block group size:    16
Filesystem created:       Thu Mar 27 20:07:38 2016
Last mount time:          Sun Mar 22 08:50:58 2016
Last write time:          Fri Aug 29 21:26:46 2016
Mount count:              391
Maximum mount count:      -1
Last checked:             Fri Aug 29 21:26:46 2016
Check interval:           0 (<none>)
Lifetime writes:          922 GB
Reserved blocks uid:      0 (user root) Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:               256
Required extra isize:     28
Desired extra isize:      28
Journal inode:            8
First orphan inode:       813409
Default directory hash:   half_md4
Directory Hash Seed:      c5d4f69f-cea8-4574-894b-166152ae5a40
Journal backup:           inode blocks
```

- Cuotas: Permiten limitar el número de bloques y/o ficheros (nodo-i) que un usuario puede usar en una partición (también se pueden establecer para grupos de usuarios). Hay dos tipos de límites:

1. Límite hard: el usuario no puede sobre pasarlo. Si lo hace, ya no podrá usar más bloques o crear más ficheros.
2. Límite soft: es interior al límite hard y se puede sobrepasar durante cierto tiempo, siempre que no se alcance el límite hard.

-. Período de gracia: tiempo durante el que se puede sobrepasar el límite soft. Se informa al usuario de que ha superado el límite y que debe liberar espacios o nodos-i (ficheros). Estos períodos y límites se establecen, de forma independiente, para bloques y nodos-i. Pasos para realizar las cuotas del disco:

1. Instalar el paquete quota.
2. Indicarlo en fstab (diferente en ext3 y ext4).
3. Remontar la partición para que se activen las opciones: mount -o remount /home.
4. quotachek -avugm; añade el contenido de los ficheros de control de cuotas.
    a. a: todos los dispositivos con cuotas.
    b. v: verbose, Método de presentación donde se relaciona mayor información y más detallada que en un informe normal.
    c. u: cuotas para usuarios.
    d. g: cuotas para grupos.
    e. m: no remontar los archivos en modo solo lectura.
5. Activar las cuotas: quotaon -avug.
6. Desactivarlas: quotaoff -avug.
7. Editar la cuota del usuario pagutierrez: edquota pagutierrez.
8. Establecer el periodo de gracia: edquota -t.
9. Copiar cuotas: edquota -up pagutierrez jsanchez.
10. Estadísticas de las cuotas: repquota /dev/sdb1.

-. RAID: Array redundante de discos independientes.Es importante que se puede implementar por software o por hardware, además varias unidades de disco se ven como una sola unidad lógica. Logical Volume Management (LVM), agrupar las particiones en volúmenes y permite redimensionar las particiones. Niveles de las RAID:

a) RAID nivel 0:

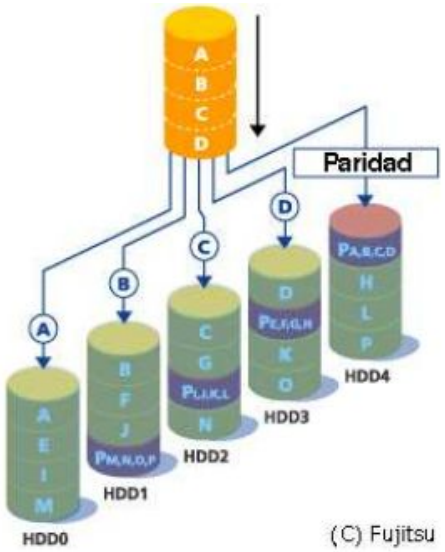
    i) Expande la información en diversos discos, que se ven como un único SF.
    ii) Aumenta el espacio según el número de discos usado.
    iii) Se consigue E/S paralela en lecturas y escrituras, siempre que los bloques a tratar no sean del mismo disco.
    iv) No hay redundancia de datos.

b) RAID nivel 1:

    i) Se utilizan dos o más discos duros, que forman un único SF (El SF está replicado en varios discos).
    ii) Son discos espejos, ya que todos guardan la misma información. Por tanto hay redundancia de datos.
    iii) Las lecturas pueden ser en paralelo, las escrituras no.
    iv) Cuando unos de los discos falla, el sistema sigue trabajando con el otro sin problemas.
    v) La recuperación de un disco transparente al usuario.

c) RAID nivel ⅘:

i) División de los datos a nivel de bloques.
ii) RAID 4: Mínimo 3 discos duros, de los cuales 1 almacenará la paridad de los otros discos, que son usados para datos.
iii) Problema: el disco con paridad es un cuello de botella. RAID 5: repartir paridad entre todos los discos y ofrece la mejor relación rendimiento-coste es un entorno con varias unidades.+
iv) Se consigue un dispositivo de almacenamiento más grande.
v) Hay redundancia de datos.
vi) La lectura y escritura es en paralelo.




-. Paridad: Cada vez que se escriben datos, se calcula el XOR bit a bit (1 número de unos impar, 0: número de unos par) de los bloques implicados en cada disco. Ejemplos:

Ejemplo: paridad

| | | |
|---|---|---|
| Disco 1: | 00101010 | (Datos) |
| Disco 2: | 10001110 | (Datos) |
| Disco 3: | 11110111 | (Datos) |
| Disco 4: | 10110101 | (Datos) |
| Disco 5: | 11100110 | (Paridad) |

| | | |
|---|---|---|
| Disco 1: | 00101010 | (Datos) |
| Disco 2: | 10001110 | (Datos) |
| Disco 3: | 11110111 | (Datos) |
| Disco 5: | 11100110 | (Paridad) |
| Disco 4: | 10110101 | (Datos) |

Ejemplo: En caso de que uno de los discos falla, el contenido se puede restaurar a partir de la paridad.

- Control de dispositivos de entrada/salida: La herramienta mdadm permite crear o administrar un dispositivo RAID, convertir un disco “normal” en un RAID… Tiene distintos modos de funcionamiento: create nos ayuda a configurar y activar los sistemas RAID. En el directorio /proc/mdstat se lista todos los sistemas RAID (dispositivos md) activos con información sobre su estado. Las particiones que forman el RAID tienen que un flag RAID (Linux raid auto), de esta manera serán detectadas y activadas en el proceso de arranque.

- Creación de un RAID1 con un disco que ya tiene datos:
  - Crear el RAID con la partición que tiene los datos:

```
mdadm --create /dev/md2 --force --level=1 --raid-devices=1 /dev/sda4
```

  - Añadir nuevo disco al RAID como disco de repuesto (`spare`):

```
mdadm /dev/md2 -a /dev/sdc3
```

  - Activar el nuevo disco: `mdadm --grow /dev/md2 -n 2`.
  - A continuación, introducirlo en `/etc/fstab`.
- Información sobre el estado:

```
mdadm --detail --scan /dev/md1
```

- Todo esto se puede configurar utilizando el fichero `/etc/mdadm.conf`.

**Tema 8: Instalación de Impresoras**

- Introducción: Las impresoras son mucho más complicadas que otros periféricos. Disponen de un SO propio, que recoge los trabajos y los imprime en papel. Reconocen formatos específicos y algunos son accesibles desde la red. En linux CUPS (Common Unix Printing System), permite realizar mucho más fáciles las tareas de administración de impresoras.

- Lenguaje de las impresoras: un trabajo de impresión puede verse como un programa escrito en un lenguaje que la impresora entiende. Un lenguaje de impresión es Page Description Languages (PDLs), describen cómo representar una página en el papel utilizando el cartucho de tinta, se usa para ello un formato vectorial. Es más rápido y fácil que transmitir la imagen en crudo, es independiente del dispositivo y de la resolución. Se podría pasar de un fichero PDL a mapa de bits: rasterizar, para ello se usan programas que hacen Raster image processing (RIP).

- Organización: Cada equipo puede gestionar muchas impresoras a la vez, cada impresora posee su propia cola de impresión en la que guardar y secuenciar los trabajos. Administración lanza órdenes para añadir impresoras, gestionar las tareas de impresión, etc.

-. Directorios de spool: son los usados por las colas de impresión. Guarda un fichero con las propiedades del trabajo de impresión, y guarda los trabajos pendientes para imprimir. Están contenidos en /var/spool.

-. Impresión en un cliente servidor: en el servidor se abre un proceso, es un demonio que realiza la impresión. Se usa un filtro de impresión, el programa modifica el fichero a imprimir, transformándolo al PDL de la impresora. Para convertirse en servidor de impresión, se tiene que dar los permisos oportunos para que la impresora pueda ser usada de forma remota.

- Elementos de CUPS, este lenguaje se basa en el protocolo HTTP:

1. Operaciones POST para imprimir y GET para ver el estado.
2. Los ficheros de configuración son muy parecidos a los de Apache.
3. Las conexiones se realizan en el puerto 631.
4. CUPS es una evolución del IPP o Internet Printing Protocol.

CUPS, sabe manejar la impresora gracias a los ficheros PPD o Postscript Printer Description (Contiene las opciones soportadas por la impresora y el lenguaje que entiende de forma nativa), junto a los filtros (Cadenas de conversores, basados en los tipo MIME, Los tipo MIME son los que describen el tipo de medio del contenido).

Foomatic es una base de datos instalable en cualquier sistemas que integra controladores de impresoras con los "spoolers" habituales en UNIX: CUPS, LPRng…

- Comandos usados en la impresión:

1. Imprimir un fichero:
    a. lp [-d impresora] fichero1 [fichero2] (System V).
    b. lpr [-P impresora] fichero1 [fichero2] (Berkeley).

2. Eliminar un trabajo de la cola de impresión:
    a. cancel id_tra1 [id_tra2] [impresora] (System V).
    b. lprm [-P impresora] id_tra1 [id_tra2] (Berkeley).

3. Eliminar una impresora (o clase):
    a. lpadmin -x impresora.

4. Consultar la cola de impresión:
    a. lpq -P impresora: listado de la cola impresión y del estado de los trabajos.

5. Crear clases de impresoras:

```
pagutierrez@Laptop:~$ lpadmin -p HP-Color -c ClasePrueba
pagutierrez@Laptop:~$ lpadmin -p Ricoh -c ClasePrueba
```

6. Crear una instancia de impresoras con opciones concretas:

```
pagutierrez@Laptop:~$ lpoptions -p HP-Color/2up -o number-up=2
pagutierrez@Laptop:~$ lpr -P HP-Color/2up tmp.ps
```

7. Añadir una impresora:

```
# -E: habilita impresora; -v URI; -m fichero.ppd
pagutierrez@PEDROLaptop:~$ lpadmin -p groucho -E -v parallel:/dev/lp0 -m pxlcolor
    .ppd
pagutierrez@PEDROLaptop:~$ lpadmin -p fezmo -E -v socket://192.168.0.12 -m
    laserjet.ppd
```

-. Habilitar/ deshabilitar impresoras:

1. cupsdisable impresora: deshabilita la impresora (se aceptan trabajos en cola, pero no los imprime).
2. cupsenable impresora: Iniciar de nuevo la impresora (imprimirá los trabajos pendientes y los que reciba nuevos).
3. cupsreject impresora: deshabilita la cola de impresión (no aceptará nuevos trabajos).
4. cupsaccept impresora: habilitará la cola de impresión (que aceptará nuevos trabajos).

El demonio de impresión es cupsd, es necesario para imprimir.

a) /etc/init.d/cups, contiene el script para lanzar al demonio

Al añadir una nueva impresora, o realizar cambios de configuración, hay que reiniciar al demonio.

-. Ficheros de configuración:

1. /etc/cups/classes.conf: información de las clases.
2. /etc/cups/cupsd.conf: configuración del demonio.
3. /etc/cups/printers.conf: información impresoras.
4. /etc/cups/ppd: Ficheros de filtro para cada impresora.
5. /var/spool/cups: directorio spool.

Término Browsing: los equipos clientes localizan y usan las impresora del servidor de impresión, sin necesidad de instalarla previamente.

**Tema 9: Copias de seguridad y restauración**

- Planes de prevención de catástrofes: En cualquier momento, algunos archivos serán totalmente ilegible por algún motivo, en consecuencia se exige capacidad de recuperación. El administrador debe planear e implementar un sistema de copias de seguridad, periódicamente hacer copias de seguridad de los ficheros y guardar las copias de seguridad en un lugar seguro. La estrategia de copias de seguridad tiene que ser efectiva, para conseguir seguridad:

1. El tiempo empleado es un esfuerzo que prevé futuras pérdidas.
2. El dinero se compensa al evitar el desastre que supone una pérdida de datos, que conlleva enormes pérdidas de trabajo y, por tanto, dinero.

-. Causas de la pérdida de información:

1. Errores de usuario.
2. Virus y software destructivo.
3. Personas malintencionadas.
4. Fallos mecánicos.
5. Fuerzas mayores: desastres naturales, electricidad estática, …

Si valoramos los costes, merece la pena incluir mecanismos/dispositivos específicos para esta labor.

También se producen errores humanos:

1. Comandos mal escritos.
2. Errores durante el redireccionamiento y uso de tuberías.

Si el usuario cuenta con acceso root, serían catastróficos que ocurrieran sobre directorios o archivos de sistema.

-. Prevención de errores humanos:

1. Uso de alias.
2. Uso de un sistema de control de versiones, conservan el archivo original y llevan un histórico de los cambios realizados sobre éste.
3. Crear copias de seguridad personales.
4. Utilizar sudo para limitar el acceso a los usuarios con privilegios de root. Se limitará el acceso únicamente a los comandos necesarios para que el usuario pueda llevar a cabo su tarea.

- Virus y software destructivo, un virus es un programa que se adhiere a un ejecutable y se propaga a otros al mismo tiempo que realiza otra acción (desde escribir un mensaje a mezclar las tablas de particiones). Tipos:

    1. Caballos de Troya: Programas que se hacen pasar por otros, funcionando como éstos, pero además realizando otras operaciones como obtener y enviar contraseñas. El grado de destrucción depende de quien los ejecute.

    2. Gusanos: Programas que se aprovechan de las debilidades de un sistema para propagarse a otros.

    3. Software destructivo: Aplicaciones no mal intencionadas pero con errores de programación que pueden ser dañinos.

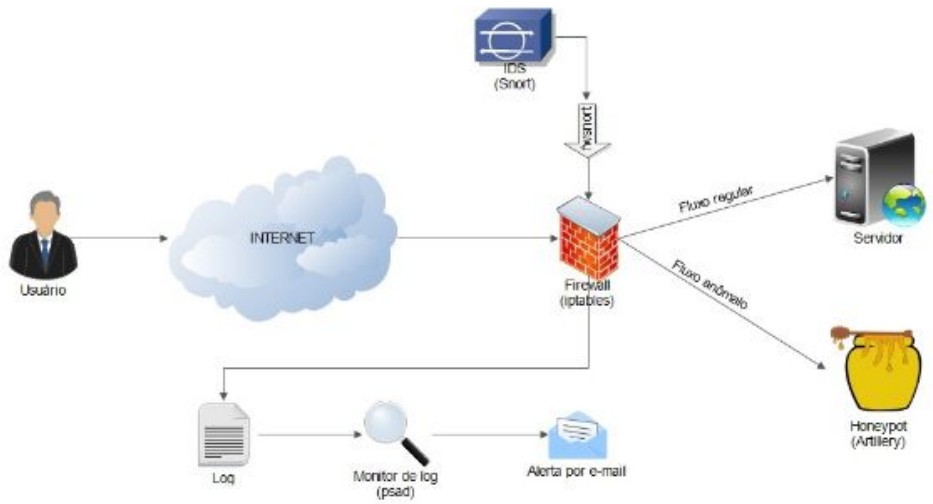
En Linux al usuario no se le dota del control total del sistema, por lo que se dificulta la propagación.

-. Medidas de prevención sencillas:

1. Software específico de búsqueda y destrucción de virus.

2. Configuración del entorno, para que el PATH no incluya la carpeta actual o donde se ejecuto el archivo.

3. Host víctimas:
    a. Se usan ciertos equipos para probar el software nuevo, asumiendo que puede causar daños (honeypots).
    b. Se suelen basar en un sistema de detección de intrusos (IDS) que genera reglas para el firewall o antivirus, separando el tráfico normal del anómalo.



-. Personas malintencionadas:

1. Crackers (!= Hacker): Son personas que entran en los sistemas de forma, a veces, ilegal con fines malintencionados.

2. Usuarios descontentos: usuario con acceso al sistema y recelo.

Medidas preventivas: Cortafuegos y seguridad física para los crackers. Seguimiento de personas sospechosas de ser “usuarios descontentos” controlando sus accesos y sus privilegios.

-. Fallos de hardware:

1. Fallo en la unidad del disco duro, aunque el Kernel suele avisar de un fallo completo.
2. Fallo en la memoria, pérdida de información por la caída del sistema o información corrupta en memoria es copiada a disco.

Prevención o recuperación:

1. Para la redundancia de información, usar RAID.
2. Supervisión de registro del sistema, con la orden dmesg y datos SMART.
3. Recuperación desde copias de seguridad.
4. Intentar leer bloques para construir una imagen con ‘dd’.
5. Recuperación en entorno estéril, con una empresa dedicada.

-. Factores a considerar en una estrategia de copias de seguridad:

1. ¿Qué ficheros se deben copiar y dónde están esos ficheros?
2. Conocer qué es lo más importante del sistema.
3. ¿Quién hará la copia?
4. ¿Dónde, cuándo y bajo qué condiciones se deben hacer? (mejor hacer las copias cuando no haya nadie trabajando).
5. Frecuencia de cambios en los ficheros y por ende frecuencia de copias.
6. ¿Cada cuánto tiempo habrá que recuperar ficheros dañados y perdidos?
7. ¿Dónde se restaurarán los datos?
8. Rutinas de restauración sencillas.
9. Proteger las copias de seguridad contra escritura.
10. Seguridad de las copias (lugar donde se almacenan,etc.).



- Estrategias de copias de seguridad, tipos:

1. Copias de seguridad completas:
    a. Se guardan todos los archivos asociados en un ordenador, como causante la restauración tarda mucho tiempo, debido a que necesita de un solo fichero.
    b. Puede ser difícil recuperar un archivo suelto.
    c. Si los ficheros no cambian muy a menudo, no hay justificación para su uso. En cambio si los ficheros cambian a diario y son vitales, están justificadas incluso a diario.
    d. Es bueno implantarlas ante grandes cambios: nuevo software, nuevo SOs…

2. Copia de seguridad parcial:
    a. Se copia sólo algunos archivos específicos. Por consiguiente, el proceso de restauración es más sencillo, ya que hay menos archivos implicados.
    b. El principal problema, es que nos dejamos archivos sin copiar.

3. Copia de seguridad incremental:
    a. Solo afecta a aquellos ficheros que hayan cambiando desde la última copia. Se realiza casi a diario.
    b. Se mantiene una copia completa del sistema,y se incorporan cambios muy pequeños, de los que se irán haciendo copias incrementales. Un ejemplo de copias incrementales sería:

        i. Nivel 0 → Backup completo (copia de los datos originales o de base del ordenador).
        ii. Nivel 1 → Todos los ficheros que han cambiado desde el último backup de nivel 0.
        iii. Nivel 2 → Todos los ficheros que han cambiado desde el último backup de nivel 1.

    c. Es importante asociar una estrategia de restauración.

- Soportes para realizar las copias de seguridad, debido a que guardar la copia de seguridad en el mismo disco , o en otro disco conectado a la máquina, no es seguro. Podemos usar multitud de dispositivos como:

1. Cintas magnéticas, es un soporte de almacenamiento de datos que se graba pistas sobre una banda plástica con un material magnetizado.
2. Discos extraíbles, son discos duros que puedes extraer sin apagar la máquina.
3. CD-Roms o DVD's regrabables.
4. Disquetes
5. Librería de cintas o jukeboxes (dispositivo que apila varias unidades CD-Roms, a los que el ordenador accede en función de sus necesidades) , stackloaders (se encargan de manejar la solicitud y carga automáticamente el cartucho correcto en una unidad de cinta, ya sea secuencialmente o en otro orden específico) y similares.

-. Para elegir el soporte se tienen que tener en cuenta:

1. Coste del soporte físico y del dispositivo.
2. Soporte Kernel
3. Capacidad de almacenamiento de datos.
4. Tasa de transferencia de datos para realizar las copias.
5. Mecanismo de cargado automático de una nueva cinta.

- Comando tar (Tape ARchiver): se encarga de realizar copias de seguridad de ficheros o dispositivos. Algunas opciones son:

1. c → Crea un fichero contenedor.
2. x → Extrae ficheros de un fichero contenedor.
3. v → Modo verbose (mayor cantidad de mensajes).
4. f → Permite especificar el nombre del fichero contenedor.
5. z → Comprime o descomprime mediante gzip.
6. j → Comprime o descomprime mediante bz2.
7. p → Conserva los permisos de los ficheros.
8. g → Guarda los ficheros con su ruta absoluta.
9. N → Considera solo archivos cuya fecha sea superior al argumento.

Algunos de los usos del comando:

- `tar cPf /dev/nst0 /home` ⇒ copia todos los ficheros del directorio `/home` en la unidad de cinta.
- `tar czvf /dev/sda1 /home` ⇒ ¿qué sucede con la partición `/dev/sda1`?
- `tar czvf /dev/nst0 /dev/sda1`
- `tar czvf practicas.tgz prac_pas`
- `tar tzvf practicas.tgz` ⇒ listar el contenido de la copia de seguridad realizada en el fichero.
- `tar xzvf practicas.tgz` ⇒ descomprimir.
- `tar xzvf practicas.tgz prac_aso/boletin1.pdf` ⇒ recuperar el fichero `boletin1.pdf` (observa que hay que indicar la ruta con la que `tar` lo almacenó).

- Comando cpio: copias de seguridad de conjuntos de ficheros seleccionados arbitrariamente. Empaqueta los datos en una cinta más eficiente que tar. También lee de la entrada estándar el nombre de los ficheros a guardar, para usarlo enlazado con otras órdenes con tuberías. Algunas de sus opciones:

1. o → Copiar “fuera” se usa para crear la copia.
2. i → Copiar “dentro” descomprime.
3. m → Conserva fecha y hora de los ficheros.
4. t → Muestra la tabla de contenidos, es decir, muestra el contenido de la copia.
5. A → Añade ficheros a un contenedor existente.
6. d → Crear directorios al descomprimir.
7. v → Modo verbose
8. F → Crear una copia en un fichero.

   Ejemplos de uso del comando:

   - `find /home | cpio -o > /dev/nst0` → se copia en la unidad de cinta.
   - `find /home | cpio -o -F h.cpio` → la copia la realiza en un fichero.
   - `cpio -i < h.cpio` → restaura la copia de seguridad de ese fichero.
   - `cpio -i -F h.cpio fichero` → restaura sólo el fichero indicado.

- Comando dump: hace copias de seguridad de un sistema de ficheros Ext2, Ext3 o Ext4, copiando la partición completa. Permite realizar copias de seguridad por niveles: desde el nivel 0, copia completa, al nivel 9, que es el valor por defecto. Actúa solo a nivel de dispositivo. Algunas opciones son:

1. 0 - 9 → Nivel de copia de seguridad, no requiere argumentos.
2. -u → Actualiza /etc/dumpdates, no requiere argumento.
3. -f → Indica fichero destino diferente al usado por defecto, sí requiere argumento. Por defecto, se usa la unidad de cinta

   Un ejemplo sería:
   - `/etc/dumpdates` → información sobre las copias de seguridad de cada SF y de qué nivel son: `/dev/sda1 0 Mon Feb 14 09:56:44 2017 +0100`

- Comando restore: Restaura copias de seguridad creadas con dump, permite recuperar los ficheros, directorios y SF enteros. Para recuperar el SFse tiene que crear y montar un SF limpio y vacío, entrar en el punto de montaje y deshacer el backup.

Algunas opciones son:

1. -r → Restaura la copia completa,  no requiere argumento.
2. -f → Indica el dispositivo o archivo donde está el backup, si requiere argumento.
3. -i → Modo interactivo, no requiere argumento.
4. -x → Extrae los archivos y directorios desde el directorio actual.
5. -t → Imprime los nombres de los archivos de la copia, no requiere argumentos.

Ejemplos de dump y restore:

- `dump 0 -u -f /dev/nst0 /dev/sda1` → Copia de nivel 0 de `/dev/sda1` en la unidad de cinta, actualizando `/etc/dumpdates`.
- `dump 1 -u -f /dev/nst0 /dev/sda1` → Copia de nivel 1 de `/dev/sda1` en la unidad de cinta, actualizando `/etc/dumpdates`.
- `dump 0 -f jj.dump /dev/sda1` → Copia de nivel 0 de `/dev/sda1` en el fichero jj.dump.
- `restore -t -f fichero_backup` → listado de la copia.
- `restore -x -f fichero_backup practicas/smallsh.c` → restaura sólo el fichero `practicas/smallsh.c`.
- `restore -r -f /dev/nst0` → restaura una copia completa.
- `restore -i -f /dev/nst0` → permite restaurar ficheros interactivamente (con ls, cd, pwd, add y extract).

- Restauración del sistema, en caso de tener una copia de todo el sistema:

1. Arrancar desde un dispositivo distinto.
2. Si es necesario, crear los ficheros especiales de dispositivos para los discos.
3. Preparar el disco duro, e.d., crear las particiones.
4. Crear el sistema de ficheros en la partición donde se restaurarán los datos y montarlo en un directorio.
5. Restaurar la copia de seguridad sobre ese sistema de ficheros.
    a. Restaurar la copia más reciente de nivel 0.
    b. Restaurar la copia más reciente del nivel más bajo después del último restaurado.
    c. Si quedan más copias por restaurar, volver al paso anterior.
6. Desmontar el sistema de ficheros restaurado.
7. Volver al paso 2, para restaurar otros SF adicionales.



**Tema 10: Gestión de comunicaciones**

- Labores mínimas en la gestión de una red: Establecer las opciones de configuración de la red más importantes, entender la configuración de red actual y programar estrategias de crecimiento de la red, para que la eficiencia pueda mantenerse.

- Demonios de red:

1. xinetd → Se usa para administrar los servicios en linux. Maneja a otros demonios, los cuales inicializa cuando hay un trabajo para ellos (sshd, ftpd, pop...). En el directorio /etc encontramos el archivo xinetd.conf, que es el fichero de configuración de este demonio. En el directorio /etc/xinetd.d, es el fichero de configuración de los demonios gestionados por xinetd.
2. ntpd → Demonio encargado de sincronizar la hora del sistema.
3. dhcpd → Demonio encargado del servicio Dynamic Host Configuration Protocol o DHCP, el servidor proporciona las IPs privadas a las máquinas que se conecten.
4. named → Demonio encargado del servicio de Domain Name System o DNS, el servidor traduce nombres de dominio.
5. sendmail → Demonio encargado del correo electrónico.
6. sshd → Demonio que permite ssh, una conexión remota segura.
7. httpd → Servidor web, normalmente apache.
8. smbd → Servicio de compartición de ficheros con Windows.

- NFS: Network File System, posibilita que un SF, que físicamente reside en un host remoto, sea usado por otros ordenadores, vía red, como si fuese un sistema de ficheros local. En el servidor se indica:

1. Qué sistema de ficheros se exportan, si un SF completo o un directorio.
2. A qué ordenadores se exportan (se les permite acceder), a un equipo concreto o a todos los equipos de una red.
3. Condiciones para la exportación.

Los equipos cliente montan el SF remoto con la orden mount y acceden a los datos como si fuesen locales. Incorporan con cada operación, una cookie que se le manda cuando montan el directorio. Al exportar un fichero, se exporta su nodo-i y sus bloques de datos. Es importante que un equipo puede ser servidor y cliente NFS al mismo tiempo.

-. Organización: Se basa en el protocolo Remove Call Procedure (RPC), para encapsular llamadas al servidor cuando se piden archivos remotos (de manera transparente para el usuario). Stateless, el servidor trabaja sin mantener información del estado de cada uno de los clientes.

a) Surge la necesidad de bloquear archivos accedidos concurrentemente por varios clientes → se crean por tanto demonios independientes. El cliente es el responsable de mantener la coherencia.

NFS tiene bastantes problemas de seguridad, uso de herramientas adicionales.

-. Configuración del lado servidor:

1. /etc/exports → Fichero en el que se indica que SFs se exportan, bajo qué condiciones y a qué ordenadores.
2. /usr/sbin/exportfs → Actualiza la información de los SFs exportados y muestra un listado con dicha información
    a. -r → Re-exporta los directorios indicados en /etc/exports.
    b. -a → exporta o deja de exportar /etc/exports.
    c. -v → muestra los directorios exportados y las opciones.
3. /usr/bin/showmount → Información en un servidor NFS.
    a. -a → clientes conectados y directorios utilizados.
    b. -d → Listado de los directorios montados.

- Demonios del lado del servidor:

1. rpcbind o portmap: Facilita la conexión entre el cliente y el servidor mediante las llamadas RPC. Tiene que estar lanzado para que NFS funcione.
2. nfsd: Implementa, en el nivel de usuario, los servicios NFS.
3. rpc.mount: Maneja las peticiones de montaje de directorios de los clientes, comprobando la petición de sistemas de ficheros exportados.

-. Opciones en el servidor: Para configurar qué demonios se exportan, bajo qué condiciones y a que equipos, todo esto se maneja en el directorio /etc/exports.

```
1 ruta dirección(opción)
```

1. ruta es el nombre del directorio a exportar vía NFS.
2. dirección a quién es exportado (IP, dirección red, etc.).
3. opción especifica el tipo de acceso al directorio.
    a. rw o ro → Modo lectura-escritura o sólo lectura.
    b. root_squash → Mapea los uid/gid anónimo.
    c. no_root_squash → No hacer lo anterior.
    d. all_squash → Mapea todos los usuarios al usuario anónimo.
    e. anonuid o anongid → Establecer el uid o el gid del usuario al que realizar el mapeo, distinto del usuario anónimo.

-. La orden mount permite montar el SF remoto:

```
1 $ mount -t nfs -o opciones_nfs 191.168.6.10:/home /datos
```

1. -t nfs: establece el tipo de SF.
2. 191.168.6.10:/home servidor y directorio remoto a montar.

Si en el fichero /etc/fstab se indica el listado de los se indica el listado de los sistemas de ficheros remotos a montar, el punto de montaje y las opciones, el montaje se puede realizar en tiempo de arranque:

```
1 191.168.6.10:/home  /datos  nfs  defaults,opciones_nfs 0 0
```

Opciones de mount:

- `soft` ⇒ Si el servidor NFS falla durante un tiempo, las operaciones que intentaban acceder a él recibirán un código de error.
- `hard` ⇒ Si un proceso está realizando una operación de E/S con un fichero vía NFS y el servidor NFS no responde, el proceso no puede ser interrumpido o matado (no acepta la señal KILL) salvo que se especifique la opción `intr`. Siempre que usemos `rw` deberíamos usar hard, para no dejar el SF remoto inconsistente.
- `intr` ⇒ Se permite señales de interrupción para los procesos bloqueados en una operación de E/S en un servidor NFS.

⋆ : `soft` va en contra de la filosofía de NFS.

- `bg` ⇒ Si el montaje del SF remoto falla, que siga intentándolo en *background*, hasta que lo consiga o desista porque se han hecho `retry` intentos
- `retry=n` ⇒ Nº de intentos que se deben hacer para montar el SF remoto, antes de desistir si la conexión falla.
- `timeo=n` ⇒ Tiempo a esperar entre cada intento de montaje si la conexión falla.
- `rsize=8192` o `wsize=8192` ⇒ Tamaño de los *buffers* de lectura o escritura.

-. NFS ejemplos:

- Ejemplos en el ***servidor*** (fichero `/etc/exports`):

```
/home 191.168.6.15(rw,root_squash) 191.168.6.16(rw,no_root_squash)
/import 191.168.8.20(rw,all_squash)
/tools 191.168.6.0/24(ro,all_squash,anonuid=500,anongid=100)
```

- Ejemplos en el ***cliente***:
  - En el fichero `/etc/fstab`:

```
julieta:/home       /home         nfs  defaults,rw,bg,hard,intr 0 0
julieta:/import     /nfs/import   nfs  defaults,rw,bg,hard,intr 0 0
191.168.6.10:/tools /nfs/tools    nfs  defaults,ro,bg,soft      0 0
```

  - También se puede realizar el montaje de forma manual:

```
$ mount /home          #(configurado /etc/fstab)
$ mount /nfs/import    #(configurado /etc/fstab)
$ mount -t nfs -o rw,bg,hard,intr julieta:/home /home
$ mount -t nfs -o rw,bg,hard,intr julieta:/import /nfs/import
$ mount -t nfs -o ro,soft,bg 191.168.6.10:/tools /nfs/tools
```

- NIS: conceptos básicos

1. Ficheros de configuración: en un entorno real, muchos ficheros de configuración son similares de una máquina a otra.

2. NIS (Network Information Service): es un servicio de red para compartir cierta información. Todos los servicios acceden a una misma base de datos de configuración, esto permite centralizar la autenticación de servicios. Los ficheros de las base de datos están en el equipo servidor y contienen información como:

    a. login names / passwords / home directories → /etc/passwd.
    b. group information → /etc/group.

En el lado del servidor, los ficheros se pre-procesan para convertirlos en un formato binario con hashing. El dominio NIS, es la clave para poder localizar al servidor. Los ficheros de las BDs residen a partir del directorio /var/yp, en un subdirectorio con el nombre del dominio.



-. Configuración de NIS:

1. Existe la posibilidad de configurar varios servidores esclavos, que tendrán una copia de las bases de datos. Un cliente puede acceder a varios servidores o dominios.

2. NSS (Name Service Switch): provee una interfaz para configurar y acceder a diferentes bases de datos de cuentas de usuarios y claves como /etc/passwd, /etc/group, /etc/hosts, LDAP, etc. También nos permite indicar cómo se resolverá cierta información de la configuración, en /etc/nsswitch.conf.

-. Demonios de NIS:

1. rpcbind o portmap: Facilita la conexión entre el cliente y servidor mediante las llamadas RPC, lanzado en cliente y servidor.
2. ypserv: Es el encargado de gestionar el servicio NIS, tiene que estar en ejecución en el servidor.
3. rpc.yppasswdd: permite la actualización de las contraseñas desde los equipos cliente, se ejecuta en el servidor.
4. ypbind: es el encargado de gestionar las peticiones en el cliente.

-. Instalación del servidor:

1. Pasos para el servidor en Ubuntu/Debian:

1. Instalar paquete `nis` (instala `portmap`). Indicar dominio a utilizar (`pas_nis`) y esperar intento fallido de *binding*.
2. Cambiar el fichero `/etc/default/nis` e indicar `NISSERVER=master`.
3. Añadir la IP del servidor al fichero `/etc/yp.conf`:

```
ypserver localhost
```

4. Configurar el servidor (crea las bases de datos): `sudo /usr/lib/yp/ypinit -m`. Este paso habrá que repetirlo cada vez que cambiemos las bases de datos.
5. Reiniciar el servicio: (`sudo /etc/init.d/nis restart`).
6. Comprobar que todo funciona: `rpcinfo -p`.
7. Configurar el NSS (`/etc/nsswitch.conf`)

```
passwd:         compat nis
group:          compat nis
shadow:         compat nis
```

2. Pasos en el cliente en Ubuntu/Debian:

1. Instalar paquete `nis` (instala `portmap`). Indicar dominio a utilizar (`pas_nis`) y esperar intento fallido de *binding*.
2. Añadir la IP del servidor al fichero `/etc/yp.conf`:

```
1 ypserver 192.168.117.23
```

3. Configurar el NSS (`/etc/nsswitch.conf`)

```
1 passwd:         compat nis
2 group:          compat nis
3 shadow:         compat nis
```

4. Reiniciar el servicio: (`sudo /etc/init.d/nis restart`).

⋆ El dominio por defecto se encuentra en `/etc/defaultdomain`.

-. Seguridad:

1. Utilidades como clientes:
    a. yppasswd: Permite que los usuarios puedan cambiar su contraseña en el servidor NIS.
    b. ypchsh: Permite cambiar el shell usuario en el servidor NIS.
    c. ypchfn: Cambia el campo gecos del usuario en el servidor NIS.
    d. ypcat: Permite conocer el contenido el contenido de un mapa NIS. Por ejemplo:
        i. ypcat passwd → visualiza el fichero de passwords.
        ii. ypcat ypservers → muestra los servidores disponibles.
    e. ypwhich: Devuelve el nombre del servidor NIS.

2. En el fichero /etc/ypserv.conf se puede indicar listas de control de acceso. Su formato es: host:nisdomain:map:security.

3. Las BDDs se indexan para mejorar su acceso, es decir, se registra ordenadamente datos e informaciones, para elaborar un índice.

- SAMBA, es un sistema de compartición de archivos e impresoras red. Permite la interconexión entre sistemas heterogéneos entre sí, como GNU/Linux y Windows. Se puede controlar el acceso de clientes Windows a servicios de red Windows o Linux. Los protocolos que sigue son:

1. SMB (Server Message Block): Compartir los recursos.
2. CIFS (Common Internet File System): Implementación mejorada de SMB.
3. NetBIOS (Network Basic Input/Output System): Servicio de nombres (Nombres lógicos en la red y sesiones entre lso nombres).

-. Demonios de SAMBA:

1. smbd → Permite la compartición de archivos e impresoras sobre una red SMB y proporciona autenticación y autorización de acceso para clientes SMB.
2. nmbd → Se ocupa de anunciar servicios, es decir, informa a las máquinas en la red de cuales son los servicios disponibles.

-. Configuración de SAMBA, se realiza a traves del fichero smb.conf. Se establecen que recursos del sistema vas a compartir y qué restricciones desea poner en ellos. Consta de varias secciones distintas que empiezan por [nombre-recurso] y siguen con:

1. [global]: define variables de carácter general y aplicables a todos los recursos.
2. [homes]: Permite a los usuarios remotos acceder a su directorio personal desde su máquina local, sean de Windows o Linux, pero han de tener cuenta en la máquina servidora.
3. [printers]: para compartir impresoras.

El inicio y parada de Samba se realiza con /etc/init.d/samba start/stop. Para crear carpetas compartidas:

```
pas@pas-virtual-debian:~$ sudo mkdir -p /home/shares/allusers
pas@pas-virtual-debian:~$ sudo chown -R root:users /home/shares/allusers/
pas@pas-virtual-debian:~$ sudo chmod -R ug+rwx,o+rx-w /home/shares/allusers/
pas@pas-virtual-debian:~$ sudo mkdir -p /home/shares/anonymous
pas@pas-virtual-debian:~$ sudo chown -R root:users /home/shares/anonymous/
pas@pas-virtual-debian:~$ sudo chmod -R ug+rwx,o+rx-w /home/shares/anonymous/
```

- Samba utiliza su propio sistema de contraseñas. Por tanto, tendremos que hacer lo siguiente por cada usuario que queramos contemplar:

```
pas@pas-virtual-debian:~$ sudo smbpasswd -a pedroa
New SMB password:
Retype new SMB password:
Added user pedroa.
```

- Para acceder a las carpetas compartidas:
  - En Windows, escribimos \\pas-virtual-debian en la barra de direcciones.
  - En GNU/Linux, escribimos smb://pas-virtual-debian en la barra de direcciones.