SCH SOLUTIONS
TECNOLOGIA QUE PROTEGE.
DESARROLLO QUE IMPULSA.

GUÍA COMPLETA DE COMANDOS LINUX

Administración, archivos, redes, procesos, seguridad, scripting y diagnóstico

Edición 2026

SCH Solutions - Tecnología que protege. Desarrollo que impulsa.

Objetivo de esta guía

Servir como referencia práctica para usuarios principiantes, técnicos, administradores de sistemas y profesionales de seguridad. Cada sección combina sintaxis, ejemplos, casos de uso, advertencias y atajos de trabajo.

Cómo usar esta guía

Los comandos están organizados por tarea. La mayoría funciona en GNU/Linux y sistemas tipo Unix; algunas opciones cambian según la distribución o la versión de la herramienta. Los ejemplos usan una shell compatible con Bash.

Convenciones

El símbolo $ representa un usuario normal y # un usuario con privilegios administrativos. No copies esos símbolos al ejecutar los comandos. Sustituye nombres como archivo.txt, usuario o 192.168.1.10 por valores reales.

Precaución

Comandos como rm, dd, mkfs, chmod -R, chown -R, iptables y operaciones sobre discos pueden causar pérdida de datos o dejar el sistema inaccesible. Verifica rutas, dispositivos y opciones antes de confirmar.

Índice temático

1. Fundamentos de terminal y ayuda

Identifica el entorno, comprende la sintaxis y consulta documentación sin salir de la consola.

whoami / id

Muestran el usuario actual y sus identificadores, grupos y contexto de pertenencia.

```
whoami
id [usuario]
```

Ejemplos prácticos

```
whoami
id
id www-data
```

> **Consejo**
>
> id es especialmente útil para verificar permisos efectivos antes de administrar servicios o archivos.

pwd

Imprime la ruta absoluta del directorio de trabajo actual.

```
pwd
```

Ejemplos prácticos

```
cd /var/log
pwd
```

man / info / help

Abren manuales y ayuda integrada.

```
man comando
info comando
comando --help
help builtin
```

Ejemplos prácticos

```
man ls
man 5 passwd
grep --help
help cd
```

> **Consejo**
>
> Las secciones de man separan comandos, llamadas del sistema, archivos de configuración y administración. man 5 passwd describe el formato del archivo, no el comando.

type / command -v / which

Indican cómo resolverá la shell un comando: alias, función, builtin o ejecutable.

```
type nombre
command -v nombre
which nombre
```

Ejemplos prácticos

```
type cd
command -v python3
which ssh
```

history

Consulta y reutiliza el historial de comandos.

```
history
history N
!N
!!
```

Ejemplos prácticos

```
history 20
!145
sudo !!
```

> **Consejo**
>
> Evita guardar secretos en la línea de comandos. En Bash puedes anteponer un espacio si HISTCONTROL incluye ignorespace.

2. Navegación y sistema de archivos

Muévete con seguridad entre rutas absolutas, relativas y directorios especiales.

ls

Lista archivos y directorios.

```
ls [opciones] [ruta]
```

Ejemplos prácticos

```
ls
ls -lah
ls -lt /var/log
ls -R proyecto | less
```

> **Consejo**
>
> -a incluye ocultos, -l muestra detalle, -h humaniza tamaños y -t ordena por fecha.

cd

Cambia el directorio actual.

```
cd [ruta]
```

Ejemplos prácticos

```
cd /etc
cd ..
cd -
cd ~
cd ~/proyectos
```

tree

Muestra la estructura jerárquica de carpetas.

```
tree [opciones] [ruta]
```

Ejemplos prácticos

```
tree -L 2
tree -a -I ".git|node_modules" proyecto
```

stat

Muestra metadatos detallados de archivos o sistemas de archivos.

```
stat archivo
stat -f ruta
```

Ejemplos prácticos

```
stat /etc/passwd
stat -f /home
```

du / df

du calcula uso por directorio; df informa espacio disponible por sistema de archivos.

```
du [opciones] ruta
df [opciones]
```

Ejemplos prácticos

```
du -sh /var/log
du -h --max-depth=1 /home | sort -h
df -hT
df -i
```

> **Consejo**
>
> Cuando un disco parece tener espacio pero no permite crear archivos, revisa inodos con df -i.

3. Archivos, directorios y enlaces

Crea, copia, mueve, elimina y enlaza información de forma controlada.

touch / mkdir

Crea archivos vacíos, actualiza marcas de tiempo y crea directorios.

```
touch archivo
mkdir [opciones] directorio
```

Ejemplos prácticos

```
touch notas.txt
mkdir proyecto
mkdir -p app/{src,tests,docs}
mkdir -m 750 privado
```

cp

Copia archivos y directorios.

```
cp [opciones] origen destino
```

Ejemplos prácticos

```
cp archivo.txt copia.txt
cp -a sitio/ respaldo/
cp -iv *.conf /tmp/config/
```

> **Consejo**
>
> -a preserva atributos y copia recursivamente. -i pregunta antes de sobrescribir y -v muestra cada operación.

mv

Mueve o renombra archivos y directorios.

```
mv [opciones] origen destino
```

Ejemplos prácticos

```
mv borrador.txt final.txt
mv *.log archivo_logs/
mv -i config.yml /etc/app/
```

rm / rmdir

Elimina archivos o directorios.

```
rm [opciones] ruta
rmdir directorio_vacio
```

Ejemplos prácticos

```
rm archivo.tmp
rm -i documento.txt
rm -r carpeta
rmdir carpeta_vacia
```

> **Consejo**
>
> No uses rm -rf con variables vacías o rutas no verificadas. Prueba primero con printf, ls o echo para inspeccionar la expansión.

ln

Crea enlaces duros o simbólicos.

```
ln origen enlace
ln -s destino enlace
```

Ejemplos prácticos

```
ln archivo.dat copia_dura.dat
ln -s /opt/app/current app_actual
readlink -f app_actual
```

file / basename / dirname

Identifican el tipo de archivo y separan componentes de una ruta.

```
file ruta
basename ruta
dirname ruta
```

Ejemplos prácticos

```
file descarga.bin
basename /var/log/nginx/access.log
dirname /var/log/nginx/access.log
```

4. Búsqueda y procesamiento de texto

Encuentra información, filtra grandes volúmenes de datos y transforma contenido desde la shell.

find

Busca archivos por nombre, tipo, tamaño, fecha, permisos y otras propiedades.

```
find ruta criterios acciones
```

Ejemplos prácticos

```
find . -type f -name "*.log"
find /var/log -type f -size +100M
find . -type f -mtime -7
find . -type f -name "*.tmp" -delete
find . -type f -exec chmod 640 {} +
```

> **Consejo**
>
> Antes de usar -delete, ejecuta la misma búsqueda sin esa acción. El orden de los criterios importa.

locate

Busca rutas mediante una base de datos indexada.

```
locate patrón
sudo updatedb
```

Ejemplos prácticos

```
locate sshd_config
locate -i "manual.pdf"
```

grep

Busca patrones dentro de archivos o flujos de texto.

```
grep [opciones] patrón [archivos]
```

Ejemplos prácticos

```
grep "ERROR" app.log
grep -Rni "password" ./config
grep -E "WARN|ERROR" sistema.log
grep -v "DEBUG" app.log
journalctl -u nginx | grep -i failed
```

> **Consejo**
>
> -R recorre subdirectorios, -n muestra líneas, -i ignora mayúsculas y -E habilita expresiones regulares extendidas.

cat / less / head / tail

Visualizan archivos completos o porciones.

```
cat archivo
less archivo
head -n N archivo
tail -n N archivo
```

Ejemplos prácticos

```
cat /etc/os-release
less /var/log/syslog
head -n 20 datos.csv
tail -f /var/log/nginx/access.log
```

> **Consejo**
>
> Para archivos grandes usa less o tail en lugar de cat.

cut / paste / tr

Extraen columnas, combinan líneas y sustituyen caracteres.

```
cut -d DELIM -f CAMPOS archivo
paste archivo1 archivo2
tr ORIGEN DESTINO
```

Ejemplos prácticos

```
cut -d: -f1 /etc/passwd
paste nombres.txt emails.txt
echo "HOLA" | tr "A-Z" "a-z"
```

sort / uniq / wc

Ordenan, deduplican y cuentan contenido.

```
sort [opciones]
uniq [opciones]
wc [opciones]
```

Ejemplos prácticos

```
sort -n numeros.txt
sort -t, -k3,3n ventas.csv
sort usuarios.txt | uniq -c | sort -nr
wc -l acceso.log
```

sed

Realiza sustituciones y ediciones de texto no interactivas.

```
sed [opciones] "expresión" archivo
```

Ejemplos prácticos

```
sed "s/http:/https:/g" config.txt
sed -n "10,20p" archivo.log
sed -i.bak "s/DEBUG=false/DEBUG=true/" .env
```

> **Consejo**
>
> Usa una copia de respaldo al editar in-place: -i.bak crea el archivo original con extensión .bak.

awk

Procesa registros y columnas con reglas y acciones.

```
awk "patrón { acción }" archivo
```

Ejemplos prácticos

```
awk -F: "{print $1, $7}" /etc/passwd
awk "{sum += $3} END {print sum}" datos.txt
awk -F, "$4 > 1000 {print $1, $4}" ventas.csv
```

xargs

Convierte entradas estándar en argumentos para otros comandos.

```
comando_productor | xargs [opciones] comando
```

Ejemplos prácticos

```
find . -type f -name "*.tmp" -print0 | xargs -0 rm -f
printf "%s\0" *.jpg | xargs -0 -n1 file
```

> **Consejo**
>
> Usa separadores nulos (-print0 y -0) para manejar nombres con espacios o saltos de línea.

5. Usuarios, grupos y permisos

Administra identidades locales y aplica el principio de mínimo privilegio.

sudo / su

Ejecutan comandos con otra identidad o abren una shell de usuario.

```
sudo comando
sudo -u usuario comando
su - usuario
```

Ejemplos prácticos

```
sudo systemctl restart nginx
sudo -u postgres psql
su - backup
```

> **Consejo**
>
> Prefiere sudo para acciones puntuales porque deja trazabilidad y reduce el tiempo como root.

useradd / adduser / usermod / userdel

Crean, modifican y eliminan usuarios.

```
sudo useradd [opciones] usuario
sudo usermod [opciones] usuario
sudo userdel [opciones] usuario
```

Ejemplos prácticos

```
sudo useradd -m -s /bin/bash ana
sudo passwd ana
sudo usermod -aG sudo,docker ana
sudo userdel -r ana
```

groupadd / groups / getent

Administran grupos y consultan bases de identidades.

```
sudo groupadd grupo
groups [usuario]
getent passwd [usuario]
getent group [grupo]
```

Ejemplos prácticos

```
sudo groupadd desarrolladores
groups ana
getent passwd www-data
getent group docker
```

chmod

Modifica permisos de lectura, escritura y ejecución.

```
chmod modo ruta
```

Ejemplos prácticos

```
chmod 640 config.ini
chmod 755 script.sh
chmod u+x deploy.sh
chmod -R g+rX proyecto/
```

> **Consejo**
>
> Los modos numéricos comunes son 600, 640, 644, 700, 750 y 755. Evita 777 salvo casos excepcionales y controlados.

chown / chgrp

Cambian propietario y grupo.

```
chown [opciones] propietario[:grupo] ruta
chgrp grupo ruta
```

Ejemplos prácticos

```
sudo chown app:app /opt/app/config.yml
sudo chown -R www-data:www-data /var/www/sitio
chgrp desarrolladores proyecto
```

umask

Define permisos que se restan al crear nuevos archivos y directorios.

```
umask
umask valor
```

Ejemplos prácticos

```
umask
umask 027
```

> **Consejo**
>
> Con umask 027, los archivos suelen crearse como 640 y los directorios como 750.

getfacl / setfacl

Consultan y aplican ACL para permisos más granulares.

```
getfacl ruta
setfacl [opciones] ruta
```

Ejemplos prácticos

```
getfacl reporte.csv
setfacl -m u:ana:rw reporte.csv
setfacl -m d:g:equipo:rwx compartido/
setfacl -x u:ana reporte.csv
```

6. Procesos, trabajos y servicios

Observa, prioriza y controla aplicaciones en ejecución, trabajos de shell y servicios del sistema.

ps / pgrep / pidof

Listan procesos y localizan PID.

```
ps [opciones]
pgrep [opciones] patrón
pidof programa
```

Ejemplos prácticos

```
ps aux
ps -ef --forest
pgrep -a nginx
pgrep -u www-data php
pidof sshd
```

top / htop

Monitorean procesos y uso de recursos en tiempo real.

```
top
htop
```

Ejemplos prácticos

```
top -o %MEM
htop
```

> **Consejo**
>
> En top: P ordena por CPU, M por memoria, k termina un proceso y 1 muestra CPUs individuales.

kill / pkill / killall

Envían señales a procesos.

```
kill [-SEÑAL] PID
pkill [-SEÑAL] patrón
killall programa
```

Ejemplos prácticos

```
kill 1234
kill -TERM 1234
kill -KILL 1234
pkill -f "python.*worker"
```

> **Consejo**
>
> Intenta TERM antes de KILL; KILL no permite limpieza ni guardado de estado.

nice / renice

Ajustan prioridad de CPU.

```
nice -n VALOR comando
renice VALOR -p PID
```

Ejemplos prácticos

```
nice -n 10 tar -czf respaldo.tar.gz datos/
sudo renice -5 -p 1234
```

jobs / bg / fg / nohup

Controlan trabajos iniciados desde una shell.

```
jobs
bg [%N]
fg [%N]
nohup comando &
```

Ejemplos prácticos

```
comando_largo &
jobs
fg %1
nohup ./proceso.sh >proceso.log 2>&1 &
```

systemctl

Administra unidades de systemd.

```
systemctl acción unidad
```

Ejemplos prácticos

```
sudo systemctl start nginx
sudo systemctl enable --now nginx
systemctl status nginx
sudo systemctl restart ssh
systemctl list-units --type=service --state=failed
```

journalctl

Consulta logs del journal de systemd.

```
journalctl [opciones]
```

Ejemplos prácticos

```
journalctl -b
journalctl -u nginx --since today
journalctl -p err..alert
journalctl -f
journalctl --disk-usage
```

7. Paquetes y actualizaciones

Instala, actualiza y elimina software según la familia de distribución.

Los gestores de paquetes resuelven dependencias y registran cambios. No mezcles repositorios de versiones distintas sin comprender sus implicancias.

APT - Debian, Ubuntu y derivadas

Gestiona paquetes .deb y repositorios APT.

```
sudo apt update
sudo apt upgrade
sudo apt install paquete
sudo apt remove paquete
```

Ejemplos prácticos

```
sudo apt update && sudo apt full-upgrade
sudo apt install curl git build-essential
apt search nginx
apt show nginx
sudo apt autoremove
```

DNF - Fedora, RHEL y derivadas

Gestiona paquetes RPM con DNF.

```
sudo dnf check-update
sudo dnf upgrade
sudo dnf install paquete
```

Ejemplos prácticos

```
sudo dnf upgrade --refresh
sudo dnf install nginx
dnf search podman
dnf info openssh-server
sudo dnf autoremove
```

Pacman - Arch Linux y derivadas

Gestiona paquetes en Arch.

```
sudo pacman -Syu
sudo pacman -S paquete
sudo pacman -Rns paquete
```

Ejemplos prácticos

```
sudo pacman -Syu
sudo pacman -S git base-devel
pacman -Ss neovim
pacman -Qi openssh
pacman -Qdtq
```

Zypper - openSUSE

Gestiona paquetes y repositorios en openSUSE.

```
sudo zypper refresh
sudo zypper update
sudo zypper install paquete
```

Ejemplos prácticos

```
sudo zypper dup
zypper search nginx
zypper info nginx
sudo zypper remove nginx
```

Snap / Flatpak

Instalan aplicaciones empaquetadas de forma distribuible.

```
snap find aplicación
sudo snap install paquete
flatpak search aplicación
flatpak install remoto app-id
```

Ejemplos prácticos

```
snap list
flatpak list
flatpak update
flatpak uninstall --unused
```

8. Discos, particiones y sistemas de archivos

Inspecciona dispositivos, monta volúmenes y administra almacenamiento sin comprometer datos.

lsblk / blkid

Listan dispositivos de bloques, particiones, UUID y tipos de sistema de archivos.

```
lsblk [opciones]
blkid
```

Ejemplos prácticos

```
lsblk -f
lsblk -o NAME,SIZE,FSTYPE,MOUNTPOINTS,MODEL
sudo blkid
```

mount / umount

Montan y desmontan sistemas de archivos.

```
sudo mount dispositivo punto
sudo umount punto
```

Ejemplos prácticos

```
sudo mkdir -p /mnt/datos
sudo mount /dev/sdb1 /mnt/datos
mount | column -t
sudo umount /mnt/datos
```

> **Consejo**
>
> Antes de desmontar, cierra archivos y procesos que usen el punto de montaje. Usa lsof o fuser para identificarlos.

fdisk / parted

Crean y modifican tablas de particiones.

```
sudo fdisk dispositivo
sudo parted dispositivo
```

Ejemplos prácticos

```
sudo fdisk -l
sudo fdisk /dev/sdb
sudo parted /dev/sdb print
```

> **Consejo**
>
> Estas herramientas modifican la estructura del disco. Confirma siempre el nombre del dispositivo con lsblk.

mkfs / fsck

Crean y verifican sistemas de archivos.

```
sudo mkfs.tipo dispositivo
sudo fsck dispositivo
```

Ejemplos prácticos

```
sudo mkfs.ext4 -L DATOS /dev/sdb1
sudo fsck -f /dev/sdb1
```

> **Consejo**
>
> Nunca ejecutes mkfs sobre un volumen con datos que quieras conservar. En general, fsck debe ejecutarse con el volumen desmontado.

dd

Copia datos a bajo nivel y crea imágenes.

```
sudo dd if=origen of=destino [opciones]
```

Ejemplos prácticos

```
sudo dd if=imagen.iso of=/dev/sdX bs=4M status=progress conv=fsync
sudo dd if=/dev/sdb of=disco.img bs=64K status=progress
```

> **Consejo**
>
> Un destino equivocado destruye datos. Verifica of= dos veces y usa rutas de dispositivo completas.

LVM

Administra volúmenes físicos, grupos y volúmenes lógicos.

```
pvs; vgs; lvs
pvcreate dispositivo
vgcreate grupo dispositivo
lvcreate ...
```

Ejemplos prácticos

```
sudo pvs
sudo vgs
sudo lvs
sudo lvextend -r -L +10G /dev/vg0/datos
```

9. Redes y conectividad

Diagnostica interfaces, rutas, DNS, puertos y tráfico.

ip

Consulta y configura interfaces, direcciones y rutas.

```
ip addr
ip link
ip route
```

Ejemplos prácticos

```
ip -br addr
ip link show
ip route
sudo ip link set eth0 up
sudo ip addr add 192.168.10.20/24 dev eth0
```

ping

Comprueba alcance y latencia ICMP.

```
ping [opciones] destino
```

Ejemplos prácticos

```
ping -c 4 1.1.1.1
ping -c 4 example.com
ping -I eth0 192.168.1.1
```

> **Consejo**
>
> Si funciona por IP pero no por nombre, el problema probablemente sea DNS.

traceroute / tracepath / mtr

Muestran el camino y pérdidas hacia un destino.

```
traceroute destino
tracepath destino
mtr destino
```

Ejemplos prácticos

```
traceroute example.com
tracepath 8.8.8.8
mtr -rwzc 20 example.com
```

ss

Muestra sockets, puertos y procesos asociados.

```
ss [opciones]
```

Ejemplos prácticos

```
ss -tulpn
ss -lnt
ss -s
ss -tn state established
```

> **Consejo**
>
> En sistemas modernos, ss reemplaza en gran medida a netstat.

dig / host / resolvectl

Consultan DNS y estado del resolvedor.

```
dig nombre [tipo]
host nombre
resolvectl status
```

Ejemplos prácticos

```
dig example.com
dig MX example.com
dig @1.1.1.1 example.com
host 8.8.8.8
resolvectl query example.com
```

curl / wget

Realizan solicitudes HTTP y descargan recursos.

```
curl [opciones] URL
wget [opciones] URL
```

Ejemplos prácticos

```
curl -I https://example.com
curl -sS https://api.example.com/health
curl -o archivo.zip https://example.com/archivo.zip
wget -c https://example.com/imagen.iso
```

> **Consejo**
>
> Para automatización usa -fS o --fail-with-body y valida códigos de salida.

nmcli

Administra NetworkManager desde terminal.

```
nmcli objeto acción
```

Ejemplos prácticos

```
nmcli device status
nmcli connection show
nmcli device wifi list
nmcli device wifi connect "SSID" password "CLAVE"
nmcli connection up "Conexión cableada 1"
```

tcpdump

Captura y filtra tráfico de red.

```
sudo tcpdump [opciones] [filtro]
```

Ejemplos prácticos

```
sudo tcpdump -i any -nn
sudo tcpdump -i eth0 port 53
sudo tcpdump -i eth0 host 192.168.1.50 -w captura.pcap
```

> **Consejo**
>
> Captura solo tráfico autorizado y protege los archivos PCAP: pueden contener credenciales, cookies y datos sensibles.

10. Acceso remoto y transferencia

Administra servidores de forma segura mediante SSH y mueve archivos con cifrado.

ssh

Abre sesiones remotas y ejecuta comandos.

```
ssh [opciones] usuario@host
```

Ejemplos prácticos

```
ssh admin@192.168.1.20
ssh -p 2222 admin@servidor
ssh -i ~/.ssh/id_ed25519 admin@host
ssh admin@host "uptime && df -h"
```

ssh-keygen / ssh-copy-id

Generan claves y autorizan acceso por clave pública.

```
ssh-keygen -t ed25519
ssh-copy-id usuario@host
```

Ejemplos prácticos

```
ssh-keygen -t ed25519 -a 100
ssh-copy-id -i ~/.ssh/id_ed25519.pub admin@host
```

scp

Copia archivos sobre SSH.

```
scp [opciones] origen destino
```

Ejemplos prácticos

```
scp archivo.txt admin@host:/tmp/
scp admin@host:/var/log/app.log .
scp -r proyecto/ admin@host:/opt/
```

sftp

Proporciona transferencia interactiva sobre SSH.

```
sftp usuario@host
```

Ejemplos prácticos

```
sftp admin@host
# Dentro de sftp: ls, lls, get, put, mkdir, cd, lcd, exit
```

rsync sobre SSH

Sincroniza eficientemente archivos locales y remotos.

```
rsync [opciones] origen destino
```

Ejemplos prácticos

```
rsync -avh --progress sitio/ admin@host:/var/www/sitio/
rsync -avh --delete backup/ admin@host:/srv/backup/
rsync -avhn --delete origen/ destino/
```

> **Consejo**
>
> Usa primero -n o --dry-run cuando incluyas --delete.

Túneles SSH

Reenvía puertos de forma cifrada.

```
ssh -L local:destino:puerto usuario@host
ssh -R remoto:destino:puerto usuario@host
ssh -D puerto usuario@host
```

Ejemplos prácticos

```
ssh -L 8080:127.0.0.1:80 admin@servidor
ssh -D 1080 admin@servidor
```

11. Compresión, copias y sincronización

Reduce tamaño, empaqueta proyectos y diseña respaldos verificables.

tar

Empaqueta y extrae árboles de archivos.

```
tar [opciones] archivo.tar [rutas]
```

Ejemplos prácticos

```
tar -cvf proyecto.tar proyecto/
tar -czvf proyecto.tar.gz proyecto/
tar -xzvf proyecto.tar.gz
tar -tf proyecto.tar.gz
tar -xzf proyecto.tar.gz -C /destino
```

gzip / bzip2 / xz

Comprimen archivos individuales con distintos balances de velocidad y tamaño.

```
gzip archivo
bzip2 archivo
xz archivo
```

Ejemplos prácticos

```
gzip -k acceso.log
gunzip acceso.log.gz
xz -T0 -9 imagen.raw
unxz imagen.raw.xz
```

zip / unzip

Crean y extraen archivos ZIP.

```
zip [opciones] archivo.zip rutas
unzip archivo.zip
```

Ejemplos prácticos

```
zip -r entrega.zip proyecto/
unzip -l entrega.zip
unzip entrega.zip -d entrega/
```

rsync local

Copia incrementalmente preservando atributos.

```
rsync [opciones] origen/ destino/
```

Ejemplos prácticos

```
rsync -aHAX --info=progress2 /datos/ /backup/datos/
rsync -aHAXn --delete /datos/ /backup/datos/
```

sha256sum

Calcula y verifica integridad mediante hash.

```
sha256sum archivo
sha256sum -c archivo.sha256
```

Ejemplos prácticos

```
sha256sum imagen.iso > imagen.iso.sha256
sha256sum -c imagen.iso.sha256
```

> **Regla 3-2-1**
>
> Mantén al menos 3 copias de los datos, en 2 medios distintos y 1 copia fuera del sitio. Un respaldo no es confiable hasta que se prueba su restauración.

12. Logs, diagnóstico y rendimiento

Encuentra cuellos de botella y evidencia útil antes de cambiar configuraciones.

uname / hostnamectl / os-release

Identifican kernel, arquitectura, hostname y distribución.

```
uname [opciones]
hostnamectl
cat /etc/os-release
```

Ejemplos prácticos

```
uname -a
uname -r
hostnamectl
cat /etc/os-release
```

uptime / free / vmstat

Muestran carga, memoria y actividad del sistema.

```
uptime
free -h
vmstat [intervalo] [conteo]
```

Ejemplos prácticos

```
uptime
free -h
vmstat 1 5
```

lscpu / lspci / lsusb

Inventarían CPU y dispositivos PCI/USB.

```
lscpu
lspci [opciones]
lsusb
```

Ejemplos prácticos

```
lscpu
lspci -nnk
lsusb -t
```

dmesg

Consulta mensajes del kernel.

```
dmesg [opciones]
```

Ejemplos prácticos

```
sudo dmesg -T | less
sudo dmesg -T --level=err,warn
sudo dmesg -w
```

lsof / fuser

Muestran archivos y sockets abiertos por procesos.

```
lsof [ruta]
fuser [opciones] recurso
```

Ejemplos prácticos

```
sudo lsof /var/log/app.log
sudo lsof -i :8080
sudo fuser -v /mnt/datos
sudo fuser -k 8080/tcp
```

iostat / pidstat / sar

Miden CPU, discos y procesos a lo largo del tiempo.

```
iostat [opciones]
pidstat [intervalo]
sar [opciones]
```

Ejemplos prácticos

```
iostat -xz 1
pidstat -dur 1
sar -u 1 5
sar -n DEV 1 5
```

strace

Traza llamadas del sistema de un proceso.

```
strace [opciones] comando
strace -p PID
```

Ejemplos prácticos

```
strace -f -o traza.log ./app
sudo strace -p 1234
strace -e trace=file cat /etc/hosts
```

> **Consejo**
>
> Puede generar mucha salida y afectar rendimiento. Úsalo de forma acotada.

13. Variables, shell y automatización

Combina comandos, controla flujos y crea scripts reproducibles.

Variables y entorno

Define variables locales, exportadas y consulta el entorno.

```
NOMBRE=valor
export NOMBRE=valor
env
printenv NOMBRE
```

Ejemplos prácticos

```
PROYECTO=/opt/app
export EDITOR=nano
echo "$PROYECTO"
printenv PATH
```

Redirecciones y tuberías

Conectan entrada, salida y errores entre comandos y archivos.

```
comando > archivo
comando >> archivo
comando 2> errores
comando1 | comando2
```

Ejemplos prácticos

```
ls -lah > listado.txt
./app >> app.log 2>&1
journalctl -u nginx | grep -i error
cat datos.txt | sort | uniq -c
```

Operadores lógicos

Ejecutan comandos según éxito, fallo o secuencia.

```
comando1 && comando2
comando1 || comando2
comando1 ; comando2
```

Ejemplos prácticos

```
mkdir build && cd build
grep patrón archivo || echo "No encontrado"
apt update; apt list --upgradable
```

Comodines y expansión

Expanden nombres, rangos y resultados.

```
*  ?  [abc]  {a,b}  {1..5}  $(comando)
```

Ejemplos prácticos

```
ls *.log
cp config.{yml,yml.bak}
mkdir -p parte-{1..4}
echo "Kernel: $(uname -r)"
```

cron

Programa tareas recurrentes.

```
crontab -e
crontab -l
```

Ejemplos prácticos

```
0 2 * * * /usr/local/bin/backup.sh >> /var/log/backup.log 2>&1
*/15 * * * * /opt/app/healthcheck.sh
```

> **Consejo**
>
> Usa rutas absolutas y un entorno explícito. Prueba el script manualmente con el mismo usuario antes de programarlo.

at

Programa una tarea única para el futuro.

```
at hora
atq
atrm ID
```

Ejemplos prácticos

```
echo "/usr/local/bin/tarea.sh" | at 23:30
at now + 30 minutes
atq
```

Plantilla de script Bash seguro

```
#!/usr/bin/env bash
set -Eeuo pipefail
IFS=$'\n\t'

log() { printf '[%s] %s\n' "$(date '+%F %T')" "$*"; }
error() { printf 'ERROR: %s\n' "$*" >&2; exit 1; }

[[ $# -ge 1 ]] || error "Uso: $0 <directorio>"
directorio=$1
[[ -d "$directorio" ]] || error "No existe: $directorio"

log "Procesando $directorio"
find "$directorio" -type f -name '*.log' -mtime +30 -print
log "Finalizado"
```

14. Seguridad y endurecimiento básico

Reduce superficie de ataque, revisa exposición y conserva trazabilidad.

passwd / chage

Gestionan contraseñas y expiración.

```
passwd [usuario]
chage [opciones] usuario
```

Ejemplos prácticos

```
sudo passwd ana
sudo chage -l ana
sudo chage -M 90 -m 1 -W 14 ana
```

ufw

Administra firewall simplificado en Debian/Ubuntu.

```
sudo ufw acción
```

Ejemplos prácticos

```
sudo ufw default deny incoming
sudo ufw default allow outgoing
sudo ufw allow OpenSSH
sudo ufw allow 443/tcp
sudo ufw enable
sudo ufw status numbered
```

firewalld

Administra zonas y reglas en Fedora/RHEL.

```
firewall-cmd [opciones]
```

Ejemplos prácticos

```
sudo firewall-cmd --get-active-zones
sudo firewall-cmd --zone=public --add-service=https --permanent
sudo firewall-cmd --reload
sudo firewall-cmd --list-all
```

iptables / nft

Gestionan reglas de filtrado de bajo nivel.

```
sudo nft list ruleset
sudo iptables -L -n -v
```

Ejemplos prácticos

```
sudo nft list ruleset
sudo iptables -S
```

> **Consejo**
>
> No apliques reglas remotas sin un canal de recuperación. Un error puede cortar tu acceso SSH.

last / lastlog / who / w

Auditan sesiones actuales e históricas.

```
last
lastlog
who
w
```

Ejemplos prácticos

```
last -a | head
lastlog | head
who -a
w
```

find para auditoría

Localiza permisos riesgosos y cambios recientes.

```
find ruta criterios
```

Ejemplos prácticos

```
sudo find / -xdev -type f -perm -4000 -print
sudo find /etc -type f -mtime -7 -ls
find /var/www -type f -perm -0002 -print
```

openssl

Inspecciona certificados, crea hashes y realiza operaciones criptográficas.

```
openssl subcomando [opciones]
```

Ejemplos prácticos

```
openssl version
openssl s_client -connect example.com:443 -servername example.com </dev/null
openssl x509 -in cert.pem -noout -subject -issuer -dates
openssl rand -hex 32
```

> **Mínimo privilegio**
>
> Deshabilita servicios innecesarios, limita puertos expuestos, aplica actualizaciones, usa claves SSH, protege secretos y revisa logs. La seguridad depende más de una configuración sostenida que de un único comando.

15. Contenedores y virtualización

Opera entornos aislados y consulta recursos comunes de Docker, Podman y máquinas virtuales.

Docker

Administra imágenes, contenedores, redes y volúmenes.

```
docker subcomando [opciones]
```

Ejemplos prácticos

```
docker version
docker ps -a
docker images
docker run --rm hello-world
docker logs -f contenedor
docker exec -it contenedor sh
docker inspect contenedor
docker system df
```

Docker Compose

Define servicios multi-contenedor.

```
docker compose subcomando
```

Ejemplos prácticos

```
docker compose config
docker compose up -d
docker compose ps
docker compose logs -f
docker compose down
```

Podman

Ofrece una experiencia compatible, con soporte rootless.

```
podman subcomando [opciones]
```

Ejemplos prácticos

```
podman ps -a
podman run --rm docker.io/library/alpine uname -a
podman logs contenedor
podman system df
```

virsh

Administra máquinas virtuales con libvirt.

```
virsh subcomando
```

Ejemplos prácticos

```
virsh list --all
virsh dominfo vm1
virsh start vm1
virsh shutdown vm1
virsh console vm1
```

16. Recetas prácticas y chuleta rápida

Secuencias de diagnóstico y administración listas para adaptar.

Encontrar qué usa un puerto

```
sudo ss -lntp | grep :8080
sudo lsof -i :8080
sudo fuser -v 8080/tcp
```

Liberar un puerto con prudencia

```
PID=$(sudo lsof -t -i :8080)
printf "PID: %s\n" "$PID"
sudo kill -TERM "$PID"
```

Investigar poco espacio en disco

```
df -hT
df -i
sudo du -xhd1 / | sort -h
sudo du -xhd1 /var | sort -h
journalctl --disk-usage
```

Diagnosticar un servicio caído

```
systemctl status servicio
journalctl -u servicio -b --no-pager | tail -n 100
sudo ss -lntp
systemctl cat servicio
```

Probar conectividad web

```
ip -br addr
ip route
ping -c 3 1.1.1.1
resolvectl query example.com
curl -vI https://example.com
```

Respaldar una carpeta con verificación

```
tar -czf backup-$(date +%F).tar.gz /ruta/datos
sha256sum backup-$(date +%F).tar.gz > backup-$(date +%F).sha256
sha256sum -c backup-$(date +%F).sha256
```

Buscar errores recientes

```
journalctl -p err..alert --since "1 hour ago"
sudo dmesg -T --level=err,warn
grep -RniE "error|failed|critical" /var/log 2>/dev/null | head
```

Inventario rápido del sistema

```
hostnamectl
cat /etc/os-release
uname -a
lscpu
free -h
lsblk -f
ip -br addr
systemctl --failed
```

Tabla de referencia rápida

Tarea	Comando
Directorio actual	`pwd`
Listar detallado	`ls -lah`
Buscar archivo	`find /ruta -name "nombre"`
Buscar texto	`grep -Rni "texto" /ruta`
Espacio en disco	`df -hT`
Tamaño de carpeta	`du -sh carpeta`
Procesos	`ps aux / top`

Tarea	Comando
Puertos	`ss -tulpn`
Servicio	`systemctl status nombre`
Logs de servicio	`journalctl -u nombre`
Red	`ip -br addr; ip route`
DNS	`dig nombre / resolvectl query nombre`
Descarga	`curl -O URL / wget URL`
Sincronizar	`rsync -avh origen/ destino/`
Permisos	`chmod / chown`
Integridad	`sha256sum archivo`

Dominar Linux es aprender a observar, comprobar y automatizar.

Usa esta guía como punto de partida, consulta los manuales y prueba primero en entornos controlados.

Publicación y uso

Material educativo preparado por SCH Solutions. Puedes compartir el PDF manteniendo la identidad visual y la atribución. Los ejemplos deben adaptarse al sistema, distribución y políticas de seguridad de cada entorno.

SCH SOLUTIONS

TECNOLOGÍA QUE PROTEGE. DESARROLLO QUE IMPULSA.