



**EBOOK GRATUITO**

# **CHULETA DE COMANDOS LINUX PARA NO PERDERSE**

**GUIA DE BOLSILLO Y  
REFERENCIA PARA  
ADMINISTRAR TU SISTEMA  
GNU/LINUX USANDO EL  
TERMINAL**

**LICENCIA CC BY-NC-SA**

**PARA MÁS INFORMACIÓN,  
VISITA [UBUNTIZANDO.COM](http://UBUNTIZANDO.COM)**





<b>Introducción:</b>	<b>1</b>
Comandos para información del sistema:	2
Comandos para apagar el sistema	3
Comandos para movernos de un directorio a otro	3
Comandos para gestión de archivos y directorios	4
Comandos para la gestión de grupos, usuarios y permisos	5
Comandos para la gestión de paquetes	7
Paquetes con formato .DEB (Debian, Ubuntu, Mint y distribuciones derivadas.):	7
Paquetes con formato .RPM (Red Hat, Fedora, SUSE Linux,etc.):	8
Gestor de paquetes YUM (Red Hat, Fedora y similares):	8
Comandos para crear copias de seguridad	8
Comandos para la gestión de redes	9
<b>Licencia:</b>	<b>11</b>

# Chuleta comandos Linux para no perderse

Última revisión 10 Marzo de 2020

## Introducción:



Aunque pueda parecer lo contrario todavía es habitual usar el terminal para gestionar nuestro sistema GNU/Linux. No es imprescindible para un usuario común pero a nada que profundizas en el sistema y vas ganando experiencia te das cuenta que sin el terminal te estás perdiendo lo mejor de la vida. Al contrario de lo que muchos pueden pensar no es más complicado en Linux que en otros sistemas y es un buen aprendizaje.

## Comandos para información del sistema:

Conocer las capacidades de nuestro sistema es clave para sacarle provecho. En GNU/Linux es posible conocer la configuración de nuestro sistema, versión y sus componentes de hardware usando el terminal. No es necesario instalar otros programas. Por ejemplo:

- **arch:** mostrará la arquitectura de tu ordenador.
- **uname -r:** mostrará la versión del kernel que está usado.
- **dmidecode -q:** mostrará los componentes de hardware presentes en el sistema.
- **hdparm -i /dev/hda:** mostrará las características del disco duro que le indiques.
- **cat /proc/cpuinfo:** mostrará la información sobre la CPU.

- **cat /proc/meminfo:** verificará el uso de memoria RAM.
- **free -m:** muestra el estado y uso de la memoria RAM.
- **cat /proc/net/dev:** mostrará los adaptadores de red y sus estadísticas.
- **cat /proc/mounts:** mostrará el sistema de archivos montado.
- **lspci -tv:** mostrará los dispositivos PCI presentes.
- **lsusb -tv:** mostrará los dispositivos USB conectados.
- **top:** mostrará las tareas de Linux usando la mayoría de CPU.
- **date:** mostrará la fecha actual del sistema.
- **cal 2018:** mostrará el calendario completo del año indicado. En este caso 2018.
- **cal 05 2018:** similar a la anterior, pero solo mostrará el mes del año indicado. En este caso, mayo de 2018.
- **man ping:** muestra las páginas del manual de ayuda para el comando indicado, en este caso el comando ping.

## Comandos para apagar el sistema

De la misma manera podemos forzar a que el sistema haga lo que nosotros queramos. Por ejemplo el forzar el apagado o reinicio de tu ordenador desde el terminal de Linux. Esto, en ocasiones puede ser incluso más útil que hacerlo desde la interfaz gráfica ya que, por ejemplo, podrás **programar el apagado de tu equipo para más tarde** `shutdown -h now`: apagará el sistema.

- **halt:** Otra forma de apagar el sistema es utilizar este comando.
- **shutdown -h 18:55 &:** Este comando programará el apagado planificado del sistema a la hora indicada. En este caso, a las 18:55.
- **shutdown -c:** cancelar un apagado planificado del sistema.
- **shutdown -r now:** reiniciará tu equipo inmediatamente.
- **reboot:** otro comando con el que reiniciar tu equipo.
- **last reboot:** mostrará un historial de cuando se realizaron los últimos reinicios.
- **logout:** cerrará tu actual sesión

## Comandos para movernos de un directorio a otro

No hace muchos años el acceso a nuestros ficheros se hacía tecleando una infinidad de instrucciones en nuestro terminal. Los que vivimos los tiempos de MS/Dos sabemos que aunque los interfaces gráficos han ayudado mucho la consola puede seguir siendo muy necesaria en ocasiones. Así es bueno conocer los comandos que nos permitirán movernos de un directorio a otro sin problemas.

- **cd /home/usuario:** Te llevará directamente hasta la ruta que indiques, en este caso hasta el directorio "usuario".
- **cd ..:** retrocederás un nivel en la jerarquía de directorios.
- **cd ../..:** retrocederás 2 niveles en la jerarquía de directorios.
- **cd:** te llevará al directorio raíz de esa unidad.
- **cd ~usuario:** irás al directorio principal del usuario que indiques.
- **cd -:** regresarás al directorio anterior.
- **pwd:** mostrará la ruta del directorio donde te encuentras actualmente.
- **ls:** mostrará los archivos y carpetas del directorio donde te encuentras.
- **ls -l:** mostrará los detalles de archivos y carpetas del directorio actual.
- **ls -a:** mostrará los archivos ocultos del directorio actual

## Comandos para gestión de archivos y directorios

Estoy seguro que tras probar los comandos anteriores te ha entrado la curiosidad de como hacer más tareas desde el terminal. Y es que se pueden hacer muchas cosas como por ejemplo **crear, copiar, mover o borrar archivos y carpetas desde el terminal de Linux**. Para ello solo necesitarás utilizar una serie de **comandos básicos para Linux con los que podrás gestionar tus archivos** .

- **mkdir Directorio:** creará una nueva carpeta o directorio con nombre que le indiques. En este caso, "Directorio".
- **mkdir Directorio1 Directorio 2:** creará dos carpetas simultáneamente.
- **rmdir Directorio:** borrará la carpeta llamada "Directorio".
- **rm -rf Directorio:** eliminará una carpeta llamada "Directorio" y todo su contenido.
- **mv Viejodirectorio Nuevodirectorio:** renombrará o moverá un archivo o carpeta.
- **cp Archivo:** copiará un archivo.
- **cp Archivo1 Archivo2:** copiará dos archivos simultáneamente.
- **cp -a Directorio:** copiará una carpeta completa, en este caso "Directorio".
- **cp -a Directorio1 Directorio2:** copiará dos directorios simultáneamente.
- **ln -s Archivo Enlacearchivo:** creará un enlace simbólico con un determinado nombre, al archivo o directorio indicado.
- **ln Archivo Enlacearchivo:** creará un enlace físico con el nombre indicado al archivo o directorio indicado

Pero, cómo podemos buscar luego entre nuestros archivos en el sistema. Muy fácil, solo tendrás que usar estos comandos para buscarlos por formato de archivo, por nombre, etc.

- **find / -name Archivo:** buscará ese archivo o directorio comenzado por la raíz del sistema.
- **find / -user Usuario:** buscará archivos y directorios propiedad del usuario "Usuario".
- **find /home/usuario -name \\*.bin:** buscará todos los archivos con la extensión que le indiques, en este caso ". bin", dentro del directorio indicado, que en el ejemplo es "/home/usuario".
- **find /usr/bin -type f -mtime -5:** buscará los archivos creados o cambiados en el sistema dentro de los últimos "5" días.
- **Whereis firefox:** mostrará la ubicación de un archivo binario, de ayuda o fuente. En este caso pregunta dónde está "Firefox".
- **Which firefox:** mostrará la ruta completa al binario/ejecutable que le indiques

## Comandos para la gestión de grupos, usuarios y permisos

Todo sistema operativo tiene un apartado de su configuración destinado a la gestión de usuarios, grupos y permisos. De otro modo aquello sería un “descontrol” total y absoluto y si eres el administrador de sistemas en una empresa sabrás lo importante que es mantener todo bajo el más absoluto control.

Los usuarios de escritorio en cambio no necesitan grandes despliegues pero en ocasiones pueden surgir conflictos con los permisos de archivo. Por ejemplo puede suceder que no puedas editar o borrar un archivo de tu ordenador o que no puedas usar según qué periférico porque no estás en el grupo adecuado.

Por suerte en Linux es posible lidiar con los permisos incluso de una manera más fácil que en Windows. Al menos yo lo encuentro mucho más intuitivo y simple.

Para **gestionar los atributos y permisos de los archivos en Linux**, solo tendrás que escribir el comando adecuado para conseguirlo. Estos son algunos de los más habituales:

- **ls -lh**: Mostrará los permisos de un archivo.
- **chown Usuario1 Archivo1**: cambiará el propietario de un determinado archivo.
- **chown -R Usuario1 directory1**: cambiará el propietario de un determinado directorio y de todos los archivos y subdirectorios contenidos dentro.
- **chattr +i Archivo1**: convierte el archivo en invariable, por lo que no puede ser eliminado, alterado, renombrado, ni enlazado.
- **chattr +s Archivo1**: permite que un archivo sea borrado de forma segura.
- **lsattr**: mostrará los atributos especiales.

Y hablando de usuarios. Cómo gestionar las tareas más simples, cambiar contraseña de un usuario, consultar en qué grupos se encuentra y cómo añadirle a nuevos.

- **whoami**: quién eres tú
- **cat /etc/passwd**: mostrará los usuarios presentes en el equipo.



- **cat /etc/group**: mostrará los grupos presentes en el equipo
- **groupadd *nombre\_del\_grupo***: crear un nuevo grupo en el sistema.
- **groupdel *nombre\_del\_grupo***: borrará un grupo ya existente en el sistema.
- **groupmod -n *nuevo\_grupo viejo\_grupo***: renombrará un grupo ya existente.
- **useradd -c "*Nombre usuario*" -g *admin* -d */home/nuevousuario* -s */bin/bashnuevousuario***: Creará un nuevo usuario perteneciente al grupo "admin".
- **useradd *Usuario1***: creará un nuevo usuario.
- **userdel -r *Usuario1***: borrará un usuario y su directorio Home.
- **passwd**: cambiará tu contraseña de usuario.
- **passwd *Usuario1***: cambiará la contraseña del usuario indicado (solo permitido para el usuario root).
- **chage -E *2019-12-31 Usuario1***: establece una caducidad para la contraseña de un usuario determinado. En este caso expira el 31 de diciembre de 2019.

## Comandos para la gestión de paquetes

Aunque ahora esto de instalar y actualizar paquetes desde el terminal de Linux es mucho más sencillo que quince años atrás, no tenemos que lidiar con dependencias y todo se reduce a seleccionar la aplicación que necesitamos desde el centro de software de nuestra distribución GNU/Linux favorita, hacerlo por terminal sigue siendo la opción más rápida. Sobre todo si precisas instalar múltiples aplicaciones en una misma ocasión.

Por desgracia instalar paquetes no es del mismo modo según la distribución que usemos. En este punto debemos diferenciar los comandos en función del sistema o gestor de paquetes que utilice tu distro Linux: RPM, DEB o YUM.

### Paquetes con formato .DEB (Debian, Ubuntu, Mint y distribuciones derivadas.):

- **dpkg -i *paquete.deb***: instalará o actualizará un paquete deb.
- **dpkg -r *paquete***: eliminará un determinado paquete deb del sistema.
- **dpkg -l**: mostrará todos los paquetes deb instalados en el sistema.

- **dpkg -s *paquete***: mostrará información del paquete indicado instalado en el sistema.
- Gestor de paquetes APT**
- **apt-get install *paquete***: instalará o actualizará un determinado paquete deb.
  - **apt-cdrom install *paquete***: instalará o actualizará un paquete deb desde un cdrom.
  - **apt-get update**: actualizará la lista de paquetes instalados.
  - **apt-get upgrade**: actualizará todos los paquetes instalados.
  - **apt-get remove *paquete***: eliminará el paquete deb indicado del sistema.
  - **apt-get check**: verificará la correcta resolución de las dependencias.
  - **apt-get clean**: limpiará la cache desde los paquetes descargados.

### Paquetes con formato .RPM (Red Hat, Fedora, SUSE Linux,etc.):

- **rpm -ivh *paquete.rpm***: instalará un paquete rpm.
- **rpm -ivh --nodeeps *paquete.rpm***: instalará un paquete rpm ignorando las peticiones de dependencias.
- **rpm -U *paquete.rpm***: actualizará un paquete rpm sin cambiar la configuración de los ficheros.
- **rpm -F *paquete.rpm***: actualizará un determinado paquete rpm solamente si este está instalado.
- **rpm -e *paquete.rpm***: eliminará el paquete rpm indicado.
- **rpm -qa**: mostrará todos los paquetes rpm instalados en el sistema.
- **rpm -qi *paquete***: ofrece la información en un paquete específico instalado.
- **rpm -q *paquete* --changelog**: mostrará el historial de revisiones de un paquete rpm.
- **rpm --checksig *paquete.rpm***: verificará la integridad de un paquete rpm.
- **rpm -qa gpg-pubkey**: verificará la integridad de todos los paquetes rpm instalados

### Gestor de paquetes YUM (Red Hat, Fedora y similares):

- **yum install *paquete***: descargará e instalará un determinado paquete rpm.
- **yum localinstall *paquete.rpm***: instalará un paquete rpm y tratará de resolver todas las dependencias usando tus repositorios.
- **yum update *paquete.rpm***: actualizará todos los paquetes rpm instalados en el sistema.
- **yum update *paquete***: actualizará un determinado paquete rpm.

- **yum remove *paquete***: eliminará el paquete rpm especificado.
- **yum list**: listará todos los paquetes de software instalados en el sistema.
- **yum search *paquete***: encontrará un paquete en repositorio rpm.

## Comandos para crear copias de seguridad

El terminal puede ser muy útil si lo que queremos es **crear nuestras propias copias de seguridad en Linux** sin depender de herramientas de terceros. Y es tan fácil que puede que te preguntes el porqué no lo has probado antes. Desde el terminal puedes crear copias de respaldo completas de tus archivos e incluso incrementales usando algunos de **los comandos más utilizados en Linux para tareas de copia de seguridad y respaldo de archivos**.

- **dump -0aj -f /tmp/home0.bak /home**: para realizar una copia de seguridad completa del directorio /home.
- **dump -1aj -f /tmp/home0.bak /home**: Similar al anterior pero para realizar una copia de seguridad incremental del directorio '/home'.
- **tar -Puf backup.tar /home/user**: Se creará una copia de seguridad incremental del directorio indicado. En este caso, /home/user.
- **find /home/Usuario -name '\*.txt' | xargs cp -av -target-directory=/home/backup/ -parents**: encontrará y copiará todos los archivos con extensión ".txt" del directorio /home/Usuario y lo copiará al directorio /home/backup.
- **mkisofs /dev/cdrom > Archivo.iso**: creará una imagen ISO de un CD.
- **mkisofs /dev/cdrom | gzip > ArchivoISO.gz**: creará una imagen ISO comprimida de un CD.
- **cdrecord -v dev=/dev/cdrom Archivo.iso**: grabar en un CD una imagen ISO.
- **gzip -dc ArchivoISO.gz | cdrecord dev=/dev/cdrom -**: grabar en un CD una imagen ISO comprimida ya existente.

## Comandos para la gestión de redes

En la actualidad conectarnos a una red en Linux es una tarea bastante sencilla y transparente al usuario pero puede suceder que en algún momento necesites recurrir al terminal para configurar tus redes o diagnosticar qué es lo que está fallando en tu red.

Los siguientes comandos te ayudarán a realizar tareas básicas para **la gestión de redes en Linux**. Estos son los **comandos que deberías conocer**:

- **ifconfig eth0**: mostrará la configuración de una tarjeta de red Ethernet.
- **ifup eth0**: activará una interface eth0.
- **ifdown eth0**: deshabilitará una interface eth0.
- **ifconfig eth0 192.168.0.1 netmask 255.255.255.0**: configurará una dirección IP.
- **dhclient eth0**: activará la interface eth0 en modo dhcp.
- **route -n**: mostrará tabla de recorrido.
- **route add -net 0/0 gw IP\_PuertaAcceso**: configurará una puerta de acceso predeterminada.
- **route del 0/0 gw IP\_PuertaAcceso**: eliminará la ruta estática.
- **hostname**: mostrará el nombre del host del sistema.
- **ip link show**: mostrará el estado de enlace de todas las interfaces.
- **mii-tool eth0**: mostrará el estado de enlace de 'eth0'.
- **ethtool eth0**: mostrar las estadísticas de tarjeta de red 'eth0'.
- **netstat -tup**: mostrar todas las conexiones de red activas y sus PID.
- **netstat -tupl**: mostrar todos los servicios de escucha de red en el sistema y sus PID.
- **iwlist scan**: mostrar las redes inalámbricas.
- **iwconfig eth1**: mostrar la configuración de una tarjeta de red inalámbrica

Y hasta aquí la chuleta de comandos para GNU/Linux. No he querido hacer de esta publicación algo demasiado extensa así que mi intención es publicar otras chuletas más adelante y así ayudar a hacernos la vida más fácil a todos los usuarios. ¿Te ha gustado? No olvides compartir esta publicación.



## Licencia:

### **Reconocimiento-NoComercial-CompartirIgual CC BY-NC-SA**

Atribución-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0)

Este es un resumen legible por humanos (y no un sustituto) de la licencia. Advertencia.

#### **Usted es libre de:**

Compartir — copiar y redistribuir el material en cualquier medio o formato

Adaptar — remezclar, transformar y construir a partir del material

La licenciante no puede revocar estas libertades en tanto usted siga los términos de la licencia

#### **Bajo los siguientes términos:**

Atribución — Usted debe dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante.

NoComercial — Usted no puede hacer uso del material con propósitos comerciales.

CompartirIgual — Si remezcla, transforma o crea a partir del material, debe distribuir su contribución bajo la misma licencia del original.

No hay restricciones adicionales — No puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia.

#### **Avisos:**

No tiene que cumplir con la licencia para elementos del material en el dominio público o cuando su uso esté permitido por una excepción o limitación aplicable.

No se dan garantías. La licencia podría no darle todos los permisos que necesita para el uso que tenga previsto. Por ejemplo, otros derechos como publicidad, privacidad, o derechos morales pueden limitar la forma en que utilice el material.