

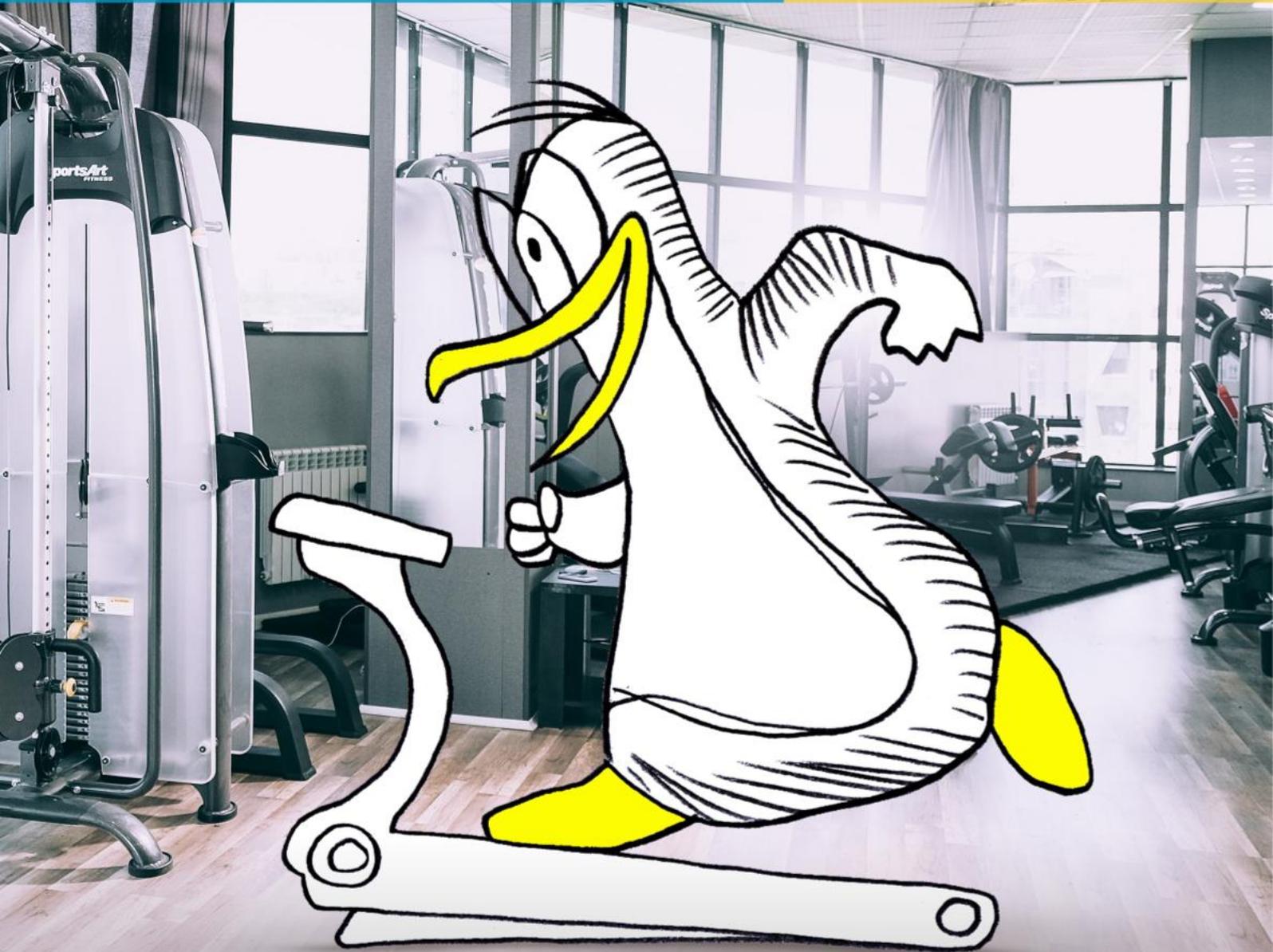
Visita nuestro sitio SoloLinux.es

# MAGAZINE SOLO LINUX

Nº  
26

Tu revista, la revista de tod@s

MARZO 2021



Deshabilitar **Network Manager** en Linux

Instalar **Debian 10 live+nonfree** en imágenes

Elegir un kernel para **CentOS 8**

Modificar el **hostname** en Linux

Uso del comando **getent** en Linux

Listar discos conectados con **lsblk** en Linux

MANUALES, SCRIPTS, SOFTWARE, HARDWARE, DISTROS LINUX, SEGURIDAD, REDES Y MUCHO MAS EN LA WEB...

## Bienvenidos a la Revista SOLOLINUX

Os presento el numero 26 de la revista SoLoLinux. Desde SOLOLINUX esperamos que os guste este numero.

Igual que en números anteriores nos gustaría animar a todos nuestros lectores para que nos envíen sus **opiniones sobre el Software Libre o sobre GNU/Linux**, pueden enviarlo a [adrian@sololinux.es](mailto:adrian@sololinux.es), con ello queremos proponer que cada mes se publicada una o varias de esas opiniones sobre lo mencionado en la nueva sección de la revista *OPINIÓN DEL LECTOR*. **Queremos saber la opinión de todos**. Se intentara incluir el máximo de opiniones en cada numero, pero si no sale la tuya este mes no desesperes, al siguiente podría tener un hueco en la revista. **ANÍMENSE Y ENVÍEN SUS OPINIONES**. Gracias.

Al igual que lo anteriormente mencionado, nos gustaría promover un espacio en la revista sobre los eventos de Software Libre y GNU/Linux en todo el mundo. Los organizadores de estos eventos pueden ponerse en contacto con migo a través de correo electrónico, [adrian@sololinux.es](mailto:adrian@sololinux.es)

Sin mas **quiero agradecer a todos** los que hacéis posible que esta revista siga adelante. Personalmente agradezco a Sergio todo su trabajo en la multitud de artículos que realiza a lo largo del mes para que esta revista pueda tener suficiente información mes a mes.

### Gracias a TOD@S

Compartan esta revista en sus redes sociales o web. Revista digital **SOLOLINUX MAGAZINE**. Tu revista, la revista de todos.

Síguenos en  
las Redes:



La revista  
SOLOLINUX esta  
realizada con  
Libre Office  
Impress 71.1.2

# www.sololinux.es

Esta revista es de **distribución gratuita**, si lo consideras oportuno puedes ponerle precio.

Tu también puedes ayudar, contamos con la posibilidad de hacer donaciones para la REVISTA, de manera muy simple a través de **PAYPAL**

AYUDANOS A SEGUIR  
CRECIENDO



### Editorial

- **Adrián Almenar** (Edición y diseño de la revista)  
- e-mail: [adrian@sololinux.es](mailto:adrian@sololinux.es)

### Redacción

- **Sergio G. B.** (Administrador y redactor artículos SoloLinux)  
- e-mail: [info@sololinux.es](mailto:info@sololinux.es)
- **Henry G. R.** (Redactor artículos SoloWordPress)  
- e-mail: [info@solowordpress.es](mailto:info@solowordpress.es)

### Diseño Portada

- **Karina Fernández**  
[@karyfernandez.design](https://www.instagram.com/karyfernandez.design)

### Publicidad

Quieres poner publicidad en la revista, ahora puedes hacerlo de forma muy simple, llegando a todo el mundo con esta revista digital de software libre y GNU/Linux en ESPAÑOL

**CON SOLOLINUX MULTIPLICARAS  
TUS CLIENTES**

Para mayor información escribe un e-mail a: [adrian@sololinux.es](mailto:adrian@sololinux.es)

### Contacto

Para cualquier consulta sobre la revista, publicidad o colaboraciones escribir un **email** a:

- [adrian@sololinux.es](mailto:adrian@sololinux.es)



Este obra se publica bajo una licencia de Creative Commons Atribución-CompartirIgual 4.0 Internacional (**CC BY-SA 4.0**)

## MANUALES



- 07. Deshabilitar Network Manager en Linux
- 12. Elegir un kernel para CentOS 8



- 16. Uso del comando getent en Linux
- 24. Buscar y reparar paquetes rotos en Linux
- 27. Como instalar LAMP en Alma Linux
- 38. Uso del comando find en Linux
- 41. Uso del comando awk en Linux



## HARDWARE



- 14. Modificar el hostname en Linux
- 20. Listar discos conectados con lsblk en Linux
- 48. Reparar un USB dañado sin perder datos



Reparar un USB dañado  
sin perder datos

## DISTROS LINUX



- 09. Instalar Debian 10 live+nonfree en imágenes



- 22. Los 5 mejores Linux para instalar en Windows WSL
- 26. Convertir CentOS 8 en Alma Linux



- 55. Parrot Security 4.11 – Listo para su descarga

## NOTICIAS



- 23. Un incendio destruye el datacenter de OVH en Estrasburgo



www.sololinux.es

## SEGURIDAD

- 25. Deshabilitar el inicio de sesión ssh de un usuario
- 30. Configurar un servidor OpenVPN en Debian 10



- 33. Configurar un cliente OpenVPN en Debian 10



- 34. Iniciar sesión ssh sin password en Linux



## SOFTWARE



- 46. Limpiar linux con Czkawka de archivos innecesarios
- 50. Comprimir imágenes con Curtail Images Compressor
- 53. Instalar GNS3 en Ubuntu y derivados



## REDES



- 51. 2.4 GHz vs 5 GHz – En que se diferencian
- 56. Descargar Cisco Packet Tracer 8.0.0 y versiones anteriores



### Publicidad:

Quieres poner publicidad en la revista, ahora puedes hacerlo de forma muy simple, llegando a todo el mundo con esta revista digital gratuita de software libre y GNU/Linux en ESPAÑOL

CON **SOLOLINUX** MULTIPLICARAS TUS CLIENTES

Para mayor información escribe un email a: [adrian@sololinux.es](mailto:adrian@sololinux.es)

[www.sololinux.es](http://www.sololinux.es)





# CUMPLIMOS 10 AÑOS

Este año cumplimos 10 años. Gracias a todos los que venís confiando en nosotros para dar el salto a Linux (y a los que habéis confiando en nosotros, con el salto ya dado).

En 2011 creamos VANT y antes de acabar el año lanzamos nuestro primer portátil: El primer VANT MOOVE, con una CPU i3-2330m, hasta 8GB de RAM, 2.3Kg de peso y Ubuntu 11.04

10 años hasta llegar a hoy, con una gama completa de sobremesas y portátiles para cubrir las necesidades de diferentes perfiles de usuario y con el ultrabook EDGE2 como tope de gama, con 1.1Kg de peso, procesadores i7-11065u, hasta 40GB de RAM y batería como para dejar tranquilamente el cargador en casa. Y con Ubuntu 20.04.2, o Linux Mint, o Debian, o KDE Neon, o...

Muchas cosas han cambiado en estos 10 años, salvo la ilusión con la que trabajamos, para convertir vuestra confianza en satisfacción, por muchos años.



la gama más completa de ordenadores linuxeros

descúbrenos en [www.vantpc.es](http://www.vantpc.es)

[@vantpc](https://twitter.com/vantpc) [f vant.pc](https://facebook.com/vant.pc) [i vantpc\\_es](https://instagram.com/vantpc_es) [t.me/vantpc](https://t.me/vantpc)



**INSTITUTO  
LINUX**



**FABIÁN AMPALIO**  
Coordinador Gral.



**GABRIEL CANEPA**  
Tutor



**ALEJANDRO DRABENCHE**  
Tutor



**JOAQUÍN AMPALIO**  
Tutor



**FRANCISCO SANDALINAS**  
Tutor



**DIEGO NOBILE**  
Tutor



# TÉCNICO LINUX 2021

**CURSO LINUX SYSTEM ADMINISTRATOR**

+

**CERTIFICACIÓN UTN-FRD**

+

**22 CLASES MAGISTRALES ON LINE ¡DE REGALO!**

+

**WORKSHOPS LPIC 1 (101-102) ¡DE REGALO!**

+

**INSCRIPCIÓN ABIERTA 2021**

**INICIA EN  
ABRIL**



+34 669 584 870



[www.carreralinux.es](http://www.carreralinux.es)

## Deshabilitar Network Manager en Linux

**Network Manager**, es una herramienta que nos ayuda a configurar los demonios relacionados con la red. Esta aplicación viene incluida de forma predeterminada, en la mayoría de los entornos de escritorio modernos.

La particularidad de **Network Manager** es, que ofrece una configuración automática para la mayoría de interfaces de red, incluyendo las cableadas e inalámbricas, así como las conexiones VPN, banda ancha móvil y las **Bluetooth**. **Network Manager** es inteligente, cuando tienes varias conexiones habilitadas cambia automáticamente a la red más potente, e informa a otras aplicaciones relacionadas de la actualización.



A pesar de los beneficios que aporta este manejador de demonios de red, es posible que no quieras usarlo. Los motivos pueden ser variados, por ejemplo: **Network Manager** no es la mejor opción para configurar redes en modo avanzado, como el **multi-homing** a más de una conexión cableada, el reenvío de tráfico de red, la configuración de puentes o alias en **Linux**, etc.

En este artículo, vemos cómo **deshabilitar Network Manager** en las **distribuciones linux** más utilizadas por los usuarios. **OJO!!!** Y el que avisa no es traidor, recuerda que si pretendes deshabilitar **Network Manager** en un host remoto (**vps o servidor**), perderás la conectividad y no podrás reconectar de forma remota.



Por ejemplo...

### Deshabilitar Network Manager en linux

Comenzamos identificando las interfaces controladas por Network Manager.

```
nmcli dev status
```

```
root@SoloLinux:~# nmcli dev status
DEVICE TYPE STATE CONNECTION
eth0 ethernet sin gestión --
eth1 ethernet conectado SoloLinux
```

Ahora vemos como **deshabilitar Network Manager**, en tu distribución linux.

#### En Ubuntu, Linux Mint y derivados

```
sudo systemctl stop NetworkManager.service
sudo systemctl disable NetworkManager.service
# o
sudo stop network-manager
echo "manual" | sudo tee /etc/init/network-manager.override
```

#### En Debian

```
# Debian 8/9/10 y superior
sudo systemctl stop NetworkManager.service
sudo systemctl disable NetworkManager.service
# Debian 7 y anteriores
sudo /etc/init.d/network-manager stop
sudo update-rc.d network-manager remove
```

#### En Fedora, CentOS, RHEL y derivados

```
# En Fedora, CentOS 7, RHEL 7 y superiores
sudo systemctl stop NetworkManager.service
sudo systemctl disable NetworkManager.service
# En CentOS 6, RHEL 6 y anteriores
sudo service NetworkManager stop
sudo chkconfig NetworkManager off
```

#### En Arch Linux, Manjaro y derivados

```
sudo systemctl stop NetworkManager.service
sudo systemctl disable NetworkManager.service
```

```
root@sololinux:~# sudo systemctl stop NetworkManager.service
root@sololinux:~# sudo systemctl disable NetworkManager.service
Removed /etc/systemd/system/multi-user.target.wants/NetworkManager.service.
Removed /etc/systemd/system/dbus-org.freedesktop.nm-dispatcher.service.
root@sololinux:~# sudo systemctl start NetworkManager.service
root@sololinux:~# sudo systemctl enable NetworkManager.service
Created symlink /etc/systemd/system/multi-user.target.wants/NetworkManager.service → /lib/systemd/system/NetworkManager.service.
Created symlink /etc/systemd/system/dbus-org.freedesktop.nm-dispatcher.service → /lib/systemd/system/NetworkManager-dispatcher.service.
root@sololinux:~#
```

www.sololinux.es

## Verificar el servicio Network Manager

Ahora verificamos si el servicio corre o no. Ejecuta el siguiente comando.

```
systemctl status NetworkManager.service
```

En el caso que lo tengamos deshabilitado, veremos algo parecido a...

```
root@sololinux:~# sudo systemctl stop NetworkManager.service
root@sololinux:~# sudo systemctl disable NetworkManager.service
Removed /etc/systemd/system/multi-user.target.wants/NetworkManager.service.
Removed /etc/systemd/system/dbus-org.freedesktop.nm-dispatcher.service.
root@sololinux:~# systemctl status NetworkManager.service
● NetworkManager.service - Network Manager
   Loaded: loaded (/lib/systemd/system/NetworkManager.service; disabled; vendor preset: enabled)
   Active: inactive (dead) since Tue 2021-03-02 12:58:46 EET; 23s ago
     Docs: man:NetworkManager(8)
   Main PID: 15021 (code=exited, status=0/SUCCESS)
  mar 02 12:58:45 sololinux systemd[1]: Stopping Network Manager...
  mar 02 12:58:45 sololinux NetworkManager[15021]: <info> [1614682725.9823] caught SIGTERM, shutting down normally.
  mar 02 12:58:45 sololinux NetworkManager[15021]: <info> [1614682725.9971] device (wlo1): state change: activated -> deactivating (reas
  mar 02 12:58:45 sololinux NetworkManager[15021]: <info> [1614682725.9986] manager: NetworkManager state is now DISCONNECTING
  mar 02 12:58:46 sololinux NetworkManager[15021]: <info> [1614682726.0100] device (wlo1): state change: deactivating -> unmanaged (reas
  mar 02 12:58:46 sololinux NetworkManager[15021]: <info> [1614682726.0282] dhcp4 (wlo1): canceled DHCP transaction, DHCP client pid 150
  mar 02 12:58:46 sololinux NetworkManager[15021]: <info> [1614682726.0292] dhcp4 (wlo1): state changed bound -> done
  mar 02 12:58:46 sololinux NetworkManager[15021]: <info> [1614682726.0777] manager: NetworkManager state is now DISCONNECTED
  mar 02 12:58:46 sololinux NetworkManager[15021]: <info> [1614682726.0963] exiting (success)
  mar 02 12:58:46 sololinux systemd[1]: Stopped Network Manager.
```

Si por el contrario está habilitado.

```
root@sololinux:~# systemctl status NetworkManager.service
● NetworkManager.service - Network Manager
   Loaded: loaded (/lib/systemd/system/NetworkManager.service; enabled; vendor p
   Active: active (running) since Tue 2021-03-02 12:43:19 EET; 10min ago
     Docs: man:NetworkManager(8)
   Main PID: 15021 (NetworkManager)
    Tasks: 4 (limit: 4521)
   CGroup: /system.slice/NetworkManager.service
           └─15021 /usr/sbin/NetworkManager --no-daemon
             └─15028 /sbin/dhclient -d -q -sf /usr/lib/NetworkManager/nm-dhcp-help

www.sololinux.es

mar 02 12:43:21 sololinux NetworkManager[15021]: <info> [1614681801.5799] devic
mar 02 12:43:21 sololinux NetworkManager[15021]: <info> [1614681801.5812] devic
mar 02 12:43:21 sololinux NetworkManager[15021]: <info> [1614681801.5878] devic
mar 02 12:43:21 sololinux NetworkManager[15021]: <info> [1614681801.5885] manag
mar 02 12:43:21 sololinux NetworkManager[15021]: <info> [1614681801.6057] manag
mar 02 12:43:21 sololinux NetworkManager[15021]: <info> [1614681801.6059] polic
mar 02 12:43:21 sololinux NetworkManager[15021]: <info> [1614681801.6071] devic
```

**Nota:** En un próximo artículo, veremos como **deshabilitar o habilitar Network Manager** en una interfaz en particular.

## Instalar Debian 10 live+nonfree en imágenes

Debian es una de las mejores **distribuciones linux** existentes, eso no cabe ninguna duda. A pesar de lo dicho, tampoco es una distribución fácil para los **recién llegados a linux**, sobre todo por los controladores que trae por defecto.

De forma no oficial, Debian mantiene una serie de paquetes e ISO's, que integran controladores y firmware privativos. Supongo que los más puristas se echaran las manos a la cabeza, pero soy de los que piensan que las herramientas que tenemos a nuestra disposición, son para usarlas y facilitarnos las tareas, no para entorpecernos aún más.

Si no eres un usuario avanzado y alguna vez has tenido problemas al instalar Debian, prueba las **versiones nonfree**, te harán la vida más fácil y, posiblemente terminen con tus quebraderos de cabeza. En el artículo de hoy, vemos como **instalar Debian 10 live+nonfree** con imágenes de ejemplo, verás que sencillo es el proceso.



## Instalar Debian 10 live+nonfree en imágenes

Comenzaremos descargando la live+nonfree de Debian 10, después procedemos a su instalación.

### Descargar Debian 10 live+nonfree

Desde los siguientes **enlaces torrent**, puedes descargar la versión que más se ajuste a tus necesidades de Debian 10-8.8 live+nonfree amd64. Es la última versión estable, publicada hasta la fecha de hoy.

- [Debian-live-10.8.0-amd64-cinnamon + nonfree.iso](#)
- [Debian-live-10.8.0-amd64-gnome + nonfree.iso](#)
- [Debian-live-10.8.0-amd64-kde + nonfree.iso](#)
- [Debian-live-10.8.0-amd64-lxde + nonfree.iso](#)
- [Debian-live-10.8.0-amd64-lxqt + nonfree.iso](#)
- [Debian-live-10.8.0-amd64-mate + nonfree.iso](#)
- [Debian-live-10.8.0-amd64-standard + nonfree.iso](#)
- [Debian-live-10.8.0-amd64-xfce + nonfree.iso](#)

Una vez descargada la iso, creas un **live USB** e inicias el sistema con él.

# www.sololinux.es

## Instalar Debian 10 live+nonfree

En la primera pantalla que nos aparece, seleccionamos la localización.



Ahora elige tu lenguaje preferido.



Se inicia Debian 10 en modo live. Si es necesario te conectas a internet y, pulsas en **instalar Debian 10 nonfree**.



En la primera pantalla, el **instalador calamares** nos da la bienvenida. Como anteriormente ya seleccionamos nuestro idioma por defecto, tan solo debes pulsar en siguiente.



Es importante que selecciones correctamente la ubicación donde te encuentras, de ello depende disponer del horario correcto.



Lo siguiente es el teclado, tu idioma y sus posibles variantes.



A continuación seleccionamos el dispositivo donde vamos a instalar Debian 10. También si quieres personalizar las particiones o no. Si no eres un usuario experto, te recomiendo marcar «**Borrar disco**» que borra todos los datos y creará las particiones de forma automática.



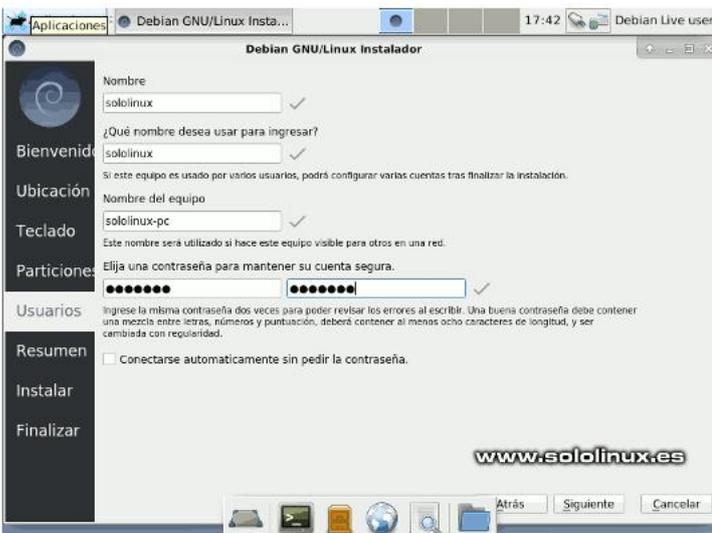
La instalación puede demorar un rato, todo depende de tu máquina. Sé paciente.



Una vez concluya la instalación de **Debian 10 nonfree**, pulsa en «**Hecho**» para reiniciar el sistema.

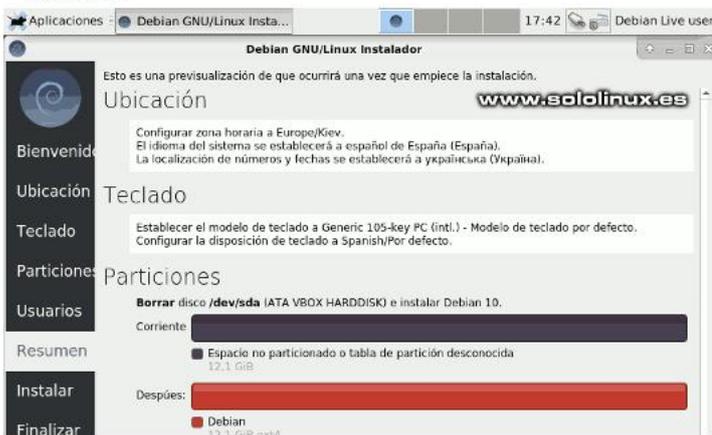


Introduce los datos que solicita. **Nombre – Equipo – Password.**



Bien, para concluir el artículo no hace falta indicaros que para iniciar Debian, tengas seleccionado «**Debian GNU/Linux**». Solo te falta configurar Debian según tus preferencias, suerte.

Antes de comenzar la instalación, revisa que todo esté correcto.



## Elegir un kernel para CentOS 8

El **Kernel Linux** es el núcleo de todas las distribuciones GNU/Linux y, **CentOS** es uno de los sistemas, a pesar de que considero que le queda poco de vida, gracias a la **decisión de Red Hat**. Como buen derivado de **Red Hat Enterprise**, sigue sus mismas actualizaciones de kernel.

Esto es bueno en servidores que requieren de cierta estabilidad, pero puede suponer un problema en máquinas modernas por el tema de hardware. Por ejemplo, el último kernel estable lanzado para **CentOS 7**, es el **3.10.0-1160.15.2.el7.x86\_64**, y el último kernel estable lanzado para **CentOS 8**, es el **4.18.0**. Creo que es inadmisibles que una distro como CentOS 7, cuyo fin de actualizaciones de mantenimiento se calcula para el 2024, continúe usando un kernel obsoleto e inadmisibles en una buena **distribución linux**.



Como siempre recomendamos desde **SOLOLINUX**, si algo funciona bien no lo toques. Pero si tienes problemas de reconocimiento de hardware, es evidente que debes buscar una solución. En este artículo, veremos como listar los kernel actualizados que sean compatibles con tu versión de CentOS, elegir un kernel e instalarlo en tu sistema. Nosotros usamos como ejemplo un CentOS 8, pero también explicamos el proceso para CentOS 7.

### End of Lifetime (EOL) Dates

	CentOS-6	CentOS-7	CentOS-8
Full Updates <sup>1</sup>	May 10th, 2017	Q4 2020	December 31, 2021
Maintenance Updates <sup>2</sup>	November 30th, 2020	June 30th, 2024	December 31, 2021

[www.sololinux.es](http://www.sololinux.es)

## Elegir un kernel para CentOS 8

Para elegir un kernel y lograr el nuestro objetivo, necesitamos el repositorio conocido como **ELRepo**. Este Proyecto se centra exclusivamente en paquetes relacionados con el hardware y, sus actualizaciones. Instalamos el nuevo **repositorio**.

### En CentOS 8:

```
sudo dnf install https://www.elrepo.org/elrepo-release-8.el8.elrepo.noarch.rpm
sudo rpm --import https://www.elrepo.org/RPM-GPG-KEY-elrepo.org
```

### En CentOS 7:

```
sudo yum install https://www.elrepo.org/elrepo-release-7.el7.elrepo.noarch.rpm
sudo rpm --import https://www.elrepo.org/RPM-GPG-KEY-elrepo.org
```

```
[root@SoloLinux ~]# dnf install https://www.elrepo.org/elrepo-release-8.el8.elrepo.noarch.rpm
Ultima comprobación de caducidad de metadatos hecha hace 0:06:26, el jue 04 mar 2021 13:07:40 CET.
elrepo-release-8.el8.elrepo.noarch.rpm      20 kB/s | 13 kB      00:00
Dependencias resueltas.
=====
Paquete      Arq.      Versión      Repositorio      Tam.
=====
Instalando:
elrepo-release  noarch    8.2-1.el8.elrepo  @commandline    13 k
Resumen de la transacción
=====
Instalar 1 Paquete

Tamaño total: 13 k
Tamaño instalado: 5.0 k
¿Está de acuerdo [s/N]?: █
```

[www.sololinux.es](http://www.sololinux.es)

Actualizamos.

```
# CentOS 8
sudo dnf update
# CentOS 7
sudo yum update
```

Listamos los kernel disponibles.

```
# En CentOS 8
sudo dnf list available --disablerepo='*' --enablerepo=elrepo-kernel
# En CentOS 7
sudo yum list available --disablerepo='*' --enablerepo=elrepo-kernel
```

Como puedes ver en la salida de ejemplo nos aparecen dos kernel, el último lanzamiento y el último LTS.

```
[root@SoloLinux ~]# sudo dnf list available --disablerepo='*' --enablerepo=elrepo-kernel
Última comprobación de caducidad de metadatos hecha hace 0:09:05, el jue 04 mar 2021 13:27:41 CET.
Paquetes disponibles
bpftool.x86_64
5.11.3-1.el8.elrepo kernel-lt.x86_64 elrepo-kernel
5.4.102-1.el8.elrepo kernel-lt-core.x86_64 elrepo-kernel
5.4.102-1.el8.elrepo kernel-lt-devel.x86_64 elrepo-kernel
5.4.102-1.el8.elrepo kernel-lt-doc.noarch elrepo-kernel
5.4.102-1.el8.elrepo kernel-lt-headers.x86_64 elrepo-kernel
5.4.102-1.el8.elrepo kernel-lt-modules.x86_64 elrepo-kernel
5.4.102-1.el8.elrepo kernel-lt-modules-extra.x86_64 elrepo-kernel
5.4.102-1.el8.elrepo kernel-lt-tools.x86_64 elrepo-kernel
5.4.102-1.el8.elrepo kernel-lt-tools-libs.x86_64 elrepo-kernel
5.4.102-1.el8.elrepo kernel-lt-tools-libs-devel.x86_64 elrepo-kernel
5.4.102-1.el8.elrepo kernel-ml.x86_64 elrepo-kernel
5.11.3-1.el8.elrepo kernel-ml-core.x86_64 elrepo-kernel
5.11.3-1.el8.elrepo kernel-ml-devel.x86_64 elrepo-kernel
5.11.3-1.el8.elrepo kernel-ml-doc.noarch elrepo-kernel
5.11.3-1.el8.elrepo kernel-ml-headers.x86_64 elrepo-kernel
5.11.3-1.el8.elrepo kernel-ml-modules.x86_64 elrepo-kernel
5.11.3-1.el8.elrepo kernel-ml-modules-extra.x86_64 elrepo-kernel
5.11.3-1.el8.elrepo kernel-ml-tools.x86_64 elrepo-kernel
5.11.3-1.el8.elrepo kernel-ml-tools-libs.x86_64 elrepo-kernel
5.11.3-1.el8.elrepo kernel-ml-tools-libs-devel.x86_64 elrepo-kernel
5.11.3-1.el8.elrepo perf.x86_64 elrepo-kernel
5.11.3-1.el8.elrepo python3-perf.x86_64 elrepo-kernel
5.11.3-1.el8.elrepo
[root@SoloLinux ~]#
```

Mi recomendación es que instales el LTS, que ELRepo denomina lt. Es tan simple como ejecutar el siguiente comando.

```
# CentOS 8
sudo dnf --enablerepo=elrepo-kernel install kernel-lt
# CentOS 7
sudo yum --enablerepo=elrepo-kernel install kernel-lt
```

```
root@SoloLinux ~]# sudo dnf --enablerepo=elrepo-kernel install kernel-lt
Última comprobación de caducidad de metadatos hecha hace 0:19:12, el jue 04 mar 2021 13:27:41 CET.
Dependencias resueltas.
-----
Paquete Arquitectura Versión Repositorio Tam.
-----
Instalando:
kernel-lt x86_64 5.4.102-1.el8.elrepo elrepo-kernel 55 k
Instalando dependencias:
kernel-lt-core x86_64 5.4.102-1.el8.elrepo elrepo-kernel 28 M
kernel-lt-modules x86_64 5.4.102-1.el8.elrepo elrepo-kernel 24 M
linux-firmware noarch 20200619-101.git3890db36.el8_3.noarch booooo 181 k
Resumen de la transacción
-----
Instalar 4 Paquetes
-----
Español total de la descarga: 154 M
Pasado (incluyendo): 499 M
Fecha de actualiz. (E02):
www.sololinux.es
```

```
Descargando paquetes:
(1/4): kernel-lt-5.4.102-1.el8.elrepo.x86_64.rpm 854 kB/s | 55 kB 00:00
(2/4): kernel-lt-modules-5.4.102-1.el8.elrepo.x86_64.rpm 45 MB/s | 24 MB 00:00
(3/4): kernel-lt-core-5.4.102-1.el8.elrepo.x86_64.rpm 37 MB/s | 28 MB 00:00
(4/4): linux-firmware-20200619-101.git3890db36.el8_3.noarch.rpm 81 MB/s | 101 MB 00:01
-----
Total 77 MB/s | 154 MB 00:02
Ejecutando verificación de operación
Verificación de operación exitosa.
Ejecutando prueba de operaciones
Prueba de operación exitosa.
Ejecutando operación
Preparando : 1/1
Instalando : linux-firmware-20200619-101.git3890db36.el8_3.noarch 1/4
Instalando : kernel-lt-core-5.4.102-1.el8.elrepo.x86_64 2/4
Ejecutando scriptlet: kernel-lt-core-5.4.102-1.el8.elrepo.x86_64 2/4
Instalando : kernel-lt-modules-5.4.102-1.el8.elrepo.x86_64 3/4
Ejecutando scriptlet: kernel-lt-modules-5.4.102-1.el8.elrepo.x86_64 3/4
Instalando : kernel-lt-5.4.102-1.el8.elrepo.x86_64 4/4
Ejecutando scriptlet: kernel-lt-core-5.4.102-1.el8.elrepo.x86_64 4/4
Failed to add dependency on unit, unit systemd-ask-password-plymouth.service does not exist.
Ejecutando scriptlet: kernel-lt-5.4.102-1.el8.elrepo.x86_64 4/4
Verificando : linux-firmware-20200619-101.git3890db36.el8_3.noarch 1/4
Verificando : kernel-lt-5.4.102-1.el8.elrepo.x86_64 2/4
Verificando : kernel-lt-core-5.4.102-1.el8.elrepo.x86_64 3/4
Verificando : kernel-lt-modules-5.4.102-1.el8.elrepo.x86_64 4/4
Instalado:
kernel-lt-5.4.102-1.el8.elrepo.x86_64
kernel-lt-core-5.4.102-1.el8.elrepo.x86_64
kernel-lt-modules-5.4.102-1.el8.elrepo.x86_64
linux-firmware-20200619-101.git3890db36.el8_3.noarch
```

Solo falta reiniciar el sistema.

```
sudo reboot
```

Ahora puedes seleccionar el nuevo kernel durante el arranque.

**ANOTACIÓN DE UN LECTOR EN EL CHAT**  
Tal y como puntualiza **OMAR** en el **CHAT DE SOLOLINIX EN TELEGRAM**, podemos usar el siguiente comando para ver el cambio del **Kernel**

```
uname -r
```

Canales de Telegram:

[Canal SoloLinux – Canal SoloWordpress](#)

Espero que este artículo te sea de utilidad, puedes colaborar con el simple gesto de compartir nuestros artículos en tu sitio web, blog, foro o redes sociales.

[Chat de SoloLinux en Telegram](#)

## Modificar el hostname en Linux

El **hostname**, también conocido como **nombre de host en linux**, es el apelativo que designamos a una máquina o sistema, con el cual se identifica en una red. No existe ninguna especificación, pero si excepciones para el **hostname**, puede ser una cadena de caracteres alfanuméricos o un nombre complejo.

Cada nombre de host (hostname) debe tener una dirección IP en formato numérico, este identificará al equipo en la red. En este artículo veremos como **modificar el hostname**, en **CentOS, Debian, Arch Linux** y todos sus derivados.



## Modificar el hostname en Linux

Antes de comenzar, debes saber que las tres fórmulas disponibles para modificar el hostname, puede intercambiarse por otra dependiendo del tipo de [distribución linux](#) derivada.

### En CentOS, RHEL y derivados

Identificamos el hostame con el siguiente comando.

```
hostnamectl
```

Ejemplo de salida (SoloLinux-demo).

```
[root@SoloLinux ~]# hostnamectl
  Static hostname: SoloLinux
    Icon name: computer-container
    Chassis: container
  Machine ID: 93543a9cd4d548fcb3dd352e355def45
  Boot ID: a1314281c5cc44c5b7b7965f84c1cd6d
  Virtualization: openvz
  Operating System: CentOS Linux 8
  CPE OS Name: cpe:/o:centos:centos:8
  Kernel: Linux 5.4.0
  Architecture: x86_64
```

Cambiar el hostname, es tan simple como ejecutar este comando (inserta tu nombre deseado).

```
hostnamectl set-hostname nuevo-hostname
```

Conviene reiniciar el sistema y verificar el cambio.

```
sudo reboot
hostnamectl
```

## En Debian, Ubuntu y derivados

Comenzamos identificando el hostname actual.

```
hostname
```

Nos dice el nombre de la máquina, en nuestro ejemplo «SoloLinux».

```
root@SoloLinux:~# hostname
SoloLinux
root@SoloLinux:~# www.sololinux.es
```

Si quieres **cambiar el hostname**, ejecuta el siguiente comando (inserta tu nombre deseado).

```
sudo hostname nuevo-hostname
```

En Debian y derivados no es suficiente con el comando anterior, debemos editar un par de archivos, **/etc/hostname** y **/etc/hosts**.

```
sudo nano /etc/hostname
```

Verás el nombre actual, lo modificas por el que insertaste en el paso anterior. Guardas y cierras el editor.

```
GNU nano 3.2 /etc/hostname
```

```
SoloLinux
www.sololinux.es
```

Ahora modificamos el archivo **/etc/hosts**

```
sudo nano /etc/hosts
```

Dependiendo si es un sistema de escritorio, servidor remoto, o incluso el tipo de derivado; lo que aparece en pantalla puede variar de formato. Lo único que tienes que hacer independientemente de la opción, es sustituir el nombre del hostname actual por el nuevo.

Vemos dos ejemplos con diferente formato.

```
127.0.0.1 localhost
127.0.1.1 SoloLinux <-----nuevo hostname
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

```
127.0.0.1 localhost
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
# Auto-generated hostname. Please do not remove this comment.
187.189.57.75 SoloLinux <----- nuevo hostname
```

Guarda el archivo y cierra el editor. Continuamos reiniciando el sistema y, verificando el nuevo hostname.

```
sudo reboot
hostname
```

## En Arch Linux, Manjaro y derivados

Verificamos el **hostname** de la máquina.

```
hostnamectl
```

En nuestro ejemplo «manjaro».

```
Terminal - manjaro@manjaro:~/Desktop
File Edit View Terminal Tabs Help
[manjaro@manjaro Desktop]$ hostnamectl
Static hostname: manjaro
Icon name: computer-vm
Chassis: vm
Machine ID: f99929d76c5bf7593a4a112160421280
Boot ID: 8ecb9087214c4a98b6089d76653cc2e4
Virtualization: oracle
Operating System: Manjaro Linux
Kernel: Linux 5.9.16-1-MANJARO
Architecture: x86-64
[manjaro@manjaro Desktop]$ www.sololinux.es
```

Puedes **modificar el hostname** con este comando.

```
sudo hostnamectl set-hostname nuevo-hostname
```

Ahora editamos el archivo **/etc/hosts**.

```
sudo nano /etc/hosts
```

Aparece algo similar a...

```
127.0.0.1 localhost
::1 localhost
127.0.1.1 manjaro.localdomain manjaro
```

Sustituye el nombre original, por el que tu quieras.

```
127.0.0.1 localhost
::1 localhost
127.0.1.1 nuevo-hostname.localdomain nuevo-hostname
```

Guarda el archivo y cierra el editor. Para concluir reiniciamos el sistema y, verificamos el **nuevo hostname**.

```
sudo reboot
hostnamectl
```

Canales de Telegram:

[Canal SoloLinux](#) – [Canal SoloWordPress](#)

Espero que este artículo te sea de utilidad, puedes colaborar con el simple gesto de compartir nuestros artículos en tu sitio web, blog, foro o redes sociales.

[Chat de SoloLinux en Telegram](#)

# www.sololinux.es

## Uso del Comando getent en Linux

El **comando getent**, es una herramienta para Linux que nos ayuda a obtener las entradas de algunos archivos de texto importantes del sistema, más conocidos como bases de datos. Estos archivos pertenecen a lo que se conoce como bibliotecas **Name Service Switch**, que están configuradas en el archivo **/etc/nsswitch.conf**.

Los archivos de **base de datos**, deben ser admitidos por la biblioteca GNU C. A continuación enumeramos todos los posibles.

- **ahost**
- **ahostsv4**
- **ahostsv6**
- **alias**
- **ethers**
- **group**
- **gshadow**
- **hosts**
- **initgroups**
- **netgroup**
- **networks**
- **passwd**
- **protocols**
- **rpc**
- **services**
- **shadow**



Esto no quiere decir que tu sistema los tenga todos, solo los que necesita. Vemos un ejemplo.

```
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the 'glibc-doc-reference' and 'info' packages installed, try:
# `info libc "Name Service Switch"' for information about this file.
passwd:          compat systemd
group:           compat systemd
shadow:         compat
gshadow:        files
hosts:          files mdns4_minimal [NOTFOUND=return] dns myhostname
networks:       files
protocols:      db files
services:       db files
ethers:         db files
rpc:            db files
netgroup:       nis
```

```
syslog:x:102:106:/:/home/syslog:/usr/sbin/nologin
messagebus:x:103:107:/:/nonexistent:/usr/sbin/nologin
_apt:x:104:65534:/:/nonexistent:/usr/sbin/nologin
uuidd:x:105:106:/:/usr/sbin/nologin
cups-ppd:x:106:117:/:/usr/sbin/nologin
kerneld:x:107:3:Kernel Daemon,,:/usr/sbin/nologin
rtkit:x:108:110:rtkit:/usr/sbin/nologin
avahi-autoipd:x:109:114:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmuxd:x:110:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
systemd-coredump:x:111:117:systemd core dump processing,,,:/run/systemd:/usr/sbin/nologin
```

**getent command**

### Uso del comando getent en linux

#### Passwd

En nuestro primer ejemplo, listamos la base de datos passwd.

```
getent passwd

sergio@sololinux:~$ getent passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
```

## Group

También podemos enumerar los grupos con `group`.

```
getent group
```

```
sergio@sololinux:~$ getent group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,sergio
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
```

## Hosts

Al ejecutar `hosts`, podemos ver el `hostname` del sistema.

```
getent hosts
```

```
sergio@sololinux:~$ getent hosts
127.0.0.1      localhost
127.0.1.1     sololinux
127.0.0.1     ip6-localhost ip6-loopback
www.sololinux.es
```

## Protocols

Los protocolos admitidos por el sistema.

```
getent protocols
```

```
sergio@sololinux:~$ getent protocols
ip          0 IP
hopopt     0 HOPOPT
icmp       1 ICMP
igmp       2 IGMP
ggp        3 GGP
ipencap    4 IP-ENCAP
st         5 ST
tcp        6 TCP
egp        8 EGP
igp        9 IGP
pup       12 PUP
udp       17 UDP
hmp       20 HMP
xns-idp   22 XNS-IDP
rdp       27 RDP
```

## Buscar el servicio de un puerto

Ahora listamos los servicios del sistema, junto a su **número de puerto**.

```
getent services
```

```
sergio@sololinux:~$ getent services
tcpmux      1/tcp
echo        7/tcp
echo        7/udp
discard     9/tcp sink null
discard     9/udp sink null
sysstat     11/tcp users
daytime     13/tcp
daytime     13/udp
netstat     15/tcp
qotd        17/tcp quote
msp         18/tcp
msp         18/udp
chargen     19/tcp ttytst source
chargen     19/udp ttytst source
ftp-data    20/tcp
ftp         21/tcp
fsp         21/udp fspd
ssh         22/tcp
telnet      23/tcp
smtp        25/tcp mail
time        37/tcp timserver
time        37/udp timserver
www.sololinux.es
```

## Uso personalizado de getent

El comando getent, no se limita a listar de forma masiva. También podemos personalizar los listados, especificando la búsqueda deseada.

### Buscar el servicio de un puerto

En este ejemplo de uso, queremos averiguar que servicio usa el **puerto 22**.

```
getent services 22
```

En la salida observamos que el puerto 22, pertenece al servicio ssh.

```
sergio@sololinux:~$ getent services 22
ssh                22/tcp
```

### Buscar el puerto de un servicio

Ahora queremos saber el puerto que utiliza el servicio telnet.

```
getent services telnet
```

El resultado es correcto, es el puerto 23.

```
sergio@sololinux:~$ getent services telnet
telnet            23/tcp
```

### Que usuario ha iniciado sesión

Identificar quien inicio la sesión, es tarea sencilla. Ejecuta el siguiente comando.

```
getent passwd `whoami`
```

El **comando getent**, nos ofrece la información solicitada.

```
sergio@sololinux:~$ getent passwd `whoami`
sergio:x:1000:1000:sergio,,,:/home/sergio:/bin/bash
```

### Información sobre un usuario

También podemos solicitar la información, de un usuario en particular por su nombre. Por ejemplo...

```
getent passwd root
# 0
getent passwd usuario
```

```
sergio@sololinux:~$ getent passwd root www.sololinux.es
root:x:0:0:root:/root:/bin/bash
sergio@sololinux:~$ getent passwd sergio
sergio:x:1000:1000:sergio,,,:/home/sergio:/bin/bash
```

### Saber la ip de un dominio

Como vemos en este artículo, la info aportada por getent es muy extensa. No podía faltar, el como saber la ip de un dominio.

```
getent hosts dominio
```

```
sergio@sololinux:~$ getent hosts sololinux.es
163.172.52.4 sololinux.es
sergio@sololinux:~$ getent hosts google.com
2a00:1450:400d:804::200e google.com
sergio@sololinux:~$ getent hosts marca.es
193.110.128.199 marca.es
```

### Opciones del comando getent

El comando getent también nos ofrece unas interesantes y útiles opciones.

Opción	Uso
-i, --no-idn	Desactivar la codificación IDN
-s, --service=CONFIG	Configuración del servicio
--usage	Ofrecer un mensaje de uso
-V, --version	Versión de la herramienta
?, --help	Ayuda de la herramienta

Como ejemplo, identificamos la versión de la **herramienta getent**.

```
getent -V
```

En Ubuntu y derivados nos ofrece este resultado.

```
sergio@sololinux:~$ getent -V
getent (Ubuntu GLIBC 2.27-3ubuntu1.4) 2.27
Copyright (C) 2018 Free Software Foundation, Inc.
Esto es software libre; vea el código fuente para las condiciones de copia.
No hay NINGUNA garantía; ni siquiera de COMERCIABILIDAD o IDONEIDAD PARA UN
FIN DETERMINADO.
Escrito por Thorsten Kukuk
```

En CentOS y derivados...

```
[root@sololinux-demo ~]# getent -V
getent (GNU libc) 2.17
Copyright (C) 2012 Free Software Foundation, Inc.
Esto es software libre; vea el código fuente para las condiciones de copia.
No hay NINGUNA garantía; ni siquiera de COMERCIABILIDAD o IDONEIDAD PARA UN
FIN DETERMINADO.
Escrito por Thorsten Kukuk.
```

## Estados de salida

El comando `getent`, en ciertas ocasiones también nos puede devolver el estado de la salida. Las vemos en la siguiente tabla.

Valor del estado	Significado del valor
0	El comando se completó correctamente
1	Faltan argumentos o no reconoce la base de datos
2	Las claves insertadas no se encuentran en la base de datos
3	La enumeración no es compatible con la base de datos

```
dovecot:!!:18574:~::~:
dovenull:!!:18574:~::~:
named:!!:18574:~::~:
popuser:!!:18574:~::~:
mhandlers-user:!!:18574:~::~:
roundcube_sysuser:!!:18574:~::~:
webalizer:!!:18574:~::~:
nginx:!!:18574:~::~:
psaftp:!!:18574:~::~:
grafana:!!:18574:~::~:
drweb:!!:18574:~::~:
memcached:!!:18574:~::~:
g54:$1$mMkQ9xE9$cI6Jm2/0ldvWX9uca0cza1:18574:0:99999:7:::
n78nn78:$1$lt9zDhPx$xCb/a.JFdpBFH7xX8A9ae1:18574:0:99999:7:::
ukm:$1$bBS3wbbP$rSSx0jfdkQVKX70DZJwLL.:18611:0:99999:7:::
v657h:$1$BS0nf90d$mtqfo0RGdzTe2mnS.25PJ.:18574:0:99999:7:::
```

[www.sololinux.es](http://www.sololinux.es)

# GETENT

## Canales de Telegram:



[Canal SoloLinux](#) – [Canal SoloWordpress](#)

Espero que este artículo te sea de utilidad, puedes colaborar con el simple gesto de compartir nuestros artículos en tu sitio web, blog, foro o redes sociales.

[Chat de SoloLinux en Telegram](#)

## Listar discos conectados con lsblk en linux

Para un buen **administrador de sistemas**, es indispensable verificar que los discos instalados en un servidor funcionan correctamente. Como es lógico, cuando te haces cargo de una nueva máquina lo primero que debes hacer es, identificar todos los **dispositivos de almacenamiento** insertados en el sistema.

Listar los discos conectados en la máquina, es tarea sencilla. Para lograr nuestro objetivo, existen muchas herramientas a nuestra disposición, pero sin duda alguna... por funciones, facilidad de uso, etc., una de las mejores es **lsblk**.



## Listar discos conectados con lsblk en linux

La fórmula más simple de enumerar los dispositivos del sistema es, haciendo uso del comando lsblk en su formato básico, sin opciones.

```
sudo lsblk
```

Vemos un ejemplo de salida, listando un raid 5.

```
[root@host ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda   8:0    0 238,5G 0 disk
├─sda1 8:1    0  512M 0 part
├─md0  9:0    0  511M 0 raid1 /boot
├─sda2 8:2    0  512M 0 part [SWAP]
├─sda3 8:3    0 232,6G 0 part
├─md1  9:1    0 464,9G 0 raid5 /
sdb   8:16   0 238,5G 0 disk
├─sdb1 8:17   0  512M 0 part
├─md0  9:0    0  511M 0 raid1 /boot
├─sdb2 8:18   0  512M 0 part [SWAP]
├─sdb3 8:19   0 232,6G 0 part
├─md1  9:1    0 464,9G 0 raid5 /
sdc   8:32   0 238,5G 0 disk
├─sdc1 8:33   0  512M 0 part
├─md0  9:0    0  511M 0 raid1 /boot
├─sdc2 8:34   0  512M 0 part [SWAP]
├─sdc3 8:35   0 232,6G 0 part
├─md1  9:1    0 464,9G 0 raid5 /
```

Otra excelente e interesante opción del comando es, que nos imprima también el sistema de archivos, con sus particiones, etc.

```
sudo lsblk -f
```

Las columnas informativas vienen con unas nomenclaturas, que explicamos a continuación.

- **NAME** – Nombre del dispositivo
- **FSTYPE** – Indica el tipo de sistema de archivos
- **LABEL** – Etiqueta del dispositivo
- **UUID** – Identificador universal del dispositivo
- **MOUNTPOINT** – Punto de montaje



Salida de ejemplo.

```
[root@host ~]# lsblk -f
NAME FSTYPE LABEL UUID
MOUNTPOINT
sda
├─sda1 linux_raid_member 0 add63980-ce81-e46f-436b-40d00ede3992
├─md0 ext4 f2d8b6dd-8d32-47a2-b427-
35151d932a82 /boot
├─sda2 swap dda0a6c7-1c0e-44e3-b3a7-2c6caa0c76a1
├─[SWAP]
├─sda3 linux_raid_member 1 14510902-de03-e7cb-5bf4-cd2dee032568
├─md1 ext4 2970de07-a351-4c73-b203-d31d1360d5cd /
sdb
├─sdb1 linux_raid_member 0 add63980-ce81-e46f-436b-40d00ede3992
├─md0 ext4 f2d8b6dd-8d32-47a2-b427-
35151d932a82 /boot
├─sdb2 swap b906ff4e-f96f-4bcc-ba47-f77b1086886a
├─[SWAP]
├─sdb3 linux_raid_member 1 14510902-de03-e7cb-5bf4-cd2dee032568
├─md1 ext4 2970de07-a351-4c73-b203-d31d1360d5cd /
sdc
├─sdc1 linux_raid_member 0 add63980-ce81-e46f-436b-40d00ede3992
├─md0 ext4 f2d8b6dd-8d32-47a2-b427-
35151d932a82 /boot
├─sdc2 swap 69e847c4-02d0-44e4-b5f2-ec8483b1b4e
├─[SWAP]
├─sdc3 linux_raid_member 1 14510902-de03-e7cb-5bf4-cd2dee032568
├─md1 ext4
```

Si prefieres imprimir el tamaño en bytes...

```
sudo lsblk -b
```

```
[root@host ~]# lsblk -b
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda   8:0    0 256060514304 0 disk
├─sda1 8:1    0  536870912 0 part
├─md0  9:0    0  535822336 0 raid1 /boot
├─sda2 8:2    0  536870912 0 part [SWAP]
├─sda3 8:3    0 249735151616 0 part
├─md1  9:1    0 499199770624 0 raid5 /
```

No es tan legible para humanos, pero si quieres exportar el listado es mejor olvidarte del formato árbol, e imprimir un listado clásico.

```
sudo lsblk -l
```

Observa la diferencia.

```
[root@host ~]# lsblk -l
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda   8:0    0 238,5G  0 disk
sda1  8:1    0   512M  0 part
md0   9:0    0   511M  0 raid1 /boot
sda2  8:2    0   512M  0 part [SWAP]
sda3  8:3    0 232,6G  0 part
md1   9:1    0 464,9G  0 raid5 /
sdb   8:16   0 238,5G  0 disk
sdb1  8:17   0   512M  0 part
md0   9:0    0   511M  0 raid1 /boot
sdb2  8:18   0   512M  0 part [SWAP]
sdb3  8:19   0 232,6G  0 part
md1   9:1    0 464,9G  0 raid5 /
sdc   8:32   0 238,5G  0 disk
sdc1  8:33   0   512M  0 part
md0   9:0    0   511M  0 raid1 /boot
sdc2  8:34   0   512M  0 part [SWAP]
sdc3  8:35   0 232,6G  0 part
md1   9:1    0 464,9G  0 raid5 /
```

Si quieres aprender más sobre este comando, puedes consultar su ayuda integrada con el siguiente comando.

```
sudo lsblk -h
```

Por ejemplo... sus muchas opciones.

```
Options:
-a, --all          print all devices
-b, --bytes        print SIZE in bytes rather than in human readable format
-d, --nodeps      don't print slaves or holders
-D, --discard      print discard capabilities
-e, --exclude <list> exclude devices by major number (default: RAM disks)
-I, --include <list> show only devices with specified major numbers
-f, --fs           output info about filesystems
-h, --help         usage information (this)
-i, --ascii        use ascii characters only
-m, --perms        output info about permissions
-l, --list         use list format output
-n, --noheadings  don't print headings
-o, --output <list> output columns
-p, --paths        print complete device path
-P, --pairs        use key="value" output format
-r, --raw          use raw output format
-s, --inverse      inverse dependencies
-t, --topology     output info about topology
-S, --scsi         output info about SCSI devices
-h, --help         display this help and exit
-V, --version      output version information and exit
```

Como puedes comprobar, estamos ante un comando extremadamente útil.

# www.sololinux.es

## Los 5 mejores linux para instalar en Windows WSL

Las vueltas que da la vida, después del repudio de Microsoft a Linux durante años, ahora podríamos decir que van de la mano. Incluso parece que monopolista **Microsoft**, ahora trabaja en estrecha colaboración con Linux (que no con la comunidad real de usuarios).

Los mayores no olvidan y, no entiendo como de pronto se debe garantizar que puedan interoperar un sistema con otro. Que ha cambiado para que esto suceda, ahhhh amigos que poco a poco se os cae el negocio lucrativo que tenáis montado, sobre una telaraña a la que le ha bastado un soplo de aire fresco para ponerla en entredicho.



Pero bueno, ya que estamos y que varios usuarios me han consultado sobre el tema, hoy vemos **los 5 mejores linux** que al parecer operan perfectamente con la capa de compatibilidad desarrollada por Microsoft, a la que ellos denominan **WSL**. No es una virtualización, que lo tengas claro, es una chapuza para no perder más usuarios en Windows.

## Los 5 mejores linux para instalar en Windows WSL

Se supone que cualquier distro que descargues, la podrías usar en **WSL**. Pero como todo en Windows, siempre falla algo, jaja. Vemos 5 **distribuciones linux** que ellos mismos ofrecen en su **store**, por tanto deberían funcionar correctamente, o no?, si eres usuario de Windows pruébalo tu mismo.

### Ubuntu



Desarrollada por **Canonical**, es evidente que viendo las últimas tendencias de Ubuntu... esta colabora estrechamente con Microsoft. Seguro que su versión WSL funciona a la perfección. Sus requisitos son mínimos, si tienes Windows 10 actualizado no tendrás ningún problema. Puedes descargar Ubuntu para Microsoft en el siguiente enlace.

- [Descargar Ubuntu para Microsoft](#)

### Kali Linux



Basada en **Debian Testing**, esta distribución linux incluye cientos de herramientas y utilidades, para pruebas de penetración y seguridad. **Kali Linux** es realmente impresionante. Puedes descargar Kali Linux para Windows, desde el siguiente enlace.

- [Descargar Kali Linux para Windows](#)

### Alpine Linux



Con un alto nivel de seguridad, ya tratamos esta distribución linux (para usuarios expertos), en un **anterior artículo**. **Alpine Linux** no viene con GUI por defecto, pero si te atreves con ella podrás instalar un entorno de escritorio sin excesivos problemas. Puedes descargar Alpine Linux para Windows desde el siguiente enlace.

- [Descargar Alpine Linux para Windows](#)

**Nota del autor:** Independientemente del **linux** elegido, no es nada recomendable usar **linux en WSL** para sistemas productivos. **Formatea tu disco e instala Linux, serás más feliz.**

### OpenSUSE



Hablar de **OpenSUSE** son palabras mayores, uno de los linux más veteranos es ampliamente utilizado a nivel empresarial. Te lo recomiendo, tal vez te sorprenda. Puedes descargar **OpenSuse** para Windows, desde el siguiente enlace.

- [Descargar OpenSuse para Windows](#)

### Debian



Conocido por su estabilidad, **Debian** es la base de Ubuntu y otras afamadas distribuciones linux. No con pocos conflictos entre sus usuarios, últimamente se intenta facilitar el acceso a los usuarios más noveles, algo que algunos agradecen, otros lo critican (sigue siendo una distro excelente). Puedes **descargar debian para Windows** desde el siguiente enlace.

- [Descargar debian para Windows](#)

## Un incendio destruye el dataCenter de OVH en Estrasburgo

Seguro que muchos de vosotros notasteis en el día de ayer, que miles de sitios web estaban offline. Lo que parecía que nunca podría pasar, paso. Un incendio en la madrugada del miércoles destruyó uno de los **centros de datos de OVH en Estrasburgo** y, parte de otro.

Por suerte, no lamentamos daños personales ni en el equipo de **OVH**, ni en los de extinción de incendios, salvo ataques de nervios que por otro lado son lógicos en estas situaciones. Recordemos que OVH es el mayor proveedor de servicios en la nube de Europa. Si descontamos **Amazon Web Services, Google Cloud y Microsoft Azure**, pocos le pueden hacer sombra a nivel mundial.

Aplicaciones caídas, miles de sitios offline, cientos de proveedores que revenden sus servicios también caídos. En fin... un auténtico desastre, sobre todo para los que piensan que nunca pasa nada y no tenían backups, todo se ha perdido.

### Un incendio destruye el datacenter de OVH en Estrasburgo

Según el **informe de incidencias** de la propia compañía, el incendio destruyó por completo el **centro de datos SBG2** y cuatro jaulas del **SBG1**. En el **SBG3** también hubo fallos y por último, el **SBG4** no tuvo ningún problema físico. Por precaución, fueron apagados los cuatro **datacenter** hasta la extinción del incendio.

La compañía ha lanzado un comunicado, en el que informa que no se restaurara la energía en el SBG1 y SBG4 hasta el lunes día 15; El **SBG3** se arrancara el viernes 19 y, el **SBG2** no tiene fecha pues deben reconstruirlo por completo.

Si nos remontamos al 2017, recordamos un gran parón también producido en Estrasburgo. Un corte de energía inesperado, tiro al suelo los datacenters de Estrasburgo. Cuarenta y cinco minutos después, el centro de datos de Roubaix perdió toda conectividad con el exterior debido a un error de software.

Habrà que replantearse la seguridad en algunos **datacenter de OVH**, pues se supone que disponen de los medios más modernos en extinción de incendios, incluyendo el agotamiento de oxígeno (o eso dicen).



El servicio de SoloLinux no se vio afectado en ningún momento, ya que no operamos con OVH, pero si a varios servidores que administramos de forma externa.

# Buscar y reparar paquetes rotos en linux

Los **administradores de paquetes en Linux**, nos permiten controlar la instalación o eliminación de paquetes (aplicaciones, software, herramientas, utilidades, etc). Además, también nos ayudan a buscar paquetes rotos en nuestro sistema y reinstalarlos, de forma que se solucionen problemas asociados a ellos.

Las herramientas que administran paquetes, tienen métodos integrados con la capacidad de manejar excepciones y errores. Pero a veces, pueden surgir problemas inesperados, la instalación falla y es incapaz de concluir la operación. Estos restos de instalaciones o desinstalaciones, es lo que conocemos como **paquetes rotos en Linux**.



*Reparar paquetes rotos en linux*

[www.sololinux.es](http://www.sololinux.es)

## Buscar y reparar paquetes rotos en linux

Cada administrador maneja diferentes tipos de paquetes. Por ejemplo, DNF, Yum y Zypper trabajan con paquetes RPM. De forma similar, apt es el frontend de dpkg en distribuciones basadas en Debian.

### Reparar paquetes en Debian, Ubuntu y derivados

Para reparar los paquetes directamente con apt, ejecutamos los siguientes comandos.

```
sudo apt update --fix-missing
# Si es necesario actualiza
# sudo apt full-upgrade
sudo apt update
sudo apt install -f
```

Si no solucionas los problemas con **apt**, atacamos directamente a **dpkg**. Comenzamos reconfigurando todos los paquetes, que aún tenemos pendientes.

```
sudo dpkg --configure -a
```

Listamos los paquetes que nos generan problemas, e intentamos desinstalarlos uno por uno.

```
sudo dpkg -l | grep ^i.r
sudo dpkg --remove --force-remove--reinstreq [paquete-roto]
```

Limpiamos nuestra caché local y, actualizamos el sistema.

```
sudo apt clean && sudo apt autoclean
sudo apt update && sudo apt dist-upgrade
```

Vemos una imagen de ejemplo sobre **Linux Mint**.

```
root@sololinux:~# sudo apt clean && sudo apt autoclean
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
root@sololinux:~# sudo apt update && sudo apt dist-upgrade
Obj:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Obj:2 http://ppa.launchpad.net/atareao/telegram/ubuntu bionic InRelease
Obj:3 http://dl.google.com/linux/chrome/deb stable InRelease
Obj:4 http://archive.ubuntu.com/ubuntu bionic InRelease
Obj:5 http://archive.canonical.com/ubuntu bionic InRelease
Obj:6 http://ppa.launchpad.net/nilarimogard/webupd8/ubuntu bionic InRelease
Ign:7 http://packages.linuxmint.com tricia InRelease
Obj:8 http://archive.ubuntu.com/ubuntu bionic-updates InRelease
Obj:9 http://archive.ubuntu.com/ubuntu bionic-backports InRelease
Obj:10 http://packages.linuxmint.com tricia Release
Obj:11 https://download.sysdig.com/stable/deb stable-amd64/ InRelease
Obj:12 http://ppa.launchpad.net/obsproject/obs-studio/ubuntu bionic InRelease
Obj:13 http://ppa.launchpad.net/quiterss/quiterss/ubuntu bionic InRelease
Obj:14 http://ppa.launchpad.net/sicklylife/filezilla/ubuntu bionic InRelease
Obj:15 http://ppa.launchpad.net/transmissionbt/ppa/ubuntu bionic InRelease
Obj:16 http://ppa.launchpad.net/ubuntu-mozilla-daily/ppa/ubuntu bionic InRelease
Obj:17 http://ppa.launchpad.net/ubuntu-mozilla-security/ppa/ubuntu bionic InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Todos los paquetes están actualizados.
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
```

### Reparar paquetes en CentOS, Fedora y derivados

Los administradores de **paquetes rpm**, son excelentes a la hora de administrar paquetes rotos. Aun lo dicho pueden surgir algunos problemas, ya que hay miles de paquetes instalados en un sistema Linux.

Verificamos los paquetes del sistema.

```
sudo rpm -Va
```

```
[root@SoloLinux ~]# sudo rpm -Va
.....P /usr/bin/ping
S.5....T. c /etc/sysconfig/saslauthd
.M..... g /run/saslauthd
.M..... /etc/cron.daily/man-db.cron
S.5....T. c /etc/firewalld/firewalld.conf
.M..... c /etc/cron.daily/logrotate
..S....T. c /etc/cron.d/dailyjobs
.....T. c /etc/crontab
.M..... g /etc/mail/access.db
.M..... g /etc/mail/domaintable.db
.M..... g /etc/mail/maillertable.db
.M..... g /etc/mail/virtusertable.db
.M...UG.. g /var/spool/clientmqueue/sm-client.st
S.5....T. c /etc/ssh/ssh_config
.M..... c /etc/cron.daily/mlocate
```

Puedes reinstalar el paquete que de problemas, con el comando...

```
sudo dnf --refresh reinstall [paquete]
sudo yum --refresh reinstall [paquete]
```

Un error común en CentOS, es el de archivos duplicados a la hora de actualizar. La solución es sencilla. Comenzamos instalando **<yum-utils>**.

```
sudo yum install yum-utils
```

Ahora listamos los paquetes con conflictos o duplicados.

```
sudo package-cleanup --dupes
```

Los borramos.

```
sudo package-cleanup --cleandupes
```

En nuestro ejemplo no tenemos paquetes problemáticos.

```
[root@SoloLinux ~]# sudo package-cleanup --cleandupes
Complementos cargados:fastestmirror
No duplicates to remove
```

## Deshabilitar el inicio de sesión ssh de un usuario



**SSH (Secure Shell)**, es un **protocolo de red** que nos permite acceder de forma segura a un sistema remoto. Pero en ocasiones podemos tener algún problema, y es que por defecto SSH permite el inicio de sesión de cualquier usuario del sistema, evidentemente siempre que tenga en su poder la contraseña correspondiente.

Este tipo de accesos (a veces incontrolados), pueden derivar en que un maligno atacante explote al usuario y, se apodere de tu sistema obteniendo acceso de administrador. Hoy, veremos cómo **deshabilitar el inicio de sesión ssh de un usuario en particular** y, si lo deseas el **acceso root** total. Es por tu seguridad.

```
GNU nano 2.3.1 Fichero: /etc/ssh/sshd_config
# $OpenBSD: sshd_config,v 1.100 2016/08/15 12:32:04 naddy Exp $
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.
# This sshd was compiled with PATH=/usr/local/bin:/usr/bin
# The strategy used for options in the default sshd config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.
# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
```

### Deshabilitar el inicio de sesión ssh de un usuario

Deshabilitar el inicio de sesión ssh de un usuario en particular, es tan sencillo como agregar una simple línea en el archivo de **configuración ssh**, normalmente «**sshd\_config**».

```
sudo nano /etc/ssh/sshd_config
```

Al final del archivo agrega lo siguiente (el usuario a bloquear).

```
DenyUsers [usuario]
```

Observa el ejemplo, bloqueamos a «**sergio**» (a mi mismo, jeje).

```
# Accept locale-related environment variables
AcceptEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY
LC_MESSAGES
AcceptEnv LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
AcceptEnv LC_IDENTIFICATION LC_ALL LANGUAGE
AcceptEnv XMODIFIERS
# override default of no subsystems
Subsystem sftp /usr/libexec/openssh/sftp-server
# Example of overriding settings on a per-user basis
Match User anoncvs
# X11Forwarding no
# AllowTcpForwarding no
# PermitTTY no
# ForceCommand cvs server
DenyUsers sergio <----- Usuario bloqueado
```

Solo falta reiniciar el servicio.

```
sudo systemctl restart ssh
# 0
sudo service ssh restart
```

Si quieres deshabilitar el inicio de **sesión root**, editamos el mismo archivo.

```
sudo nano /etc/ssh/sshd_config
```

Busca la siguiente línea...

```
#PermitRootLogin yes

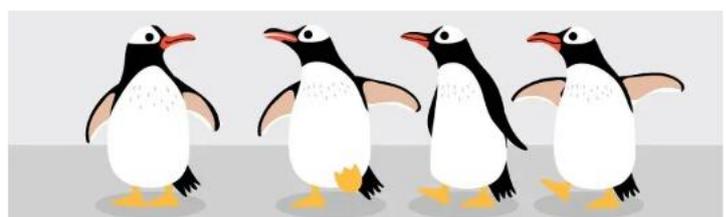
# Authentication:
#LoginGraceTime 2m
#PermitRootLogin yes <-----
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Descomentas la línea marcada y, editas el «**yes**» por «**no**». Debe quedar como en el ejemplo...

```
# Authentication:
#LoginGraceTime 2m
#PermitRootLogin no <-----
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Listo, guardas el archivo, cierras el editor y reinicias el servicio.

```
sudo systemctl restart ssh
# 0
sudo service ssh restart
```



Canales de Telegram:

[Canal SoloLinux – Canal SoloWordPress](#)

Espero que este artículo te sea de utilidad, puedes colaborar con el simple gesto de compartir nuestros artículos en tu sitio web, blog, foro o redes sociales.

[Chat de SoloLinux en Telegram](#)

# Convertir CentOS 8 en Alma Linux

Harto de la espera, y porque no decirlo, «**enfadado**» con la drástica decisión de **RED-HAT sobre CentOS**, he tomado una decisión. Voy a intentar actualizar, o mejor dicho **convertir CentOS 8 en Alma Linux**.

Es evidente que para el experimento, tomo un servidor de pruebas. El resultado... realmente sorprendente, ha sido un proceso rápido, seguro y fiable. Soy consciente de que tal vez es un poco precipitado, pues **Alma Linux** aún está en versión beta, pero la verdad es que se me comen los nervios y, como aún no tenemos lista ninguna versión de **Rocky Linux**, me lanzo a la piscina sin saber si tiene agua.

En este artículo de hoy, vemos como **convertir CentOS 8 en Alma Linux**, o si lo prefieres.... **actualizar CentOS 8 a Alma Linux**. Antes de comenzar dos puntualizaciones, no te recomiendo realizar este proceso en servidores con producción, también te indico que hemos usado un **CentOS 8 minimal** en servidor remoto sin GUI.



## Convertir CentOS 8 en Alma Linux

Lo primero que hacemos, es verificar la versión instalada de CentOS en nuestro sistema.

```
sudo dnf update
cat /etc/os-release
```

En la salida, observamos que tenemos instalada la última versión de CentOS 8.

```
[root@SoloLinux ~]# cat /etc/os-release
NAME="CentOS Linux"
VERSION="8"
ID="centos"
ID_LIKE="rhel fedora"
VERSION_ID="8"
PLATFORM_ID="platform:el8"
PRETTY_NAME="CentOS Linux 8"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:centos:centos:8"
HOME_URL="https://centos.org/"
BUG_REPORT_URL="https://bugs.centos.org/"
CENTOS_MANTISBT_PROJECT="CentOS-8"
CENTOS_MANTISBT_PROJECT_VERSION="8"
[root@SoloLinux ~]#
```

Bien, todo es perfecto. Ahora descargamos un **script sh** que los desarrolladores de Alma Linux ponen a nuestra disposición.

```
curl -O
https://raw.githubusercontent.com/AlmaLinux/almalinux-deploy/master/almalinux-deploy.sh
```

```
[root@SoloLinux ~]# curl -O
https://raw.githubusercontent.com/AlmaLinux/almalinux-deploy/master/almalinux-deploy.sh
% Total % Received % Xferd Average Speed Time Time Time
Current % Dload Upload Total Spent Left
Speed
100 9607 100 9607 0 0 60043 0 ---:---:---:---:
--:--:-- 60043
[root@SoloLinux ~]#
```

Una vez descargado el script, lo ejecutamos con el siguiente comando.

```
sudo bash almalinux-deploy.sh

[root@SoloLinux ~]# sudo bash almalinux-deploy.sh
Check root privileges 0%
Check Secure Boot status 0%
Check centos-8_alm 84 is supported 0%
Download RPM-RPG-Kernel-headers 0%
Download almalinux-release package 0%
Verify almalinux-release package 0%
Remove centos-release package 0%
Remove centos-gpp-key package 0%
Remove centos-linux-root package 0%
Verifying... ***** [100%]
Preparando... ***** [100%]
Actualizando / instalando... ***** [100%]
1.0almalinux-release-8.3-3.el8 ***** [100%]
buscando el servidor más asociado (a servidores)... done.
AlmaLinux 8 - BaseOS 3.0 MB/s 3.3 MB 00:01
AlmaLinux 8 - AppStream 28 MB/s 6.9 MB 00:00
AlmaLinux 8 - PowerTools 19 MB/s 1.9 MB 00:00
AlmaLinux 8 - Extras 26 MB/s 3.3 MB 00:00
```

Todo el proceso es automático. Al concluir recibirás un aviso.

```
vim-common-2:8.0.1763-15.el8.x86_64
vim-enhanced-2:8.0.1763-15.el8.x86_64
vim-filesystem-2:8.0.1763-15.el8.noarch
vim-minimal-2:8.0.1763-15.el8.x86_64
wget-1.19.5-10.el8.x86_64
which-2.21-12.el8.x86_64
xkeyboard-config-2.28-1.el8.noarch
5.2.4-3.el8.x86_64
xz-libs-5.2.4-3.el8.x86_64
yum-4.2.23-4.el8.noarch
zip-3.0-23.el8.x86_64
zlib-1.2.11-16.el8_2.x86_64
¡Listo!
Run dnf distro-sync -y
OK
Migration to AlmaLinux is completed
```

Listo, ya tienes la migración realizada. Puedes verificar que los repositorios han cambiado, lo que significa que a partir de hoy... tu Alma Linux se actualiza de forma oficial.

```
cd /etc/yum.repos.d
dir
```

Repositorios de Alma Linux oficiales.

```
[root@SoloLinux ~]# cd /etc/yum.repos.d
[root@SoloLinux yum.repos.d]# dir
almalinux-ha.repo almalinux.repo
```

También podemos ejecutar....

```
cat /etc/os-release
```

Revisa la imagen de ejemplo, la operación ha sido un éxito.

```
[root@SoloLinux ~]# cat /etc/os-release
NAME="AlmaLinux"
VERSION="8.3 (Purple Manul)"
ID="almalinux"
ID_LIKE="rhel centos fedora"
VERSION_ID="8.3"
PLATFORM_ID="platform:el8"
PRETTY_NAME="AlmaLinux 8.3 RC (Purple Manul)"
ANSI_COLOR="0;34"
CPE_NAME="cpe:/o:almalinux:almalinux:8.3:rc"
HOME_URL="https://almalinux.org/"
BUG_REPORT_URL="https://bugs.almalinux.org/"
ALMALINUX_MANTISBT_PROJECT="AlmaLinux-8"
ALMALINUX_MANTISBT_PROJECT_VERSION="8.3"
```

Para dar por terminada la migración, reiniciamos el servidor.

```
sudo reboot
```

# Como instalar LAMP en Alma Linux



En contra de lo que algunos piensan, **LAMP** no es un software empaquetado como tal, sino un conjunto de aplicaciones **opensource**, que se utiliza en el desarrollo de aplicaciones web o como simple servidor de sitios.

Basado en sistemas Linux, consta de un **servidor http Apache**, una base de datos MySQL o MariaDB y el conocido lenguaje de programación PHP.

Aprovechando que después del [artículo anterior](#), tengo un servidor Alma Linux corriendo; quiero probar si aún siendo una versión beta ya podemos instalar un servidor LAMP. Los requisitos son fáciles de cumplir.

- Un servidor **Alma Linux**.
- Acceso a la terminal / línea de comandos.
- Usuario **root** o con privilegios de administrador.
- Administrador de paquetes **dnf** y **RPM** (incluidos por defecto).



## Como instalar LAMP en Alma Linux

Como es lógico, primero actualizamos Alma Linux.

```
sudo dnf update
```

### Instalar Apache en Alma Linux

Instalamos Apache.

```
sudo dnf -y install httpd
```

```
[root@SoloLinux ~]# sudo dnf -y install httpd
Última comprobación de caducidad de metadatos hecha hace 0:13:21, el
sáb 13 mar 2021 09:29:51 CET.
Dependencias resueltas.
```

Paquete	Arquitectura Repositorio	Versión Tam.
-----		
Instalando:		

```
httpd
30.module_el8.3.0+2016+8bf57d29.alma
1.4 M
Instalando dependencias:
almalinux-logos-httpd
noarch 81.2-1.el8
appstream 27 k
apr
x86_64 1.6.3-11.el8
appstream 124 k
apr-util
x86_64 1.6.1-6.el8
appstream 105 k
httpd-filesystem
noarch 2.4.37-
30.module_el8.3.0+2016+8bf57d29.alma
appstream
36 k
httpd-tools
x86_64 2.4.37-
30.module_el8.3.0+2016+8bf57d29.alma
appstream
103 k
mailcap
noarch 2.1.48-3.el8
baseos 39 k
mod_http2
x86_64 1.15.7-
2.module_el8.3.0+2016+8bf57d29
appstream
153 k
Instalando dependencias débiles:
apr-util-bdb
x86_64 1.6.1-6.el8
appstream 24 k
apr-util-openssl
x86_64 1.6.1-6.el8
appstream 27 k
Activando flujos de módulos:
httpd 2.4
```

```
Resumen de la transacción
-----
Instalar 10 Paquetes
Tamaño total de la descarga: 2.0 M
Tamaño instalado: 5.4 M
Descargando paquetes:
(1/10): mailcap-2.1.48-3.el8.noarch.rpm
976 kB/s | 39 kB 00:00
(2/10): almalinux-logos-httpd-81.2-1.el8.noarch.rpm
638 kB/s | 27 kB 00:00
(3/10): apr-1.6.3-11.el8.x86_64.rpm
2.5 MB/s | 124 kB 00:00
(4/10): apr-util-bdb-1.6.1-6.el8.x86_64.rpm
```

```
Instalado:
almalinux-logos-httpd-81.2-1.el8.noarch
apr-1.6.3-11.el8.x86_64
apr-util-1.6.1-6.el8.x86_64
apr-util-bdb-1.6.1-6.el8.x86_64
apr-util-openssl-1.6.1-6.el8.x86_64
httpd-2.4.37-30.module_el8.3.0+2016+8bf57d29.alma.x86_64
httpd-filesystem-2.4.37-30.module_el8.3.0+2016+8bf57d29.alma.noarch
httpd-tools-2.4.37-30.module_el8.3.0+2016+8bf57d29.alma.x86_64
mailcap-2.1.48-3.el8.noarch
mod_http2-1.15.7-2.module_el8.3.0+2016+8bf57d29.x86_64
```

Iniciamos el servicio.

```
sudo systemctl start httpd.service
```

Para verificar que está corriendo, inserta la **ip del servidor** en tu navegador web. Si tienes un server en local, puedes probar con «localhost» o «10.0.0.1». Verás una imagen similar a...



### Instalar MariaDB en Alma Linux

Ahora instalamos MariaDB.

```
sudo dnf install mariadb-server
```

```

Última comprobación de cantidad de paquetes hecha hace 0:10:41, el sáb 13 mar 2021 09:20:51 CET.
Dependencias Resueltas:
Paquete Arquitectura Versión Repositorio Tam.
Instalando:
mariadb-server x86_64 3:10.3.27-3.module_el8.3.0-2028-5c3224e9 appstream 16 M
Instalamos dependencias:
libaio x86_64 6.3.112-1.el8 basos 32 k
mariadb x86_64 3:10.3.27-3.module_el8.3.0-2028-5c3224e9 appstream 611 M
mariadb-common x86_64 3:10.3.27-3.module_el8.3.0-2028-5c3224e9 appstream 62 k
mariadb-connector-c x86_64 3:1.11.7-1.el8 appstream 194 k
mariadb-connector-c-config noarch 3:1.11.2-3.el8 appstream 15 k
mariadb-errmsg x86_64 3:10.3.27-3.module_el8.3.0-2028-5c3224e9 appstream 234 k
perl-DBD-mysql x86_64 4.030-3.module_el8.3.0-2028-5c3224e9 appstream 150 k
perl-DBI x86_64 1.641-3.module_el8.3.0-2028-5c3224e9 appstream 738 k
Instalamos dependencias débiles:
mariadb-backup x86_64 3:10.3.27-3.module_el8.3.0-2028-5c3224e9 appstream 6.8 M
mariadb-libs x86_64 3:10.3.27-3.module_el8.3.0-2028-5c3224e9 appstream 58 k
mariadb-server-utils x86_64 3:10.3.27-3.module_el8.3.0-2028-5c3224e9 appstream 1.1 M
Activamos el servicio de inicio:
mariadb 10.3
perl-DBD-mysql 4.030
perl-DBI 1.641
Resumen de la transacción
Instalar 12 Paquetes
Cambio total de la descarga: 31 M
Tiempo instalado: 15 s
¿Está de acuerdo [Y/n]?
www.sololinux.es
    
```

```

Última comprobación de cantidad de paquetes hecha hace 0:10:41, el sáb 13 mar 2021 09:20:51 CET.
Dependencias Resueltas:
Paquete Arquitectura Versión Repositorio Tam.
Instalando:
php x86_64 7.2.24-1.module_el8.3.0-2028-5c3224e9 appstream 1.6 M
php-cli x86_64 7.2.24-1.module_el8.3.0-2028-5c3224e9 appstream 3.1 M
php-common x86_64 7.2.24-1.module_el8.3.0-2028-5c3224e9 appstream 650 k
php-fpm x86_64 7.2.24-1.module_el8.3.0-2028-5c3224e9 appstream 1.6 M
php-gd x86_64 7.2.24-1.module_el8.3.0-2028-5c3224e9 appstream 70 k
php-ldap x86_64 7.2.24-1.module_el8.3.0-2028-5c3224e9 appstream 221 k
php-mbstring x86_64 7.2.24-1.module_el8.3.0-2028-5c3224e9 appstream 48 k
php-pecl-zip x86_64 1.13.3-1.module_el8.3.0-2028-5c3224e9 appstream 48 k
Instalando dependencias:
libcap x86_64 1.5.1-2.module_el8.3.0-2028-5c3224e9 appstream 61 k
logrotate-noarch noarch 1:1.14.1-9.module_el8.3.0-2028-5c3224e9 appstream 24 k
Resumen de la transacción
Instalar 10 Paquetes
Cambio total de la descarga: 7.0 M
Tiempo instalado: 27 M
Descargando paquetes:
(1/10): main-filesystem-1.14.1-9.module_el8.3.0-2028-5c3224e9.noarch.rpm 401 kB/s | 24 kB 00:00
(2/10): libzip-1.5.1-2.module_el8.3.0-2028-5c3224e9.x86_64.rpm 1.1 MB/s | 81 kB 00:00
(3/10): php-common-7.2.24-1.module_el8.3.0-2028-5c3224e9.x86_64.rpm 23 MB/s | 650 kB 00:00
(4/10): php-7.2.24-1.module_el8.3.0-2028-5c3224e9.x86_64.rpm 16 MB/s | 1.5 MB 00:00
www.sololinux.es
    
```

Iniciamos y habilitamos el servicio.

```

systemctl start mariadb
systemctl enable mariadb

[root@SoloLinux ~]# systemctl start mariadb
[root@SoloLinux ~]# systemctl enable mariadb
Created symlink /etc/systemd/system/mysql.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/mysqld.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /usr/lib/systemd/system/mariadb.service.
[root@SoloLinux ~]#
    
```

Verificamos que el servicio inicio correctamente.

```

systemctl status mariadb

[root@SoloLinux ~]# systemctl status mariadb
● mariadb.service - MariaDB 10.3 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2021-03-13 09:47:37 CET; 32s ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
   Main PID: 55471 (mysqld)
   Status: "Taking your SQL requests now..."
     Tasks: 30 (limit: 104857)
    Memory: 72.7M
     CGroup: /system.slice/mariadb.service
            └─55471 /usr/libexec/mysqld --basedir=/usr
    
```

Es fundamental que asegures la instalación de MariaDB.

```
mysql_secure_installation
```

El motor nos lanza una serie de preguntas, las respuestas que yo te propongo son una buena opción.

- **Enter current password for root (enter for none):**  
Pulsa Enter
- **Set root password? [Y/n]:** Y
- **New password:** Introduce el password
- **Re-enter new password:** Repite el password
- **Remove anonymous users? [Y/n]:** Y
- **Disallow root login remotely? [Y/n]:** Y
- **Remove test database and access to it? [Y/n]:** Y
- **Reload privilege tables now? [Y/n]:** Y

Solo falta reiniciar MariaDB.

```
sudo systemctl restart mariadb.service
```

Opcionalmente... puedes verificar la versión instalada de MariaDB.

Instalar PHP en Alma Linux

En el artículo «Instalar LAMP en Alma Linux», no podía faltar php con sus módulos más comunes.

```
sudo dnf -y install php php-pdo php-pecl-zip php-json php-common php-fpm php-mbstring php-cli
```

**Este proceso es opcional (también es posible hacerlo antes de instalar PHP).** Al igual que CentOS 8, Alma Linux instala por defecto una versión de php un tanto obsoleta, la 7.2.x. Por suerte tenemos el **repositorio remi** que nos ayuda a solucionar este problema. Lo descargamos e instalamos.

```
wget http://rpms.famillecollet.com/enterprise/remi-release-8.rpm
```

```
sudo rpm -Uvh remi-release-8*.rpm
```

```

[root@SoloLinux ~]# sudo rpm -Uvh remi-release-8*.rpm
advertencia:remi-release-8.rpm: EncabezadoV4 RSA/SHA256 Signature, ID de clave 5f11735a: NOKEY
Verifying...
##### [100%]
Preparando...
##### [100%]
Actualizando / instalando...
 1:remi-release-8.3-1.el8.remi
##### [100%]
    
```

Bien, ahora habilitamos el repo.

```
sudo nano /etc/yum.repos.d/remi.repo
```

Al principio del archivo, verás esto.

```

# Repository: https://rpms.remirepo.net/
# Blog: https://blog.remirepo.net/
# Forum: https://forum.remirepo.net/
[remi]
name=Remi's RPM repository for Enterprise Linux 8 - $basearch
#baseurl=http://rpms.remirepo.net/enterprise/8/remi/$basearch/
#mirrorlist=https://rpms.remirepo.net/enterprise/8/remi/$basearch/
httpsmirror
mirrorlist=http://cdn.remirepo.net/enterprise/8/remi/$basearch/mirror
enabled=0
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-remi.el8
    
```

Para habilitarlo sustituye enabled=0, por enabled=1.

```

# Repository: https://rpms.remirepo.net/
# Blog: https://blog.remirepo.net/
# Forum: https://forum.remirepo.net/
[remi]
name=Remi's RPM repository for Enterprise Linux 8 - $basearch
#baseurl=http://rpms.remirepo.net/enterprise/8/remi/$basearch/
#mirrorlist=https://rpms.remirepo.net/enterprise/8/remi/$basearch/
httpsmirror
mirrorlist=http://cdn.remirepo.net/enterprise/8/remi/$basearch/mirror
enabled=1 <-----1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-remi.el8
    
```

Guarda el archivo y cierra el editor.

Actualizamos.

```
sudo dnf update
```

Si quieres instalar PHP 7.4, ejecutas lo siguiente...

```
sudo dnf -y install php74
```

En nuestro caso, como queremos **instalar LAMP en Alma Linux** que sea novedoso, nos decantamos por instalar **PHP 8.0**.

```
sudo dnf -y install php80
```

Comienza la instalación...

```
[root@SoloLinux]# sudo dnf -y install php80
Última comprobación de caducidad de metadatos hecha hace 0:13:05, el lun 15 mar 2021 13:33:18 CET.
Dependencias resueltas.
```

Paquete	Arquitectura	Versión	Repositorio	Tam.
Instalando:				
php80	x86_64	1.0-3.el8.remi	remi-safe	7.0 k
Instalando dependencias:				
checkpolicy	x86_64	2.9-1.el8	baseos	347 k
environment-modules	x86_64	4.5.2-1.el8	baseos	421 k
php80-php-cli	x86_64	8.0.3-1.el8.remi	remi-safe	3.2 M
php80-php-common	x86_64	8.0.3-1.el8.remi	remi-safe	720 k
php80-runtime	x86_64	1.0-3.el8.remi	remi-safe	1.1 M
polycoreutils-python-utils	noarch	2.9-9.el8	baseos	251 k
python3-audit	x86_64	3.0-0.17.20191104git1c2f876.el8	baseos	86 k
python3-libsemanage	x86_64	2.9-3.el8	baseos	126 k
python3-polycoreutils	noarch	2.9-9.el8	baseos	2.2 M
python3-setools	x86_64	4.3.0-2.el8	baseos	625 k
scl-utils	x86_64	1:2.0.2-12.el8	appstream	47 k
tcl	x86_64	1:8.6.8-2.el8	baseos	1.1 M
Resumen de la transacción				
Instalar 13 Paquetes				
Tamaño total de la descarga: 10 M				
Tamaño instalado: 36 M				
Descargando paquetes:				
(1/13): environment-modules-4.5.2-1.el8.x86_64.rpm			5.1 MB/s   421 kB	00:00
(2/13): polycoreutils-python-utils-2.9-9.el8.noarch.rpm			2.9 MB/s   251 kB	00:00
(3/13): python3-audit-3.0-0.17.20191104git1c2f876.el8.x86_64.rpm			11 MB/s   86 kB	00:00
(4/13): checkpolicy-2.9-1.el8.x86_64.rpm				

Una vez termine el proceso, reiniciamos el servidor.

```
sudo reboot
```

**Nota final:** Recuerda que instalar **LAMP en Alma Linux** aún no es seguro, es una beta. No se recomienda usar en producción.

## Canales de Telegram:

### Canal SoloLinux – Canal SoloWordpress

Espero que este artículo te sea de utilidad, puedes colaborar con el simple gesto de compartir nuestros artículos en tu sitio web, blog, foro o redes sociales.

### Chat de SoloLinux en Telegram

## Configurar un servidor OpenVPN en Debian 10



Una **red privada virtual**, también conocida como **VPN**, es una conexión segura y encriptada entre dos puntos. Su uso puede parecer un poco exagerado para usuarios normales, pero te equivocas, ya que protege nuestra conexión a Internet y también nuestra privacidad en línea.

**OpenVPN** es una herramienta VPN gratuita y de código abierto. Implanta técnicas y reglas que crean conexiones punto a punto, de forma segura en la configuración. Es compatible con los principales sistemas operativos, pero claro... si usas **linux** mejor que mejor.



## Configurar un servidor OpenVPN en Debian 10

Antes de comenzar la instalación de nuestro servidor, actualizamos el sistema.

```
sudo apt update -y
sudo apt upgrade -y
```

### Instalar OpenVPN

**Debian 10** incluye **OpenVPN** en sus repositorios oficiales, por tanto, instalarlo es tan fácil como ejecutar este comando.

```
sudo apt install openvpn -y
```

```
root@SoloLinux:~# sudo apt install openvpn -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
easy-rsa libccid liblzo2-2 libpcsc-lite libpkcs11-helper1 libusb-
1.0-0
opencsc opencsc-pkcs11 pcscd
Paquetes sugeridos:
pcmciautils openvpn-systemd-resolved
Se instalarán los siguientes paquetes NUEVOS:
easy-rsa libccid liblzo2-2 libpcsc-lite libpkcs11-helper1 libusb-
1.0-0
opencsc opencsc-pkcs11 openvpn pcscd
```

### Instalar EasyRSA

**EasyRSA** es una herramienta que nos facilita la tarea de crear y administrar, una **CA PKI (public key infrastructure)**. Con esta aplicación en línea de comandos, podremos generar múltiples tipos de certificados. En OpenVPN la necesitamos para crear la autoridad de certificación raíz, también para solicitar y firmar los certificados de OpenVPN.

Descargamos la última versión de su [zona oficial de Github](https://github.com/OpenVPN/easy-rsa).

```
wget
https://github.com/OpenVPN/easy-rsa/releases/download/v3.0.8/EasyRSA-3.0.8.tgz
```

Extraemos el archivo.

```
tar -xvzf EasyRSA-3.0.8.tgz
```

Ahora lo copiamos en el directorio de OpenVPN.

```
cp -r EasyRSA-3.0.8 /etc/openvpn/easy-rsa
```

### Crear la autoridad de certificación

Si quieres configurar correctamente un servidor OpenVPN en Debian 10, estos pasos son necesarios, no te asustes... no es tan difícil.

Nos movemos al directorio de **EasyRSA**.

```
cd /etc/openvpn/easy-rsa
```

Ahora creamos un archivo de configuración, al cual llamaremos «**vars**».

```
nano vars
```

Copia y pega lo siguiente con tus datos. **Ojo, no olvides las comillas.**

```
set_var EASYRSA "$PWD"
set_var EASYRSA_PKI "$EASYRSA/pki"
set_var EASYRSA_DN "cn_only"
set_var EASYRSA_REQ_COUNTRY "SPAIN" # pais
set_var EASYRSA_REQ_PROVINCE "Huesca" # provincia
set_var EASYRSA_REQ_CITY "Monzon" # localidad
set_var EASYRSA_REQ_ORG "SOLOLINUX CERTIFICATE AUTHORITY" #
Organizacion
set_var EASYRSA_REQ_EMAIL "admin@tuemail.com" # Email del
administrador
set_var EASYRSA_REQ_OU "SOLOLINUX EASY CA" # Nombre
set_var EASYRSA_KEY_SIZE "2048"
set_var EASYRSA_ALGO "rsa"
set_var EASYRSA_CA_EXPIRE "7500"
set_var EASYRSA_CERT_EXPIRE "365"
set_var EASYRSA_NS_SUPPORT "no"
set_var EASYRSA_NS_COMMENT "SOLOLINUX CERTIFICATE AUTHORITY"
#Comentario
set_var EASYRSA_EXT_DIR "$EASYRSA/x509-types"
set_var EASYRSA_SSL_CONF "$EASYRSA/openssl-easyrsa.cnf"
set_var EASYRSA_DIGEST "sha256"
```

Guarda el archivo y cierra el editor.

```
GNU nano 3.2 vars
set_var EASYRSA "$PWD"
set_var EASYRSA_PKI "$EASYRSA/pki"
set_var EASYRSA_DN "cn_only"
set_var EASYRSA_REQ_COUNTRY "SPAIN" # pais
set_var EASYRSA_REQ_PROVINCE "Huesca" # provincia
set_var EASYRSA_REQ_CITY "Monzon" # localidad
set_var EASYRSA_REQ_ORG "SOLOLINUX CERTIFICATE AUTHORITY" # Organizacion
set_var EASYRSA_REQ_EMAIL "admin@tuemail.com" # Email del administrador
set_var EASYRSA_REQ_OU "SOLOLINUX EASY CA" # Nombre
set_var EASYRSA_KEY_SIZE "2048"
set_var EASYRSA_ALGO "rsa"
set_var EASYRSA_CA_EXPIRE "7500"
set_var EASYRSA_CERT_EXPIRE "365"
set_var EASYRSA_NS_SUPPORT "no"
set_var EASYRSA_NS_COMMENT "SOLOLINUX CERTIFICATE AUTHORITY" #Comentario
set_var EASYRSA_EXT_DIR "$EASYRSA/x509-types"
set_var EASYRSA_SSL_CONF "$EASYRSA/openssl-easyrsa.cnf"
set_var EASYRSA_DIGEST "sha256"
```



Copiamos los certificados y claves al directorio del cliente.

```
cp pki/ca.crt /etc/openvpn/client/  
cp pki/issued/vpnclient.crt /etc/openvpn/client/  
cp pki/private/vpnclient.key /etc/openvpn/client/
```

## Configurar el servidor OpenVPN

Bueno, ya tenemos los certificados y claves listas para funcionar. Vamos a la configuración del servidor OpenVPN propiamente dicha y, para ello necesitamos un archivo de configuración.

```
nano /etc/openvpn/server.conf
```

Copia y pega lo siguiente.

```
port 1194  
proto udp  
dev tun  
ca /etc/openvpn/server/ca.crt  
cert /etc/openvpn/server/vpnserver.crt  
key /etc/openvpn/server/vpnserver.key  
dh /etc/openvpn/server/dh.pem  
server 10.8.0.0 255.255.255.0  
push "redirect-gateway def1"  
push "dhcp-option DNS 208.67.222.222"  
push "dhcp-option DNS 208.67.220.220"  
duplicate-cn  
cipher AES-256-CBC  
tls-version-min 1.2  
tls-cipher TLS-DHE-RSA-WITH-AES-256-GCM-SHA384:TLS-DHE-RSA-WITH-AES-256-CBC-SHA256:TLS-DHE-RSA-WITH-AES-128-GCM-SHA256:TLS-DHE-RSA-WITH-AES-128-CBC-SHA256  
auth SHA512  
auth-nocache  
keepalive 20 60  
persist-key  
persist-tun  
compress lz4  
daemon  
user nobody  
group nogroup  
log-append /var/log/openvpn.log  
verb 3
```

Guarda el archivo y cierra el editor.

**Por fin!!!**, iniciamos y habilitamos el servicio.

```
systemctl start openvpn@server  
systemctl enable openvpn@server
```

## Habilitar el reenvío de IP

El reenvío de IP permite al sistema aceptar paquetes de red entrantes y, reenviarlos a otra red. Para habilitar el servicio debemos editar su archivo de configuración.

```
sudo nano /etc/sysctl.conf
```

Busca la línea... Y la descomentas...

```
#net.ipv4.ip_forward=1
```

```
Turn on Source Address Verification in all interfaces to  
prevent some spoofing attacks  
#net.ipv4.conf.default.rp_filter=1  
#net.ipv4.conf.all.rp_filter=1  
  
# Uncomment the next line to enable TCP/IP SYN cookies  
# See http://lwn.net/Articles/277146/  
# Note: This may impact IPv6 TCP sessions too  
#net.ipv4.tcp_syncookies=1  
  
# Uncomment the next line to enable packet forwarding for IPv4  
net.ipv4.ip_forward=1  
  
# Uncomment the next line to enable packet forwarding for IPv6  
# Enabling this option disables Stateless Address Autoconfiguration  
# based on Router Advertisements for this host  
#net.ipv6.conf.all.forwarding=1
```

www.sololinux.es

Guarda el archivo y cierra el editor, nuestro último paso es aplicar los cambios.

```
sysctl -p
```

La configuración de un **servidor OpenVPN en Debian 10**, ha concluido. En el siguiente artículo instalaremos y configuraremos un cliente externo.

# SoloWordPress

Manuales

Noticias

Temas

Plugins

SEO

Seguridad

Entra en: [www.solowordpress.es](http://www.solowordpress.es)



# Configurar un Cliente OpenVPN en Debian 10



```
client
dev tun
proto udp
remote IP-del-VPNSERVER 1194
ca ca.crt
cert vpnclient.crt
key vpnclient.key
cipher AES-256-CBC
auth SHA512
auth-nocache
tls-version-min 1.2
tls-cipher TLS-DHE-RSA-WITH-AES-256-GCM-SHA384:TLS-DHE-RSA-WITH-AES-256-CBC-SHA256:TLS-DHE-RSA-WITH-AES-128-GCM-SHA256:TLS-DHE-RSA-WITH-AES-128-CBC-SHA256
resolv-retry infinite
compress lz4
nobind
persist-key
persist-tun
mute-replay-warnings
verb 3
```

Si en el artículo anterior, aprendimos a [instalar un servidor OpenVPN](#); hoy instalaremos y configuraremos su cliente nativo. Existen otras alternativas, pero como este ninguno.

Al contrario de la instalación del servidor, el cliente lo puedes instalar y configurar en apenas cinco o 10 minutos. Observa que fácil es...



## Configurar un cliente OpenVPN en Debian 10

Instalamos OpenVPN en Debian 10.

```
sudo apt install openvpn -y
```

Una vez instalado OpenVPN, copiamos el certificado y la **key del cliente** desde el servidor que instalamos anteriormente, a la máquina del cliente. Puedes clonar los datos con el siguiente comando (con tus datos).

```
scp -r root@ip-vpnserver:/etc/openvpn/client
```

Nos movemos al directorio del cliente.

```
cd /etc/openvpn/client
```

```
root@sololinux:~# cd /etc/openvpn/client
root@sololinux:/etc/openvpn/client#
```

Creamos el archivo de configuración.

```
sudo nano client.ovpn
```

Copia y pega lo siguiente, no te olvides insertar datos válidos.

Guarda el archivo y cierra el editor. Imagen de ejemplo...



Ya puedes conectarte con el siguiente comando.

```
openvpn --config client.ovpn
```

Si la conexión es correcta, recibirás una respuesta como esta o similar.

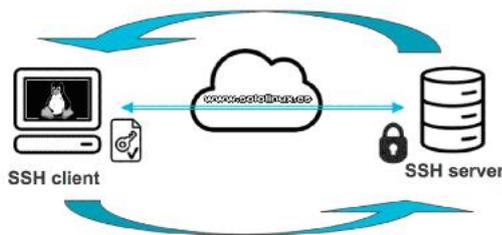
```
Mie Mar 17 15:24:47 2021 SENT CONTROL [vpnserver]: 'PUSH_REQUEST' (status=1)
Mie Mar 17 15:24:47 2021 PUSH: Received control message: 'PUSH_REPLY,redirect-gateway def1,dhcp-option DNS 208.67.222.222,dhcp-option DNS 208.67.220.220,route 10.8.0.1,topology net30,ping 20,ping-restart 60,ifconfig 10.8.0.6 10.8.0.5,peer-id 0,cipher AES-256-GCM'
Mie Mar 17 15:24:47 2021 OPTIONS IMPORT: timers and/or timeouts modified
Mie Mar 17 15:24:48 2021 OPTIONS IMPORT: --ifconfig/up options modified
Mie Mar 17 15:24:48 2021 OPTIONS IMPORT: route options modified
```

Ya puedes comenzar a usar tu servidor OpenVPN.

## Iniciar sesión ssh sin password en linux

Por seguridad, es algo común que al **iniciar sesión ssh** nos solicite nuestra contraseña (password). Si administras varios **sistemas linux**, esta operación puede volverse pesada además de insegura, ya que transmites los caracteres en texto plano.

Tal vez debas plantearte, **iniciar sesión ssh sin password**. Si te preguntas como puedes hacerlo, te digo que es muy fácil, tan solo debes generar unas **claves rsa** (pública y privada). Es posible que su propio nombre te asuste, tranquilo es un proceso muy sencillo y rápido, que aprenderás en el artículo de hoy.



### Iniciar sesión ssh sin password en linux

Para generar las claves por primera vez, debes **iniciar sesión ssh** como lo haces habitualmente, en tu servidor remoto. Una vez en el sistema, ejecuta el siguiente comando (debes ser **root**).

```
ssh-keygen -t rsa
```

La herramienta nos pregunta donde **guardar la key**. Te recomiendo pulsar **«enter»**, para que se guarde en el sitio definido por defecto.

```
root@SoloLinux:~# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
```

La **passphrase** es una cadena de caracteres, que se utiliza para cifrar la clave privada; no es obligatoria, déjala en blanco y pulsa **«enter»** dos veces.

```
Enter passphrase (empty for no passphrase):
Enter same passphrase again: www.sololinux.es
```

Listo, la key está preparada para su uso.

```
root@SoloLinux:~# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:sT/VXaFr0gR25RPHAJU7VsLkh0Eq6fEGTTLnNVmpHJs root@SoloLinux
The key's randomart image is:
```

```
+---[RSA 3072]---+
  oo0B0.|
  .,.=B *
  + o.=B.*
  @ oEo=.o
  S +. . . .
  . * .o
  .
  .
+---[SHA256]---+
root@SoloLinux:~#
```

**Copiar la key remota a nuestro sistema local**  
Como es evidente... necesitamos la key en nuestro sistema local, para que se pueda verificar en el sistema remoto. Existen tres formas de hacer esta operación.

**Comando ssh-copy-id**  
Este método es el más común.

```
ssh-copy-id -i id_rsa.pub root@ip-remota
```

**Con el comando cat**  
En este caso nos aprovechamos de las múltiples funciones del **comando cat**.

```
cat ~/.ssh/id_rsa.pub | ssh root@ip-remota "mkdir -p ~/.ssh && cat >>
~/.ssh/authorized_keys"
```

**Copiar la key manualmente**  
Es el menos recomendable, pero si tienes problemas con otros métodos...

```
cat ~/.ssh/id_rsa.pub
root@SoloLinux:~# cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQg0DpnhqPgZb5YcuxbTB5GqXX8R860SxvEhI514D5Hmxz6ct5KxLxh
t2EGCS9+vd12Yz0-JM2vUfH4wLe6B:80b0lxr080Y0pt08M-SH0N17cnbqHDpd6PYG9BcW1X0j1YhBnYyC2yukR
3B6AgnJ24Dp070mYFqT/Zen0TcFG24k80Tcy2s7WA+50m3Mx1BRmmBt23grw1GCjFDxtYJvZDhqr0VpnsWx6PBu
djrlna1wh0eGA511fu00s1wvaLGLaYlulLboJ3uW+MdTI8T/xoIQgJK4wt56+2utK2Xao1Va10twZnt1XVeSTHmU
A1u5Rc4MouYFehl4/JkTx4YRb2E7FVA+Ys48h2YkR/nz1NqT4IneS230AmLITnOQLrXsFK9CjUWvR8kU1dFU100V
EuEdana2uAZyD30SN7sy3Cc4u04p6CnMUD2PzLzYNg9e66rLfk1ppcIKEQA30bXUfU0jtcBL/P21zrwV9bRp0YEz
/d7xcPG0BaTXaoMTVMU09ck= root@SoloLinux
root@SoloLinux:~# www.sololinux.es
```

**Nota final**  
Al generar la key con el comando...

```
ssh-keygen -t rsa
```

Se crea una clave que de forma predeterminada es de 2048 bits. Si te parece poco y quieres ampliar seguridad, puedes generarla de 4096 bits, con el siguiente comando.

```
ssh-keygen -t rsa -b 4096
```

# 10 Consejos para proteger un servidor linux

De forma predeterminada, los servidores Linux tienen un alto grado de seguridad. Por este motivo, la gran mayoría de **servidores web** usan linux, y si hablamos de **supercomputadoras** se dispara al 100%.

Como todo en la vida, nada es perfecto, siempre hay algo que pueden salir mal si no se toman las precauciones adecuadas. Debido a la naturaleza de linux (**open source**), es algo lógico que aparezcan vulnerabilidades, aunque rápidamente se lanzan parches de seguridad.

Además de contar con la propia solidez del **núcleo**, tenemos muchas herramientas de seguridad que no podemos olvidar si queremos proteger un servidor linux. Es imprescindible que los administradores de sistemas y usuarios en general, tomen las debidas precauciones y sigan algunos pasos que en este artículo recomendamos.

## 10 consejos para proteger un servidor linux

Cuando encontramos un servidor o sistema de escritorio comprometido, no te engañes, el 99,99% de las veces es culpa del desconocimiento de un usuario, o de algún **sysadmin** poco experimentado. Cuando analizas el problema, te das cuenta de que la prevención era muy simple. En este artículo usamos Ubuntu 20.04 server, pero es válido para la mayoría de **distribuciones linux**. Comenzamos.

### Configurar un usuario no root

Al contrario que en las instalaciones de escritorio, al instalar una distribución de servidor, esta viene con el usuario root configurado y listo para iniciar sesión. Una vez que el **usuario root** inicia sesión, no necesita ningún permiso extra para realizar cualquier acción en el servidor.

Esta práctica se considera insegura, si quieres **proteger un servidor** es mucho mejor configurar algún usuario en el **«grupo sudo»**, que requiera autenticarse si quieres realizar cambios importantes en el servidor.

Puedes agregar un nuevo usuario con el siguiente comando.

```
adduser [usuario]
```

```
root@SoloLinux:~# adduser usuario
Adding user `usuario' ...
Adding new group `usuario' (1001) ...
Adding new user `usuario' (1001) with group `usuario' ...
Creating home directory `/home/usuario' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for usuario
Enter the new value, or press ENTER for the default
Full Name []: sololinux
Room Number []: sololinux
Work Phone []:
Home Phone []:
Other []:
Is the information correct? [Y/n] y
root@SoloLinux:~#
```

Para poder realizar tareas administrativas, debes agregar el nuevo [usuario] al grupo sudo.

```
usermod -aG sudo [usuario]
```

Actualiza. La próxima vez que reinicies el sistema podrás acceder con el nuevo usuario.

```
sudo apt upgrade
```

### Proteger el inicio de sesión

Los nuevos usuarios suelen establecer contraseñas fáciles de recordar, pero a la vez inseguras. Debes cerciorarte de configurar contraseñas. Una combinación de números, letras y símbolos, es buena idea. Si puedes generar keys de acceso tal como explicamos en el **artículo anterior**, mejor que mejor.



## Mantente actualizado

Las distribuciones Linux reciben actualizaciones de seguridad constantemente. No te dé pereza, debes buscar actualizaciones por lo menos una vez al día; Un **administrador de sistemas** no debe ser perezoso, pero por si acaso puedes habilitar las actualizaciones automáticas, todas las distribuciones lo permiten.

## Aplicaciones y servicios innecesarios

Este punto debemos tenerlo muy claro, no es lo mismo un servidor que una máquina de escritorio. En el servidor solo debes tener instaladas las aplicaciones, herramientas y servicios que necesitas. Todo paquete que no utilizas, es un potencial problema de seguridad en el futuro.

Si quieres proteger un servidor correctamente, debes analizar la función de cada paquete instalado. En el caso de sistemas de escritorio es diferente, pues sus múltiples funciones son muchas y variadas.

## Cierra los puertos que no usas

Cada **puerto del sistema** permite un tipo de tráfico en particular. Es un error común, pero a la vez garrafal, no revisar los puertos que están abiertos después de **instalar linux**. Cada puerto que tengas abierto de forma innecesaria, supone una grave amenaza de seguridad.

Existen bots lanzados por piratillas (mejor dicho **lamers**), que escanean continuamente rangos completos de ip buscando máquinas con puertos abiertos, para intentar colar algún **exploit** o cualquier otro bicho en tu sistema. Si eres un usuario novel, te recomiendo que cierres todos los puertos desde tu firewall y, vayas abriendo los que necesitas realmente. Ojo con los paneles de control web estilo **Cpanel**, **Plesk**, etc, suelen abrir puertos de control para conectarse ellos.

## Instala y configura fail2ban

**Fail2Ban** es una herramienta programada para analizar y contabilizar los registros del sistema, según el número y tipo de intento de acceso, bloquea las ip al considerarlas ataques maliciosos. **Fail2Ban** también bloquea los ataques de fuerza bruta que intentan conseguir las contraseñas de forma repetitiva.

En [sololinux.es](http://sololinux.es) tenemos muchos artículos sobre esta herramienta, pulsa [aquí](#) para verlos.



## Configurar 2FA

La autenticación de dos factores también conocida como **2FA**, es una capa adicional de seguridad para autenticar usuarios en el servidor. Si usas un panel de control web en tu servidor, no te resultara difícil implantarlo, la mayoría tienen sus propios plug-ins.

**2FA** requiere de una contraseña adicional de un solo uso, que se envía a tu smartphone o también por mail. El usuario solo podrá acceder después de ingresar el nombre de usuario, su contraseña y la key de un solo uso. 10 consejos para proteger un servidor linux.

## Modificar el puerto SSH

Si quieres proteger tu servidor, te sugiero modificar los puertos predeterminados que son continuamente rastreados. SSH y FTP se llevan la palma. SSH se usa para acceder al servidor remoto y, FTP es un protocolo de transferencia de archivos bidireccional contra una máquina remota. Puedes aprender a modificar el puerto SSH, [en este artículo](#).

## Usa SpamAssassin

Si también tienes un servidor de correo electrónico, que mejor herramienta que **SpamAssassin**. Esta escanea todos los mails, bloqueando los que detecte como maliciosos. Es una buena forma de proteger un servidor. La mayoría de las distribuciones linux lo tienen en sus repositorios, por tanto su instalación es simple. Vemos un ejemplo en Ubuntu 20.04.

```
sudo apt install spamassassin
```

```
root@SoloLinux:~# sudo apt install spamassassin
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
 binutils binutils-common binutils-x86-64-linux-gnu cpp cpp-9 gcc gcc-9
 gcc-9-base libasan5 libatomic1 libauthen-sasl-perl libbinutils libc-dev-bin
 libc6-dev libcc1-0 libcommon-sense-perl libcrypt-dev
 libcrypt-openssl-bignum-perl libcrypt-openssl-random-perl
 libcrypt-openssl-rsa-perl libctf-nobfd0 libctf0 libdigest-bubblebabble-perl
 libdigest-hmac-perl libencode-locale-perl liberror-perl libgcc-9-dev
 libhtml-parser-perl libhtml-tagset-perl libhttp-date-perl
 libhttp-message-perl libio-html-perl libio-socket-inet6-perl
 libio-socket-ssl-perl libisl22 libitm1 libjson-perl libjson-xs-perl liblsan0
 liblwp-mediatypes-perl libmail-authenticationresults-perl libmail-dkim-perl
 libmail-spf-perl libmailtools-perl libmpc3 libnet-dns-perl
 libnet-dns-sec-perl libnet-ip-perl libnet-libidn-perl libnet-smtp-ssl-perl
 libnet-ssleay-perl libnetaddr-ip-perl libperl4-corelibs-perl libquadmath0
 libsocket6-perl libsys-hostname-long-perl libtimedate-perl libtsan0
 libtypes-serializer-perl libubsan1 liburi-perl linux-libc-dev make
 manpages-dev perl-openssl-defaults re2c sa-compile spamc
Suggested packages:
 binutils-doc cpp-doc gcc-9-locales gcc-multilib autoconf automake libtool
 flex bison gdb gcc-doc gcc-9-multilib gcc-9-doc libgssapi-perl glibc-doc
 libdata-dump-perl libwww-perl make-doc razor libdbi-perl pyzor
 libencode-detect-perl libgeoip2-perl libnet-patricia-perl
 libbsd-resource-perl
The following NEW packages will be installed:
 binutils binutils-common binutils-x86-64-linux-gnu cpp cpp-9 gcc gcc-9
 gcc-9-base libasan5 libatomic1 libauthen-sasl-perl libbinutils libc-dev-bin
 libc6-dev libcc1-0 libcommon-sense-perl libcrypt-dev
 libcrypt-openssl-bignum-perl libcrypt-openssl-random-perl
 libcrypt-openssl-rsa-perl libctf-nobfd0 libctf0 libdigest-bubblebabble-perl
 libdigest-hmac-perl libencode-locale-perl liberror-perl libgcc-9-dev
 libhtml-parser-perl libhtml-tagset-perl libhttp-date-perl
```

www.sololinux.es

Una vez concluya la instalación, te sugiero que actualices las reglas con este comando.

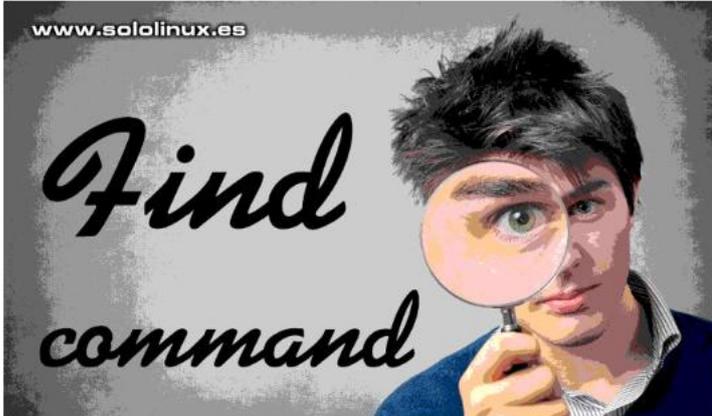
```
sudo sa-update
```

## Auditoría de seguridad

La seguridad no es revisar hoy y se acabó. Debes auditar la seguridad de tu sistema por lo menos una vez a la semana, sobre todo si hablamos de un servidor en producción. No es común, pero puede suceder que algún servicio importante no inicie con el sistema, asegúrate de que las herramientas de seguridad indispensables como **fail2ban** están activas.

Supervisa los dispositivos de almacenamiento, el uso de la memoria y los registros del sistema de forma manual. Debes estar siempre atento a los posibles ataques, así como a los recursos consumidos por el sistema, pequeños detalles pueden delatar una intrusión o un fallo de seguridad.

# Uso del Comando find en linux



El comando **find** en Linux es una herramienta en línea de comandos, con la capacidad de recorrer íntegramente una jerarquía de archivos, buscando directorios y archivos según las opciones que le hemos indicado.

Su uso es amplio, por ejemplo, puedes ordenar búsquedas precisas y posteriormente realizar operaciones sobre lo encontrado. Admite búsquedas que otros comandos no soportan, por archivo, carpeta, nombre, fecha de creación, fecha de modificación, propietario y permisos, realmente impresionante.

El comando **find** admite el uso de «**-exec**», de manera que puedas ejecutar algún **comando** sobre el archivo o directorio localizado en la búsqueda.



## Uso del comando find en linux

En el artículo de hoy vemos algunos ejemplos de uso, pero antes conviene conocer sus interesantes opciones (las más habituales). Las vemos.

- **-exec [comando]** : Ejecuta un comando si se cumplen con los criterios establecidos.
- **-ok [comando]** : Lo mismo que -exec pero solicitando un usuario.
- **-inum [número de inodo]** : Busca archivos por el número de inodo.
- **-links [enlaces]** : Buscar archivos por enlaces.
- **-name [archivo]** : Especificar el archivo a buscar.
- **-newer [archivo]** : Buscar archivos creados o modificados después de la fecha del indicado.
- **-perm [octal]** : Buscar archivos con el permiso octal.
- **-print** : Imprime la ruta de los archivos encontrados.
- **-empty** : Busca directorios y archivos sin contenido.
- **-user name** : Buscar los archivos del usuario, que podemos definir por nombre o por su ID.

### Ejemplos de uso del comando find

#### Buscar archivos

Todos los archivos del directorio raíz (/).

```
find / -type f
```

Todos los archivos de un directorio en particular, en nuestro caso «**/etc**».

```
find /etc/ -type f
```

```
sergio@sololinux:~$ find /etc/ -type f
/etc/xdg/tumbler/tumbler.rc
/etc/xdg/user-dirs.conf
/etc/xdg/menus/xfce-settings-manager.menu
/etc/xdg/menus/gnome-applications.menu
/etc/xdg/menus/xfce-applications.menu
/etc/xdg/Thunar/uca.xml
/etc/xdg/Trolltech.conf
/etc/xdg/autostart/orca-autostart.desktop
/etc/xdg/autostart/blueberry-obex-agent.desktop
/etc/xdg/autostart/indicator-sound.desktop
/etc/xdg/autostart/org.gnome.SettingsDaemon.MediaKeys.desktop
/etc/xdg/autostart/org.gnome.SettingsDaemon.Rfkill.desktop
/etc/xdg/autostart/org.gnome.SettingsDaemon.Smartcard.desktop
/etc/xdg/autostart/light-locker.desktop
/etc/xdg/autostart/org.gnome.SettingsDaemon.Mouse.desktop
/etc/xdg/autostart/org.gnome.SettingsDaemon.allySettings.desktop
/etc/xdg/autostart/pulseaudio.desktop
/etc/xdg/autostart/org.gnome.SettingsDaemon.Datetime.desktop
```

#### Buscar directorios

Todos los directorios de la raíz del sistema «**/**».

```
find / -type d
```

Buscar los directorios de un directorio en particular, en nuestro caso de «**/etc**».

```
find /etc/ -type d
```

```
sergio@sololinux:~$ find /etc/ -type d
/etc/
/etc/xdg
/etc/xdg/tumbler
/etc/xdg/menus
/etc/xdg/menus/applications-merged
/etc/xdg/systemd
/etc/xdg/Thunar
/etc/xdg/autostart
/etc/xdg/xfce4
/etc/xdg/xfce4/xfconf
/etc/xdg/xfce4/xfconf/xfce-perchannel-xml
/etc/xdg/xfce4/panel
/etc/xdg/xfce4/whiskermenu
/etc/rc5.d
/etc/dhcp
/etc/dhcp/dhclient-enter-hooks.d
/etc/dhcp/dhclient-exit-hooks.d
/etc/appport
/etc/appport/crashdb.conf.d
/etc/appport/blacklist.d
/etc/appport/native-origins.d
/etc/depmod.d
```

www.sololinux.es

#### Buscar archivos por nombre

Si conoces la ruta, nombre o extensión, la búsqueda es muy rápida. Vemos un ejemplo en el que añadimos sudo para evitar que nos bloquee archivos restringidos con la extensión sh.

```
sudo find /home/sergio -iname *.sh
```

```
sergio@sololinux:~$ sudo find /home/sergio -iname *.sh
/home/sergio/libera.sh
/home/sergio/Escritorio/libera.sh
/home/sergio/Escritorio/sololinux/libera.sh
/home/sergio/backs/000-PenDRIVE/libera.sh
/home/sergio/scripts/liberados.sh
/home/sergio/scripts/demo.sh
/home/sergio/scripts/sololinux.sh
/home/sergio/scripts/aporte.sh
/home/sergio/scripts/anal-remote.sh
```

Con «**-name**», también funciona.

```
sudo find /home/sergio -name libera.sh
```

## Excluir archivos o extensiones de la búsqueda

En este caso excluimos de la búsqueda todos los archivos con **extensión txt**.

```
find /etc -not -name "*.txt"
```

```
sergio@sololinux:~$ find /etc -not -name "*.txt"
/etc
/etc/xdg
/etc/xdg/xdg-default.desktop
/etc/xdg/tumbler
/etc/xdg/tumbler/tumbler.rc
/etc/xdg/user-dirs.conf
/etc/xdg/menus
/etc/xdg/menus/gnome-applications-merged
/etc/xdg/menus/applications-merged
/etc/xdg/menus/xfce-settings-manager.menu
/etc/xdg/menus/gnome-applications.menu
/etc/xdg/menus/xfce-applications.menu
```

## Buscar archivos o directorios por tamaño

Si conoces el tamaño específico de un archivo o directorio, puedes encontrarlo con este comando.

```
find /etc -type f -size 5M
```

## Buscar archivos por permisos

Esta opción es realmente interesante. Ampliamente utilizada por usuarios y herramientas automatizadas, que deben verificar los permisos.... por ejemplo de sitios web. En nuestro caso buscamos los archivos con **permisos 0644**.

```
find /etc -type f -perm 0644
```

Otra opción para buscar archivos de solo lectura, con el **comando find**, es...

```
find /etc -type f -perm /u=r
```

```
sergio@sololinux:~$ find /etc -type f -perm 0644
/etc/xdg/tumbler/tumbler.rc
/etc/xdg/user-dirs.conf
/etc/xdg/menus/xfce-settings-manager.menu
/etc/xdg/menus/gnome-applications.menu
/etc/xdg/menus/xfce-applications.menu
/etc/xdg/Thunar/uca.xml
/etc/xdg/Trolltech.conf
/etc/xdg/autostart/orca-autostart.desktop
/etc/xdg/autostart/blueberry-obex-agent.desktop
/etc/xdg/autostart/indicator-sound.desktop
/etc/xdg/autostart/org.gnome.SettingsDaemon.MediaKeys.desktop
/etc/xdg/autostart/org.gnome.SettingsDaemon.Rfkill.desktop
/etc/xdg/autostart/org.gnome.SettingsDaemon.Smartcard.desktop
/etc/xdg/autostart/light-locker.desktop
```

## Buscar archivos por usuario o grupo

Para buscar archivos por usuario, te recomiendo utilizar **sudo**. Si no usas **sudo**, talvez se omitan algunos archivos.

```
sudo find / -user sergio
```

```
sergio@sololinux:~$ sudo find / -user sergio
[sudo] contraseña para sergio:
/var/tmp/flatpak-cache-NBTLR0
/var/tmp/flatpak-cache-8J8VY0
/var/tmp/flatpak-cache-395WS0
/var/tmp/flatpak-cache-0Z6J50
/var/tmp/flatpak-cache-2A05Q0
/var/tmp/flatpak-cache-4AF4R0
/var/tmp/flatpak-cache-T0DUX0
/var/tmp/flatpak-cache-G2ZDZ0
/var/tmp/flatpak-cache-XI0FT0
/var/tmp/flatpak-cache-KDLNQ0
```

También podemos buscar archivos por el grupo de usuario.

```
find / -group sergio
```

## Buscar archivos por tiempo de acceso o modificación

En las siguientes cadenas de comandos, puedes insertar el valor numérico que más te convenga. Por ejemplo... para buscar archivos que han sido modificados hace un máximo de siete días, ejecutamos este **comando** (recomiendo utilizar **sudo**).

```
sudo find / -mtime 7
```

```
sergio@sololinux:~$ sudo find / -mtime 7
[sudo] contraseña para sergio:
/var/backups/alternatives.tar.2.gz
find: '/run/user/1000/gvfs': Permiso denegado
/home/sergio/.local/share/TelegramDesktop/tdata/user_data/media_cache/0/6D
/home/sergio/.local/share/TelegramDesktop/tdata/user_data/media_cache/0/6D/EEB80EB36083
/home/sergio/.local/share/TelegramDesktop/tdata/user_data/media_cache/0/75
/home/sergio/.local/share/TelegramDesktop/tdata/user_data/media_cache/0/49
/home/sergio/.local/share/TelegramDesktop/tdata/user_data/media_cache/0/2E
```

Ahora los que han cambiado hace menos de 5 minutos.

```
sudo find / -mmin -5
```

También los que han sido modificados. Esta opción es similar a la anterior (no igual).

```
sudo find / -mtime -5
```

Por último, listamos los archivos a los cuales accediste en los últimos seis minutos.

```
sudo find / -amin -6
```

```
sergio@sololinux:~$ sudo find / -amin -6
[sudo] contraseña para sergio:
/var/spool/anaconda/cron.daily
/var/spool/anaconda/cron.weekly
/var/spool/anaconda/cron.monthly
/var/lib/systemd/timers/stamp-anaconda.timer
/usr/lib/firefox-addons/extensions/langpack-es-AR@firefox.mozilla.org.xpi
/usr/lib/firefox-addons/extensions/langpack-es-ES@firefox.mozilla.org.xpi
/usr/lib/x86_64-linux-gnu/libasyncln.so.0.3.1
/usr/lib/x86_64-linux-gnu/libGL.so.1.7.0
/usr/lib/x86_64-linux-gnu/libsqlite3.so.0
/usr/lib/x86_64-linux-gnu/libGLdispatch.so.0.0.0
/usr/lib/x86_64-linux-gnu/pulseaudio/libpulsecommon-11.1.so
/usr/lib/x86_64-linux-gnu/libGLX.so.0
/usr/lib/x86_64-linux-gnu/libpulse.so.0.20.2
/usr/lib/x86_64-linux-gnu/libsndfile.so.1
/usr/lib/x86_64-linux-gnu/libdbusmenu-glib.so.4.0.12
```

## Buscar archivos y ejecutar un comando

En este ejemplo de uso, buscamos todos los archivos de un usuario y, los listamos con el «**comando ls -ld**».

```
find /home/sergio -exec ls -ld {} \;
```

```
sergio@sololinux:~$ find /home/sergio -exec ls -ld {} \;
drwxr-xr-x 43 sergio sergio 4096 mar 21 11:45 /home/sergio
-rw-r--r-- 1 sergio sergio 23 sep 2 2020 /home/sergio/.dmrc
-rw-r--r-- 1 sergio sergio 22 sep 2 2020 /home/sergio/.gtkrc-2.0
-rw-r--r-- 1 sergio sergio 1986 dic 7 15:35 /home/sergio/libera.sh
-rw-rw-r-- 1 sergio sergio 25294 mar 18 21:36 /home/sergio/Database.kdbx
drwxr-xr-x 8 root root 4096 sep 2 2020 /home/sergio/rtl8723de
-rw-r--r-- 1 root root 34961 sep 2 2020
/home/sergio/rtl8723de/hal/hal_intf.c
-rw-r--r-- 1 root root 82905 sep 2 2020 /home/sergio/rtl8723de/hal/hal_mp.c
drwxr-xr-x 2 root root 4096 sep 2 2020 /home/sergio/rtl8723de/hal/hal_hci
-rw-r--r-- 1 root root 958 sep 2 2020
/home/sergio/rtl8723de/hal/hal_hci/hal_pci.c
drwxr-xr-x 4 root root 4096 sep 2 2020 /home/sergio/rtl8723de/hal/phydm
```

Otra opción interesante y como último ejemplo del artículo, es buscar todos los archivos de un directorio con una extensión en particular, y borrarlos con el **comando rm**. En nuestro ejemplo, eliminamos todos los **archivos txt** del directorio «**/etc**», de un plumazo.

```
sudo find /etc -type f -name *.txt -size -1M -exec rm -f {} \;
```

## Ayuda del comando find

Al igual que otros comandos de Unix/Linux, find, también viene con su ayuda integrada.

```
find -help
```

Se imprime algo similar a...

```
sergio@sololinux:~$ find -help
Usage: find [-H] [-L] [-P] [-Olevel] [-D debugopts] [path...] [expression]

la ruta de acceso predeterminada es el directorio actual; la expresión predeterminada es -print
la expresión puede ser: operadores, opciones, pruebas y acciones:
operadores (prioridad decreciente; se supone -and si no hay otros):
    ( EXPR ) ! EXPR -not EXPR EXPR1 -a EXPR2 EXPR1 -and EXPR2
    EXPR1 -o EXPR2 EXPR1 -or EXPR2 EXPR1 , EXPR2
opciones de posición (siempre verdaderas): -daystart -follow -regextype

opciones normales (siempre verdaderas, se ponen después de otras expresiones):
    -depth --help -maxdepth NIVELES -mindepth NIVELES -mount -noleaf
    --version -xdev -ignore_readdir_race -noignore_readdir_race
evaluaciones (N puede ser +N, -N o N): -amin N -anewer FICHERO -atime N
    -cmin N -cnewer ARCHIVO -ctime N -empty -false -fstype TIPO -gid N
    -group NOMBRE -ilname EXPR-REG -iname EXPR-REG -inum N
    -iwholename EXPR-REG -iregex EXPR-REG -links N -lname EXPR-REG
    -mmin N -mtime N -name EXPR-REG -newer ARCHIVO
    -nouser -nogroup -path PATTERN -perm [-/]MODE -regex PATTERN
    -readable -writable -executable
    -wholename PATTERN -size N[bcwkMG] -true -type [bcdpflsD] -uid N
    -used N -user NAME -xtype [bcdpfls] -context CONTEXTO

actions: -delete -print0 -printf FORMAT -fprintf FILE FORMAT -print
    -fprintf0 FILE -fprintf FILE -ls -fls FILE -prune -quit
    -exec COMMAND ; -exec COMMAND {} + -ok COMMAND ;
    -execdir COMMAND ; -execdir COMMAND {} + -okdir COMMAND ;

Valid arguments for -D:
exec, help, opt, rates, search, stat, time, tree
Use '-D help' for a description of the options, or see find(1)

Please see also the documentation at http://www.gnu.org/software/findutils/.
You can report (and track progress on fixing) bugs in the "find"
program via the GNU findutils bug-reporting page at
https://savannah.gnu.org/bugs/?group=findutils or, if
```

[www.sololinux.es](http://www.sololinux.es)

[www.sololinux.es](http://www.sololinux.es)

### ¿Te interesa colaborar con SoloLinux?

Puedes enviarnos tus ideas, propuestas, artículos, opiniones a nuestra **dirección de correo:**  
[adrian@sololinux.es](mailto:adrian@sololinux.es)

También puedes contactar con nosotros mediante **TELEGRAM** en el canal  
[@sololinux](https://t.me/sololinux).

### Publicidad:

Quieres poner publicidad en la revista, ahora puedes hacerlo de forma muy simple, llegando a todo el mundo con esta revista digital gratuita de software libre y GNU/Linux en ESPAÑOL

### CON SOLOLINUX MULTIPLICARAS TUS CLIENTES

Para mayor información escribe un email a:  
[adrian@sololinux.es](mailto:adrian@sololinux.es)



## Uso del Comando awk en linux

El **comando awk**, ofrece un tipo de lenguaje de programación específico para procesar texto o caracteres. Usado ampliamente para manipular datos y generar informes, el **comando awk** no requiere compilación. Destacamos que se permite el uso de variables, también funciones numéricas, funciones de cadena y operadores lógicos.

En definitiva, el comando awk es un lenguaje de secuencias de comandos muy poderoso, con el cual puedes...

- Definir variables
- Usar cadenas y operadores aritméticos
- Hacer uso de control de flujo y ciclos
- Generar reportes con formato



*En este artículo, haremos una pequeña introducción en el mundo de awk.*

```
if(PATTERN1){...print something..}
```

www.sololinux.es

## Uso del comando awk en linux

La **sintaxis del comando AWK** es un poco diferente si la comparamos con los comandos habituales, aun así, no es tan difícil como parece en un principio.

```
awk [options] [program] [file]
```

Estas son las opciones:

Opción	Su uso
-F / fs	Especificar el separador de archivos
-f / file	Indicar el archivo con el script awk
-v / var=value	Declarar una variable

### Leer un script awk

Para definir un script awk, debes incluirlo entre comillas dobles que a su vez (el conjunto), estará entre comillas simples. Como ejemplo imprimimos un texto.

```
awk '{print "Bienvenido a SoloLinux "}'
```

Para salir del modo programación, pulsa las teclas «**Ctrl + d**». Vemos un ejemplo del resultado.

```
sergio@sololinux:~$ awk '{print "Bienvenido a SoloLinux "}'
```

```
Bienvenido a SoloLinux
sergio@sololinux:~$
```

## Usar variables en awk

Agregar variables para extraer campos de un archivo de texto es tarea sencilla. Por ejemplo...

- **\$0** – Todos los campos de la línea.
- **\$1** – El primer campo.
- **\$2** – El segundo campo.
- **\$3** – El tercer campo.

Tenemos un archivo de texto denominado «**demo.txt**», con este contenido.

```
Yo uso Arch Linux
Yo uso Debian
Yo uso Alpine
Yo uso Slackware
```

A modo de ejemplo, vemos el resultado de las cuatro variables anteriores.

```
sergio@sololinux:~$ awk '{print $0}' demo.txt
Yo uso Arch Linux
Yo uso Debian
Yo uso Alpine
Yo uso Slackware
sergio@sololinux:~$ awk '{print $1}' demo.txt
Yo
Yo
Yo
Yo
sergio@sololinux:~$ awk '{print $2}' demo.txt
uso
uso
uso
uso
sergio@sololinux:~$ awk '{print $3}' demo.txt
Arch
Debian
Alpine
Slackware
```

Si observas detenidamente el ejemplo anterior, los espacios en blanco o tabulaciones son los separadores por defecto de **awk**. Entonces... ¿Qué sucede cuando los separadores por defectos son otros caracteres?. Vemos un ejemplo claro en «**/etc/passwd**».

```
cat /etc/passwd
```

En este caso, los campos están delimitados por dos puntos «:».

```
sergio@sololinux:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
```

Forzamos el separador correspondiente con «**-F:**» (no olvides la variable).

```
awk -F: '{print $1}' /etc/passwd
```

```
sergio@sololinux:~$ awk -F: '{print $1}' /etc/passwd
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
```

## Uso de varios comandos a la vez

Puedes agregar varios comandos a la vez. Nosotros ejecutamos el **comando echo** con dos campos, a continuación... con **awk** y su variable sustituimos el segundo campo del comando echo, por el campo insertado en **awk**, e imprimimos el resultado total. Soy consciente que puede parecer un poco lioso, pero te aseguro que no lo es.

```
echo "Hola Sergio" | awk '{$2="SoloLinux"; print $0}'
```

```
sergio@sololinux:~$ echo "Hola Sergio" | awk '{$2="SoloLinux"; print $0}'
Hola SoloLinux
sergio@sololinux:~$
```

## Pre-procesar con awk

Con esta acción podemos crear títulos a la hora de imprimir archivos. Nos ayudamos de «**BEGIN**» y del archivo **demo.txt** que creamos anteriormente.

```
awk 'BEGIN {print "Que linux usas:"}
> {print $0}' demo.txt
```

```
sergio@sololinux:~$ awk 'BEGIN {print "Que linux usas:"}
> {print $0}' demo.txt
Que linux usas:
Yo uso Arch Linux
Yo uso Debian
Yo uso Alpine
Yo uso Slackware
sergio@sololinux:~$
```

[www.sololinux.es](http://www.sololinux.es)

## Variables integradas

Junto al **separador FS**, anteriormente vimos las variables de campo para extraer datos, \$1, \$2, \$3, etc. Debes saber que no son las únicas, también tenemos a nuestra disposición otras variables integradas, las más comunes son las de la siguiente tabla.

Variable	Uso
FIELDWIDTHS	Permite especificar el ancho del campo
RS	Permite especificar el separador de campos
FS	Permite especificar un separador de campos
ORS	Permite especificar el separador de salida
OFS	Permite especificar un separador de salida

De forma predeterminada, la variable **OFS** equivale al espacio, haciendo uso de ella puedes seleccionar el separador que tú quieras. Vemos un ejemplo.

```
awk 'BEGIN{FS=":"; OFS="-"} {print $1,$2,$3,$4,$5}' /etc/passwd
```

```
sergio@sololinux:~$ awk 'BEGIN{FS=":"; OFS="-"} {print $1,$2,$3,$4,$5}' /etc/passwd
root-x-0-0-root
daemon-x-1-1-daemon
bin-x-2-2-bin
sys-x-3-3-sys
sync-x-4-65534-sync
games-x-5-60-games
man-x-6-12-man
lp-x-7-7-lp
mail-x-8-8-mail
news-x-9-9-news
```

En el caso de que existan varios tipos de separadores podemos usar la variable **FIELDWIDTHS**, que además nos ayudara a delimitar el tamaño de los campos, en el caso que sean desiguales.

```
awk 'BEGIN{FIELDWIDTHS="3 4 3"}{print $1,$2,$3}' demo.txt
```

```
sergio@sololinux:~$ awk 'BEGIN{FIELDWIDTHS="3 4 3"}{print $1,$2,$3}' demo.txt
Yo uso Arc
Yo uso Deb
Yo uso Alp
Yo uso Sla
sergio@sololinux:~$
```

## Comandos estructurados

Una de las particularidades del **comando awk**, es su lenguaje de scripting que por ejemplo incluye soporte para **sentencias condicionales if**. Supongamos que tenemos un archivo con el siguiente contenido.

```
9
180
3
27
49
31
```

Ahora imprimimos los números con valor superior a 25.

```
awk '{if ($1 > 25) print $1}' dem01.txt
```

```
sergio@sololinux:~$ awk '{if ($1 > 25) print $1}' dem01.txt
180
27
49
31
sergio@sololinux:~$ www.sololinux.es
```

Usamos llaves para ejecutar varias sentencias.

```
awk '{
>
> if ($1 > 25)
> {
> x = $1 * 3
> print x
> }
> }' dem01.txt
```

```
sergio@sololinux:~$ awk '{
>
> if ($1 > 25)
> {
> x = $1 * 3
> print x
> }
> }' dem01.txt
540
81
147
93
sergio@sololinux:~$
```

También agregamos una «else».

```
awk '{
>
> if ($1 > 25)
> {
> x = $1 * 3
> print x
> } else
> {
> x = $1 / 2
> print x
> }
> }' dem01.txt
```

```
sergio@sololinux:~$ awk '{
>
> if ($1 > 25)
> {
> x = $1 * 3
> print x
> } else
> {
> x = $1 / 2
> print x
> }
> }' dem01.txt
4.5
0
540
0
1.5
0
81
0
147
0
93
sergio@sololinux:~$
```

## Bucle While

El **bucle While**, es un ciclo repetitivo basado en los resultados de una expresión lógica. Puedes usarlo para iterar sobre los datos con una condición.

```
awk '{
>
> sum = 0
>
> i = 1
>
> while (i < 8)
> {
> sum += $i
> i++
> }
>
> average = sum / 4
>
> print "Average:",average
> }' dem01.txt
```

```
sergio@sololinux:~$ awk '{
>
> sum = 0
>
> i = 1
>
> while (i < 8)
> {
> sum += $i
> i++
> }
>
> average = sum / 4
>
> print "Average:",average
> }' dem01.txt
Average: 2.25
Average: 0
Average: 45
Average: 0
Average: 0.75
Average: 0
Average: 6.75
Average: 0
Average: 12.25
Average: 0
Average: 7.75
sergio@sololinux:~$ www.sololinux.es
```

## Bucle For

También se admiten sentencias de **bucles for**.

```
awk '{
>
> total = 0
> for (var = 1; var < 8; var++)
> {
> {
> total += $var
> }
> }
> avg = total / 4
> print "Average:",avg
> }' demo1.txt
```

## Funciones Matemáticas

Se permite el uso de las siguientes funciones matemáticas:

- **sin(x)**
- **cos(x)**
- **sqrt(x)**
- **exp(x)**
- **log(x)**
- **rand()**

```
awk 'BEGIN{x=cos(10); print x}'
```

```
sergio@sololinux:~$ awk 'BEGIN{x=cos(10); print x}'
-0.839072
```

## Funciones String

Tienes a tu disposición muchas **funciones string** (de cadena de caracteres), en nuestro caso usaremos «**toupper**», que convierte el texto a texto en mayúsculas.

```
awk 'BEGIN{x = "sololinux"; print toupper(x)}'
```

Obtenemos la palabra **sololinux** en mayúsculas.

```
sergio@sololinux:~$ awk 'BEGIN{x = "sololinux"; print toupper(x)}'
SOLOLINUX
sergio@sololinux:~$
```

## ¿Te interesa colaborar con SoloLinux?

Puedes enviarnos tus ideas, propuestas, artículos, opiniones a nuestra **dirección de correo**:  
adrian@sololinux.es

También puedes contactar con nosotros mediante **TELEGRAM** en el canal @sololinux.

## Dar formato a la salida

Como último apunte de este artículo, vemos como dar formato a la salida del comando awk. La sintaxis es sencilla.

```
%[formato]control-letter
```

Normalmente, el formato se aplica a «printf». Estas son las opciones más comunes.

- **c** – Se imprimen las salidas numéricas como cadena de caracteres.
- **d** – Valores enteros.
- **e** – Se imprime numeración científica.
- **f** – Valores numéricos incluyendo los decimales.
- **o** – Los valores se imprimen en octal.
- **s** – Imprime cadenas de texto.

En nuestro ejemplo necesitamos los decimales.

```
awk 'BEGIN{
>
> x = 123 / 89
>
> printf "El resultado es: %f\n", x
>
> }'
```

```
sergio@sololinux:~$ awk 'BEGIN{
>
> x = 123 / 89
>
> printf "El resultado es: %f\n", x
>
> }'
El resultado es: 1.382022
sergio@sololinux:~$
```

## Publicidad:

Quieres poner publicidad en la revista, ahora puedes hacerlo de forma muy simple, llegando a todo el mundo con esta revista digital gratuita de software libre y GNU/Linux en ESPAÑOL

CON **SOLOLINUX** MULTIPLICARAS TUS CLIENTES

Para mayor información escribe un email a:  
adrian@sololinux.es



## Limpiar linux Con Czkawka de archivos innecesarios



Czkawka es una herramienta gratuita y de código abierto, específica para limpiar tu sistema fácilmente de archivos innecesarios que se van acumulando con el tiempo.

Escrito en **rust** y publicado bajo **licencia MIT**, incluye soporte de caché para un escaneo repetitivo mucho más rápido. Tiene la capacidad de **limpiar linux**, buscando y localizando archivos duplicados, carpetas vacías, archivos demasiado grandes, archivos vacíos, archivos temporales, archivos multimedia e imágenes que sean iguales, además de los que están corruptos, simbólicos inválidos y archivos rotos.

Su uso es bastante simple e intuitivo, por tanto no requiere de ninguna curva de aprendizaje. En el artículo de hoy procedemos a **instalar Czkawka** y, conocer un poco su forma de operar.



# Czkawka

[www.sololinux.es](http://www.sololinux.es)

## Limpiar linux con Czkawka de archivos innecesarios

Lo primero que tienes que hacer es cerciorarte de la última versión disponible, que puedes verificar en su **zona de descargas Github oficial de Czkawka**. A fecha de hoy, la 3.0.0 – 11.03.2021r.

Bien, ya sabemos que a fecha de hoy es la 3.0.0, así que la descargamos con el **comando wget**.

```
wget
https://github.com/qarmin/czkawka/releases/download/3.0.0/linux_czkawka_gui.AppImage
```

```
sergio@sololinux:~$ wget
https://github.com/qarmin/czkawka/releases/download/3.0.0/linux_czkawka_gui.AppImage
--2021-03-24 14:18:26--
https://github.com/qarmin/czkawka/releases/download/3.0.0/linux_czkawka_gui.AppImage
Resolviendo github.com (github.com)... 140.82.121.3
Conectando con github.com (github.com)[140.82.121.3]:443... conectado.
Petición HTTP enviada, esperando respuesta... 302 Found
Ubicación:
https://github-releases.githubusercontent.com/292065578/dab3bf00-82b0-11eb-89ae-b2227c4d770a?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20210324%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20210324T121827Z&X-Amz-Expires=300&X-Amz-Signature=33803e2d43943cd27de5e1d8eefabf10d7bf992ff889d6132890d7481d367621&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=292065578&response-content-disposition=attachment%3B%20filename%3Dlinux_czkawka_gui.AppImage&response-content-type=application%2Foctet-stream [siguiente]
--2021-03-24 14:18:27-- https://github-releases.githubusercontent.com/292065578/dab3bf00-82b0-11eb-89ae-b2227c4d770a?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20210324%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20210324T121827Z&X-Amz-Expires=300&X-Amz-Signature=33803e2d43943cd27de5e1d8eefabf10d7bf992ff889d6132890d7481d367621&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=292065578&response-content-disposition=attachment%3B%20filename%3Dlinux_czkawka_gui.AppImage&response-content-type=application%2Foctet-stream
Resolviendo github-releases.githubusercontent.com (github-releases.githubusercontent.com)... 185.199.109.154, 185.199.110.154, 185.199.111.154, ...
Conectando con github-releases.githubusercontent.com (github-releases.githubusercontent.com)[185.199.109.154]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 27432128 (26M) [application/octet-stream]
Guardando como: "linux_czkawka_gui.AppImage"
linux_czkawka_gui.AppImage 100%
[=====>]
26,16M 4,57MB/s en 6,5s
2021-03-24 14:18:34 (4,01 MB/s) - "linux_czkawka_gui.AppImage" guardado [27432128/27432128]
```

Ahora le concedemos los permisos correspondientes y, ejecutamos la app de la herramienta.

```
sudo chmod +x ~/linux_czkawka_gui.AppImage
./linux_czkawka_gui.AppImage
```

Prácticamente al instante se abre la aplicación para **limpiar linux**, por algo está escrita en **rust**, jeje.

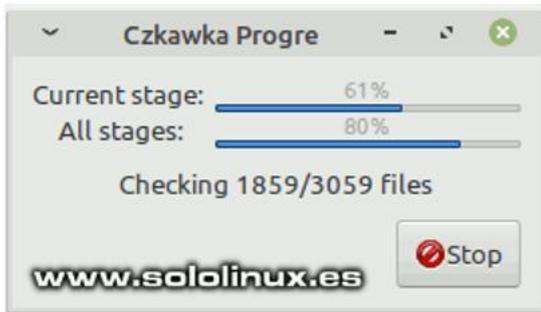


Como podemos observar en la anterior imagen, dispone de una interfaz simple y sencilla. No te dejes engañar por su apariencia, **Czkawka** es muy potente. Como ejemplo nosotros vamos a buscar archivos rotos.

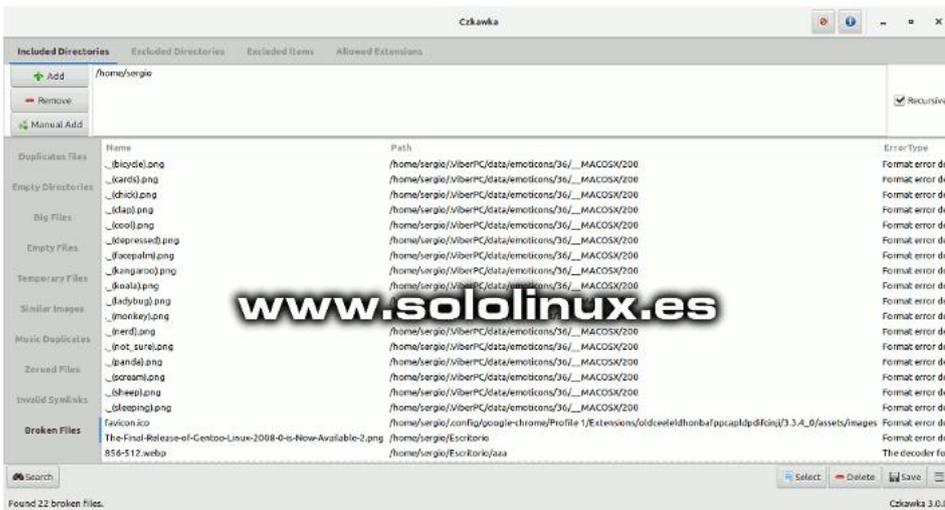
- Selecciona la ruta
- Selecciona lo que te interese buscar
- Pulsa sobre «search»



Para limpiar linux correctamente, Czkawka toma su tiempo en buscar todos los archivos rotos.



Una termina el proceso, analiza visualmente los archivos encontrados y borra los que consideras inútiles pulsando en «Remove».



## ¿Te interesa colaborar con SoloLinux?

Puedes enviarnos tus ideas, propuestas, artículos, opiniones a nuestra **dirección de correo:**  
adrian@sololinux.es

También puedes contactar con nosotros mediante **TELEGRAM** en el canal @sololinux.

## SoloWordPress

- Revista Bimestral
- Manuales
- Noticias
- Temas
- Plugins
- SEO
- Seguridad
- Entra en: [www.solowordpress.es](http://www.solowordpress.es)



# www.sololinux.es

## Reparar un USB dañado sin perder datos

www.sololinux.es



## Reparar un USB dañado sin perder datos

Las unidades de **almacenamiento flash** son algo habitual hoy en día, pequeñas, ligeras y sobre todo muy económicas. Este tipo de dispositivos nos facilitan nuestras labores diarias, al permitirnos un almacenamiento externo de nuestros datos y su transferencia, de forma ágil y eficaz.

Sin embargo, es posible que alguna vez nos encontremos con el caso, que nuestra unidad está dañada o corrupta. Por suerte nosotros usamos **linux** y, tenemos a nuestra disposición muchas herramientas que nos ayudaran a solucionar este problema.

## Reparar un USB dañado sin perder datos

Como ya comentamos anteriormente, comenzamos por extraer los datos. Nosotros usaremos el **comando dd** y **gzip**. Identificamos el dispositivo corrupto con el siguiente comando.

```
ls /dev/disk/by-id
```

En nuestro ejemplo aparece el USB y nuestro disco SSD

```
sergio@sololinux:~$ ls /dev/disk/by-id
ata-LITEON_CV8-8E512-HP_002949101DPC
ata-LITEON_CV8-8E512-HP_002949101DPC-part1
ata-LITEON_CV8-8E512-HP_002949101DPC-part2
usb-_USB_FLASH_DRIVE_07190880872A7198-0:0
usb-_USB_FLASH_DRIVE_07190880872A7198-0:0-part1
sergio@sololinux:~$
```

Toma nota del dispositivo en cuestión (ejemplo – **usb-\_USB\_FLASH\_DRIVE\_07190880872A7198-0:0**), debes indicarlo para realizar la copia. Usamos esta sintaxis.

```
sudo dd if=/dev/disk/by-id/dispositivo-usb status=progress | gzip -c > /ruta/archivo.img.gz
```

Vemos un ejemplo de nuestro comando...

```
sudo dd if=/dev/disk/by-id/usb-_USB_FLASH_DRIVE_07190880872A7198-0:0 status=progress | gzip -c > /home/sergio/BACKUP_USB.img.gz
```

Comienza el volcado de datos...

```
sergio@sololinux:~$ sudo dd if=/dev/disk/by-id/usb-_USB_FLASH_DRIVE_07190880872A7198-0:0 status=progress | gzip -c > /home/sergio/BACKUP_USB.img.gz
[sudo] contraseña para sergio:
319355392 bytes (319 MB, 305 MiB) copied, 26 s, 12,3 MB/s
```

El **backup** íntegro del dispositivo ha concluido.

```
sergio@sololinux:~$ sudo dd if=/dev/disk/by-id/usb-_USB_FLASH_DRIVE_07190880872A7198-0:0 status=progress | gzip -c > /home/sergio/BACKUP_USB.img.gz
[sudo] contraseña para sergio:
31035291136 bytes (31 GB, 29 GiB) copied, 2540 s, 12,2 MB/s
60628992+0 registros leídos
60628992+0 registros escritos
31042043904 bytes (31 GB, 29 GiB) copied, 2540,48 s, 12,2 MB/s
sergio@sololinux:~$
```

La práctica totalidad de dispositivos USB corruptos, deben su error a bloques de archivos defectuosos. Esto lo podemos reparar con uno de los comandos más potentes de linux, el **comando fsck**.

Recordamos la identificación anterior de los dispositivos del sistema.

```
sergio@sololinux:~$ ls /dev/disk/by-id
ata-LITEON_CV8-8E512-HP_002949101DPC
ata-LITEON_CV8-8E512-HP_002949101DPC-part1
ata-LITEON_CV8-8E512-HP_002949101DPC-part2
usb-_USB_FLASH_DRIVE_07190880872A7198-0:0
usb-_USB_FLASH_DRIVE_07190880872A7198-0:0-part1
sergio@sololinux:~$
```

Para reparar un USB dañado con el comando **fsck**, tomaremos la identificación de la partición del dispositivo. En nuestro caso «**usb-\_USB\_FLASH\_DRIVE\_07190880872A7198-0:0-part1**». La sintaxis es fácil de recordar.

```
sudo fsck -v -a /dev/disk/dispositivo-usb
```

```
sudo fsck -v -a /dev/disk/by-id/usb-_USB_FLASH_DRIVE_07190880872A7198-0:0-part1
```

```
sergio@sololinux:~$ sudo fsck -v -a /dev/disk/by-id/usb-_USB_FLASH_DRIVE_07190880872A7198-0:0-part1
[sudo] contraseña para sergio:
fsck de util-linux 2.31.1
fsck.fat 4.1 (2017-01-24)
Checking we can access the last sector of the filesystem
0x41: Dirty bit is set. Fs was not properly unmounted and some data
may be corrupt.
Automatically removing dirty bit.
Boot sector contents:
System ID "mkfs.fat"
Media byte 0xf8 (hard disk)
512 bytes per logical sector
16384 bytes per cluster
32 reserved sectors
First FAT starts at byte 16384 (sector 32)
2 FATs, 32 bit entries
7585792 bytes per FAT (= 14816 sectors)
Root directory start at cluster 2 (arbitrary size)
Data area starts at byte 15187968 (sector 29664)
1893665 data clusters (31025807360 bytes)
32 sectors/track, 64 heads
2048 hidden sectors
60626944 sectors total
Reclaiming unconnected clusters.
Reclaimed 9251 unused clusters (151568384 bytes) in 2 chains.
Checking free cluster summary.
Performing changes.
/dev/sdb1: 44 files, 1275558/1893665 clusters
```

Como norma general ya deberías de tener acceso a los datos del dispositivo, si no es así... recuperamos el backup que realizamos anteriormente.

```
sudo gzip -c /ruta-del-backup | sudo dd of=/dev/disk/by-id/dispositivo-usb status=progress
```

El backup realizado por nosotros...

```
sudo gzip -c /home/sergio/BACKUP_USB.img.gz | sudo dd of=/dev/disk/by-id/usb-_USB_FLASH_DRIVE_07190880872A7198-0:0 status=progress
```

Imagen de ejemplo.

```
sergio@sololinux:~$ sudo gzip -c /home/sergio/BACKUP_USB.img.gz | sudo dd of=/dev/disk/by-id/usb-USB_FLASH_DRIVE_07190880872A7198-0:0 status=progress
474108416 bytes (474 MB, 452 MiB) copied, 335 s, 1,4 MB/s
```

[www.sololinux.es](http://www.sololinux.es)

Si con todo lo realizado sigues sin poder acceder, no queda otra que formatear. Te propongo dos artículos que te ayudaran a realizar el proceso correctamente.

- [Formatear dispositivos USB correctamente – 1](#)
- [Formatear dispositivos USB correctamente – 2](#)

Si el desastre ha sido tal, que no puedes recuperar los datos... solo queda un recurso, armarte de paciencia y usar la herramienta testdisk, no falla.

- [Recuperar datos borrados en linux con testdisk \(photorec\)](#)

**Nota final:** Todos los comandos propuestos en este artículo, están verificados por el equipo de **sololinux**. Aun siendo así, debes actuar con precaución y leer bien los pasos indicados. Nosotros no somos responsables de posibles daños.

*Si algo funciona, no lo toques, si no sabes lo que haces... no lo hagas.*



# TestDisk

[www.sololinux.es](http://www.sololinux.es)

## Publicidad:

Quieres poner publicidad en la revista, ahora puedes hacerlo de forma muy simple, llegando a todo el mundo con esta revista digital gratuita de software libre y GNU/Linux en ESPAÑOL

CON **SOLOLINUX** MULTIPLICARAS TUS CLIENTES

Para mayor información escribe un email a: [adrian@sololinux.es](mailto:adrian@sololinux.es)



# Comprimir imágenes Con Curtail Images Compressor



De la mano de **Hugo Posnic**, nos llega un buen **compresor de imágenes** gratuito y de **código abierto**. **Curtail Image Compressor**, nos permite comprimir archivos JPEG y PNG. Destacamos que admite compresión con pérdida o sin pérdida de calidad.

La **herramienta Curtail** es simple y fácil de usar, simplemente debemos arrastrar y soltar la imagen para que se comprima automáticamente. Entre otras cosas, tienes las opciones de eliminar los **metadatos** y la compatibilidad con el **JPG de codificación progresiva**.



## Comprimir imágenes con Curtail Images Compressor

Te recomiendo instalar la herramienta como **paquete flatpak**. Te evitarás muchos quebraderos de cabeza, además de ser un proceso muy rápido. Ejecuta el siguiente comando.

```
flatpak install flathub com.github.huluti.Curtail
```

Cuando veas lo siguiente en tu consola o terminal, pulsa «Y».

```
sergio@sololinux:~$ flatpak install flathub com.github.huluti.Curtail
Installing in system:
com.github.huluti.Curtail/x86_64/stable      flathub 4537b2ea49ef
permissions: ipc, fallback-x11, wayland, x11
file access: home
com.github.huluti.Curtail.Locale/x86_64/stable flathub 196cd51c2251
Is this ok [y/n]:
```

```
sergio@sololinux:~$ flatpak install flathub com.github.huluti.Curtail
Installing in system:
com.github.huluti.Curtail/x86_64/stable      flathub 4537b2ea49ef
permissions: ipc, fallback-x11, wayland, x11
file access: home
com.github.huluti.Curtail.Locale/x86_64/stable flathub 196cd51c2251
Is this ok [y/n]: y
Instalando: com.github.huluti.Curtail/x86_64/stable desde flathub
[#####] 36 metadata, 34 content objects fetched; 139 KiB transfer
Ahora a las 4537b2ea49ef.
Instalando: com.github.huluti.Curtail.Locale/x86_64/stable desde flathub
[#####] 7 metadata, 2 content objects fetched; 6 KiB transferred
Ahora a las 196cd51c2251.
sergio@sololinux:~$
```

Puedes ejecutar el compresor de imágenes desde el menú de aplicaciones, de tu entorno de escritorio preferido. También con este comando...

```
flatpak run com.github.huluti.Curtail
```

Observa que fácil es el uso de la herramienta.



Puedes desinstalar el compresor de imágenes Curtail, con el siguiente comando.

```
flatpak remove com.github.huluti.Curtail/x86_64/stable
```

Se ofrece la opción de instalarlo de forma manual, pero no te la recomiendo. Necesitarás instalar aparte **git**, **pip** y **meson** en su versión 0.50 o superior.

```
git clone https://github.com/Huluti/Curtail.git
cd Curtail
meson _build
cd _build
ninja
sudo ninja install
```

Puedes desinstalar con el siguiente comando.

```
sudo ninja uninstall
```

## 2.4 GHz vs 5 GHz – En que se diferencian

La gran mayoría de los **enrutadores Wi-Fi**, que se comercializan en la actualidad son de doble banda. Esto quiere decir que tienen la capacidad de transmitir una **señal Wi-Fi** en dos frecuencias, **2,4 GHz y 5 GHz**.

Pero... ¿Sabes que son las bandas **2.4 y 5 GHz**?, ¿forman parte de la misma **red Wi-Fi**?, ¿necesitas dos señales wifi diferentes?, ¿cuál de las dos es mejor?. En el artículo de hoy respondemos a tus preguntas. **2.4 GHz vs 5 GHz**.



### 2.4 GHz vs 5 GHz – En que se diferencian

#### Qué son las Wifis de 2.4 GHz y de 5 GHz

Primero debes tener claro que las bandas 2.4 GHz y 5 GHz, pertenecen a la misma red wifi. En los enrutadores de doble banda el flujo entrante de internet es el mismo, pero tienen la capacidad de emitir en dos frecuencias diferentes, 2.4 GHz y 5 GHz.

Como son dos bandas totalmente separadas, no interfieren entre sí, por eso los enrutadores modernos transmiten en las dos frecuencias a la vez sin que afecte a la potencia de señal wifi. Muchos de estos dispositivos, vienen preconfigurados para que ambos canales transmitan en las dos bandas, sin tener que configurar nada.

#### 2.4 GHz vs 5 GHz

Las principales diferencias entre las bandas 2.4 y 5 GHz, son dos: velocidad y alcance. Una **conexión Wi-Fi de 5 GHz** es mucho más rápida que la de 2,4 GHz, pero su rango de acceso es más corto. Entonces, es evidente que si la distancia con el enrutador es grande, debes usar la de 2.4 GHz si quieres tener una conexión estable.

Lo explicado en el párrafo anterior, es el motivo principal por el cual los enrutadores de doble banda ofrecen a los usuarios la posibilidad de elegir qué canal se adapta mejor a sus necesidades.

#### Por qué usar la banda de 5 GHz

Como mencionamos anteriormente, un Wi-Fi corriendo en 5 GHz es más rápido que 2.4 GHz. Esto se debe a que la frecuencia de 5 GHz admite velocidades de datos más altas, por tanto, si tiene capacidad de transmitir grandes cantidades de datos a un dispositivo está claro que la conexión es más rápida.

También debemos tener en cuenta, que la banda de 5 GHz es menos utilizada que la de 2,4 GHz; por tanto la conexión es más estable y menos propensa a las interferencias de otros dispositivos. En los tiempos que corren del **internet de las cosas**, hasta los microondas se conectan a internet, y este tipo de dispositivos usan 2,4 GHz.

Los dispositivos de **2.4 GHz** compiten entre sí por acaparar el mayor flujo posible, esto provoca un rendimiento deficiente. Cuantos más dispositivos se conectan, más despacio funciona.

Dicho lo anterior, los **5 GHz** tienen un rango más limitado si lo comparamos con **2,4 GHz**. Si eres consumidor de grandes cantidades de datos continuos, como por ejemplo la transmisión de **video HD** o eres fan de los **videojuegos online**, deberías cambiar a **5 GHz**.



## Por qué usar la banda de 2.4 GHz

Bueno, a estas alturas del artículo todos tenemos claro que la 2,4 GHz es un poco más lenta, sin embargo, ofrece un rango superior (tema muy importante). Cuando hablamos de transmisiones inalámbricas, no pienses en un terreno abierto jaja, considera el hogar, tu empresa, tu oficina, etc.

Todo a nuestro alrededor está lleno de impedancia (resistencia por obstáculos), paredes, techos, puertas, etc, ralentizan nuestra red inalámbrica. Gracias a que la 2.4 GHz tiene una longitud de onda más larga, es mucho mejor para atravesar obstáculos sin perder excesiva potencia.

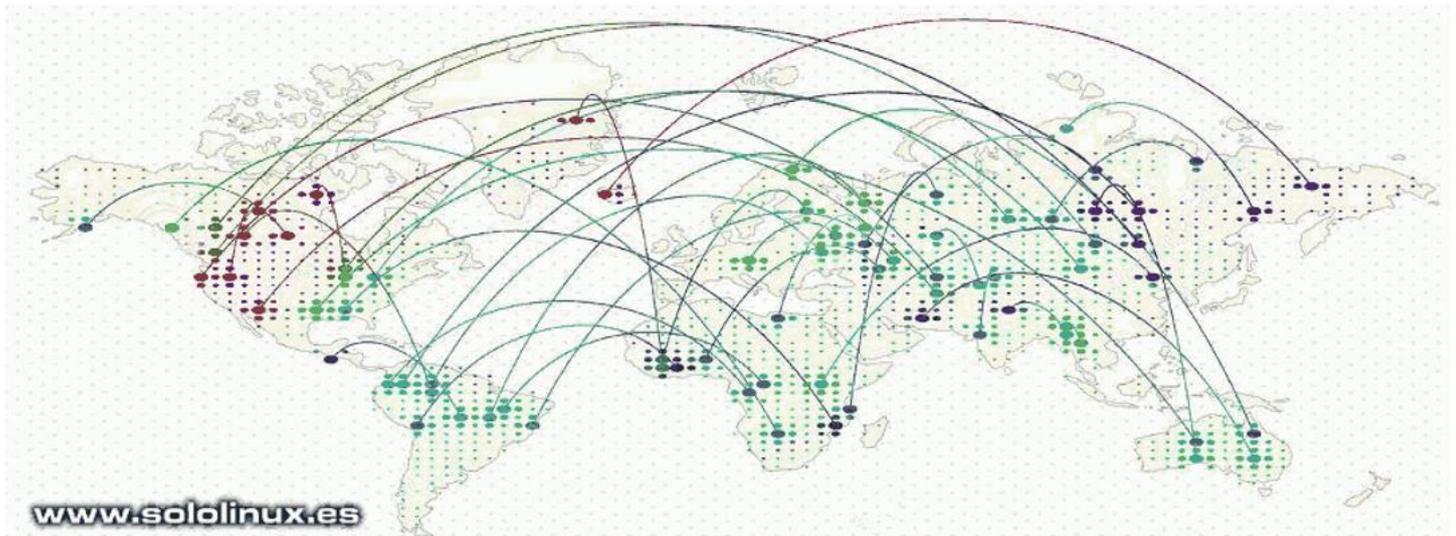
Como ya comente anteriormente, a no ser que necesites transmitir video u otras tareas con uso intensivo de datos, lo normal es utilizar 2.4 GHz. Para visitar **sitios web**, revisar el **correo electrónico** y otras tareas comunes, la banda de 2.4 GHz es perfecta, ni siquiera notarás la diferencia.

## Conclusión

Tener dos bandas nos permite distribuir los dispositivos a través de los canales, esto ayudará a reducir la masificación de dispositivos y a mejorar el rendimiento de la red en general.

Debes recordar que cuanto más lejos esté un dispositivo del enrutador, más longitud de onda necesitas para tener una buena conexión, debes usar la banda 2,4 GHz, sí o sí. En casas, pisos, oficinas, empresas, etc, es recomendable que te decantes por la de 2.4 GHz.

Los requisitos de cada usuario son diferentes, te recomiendo que hagas pruebas con sus dispositivos hasta que des con la banda que mejor funcione para ti. 2.4 GHz vs 5 GHz.



## Instalar GNS3 en Ubuntu y derivados

**GNS3** (Graphical Network Simulator-3) es un emulador al estilo **Cisco Packet Tracer**, pero con algunas diferencias sustanciales que lo hacen más poderoso. Este software de red, tiene la capacidad de ayudar a los profesionales en redes a diseñar, construir, conectar y mejorar redes.

También es de mucha utilidad a la hora de configurar, probar y solucionar problemas emulando dispositivos de hardware reales. Miles de profesionales lo usan, porque tú no. Tiene la capacidad de operar con dispositivos de hardware diversos incluyendo varios fabricantes, incluidos los virtuales de Cisco, ASA de Cisco, **vRouters Brocade**, de **Cumulus Linux**, instancias **Docker**, VSR de HPE y más.



Puedes usar **GNS3** para probar la interoperabilidad y, las experiencias en un mundo real antes de adquirir cualquier dispositivo. Siguiendo los pasos de este artículo, podrás **instalar GNS3** en Ubuntu 20.04 o Ubuntu 18.04 incluyendo todos sus derivados.

## Instalar GNS3 en Ubuntu y derivados

Disponemos de diversas formas de instalar GNS3 en Ubuntu 20.04, pero la más rápida y confiable es instalar la aplicación desde su **repositorio oficial de PPA**. Puedes ejecutar los siguientes comandos para agregar el PPA.

```
sudo add-apt-repository ppa:gns3/ppa
```

Actualizamos e instalamos GNS3.

```
sudo apt update
sudo apt install gns3-gui gns3-server
```

Dependiendo de tu versión de Ubuntu, puede demorar un poco, sé paciente.

```
root@SoloLinux:~# sudo apt install gns3-gui gns3-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  adwaita-icon-theme at-spi2-core cpu-checker cpulimit dmeventd dmidecode
  dns-root-data dnsmasq-base dynamips gstreamer1.0-plugins-base
  gstreamer1.0-plugins-good gstreamer1.0-x gtk-update-icon-cache
  hicolor-icon-theme humanity-icon-theme i965-va-driver ibverbs-providers
  intel-media-va-driver ipxe-qemu ipxe-qemu-256k-compat-efi-roms libaa1
  libaiol libasyncons0 libatk-bridge2.0-0 libatk1.0-0 libatk1.0-data
  libatspi2.0-0 libavahi-client3 libavahi-common-data libavahi-common3
  libavahi-glib1 libavahi-gobject0 libavahi-ui-gtk3-0 libavc1394-0
  libboost-iostreams1.71.0 libboost-thread1.71.0 libbrlapi0.7 libc-ares2
  libcaca0 libcacard0 libcairo-gobject2 libcairo2 libcdparanoia0 libcolord2
  libcups2 libcurl3-gnutls libdatrie1 libdevmapper-event1.02.1
  libdouble-conversion3 libdrm-amdgpu1 libdrm-common libdrm-intel1
  libdrm-nouveau2 libdrm-radeon1 libdrm2 libdv4 libegl-mesa0 libegl1 libepoxy0
  libevdev2 libfdt1 libflac8 libfontenc1 libfreerdp2-2 libfribidi0 libfuse2
  libgbm1 libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-common
  libgl1 libgl1-mesa-dri libglapi-mesa libglvnd0 libglx-mesa0 libglx0
  libgraphite2-3 libgstreamer-plugins-base1.0-0 libgstreamer-plugins-good1.0-0
  libgtk-3-0 libgtk-3-bin libgtk-3-common libgtk-vnc-2.0-0 libgudev-1.0-0
  libgvnc-1.0-0 libharfbuzz0b libibverbs1 libice6 libiec61883-0 libigdgmm11
  libinput-bin libinput10 libiscsi7 libjack-jackd2-0 libjbig0 libjpeg-turbo8
  libjpeg8 libjson-glib-1.0-0 libjson-glib-1.0-common liblcms2-2 libllvm11
```

www.sololinux.es

Te aparecen un par de preguntas.

- Pueden usarlo los usuarios no root
- Pueden capturar los usuarios no root

En ambas preguntas responde que si (Y).

```
Ubridge can be installed in a way that allows members of the 'ubridge' system group to capture packets. This is recommended over the alternative of running GNS3 directly as root, because less of the code will run with elevated privileges.
All users members of 'sudo' or 'admin' group will be added to this group by default.
Without that most GNS3 features will not work.
Should non supervisors be able to run GNS3?
www.sololinux.es
```

**GNS3 necesita soporte IOU**, por tanto lo instalamos.

```
sudo dpkg --add-architecture i386
sudo apt update
sudo apt install gns3-iou
```

```
root@SoloLinux:~# sudo apt install gns3-iou
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
gcc-10-base:i386 lib32z1 libcc1:i386 libcc1-0:i386 libcrypt1:i386
libgcc-s1:i386 libidn2-0:i386 libssl1.1:i386 libunistring2:i386
Suggested packages:
glibc-doc:i386 locales:i386
The following NEW packages will be installed:
gcc-10-base:i386 gns3-iou lib32z1 libcc1:i386 libcc1-0:i386
libcrypt1:i386 libgcc-s1:i386 libidn2-0:i386 libssl1.1:i386
libunistring2:i386
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 7260 kB of archives.
After this operation, 34.2 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

De forma predeterminada, GNS3 utiliza **contenedores Docker** para crear jaulas de redes. Lo instalamos con estos comandos.

```
sudo apt install apt-transport-https ca-certificates curl software-properties-common
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

```
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

```
sudo apt update
sudo apt install docker-ce
```

```
root@SoloLinux:~# sudo apt install docker-ce
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
apparmor containerd.io docker-ce-cli docker-ce-rootless-extras git
git-man liberror-perl pigz slirp4netns
Suggested packages:
apparmor-profiles-extra apparmor-utils aufs-tools cgroupfs-mount |
cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-el
git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
apparmor containerd.io docker-ce docker-ce-cli docker-ce-rootless-
extras git git-man liberror-perl pigz slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 110 MB of archives.
After this operation, 492 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

No olvides agregar nuestro usuario al grupo «Docker».

```
sudo usermod -a -G docker TU-USUARIO
```

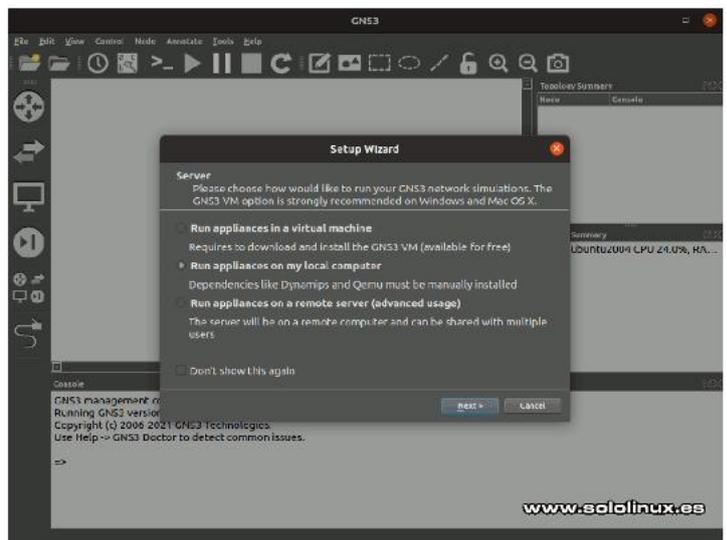
También a los grupos «ubridge», «libvirt», «kvm» y «wireshark».

```
sudo usermod -aG ubridge TU-USUARIO
sudo usermod -aG libvirt TU-USUARIO
sudo usermod -aG kvm TU-USUARIO
sudo usermod -aG wireshark TU-USUARIO
```

Si todo salió bien la instalación ha concluido, pero necesario reiniciar el sistema.

```
sudo reboot
```

Ahora mismo, la **herramienta GNS3** debería estar instalada, pre-configurada y lista para usar. Encontrarás el lanzador, en el menú de aplicaciones de tu entorno de escritorio preferido.



## Publicidad:

Quieres poner publicidad en la revista, ahora puedes hacerlo de forma muy simple, llegando a todo el mundo con esta revista digital gratuita de software libre y GNU/Linux en ESPAÑOL

CON **SOLOLINUX** MULTIPLICARAS TUS CLIENTES

Para mayor información escribe un email a: [adrian@sololinux.es](mailto:adrian@sololinux.es)



## Parrot Security 4.11 – Listo para su descarga

Diseñada específicamente para realizar **pruebas de penetración y evaluación de vulnerabilidades**, Parrot 4.11 ya está listo para su descarga. Es evidente que llega para competir (como no podía ser menos) con Kali Linux 2021.1.

Parrot Security 4.11 es una distribución forense, con grandes capacidades (incluso más que Kali), incluyendo la detección de vulnerabilidades en sistemas operativos y redes. Basado en **Debian Testing**, esta bestia de la seguridad incluye todas las actualizaciones incluidas en el repositorio oficial testing, también los del futuro Debian 11 ("Bulseye").

Este sistema operativo no solo trata temas **hacking**, su principal particularidad es la protección al usuario que logra enviando todo el tráfico de datos a través de la **red Tor**; Por tanto se permite el lujo de garantizar (en la medida de lo posible) el máximo anonimato y seguridad de los datos. Ofrece el control desde una consola de red anónima descentralizada, o usando una interfaz gráfica de usuario.

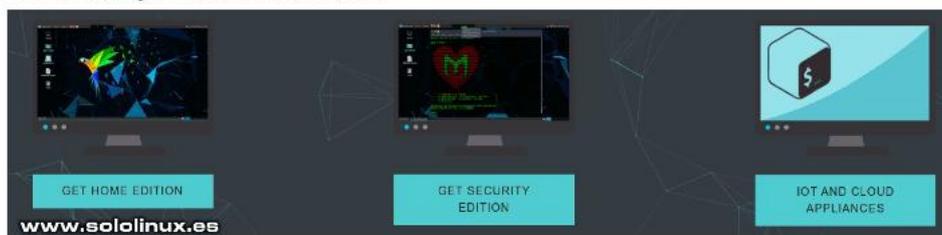
### Parrot Security 4.11 – Listo para su descarga

A la espera del kernel 5.11, de momento viene con el **5.10 LTS**. Todas sus herramientas han sido actualizadas, incluyendo las propias para un uso cotidiano como pc de escritorio, por ejemplo LibreOffice 7.1.1 o Firefox 87.

En esta nueva versión de Parrot Security desaparece Python 2.X y su compatibilidad, viene con Python 3.9. Otras actualizaciones a destacar son el marco de Metasploit, el GCC se actualiza a la versión 10.2.1 y, se agrega la compatibilidad con Pisces y Zsh.

#### Descargar Parrot 4.11

Parrot Security 4.11 se distribuye en tres versiones.



#### Home Edition

**Parrot Home Edition** es un sistema de uso general, diseñado para uso diario, incluyendo privacidad y desarrollo de software. Se permite instalar de forma manual **Parrot Tools**, para que montes tu propio entorno de pentesting totalmente personalizado. Puedes descargar la versión de escritorio **XFCE, Cinnamon o KDE**.

#### Security Edition

**Parrot Security Edition** está diseñado para lanzar pruebas de penetración y, otras operaciones diversas en redes. Viene con una tonelada de **herramientas pentest** listas para usar, según tus necesidades. Puedes descargar la versión **Mate o KDE**.

#### IOS y Cloud Appliances

**Parrot IoT y Cloud Appliances** son ediciones específicas, para dispositivos, entornos en la nube, máquinas virtuales y otras integraciones especiales. Elige la versión que más se ajuste a tus necesidades, **OVA Home, OVA Security y Docker Images**.

Enlaces de descarga:

- [Parrot Home Edition](#)
- [Parrot Security Edition](#)
- [Parrot IoT / Cloud Appliances](#)

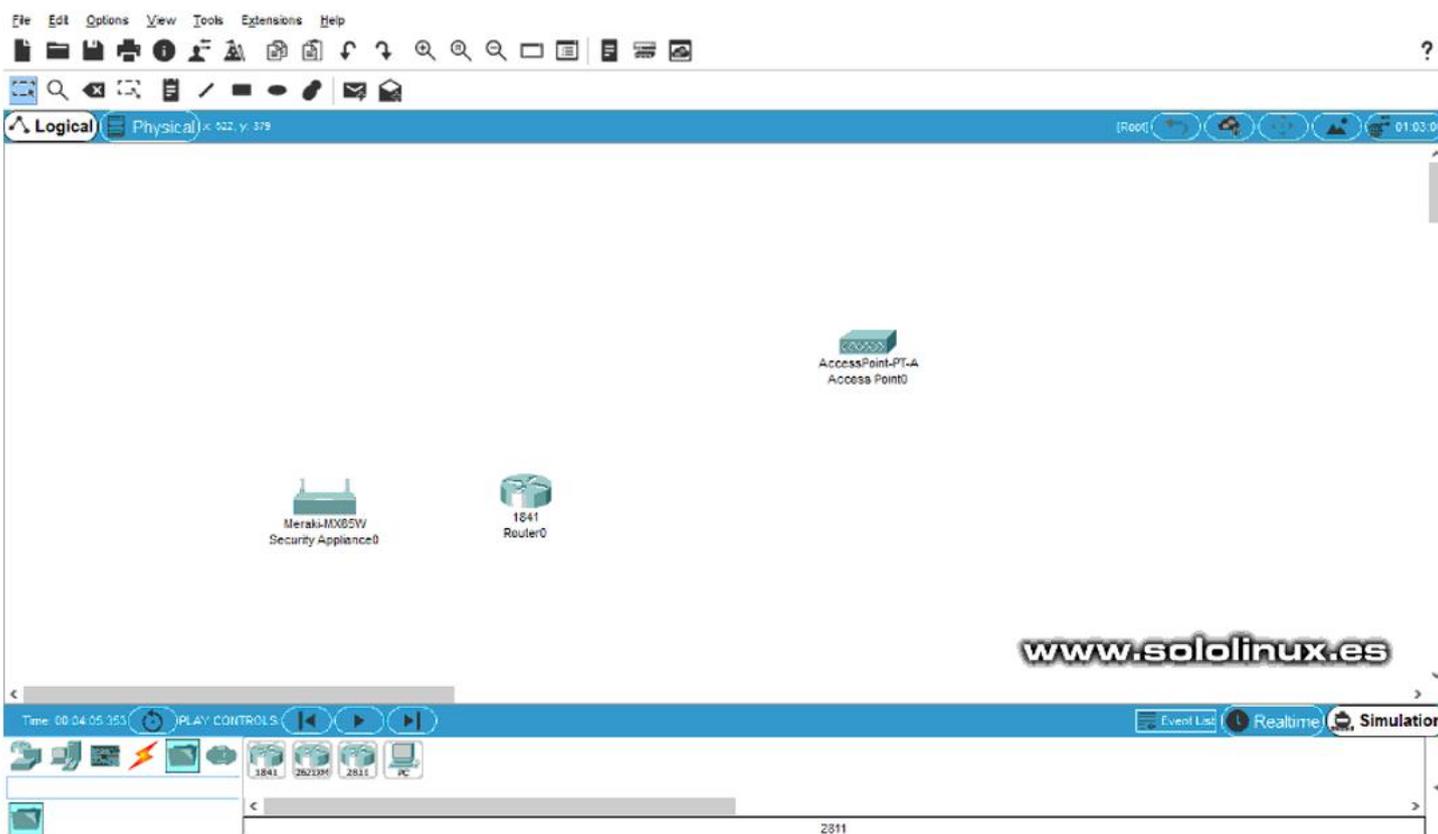


## Descargar Cisco Packet Tracer 8.0.0 y versiones anteriores

Cisco Packet Tracer es una poderosa herramienta desarrollada por **Cisco Systems**, con la capacidad de simular el comportamiento de una red incluyendo todos sus dispositivos. Al contrario de otras herramientas emuladoras como **GNS3**, Packet Tracer simula, no emula.

Con **Cisco Packet Tracer**, puedes diseñar topologías de red sencillas y complejas, para luego aplicarlas a un mundo real. También es útil para el diseño de máquinas virtuales, enrutadores, conmutadores y otros dispositivos. Todo es posible con esta aplicación, que nos ayuda a verificar un funcionamiento correcto a la hora de la implantación.

Las principales características de la versión Packet Tracer 8.0.0 son, el controlador de red SDN, así como las capacidades de programación API. La nueva apariencia gráfica, incluyendo la pantalla de inicio son realmente sorprendentes. En este artículo ofrecemos los enlaces de descarga gratuitos de descarga, no solo para **linux**, también los de **Windows**. No abuses de ellos por favor, entorpeces al resto de usuarios.



## Descargar Cisco Packet Tracer 8.0.0 y otras versiones

Descargar Packet Tracer 8.0.0

- Cisco Packet Tracer 8.0.0 para Windows (10, 8.1, 7.0) edición de 64 bits
- Cisco Packet Tracer 8.0.0 para Windows (10, 8.1, 7.0) edición de 32 bits
- Cisco Packet Tracer 8.0.0 para linux de 64 bits

Descargar Packet Tracer 7.3.1

- Cisco Packet Tracer 7.3.1 para Windows (10, 8.1, 7.0) edición de 64 bits
- Cisco Packet Tracer 7.3.1 para Windows (10, 8.1, 7.0) edición de 32 bits
- Cisco Packet Tracer 7.3.1 para linux de 64 bits

#### Descargar Packet Tracer 7.3.0

- Cisco Packet Tracer 7.3.0 para Windows (10, 8.1, 7.0) edición de 64 bits
- Cisco Packet Tracer 7.3.0 para Windows (10, 8.1, 7.0) edición de 32 bits
- Cisco Packet Tracer 7.3.0 para linux de 64 bits

#### Descargar Packet Tracer 7.2.2

- Cisco Packet Tracer 7.2.2 para Windows (10, 8.1, 7.0) edición de 64 bits
- Cisco Packet Tracer 7.2.2 para Windows (10, 8.1, 7.0) edición de 32 bits
- Cisco Packet Tracer 7.2.2 para linux de 64 bits

#### Descargar Packet Tracer 7.2.1

- Cisco Packet Tracer 7.2.1 para Windows (10, 8.1, 7.0) edición de 64 bits
- Cisco Packet Tracer 7.2.1 para Windows (10, 8.1, 7.0) edición de 32 bits
- Cisco Packet Tracer 7.2.1 para linux de 64 bits

#### Descargar Packet Tracer 7.2.0

- Cisco Packet Tracer 7.2.0 para Windows (10, 8.1, 7.0) edición de 64 bits
- Cisco Packet Tracer 7.2.0 para Windows (10, 8.1, 7.0) edición de 32 bits
- Cisco Packet Tracer 7.2.0 para linux de 64 bits

#### Descargar Packet Tracer 7.1.1

- Cisco Packet Tracer 7.1.1 para Windows (10, 8.1, 7.0) edición de 64 bits
- Cisco Packet Tracer 7.1.1 para Windows (10, 8.1, 7.0) edición de 32 bits
- Cisco Packet Tracer 7.1.1 para linux de 64 bits

#### Descargar Packet Tracer 7.1.0

- Cisco Packet Tracer 7.1.0 para Windows (10, 8.1, 7.0) edición de 64 bits
- Cisco Packet Tracer 7.1.0 para Windows (10, 8.1, 7.0) edición de 32 bits
- Cisco Packet Tracer 7.1.0 para linux de 64 bits

#### Descargar Packet Tracer 7.0.0

- Cisco Packet Tracer 7.0.0 para Windows edición de 64 bits
- Cisco Packet Tracer 7.0.0 para Windows edición de 32 bits
- Cisco Packet Tracer 7.0.0 para linux de 64 bits
- Cisco Packet Tracer 7.0.0 para linux de 32 bits

#### Descargar Packet Tracer 6.3

- Cisco Packet Tracer 6.3 para Windows
- Cisco Packet Tracer 6.3 para Linux

#### Descargar Packet Tracer 5.0

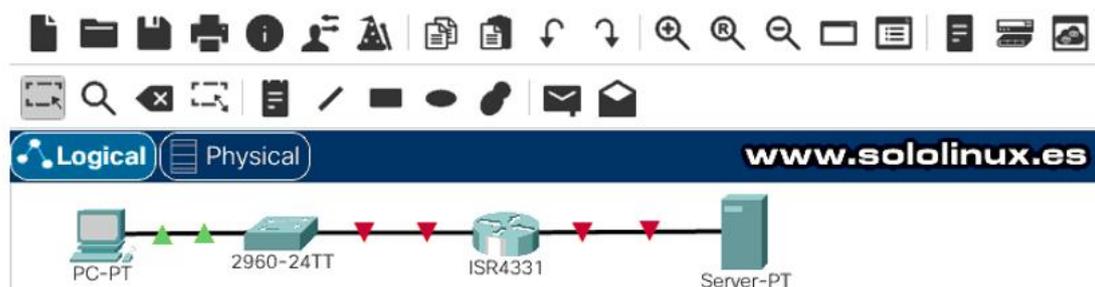
- Cisco Packet Tracer 5.0 para Windows

#### Descargar Packet Tracer 4.0

- Cisco Packet Tracer 4.0 para Windows

**NOTA:** Las últimas versiones publicadas para linux son **\*.deb**, exclusivamente compatibles con Ubuntu, Debian y derivados.

**Las versiones obsoletas pueden ser extremadamente útiles, si utilizas una maquina obsoleta para realizar tus pruebas. Descargar Cisco Packet Tracer 8.0 y versiones anteriores.**



A brick wall with a large red and black geometric shape on the right side. The red shape is a large, irregular polygon that points downwards. The black shape is a large, irregular polygon that points upwards. The wall is made of dark grey bricks with light grey mortar. The red and black shapes are painted on the wall.

[www.sololinux.es](http://www.sololinux.es)