

MAGAZINE SOLO LINUX

Nº
23

Tu revista, la revista de tod@s

DICIEMBRE 2020



Características del **Kernel 5.10 LTS**
(disponible la RC-6)

Crear y administrar máquinas virtuales
con **Virt-Manager**

Cómo configurar **Firewalld** en **CentOS 8**

Instalar y configurar **Firewalld** en **Debian 10**

Qué son los **inodos**
en Linux

Instalar el repositorio **RPM Fusion** en **Fedora**,
CentOS y **RHEL**

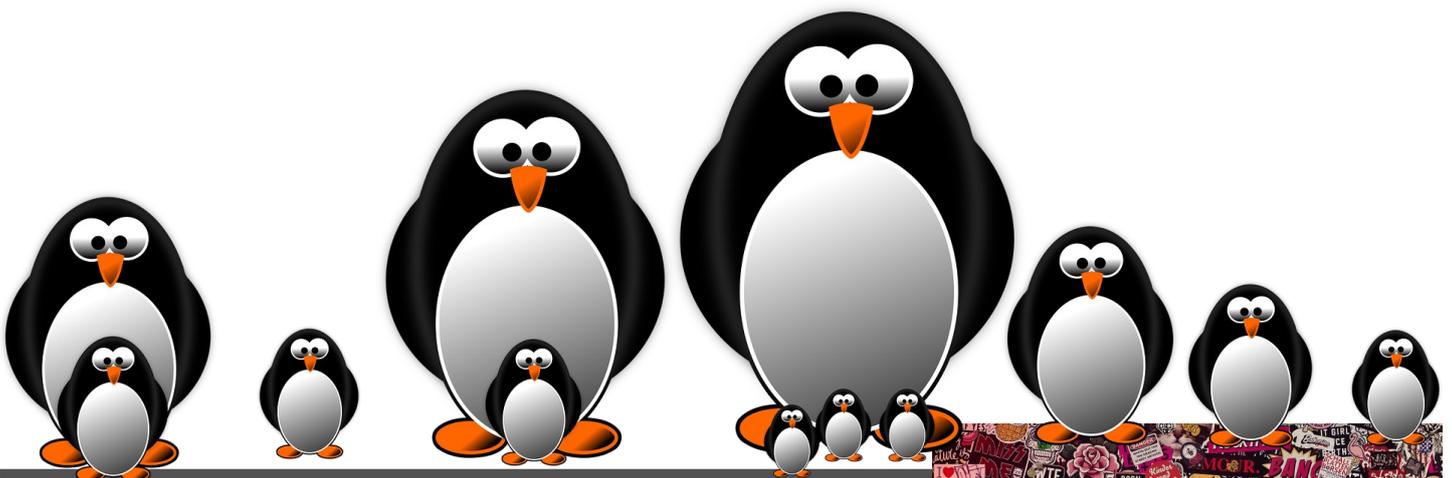
MANUALES, SCRIPTS, SOFTWARE, HARDWARE, DISTROS LINUX,
SEGURIDAD, REDES Y MUCHO MAS EN LA WEB...

SoloLinux

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



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



By: MichaelGaida. Extraída de Pixabay.

BIENVENIDO A LA REVISTA SOLOLINUX

Buenos días, tardes o noches, dependiendo del lugar del mundo donde se encuentren ahora mismo.

Os presentamos el número 23 de la **Revista SoloLinux**.

Quiero agradecer a todos los que hacéis que este proyecto siga adelante. Desde el equipo de SOLOLINUX esperamos que os guste este nuevo número.

Feliz Navidad linuxera y mucho OpenSource para el nuevo año.

Un año difícil para muchos de nosotros, pero que hemos sacado adelante con la ayuda de todos. Gracias por seguir leyendo nuestras publicaciones.

Sergio, Adrián y Henry os deseamos...

Feliz Navidad linuxera con mucho OpenSource, para estos días tan emotivos.

Gracias a todos por acompañarnos en esta aventura.

Compartan esta revista en sus redes sociales o web.
Revista digital **SOLOLINUX MAGAZINE**.

**Tu revista, la revista
de todos.**

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**



Síguenos en
las Redes:



La revista
SOLOLINUX esta
realizada con
Libre Office
Impress 7.0.0.3

Editorial

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

Redacción

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

Diseño Portada

- **Karina Fernández**
@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

Contacto

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

- adrian@sololinux.es



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

SEGURIDAD

- 08. Como configurar Firewalld en CentOS 8
- 20. Instalar y configurar Firewalld en Debian 10
- 23. Escanear un servidor linux con Nikto
- 25. Permisos de archivo estándar simples en linux



MANUALES

- 10. Instalar el repositorio RPM Fusion en Fedora, CentOS y RHEL
- 13. Crear y administrar máquinas virtuales con Virt-Manager
- 18. Uso del comando stat en linux
- 26. Anarchy – Instalar Arch Linux para novatos
- 32. Instalar el Kernel 5.10 LTS en Ubuntu o Linux Mint
- 36. Instalar un servidor Debian 10 con Yunohost
- 40. Montar un servidor casero con FreedomBox
- 50. Instalar XFCE 4.16 en Ubuntu 20.04 o Linux Mint
- 51. Uso del comando Nslookup en linux
- 53. Uso del comando Netstat en linux

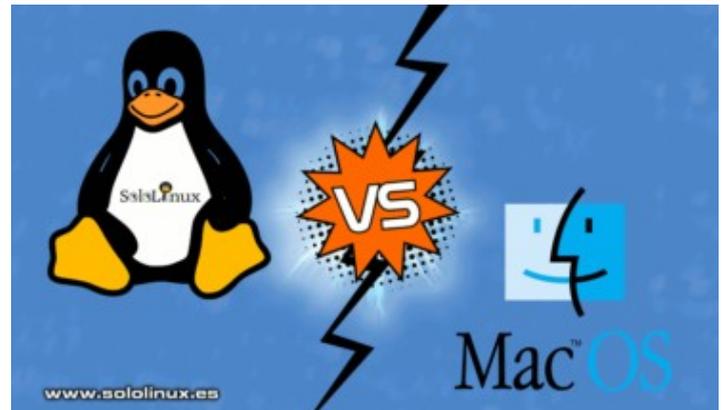


REDES

- 34. Monitorizar la red con Iprtraf
- 38. Uso del comando route en linux

NOTICIAS

- 12. Características del Kernel 5.10 LTS (disponible la RC-6)
- 30. CentOS se muere – Bienvenido Rocky Linux
- 42. Linux vs MacOS – Sus principales diferencias
- 46. Entrevista a un Sysadmin Linux en 17 preguntas



SOFTWARE

- 16. Instalar Tixati Torrent en Ubuntu 20.04 y otros linux



HARDWARE

- 22. Que son los inodos en linux
- 33. Mantener los controladores Mesa actualizados

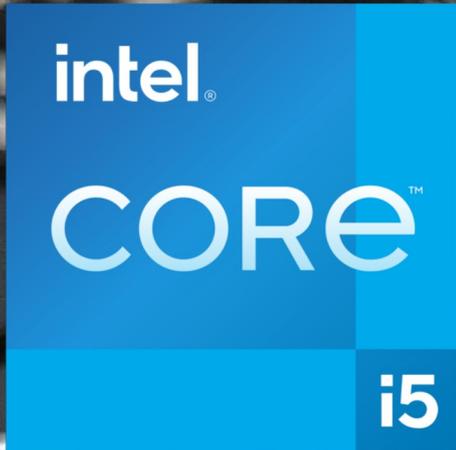
www.sololinux.es

OPINIÓN DEL LECTOR

- 55. La opinión de Raydel Comas Rodríguez
- 55. La opinión de Erwin Andres Espitia

VANT

SOMOS LINUXEROS



edge²

Nuestro ultrabook más ligero y con mayor autonomía, ahora más potente con Intel Core de 11ª generación y gráficos Intel Iris Xe

descúbreanos en 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




THANKS!

¿QUIERES PUBLICITARTE EN LA REVISTA?



MAGAZINE
SoloLinux



Puedes hacerlo de una forma muy simple, llegando a todo el mundo con la única revista digital de Software libre y GNU/Linux en Español

CON SOLOLINUX MULTIPLICARA SUS CLIENTES



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



www.sololinux.es



**INSTITUTO
LINUX**



INSCRIPCIÓN ABIERTA

TODO POR

€ 397



CARRERA LINUX 2021

CURSO LINUX SYSTEM ADMINISTRATOR

+

CERTIFICACIÓN UTN-FRD

+

WORKSHOPS LPIC 1 (101-102)

+

**VIDEOS EN ESPAÑOL PARA LFS 201
LINUX SYSTEM ADMINISTRATOR**

Tutorías de Fabián Ampalio



+54 9 11 6969 9993



@fabianampalio

Como Configurar Firewalld en CentOS 8



El uso de **Firewall**, es una de las mejores formas de proteger tu sistema de tráfico no deseado procedente de redes externas. Una herramienta perfecta para tal efecto, es **Firewalld**, que además es **Open Source**.

Existe una falsa creencia entre los profanos de la **administración de servidores**; No sería la primera vez que alguien me dice que no necesita **iptables** (o **nftables**), tiene **UFW** o **Firewalld**. Es evidente que existe un gran desconocimiento al respecto, UFW, Firewalld y otros muchos, son simples controladores frontend para **iptables** o **nftables**, que ofrecen una interfaz cómoda en línea de comandos para implementar reglas Firewall (existen versiones GUI).

Firewalld usa zonas y servicios, en vez de las cadenas y reglas típicas de **iptables**, además tiene la capacidad de administrar los conjuntos de reglas dinámicamente. La herramienta en línea de comandos para administrar el tiempo de ejecución y su configuración permanente, se conoce como **firewall-cmd**. **Configurar Firewalld**

Esta herramienta opera de forma nativa en las siguientes **distribuciones Linux**:

- * **RHEL 7 y posteriores.**
- * **CentOS 7 y posteriores.**
- * **Fedora 18 y más nuevos.**
- * **SUSE 15 y posteriores.**
- * **OpenSUSE 15 y más recientes.**



www.sololinux.es

Como configurar Firewalld en CentOS 8 y Fedora

Firewalld ofrece grandes mejoras respecto al Firewall de **Netfilter**, antes de comenzar a configurar **Firewalld** vemos sus principales características.

- API D-Bus completa.
- Soporte completo para IPv4, IPv6, bridge e ipset.
- Compatible con NAT IPv4 e IPv6.
- Especificar zonas de cortafuegos.
- Lista predefinida de zonas, servicios y tipos de icmp.

- Servicio simple, puertos, protocolos, puertos de origen, enmascaramiento, reenvío de puertos, filtro icmp, reglas enriquecida, completa interfaz y el control de direcciones de origen en las zonas.
- Definición del servicio simple con puertos, protocolos, puertos de origen, módulos de netfilter y manejo de direcciones de destino.
- Lenguaje enriquecido que nos permiten crear reglas más flexibles y complejas.
- Reglas de firewall por tiempo en las zonas.
- Registro simple de paquetes denegados.
- Excelente interfaz.
- Lista blanca de aplicaciones que pueden modificar el firewall.
- Carga automatizada de los módulos del kernel Linux.
- Integración con Puppet.
- Configuración online y offline.
- Herramienta de configuración gráfica mediante gtk3.
- Applet para Qt4.

Firewalld viene instalado por defecto en las distribuciones mencionadas anteriormente, pero tal vez no lo tengas si instalaste un minimal server. Por si acaso... instalamos.

CentOS 7, RHEL 7 y derivados:

```
sudo yum install firewalld -y
```

Fedora, CentOS 8, RHEL 8 y derivados:

```
sudo dnf install firewalld -y
```

Open Suse, Suse y derivados:

```
sudo zypper install firewalld -y
```

Una vez instalada, la iniciamos y habilitamos (válido para todas las Distros mencionadas).

```
systemctl start firewalld
systemctl enable firewalld
```

Verificamos su estado.

```
firewall-cmd --state
```

Imagen de ejemplo de salida correcta.

```
[root@sololinux-demo ~]# systemctl start firewalld
[root@sololinux-demo ~]# systemctl enable firewalld
[root@sololinux-demo ~]# firewall-cmd --state
running
[root@sololinux-demo ~]#
```

www.sololinux.es

Bien... lo primero que haremos es identificar las zonas disponibles de manera predeterminada.

```
firewall-cmd --get-zones
```

Aparecen las zonas predefinidas...

```
[root@sololinux-demo ~]# firewall-cmd --get-zones
block dmz drop external home internal public trusted work
[root@sololinux-demo ~]#
```

Ahora identificamos la zona que tenemos activa.

```
firewall-cmd --get-active-zones
```

Como podemos observar, en nuestro caso es la public.

```
[root@sololinux-demo ~]# firewall-cmd --get-active-zones
public
  interfaces: venet0
[root@sololinux-demo ~]#
```

También podemos identificar la zona configurada de forma predeterminada por el sistema.

```
firewall-cmd --get-default-zone
```

Nos indica la public...

```
[root@sololinux-demo ~]# firewall-cmd --get-default-zone
public
[root@sololinux-demo ~]#
```

Puedes modificar la zona aplicada, por cualquiera de las que vimos en el ejemplo anterior con el siguiente comando. En el comando de ejemplo la **zona home**.

```
firewall-cmd --set-default-zone=home --permanent
```

Podemos visualizar la información sobre la zona con este comando (nosotros usamos un servidor, por tanto la public).

```
firewall-cmd --info-zone public
```

Ejemplo...

```
[root@sololinux-demo ~]# firewall-cmd --info-zone public
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: cockpit dhcpv6-client ssh
  ports: 80/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Si quieres, también puedes enumerar los servicios habilitados con el siguiente comando.

```
firewall-cmd --get-services
```

En este paso vamos a permitir un servicio, en este caso el puerto 22 (ssh).

```
firewall-cmd --zone=public --permanent --add-port=22/tcp
```

Reiniciamos la aplicación y verificamos que el **servicio ssh** está habilitado.

```
firewall-cmd --reload
# Con tu zona definida (public)
firewall-cmd --info-zone public
```

En la siguiente imagen vemos el puerto 22 abierto.

```
[root@sololinux-demo ~]# firewall-cmd --info-zone public
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: cockpit dhcpv6-client ssh
  ports: 80/tcp 22/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
www.sololinux.es
```

Para denegar el acceso al servicio o puerto ejecuta el siguiente comando (por ejemplo el puerto 80 httpd).

```
firewall-cmd --zone=public --permanent --remove-port=80/tcp
```

Como puedes observar, permitir o denegar servicios es tarea sencilla. Por ejemplo ftp.

```
# permitir servicio
firewall-cmd --zone=public --permanent --add-service=ftp
firewall-cmd --reload
# denegar servicio
firewall-cmd --zone=public --permanent --remove-service=ftp
firewall-cmd --reload
```

Últimamente, enmascarar la ip es algo común. Es una forma de que el receptor no identifique tu ip privada, pero si la del servidor.

Habilitamos el enmascaramiento.

```
firewall-cmd --zone=public --add-masquerade
firewall-cmd --reload
```

```
[root@sololinux-demo ~]# firewall-cmd --zone=public --add-masquerade
success
[root@sololinux-demo ~]# firewall-cmd --reload
success
www.sololinux.es
```

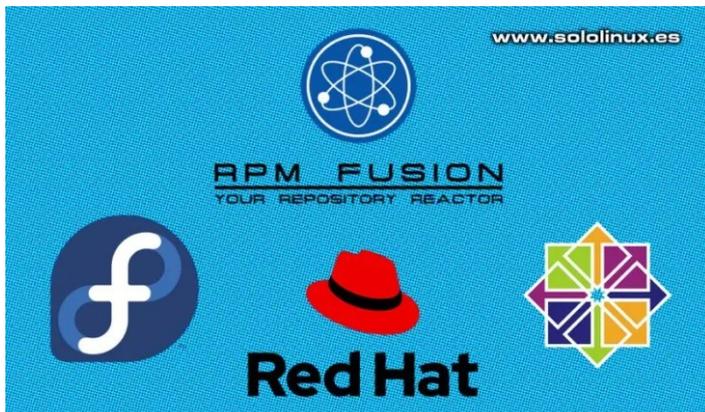
Ahora lo deshabilitamos.

```
firewall-cmd --zone=public --remove-masquerade
firewall-cmd --reload
```

Hemos aprendido a usar FirewallD, bloquear el tráfico no deseado en nuestro sistema, desbloquearlo. Son tareas necesarias si queremos estar protegidos de atacantes.

Canales de Telegram:
Canal SoloLinux
Canal SoloWordpress

Instalar el repositorio RPM Fusion en Fedora, CentOS y RHEL



RPM Fusion es un repositorio de **Fedora Project**, que ofrece software y aplicaciones que **Fedora** o **Red Hat**, no envían en sus repositorios oficiales. Las herramientas se envían como RPM precompilados, para las versiones actuales de Fedora, Red Hat Enterprise, CentOS y todos sus derivados.

El repositorio **RPM Fusion en Fedora** tiene dos versiones, el primero ofrece exclusivamente software gratuito, el segundo contiene paquetes compilados de software de código cerrado o, de código abierto pero no comercial. En el artículo de hoy, vemos como instalar el repositorio **RPM Fusion free** y **RPM Fusion nonfree**, en las siguientes distribuciones Linux

- **RHEL 8**
- **CentOS 8**
- **Fedora**
- **Fedora Silverblue** con rpm-ostree

RPM Fusion free  *RPM Fusion nonfree*

Instalar el repositorio RPM Fusion en Fedora, CentOS y RHEL

Lo primero que debemos hacer es, verificar que no tenemos instalado **RPM Fusion**. Ejecuta el siguiente comando.

```
dnf repolist | grep rpmfusion
```

Si los repositorios no están instalados, no recibirás ninguna respuesta. Si ya los tienes, verás algo similar a la siguiente imagen.

```
[root@sololinux-demo ~]# dnf repolist | grep rpmfusion
rpmfusion-free-updates RPM Fusion for EL 8 - Free - Updates
rpmfusion-nonfree-updates RPM Fusion for EL 8 - Nonfree - Updates
[root@sololinux-demo ~]#
```

RPM Fusion en RHEL 8

```
sudo dnf install --nogpgcheck
https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

```
sudo dnf install --nogpgcheck
https://download1.rpmfusion.org/free/el/rpmfusion-free-release-8.noarch.rpm
```

```
sudo dnf install --nogpgcheck
https://download1.rpmfusion.org/nonfree/el/rpmfusion-nonfree-release-8.noarch.rpm
```

```
sudo subscription-manager repos --enable "codeready-builder-for-rhel-8-$(uname -m)-rpms"
```

RPM Fusion en CentOS 8

```
sudo dnf install --nogpgcheck
https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

```
sudo dnf install --nogpgcheck
https://download1.rpmfusion.org/free/el/rpmfusion-free-release-8.noarch.rpm
```

```
sudo dnf install --nogpgcheck
https://download1.rpmfusion.org/nonfree/el/rpmfusion-nonfree-release-8.noarch.rpm
```

```
sudo dnf config-manager --enable PowerTools
```

Si aparece algún error con el último comando, por ejemplo...

```
[root@sololinux-demo ~]# sudo dnf config-manager --enable PowerTools
No existe el comando: config-manager. Por favor, utilice /bin/dnf --help
It could be a DNF plugin command, try: "dnf install 'dnf-command(config-manager)'"
```

Lo puedes solucionar ejecutando los siguientes comandos.

```
sudo dnf install 'dnf-command(config-manager)'
sudo dnf config-manager --enable PowerTools
```

```
[root@sololinux-demo ~]# dnf install 'dnf-command(config-manager)'
Última comprobación de caducidad de metadatos hecha hace 0:00:33, el jue 03 dic 2020 12:09:22 CET.
Dependencias resueltas.
=====
Paquete           Arq.      Versión      Repo          Tam.
-----
Instalando:
dnf-plugins-core  noarch   4.0.12-4.el8_2  Base05       64 k
Instalando dependencias:
python3-dateutil  noarch   1:2.6.1-6.el8  Base05       251 k
python3-dnf-plugins-core  noarch   4.0.12-4.el8_2  Base05       204 k
Resumen de la transacción
=====
Instalar 3 Paquetes                               www.sololinux.es
Tamaño total de la descarga: 519 k
Tamaño instalado: 1.2 M
¿Está de acuerdo [s/N]? s
Descargando paquetes:
(1/3): dnf-plugins-core-4.0.12-4.el8_2.noarch.r 2.1 MB/s | 64 kB 00:00
(2/3): python3-dnf-plugins-core-4.0.12-4.el8_2. 5.5 MB/s | 204 kB 00:00
(3/3): python3-dateutil-2.6.1-6.el8.noarch.rpm 5.8 MB/s | 251 kB 00:00
-----
Total                                           5.1 MB/s | 519 kB 00:00
Ejecutando verificación de operación
Verificación de operación exitosa.
Ejecutando prueba de operaciones
Prueba de operación exitosa.
Ejecutando operación
Preparando      : python3-dateutil-1:2.6.1-6.el8.noarch          1/1
Instalando     : python3-dnf-plugins-core-4.0.12-4.el8_2.noarch 1/3
Instalando     : dnf-plugins-core-4.0.12-4.el8_2.noarch        2/3
Instalando     : python3-dateutil-2.6.1-6.el8.noarch.rpm       3/3
Ejecutando scriptlet: dnf-plugins-core-4.0.12-4.el8_2.noarch 3/3
Verificando    : dnf-plugins-core-4.0.12-4.el8_2.noarch        1/3
```

RPM Fusion en Fedora

```
sudo dnf install
https://download1.rpmfusion.org/free/fedora/rpmfusion-free-release-$(rpm -E %fedora).noarch.rpm
```

```
sudo dnf install
https://download1.rpmfusion.org/nonfree/fedora/rpmfusion-nonfree-release-$(rpm -E %fedora).noarch.rpm
```

RPM Fusion en Fedora Silverblue con rpm-ostree

```
sudo rpm-ostree install
https://download1.rpmfusion.org/free/fedora/rpmfusion-free-release-$(rpm -E %fedora).noarch.rpm
```

```
sudo rpm-ostree install
https://download1.rpmfusion.org/nonfree/fedora/rpmfusion-nonfree-release-$(rpm -E %fedora).noarch.rpm
```

Otros comandos RPM Fusion

RPM Fusion, también ofrece soporte para GNOME o KDE Discover. Lo instalamos.

```
sudo dnf groupupdate core
```

Si quieres reproducir archivos multimedia adicionales, como por ejemplo gstreamer. También es posible.

```
sudo dnf groupupdate multimedia --setopt="install_weak_deps=False" --exclude=PackageKit-gstreamer-plugin
sudo dnf groupupdate sound-and-video
```

Es posible que necesites la biblioteca **libdvdcss**, para reproducir el formato DVD.

```
sudo dnf install rpmfusion-free-release-tainted
sudo dnf install libdvdcss
```

Habilitamos los paquetes de firmware non-FLOSS (no libres).

```
sudo dnf install rpmfusion-nonfree-release-tainted
sudo dnf install *-firmware
```

Para concluir la instalación debes reiniciar el sistema, pero antes actualizamos.

```
sudo dnf check-update
sudo dnf update
sudo reboot
```

Ejemplo de salida...

```
[root@sololinux-demo ~]# sudo dnf check-update
Última comprobación de caducidad de metadatos hecha hace 0:43:50, el jue 03 dic 2020 12:09:22 CET.
[root@sololinux-demo ~]# sudo dnf update
Última comprobación de caducidad de metadatos hecha hace 0:43:57, el jue 03 dic 2020 12:09:22 CET.
Dependencias resueltas.
Nada por hacer.
¡Listo!
[root@sololinux-demo ~]# sudo reboot
Connection to 185.185.40.142 closed by remote host.
Connection to 185.185.40.142 closed.
```



Características del Kernel 5.10 LTS (disponible la RC-6)



Después del **Kernel 5.4 LTS** (último núcleo a largo plazo), acaba de salir el **Kernel 5.10 LTS RC-6**. La versión **RC-6** es muy importante, pues la siguiente será la final estable. Todo indica, que en breve estará a disposición de cualquier **distribución Linux**

La lista de modificaciones, correcciones y cambios es bastante grande en la nueva versión. Conocemos los principales cambios y como afectaran a nuestro sistema Linux



Características del Kernel 5.10 LTS (disponible la RC-6)

Vemos las características que más nos llaman la atención, alguna realmente innovadora.

Sistema de archivos y almacenamiento

- El sistema de **archivos XFS** soluciona el problema del año 2038, ahora es compatible hasta julio de 2486.
- Un significativo cambio en el **inodo** del grupo de asignación, **acelera los tiempos de montaje**.
- Se agregan modificaciones en el código XFS para desaprobar el **formato del sistema de archivos V4** en 2030.
- **Btrfs**, este sistema de archivos ha sido mejorado. Esto repercute en el rendimiento, sobre todo en la sincronización.
- El sistema de archivos **EXT4**, sigue siendo el más utilizado en las distribuciones Linux. Ahora admite confirmaciones rápidas y, se mejora el rendimiento de sobrescritura de archivos en los modos **DIO / DAX**.

Mejoras en el ámbito del procesador

- Se agrega el soporte para las futuras **CPU Intel Rocket Lake**, que será la arquitectura para CPU de escritorio de próxima generación desarrollada por Intel. También se añade soporte para las CPU **Alder Lake** y **Meteor Lake** de Intel.
- La **CPU AMD Zen 3** obtiene mejoras y correcciones en el rendimiento.
- Soporte para un arranque inicial **RISC-V** por **EFI**.
- Se mitigan las vulnerabilidades de **Spectre** en **ARM**.
- Muchas correcciones y mejoras en procesadores **AMD** e **Intel**. Caché, paginación y nuevas instrucciones de compatibilidad para Intel.

Sistema de gráficos

- La compatibilidad con tarjetas gráficas de próxima generación de Intel, ya está insertada en el nuevo kernel. Se mejora el soporte de **Tiger Lake** para extender la duración de la batería, también el soporte **Gen12** para **Intel Rocket Lake**.
- La arquitectura **AMD RDNA 2** y su compatibilidad con **Radeon RX 600**, fue mejorada de manera notable.
- El Kernel 5.10 LTS, trae muchas actualizaciones en el **DRM** (administrador de renderizado directo).
- Se añade soporte para pantalla en el **Raspberry Pi 4 / BCM2711** con el controlador **VC4 DRM**.

Mejoras en hardware e interfaces

- **USB4**: La interfaz USB de última generación, viene con grandes mejoras en el rendimiento.
- Se incluye soporte para los nuevos paneles táctiles de **portátiles Synaptics**. También para dispositivos **ThinkPad X1, X1E, P1**.
- Adición de **DeviceTree** en dispositivos **ARM Librem 5**.
- La tarjeta de sonido **Creative SoundBlaster Ae-7**, ahora es compatible.
- Se mejora el controlador de **Nintendo Switch**.

Descargar el Kernel 5.10 LTS (RC-6)

Recuerda que no es recomendable usar este kernel en producción (falta poco para la versión estable final), pero sí que puedes probar toda su potencia. Descarga la versión 5.10 LTS RC-6, desde la siguiente url.

- [Descargar la RC-6 del núcleo 5.10 LTS.](#)



Crear y administrar máquinas virtuales Con Virt-Manager



Existen muchos administradores de máquinas virtuales, y sin duda alguna el más conocido entre los usuarios normales es... **VirtualBOX** de **Oracle**. Vale, estamos de acuerdo.

Pero si te digo que existe un administrador mucho más ligero, más rápido, más fácil de usar y, que no te obliga a que tu procesador tenga esa capacidad?, entonces que me dices. La cosa cambia, eh.

Esa herramienta existe y se llama **Virt-Manager**. Es evidente que si agregas **KVM** funcionara mejor, pero no es obligatorio. Crear y **administrar máquinas virtuales con Virt-Manager** es otro mundo, se acabaron los problemas con el mouse, con internet, etc., todo es más fácil y además lo tienes a tu alcance.



Crear y administrar máquinas virtuales con Virt-Manager

La aplicación **virt-manager** necesita la biblioteca **libvirt**, para ofrecer los servicios de administración de máquinas virtuales. Tiene una interfaz de escritorio que te ayuda de manera simple a crear, eliminar y administrar una o varias máquinas virtuales.

Su mejor rendimiento se ofrece en máquinas virtuales KVM, pero también admite otros hipervisores, incluso los propios del sistema Linux

La interfaz de usuario nos brinda una vista resumida de todas las VM, su estado, uso de CPU, gráficos de rendimiento en vivo, estadísticas del uso de los recursos y mucho más. Es importante el detalle que no requiere de la tediosa configuración de la aplicación de Oracle, también destacamos su rapidez.

No perdamos más tiempo con relleno de palabras para alargar el artículo, vamos a lo que vamos. La mayoría de **distribuciones Linux** lo tienen en sus repositorios, así que lo instalamos.

Instalar Virt-Manager En Ubuntu, Debian, Linux Mint y derivados:

```
sudo apt-get install virt-manager
```

En Fedora, RHEL y derivados:

```
sudo dnf install virt-manager
```

En Arch Linux, Manjaro y derivados:

```
pacman -S virt-manager
```

En Gentoo y derivados:

```
emerge virt-manager
```

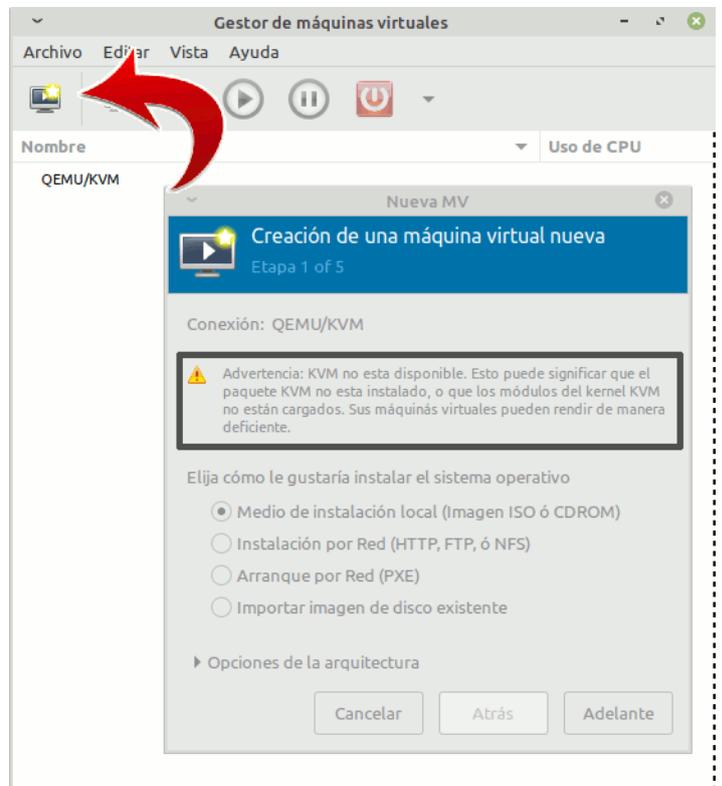
OpenBSD, FreeBSD y derivados:

```
pkg_add virt-manager
```

Una vez instalada la herramienta, la puedes encontrar en tu menú de aplicaciones.

Crear una máquina virtual con Virt-Manager

Abres la herramienta y seleccionas el medio de instalación, normalmente una **iso**.



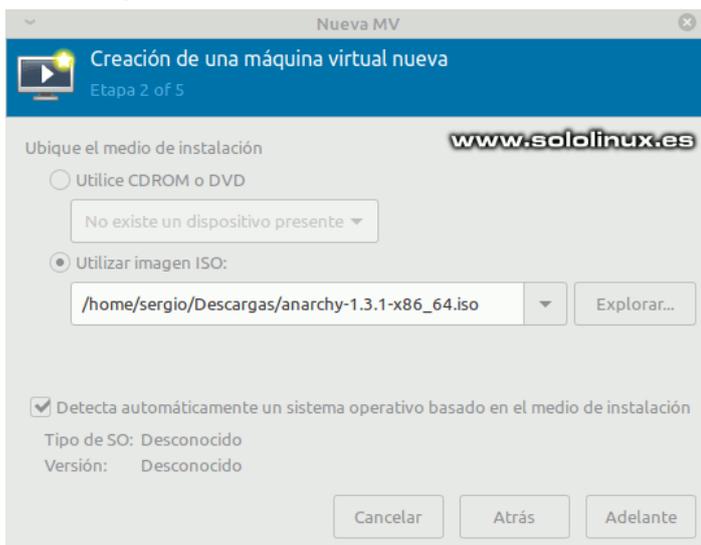
Como puedes ver en la anterior imagen, recibes un aviso donde te indica que **KVM** no está en el sistema. No es necesario, pero si quieres lo puedes instalar. En Ubuntu Debian, Linux Mint y derivados ejecutaríamos el siguiente comando.

```
sudo apt install -y qemu qemu-kvm libvirt-daemon libvirt-clients bridge-utils virt-manager
```

Es importante que después de instalar Virt-Manager, reinicies el sistema.

```
sudo reboot
```

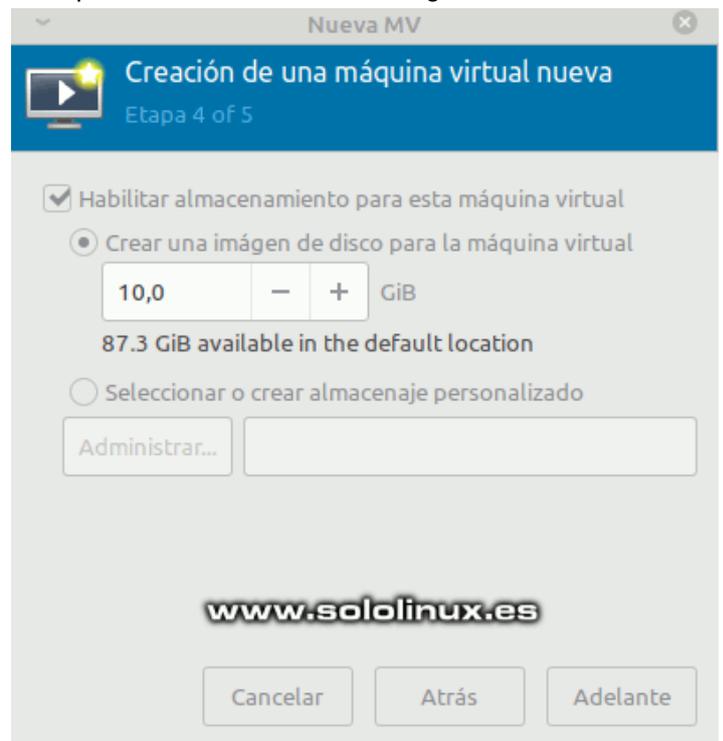
Una vez iniciada de nuevo la aplicación de administrar máquinas virtuales, creamos una nueva máquina como te indica la imagen anterior. Pulsa en «adelante» y selecciona la iso que quieres virtualizar. En nuestro artículo de ejemplo, nos decantamos por **Anarchy Linux**.



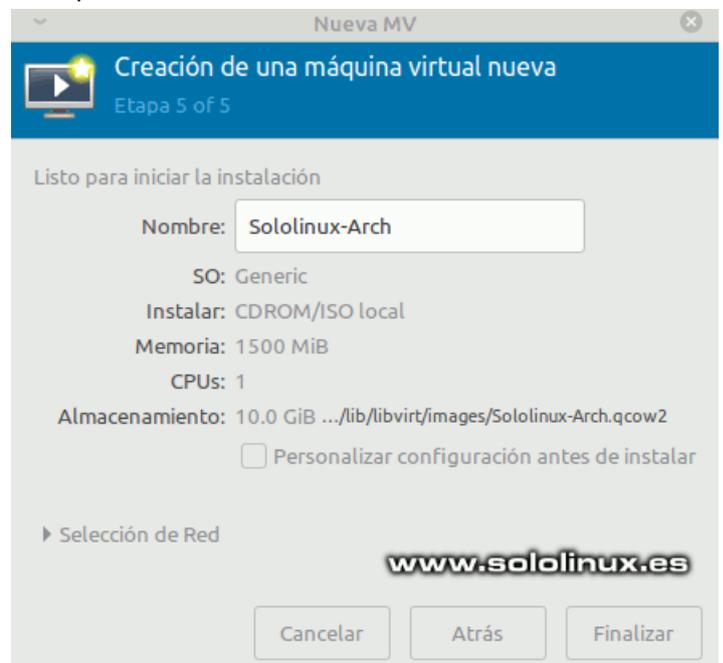
Pulsamos otra vez en «adelante», para configurar la ram y los núcleos de la nueva máquina virtual.



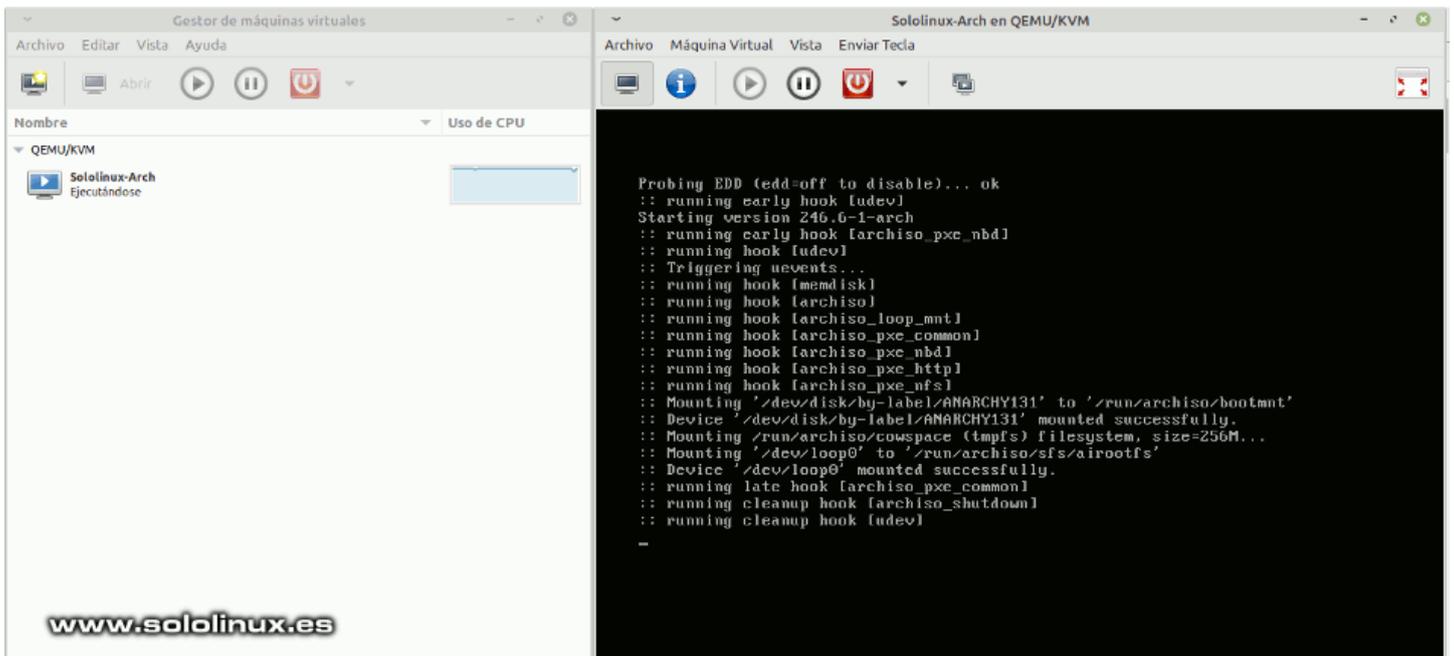
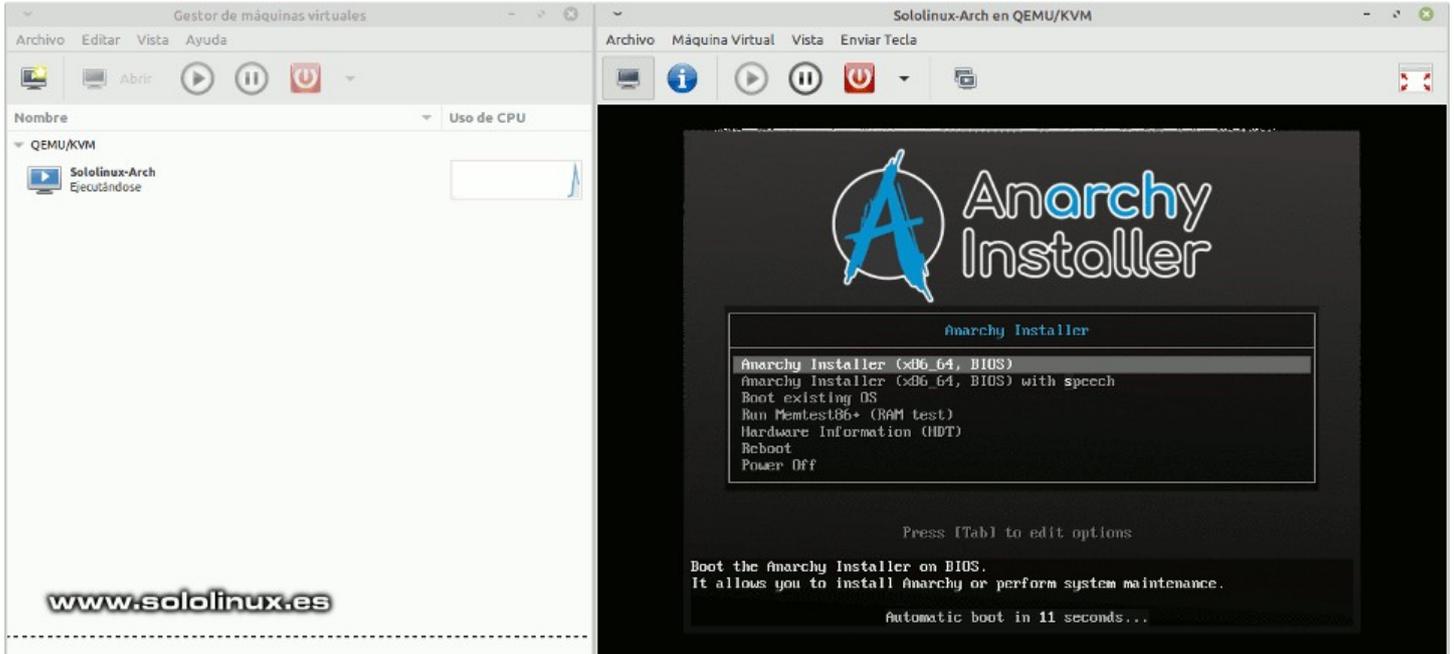
También el tamaño del disco. Crear y administrar máquinas virtuales con Virt-Manager.



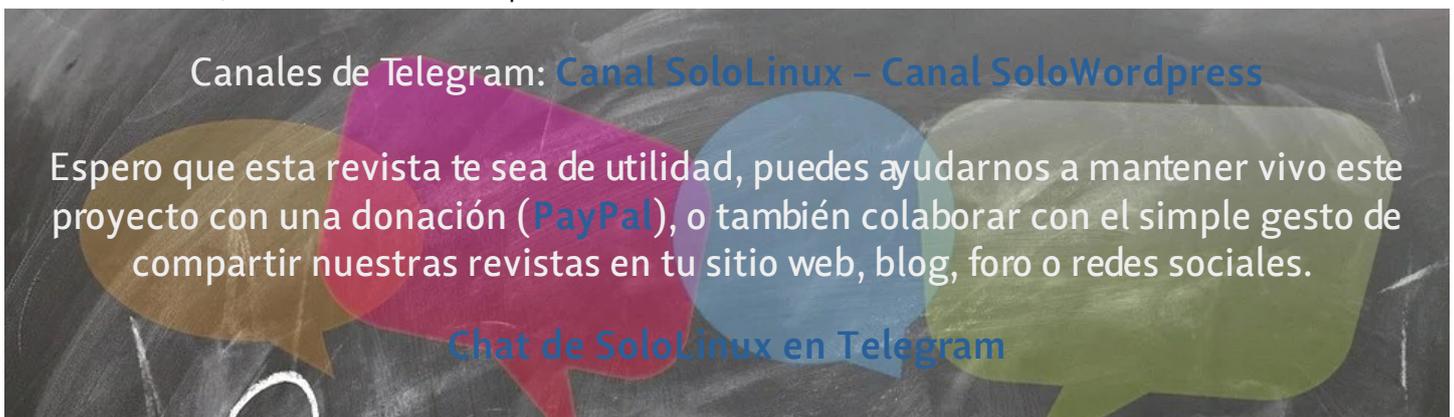
Hacemos clic otra vez en «adelante». Para crear la **máquina virtual**, solo nos falta ingresar el nombre de la máquina.



El inicio es prácticamente instantáneo. Si por algún extraño caso desaparece el puntero del ratón, pulsa las teclas «**Ctrl + Alt**».



NOTA: A no ser que tengas un sistema de última generación, conviene instalar KVM como te indique anteriormente. De todas formas, esta herramienta te sorprenderá.

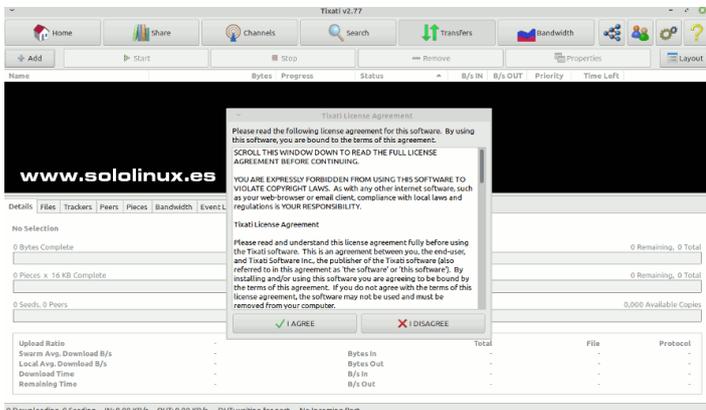


Instalar Tixati Torrent en Ubuntu 20.04 y otros linux



Tixati es una herramienta de intercambio de archivos, bajo el protocolo BitTorrent. Con tan solo un click tiene la capacidad de abrir archivos Torrent, además de los enlaces Magnet. Además de una útil función de chat, este cliente torrent incluye detalles de la red en su medio mediante varias gráficas. Por ejemplo, nos detalla la velocidad y las estadísticas de datos en tiempo real.

Es compatible con DHT, PEX y Magnet Link, permite la selección de pares y bloqueos; Cifrado de conexión RC4 para una mayor seguridad, enrutadores NAT, RSS, filtrado de IP, programador de eventos y más. Como puedes comprobar, este ligero cliente de torrent es una opción a tener en cuenta.



Instalar Tixati Torrent en Ubuntu 20.04 y otros linux

Nosotros instalamos la última versión disponible, en este caso Tixati 2.77-1. De todas formas, puedes verificar la última versión en su zona de descarga oficial. No perdamos más tiempo, vamos a instalar Tixati Torrent.

Instalar Tixati en linux con deb

Compatible con Debian, Ubuntu, Linux Mint y muchos más.

Versión 64 bits

```
wget https://download2.tixati.com/download/tixati_2.77-1_amd64.deb
sudo apt install ./tixati_*_amd64.deb
```

Versión 32 bits

```
wget https://download2.tixati.com/download/tixati_2.77-1_i686.deb
sudo apt install ./tixati_*_i686.deb
```

```
sergio@sololinux:~$ sudo apt install ./tixati_*_amd64.deb
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Nota, seleccionando «tixati» en lugar de «./tixati_2.77-1_amd64.deb»
Se instalarán los siguientes paquetes NUEVOS:
  tixati
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 0 B/10,8 MB de archivos.
Se utilizarán 35,8 MB de espacio de disco adicional después de esta operación.
Des:1 /home/sergio/tixati_2.77-1_amd64.deb tixati amd64 2.77-1 [10,8 MB]
Seleccionando el paquete tixati previamente no seleccionado.
(Leyendo la base de datos ... 342704 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../sergio/tixati_2.77-1_amd64.deb ...
Desempaquetando tixati (2.77-1) ...
Configurando tixati (2.77-1) ...
Procesando disparadores para hicolor-icon-theme (0.17-2) ...
Procesando disparadores para mime-support (3.60ubuntu1) ...
.....
```



Instalar Tixati en linux con rpm

Compatible con RHEL, CentOS, Fedora, Open Suse, Mandriva y muchos más.

Versión 64 bits

```
wget https://download2.tixati.com/download/tixati-2.77-1.x86_64.rpm
sudo dnf -i ./tixati-2.77-1.x86_64.rpm
# 0
sudo yum -i ./tixati-2.77-1.x86_64.rpm
```

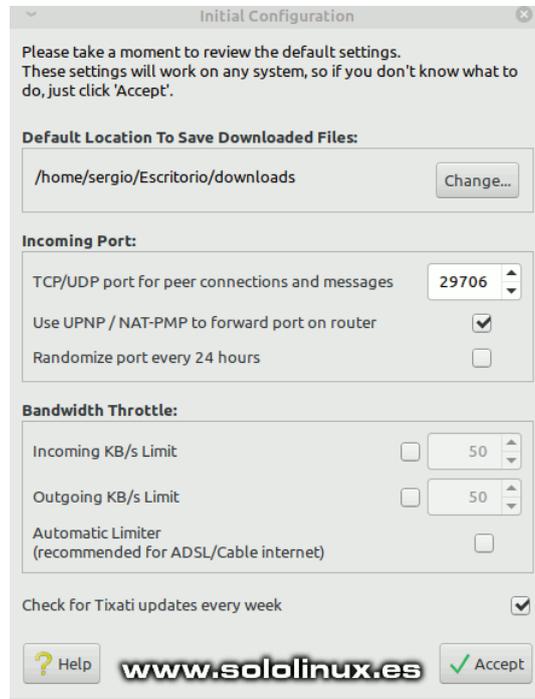
Versión 32 bits

```
wget https://download2.tixati.com/download/tixati-2.77-1.i686.rpm
sudo dnf -i ./tixati-2.77-1.i686.rpm
# 0
sudo yum -i ./tixati-2.77-1.i686.rpm
```

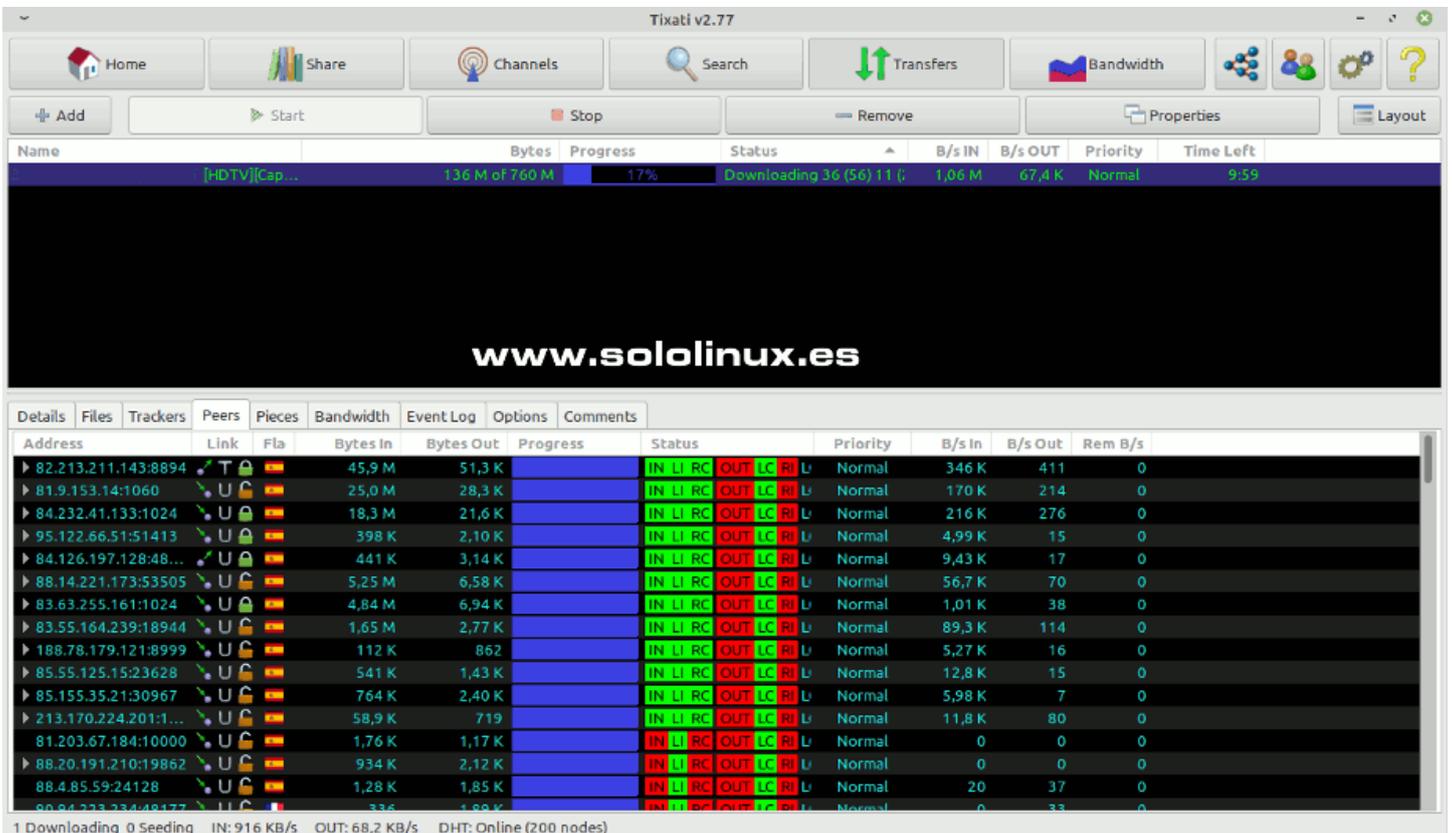
Canales de Telegram:
Canal Sololinux
Canal Solo Wordpress

Ejecutar Tixati

Una vez instales **Tixati**, cualquier archivo torrent se abrirá de forma automática con este **cliente torrent**. La primera vez que inicie, te solicitará aceptar la configuración por defecto o modificar algún valor, es tu decisión.



Como último ejemplo, vemos una imagen de Tixati en acción. Instalar Tixati Torrent.



Uso del Comando stat en linux

El comando **stat**, es un comando simple que podíamos definir como una versión mejorada de la herramienta «**ls -l**». Con «**ls**», el indicador «**-l**» nos ofrece más detalles de los archivos, como el propietario y los permisos, pero el **comando stat** profundiza aún más en ellos, imprimiendo en la consola una gran cantidad de información sobre el archivo requerido.

A diferencia de «**ls**», la herramienta en consola **stat** no tiene tantas opciones. Sin embargo lo que hace, lo hace muy bien, incluyendo datos del archivo que realmente te sorprenderán. **Stat** viene por defecto en cualquier **distribución linux** que se precie, por tanto no tendrás ningún problema con ella. En este artículo vemos algunos ejemplos de uso.



```
[~]$ stat
```

www.sololinux.es

Uso del comando stat en linux

En su sintaxis más simple, sin ningún parámetro adicional, el **comando stat** muestra la salida predeterminada. Esto incluye mucha y valiosa información, como el tamaño y el patrón de archivo, también el tipo de dispositivo, el número de **inodo**, UID, GID, y también el número de enlaces además de las fechas de acceso y modificación del archivo.

Para crear el artículo, nosotros tomamos como ejemplo un **script bash** conocido por todos los seguidores de **sololinux**, dada su gran utilidad a la hora de limpiar **servidores linux**, es evidente que hablamos de «**libera.sh**».

```
stat libera.sh
```

Observa la información aportada en el siguiente ejemplo...

```
sergio@sololinux:~$ stat libera.sh
  Fichero: libera.sh
  Tamaño: 1986      Bloques: 8      Bloque E/S: 4096  fichero regular
Dispositivo: 802h/2050d Nodo-i: 4984706  Enlaces: 1
Acceso: (0644/-rw-r--r--) Uid: ( 1000/ sergio)  Gid: ( 1000/ sergio)
Acceso: 2020-12-07 17:35:23.682788999 +0200
Modificación: 2020-12-07 15:35:56.184869372 +0200
      Cambio: 2020-12-07 15:35:56.184869372 +0200
  Creación: -
```

Con el comando **stat**, también es posible identificar dos archivos o más.

```
stat libera.sh libera1.sh
```

Veremos una salida similar a...

```
sergio@sololinux:~$ stat libera.sh libera1.sh
  Fichero: libera.sh
  Tamaño: 1986      Bloques: 8      Bloque E/S: 4096  fichero regular
Dispositivo: 802h/2050d Nodo-i: 4984706  Enlaces: 1
Acceso: (0644/-rw-r--r--) Uid: ( 1000/ sergio)  Gid: ( 1000/ sergio)
Acceso: 2020-12-07 17:35:23.682788999 +0200
Modificación: 2020-12-07 15:35:56.184869372 +0200
      Cambio: 2020-12-07 15:35:56.184869372 +0200
  Creación: -
  Fichero: libera1.sh
  Tamaño: 5840      Bloques: 16     Bloque E/S: 4096  fichero regular
Dispositivo: 802h/2050d Nodo-i: 4989994  Enlaces: 1
Acceso: (0644/-rw-r--r--) Uid: ( 1000/ sergio)  Gid: ( 1000/ sergio)
Acceso: 2020-12-07 17:35:43.247842441 +0200
Modificación: 2020-12-07 17:38:22.044392534 +0200
      Cambio: 2020-12-07 17:55:56.909189646 +0200
  Creación: -
```

La información aportada nos indica estos datos:

- **Fichero** : Nombre del archivo.
- **Tamaño** : Tamaño del archivo en bytes.
- **Bloques** : Número de bloques asignados al archivo.
- **Bloque E/S (IO)** : Bytes de cada bloque.
- **Dispositivo** : Número de dispositivo en formato hexadecimal o decimal.
- **Inode** : Inodo del archivo.
- **Enlaces** : Enlaces físicos asociados al archivo.
- **Acceso** : Indica los permisos del archivo en formato simbólico o numérico.
- **Uid** : ID del usuario y nombre del propietario.
- **Gid** : ID del grupo y nombre del propietario.
- **Acceso** : Nos muestra la última vez que se accedió al archivo.
- **Modificación**: Última vez que se modificó el contenido del archivo.
- **Cambio** : Última vez que los metadatos de un archivo se modificaron, por ejemplo, los permisos y propiedad.

Si te parece un poco apabullante tanto texto, puedes concentrarlo todo aplicando la opción «-t».

```
stat -t libera.sh
```

Vemos una imagen de ejemplo...

```
sergio@sololinux:~$ stat -t libera.sh
libera.sh 1986 8 81a4 1000 1000 802 4984706 1 0 0 1607355323 1607348156 1607348156 0 4096
```

También podemos visualizar el estado de un directorio, por ejemplo de nuestra carpeta /home. **Comando stat.**

```
stat -f /home
```

Nos aparece lo siguiente....

```
sergio@sololinux:~$ stat -f /home
Fichero: "/home"
ID: 3f5dcdd55de413a9 Longnombre: 255 Tipo:
ext2/ext3
Tam. bloque: 4096 Tam. bloque fundamental: 4096
Bloques: Total: 30507684 Libres: 20668236 Disponibles:
19107776
Nodos-i: Total: 7782400 Libres: 7335692
```

Con «-help», visualizamos la ayuda del comando stat.

```
stat --help
```

```
sergio@sololinux:~$ stat --help
Modo de empleo: stat [OPCIÓN]... FICHERO...
Muestra el estado del fichero o del sistema de ficheros.
Los argumentos obligatorios para las opciones largas son también
obligatorios
para las opciones cortas.
-L, --dereference sigue los enlaces
-f, --file-system muestra el estado del sistema de ficheros en
lugar del
estado del fichero
-c --format=FORMATO usa el FORMATO especificado en vez del
predeterminado;
muestra una nueva línea después de cada uso
del FORMATO
--printf=FORMATO como --format, pero interpreta las secuencias
de escape,
y no muestra un carácter de nueva línea
final
obligatoriamente; si quiere un carácter de
nueva
línea, incluya \n en el FORMATO
```

```
-t, --terse muestra la información de manera escueta
--help muestra esta ayuda y finaliza
--version informa de la versión y finaliza
The valid format sequences for files (without --file-system):
%a access rights in octal (note '#' and '0' printf flags)
%A access rights in human readable form
%b number of blocks allocated (see %B)
%B the size in bytes of each block reported by %B
%C SELinux security context string
%d número de dispositivo en decimal
%D número de dispositivo en hexadecimal
%f modo en hexadecimal
%F tipo de fichero
%g ID del grupo del propietario
%G nombre de grupo del propietario
%h número de enlaces duros
%i número de nodo-i
%m punto de montaje
%n nombre del fichero
%N nombre de fichero entrecomillado desreferenciado si era un
enlace
simbólico
%o indicación del tamaño óptimo de bloque de E/S
%s tamaño total, en bytes
%t tipo de dispositivo principal en hexadecimal, para ficheros
especiales
de caracteres o dispositivos de bloques
%T tipo de dispositivo secundario en hexadecimal, para ficheros
especiales
de caracteres o dispositivos de bloques
%u ID de usuario del propietario
%U nombre de usuario del propietario
%w fecha de creación del fichero, formato legible, - si es
desconocido
%W fecha de creación, segundos desde la Época, 0 si es
desconocido
%x fecha de último acceso, formato legible
%X fecha de último acceso, segundos desde la Época
%y fecha de última modificación en los datos, formato legible
%Y fecha de última modificación en los datos, segundos desde la
Época
%z fecha de último cambio en el estado, formato legible
%Z fecha de último cambio en el estado, segundos desde la Época
Formatos válidos para sistemas de ficheros:
%a bloques libres disponibles para el no superusuario
%b total de bloques de datos en el sistema de ficheros
%c total de nodos de ficheros en el sistema de ficheros
%d nodos de ficheros libres en el sistema de ficheros
%f bloques libres en el sistema de ficheros
%i id del sistema de ficheros en hexadecimal
%l longitud máxima de los nombres de ficheros
%n nombre del fichero
%s tamaño de bloque de transferencia óptima
%S tamaño de bloque fundamental (para contar bloques)
%t tipo en hexadecimal
%T tipo de forma legible
NOTA: su shell puede tener su propia versión de stat, que usualmente
tiene prioridad sobre la versión que se describe aquí. Por favor
acuda
a la documentación de su shell para saber los detalles sobre las
opciones que admite.
ayuda en línea sobre GNU coreutils:
<http://www.gnu.org/software/coreutils/>
Informe de errores de traducción en stat a
<http://translationproject.org/team/es.html>
Full documentation at: <http://www.gnu.org/software/coreutils/stat>
or available locally via: info '(coreutils) stat invocation'
```

Como puedes verificar, la opción help del comando stat nos ofrece una valiosa información sobre la herramienta. Nosotros concluimos la presentación de este comando, identificando la versión instalada en nuestro sistema.

```
stat --version
```

Imagen de ejemplo, con la versión instalada en nuestro linux.

```
sergio@sololinux:~$ stat --version
stat (GNU coreutils) 8.28
Copyright © 2017 Free Software Foundation, Inc.
Licencia GPLv3+: GPL de GNU versión 3 o posterior
<http://gnu.org/licenses/gpl.html>.
Esto es software libre: usted es libre de cambiarlo y redistribuirlo.
No hay NINGUNA GARANTÍA, hasta donde permite la ley.
Escrito por Michael Meskes. www.sololinux.es
```

Instalar y Configurar FirewallD en Debian 10

En un artículo anterior, vimos como [configurar FirewallD en CentOS 8](#). Esta herramienta es nativa y creada específicamente para RHEL, CentOS, Fedora y demás distribuciones que usan **rpm**.

Aun siendo así, **FirewallD** también se puede instalar en otro tipo de **distribuciones linux**, en nuestro artículo lo haremos en un **Debian 10 Desktop XFCE**. Es cierto que Debian 10 viene por defecto con **UFW**, pero algunos usuarios confían ciegamente en la potencia de **FirewallD**.



Instalar y configurar FirewallD en Debian 10

Tal vez te preguntes por qué **FirewallD** es tan bueno, sus características hablan por si solas.

- API D-Bus completa.
- Soporte completo para IPv4, IPv6, bridge e ipset.
- Compatible con NAT IPv4 e IPv6.
- Especificar zonas de cortafuegos.
- Lista predefinida de zonas, servicios y tipos de icmp.
- Servicio simple, puertos, protocolos, puertos de origen, enmascaramiento, reenvío de puertos, filtro icmp, reglas enriquecida, completa interfaz y el control de direcciones de origen en las zonas.
- Definición del servicio simple con puertos, protocolos, puertos de origen, módulos de netfilter y manejo de direcciones de destino.
- Lenguaje enriquecido que nos permiten crear reglas más flexibles y complejas.
- Reglas de firewall por tiempo en las zonas.
- Registro simple de paquetes denegados.
- Excelente interfaz.
- Lista blanca de aplicaciones que pueden modificar el firewall.
- Carga automatizada de los módulos del kernel Linux.
- Integración con Puppet.
- Configuración online y offline.
- Herramienta de configuración gráfica mediante gtk3.
- Applet para Qt4.

Esta herramienta de Firewall, viene incluida en los repositorios oficiales de **Debian 10 Buster**, su instalación es tan simple como ejecutar los siguientes comandos.

```
sudo apt update
sudo apt -y install firewalld
```

Ejemplo...

```
root@sololinux-demo:~# sudo apt -y install firewalld
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
gir1.2-glib-2.0 ipset libgirepository-1.0-1 libipset11
python3-dbus
python3-decorator python3-gi python3-selinux python3-six
python3-slip
python3-slip-dbus
Paquetes sugeridos:
python-dbus-doc python3-dbus-dbg
Se instalarán los siguientes paquetes NUEVOS:
firewalld gir1.2-glib-2.0 ipset libgirepository-1.0-1
libipset11
python3-dbus python3-decorator python3-gi python3-selinux
python3-six
python3-slip python3-slip-dbus
0 actualizados, 12 nuevos se instalarán, 0 para eliminar y
0 no actualizados.
Se necesita descargar 1.265 kB de archivos.
Se utilizarán 7.286 kB de espacio de disco adicional
después de esta operación.
Des:1 http://ftp.debian.org/debian buster/main amd64
libgirepository-1.0-1 amd64 1.58.3-2 [92,8 kB]
Des:2 http://ftp.debian.org/debian buster/main amd64
gir1.2-glib-2.0 amd64 1.58.3-2 [143 kB]
Des:3 http://ftp.debian.org/debian buster/main amd64
python3-dbus amd64 1.2.8-3 [103 kB]
Des:4 http://ftp.debian.org/debian buster/main amd64
python3-gi amd64 3.30.4-1 [180 kB]
.....
```

Verificamos los detalles de Firewalld en nuestro Debian.

```
sudo apt policy firewalld

root@sololinux-demo:~# sudo apt policy firewalld
firewalld:
  Instalados: 0.6.3-5
  Candidato: 0.6.3-5
  Tabla de versión:
*** 0.6.3-5 500
    500 http://ftp.debian.org/debian buster/main amd64 Packages
    100 /var/lib/dpkg/status
```

Iniciamos y habilitamos el servicio para que inicie con el sistema.

```
systemctl start firewalld
systemctl enable firewalld
```

Ahora verificamos que tenemos corriendo firewalld en nuestro sistema.

```
sudo firewall-cmd --state
```

También su estado.

```
systemctl status firewalld
```

Verás algo similar a...

```
root@sololinux-demo:~# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/lib/systemd/system/firewalld.service;
   enabled; vendor preset
   Active: active (running) since Sun 2020-12-08 14:50:14
   EDT; 4min 51s ago
     Docs: man:firewalld(1)
    Main PID: 7955 (firewalld)
       Tasks: 2 (limit: 2319)
      Memory: 24.5M
    CGroup: /system.slice/firewalld.service
           └─7955 /usr/bin/python3 /usr/sbin/firewalld --
   nofork --nopic
```

Si obtienes errores al ejecutar algún comando de los anteriores, [revisa este artículo](#).

Llegado este punto, es importante deshabilitar UFW y reiniciar el sistema.

```
sudo ufw disable
sudo reboot
```

```
root@sololinux-demo:~# sudo ufw disable
Firewall stopped and disabled on system startup
root@sololinux-demo:~# sudo reboot
Connection to 185.185.40.142 closed by remote host.
Connection to 185.185.40.142 closed.
```

Configurar Firewalld en Debian 10
Llego el momento de **configurar Firewalld en Debian 10 Buster**. Comenzamos listando las reglas establecidas por defecto.

```
sudo firewall-cmd --list-all
```

Nos aparecen la reglas establecidas. Los servicios ssh y dhcpv6-client, están habilitados de forma predeterminada al iniciar firewalld.

```
root@sololinux-demo:~# sudo firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: wlan0
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Listamos los servicios que se permite habilitar o deshabilitar.

```
sudo firewall-cmd --get-services
```

La sintaxis para habilitar un servicio es básica.

```
sudo firewall-cmd --add-service="servicio" --permanent
```

En el siguiente ejemplo habilitamos http, posteriormente reiniciamos el servicio.

```
sudo firewall-cmd --add-service="http" --permanent
success
sudo firewall-cmd --reload
```

Listamos los servicios habilitados.

```
sudo firewall-cmd --add-service={http,https,smtp,imap} --
permanent --zone=public
```

Si por ejemplo queremos habilitar TCP...

```
sudo firewall-cmd --add-port=port/tcp --permanent
sudo firewall-cmd --reload
```

También los puertos **8080** y **8443**.

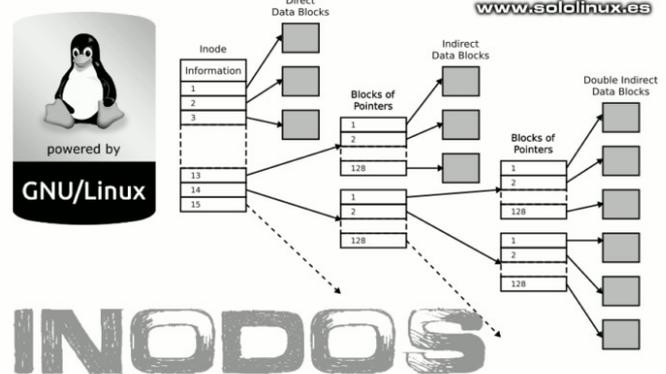
```
sudo firewall-cmd --zone=public --add-port=8080/tcp --
permanent
sudo firewall-cmd --zone=public --add-port={8080,8443}/tcp
--permanent
sudo firewall-cmd --reload
```



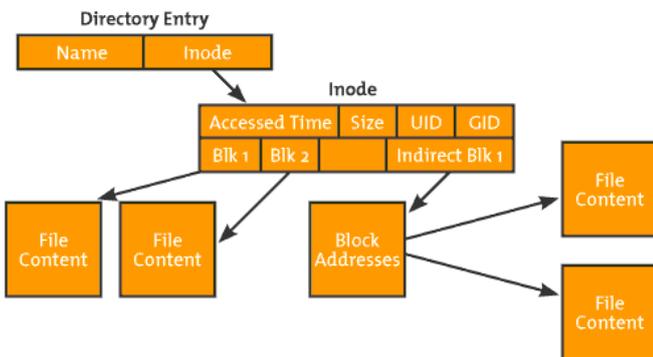
Que son los inodos en Linux

El **inodo** es una estructura de datos contenedora de los **metadatos** de un archivo. Cuando nuestro **sistema de archivos** almacena uno nuevo, no solo se guarda el contenido (datos) del archivo, también otras propiedades adicionales como el nombre del archivo, la fecha de creación, permisos, el propietario y más.

Exceptuando el nombre y el contenido, el resto de información se almacena en un inodo específico del archivo. En este artículo vemos como listarlos, así como su contenido y otros datos que pueden resultar importantes.



INODOS



Que son los inodos en linux

El comando **ls -li**, nos mostrará información del inodo, como podemos comprobar a continuación.

Ver los inodos

Inodos de la carpeta actual:

```
ls -li
sergio@sololinux:~$ ls -li
total 92
-rw-r--r-- 1 sergio sergio 20480 sep 12 17:33 Databases.db
drwxr-xr-x 6 sergio sergio 12288 dic 9 13:05 Descargas
drwxr-xr-x 7 sergio sergio 4096 nov 13 12:11 Documentos
drwxr-xr-x 6 sergio sergio 4096 dic 9 14:53 Escritorio
drwxr-xr-x 4 sergio sergio 4096 sep 5 09:10 Imágenes
-rw-r--r-- 1 sergio sergio 1986 dic 7 15:35 libera.sh
drwxr-xr-x 4 sergio sergio 4096 dic 5 16:43 Música
drwxr-xr-x 8 root root 4096 sep 2 16:53 rtl8723de
drwxrwxr-x 3 sergio sergio 4096 sep 16 12:25 rtw88
```

En este caso especificamos una ruta.

```
ls -ld /carpeta/archivo
```

Observa el inodo de **/etc**.

```
root@sololinux-demo:~# sudo ls -ld /etc
drwxr-xr-x 81 root root 4096 dic 9 13:55 /etc
```

Que son los inodos. La tabla de inodos contiene todos los inodos de nuestro **Linux**, y se genera al crear el sistema de archivos. Puedes usar el comando **df -i** para ver cuántos inodos están en uso y cuántos están libres. También podemos identificar los detalles de todos los volúmenes montados.

```
df -i
sergio@sololinux:~$ df -i
S.ficheros Nodos-i NUsados NLibres NUsos% Montado en
udev 482292 592 481700 1% /dev
tmpfs 491747 887 490860 1% /run
/dev/sda2 7782400 448537 7333863 6% /
tmpfs 491747 126 491621 1% /dev/shm
tmpfs 491747 6 491741 1% /run/lock
tmpfs 491747 18 491729 1% /sys/fs/cgroup
/dev/sda1 0 0 0 - /boot/efi
tmpfs 491747 28 491719 1% /run/user/1000
/dev/sdb1 0 0 0 -
/media/sergio/DESCARGAS
/dev/sdc1 0 0 0 -
/media/sergio/204E-01F8
```

Número de inodos

El número de inodo, también conocido como número de índice. Es un número único que se asigna a los archivos y directorios en el momento de su creación. El número de inodo es único en todo el sistema de archivos.

Podemos ver los números de inodo con el comando **ls -li** (se admiten rutas). Excepto el nombre del archivo, todos los datos aportados se contienen en el inodo.

```
ls -li /etc
root@sololinux-demo:~# ls -li /etc
total 700
524584 -rw-r--r-- 1 root root 2981 jul 2 12:10 adduser.conf
524357 drwxr-xr-x 2 root root 4096 dic 8 13:39 alternatives
524867 drwxr-xr-x 3 root root 4096 dic 8 11:10 apparmor.d
524318 drwxr-xr-x 7 root root 4096 jul 2 12:11 apt
524369 -rw-r--r-- 1 root root 2008 jul 2 12:11 bash.bashrc
524300 -rw-r--r-- 1 root root 367 mar 2 2018 bindresvport.blacklist
524526 drwxr-xr-x 2 root root 4096 abr 27 2020 binfmt.d
524897 drwxr-xr-x 3 root root 4096 jul 2 12:10 ca-certificates
525086 -rw-r--r-- 1 root root 5434 jul 2 12:11 ca-certificates.conf
524860 drwxr-xr-x 2 root root 4096 jul 2 12:10 calendar
524899 drwxr-xr-x 2 root root 4096 jul 2 12:11 cifs-utils
524877 drwxr-xr-x 2 root root 4096 jul 2 12:10 console
524904 drwxr-xr-x 2 root root 4096 jul 2 12:11 cracklib
524836 drwxr-xr-x 2 root root 4096 jul 2 12:10 cron.d
524359 drwxr-xr-x 2 root root 4096 jul 2 12:11 cron.daily
524842 drwxr-xr-x 2 root root 4096 jul 2 12:10 cron.hourly
524844 drwxr-xr-x 2 root root 4096 jul 2 12:10 cron.monthly
524298 -rw-r--r-- 1 root root 1039 dic 8 11:07 crontab
524846 drwxr-xr-x 2 root root 4096 jul 2 12:11 cron.weekly
524509 drwxr-xr-x 4 root root 4096 dic 8 11:15 dbus-1
524321 -rw-r--r-- 1 root root 2969 feb 26 2019 debconf.conf
525422 -rw-r--r-- 1 root root 5 nov 22 13:37 debian version
524305 drwxr-xr-x 2 root root 4096 dic 8 13:00 default
524511 -rw-r--r-- 1 root root 604 jun 26 2016 deluser.conf
524527 drwxr-xr-x 4 root root 4096 jul 2 12:09 dhcp
524343 drwxr-xr-x 5 root root 4096 jul 2 12:10 dpkg
524680 -rw-r--r-- 1 root root 0 jul 2 12:10 environment
525038 drwxr-xr-x 7 root root 4096 dic 8 13:39 firewalld
524911 drwxr-xr-x 4 root root 4096 jul 2 12:11 fonts
524301 -rw-r--r-- 1 root root 144 dic 8 13:54 fstab
524306 -rw-r--r-- 1 root root 2584 ago 1 2018 gai.conf
524864 drwxr-xr-x 2 root root 4096 jul 2 12:10 groff
525483 -rw-r--r-- 1 root root 703 dic 8 11:15 group
```

Escanear un servidor Linux Con Nikto



SCAN SERVER LINUX

www.sololinux.es

Nikto es una herramienta con una finalidad concreta, escanear un **servidor Linux** (la seguridad) de forma rápida y eficiente, con el fin de identificar posibles vulnerabilidades, así como herramientas potencialmente peligrosas instaladas en el mismo.

Fácil de usar y aún más de instalar, **Nikto** se basa en **LibWhisker2** (por RFP) y puede ejecutarse en cualquier plataforma de **servidor Linux** que tenga un entorno Perl. Es compatible con SSL, con proxy, con autenticación de host, codificación de ataques y mucho más. Además se permite actualizar de forma automática desde tu terminal / consola.

También verifica todos los elementos de configuración del servidor, como los archivos del índice, las opciones y elementos del **servidor HTTP**, y el software instalado. El resultado de **escanear un servidor Linux con Nikto**, se imprime en línea de comandos. Podrás identificar rápidamente donde tienes los problemas, incluso si estás usando versiones obsoletas. Escanear un servidor Linux con Nikto.

Escanear un servidor linux con Nikto

Nikto es una herramienta muy rápida, por tanto... puedes deducir que de manera predeterminada no es sigilosa, deja rastros en el servidor linux escaneado. **No lo utilices en servidores o dominios que no sean de tu propiedad.**

Características de Nikto

- Trabaja con SSL.
- Soporte completo para proxy HTTP.
- Verificación de componentes del servidor obsoletos.
- Guarda informe en texto plano, XML, HTML, NBE o CSV.
- Permite personalizar los informes.
- Capacidad de escanear varios puertos de un servidor (o varios servidores), desde un archivo de entrada (por ejemplo la salida de **nmap**).
- Técnicas de codificación IDS de LibWhisker.
- Fácil de actualizar desde línea de comandos.
- Identifica el software instalado desde los encabezados, favicons y archivos.
- Autenticación del host con Basic y NTLM.
- Localiza subdominios.
- Lista el nombre de usuario de Apache y cgiwrap.
- Técnicas de mutación para mejorar la captura del contenido en servidores web.
- Permite incluir o excluir clases completas de verificaciones de vulnerabilidad.
- Se limitan los falsos positivos a través de los encabezados, el contenido de la página y su hash.
- Informa de encabezados extraños.
- Estado interactivo, pausa y cambios en la configuración.
- Guardar solicitudes y respuestas completas.
- Permite configurar el tiempo máximo de ejecución por objetivo.
- Permite configurar una pausa automática.
- Iniciar sesión en **Metasploit**.
- Excelente [documentación oficial](#).



Instalar Nikto en linux

Instalamos la herramienta Nikto en nuestra distribución linux favorita.

Instalar Nikto en Debian, Ubuntu y derivados

```
sudo apt install nikto libnet-ssleay-perl
```

Instalar Nikto en Fedora y derivados

```
sudo dnf install nikto
```

Instalar Nikto en Arch Linux, Manjaro y derivados

```
sudo pacman -S nikto
```

Instalar Nikto en Open Suse y derivados

```
sudo zypper install nikto
```

Instalar Nikto en linux con GIT

```
git clone https://github.com/sullo/nikto
cd nikto/program
# Ejecutar desde el interprete
./nikto.pl -h [servidor]
# Ejecutar con perl
perl nikto.pl -h [servidor]
```

Ejecutar Nikto

Una vez instalada la herramienta verifica que tienes la última versión.

```
nikto -update
```

Ahora lanzamos un escaneo normal contra una ip o dominio.

```
nikto -h [ip-dominio]
```

Vemos un ejemplo de salida, incluyendo sus errores.

```
root@sololinux:~# nikto -h 198.157.21.85
- Nikto v2.1.5
-----
+ Target IP:          198.157.21.85
+ Target Hostname:    hostame-demo
+ Target Port:        80
+ Start Time:         2020-12-10 14:03:28 (GMT2)
-----
+ Server: nginx
+ Server leaks inodes via ETags, header found with file /,
fields: 0xf09 0x5a8592b5feed6
+ The anti-clickjacking X-Frame-Options header is not
present.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 6544 items checked: 0 error(s) and 5 item(s) reported on
remote host
+ End Time:           2020-12-10 14:10:02 (GMT2) (394
seconds)
-----
+ 1 host(s) tested
```

```
root@sololinux:~# nikto -Tuning x 6 -h [redacted]
- Nikto v2.1.5
-----
+ Target IP:          [redacted]
+ Target Hostname:    www.sololinux.es
+ Target Port:        80
+ Start Time:         2020-12-10 14:16:04 (GMT2)
-----
+ Server: nginx
+ Server leaks inodes via ETags, header found with file /, fields: 0xf09 0x5a8592b5feed6
+ The anti-clickjacking X-Frame-Options header is not present.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 6544 items checked: 0 error(s) and 5 item(s) reported on host
+ End Time:           2020-12-10 14:22:36 (GMT2) (392 seconds)
-----
+ 1 host(s) tested
```

Otro ejemplo interesante es con salida detallada.

```
nikto -Display V -h [ip-dominio]
```

La salida del comando anterior, puede resultar extremadamente grande.

```
root@sololinux:~# nikto -Display V -h 147.135.138.64
- Nikto v2.1.5
-----
V:Thu Dec 10 15:31:14 2020 - Initialising plugin
nikto_headers
V:Thu Dec 10 15:31:14 2020 - Loaded "HTTP Headers" plugin.
V:Thu Dec 10 15:31:14 2020 - Initialising plugin
nikto_tests
V:Thu Dec 10 15:31:14 2020 - Loaded "Nikto Tests" plugin.
V:Thu Dec 10 15:31:14 2020 - Initialising plugin
nikto_report_nbe
V:Thu Dec 10 15:31:14 2020 - Loaded "NBE reports" plugin.
V:Thu Dec 10 15:31:14 2020 - Initialising plugin
nikto_report_msf
V:Thu Dec 10 15:31:14 2020 - Loaded "Report data to
Metasploit" plugin.
V:Thu Dec 10 15:31:14 2020 - Initialising plugin
nikto_subdomain
V:Thu Dec 10 15:31:14 2020 - Loaded "Sub-domain forcer"
plugin.
V:Thu Dec 10 15:31:14 2020 - Initialising plugin
nikto_put_del_test
V:Thu Dec 10 15:31:14 2020 - Loaded "Put/Delete test"
plugin.
V:Thu Dec 10 15:31:14 2020 - Initialising plugin
nikto_paths
V:Thu Dec 10 15:31:14 2020 - Loaded "Path Search" plugin.
V:Thu Dec 10 15:31:14 2020 - Initialising plugin
nikto_favicon
V:Thu Dec 10 15:31:14 2020 - Loaded "Favicon" plugin.
V:Thu Dec 10 15:31:14 2020 - Initialising plugin
nikto_report_text
V:Thu Dec 10 15:31:14 2020 - Loaded "Text reports" plugin.
V:Thu Dec 10 15:31:14 2020 - Initialising plugin
nikto_apacheusers
V:Thu Dec 10 15:31:14 2020 - Loaded "Apache Users" plugin.
V:Thu Dec 10 15:31:14 2020 - Initialising plugin nikto_ssl
V:Thu Dec 10 15:31:14 2020 - Loaded "SSL and cert checks"
plugin.
V:Thu Dec 10 15:31:14 2020 - Initialising plugin
nikto_multiple_index
.....
.....
```

Como puedes comprobar su uso es bastante simple, pero si quieres profundizar en la herramienta te sugiero que revises su ayuda integrada con este comando.

```
nikto -help
```

Ejecutar Nikto con perl

En el caso de que instales la herramienta Nikto con GIT, tal vez te interese trabajar con ella en perl. Entonces se debe modificar el comando principal, pero las opciones son las mismas que vimos anteriormente. Vemos la sintaxis con perl.

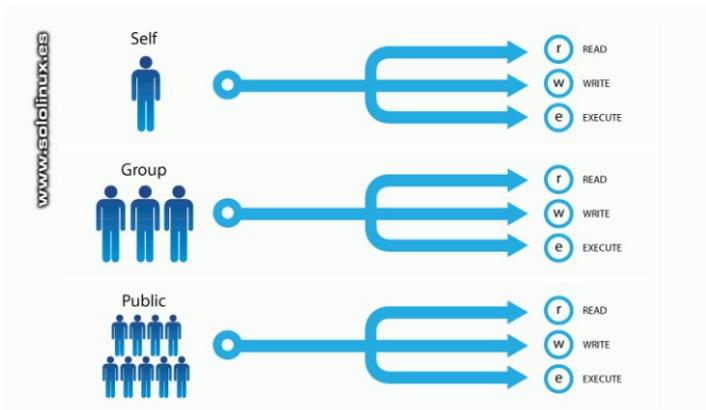
```
perl nikto.pl [opcion] [ip-o-dominio]
```

Manual de Nikto

Te recomiendo encarecidamente que revises su manual oficial, es muy bueno.

- [Manual de Nikto](#)

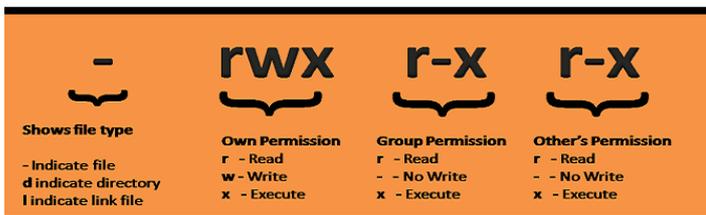
Permisos de archivo estándar simples en linux



Una de las particularidades del sistema operativo Linux, es que permite a varios usuarios trabajar en el mismo sistema de forma simultánea sin entorpecerse entre ellos. Compartir el **acceso a los archivos**, representa el riesgo de exponer información personal con otros o usuarios, o incluso en casos extremos, la pérdida de datos.

Para abordar este potencial problema, nuestro **sistema linux** nos permite especificar los **permisos** que cada usuario tiene sobre un archivo o directorio determinado. También es posible aplicar permisos a **grupos de usuarios**.

Los permisos se basan en tres caracteres:
R-W-X / Leer – Escribir – Ejecutar



Permisos de archivo estándar en linux

Si observas la imagen anterior, te darás cuenta de que los nueve caracteres que siguen al tipo de archivo establecen los permisos. Un permiso puede ser «**r**» para conceder acceso de lectura, «**w**» para acceso de escritura y «**x**» para permitir su ejecución.

Por ejemplo, si quieres listar contenido, necesitas el permiso «**r**». Si necesitas abrir, el permiso «**x**». Con el permiso «**w**» podrás crear o eliminar archivos de un directorio. **Permisos de archivo estándar simples en linux**. Vemos un ejemplo con permisos limitados.

```
sergio@sololinux:~$ ls -l libera.sh
-rw-r--r-- 1 sergio sergio 1986 dic 7 15:35 libera.sh
```

Otro ejemplo de la carpeta de usuario.

```
sergio@sololinux:~$ ls -lh
total 92K
-rw-r--r-- 1 sergio sergio 20K sep 12 17:33 Databases.db
drwxr-xr-x 6 sergio sergio 12K dic 11 16:27 Descargas
drwxr-xr-x 7 sergio sergio 4,0K nov 13 12:11 Documentos
drwxr-xr-x 4 sergio sergio 4,0K sep 5 09:10 Imágenes
-rw-r--r-- 1 sergio sergio 2,0K dic 7 15:35 libera.sh
drwxr-xr-x 4 sergio sergio 4,0K dic 5 16:43 Música
drwxr-xr-x 2 sergio sergio 4,0K sep 2 16:49 Público
drwx----- 6 sergio sergio 4,0K sep 11 16:49 Viber
drwxr-xr-x 6 sergio sergio 4,0K dic 11 17:06 Vídeos
```

Bien, ahora ya tenemos claro el concepto de permisos. Vemos unos ejemplos comunes de uso.

Para conceder permisos de ejecución.

```
chmod u+x [archivo/carpeta]
```

Eliminar el permiso de lectura a los propietarios del grupo.

En este caso eliminamos los permisos de lectura a los propietarios del grupo.

```
chmod g-r [archivo/carpeta]
```

Eliminar el permiso de lectura al resto de usuarios.

```
chmod o-r [archivo/carpeta]
```

Todos los usuarios tienen permisos de escritura.

```
chmod a+w [archivo/carpeta]
# o
chmod +x [archivo/carpeta]
```

Establecer permisos específicos.

```
chmod u=rw [archivo/carpeta]
```

Combinar permisos.

```
chmod u=rwx, ug+rw, o=r [archivo/carpeta]
```

Este sencillo artículo es una primera toma de contacto para los usuarios más noveles, en el próximo (también indicado para los recién llegados a linux), veremos los permisos avanzados en sistemas linux.



Anarchy – Instalar Arch Linux para novatos



Instalar Arch Linux puede resultar una tarea tediosa para los usuarios menos experimentados. Sorprendentemente... pocos son los usuarios que recuerdan **Architect Linux**, un instalador gráfico en línea de comandos (ya desaparecido) que te guiaba en la instalación de **Arch Linux** de forma simple.

Afortunadamente surgieron dos grandes derivados, que actualmente cada uno sigue su camino. Por un lado tenemos **Manjaro Architect** que permite crear tu propio sistema, pero... bajo las directrices marcadas por **Manjaro**, y por el otro **Anarchy Linux**, también conocido como **Anarchy Installer**. Este último es más purista, por tanto el preferido por los profesionales.

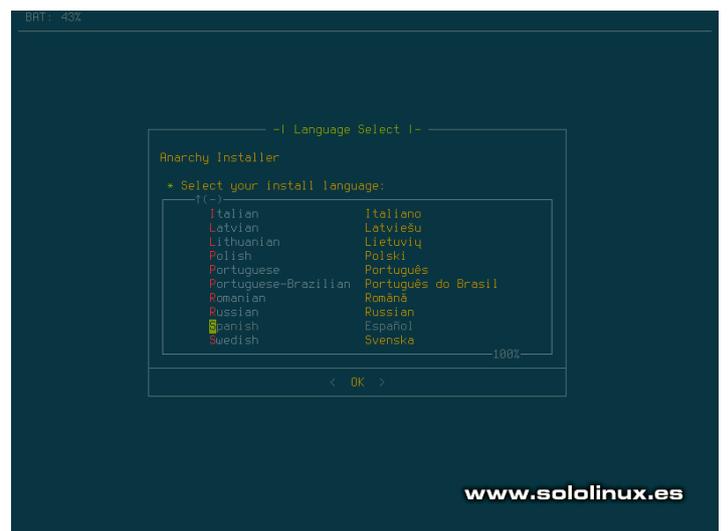
Anarchy Linux es un instalador en línea de comandos de Arch, fácil, intuitivo y sorprendentemente sencillo de usar. Su misión es centrarse en mejorar la experiencia del usuario, ofreciendo unas características con respecto a la personalización difíciles de igualar.

- Permite elegir entre múltiples entornos de escritorio y administradores de ventanas.
- El particionado del disco puede ser automático o manual.
- Elige tu propio kernel.
- Puedes seleccionar entre las shells Bash, ZSH, Fish y más,
- Funciones increíbles, como instalar paquetes de aplicaciones personalizados.
- Mucho más...



Anarchy – Instalar Arch Linux para novatos
Anarchy se aloja en **Gitlab** (anteriormente en **Gitnuh**), puedes descargar la última versión desde este [enlace](#). Copias la iso en un dispositivo USB, e inicias la instalación desde el mismo.

En las dos primeras pantallas, nos pide nuestro lenguaje y el del teclado.



En la siguiente imagen verás que se ofrecen varias opciones, actualizar, instalar o volver a la **shell**.



Recuerda que este tutorial se crea para usuarios noveles, por tanto lo haremos todo lo más fácil y rápido posible. No te compliques la vida, selecciona en descargar las mejores replicas.



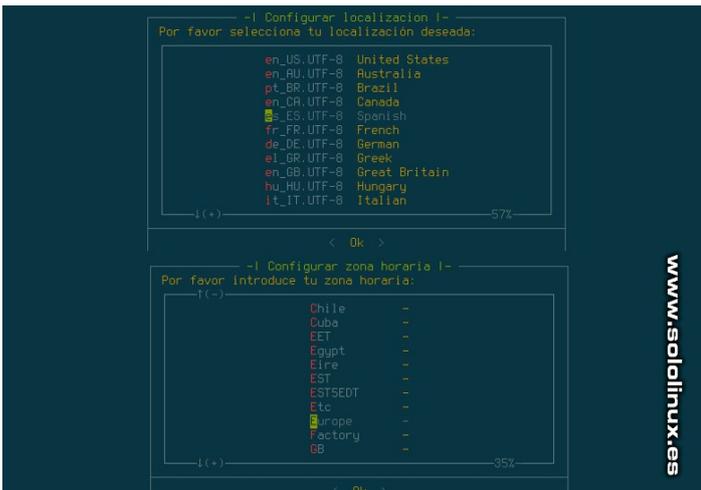
Ahora el sistema de archivos. Elegimos «ext4» (recuerda que somos novatos).



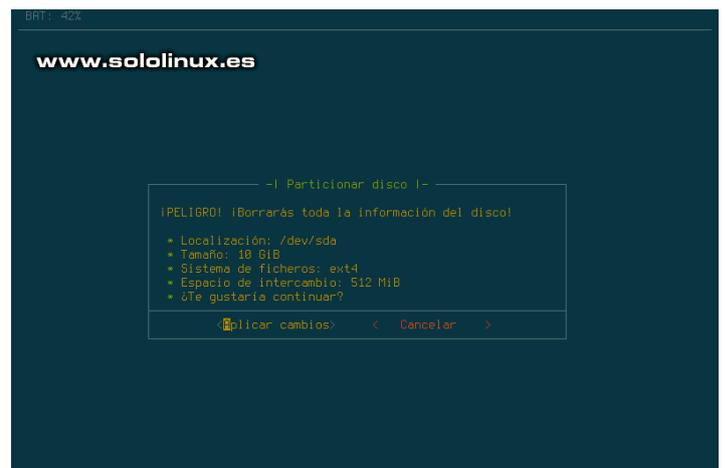
Creamos la Swap y, aplicamos todos los cambios.



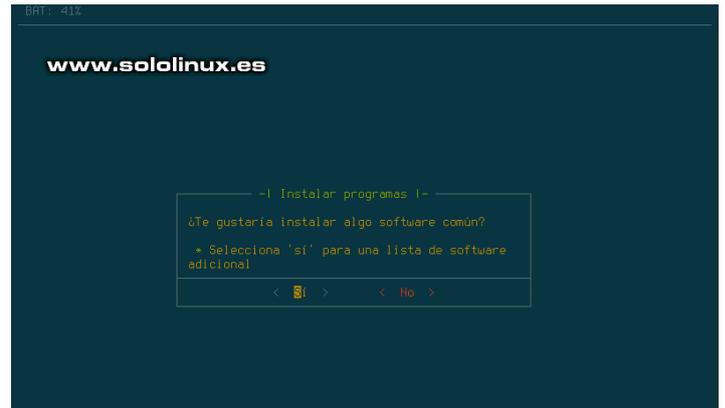
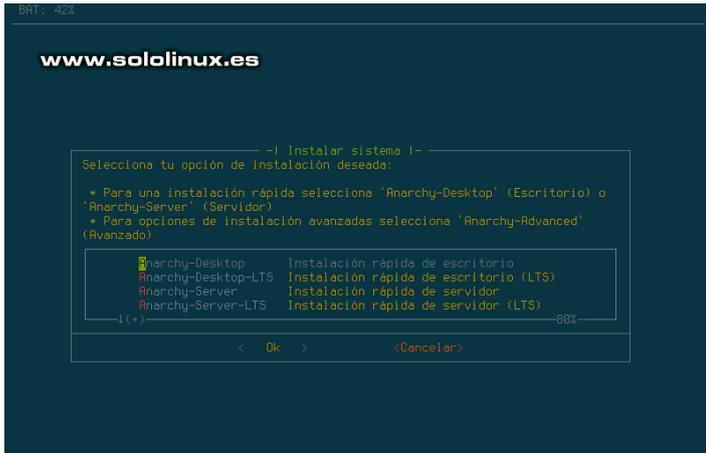
También necesitas especificar la zona horaria.



Llegó el momento de la verdad, vamos a **instalar Arch Linux**. Como somos novatos seleccionamos el particionado automático y, el disco de destino.



Nos pregunta el tipo de instalación deseada.

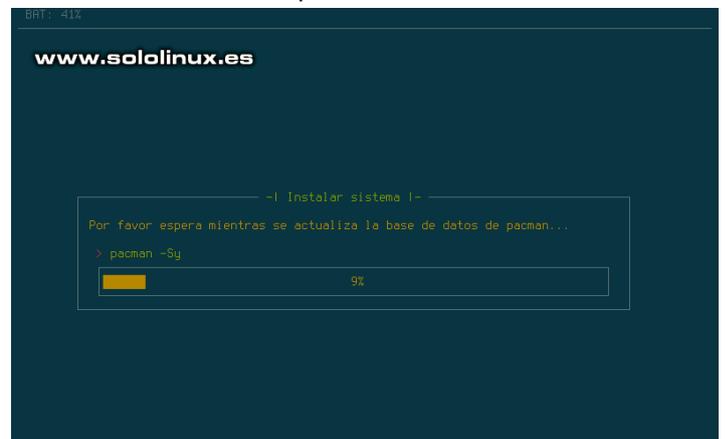


Nos solicita un nombre para el sistema.



Una vez hayas seleccionado todos los componentes de tu instalación de Arch, procedemos...

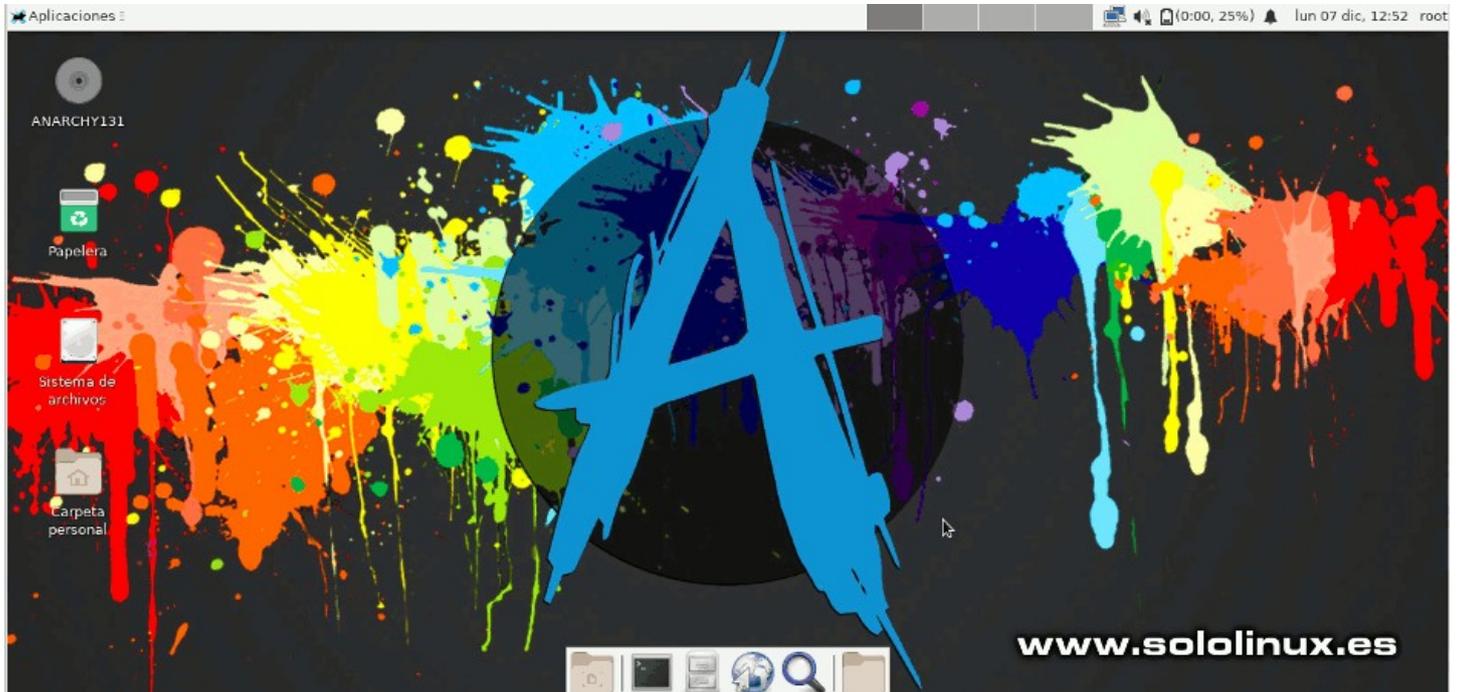
Insertamos la key del root. También podemos crear un nuevo usuario (recomendable).



En el siguiente punto te pregunta si quieres instalar algún software adicional. Mi recomendación es que no, **Arch Linux con XFCE** ya viene con todo lo necesario, incluyendo **Gimp, LibreOffice, etc.**



Pulsa en instalar el nuevo sistema. Sé paciente, pues dependiendo de la velocidad de tu sistema tardara más o menos.



Nota final: Recuerda que para moverte por los menús debes usar la tecla «TAB». Por defecto Anarchy instala ZSH como shell predeterminada, en el proceso de instalación tienes la opción recomendable (después de crear el usuario), de sustituirla por ssh.

SoloLinux

Canales de Telegram: [Canal SoloLinux](#) – [Canal SoloWordpress](#)

Espero que esta revista te sea de utilidad, puedes ayudarnos a mantener vivo este proyecto con una donación (PayPal), o también colaborar con el simple gesto de compartir nuestras revistas en tu sitio web, blog, foro o redes sociales.

[Chat de SoloLinux en Telegram](#)

CentOS se muere – Bienvenido Rocky Linux



Hace varios días que tengo en mente este artículo, pero la verdad, la tristeza que me embarga no me lo permitía. **CentOS se muere**, y no es una broma. Cuando hace seis años, **la comunidad de CentOS** paso a ser patrocinada por **RedHat**, muchos ya sabíamos como acabaría la historia, son muchos años al pie del cañón en este mundillo.

No entiendo como **CentOS** entrego su marca a **RedHat**, aún menos el futuro de la distribución, bueno... sí que lo llego a comprender, parte de sus principales desarrolladores pasaron a engrosar la enorme plantilla de la compañía. Los que llevamos tantos años con **CentOS Server** estamos de luto, fue una muerte anunciada.

Realmente, CentOS no desaparece como distribución linux, simplemente pasa a engordar la lista de distribuciones linux que prueban los últimos desarrollos de RHEL, para que me entiendas... será un nuevo **Fedora** pero con otro nombre, **CentOS Stream**. Recuerda que hace poco más de un año, mataron a uno de sus mejores derivados, Mi amado **Scientific Linux Desktop**.



Nota de CentOS Team

El futuro del Proyecto CentOS es CentOS Stream, y durante el próximo año cambiaremos el enfoque de CentOS Linux, la reconstrucción de Red Hat Enterprise Linux (RHEL), a CentOS Stream, que rastrea justo antes de una versión actual de RHEL. CentOS Linux 8, como una reconstrucción de RHEL 8, finalizará a fines de 2021. CentOS Stream continúa después de esa fecha y actúa como la rama ascendente (desarrollo) de Red Hat Enterprise Linux.

Mientras tanto, entendemos que muchos de ustedes están profundamente interesados en CentOS Linux 7, y continuaremos produciendo esa versión durante el resto del ciclo de vida de RHEL 7.

CentOS Stream también será la pieza central de un cambio importante en la colaboración entre los Grupos de Interés Especial (SIG) de CentOS. Esto asegura que los SIG se estén desarrollando y probando contra lo que se convertirá en la próxima versión de RHEL. Esto también proporciona a los SIG un objetivo único claro, en lugar de tener que crear y probar dos versiones. Le da a la comunidad de colaboradores de CentOS una gran influencia en el futuro de RHEL. Y elimina la confusión sobre lo que significa «CentOS» en el ecosistema de distribución de Linux.

Cuando finalice CentOS Linux 8 (la reconstrucción de RHEL8), su mejor opción será migrar a CentOS Stream 8, que es un pequeño delta de CentOS Linux 8, y tiene actualizaciones regulares como las versiones tradicionales de CentOS Linux. Si está utilizando CentOS Linux 8 en un entorno de producción y le preocupa que CentOS Stream no satisfaga sus necesidades, le recomendamos que se ponga en contacto con Red Hat para conocer las opciones.

*Tenemos **preguntas frecuentes** para ayudarlo con sus necesidades de información y planificación, a medida que descubre cómo este cambio de enfoque del proyecto podría afectarlo. CentOS se muere.*

CentOS se muere – Bienvenido Rocky Linux

Para los amantes de servidores basados en **rpm**, casi no quedan alternativas fiables, **Fedora** puede servir para servidores caseros o de pequeños negocios, nada más. Podría servirnos **Oracle Linux**, pero su política empresarial en otros puntos no coinciden con mis ideas.

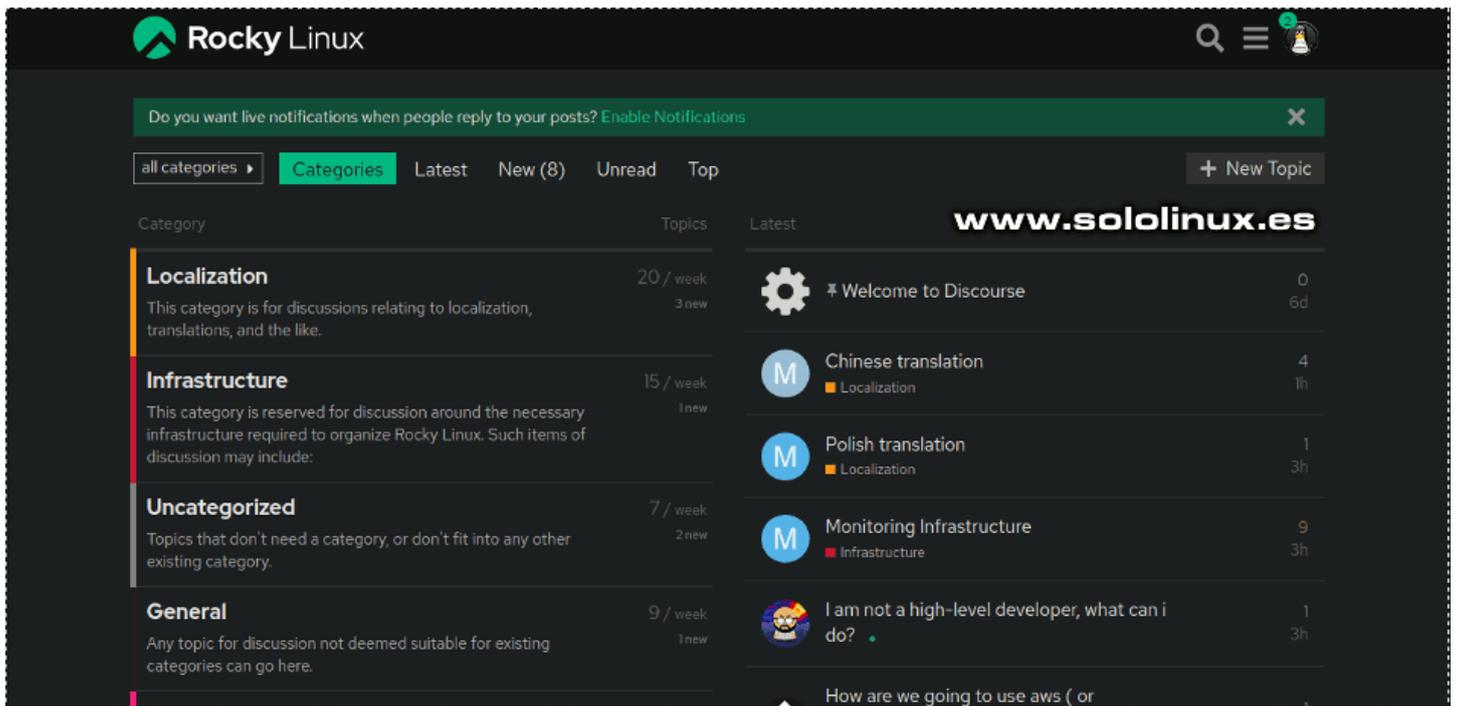
Está claro que **CentOS se muere en servidores web**; como **RedHat** no puede con su competencia gratuita, la compra (mal, muy mal). Por suerte... el fundador y principal desarrollador del proyecto CentOS «Gregory Kurtzer», ha dado un puñetazo en la mesa, y con un hasta aquí hemos llegado inicia un nuevo proyecto llamado «**Rocky Linux**». Rocky Linux tiene la intención de seguir la senda de CentOS, tal como Red Hat emite sus paquetes estables. CentOS se muere.

El anuncio de CentOS cayo como un jarro de agua fría en la comunidad, no hay tiempo de reacción. Por lo dicho, **Rocky Linux** es todavía un proyecto que solo comienza a caminar. La información disponible de Rocky Linux es poca, pero aun así... puedes revisar los siguientes enlaces propuestos, e incluso colaborar con tus ideas en el foro.

- [Página web oficial](#)
- [Código fuente @ GitHub](#)
- [Foro oficial de Rocky Linux](#)
- [Twitter](#)
- [Reddit](#)

Todas las páginas se mantienen actualizadas constantemente, pero si necesitas más información sobre el nuevo proyecto puedes revisar su wiki.

- [Wiki de Rocky Linux](#)



Otros actores entran en escena:

Al contrario que **Scientific Linux** y **Oracle Linux**, El derivado de RHEL, **CloudLinux** no es un mero espectador. **CloudLinux** es una distribución linux específica, cuya utilidad es crear servidores web (de pago) de alojamiento compartido, que introduce cambios dirigidos a controlar y optimizar el uso del sistema para tal efecto.

Los responsables de esta distribución, **anunciaron** hace unos días el desarrollo de una nueva distribución paralela, **open source** y gratuita que continúe el camino andado de CentOS. Ellos cuentan con toda la infraestructura, el software y la experiencia necesaria. Aún no sabemos nada al respecto, pero su lanzamiento se prevé para el primer trimestre del 2021. Veremos como termina todo este jaleo. Muchas gracias RedHat, en serio... gracias por todo.



Instalar el Kernel 5.10 LTS en Ubuntu o Linux Mint

Tal como anunciamos en un [artículo anterior](#), el **Kernel 5.10 LTS** estaba a punto de ser lanzado en su versión estable. Bien, ya lo tenemos entre nosotros con nuevas características y mejoras.

Se amplía el soporte de hardware incluyendo los nuevos Intel Rocket Lake y Alder Lake.

- Ahora soporta el monitoreo de la temperatura en AMD Zen 3.
- El nuevo kernel es compatible con Raspberry Pi VC4.
- La tarjeta de sonido Creative Labs SoundBlaster AE-7 también es compatible.
- Soporte para Nintendo Switch.
- Soporte para NVIDIA Orin.
- Y muchos cambios más.



The Linux Kernel Archives

[About](#) [Contact us](#) [FAQ](#) [Releases](#) [Signatures](#) [Site news](#)

www.sololinux.es

Protocol	Location	Latest Release
HTTP	https://www.kernel.org/pub/	<div style="background-color: #ffcc00; padding: 5px; border: 1px solid black;"> 5.10.1 ↓ </div>
GIT	https://git.kernel.org/	
RSYNC	rsync://rsync.kernel.org/pub/	

mainline:	5.10	2020-12-13	[tarball]	[pgp]	[patch]	[view diff]	[browse]		
stable:	5.10.1	2020-12-14	[tarball]	[pgp]	[patch]	[view diff]	[browse]	[changelog]	
stable:	5.9.14	2020-12-11	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.4.83	2020-12-11	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.19.163	2020-12-11	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]



Instalar el Kernel 5.10 LTS en Ubuntu o Linux Mint

Descargar e instalar los binarios del kernel es tarea sencilla, tan solo debes ejecutar los comandos que te propongo en la terminal de tu sistema.

Usamos el **comando wget** para descargar los archivos necesarios.

```
cd /tmp/
wget -c https://kernel.ubuntu.com/~kernel-ppa/mainline/v5.10/amd64/linux-headers-5.10.0-051000_5.10.0-051000.202012132330_all.deb
wget -c https://kernel.ubuntu.com/~kernel-ppa/mainline/v5.10/amd64/linux-headers-5.10.0-051000-generic_5.10.0-051000.202012132330_amd64.deb
wget -c https://kernel.ubuntu.com/~kernel-ppa/mainline/v5.10/amd64/linux-image-unsigned-5.10.0-051000-generic_5.10.0-051000.202012132330_amd64.deb
wget -c https://kernel.ubuntu.com/~kernel-ppa/mainline/v5.10/amd64/linux-modules-5.10.0-051000-generic_5.10.0-051000.202012132330_amd64.deb
```

Una vez los tengamos en la carpeta **/tmp**, instalamos el nuevo kernel.

```
sudo dpkg -i *.deb
```

Reinicia el sistema y selecciona el nuevo Kernel. Es así de fácil.

```
sudo reboot
```

Si te causa algún problema, puedes desinstalar el Kernel 5.10 LTS con...

```
sudo dpkg --purge linux-image-unsigned-5.10.0-051000-generic
```

Mantener los Controladores Mesa actualizados



Mesa es una biblioteca de **gráficos 3D** (código abierto), que proporciona a nuestro sistema un renderizado gráfico tridimensional. Esto nos permite una mejora considerable en la experiencia de usuario con aplicaciones comunes o, incluso con juegos en alta calidad.

El hardware gráfico de Intel y AMD, son los que mejor aprovechan la potencia de Mesa, pero también es compatible con otros fabricantes. El paquete Mesa contiene las siguientes **API**:

- **OpenGL**
- **OpenGL ES**
- **Vulkan**
- **EGL**
- **OpenMAX**
- **OpenCL**
- **VDDPAU**
- **VA API**
- **XvMC**

La mayoría de **distribuciones linux**, agregan el paquete Mesa al lanzar su última versión y, después se olvidan de él, apenas lo actualizan. Nosotros vamos a solucionar este problema.



Mantener los controladores Mesa actualizados

Según se indica en su [página oficial](#), en diciembre de 2020 la última versión disponible es la 20.3.0. Si tu hardware es muy nuevo, tal vez necesites la versión indicada. Pero primero vamos a verificar nuestra biblioteca Mesa, para ello ejecuta el siguiente **comando**.

```
glxinfo | grep Mesa
```

Obtendrás algo similar a...

```
sergio@sololinux:~$ glxinfo | grep Mesa
client glx vendor string: Mesa Project and SGI
Device: Mesa DRI Intel(R) HD Graphics 400 (BSW)
(0x22b1)
OpenGL renderer string: Mesa DRI Intel(R) HD Graphics 400 (BSW)
OpenGL core profile version string: 4.6 (Core Profile) Mesa 20.0.8
OpenGL version string: 3.0 Mesa 20.0.8
OpenGL ES profile version string: OpenGL ES 3.1 Mesa 20.0.8
```

En Ubuntu, Linux Mint y derivados, tenemos la suerte de disponer de un repositorio que siempre mantiene la última versión estable actualizada. Lo agregamos.

```
sudo add-apt-repository ppa:kisak/kisak-mesa
```

Actualizar Mesa.

```
sudo apt update
sudo apt full-upgrade
```

Reiniciamos el sistema y verificamos de nuevo la versión instalada.

```
sudo reboot
```

```
glxinfo | grep Mesa
```

En el siguiente ejemplo vemos la versión actual de Mesa, la 20.3.0 (última).

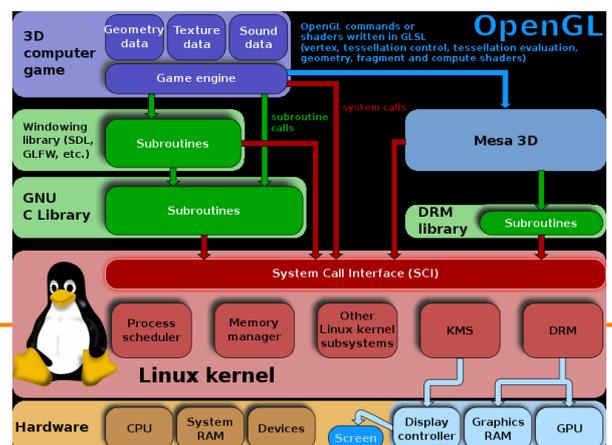
```
sergio@sololinux:~$ glxinfo | grep Mesa
client glx vendor string: Mesa Project and SGI
Device: Mesa DRI Intel(R) HD Graphics 400 (BSW)
(0x22b1)
OpenGL renderer string: Mesa DRI Intel(R) HD Graphics 400 (BSW)
OpenGL core profile version string: 4.6 (Core Profile) Mesa 20.3.0 - kisak-mesa PPA
OpenGL version string: 3.0 Mesa 20.3.0 - kisak-mesa PPA
OpenGL ES profile version string: OpenGL ES 3.1 Mesa 20.3.0 - kisak-mesa PPA
```

Si quieres retornar a la versión anterior, es tan sencillo como borrar el repositorio agregado y actualizar de nuevo.

```
sudo ppa-purge ppa:kisak/kisak-mesa
sudo apt update
sudo apt full-upgrade
```

Otra opción (solo para desarrolladores) es agregar el repositorio de desarrollo continuo de controladores Mesa. No lo recomiendo para un sistema en producción, dado que se actualiza frecuentemente y no es cien por cien estable. Si lo quieres probar... es tu decisión.

```
sudo add-apt-repository ppa:oibaf/graphics-drivers
```



Monitorizar la red Con Iptraf

```

iptraf-ng 1.1.4
Statistics for eth0

Total      Total      Incoming  Incoming  Outgoing  Outgoing
Packets   Bytes     Packets   Bytes     Packets   Bytes
-----
Total:    629      128040    310      34884     319      79996
IPv4:    600      117716    297      33714     303      79996
IPv6:     29       192       13       192       16       0
TCP:     614      111229    295      31232     319      79996
UDP:     12       7481      15       2881      16       0
ICMP:     3       192       0       0         0       0
Other IP: 0         0         0         0         0       0
Non-IP:  0         0         0         0         0       0

Total rates  71.64 kbps  Broad  11
Incoming rates  20.25 kbps  Broad  2352
Outgoing rates  53.34 kbps  Broad

IP checks errors:

www.sololinux.es
  
```

IPTraf es una utilidad de monitoreo de red en Linux, que nos muestra valiosa información sobre el tráfico circulante sobre la IP. Con esta herramienta, podemos monitorar mucha información, vemos sus principales características.

- Conexiones TCP actuales.
- Se aceptan paquetes UDP, ICMP, OSPF y otros tipos.
- Medición de paquetes y bytes en conexiones TCP, UDP, ICMP y más.
- Medición de TCP / UDP por puertos.
- Recolección de paquetes por tamaño.
- Recuento de paquetes y bytes por dirección IP.
- Estados de los paquetes TCP.
- Estadísticas generales de la red.
- Interfaz de usuario agradable (en consola).

Si especificamos todos los protocolos soportados...

- IP
- TCP
- UDP
- ICMP
- IGMP
- IGP
- IGRP
- OSPF
- ARP
- RARP

También las interfaces admitidas...

- Loopback local.
- Todas Interfaces Ethernet.
- Todas Interfaces FDDI.
- SLIP.
- PPP asíncrono.
- Síncrono PPP sobre ISDN.
- ISDN con encapsulamiento IP Raw.
- ISDN con encapsulamiento HDLC Cisco.
- IP en Línea Paralela.
- Alguno más...

Esta veterana herramienta nos puede sacar de más de un apuro a la hora de descubrir fugas en nuestra red, pruébala no te arrepentirás. En este artículo vemos como instalar la aplicación y, también algunos ejemplos de uso. Aclaramos que **Iptraf** hace años que abandono

su desarrollo, actualmente usamos su evolución **Iptraf-ng**. Monitorizar la red con Iptraf.

IPTraf

IP Network Monitoring Software

www.sololinux.es

Monitorizar la red con Iptraf-ng

Iptraf o su evolución Iptraf-ng, no viene instalado en casi ninguna distribución linux. Lo instalamos.

Instalar Iptraf en Debian, Ubuntu y derivados

```
sudo apt install iptraf-ng
```

Instalar Iptraf en CentOS, Fedora, RHEL y derivados

```
sudo dnf install iptraf-ng
# 0
sudo yum install iptraf-ng
```

Instalar Iptraf en Arch Linux, Manjaro y derivados

```
sudo pacman -S iptraf-ng
```

Instalar Iptraf en Open Suse y derivados

```
sudo zypper install iptraf-ng
```

Ejecutar Iptraf

Una vez instalada la herramienta la ejecutas con el siguiente comando.

```
sudo iptraf-ng
```

Aparece una pantalla menú, en la cual puedes buscar la información que necesites (flechas del teclado y Enter).

```

iptraf-ng 1.1.4

IP traffic monitor
General interface statistics
Detailed interface statistics
Statistical breakdowns...
LAN station monitor

Filters...
Configure...
About...
Exit

www.sololinux.es

Displays current IP traffic information
Up/Down-Move selector Enter-execute
  
```

Un ejemplo...

```

iptraf-ng 1.1.4

192.168.0.107:46126 > 213 8520 --A- wlo1
195.10.2.73:443 > 221 324630 --PA- wlo1
192.168.0.107:38578 > 24 1793 --A- wlo1
31.13.81.9:443 > 24 1600 --PA- wlo1
192.168.0.107:47380 > 1 40 --A- wlo1
104.16.86.20:443 > 1 40 --A- wlo1
192.168.0.107:47382 > 1 40 --A- wlo1
104.16.86.20:443 > 1 40 --A- wlo1
192.168.0.107:38570 > 22 1624 --A- wlo1
31.13.81.9:443 > 22 1466 --PA- wlo1
192.168.0.107:46512 > 27 2175 --A- wlo1
163.172.52.4:443 > 25 16391 --PA- wlo1
179.24.126.58:29583 = 0 0 ---- wlo1
192.168.0.107:40859 www.sololinux.es = 1 60 S--- wlo1

UDP (58 bytes) from 192.168.0.107:61233 to 83.47.208.21:40694 on wlo1
UDP (58 bytes) from 192.168.0.107:61233 to 213.94.39.80:43094 on wlo1
UDP (58 bytes) from 192.168.0.107:61233 to 84.232.116.26:45960 on wlo1
UDP (58 bytes) from 192.168.0.107:61233 to 213.194.180.139:6263 on wlo1
UDP (58 bytes) from 192.168.0.107:61233 to 62.117.202.241:2265 on wlo1
UDP (58 bytes) from 192.168.0.107:61233 to 31.4.159.35:49917 on wlo1

Packets captured: 3792 | TCP flow rate: 1.82 kbps
Up/Dn/PqUp/PqDn-scroll H-more TCP info W-chg actv win S-sort TCP X-exit
  
```

Navega por el menú para descubrir todas sus posibilidades. También puedes modificar las opciones de la herramienta, incluyendo su visualización.

iptraf-ng 1.1.4

Reverse DNS Lookups	Current Settings
Reverse DNS Lookups	Reverse DNS lookups: Off
TCP/UDP service names	Service names: Off
Force promiscuous mode	Promiscuous: Off
Color	Color: On
Logging	Logging: Off
Activity mode	Activity mode: kbits/s
Source MAC addr's in traffic monitor	MAC addresses: Off
Show v6-in-v4 traffic as IPv6	v6-in-v4 as IPv6: On
Timers...	TCP timeout: 15 mins
Additional ports...	Log interval: 60 mins
Delete port/range...	Update interval: 0 secs
Ethernet/PLIP host descriptions...	Closed/idle persist: 0 mins
FDDI/Token Ring host descriptions...	
Exit configuration	

www.sololinux.es

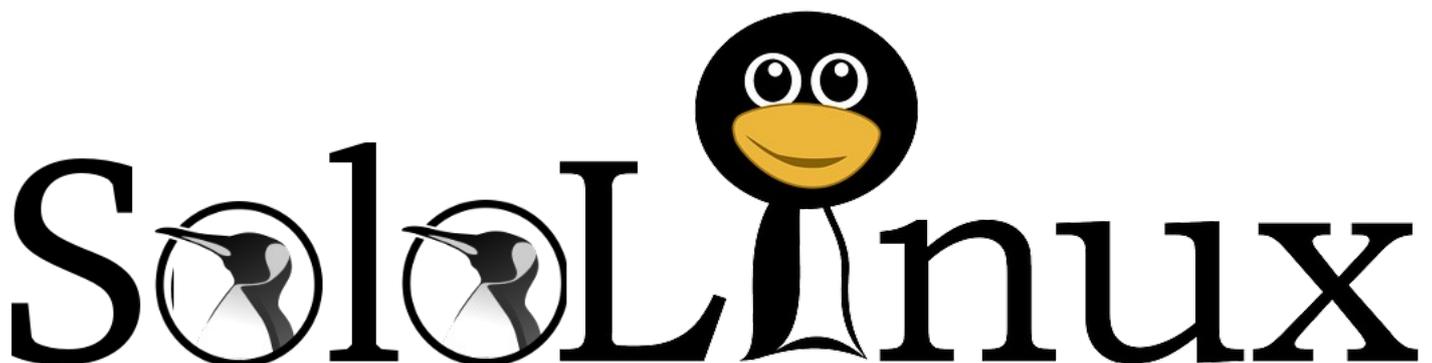
Toggles resolution of IP addresses into host names

Up/Down-Move selector Enter-execute

Puedes **monitorizar la red** sobre una medición específica, lanzando **Iptraf-ng**, desde la línea de comandos. Son pocas opciones fáciles de recordar, de todas formas... te recomiendo que revises su ayuda con «**help**».

```
sudo iptraf-ng -h
```

```
root@sololinux:~# iptraf-ng -h
usage: iptraf-ng [options]
or: iptraf-ng [options] -B [-i <iface> | -d <iface> | -s <iface> | -z <iface> | -l <iface> | -g]
-h, --help                show this help message
-i <iface>                start the IP traffic monitor (use '-i all' for all interfaces)
-d <iface>                start the detailed statistics facility on an interface
-s <iface>                start the TCP and UDP monitor on an interface
-z <iface>                shows the packet size counts on an interface
-l <iface>                start the LAN station monitor (use '-l all' for all LAN interfaces)
-g                        start the general interface statistics
-B                        run in background (use only with one of the above parameters)
-f                        clear all locks and counters
-t <n>                    run only for the specified <n> number of minutes
-L <logfile>              specifies an alternate log file
```



Instalar un servidor Debian 10 con Yunohost



YunoHost es un sistema operativo **basado en Debian**, que tiene como objetivo simplificar lo máximo posible la configuración de un servidor para un único usuario, o grupo de colaboradores.

YunoHost tiene unas excelentes características, de forma predeterminada instala y configura un **servidor web nginx, MariaDB**, el agente de correo Postfix y el servidor IMAP Dovecot, incluyendo el filtro spam, Rspamd. También se incluye el servidor Metronome IM XMPP, OpenLDAP, Dnsmasq y, un sistema de autenticación transparente SSOwat para el inicio de sesión. Como es lógico no falta el generador de certificados SSL, **Let's Encrypt**.

Instalar un servidor Debian 10 con YunoHost ofrece otras ventajas, como por ejemplo un repositorio comunitario de scripts validados, que nos permiten instalar otros servicios y aplicaciones (WordPress, Drupal, etc). El repositorio proporciona versiones personalizadas de aplicaciones ya preconfiguradas, y preparadas para integrarse en el entorno de inicio de sesión único de **YunoHost**.



Instalar un servidor Debian 10 con Yunohost

El único requisito para instalar Yunohost, es tener un VPS o servidor con Debian 10 instalado. Iniciamos la instalación con el siguiente comando.

```
sudo curl https://install.yunohost.org | bash
```

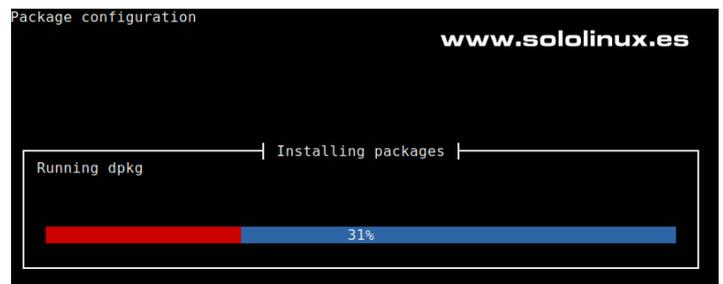
Verás algo similar a la imagen, responde «**YES**».



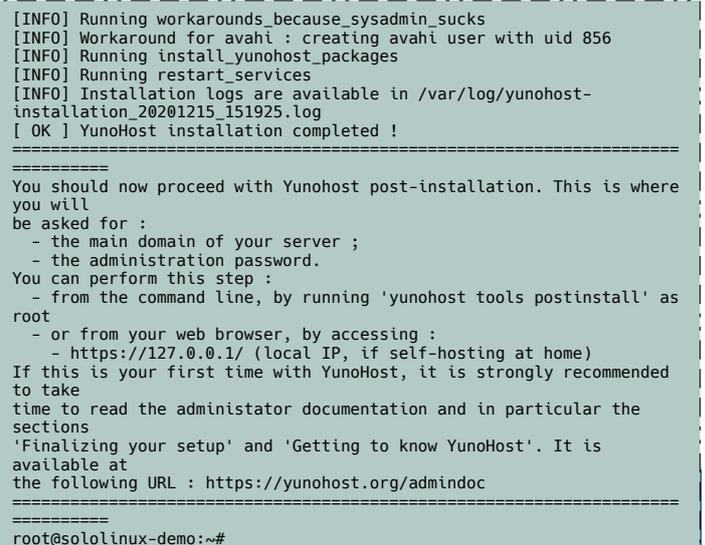
También te pregunta si quieres que **Yunohost** maneje el acceso root a ssh, si aceptas no podrás acceder desde root por ssh al servidor. Es tu decisión.



Comienza la instalación.



Al concluir el proceso de instalación (es rápido), veremos lo siguiente.



Para acceder a **Yunohost** escribes la **ip del servidor**, en tu navegador preferido. Antes de **acceder a Yunohost** debemos configurar algunas cosas como el dominio (no obligatorio), el usuario, la contraseña, etc.



Notas finales a tener en cuenta

- Si quieres **instalar Yunohost** en tu propio servidor local sin VPN, asegúrate de redirigir correctamente los puertos en tu enrutador.
- Si utilizas tu propio nombre de dominio debes configurarlo correctamente, y sin ningún error.
- Recuerda que hasta que obtengas el certificado Let's Encrypt, pueden aparecer advertencias de peligro.
- Yunohost es un servidor con propósito de usuario único, no des acceso a invitados a no ser que sean de absoluta confianza.
- El listado de aplicaciones y herramientas es enorme, revísalo bien, seguro que localizas lo que necesitas.

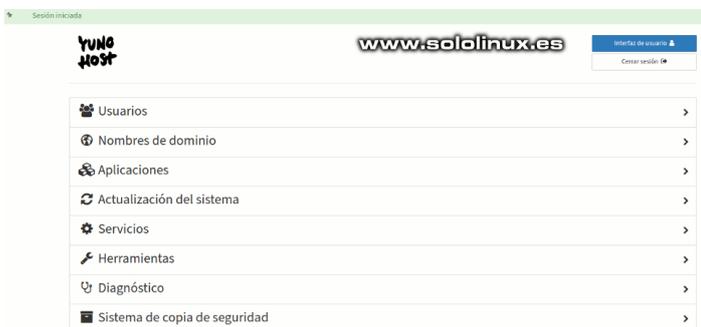


Dominio

Este es el primer nombre de dominio vinculado a su servidor YunoHost, pero también el que será utilizado por los usuarios para acceder al portal de identificación. Será visible por todos, elegir con cuidado.



Una vez tengamos todo configurado, accedemos al menú general del servidor Debian con **Yunohost**.



Como puedes observar en la imagen anterior, disponemos de un menú sencillo y fácil de usar. Como último ejemplo del artículo «**Instalar un servidor Debian 10 con Yunohost**», accedemos al menú de aplicaciones donde encontraras las categorías principales, navega por ellas, te sorprenderá.



Uso del Comando route en linux



Para cualquier **sysadmin**, es importante comprender el escalado de red de los sistemas operativos Linux, así el cómo acceder a información relacionada con la misma, de forma que podamos corregir posibles errores. Es evidente que este artículo es más apropiado para un administrador de redes, aun siendo así, los administradores de **sistemas linux** también deben tener nociones importantes al respecto.

En **sololinux** hemos escrito artículos sobre el uso de los comandos **ifconfig** e **ip**, que nos permiten agregar, eliminar o actualizar direcciones IP en un servidor Linux. El enrutamiento es un término que parece similar, pero no lo es. En términos simples, el enrutamiento es un método de enviar un paquete IP de un punto a otro, cuyo contenido puede ser cualquier información transmitida a través de la red.

En los sistemas operativos Linux, la información sobre cómo se reenviarán los paquetes se almacena estructuralmente en la tabla de enrutamiento en el **kernel**, por tanto es posible que a veces necesitemos usar el **comando route**. Soy consciente que dicho comando está en desuso, pues poco a poco es sustituido por el «**comando ip route**», aun siendo así, sigue siendo de gran utilidad y continúa funcionando en cualquier **distribución linux** como el primer día.

```
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default 62.210.131-1.re 0.0.0.0 UG 100 0 0 eth0
62.210.131.0 0.0.0.0 255.255.255.0 U 100 0 0 eth0
```

Uso del comando route en linux

El **comando route** se usa para manipular las tablas de enrutamiento del kernel. Su uso principal es configurar rutas estáticas sobre redes u hosts específicos. Debemos indicar que las rutas pueden ser estáticas o dinámicas.

Las rutas dinámicas se agregan en los enrutadores específicos, mientras que con respecto al sistema operativo... solo utilizamos las rutas estáticas. En este artículo vemos algunos ejemplos sencillos de cómo usar el **comando route** para mostrar y manipular la tabla de enrutamiento en Linux.

Por ejemplo... para ver todas las rutas existentes (con conexión), ejecutamos el siguiente comando.

```
route
```

Veremos algo similar a...

```
sergio@sololinux:~$ route
Tabla de rutas IP del núcleo
Destino Pasarela Genmask Indic
Métric Ref Uso Interfaz
default _gateway 0.0.0.0 UG 600
0 0 wlo1
link-local 0.0.0.0 255.255.0.0 U 1000
0 0 wlo1
192.168.0.0 0.0.0.0 255.255.255.0 U 600
0 0 wlo1
```

Como puedes ver en el anterior ejemplo, la ruta de enlace se identifica perfectamente. En el caso de que la quisiéramos modificar manualmente (por ejemplo 192.168.100.25), lanzaríamos este comando.

```
route add default gw 192.168.100.25
```

Verificamos que ha sido modificada correctamente.

```
sergio@sololinux:~$ route -n
Tabla de rutas IP del núcleo
Destino Pasarela Genmask Indic
Métric Ref Uso Interfaz
0.0.0.0 192.168.0.1 0.0.0.0 UG 600
0 0 wlo1
192.168.100.26 0.0.0.0 255.255.0.0 U 1000
0 0 wlo1
192.168.100.25 0.0.0.0 255.255.255.0 U 600
0 0 wlo1
```



También podemos agregar la ruta de la red y la ruta del host, de forma manual.

```
route add -net 192.168.100.25 netmask 255.255.255.0 gw 192.168.100.26
```

En el caso de que quieras eliminar lo agregado anteriormente...

```
route del -net 192.168.100.25 netmask 255.255.255.0 gw 192.168.100.26
```

Como último ejemplo, debes saber que el **comando route** nos permite rechazar cualquier acceso a una conexión (u host) en particular.

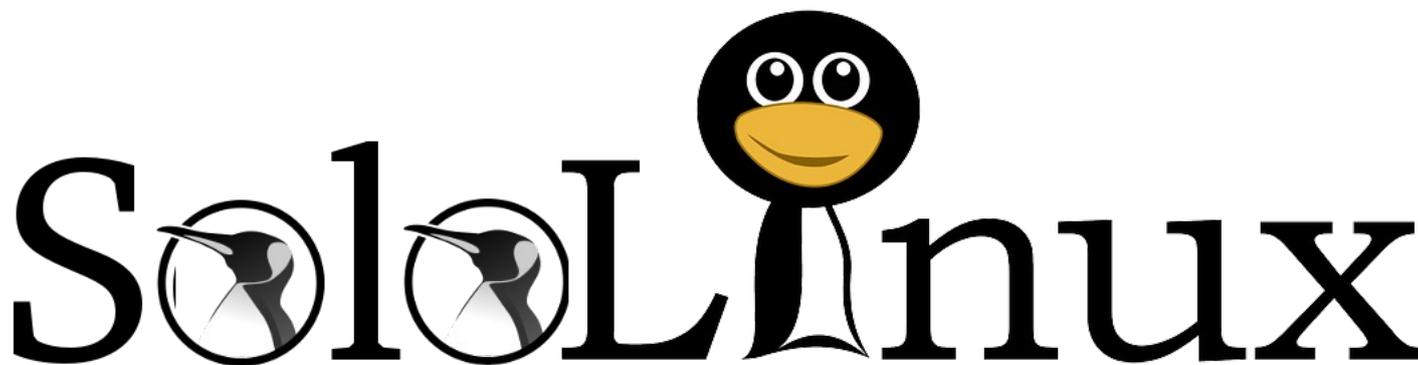
```
route add -host 193.168.250.26 reject
```

Para ver la ayuda del comando.

```
route -h
```

Salida.

```
sergio@sololinux:~$ route -h
Uso: route [-nNvee] [-FC] [<AF>]
      route [-v] [-FC] {add|del|flush} ...  Modifica la tabla de ruteado para AF
      route {-h|--help} [<AF>]             Sintaxis detallada de uso para el AF indicado.
      route {-V|--version}                 Muestra la/el versión/autor y sale.
-v, --verbose descripción amplia
-n, --numeric no se resolverán nombres
-e, --extend muestra otra/más información
-F, --fib muestra la base de información hacia adelante (predeterminado)
-C, --cache muestra la caché de ruteado en vez de la FIB
<AF>=Use -4, -6, '-A <af>' or '--<af>'; default: inet
Lista de posibles familias de direcciones (que soportan el ruteado):
inet (DARPA Internet) inet6 (IPv6) ax25 (AMPR AX.25)
netrom (AMPR NET/ROM) ipx (Novell IPX) ddp (Appletalk DDP)
x25 (CCITT X.25)
```



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



Canales de Telegram: [Canal SoloLinux](#) – [Canal SoloWordpress](#)

Espero que esta revista te sea de utilidad, puedes ayudarnos a mantener vivo este proyecto con una donación (PayPal), o también colaborar con el simple gesto de compartir nuestras revistas en tu sitio web, blog, foro o redes sociales.

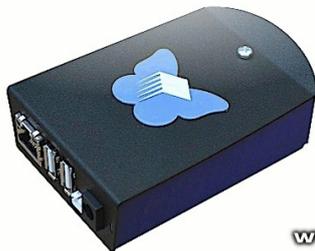
[Chat de SoloLinux en Telegram](#)

Montar un servidor Casero Con FreedomBox



FreedomBox es un proyecto de **código abierto** basado en **Debian 10**, que permite a cualquier usuario ajeno al mundo de los servidores, configurar su propio servidor casero (doméstico) y para alojar su blog, wiki, etc.

Se ofrece a la venta un pequeño **hardware FreedomBox** que puedes usar como almacenamiento de datos, servidor VPN, servidor proxy, correo y mucho más. Realmente no es necesario, pues si tienes un viejo ordenador que tenga la capacidad de mover Debian 10 o Debian 9, también se ofrece el software de forma totalmente gratuita.



Montar un servidor casero con FreedomBox

El **software FreedomBox** instala un completo servidor basado en Apache, en el cual no echaras nada de menos. Se distribuye a través de los repositorios oficiales de Debian, por tanto también lo encontrarás en **Ubuntu**, **Linux Mint** y derivados (se recomienda instalar sobre Debian 10).

Su instalación es sencilla, aunque si requiere descargar muchos paquetes. Vemos como instalar FreedomBox en Debian 10.

```
sudo apt update
sudo apt install freedombox-setup
```

Comienza la instalación...

```
sergio@sololinux:~$ apt install freedombox-setup
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  apache2 apache2-bin apache2-data apache2-utils augeas-
  lenses augeas-tools
  batctl bridge-utils certbot devio ebttables fail2ban
  firewallld flite
  fonts-lato gir1.2-nm-1.0 haveged iftop ipset javascript-
  common ldap-utils
  ldapscripts libapache2-mod-auth-pubtk libapache2-mod-
  gnutls
  libapache2-mod-php libapache2-mod-php7.2 libapr1
  libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libaugeas0
  libcurl3-nss libhavege1
  libipset3 libjs-bootstrap libjs-jquery libjs-modernizr
  libmsv1
  libnl-route-3-200 libnss-gw-name libnss-ldapd libodbc1
  libpam-abi
  libpam-ldapd libpam-tmpdir locales-all ncurses-term nscd
  nslcd nslcd-utils
  openssh-server openssh-sftp-server php-common php7.2-cli
  php7.2-common
  php7.2-json php7.2-opcache php7.2-readline plinth pppoe
  pwen
  python-django-common python-pyicu python3-acme python3-
  asn1crypto
  python3-augeas python3-bootstrapform python3-certbot
  python3-cffi-backend
  python3-cherrypy3 python3-configargparse python3-
  cryptography
  python3-decorator python3-django python3-django-appconf
  python3-django-axes
  python3-django-captcha python3-django-ipware python3-
  django-ranged-response
  python3-django-stronghold python3-future python3-josepy
  python3-mock
  python3-openssl python3-parsedatetime python3-pbr
  python3-pyinotify
  python3-repoze.lru python3-requests-toolbelt python3-
  rfc3339 python3-routes
  .....
  .....
```

Una vez instalada la aplicación, identificamos la **ip de acceso** con el comando...

```
ip addr
```

Vemos un ejemplo.

```
sololinux@sololinux-demo:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:26:c3:d6 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic
noreferrer enp0s3
        valid_lft 86124sec preferred_lft 86124sec
    inet6 fe80::61fb:f95a:c968:97f8/64 scope link
noreferrer
        valid_lft forever preferred_lft forever
```

Normalmente debes acceder a la ip 10.0.2.0

Nos aparece una pantalla en la cual nos indica que debe concluir la instalación.

Por favor espere a que FreedomBox termine la instalación. Podrá empezar a usar su FreedomBox una vez finalice.

www.sololinux.es



Una vez inicies sesión en tu nuevo servidor local, te recomiendo que te muevas por él. Podrás comprobar lo ágil y poderoso que es.



Te recomiendo que configures el sistema de tu nuevo servidor casero.



Como último punto, te recomiendo que antes de poner tu servidor en producción, abras la pestaña **configuración del sistema** y realices las modificaciones necesarias.



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 **SOLO LINUX** MULTIPLICARAS TUS CLIENTES

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



Canales de Telegram: **Canal SoloLinux**
- **Canal SoloWordPress**

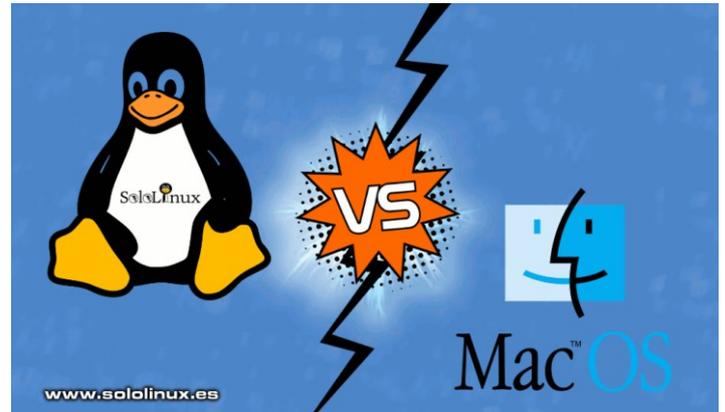
Espero que esta revista te sea de utilidad, puedes ayudarnos a mantener vivo este proyecto con una donación (**PayPal**), o también colaborar con el simple gesto de compartir nuestras revistas en tu sitio web, blog, foro o redes sociales.

Chat de SoloLinux en Telegram

Linux vs MacOS – Sus principales diferencias

La discusión sobre quien es mejor, entre **Linux** y **MacOS** se alarga en el tiempo. Existe la falsa creencia de que macOS es para los ricos y Linux es para expertos. Antes de comenzar este análisis, debemos aclarar que ningún sistema operativo es perfecto, pero es cierto que existen usuarios y máquinas perfectas para cada sistema operativo.

La virtual superioridad existente de **MacOS** sobre Linux, tiene un coste, además considero que desproporcionado por ofrecer «un nada», es evidente que es un tema de marketing. Este artículo no pretende comenzar una guerra, y mucho menos una cruzada. Tan solo intentaremos arrojar algo de luz sobre este asunto, el porqué y el cómo quien es quien.



Linux vs MacOS – Sus principales diferencias

Insisto que este artículo no es una pelea entre Linux y MacOS, simplemente vemos lo que cada sistema operativo nos ofrece, en comparación con el otro. Es evidente que cada uno tiene sus preferencias y, este efecto pueden favorecer a un sistema sobre el otro. Nosotros nos centramos en la flexibilidad, estabilidad y rendimiento que nos ofrece cada uno.

Ya es hora de romper tabús – Linux vs MacOS

Principales diferencias entre Linux y MacOS

Tal vez la definición de **sistema operativo** no define realmente a Linux, simplemente es costumbre. Realmente al hablar de un S.O. linux, deberíamos tener en cuenta que es una combinación de hardware, software y protocolos para una correcta comunicación operacional.

La definición real de Linux, es un núcleo básico preempaquetado. Este núcleo lo usan muchas distribuciones Linux como Ubuntu, Debian, Fedora, RHEL, Arch Linux, que integran con mejor o peor acierto su software, controladores, entornos de escritorio y más.

Por lo dicho, Linux es la conjunción de un núcleo básico preempaquetado, con una distribución de Linux específica. El núcleo de la base linux es el kernel, y destacamos que son incontables los voluntarios que desarrollan o crean distribuciones Linux para la comunidad.

En MacOS cambia todo, esta responsabilidad es exclusiva de los ingenieros de Apple. Todo el software, controladores, entornos de escritorio y núcleos se desarrollan bajo un mismo techo. Linux trata de involucrar a la comunidad en el desarrollo de sus distribuciones, MacOS... **no**.

Estabilidad y rendimiento – Linux vs MacOS



En este apartado, ambos sistemas operativos son vencedores. Los dos se comportan de forma estable con tareas normales. Debo indicar, que **MacOS** tiene integrado protocolos de actualización del sistema en segundo plano, y esto puede ser beneficioso o negativo, depende del hardware y del usuario.

Como norma general, la estabilidad y el rendimiento en Linux dependen de las distribuciones y de la máquina donde lo instalas. Recuerda que las **distribuciones linux**, vienen con diferentes entornos de escritorio que dependen enormemente del propósito final necesario por el usuario.

Cada distribución opera de una forma diferente, sobre todo dependiendo de su hardware; por tanto, la máquina es importante. Una distribución ligera favorecerá la experiencia de usuario, en un **hardware** obsoleto. Sin embargo, **MacOS** suele ser correlativo con sus máquinas, por tanto si quieres un buen rendimiento... solo debes usar el hardware de alta gama indicado por **Apple**.

Independientemente de las muchas pruebas que realicemos sobre su rendimiento, está claro que Linux tiene la capacidad de adaptarse tanto a máquinas de alto rendimiento como de bajo rendimiento.

Soporte de hardware y de uso

Esta sección puede ser la más polémica, pues Mac se basa exclusivamente en hardware fabricado por Apple y para **Apple**. Podemos pensar que el objetivo de esta decisión es una integración hardware-software estable, sin embargo también la podemos acusar de prácticas monopolistas en contra de las leyes de competencia legal. Los usuarios de MacOS afirman que gracias a este efecto, consiguen un buen rendimiento software-hardware (aunque realmente con un coste fuera de mercado). Sus tarifas de reparación, también son abusivas.

La **comunidad Linux** tuvo que luchar contra viento y marea, para que los fabricantes de hardware les ofrecieran su código. Por suerte, hoy en día los proveedores de hardware han visto el potencial de linux y, ya nadie los puede parar.

Tipo de usuarios

Umm... **Linux vs MacOS** y sus usuarios, suponen una gran controversia. Si tienes un Mac, tienes una joya, eres único. Sin embargo, hay una diferencia entre que te guste un sistema operativo y un sistema operativo que te guste.

Mac OS ofrece un entorno de escritorio limpio y poderoso, que funciona bien y rápido. Si te gustan los gráficos geniales o trabajas con diseños gráficos, sonido o edición de video, MacOS es muy bueno independientemente de tus conocimientos.

Linux también da la bienvenida tanto a los usuarios principiantes como a los profesionales. Pero todo depende de la primera distribución que usen, o lo aman o lo odian. Es importante conocer las distribuciones que favorecen a los **usuarios novatos**, así como las más indicadas para profesionales, son un mundo diferente.

Recuerda que al derivar de Unix, en MacOS también puedes operar en terminal, pero su desconocimiento es grande en este sentido. Los usuarios de Linux están más acostumbrados a manejar comandos. Linux vs MacOS.

Coste de Linux vs MacOS

Si hablamos de Linux, suele ser libre en lo que respecta al software, realmente en linux casi todo es gratuito y de código abierto, sin precios o cláusulas ocultas. Ten en cuenta que hablamos de herramientas y aplicaciones de alto rendimiento, algunas incluso mejoran otras aplicaciones de pago.

Si hablamos de **MacOS**, la cosa es diferente. Tanto el software como el hardware tienen sus precios establecidos, insisto en que abusivos y más si hablamos que su código fuente lo tomaron de **UNIX**. Solo tú, podrás valorar si el dinero pagado por una máquina de esas características vale la pena, o no.



Seguridad y privacidad

En Linux la privacidad y la seguridad están aseguradas en su base. Es algo común que cualquier profesional de TI, presuma de su uso (aunque no sea cierto), es por su reputación.

Si tú quieres, Linux tiene la capacidad de no dejar ningún rastro de huellas digitales. En términos de seguridad, es casi imposible crear una puerta trasera y obtener acceso a un sistema Linux seguro. OJO!!!, el potencial de una brecha en linux depende de la propia configuración del sistema, por eso es mejor no tocar lo que no conoces.

En MacOS.... uff, la privacidad y seguridad parece que es un jardín floreado en plena primavera rodeado de muros gigantes, o eso es lo que nos dicen.

Tal vez no sepas que Apple, rastrea cualquier movimiento de sus usuarios. Nos dicen que con el único objetivo de mejorar la experiencia de los clientes, para favorecer la evolución de nuevos productos o mejorar los existentes. Como su software no es de **código abierto**, es imposible acceder al sistema operativo y entender que hacer realmente con los datos de los usuarios. Mal, muy mal. Linux vs MacOS.

Personalización del usuario

Apple desarrolla cada versión de **MacOS** de una forma muy definida. El software tiene un estado fijo, lo que equivale a que el usuario no puede modificar el diseño del sistema y ajustarlo a sus preferencias. Es triste que no puedas calibrar un sistema operativo de tal categoría, para que funcione como tú quieras.

Al hablar de Linux en términos de personalización, jaja, esto es lo mejor. Solo tienes que querer, y tus deseos serán cumplidos. Puedes personalizar desde la GUI hasta la **terminal**, es increíble.



Plataforma de programación

Si te dedicas a la programación profesional en codificación, no busques más. Linux es la solución a todos tus quebraderos de cabeza.

Cientos de editores de código, todos los lenguajes de programación que existen en la actualidad tienen una asociación directa con el entorno Linux. Todo funciona a las mil maravillas en un entorno Linux.

Plataforma de juegos

Mac OS supera a Linux como plataforma para desarrollar y probar juegos de alta gama. Aunque linux hace un gran esfuerzo al respecto, en estos momentos el desarrollo de Apple es superior.



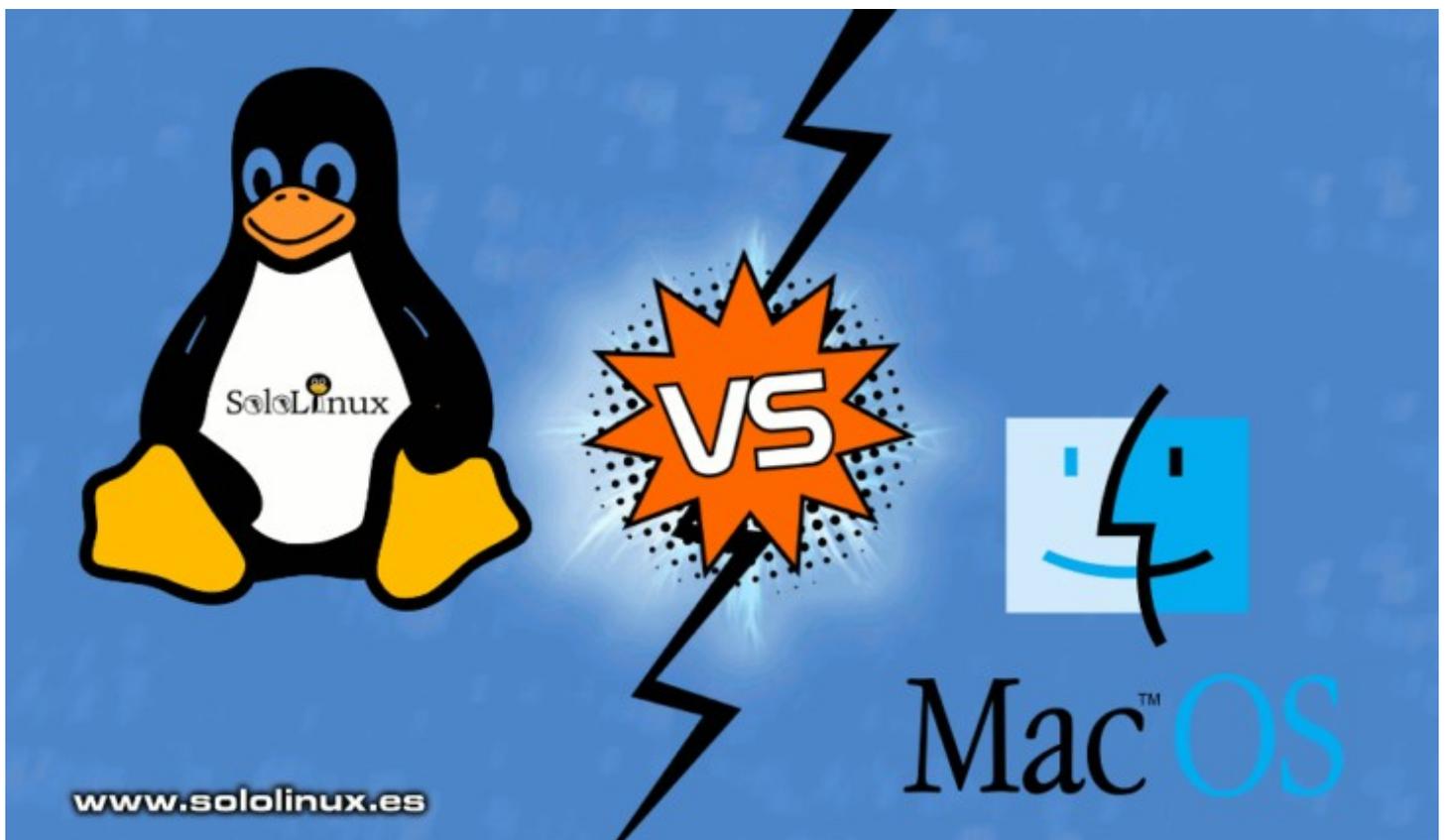
Tareas automatizadas

Linux es de código abierto, sus usuarios tienen el privilegio de interactuar con el sistema hasta su núcleo como les plazca. Es evidente que se contempla la automatización de tareas y mucho más. MacOS no otorga a sus usuarios estos privilegios de acceso por defecto.

Es lógico, que Linux sea el sistema operativo preferido por los programadores y profesionales de TI. El terminal o consola de Linux, ofrecen la puerta de enlace perfecta para automatizar tareas y mucho más.

Nota del autor

Con este artículo no pretendo generar polémica, tan solo ayudar al usuario final a decidirse si opta por Linux o por MacOS. Está claro que nuestra preferencia es linux, pero tal vez tus circunstancias sean diferentes. Analiza en términos de flexibilidad, funcionalidad, seguridad, privacidad y, sobre todo del uso que le vayas a dar. **Linux vs MacOS.**



SoloLinux

Entrevista a un Sysadmin Linux en 17 preguntas

La **industria del TI** cada día depende más de Linux. Por tanto para acceder a un puesto de trabajo relacionado, debes tener altos conocimientos sobre este sistema. En la actualidad, estos conocimientos son muy valorados por las empresas contratantes.

El **código abierto** está en la cima cuando hablamos de entornos de desarrollo, entre pequeñas y medianas empresas y es evidente que su máximo exponente es linux. La demanda de administradores de sistemas cada día es más alta, ya sea para el manejo de servidores o redes, como para tecnologías en la nube tan de moda actualmente.



En este artículo tomamos como base un ejemplo de entrevista real, que fue realizada por una pequeña empresa que buscaba un **sysadmin** (administrador de sistemas), para el manejo de sus servidores. Estos contenían el trabajo diario de la empresa, así como varios sitios web y tiendas virtuales.



Entrevista a un Sysadmin Linux

Comenzamos la entrevista.

Cómo defines linux – dime algo básico

Linux es un sistema operativo de código abierto basado en Unix, creado por **Linux Torvalds**. Su uso es gratuito y se basa en el kernel de Linux, debo indicar que está dirigido principalmente a sistemas donde el usuario no quiere excesivos costes y, que se le permita modificar el sistema a su antojo. Estas circunstancias no se dan ni en **Windows**, ni en **MacOS**.

Las distribuciones más conocidas personalizan el kernel, otras no lo compilan y ofrecen al usuario una experiencia única. No nos podemos olvidar, que Linux puede ejecutarse en hardware común independientemente de su fabricante (HP, Intel, SPARC, IBM, etc).

Para qué sirve el kernel linux

El **kernel Linux** es el núcleo del sistema operativo. El kernel maneja la interacción con el hardware y los comandos del usuario. Los usuarios también pueden interactuar directamente con el kernel desde la terminal.

Como el kernel de Linux es de código abierto, cualquier usuario puede descargarlo y editarlo sin permiso. Linux se lanza bajo la Licencia Pública General (GPL), esta quiere decir que se puede modificar y editar según las necesidades o requisitos del usuario.

Principales diferencias entre Linux y Unix

Una de las principales diferencias entre **Linux y Unix**, es el costo asociado. Linux es de código abierto y gratuito, mientras que Unix tiene un coste. La **GUI** para Linux incluye varios entornos de escritorio que puedes elegir, Unix solo ofrece un entorno común. Entrevista a un Sysadmin Linux.

Otra gran diferencia entre ellos es a quien va dirigido, Linux es más fácil de usar que Unix. También debo decir que las actualizaciones en Linux son constantes, en Unix, no.

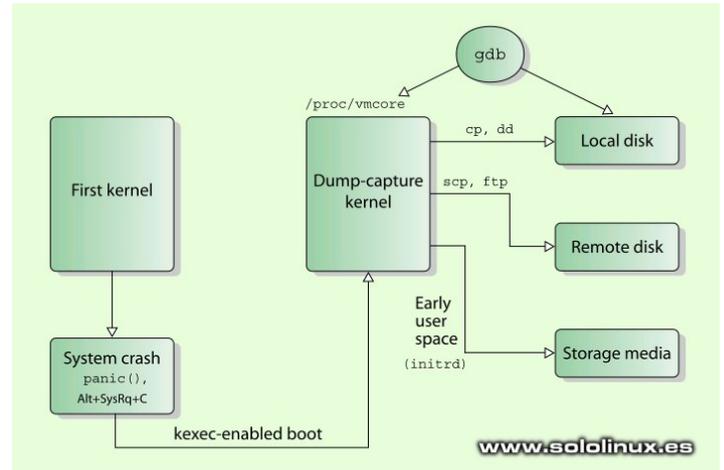
Qué es Kdump

Kdump es un mecanismo del Kernel Linux, que captura los volcados del mismo en caso de errores o caída del sistema. Los volcados también se generan con un **Kernel panic**. Es muy útil para detectar los errores del sistema y, analizar posteriormente la causa. Un administrador Linux puede configurar donde y como guardar estos volcados.

Qué es LILO

LILO o mejor dicho... Linux Loader. Es un cargador de arranque para Linux, que permite al sistema operativo cargarse en la memoria e iniciar su ejecución.

También se asegura de que la BIOS o el UEFI inicien, para verificar todo el sistema y asegurarse de que todo es correcto. Una vez que la BIOS o el UEFI comprueban todo, transfieren el control al registro de arranque maestro. En este momento LILO toma el control y, carga el sistema operativo Linux para su uso. El proceso es muy rápido.



Componentes básicos de un sistema linux

Los componentes básicos de un **sistema operativo Linux**, son cinco. El kernel, la shell, la GUI, las utilidades del sistema y las aplicaciones.

- **Kernel:** Es el núcleo del sistema Linux, encargado de gestionar la conexión entre el hardware y los usuarios.
- **GUI:** La interfaz gráfica de usuario (GUI), facilita al usuario la interacción con el sistema (de forma sencilla).
- **Shell:** Shell ejecuta comandos en Linux como intérprete.
- **Utilidades del sistema:** Las utilidades del sistema tienen la capacidad de manejar las funcionalidades de la máquina.
- **Aplicaciones:** Son los programas y herramientas diseñados para tareas específicas.

Qué es Automounter

Automounter es un servicio de Linux, encargado de manejar el montaje del sistema y sus archivos, ya sean locales o remotos. Este servicio monta automáticamente las imágenes necesarias, cuando el sistema las solicita.

Esto quiere decir que el sistema no necesita seguir montando unidades continuamente. El servicio de montaje automático **autofs**, solo monta el sistema de archivos cuando es necesario. Entrevista a un sysadmin.

Qué es, y como habilitar las políticas de contraseña

Las políticas de contraseña garantizan que estas sean seguras y, que los usuarios sigan las directrices marcadas a la hora de crear contraseñas más o menos largas y difíciles de descifrar. Para habilitar la política de contraseñas en Linux, usamos PAM (Módulo de autenticación conectable).

Hablamos de la Swap

Swap, también conocido como espacio de intercambio, es la cantidad de memoria adicional asignada por Linux para ejecutar programas de forma simultánea cuando sea necesario.

Para garantizar un rendimiento y uso óptimos del sistema, la swap crea temporalmente un espacio en el disco para liberar de carga a la memoria ram. Una vez hecho esto, la RAM vuelve a ser accesible para que la usen otras herramientas o aplicaciones. La definición correcta sería que... el espacio de intercambio se usa principalmente cuando la RAM no tiene suficiente tamaño para ejecutar las demandas solicitadas.



Qué es la cuenta root

La cuenta root, también conocida como raíz es la cuenta principal del administrador del sistema. Con esta cuenta puedes realizar todos los cambios necesarios en el sistema, lo controla por completo. Además, es la cuenta operativa predeterminada en todos los sistemas basados en Linux. Entrevista a un Sysadmin Linux.

Qué es la shell en linux

La Shell nos permite a través de un terminal, introducir comandos para indicar al sistema la acción que debe ejecutar. Linux ofrece varias opciones de Shell; sin embargo, te enumero las más conocidas y populares.

- **bash:** Bourne Again Shell (bash), es la shell más popular, ya que viene por defecto en la mayoría de distribuciones de Linux.
- **ksh:** Korn Shell (ksh), es un shell de lenguaje de programación de alto nivel, creado para administradores de sistemas y programadores.
- **zsh:** Z Shell (zsh), ofrece características únicas que incluyen comentarios de cierre, archivos de inicio, generación de nombres de archivo, etc.
- **csh:** C Shell (csh), tiene la particularidad de usar una sintaxis similar a C.

Verificar la memoria consumida

En la entrevista a un Sysadmin, nos preguntan como verificar la memoria de un sistema linux. Esto es algo realmente sencillo, pues disponemos de muchos comandos y herramientas que nos ayudan al respecto. Vemos algunos de los más conocidos.

```
vmstat
free -m
top
htop
```

```
[root@ ~]# vmstat
procs -----memory----- --swap-- -----io----- -system-- -----cpu-----
r b  swpd  free  buff  cache  si  so  bi  bo  in  cs  us  sy  id  wa  st
0  0      0 27800496 448584 3020924  0  0  27  67  0  1  5  1  94  0  0
```

Qué es el Master Boot Record

El Master Boot Record (MBR), se encarga de detectar el dispositivo principal de arranque. Contiene 512 bytes de datos y normalmente contiene el cargador de arranque. Las partes principales del MBR son las siguientes:

Los primeros 446 bytes contienen información del cargador de arranque principal.

En los siguientes 64 bytes se almacena la información de la tabla de particiones.

En los últimos 2 bytes está la verificación de validación MBR.

Qué es el nivel de ejecución en linux

El nivel de ejecución en Linux define como se ejecuta el sistema. En linux tenemos siete niveles de ejecución diferentes, y cada uno tiene su función.

- **0** : Se utiliza para detener el sistema, es decir, apagar la máquina.
- **1** : Es un nivel de ejecución en modo de usuario único.
- **2** : Es el modo multiusuario básico.
- **3** : Modo multiusuario con funciones completas.
- **4** : Es experimental, no se utiliza.
- **5** : Modo multiusuario basado en GUI.
- **6** : Reiniciar el sistema.

Linux es seguro

Linux es seguro si lo comparamos con otros sistemas operativos. Al ser de código abierto, son muchos los desarrolladores que trabajan en el código. Otras razones básicas que hacen que Linux sea más seguro, son las siguientes.

- Mejor gestión de usuarios garantizando las restricciones de bajo nivel. Así, los virus no pueden entrar en archivos o carpetas locales y se garantiza que el daño sea menor.
- Se puede especificar los permisos de una aplicación, de manera que garantizamos las mejores medidas de seguridad, posibles.
- Linux tiene un poderoso sistema de auditoría.
- Se ofrece el acceso a los registros de forma detallada, lo que facilita a los administradores descubrir accesos no autorizado u otras violaciones en el sistema



Comandos comunes para administrar una red

Linux es un sistema operativo que ofrece funciones completas de red. El administrador del sistema es el encargado de su configuración, así como de solucionar los problemas que puedan presentarse. Vemos algunos de los comandos que nos ayudaran a solucionar fallos en la red.

- **hostname** : Visualizar la dirección IP y el dominio del nombre de host. También permite cambiar el nombre de host.
- **ifconfig** : Permite manipular y mostrar las interfaces y rutas de la red.
- **tracpath** : Rastrea los saltos de un paquete hasta llegar a su destino.
- **route** : Imprime y edita la tabla de enrutamiento IP.
- **mtr** : Rastrear una ruta y hacer ping con solo un comando.
- **netstat** : Nos permite ver las tablas de enrutamiento, las conexiones de red y las estadísticas de la interfaz.
- **dig** : Consulta servidores de nombres DNS.
- **ping** : Verificar que un sistema remoto está disponible.

```
sergio@sololinux:~$ tracpath sololinux.es
1?: [LOCALHOST] pmtu 1500 www.sololinux.es
1: _gateway 3.130ms
1: _gateway 3.144ms
2: 10.132.0.28 10.465ms
3: 172.20.201.2 84.131ms
4: odin-vgw3.te.net.ua 2.279ms
5: br3-co-core3-dca.te.net.ua 20.386ms
6: odessa1-ge-0-0-0-857.ett.ua 7.159ms
7: fft1-od0.ett.ua 30.799ms
```

Qué son los demonios

Los demonios también conocidos como **Daemons**, permiten que nuestro sistema pueda ejecutar un proceso en segundo plano, el cual no puede hacerlo de forma nativa. Daemons es un enfoque seguro cuando hablamos de manejar solicitudes periódicas.

Nota final de la Entrevista a un Sysadmin

Llegamos al final de nuestras 17 preguntas de la entrevista a un sysadmin Linux. Recuerda que las preguntas formuladas en el artículo, tan solo son una base. Además, si pasas la selección de la entrevista lo más probable es que después realicen pruebas prácticas.

Instalar XFCE 4.16 en Ubuntu 20.04 o Linux Mint



Hace unos días, se anunció el lanzamiento de la nueva versión principal estable de **XFCE**, hablamos de **XFCE 4.16**. No nos vamos a explayar mucho en sus nuevas características y mejoras, ya lo hicimos en este artículo [anterior](#), aun así, recordamos las principales características y novedades.

Las características de Xfce 4.16 incluyen:

- Nuevos iconos y paleta.
- El nuevo marco de diálogo de «Aplicaciones predeterminadas», se fusiona con los de «Configuración de Mime» y «Aplicaciones preferidas».
- Se mejora la compatibilidad con HiDPI (escala fraccionada).
- Más atajos de teclado de forma predeterminada.
- Se añade un botón de pausa en los cuadros de diálogo de copia y movimiento de archivos en Thunar.
- Modo de panel oscuro (habilitado de forma predeterminada).
- El nuevo complemento «Status Tray», combina el «Systray» heredado y «StatusNotifier».
- Nuevo marco de diálogo (Acerca de...), con información básica del sistema.
- Se mejora la configuración del administrador de energía.
- Puedes descubrir más características [aquí](#).

Si tu versión preferida de Ubuntu es Xubuntu, o Ubuntu con XFCE, es recomendable que actualices el XFCE. En este artículo vemos como hacerlo de forma simple y rápida, además también es válido para Linux Mint y otros derivados de Ubuntu. **Instalar XFCE 4.16**.



Instalar XFCE 4.16 en Ubuntu 20.04

Xubuntu Developers nos ofrece un ppa que facilita su instalación, por tanto lo agregamos.

```
sudo add-apt-repository ppa:xubuntu-dev/staging
```

Como ya tienes XFCE instalado en el sistema, tan solo debemos actualizar nuestro sistema íntegramente.

```
sudo apt update
sudo apt full-upgrade
```

Acepta la solicitud del sistema pulsando «Y». La instalación comienza inmediatamente, vemos un ejemplo.

```
root@sololinux-demo:~# apt full-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer
required:
  libunique-1.0-0 xfce4-notes
Use 'apt autoremove' to remove them.
The following packages will be REMOVED:
  libexo-1-0 libexo-helpers xfce4-mailwatch-plugin xfce4-notes-plugin
The following NEW packages will be installed:
  gir1.2-xfconf-0 xfce4-helpers
The following packages will be upgraded:
  catfish elementary-xfce-icon-theme exo-utils libexo-2-0 libexo-
common
  libthunarx-3-0 libtumbler-1-0 libxfce4panel-2.0-4 libxfce4ui-2-0
  libxfce4ui-common libxfce4ui-utils libxfce4util-bin libxfce4util-
common
  libxfce4util7 libxfconf-0-3 mousepad thunar thunar-data thunar-
volman
  tumbler tumbler-common xfce4-appfinder xfce4-dict xfce4-notifyd
  xfce4-panel
  xfce4-power-manager xfce4-power-manager-data xfce4-power-manager-
plugins
  xfce4-pulseaudio-plugin xfce4-screensaver xfce4-screenshooter
  xfce4-session
  xfce4-settings xfce4-taskmanager xfce4-terminal xfce4-weather-
plugin
  xfce4-whiskermenu-plugin xfconf xfdesktop4 xfdesktop4-data xfwm4
41 upgraded, 2 newly installed, 4 to remove and 0 not upgraded.
Need to get 16.9 MB of archives.
After this operation, 1550 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Observa en la siguiente imagen como se actualiza XFCE, a XFCE 4.16.

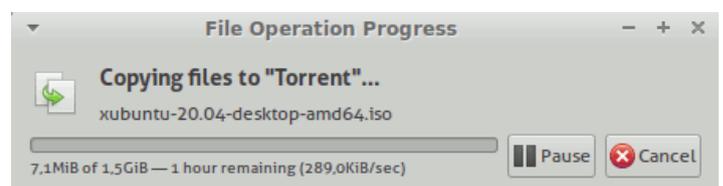
```
Preparing to unpack .../18-xfce4-panel_4.16.0-1-20.04_amd64.deb ...
Unpacking xfce4-panel (4.16.0-1-20.04) over (4.14.3-1) ... www.sololinux.es
Preparing to unpack .../19-xfce4-power-manager_4.16.0-1-20.04_amd64.deb ...
Unpacking xfce4-power-manager (4.16.0-1-20.04) over (1.6.6-1) ...
Preparing to unpack .../20-xfce4-power-manager-data_4.16.0-1-20.04_all.deb ...
Unpacking xfce4-power-manager-data (4.16.0-1-20.04) over (1.6.6-1) ...
Preparing to unpack .../21-xfce4-power-manager-plugins_4.16.0-1-20.04_amd64.deb
```

Para una instalación limpia ejecuta lo siguiente.

```
sudo apt update
sudo apt install xubuntu-desktop
```

Para disfrutar de tu nuevo XFCE, solo falta reiniciar el sistema.

```
sudo reboot
```



Uso del Comando Nslookup en linux



Hoy hablamos de un comando desconocido por los usuarios más noveles, pero tremendamente útil a la hora de solucionar cualquier problema que tengas con las **DNS**. El **comando Nslookup** tiene la capacidad de localizar la dirección IP de un host, el dominio de una dirección IP, o incluso localizar los **servidores de correo** de un dominio.

Esta herramienta se puede utilizar en dos formas.

- Modo interactivo
- Modo no interactivo

El modo interactivo se usa para lanzar consultas a un servidor DNS sobre sus dominios y hosts. Por el contrario, el modo no interactivo lo utilizaremos para solicitar información sobre un host o dominio.

Este comando suele venir instalado en la mayoría de **distribuciones linux**, por si acaso no es tu caso, además de aprender a usarlo, también veremos como se instala en las distribuciones linux más conocidas.

```
sergio@sololinux:~$ nslookup
> google.es
Server:      127.0.0.53
Address:    127.0.0.53#53

Non-authoritative answer:
Name:   google.es
Address: 142.250.75.3
Name:   google.es
Address: 2a00:1450:401b:801::2003
> sololinux.es
Server:      127.0.0.53
Address:    127.0.0.53#53
```

www.sololinux.es

Instalar el comando Nslookup en linux

Como ya dije anteriormente, la herramienta no siempre viene instalada por defecto. La instalamos.

Instalar Nslookup en Debian, Ubuntu y derivados

```
sudo apt install dnsutils -y
```

Instalar Nslookup en CentOS, Fedora y derivados

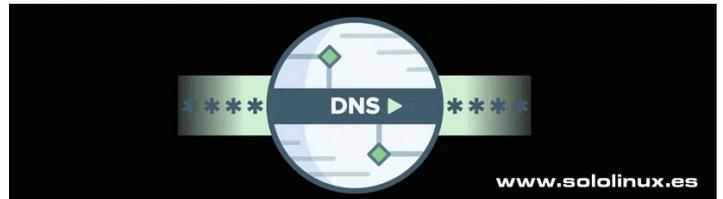
```
sudo dnf install bind-utils
# 0
sudo yum install bind-utils
```

Instalar Nslookup en Arch Linux, Manjaro y derivados

```
sudo pacman -S dnsutils
```

Si ya lo tienes instalado, recibirás un mensaje similar a...

```
root@sololinux:# sudo apt install dnsutils -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
dnsutils ya está en su versión más reciente (1:9.11.3+dfsg-1ubuntu1.13).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
```



Uso del comando Nslookup en linux

En modo interactivo

Para acceder al modo interactivo, tan solo debes ejecutar el siguiente comando.

```
nslookup
```

Como vemos en el siguiente ejemplo, accedemos a la consola de la herramienta.

```
sergio@sololinux:~$ nslookup
>
```

Para obtener resultados, inserta un dominio o una ip (en el ejemplo «**google.es**»).

```
sergio@sololinux:~$ nslookup
> google.es
Server: 127.0.0.53
Address: 127.0.0.53#53
Non-authoritative answer:
Name: google.es
Address: 142.250.75.3
Name: google.es
Address: 2a00:1450:401b:801::2003
>
```

También podemos especificar el tipo de búsqueda. En nuestro caso las DNS.

```
set type=ns
```

```
sergio@sololinux:~$ nslookup
> set type=ns
> google.es
Server: 127.0.0.53
Address: 127.0.0.53#53
Non-authoritative answer:
google.es nameserver = ns3.google.com.
google.es nameserver = ns2.google.com.
google.es nameserver = ns4.google.com.
google.es nameserver = ns1.google.com.
Authoritative answers can be found from:
>
```

Incluso podemos identificar rápidamente los servidores de correo (mail).

```
set type=mx

sergio@sololinux:~$ nslookup
> set type=mx
> google.es
Server: 127.0.0.53
Address: 127.0.0.53#53
Non-authoritative answer:
google.es mail exchanger = 20 alt1.aspmx.l.google.com.
google.es mail exchanger = 30 alt2.aspmx.l.google.com.
google.es mail exchanger = 50 alt4.aspmx.l.google.com.
google.es mail exchanger = 40 alt3.aspmx.l.google.com.
google.es mail exchanger = 10 aspmx.l.google.com.
Authoritative answers can be found from:
>
```

Impresionante verdad?, ahora vemos el modo no interactivo.

```
sergio@sololinux:~$ nslookup
> set type=ns
> google.es
Server: 127.0.0.53
Address: 127.0.0.53#53

Non-authoritative answer:
google.es nameserver = ns3.google.com.
google.es nameserver = ns2.google.com.
google.es nameserver = ns4.google.com.
google.es nameserver = ns1.google.com.

Authoritative answers can be found from:
> set type=mx
> google.es
Server: 127.0.0.53
Address: 127.0.0.53#53

Non-authoritative answer:
google.es mail exchanger = 10 aspmx.l.google.com.
google.es mail exchanger = 40 alt3.aspmx.l.google.com.
google.es mail exchanger = 50 alt4.aspmx.l.google.com.
google.es mail exchanger = 30 alt2.aspmx.l.google.com.
google.es mail exchanger = 20 alt1.aspmx.l.google.com.

Authoritative answers can be found from:
>
```

En modo no interactivo

En el modo no interactivo, insertamos el comando seguido de la solicitud. Vemos un ejemplo.

```
nslookup google.es

sergio@sololinux:~$ nslookup google.es
Server: 127.0.0.53
Address: 127.0.0.53#53
Non-authoritative answer:
Name: google.es
Address: 142.250.75.3
Name: google.es
Address: 2a00:1450:401b:801::2003
sergio@sololinux:~$
```

También podemos ver los registros SOA.

```
nslookup -type=soa

sergio@sololinux:~$ nslookup -type=soa google.es
Server: 127.0.0.53
Address: 127.0.0.53#53
Non-authoritative answer:
google.es
origin = ns1.google.com
mail addr = dns-admin.google.com
serial = 349260167
refresh = 900
retry = 900
expire = 1800
minimum = 60
Authoritative answers can be found from:
sergio@sololinux:~$
```

Como ultima introducción en este comando, vemos otra opción interesante con la cual podrás visualizar todos los registros posibles.

```
nslookup -query=any

sergio@sololinux:~$ nslookup -query=any google.es
Server: 127.0.0.53
Address: 127.0.0.53#53

Non-authoritative answer:
Name: google.es
Address: 142.250.75.3
Name: google.es
Address: 2a00:1450:401b:801::2003
google.es
origin = ns1.google.com
mail addr = dns-admin.google.com
serial = 349260167
refresh = 900
retry = 900
expire = 1800
minimum = 60
www.sololinux.es
google.es nameserver = ns1.google.com.
google.es nameserver = ns4.google.com.
google.es nameserver = ns2.google.com.
google.es nameserver = ns3.google.com.

Authoritative answers can be found from:
```



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 SOLO LINUX MULTIPLICARAS TUS CLIENTES

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



Uso del Comando Netstat en Linux



Netstat es una interesante herramienta en línea de comandos, con la capacidad de enumerar todas las conexiones de red existentes en nuestro sistema linux.

Enumera todas las conexiones de los **socket tcp, udp** y las conexiones privadas de los sistemas derivados de UNIX. También muestra interesante información sobre la red, como por ejemplo las conexiones, las tablas de enrutamiento, estadísticas de la interfaz, conexiones enmascaradas, membresías de multidifusión, y mucho más.

En este sencillo artículo, veremos cómo usar la utilidad para recopilar información sobre las conexiones de red y, puertos abiertos en el sistema. Es la info que por circunstancias diversas más necesitamos.

Uso del comando Netstat en linux

Listar todas la conexiones

```
sergio@sololinux:~$ netstat -a
Conexiones activas de Internet (servidores y establecidos)
Proto Recib Envíad Dirección local Dirección remota Estado
tcp 0 0 localhost:ipp 0.0.0.0:* ESCUCHAR
tcp 0 0 localhost:domain 0.0.0.0:* ESCUCHAR
tcp 0 0 sololinux:39380 static.200.4.202.:https ESTABLECIDO
tcp 0 0 sololinux:49836 104.18.3.159:https ESTABLECIDO
tcp 0 0 sololinux:34796 199.232.16.134:https ESTABLECIDO
tcp 0 0 sololinux:39770 flucast02-06.bt.b:https ESTABLECIDO
tcp 0 0 sololinux:41102 li-in-f188.1e100.:https ESTABLECIDO
tcp 0 0 sololinux:40566 ae9d0fe0e9c2cb798:https ESTABLECIDO
TIME_WAIT
tcp 0 0 sololinux:39390 static.183.67.216:https ESTABLECIDO
tcp 0 0 sololinux:60382 104.18.169.19:https ESTABLECIDO
tcp6 0 0 ip6-localhost:ipp [::]:* ESCUCHAR
udp 0 0 localhost:domain 0.0.0.0:*
udp 0 0 0.0.0.0:bootpc 0.0.0.0:*
udp 0 0 sololinux:ntp 0.0.0.0:*
udp 0 0 localhost:ntp 0.0.0.0:*
udp 0 0 0.0.0.0:ntp 0.0.0.0:*
udp 0 0 0.0.0.0:39063 0.0.0.0:*
udp 0 0 0.0.0.0:ipp 0.0.0.0:*
udp 0 0 sololinux:48159 0.0.0.0:*
udp 0 0 224.0.0.251:mdns 0.0.0.0:*
udp 0 0 224.0.0.251:mdns 0.0.0.0:*
udp 0 0 0.0.0.0:mdns 0.0.0.0:*
udp 0 0 sololinux:50461 waw02s07-in-f14.1e1:443 ESTABLECIDO
```

Identificar las interfaces de red

```
netstat -i

sergio@sololinux:~$ netstat -i
Tabla de la interfaz del núcleo
Iface MTU RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg
eno1 1500 0 0 0 0 0 0 0 0
0 0 0 BMU
lo 65536 12937 0 0 0 12937 0 0 0
0 0 0 LRU
wlo1 1500 751088 0 0 0 424640 0 0
0 0 0 BMRU
```

Otra opción más legible y con más datos.

```
netstat -ie

sergio@sololinux:~$ netstat -ie
Tabla de la interfaz del núcleo
eno1: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
ether 38:22:e2:ba:27:73 txqueuelen 1000 (Ethernet)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Bucle local)
RX packets 12961 bytes 1287919 (1.2 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 12961 bytes 1287919 (1.2 MB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlo1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.0.109 netmask 255.255.255.0 broadcast 192.168.0.255
inet6 fe80::e8f4:fa6f:11f1:74f8 prefixlen 64 scopeid 0x20<link>
ether 70:66:55:01:02:4d txqueuelen 1000 (Ethernet)
RX packets 753512 bytes 911881727 (911.8 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 426779 bytes 63666691 (63.6 MB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Listar las conexiones activas del sistema

```
netstat -atnp | grep ESTA

sergio@sololinux:~$ netstat -atnp | grep ESTA
(No todos los procesos pueden ser identificados, no hay información de propiedad del proceso)
no se mostrarán, necesita ser superusuario para verlos todos.)
tcp 0 0 192.168.0.109:56560 163.172.52.4:22 ESTABLECIDO -
tcp 0 0 192.168.0.109:43042 62.210.131.25:232 ESTABLECIDO -
tcp 0 0 192.168.0.109:56906 104.244.42.8:443 ESTABLECIDO 1792/chromium-brows
tcp 0 0 192.168.0.109:39770 195.10.2.92:443 ESTABLECIDO 1792/chromium-brows
tcp 0 0 192.168.0.109:41102 64.233.162.188:443 ESTABLECIDO 1792/chromium-brows
tcp 0 0 192.168.0.109:45458 52.7.174.108:443 ESTABLECIDO 1792/chromium-brows
tcp 0 0 192.168.0.109:39402 116.202.4.200:443 ESTABLECIDO 1792/chromium-brows
tcp 0 0 192.168.0.109:48044 199.232.16.157:443 ESTABLECIDO 1792/chromium-brows
tcp 0 0 192.168.0.109:54856 195.10.2.75:443 ESTABLECIDO 1792/chromium-brows
tcp 0 0 192.168.0.109:35294 163.172.52.4:443 ESTABLECIDO 1792/chromium-brows
tcp 0 0 192.168.0.109:34934 149.154.167.51:443 ESTABLECIDO 23756/telegram
tcp 0 0 192.168.0.109:35354 163.172.52.4:443 ESTABLECIDO 1792/chromium-brows
```



Listar las conexiones TCP

```
netstat -at
```

```
sergio@sololinux:~$ netstat -at
Conexiones activas de Internet (servidores y establecidos)
Proto Recib Enviad Dirección local      Dirección
remota      Estado
tcp        0      0 localhost:ipp      0.0.0.0:*
          ESCUCHAR
tcp        0      0 localhost:domain  0.0.0.0:*
          ESCUCHAR
tcp        0      0 sololinux:56906   104.244.42.8:https ESTABLECIDO
tcp        0      0 sololinux:39770   06.bt.b:https     ESTABLECIDO
tcp        0      0 sololinux:41102   f188.1e100.:https ESTABLECIDO
tcp        0      0 sololinux:39412   static.200.4.202.:https ESTABLECIDO
tcp        0      0 sololinux:39410   static.200.4.202.:https ESTABLECIDO
tcp        0      0 sololinux:45458   108.:https        ESTABLECIDO
tcp        0      0 sololinux:48044   199.232.16.157:https TIME_WAIT
tcp        0      0 sololinux:54856   04.bt.b:https     ESTABLECIDO
tcp        0      0 sololinux:34934   149.154.167.51:https ESTABLECIDO
tcp6       0      0 ip6-localhost:ipp [::]:*
          ESCUCHAR
```

Listar las conexiones UDP

```
netstat -au
```

```
sergio@sololinux:~$ netstat -au
Conexiones activas de Internet (servidores y establecidos)
Proto Recib Enviad Dirección local      Dirección remota      Estado
udp    0      0 sololinux:38738   sof01s12-in-f10.1e1:443 ESTABLECIDO
udp    0      0 localhost:domain  0.0.0.0:*
udp    0      0 0.0.0.0:bootpc   0.0.0.0:*
udp    0      0 sololinux:ntp     0.0.0.0:*
udp    0      0 localhost:ntp     0.0.0.0:*
udp    0      0 0.0.0.0:ntp      0.0.0.0:*
udp    0      0 sololinux:53385   waw02s17-in-f10.1e1:443 ESTABLECIDO
udp    0      0 0.0.0.0:39063    0.0.0.0:*
udp    0      0 0.0.0.0:ipp      0.0.0.0:*
udp    0      0 sololinux:40008   waw02s07-in-f174.1e:443 ESTABLECIDO
udp    0      0 224.0.0.251:mdns 0.0.0.0:*
udp    0      0 224.0.0.251:mdns 0.0.0.0:*
udp    0      0 0.0.0.0:mdns     0.0.0.0:*
udp6   0      0 sololinux:ntp     [::]:*
udp6   0      0 ip6-localhost:ntp [::]:*
udp6   0      0 [::]:ntp          [::]:*
udp6   0      0 [::]:mdns         [::]:*
udp6   0      0 [::]:36455        [::]:*
www.sololinux.es
```

Puertos abiertos que escuchan un servicio

```
netstat -tnl
```

```
sergio@sololinux:~$ netstat -tnl
Conexiones activas de Internet (solo servidores)
Proto Recib Enviad Dirección local      Dirección
remota      Estado
tcp        0      0 127.0.0.1:631    0.0.0.0:*
          ESCUCHAR
tcp        0      0 127.0.0.53:53    0.0.0.0:*
          ESCUCHAR
tcp6       0      0 ::1:631          :::*
```

Si quieres conocer el servicio (solo en servidores o vps), usa la opción propuesta a continuación.

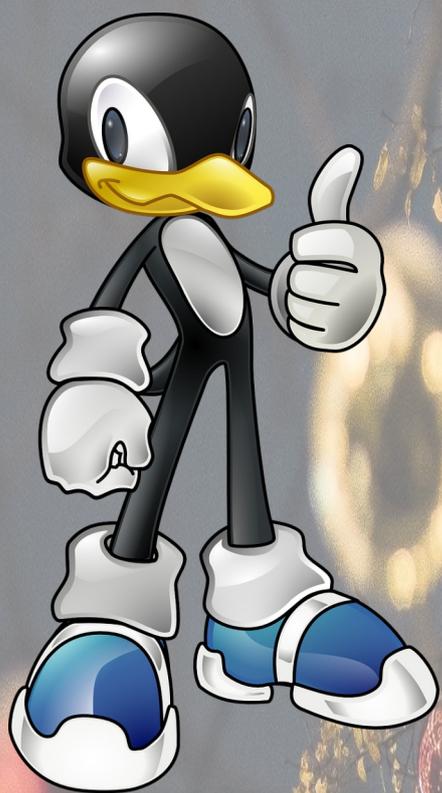
```
netstat -nlpt
```

Estadísticas de la red

Como ultima opción propuesta en este artículo, vemos las estadísticas detalladas de la red con el comando Netstat.

```
netstat -s
```

```
sergio@sololinux:~$ netstat -s
Ip:
  Forwarding: 2
  787008 total packets received
  2 with invalid addresses
  0 forwarded
  0 incoming packets discarded
  762528 incoming packets delivered
  455462 requests sent out
Icmp:
  244 ICMP messages received
  49 input ICMP message failed
  histograma de entrada ICMP:
    destination unreachable: 233
    timeout in transit: 11
  18 ICMP messages sent
  0 ICMP messages failed
  histograma de salida ICMP:
    destination unreachable: 18
IcmpMsg:
  InType3: 233
  InType11: 11
  OutType3: 18
Tcp:
  2477 active connection openings
  3 passive connection openings
  88 failed connection attempts
  301 connection resets received
  11 connections established
  448752 segments received
  368043 segments sent out
  350 segments retransmitted
  6 bad segments received
  1687 resets sent
  InCsumErrors: 1
Udp:
  309178 packets received
  18 packets to unknown port received
  4609 packet receive errors
  89493 packets sent
  4609 receive buffer errors
  0 send buffer errors
  IgnoredMulti: 21
UdpLite:
TcpExt:
  626 TCP sockets finished time wait in fast timer
  1 packets rejected in established connections because of
  timestamp
  83348 delayed acks sent
  5 delayed acks further delayed because of locked socket
  Quick ack mode was activated 592 times
  331628 packet headers predicted
  17077 acknowledgments not containing data payload received
  20129 predicted acknowledgments
  TCPSackRecovery: 14
  Detected reordering 284 times using SACK
  TCPSACKUndo: 11
  4 congestion windows recovered without slow start after partial
  ack
  TCPLostRetransmit: 88
  29 fast retransmits
  TCPTimeouts: 140
  TCPLossProbes: 201
  TCPLossProbeRecovery: 8
  TCPBacklogCoalesce: 2320
  TCPSACKOldSent: 592
  TCPSACKOffoSent: 173
  TCPSACKRecv: 151
  265 connections reset due to unexpected data
  183 connections reset due to early user close
  5 connections aborted due to timeout
  TCPSACKIgnoredNoUndo: 36
  TCPSpuriousRTOs: 1
  TCPSackShifted: 4
  TCPSackMerged: 6
  TCPSackShiftFallback: 261
  TCPRecvCoalesce: 157854
  TCPDFoQueue: 18630
  TCPDFoMerge: 173
  TCPChallengeACK: 7
  TCPSYNChallenge: 7
  TCPSpuriousRtxHostQueues: 12
  TCPAutoCorking: 4163
  TCPFromZeroWindowAdv: 1
  TCPToZeroWindowAdv: 3
  TCPWantZeroWindowAdv: 93
  TCPSynRetrans: 50
  TCPOrigDataSent: 38949
  TCPHystartTrainDetect: 1
  TCPHystartTrainCwnd: 22
  TCPHystartDelayDetect: 4
  TCPHystartDelayCwnd: 170
  TCPKeepAlive: 7724
  TCPDelivered: 40786
  TCPAckCompressed: 7948
IpExt:
  InMcastPkts: 1329
  OutMcastPkts: 1929
  InBcastPkts: 21
  OutBcastPkts: 6
  InOctets: 929296492
  OutOctets: 48770607
  InMcastOctets: 99055
  OutMcastOctets: 323396
  InBcastOctets: 5486
  OutBcastOctets: 468
  InNoECTPkts: 787008
```

TU REVISTA
LA DE TODOS

www.sololinux.es