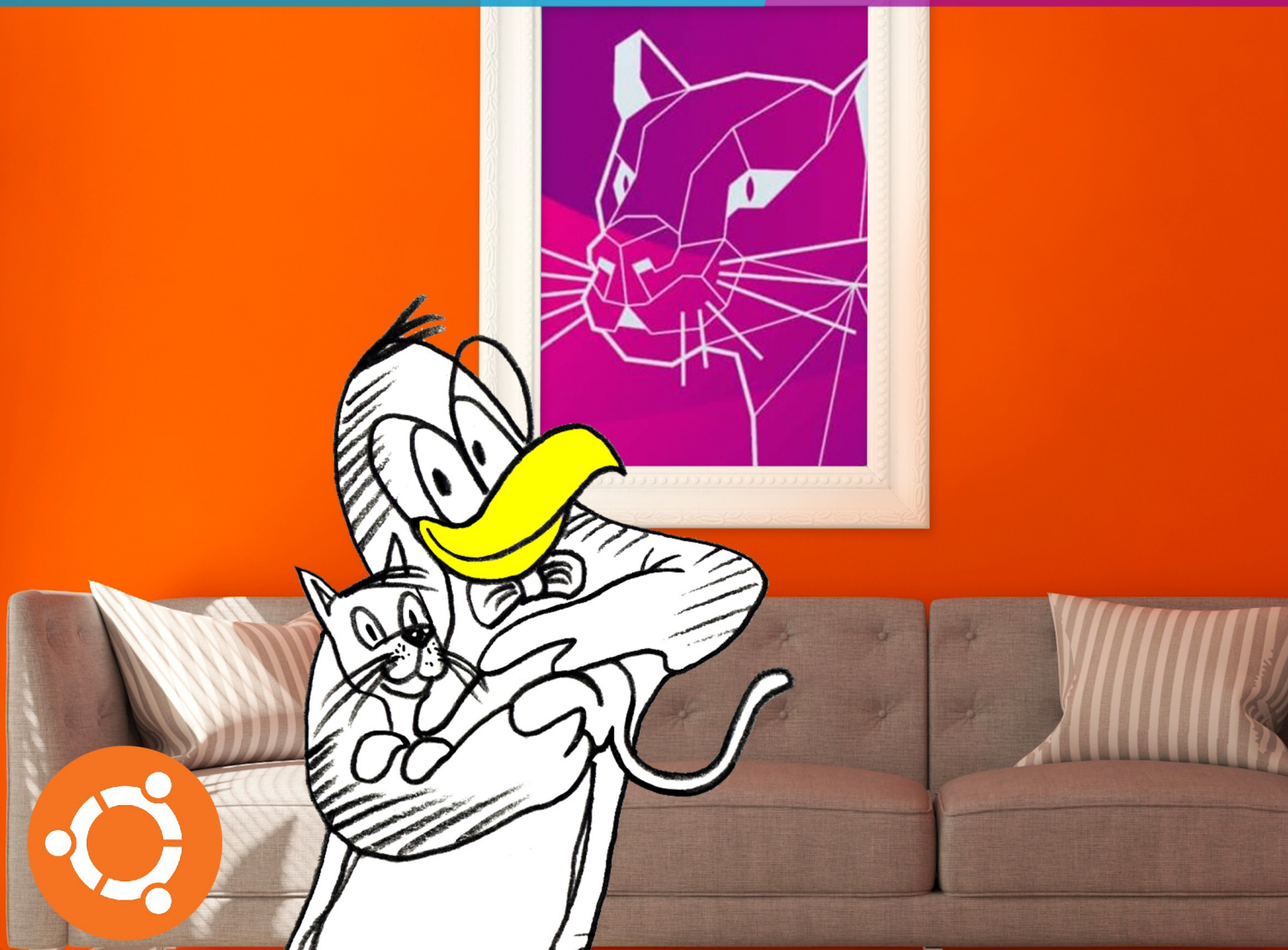


MAGAZINE SOLO LINUX

Nº
16

Tu revista, la revista de tod@s

MAYO 2020



Ubuntu 20.04 LTS Focal Fossa

#QuedateEnCasa

Qué hacer después de
instalar **Ubuntu 20.04
Focal Fossa**

Instalar **Unity Desktop
Environment** en **Ubuntu
20.04 LTS**

Crear directorios y
subdirectorios desde un
archivo de texto

El nuevo **Tails 4.6** ya está
listo para su descarga

Por qué Linux es tan popular
en alojamientos web

Como configurar una red
con **nmtui** en Linux

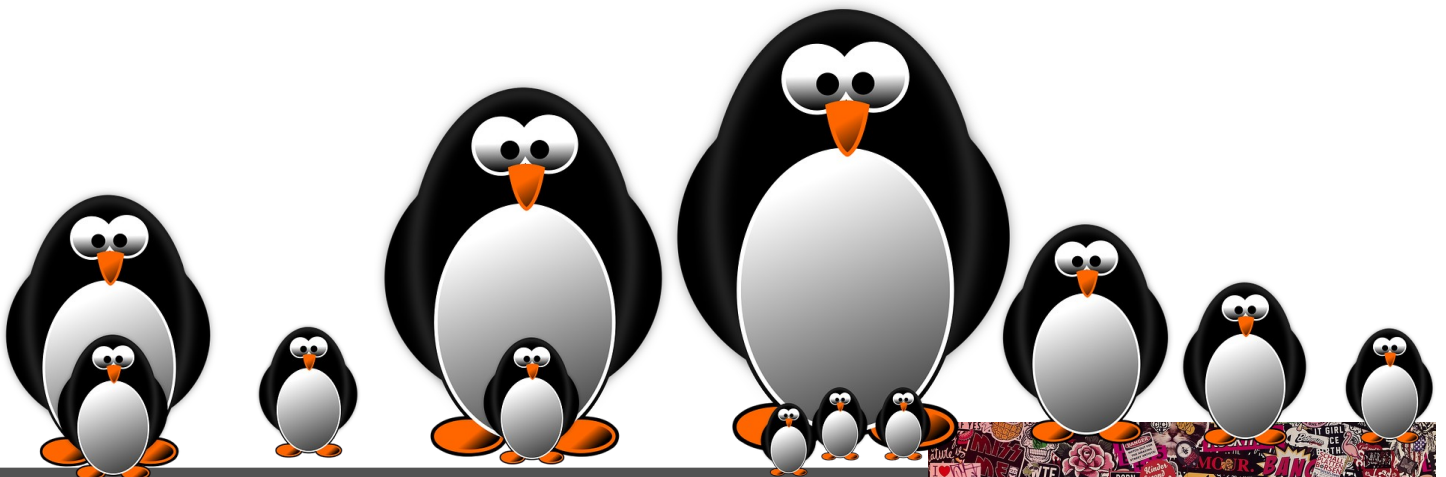
MANUALES, SCRIPTS, SOFTWARE, HARDWARE, DISTROS LINUX,
SEGURIDAD, REDES Y MUCHO MAS EN LA WEB...

¿Te interesa colaborar con SoloLinux?

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

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

Ayúdanos a crecer, vota por SoloLinux en la siguiente [encuesta](#).
Gracias por colaborar.



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



Bienvenido a la Revista SOLOLINUX

Aquí tenemos el número 16 de la revista SoLoLinux. Gracias a todos los que hacéis posible que esto siga adelante. Desde SOLOLINUX esperamos que os guste este número.

Igual que en números anteriores nos gustaría animar a todos nuestros lectores para que nos envíen sus **opiniones sobre el Software Libre o sobre GNU/Linux**, pueden enviarlo a adrian@sololinux.es, con ello queremos proponer que cada mes se publicada una o varias de esas opiniones sobre lo mencionado en la nueva sección de la

revista **OPINIÓN DEL LECTOR**. Queremos saber la **opinión de todos**. Se intentará incluir el máximo de opiniones en

cada número, pero si no sale la tuya este mes no desesperes, al siguiente podría tener un hueco en la revista. **ANÍMENSE Y ENVÍEN SUS OPINIONES**. Gracias.

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

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

Gracias a TOD@S

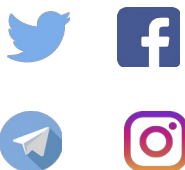
Compartan esta revista en sus redes sociales o web. Revista digital **SOLOLINUX MAGAZINE**. Tu revista, la revista de todos.

Adrián A. A.

**Ayúdanos a crecer, vota por
SoloLinux en la siguiente encuesta.
Gracias por colaborar.**

SoloLinux

Síguenos en las Redes:



La revista SOLOLINUX
esta realizada con Libre
Office Impress 6.2.8.

Nuestras Webs:

www.sololinux.es
www.solowordpress.es

EDITORIAL

Edición:

• **Adrián A. A.**
adrian@sololinux.es

Redacción:

• **Sergio G. B.**
(Administrador y redactor artículos
SoloLinux)
info@sololinux.es

• **Henry G. R.**
(Redactor artículos SoloWordPress)
info@solowordpress.es

Agradecimientos:

• **Ricardo Fernández** (Donación)
• **OpenIT** (Donación)

Diseños de 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 email a: adrian@sololinux.es

Colabora:

Quieres colaborar en la revista. Para mayor información escribe un email a: adrian@sololinux.es

La **Revista SOLOLINUX**, se distribuye gratuitamente en forma digital para todo el mundo que quiere disfrutar de ella. Si quieres imprimirla es cosa tuya. Si os cobran por ella, os están timando. :)

Contacto:

Para cualquier consulta sobre las revistas, publicidad o colaboraciones escribir un email a: adrian@sololinux.es



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

MANUALES

- 08. Que hacer después de instalar Ubuntu 20.04 Focal Fossa
- 12. Instalar Chromium en Ubuntu, Linux Mint y derivados
- 13. Instalar Unity Desktop Environment en Ubuntu 20.04 LTS
- 16. Unir imágenes con el comando convert de Image Magik
- 18. Como instalar aMule en Ubuntu 20.04 LTS Focal Fossa
- 20. Crear directorios y subdirectorios desde un archivo de texto
- 28. Como ejecutar un script al iniciar nuestro sistema
- 31. Administrar tarjetas ethernet con el comando ethtool
- 34. Proteger un servidor contra ataques SYN y suplantación de IP
- 35. Solución al error: sudo: add-apt-repository: command not found
- 38. Como instalar Mono (.NET Framework) en Ubuntu 20.04
- 39. Instalar Samba en Ubuntu 20.04 LTS Focal y derivados
- 41. Instalar el tema MacOS Catalina en linux con XFCE
- 43. Instalar NetSurf en Linux – El navegador web ligero
- 44. Conectar un Android o iPhone con Manjaro y Arch Linux
- 45. Obtener información del hardware con hwinfo
- 47. VestaCP Plugin Manager – El administrador definitivo
- 50. Como encriptar un Pendrive con VeraCrypt en linux
- 53. Limitar los recursos de procesos con systemd en linux
- 57. Comprimir archivos con zip y protegerlos con contraseña
- 58. Solución: Error Minimal BASH like line editing is supported
- 59. Uso del comando uptime – Tiempo de actividad del sistema
- 60. Habilitar el servicio ssh en Ubuntu 20.04 y derivados

SSH

service

www.sololinux.es

DISTROS LINUX

- 14. El nuevo Tails 4.6 ya está listo para su descarga
- 23. Por qué Linux es tan popular en alojamientos web

 **Tails**
the amnesic incognito live system
4.6

www.sololinux.es

SCRIPTS

- 15. Bashtop – El monitor de linux escrito en bash
- 36. Verificar la conexión SSH de un host remoto
- 55. Eliminar directorios vacíos con un script bash
- 56. Bloquear el acceso a los usuarios que utilizan la red Tor

SEGURIDAD

- 25. Verificar los password con John The Ripper en Ubuntu
- 26. Instalar las herramientas de Kali en Ubuntu 20.04 LTS

REDES

- 27. Como configurar una red con nmtui en linux
- 30. Extraer las claves wifi guardadas en linux



HARDWARE

- 11. Los nuevos Lenovo ThinkPad con Fedora 32 están a punto
- 19. Instalar los drivers de impresoras Canon en Ubuntu 20.04
- 21. Analizar la velocidad de tu disco duro con FIO
- 42. Nuevo Hplib 3.20.5 con soporte para debian 10.3

SOFTWARE

- 17. Instalar LMMS Linux Multimedia Studio en Ubuntu 20.04 LTS
- 24. Dropbear SSH la alternativa ligera a OpenSSH
- 49. Instalar Transmission 3.00 en cualquier distribución Linux



VANT

SOMOS LINUXEROS



la gama mas completa de ordenadores con GNU/Linux



EDGE

CPU Intel 10ª generación
Pantalla FHD de 14"
Hasta 40GB de RAM
SSD Samsung 970EVO Plus
Ligero (1,1 Kg de peso)
+10 horas de autonomía
GNU/Linux

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



- Revista Bimestral
- Noticias
- Manuales
- Temas
- Seguridad
- Plugins
- SEO
- OTROS

¡Si crees que puedes ayudar contacta con nosotros!

SoloWordpress

Tu revista sobre Wordpress



Si el formato digital no te convence, también tenemos todo el contenido en una **Página Web**

¡Visítanos!
www.solowordpress.es



APRENDE LINUX

**INSTAGRAM LIVES:
TODOS LOS LUNES,
MIÉRCOLES Y VIERNES**

@FABIANAMPALIO

 +54 9 11 6969 9993

 www.linkedin.com/in/Fabian-Ampalio

@exameneslinux 

@fabianampalio 

Que hacer después de instalar Ubuntu 20.04 Focal Fossa

Después del esperado lanzamiento de la última **versión LTS** del sistema operativo de **Canonical**, conocido como **Ubuntu 20.04 Focal Fossa**. Ha llegado el momento de analizarlo, e instalar o configurar aplicaciones y pequeños detalles que nos pueden ser muy útiles.

Las **distribuciones Linux**, ofrecen una experiencia de usuario excelente en sus diferentes entornos gráficos, además... cada día más interactivas. Esto me genera la duda de si crear uno de mis clásicos artículos del tipo «**que hacer después de...**» operando con la terminal del sistema, o en modo exclusivamente gráfico sobre el escritorio.

Después de analizar y recabar opiniones entre algunos usuarios, la cosa está clara. Haremos el artículo clásico, siempre trabajando en nuestra consola. Te preguntaras el porqué...

Es evidente que como en la **última versión de Ubuntu**, también redactaremos uno en modo gráfico. Pero a pesar de ir a contracorriente (de otras publicaciones), considero que la principal debe ser en consola, si no fuera así perderíamos nuestra esencia y principios sobre linux. Así que vamos a ello...

Que hacer después de instalar Ubuntu 20.04

Si no quieres estar escribiendo continuamente **sudo**, puedes operar como **root** (**bajo tu responsabilidad**) y trabajar más cómodo. Introduce el password del root dos veces.

```
sudo -i
sudo passwd root
```

Bueno... es importante actualizar nuestro sistema antes de comenzar.

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

No desespere... dependiendo de la velocidad de tu conexión, y del servidor de tu región, tardara más o menos.

```
Get:29 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 DEP-11 Metadata [48.4 kB]
Get:30 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [35.5 kB]
Get:31 http://archive.ubuntu.com/ubuntu focal-updates/main Translation-en [15.0 kB]
Get:32 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [14.6 kB]
Get:33 http://archive.ubuntu.com/ubuntu focal-updates/main DEP-11 48x48 Icons [1175 B]
Get:34 http://archive.ubuntu.com/ubuntu focal-updates/main DEP-11 64x64 Icons [1410 B]
Get:35 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [1232 B]
Get:36 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [2948 B]
Get:37 http://archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [936 B]
Get:38 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [12.3 kB]
Get:39 http://archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [5676 B]
Get:40 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [212 B]
Get:41 http://archive.ubuntu.com/ubuntu focal-updates/universe DEP-11 48x48 Icons [29 B]
Get:42 http://archive.ubuntu.com/ubuntu focal-updates/universe DEP-11 64x64 Icons [29 B]
Get:43 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [788 B]
Fetched 31.1 MB in 1min 1s (511 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
32 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following NEW packages will be installed:
  linux-headers-5.4.0-28 linux-headers-5.4.0-28-generic linux-image-5.4.0-28-generic linux-modules-5.4.0-28-generic linux-modules-extra-5.4.0-28-generic
The following packages will be upgraded:
  cups-bsd cups-client cups-common cups-core-drivers cups-daemon cups-ipp-utils cups-ppdc cups-server-common distro-info-data gir1.2-javascriptcoregtk-4.0 gir1.2-webkit2-4.0
  libcups2 libcupsimage2 libjavascriptcoregtk-4.0-18 libldb2 libnetplan0 libopenexr24 libpython3.8 libpython3.8-minimal libpython3.8-stdlib libsmclient libwbclient0
  libwebkit2gtk-4.0-37 linux-generic linux-headers-generic linux-image-generic netplan.io python3.8 python3.8-minimal samba/libs strace
32 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 107 MB of archives.
After this operation, 359 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Imagen 1. Actualizar Ubuntu 20.04

Opcional, bajo tu responsabilidad (no recomendado si eres un usuario novel). Si quieres instalar los **drivers** privativos los puedes descargar desde:

- **Nvidia:** <http://www.nvidia.es/Download/index.aspx?lang=es>
- **ATI:** <https://support.amd.com/en-us/download>

En este punto debo recordaros que **Intel** ya no ofrece actualizaciones de sus drivers privativos, vienen con el sistema. Para los que echen de menos el **ppa:xorg-edgers**, os comunico que tampoco se mantiene, además sus drivers también están incluidos en el sistema.

Si tu elección fue **Ubuntu 20.04** con su sabor original, ya sabes que estás usando el entorno de escritorio **Gnome**. Es interesante que instales **Gnome Tweaks**, con esta aplicación podrás modificar tu escritorio, según tus necesidades.

- Modificar la apariencia de **Ubuntu 20.04 Desktop**
- Cambiar las fuentes predeterminadas
- Mover los botones de ventana
- Gestionar eficazmente todas las extensiones de **GNOME**
- Habilitar / Deshabilitar los iconos del escritorio

Si no te convence la apariencia visual del nuevo Gnome, siempre puedes volver a **Gnome Classic**. Revisa este [artículo específico](#) sobre el tema.

Ubuntu reconoce la mayoría de formatos (de dispositivos), aun así es recomendable que amplíes la lista.

```
sudo apt install exfat-fuse exfat-utils hfsplus hfsutils ntfs-3g
```

Por defecto, **Ubuntu 20.04 Focal Fossa** no instala los codecs restringidos, las **ttf-mscorefonts** y un largo etcétera de herramientas que seguro necesitaras en un futuro. La instalación de este paquete también puede demorar dependiendo de la velocidad de tu red, paciencia.

```
sudo apt install ubuntu-restricted-extras
```

Es necesario aceptar las condiciones para poder instalar las **TTF-mscorefonts**.

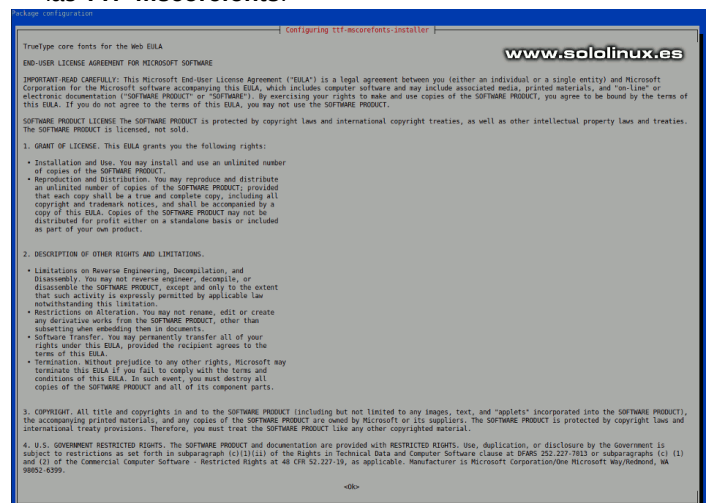


Imagen 3. Instalar las ttf mscorefonts de Microsoft

Instalamos **libdvdcss2**. Nos lanzara unas ventanas de aviso, responde «YES» a todas. Después reconfiguramos y recargamos la cache de las fuentes para concluir la instalación de codecs y fuentes.

```
sudo apt-get install libdvdcss2
sudo dpkg-reconfigure libdvd-pkg
sudo fc-cache -fv
```

Por defecto, el firewall **UFW** ya viene instalado en Ubuntu. No todas las versiones desktop incluyen su GUI para interactuar con el de manera simple y rápida. Instalamos.

```
sudo apt install gufw
```



Imagen 4. Instalar la GUI de UFW

Si prefieres manejar el **firewall** desde la terminal, puedes revisar este [artículo anterior](#).

El **comando aptitude**, y la herramienta de instalación de **paquetes deb**, pueden sacarte de más de un apuro. Las instalamos las dos.

```
sudo apt install aptitude
sudo apt install gdebi-core gdebi
```

El veterano administrador de **paquetes Synaptic**, tampoco puede faltar en nuestro flamante Ubuntu 20.04.

```
sudo apt install synaptic
```

Algunos descompresores interesantes.

```
sudo apt install p7zip-full p7zip-rar unrar
```

Ahora mismo nuestro sistema tiene muchos archivos que son innecesarios, los borramos.

```
sudo apt autoremove
```

Si estás usando un sistema portátil, es interesante instalar la **herramienta tlp**. Ahorrarás batería y evitaras sobrecalentamientos, así que la instalamos y habilitamos.

```
sudo apt install tlp tlp-rdw
sudo tlp start
```

La gran mayoría de impresoras son soportadas por nuestro Ubuntu 20.04, por si acaso ampliamos el listado de drivers.

```
sudo apt install printer-driver-all
```

La utilidad Transmission para descargar **archivos torrent**, no siempre reconoce los **magnet**. Ejecuta el siguiente comando para evitar errores.

```
xdg-mime default transmission-gtk.desktop x-scheme-handler/magnet
```

Transmission funciona realmente bien y es muy ligero, pero existe un cliente que es buenísimo y gráficamente nos aporta muchísima más información. Está claro que hablamos de **qBittorrent** (Recomendado).

Agregamos su repositorio oficial y lo instalamos.

```
sudo add-apt-repository
ppa:qbittorrent-team/qbittorrent-stable
sudo apt update && sudo apt-get install qbittorrent
```

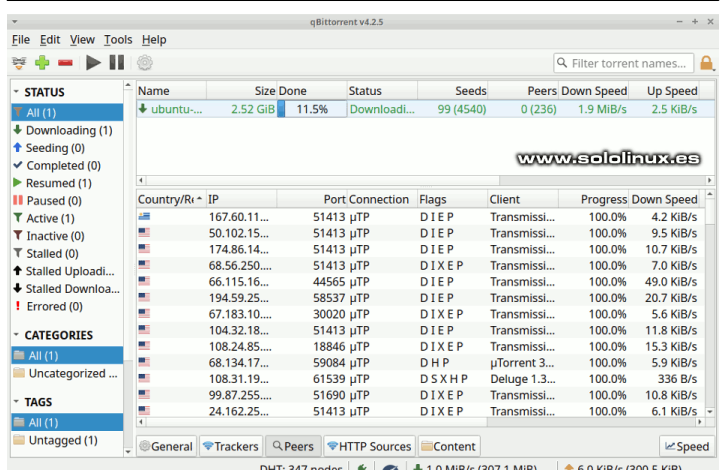


Imagen 5. Instalar qBittorrent en Ubuntu 20.04

Como alternativa puedes instalar otros navegadores web, por ejemplo **Google Chrome** o **Chromium**.

```
# Instalar Google Chrome
wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb
sudo gdebi google-chrome-stable_current_amd64.deb
```

```
# Instalar Chromium
sudo apt install -y chromium-browser
```

Otras herramientas...

```
sudo apt install bluefish
sudo apt install bleachbit
sudo apt install htop
sudo apt install smplayer
etc...
```

Una vez hayas terminado, es conveniente limpiar de nuevo tu Ubuntu 20.04 y reiniciar el sistema.

```
sudo apt autoremove
sudo reboot
```


Los nuevos Lenovo ThinkPad con Fedora 32 están a punto

Uff... aun recuerdo hace más de 25 años lo que me costo adquirir mi primer **IBM ThinkPad**, madre mía. Hablamos de un ThinkPad 300, duro donde los haya, pero suave y fino internamente como el terciopelo.

Cuantas veces fue al suelo con los juegos de los niños entonces pequeños, nunca le paso nada. Ni un misero rasguño... eso me hizo adicto a los IBM, eran como rocas pero su tacto externo era suave. De repente todo cambio... una empresa china se hizo con la división de computadoras personales de **IBM**.

Pasados los años todo cambio, mi colección de IBM ya no eran IBM, eran **Lenovo**. Supongo que por desconfianza del país asiático deje de un lado esa marca.

A pesar de que los comienzos de lenovo no fueron buenos (incluyendo sus más y sus menos con la comunidad GNU). Por fin retomaron la senda de IBM, que aunque desarrollaba su propio sistema participaba activamente con el **opensource**.

Los nuevos Lenovo ThinkPad con Fedora 32

Hoy en día, Lenovo es uno de los mayores fabricantes de ordenadores personales y portátiles del mundo. Comprometidos con **linux**, acaban de anunciar que en muchos modelos fabricados, incluyendo los ThinkPad P1 , ThinkPad P53 y ThinkPad X1 (Gen 8) podrás decidir que sistema operativo quieres por defecto. Fedora o Windows.

Parece una tontería, pero hablamos de un paso adelante para la comunidad linux. Recuerda que al ser una instalación oficial, obtienes exactamente el mismo soporte que sus sistemas basados en Windows, y si algo no te funciona puedes contactar con ellos y te enviaran un driver personalizado. Parece mentira verdad?, pues no lo es.

Esta revolución está próxima, será a partir de **Fedora 32** que se lanzara el próximo 28 de abril del año en curso (2020). Hoy... **Lenovo ThinkPad** ha dado una lección al resto de fabricantes y aquí inserto su video promocional.

<https://youtu.be/dnnpY1VZGW0>

“Esto y el compromiso de Lenovo para solucionar problemas formando parte de la comunidad significa que todos se benefician de su trabajo de la ingeniería de Linux en el verdadero espíritu de colaboración de código abierto”

Muchos diréis que HP también ofrece maquinas con Ubuntu, yo respondo que si, pero con un soporte casi nulo y con poca implicación.

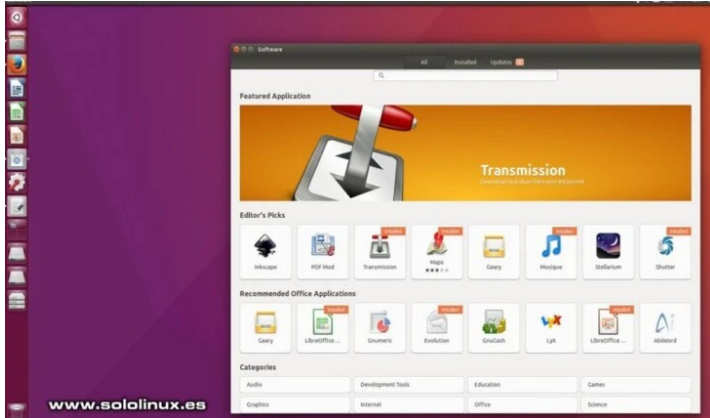
Otros diréis y Dell, que?, umm... Dell es un ensamblador, grande pero ensamblador. Nada más, no es Lenovo ThinkPad.

¿Te interesa colaborar
con SoloLinux?

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

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

Instalar Unity Desktop Environment en Ubuntu 20.04 LTS



Ubuntu 17.04 Zesty fue el último lanzamiento de **Canonical** con el **entorno de escritorio Unity** por defecto.

Odiado por muchos y deseado por otros, es evidente que no dejó indiferente a nadie. Personalmente nunca me gustó, pero debo reconocer que en máquinas con pantallas muy pequeñas era bastante cómodo e intuitivo, aun así... nunca me convenció totalmente.

Su facilidad para instalar extensiones y personalizar el escritorio, le hizo tener una legión de seguidores acérrimos que sufrieron una gran frustración al anunciarse que Ubuntu regresaba a **Gnome**. Por suerte para ellos, una comprometida comunidad retomó el proyecto Unity; además de forma muy activa. Puedes visitar su [Github](#).

En el artículo de hoy (para los nostálgicos), vemos como **instalar Unity Desktop** en nuestro flamante **Ubuntu 20.04 LTS Focal Fossa**.

Instalar Unity Desktop en Ubuntu 20.04

El escritorio **Unity 7** viene en los repositorios oficiales de Ubuntu 20.04. Su instalación es tan sencilla como ejecutar los comandos propuestos y seguir los pasos indicados a continuación.

```
sudo apt update
sudo apt install ubuntu-unity-desktop
```

Durante el proceso de instalación, debes elegir el administrador de pantalla predeterminado. Tienes dos opciones, **lightdm** para el inicio de sesión de **Unity** o **gdm** para el escritorio **GNOME**. En nuestro caso seleccionamos **lightdm**.



Imagen 7. Seleccionamos lightdm para Unity Desktop

Cuando termine la instalación debes reiniciar el sistema.

```
sudo shutdown -r now
```

Nos aparece la pantalla de inicio de sesión de Unity.

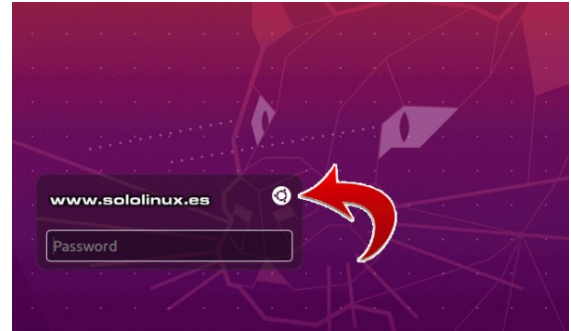


Imagen 8. Iniciar sesión en Unity Desktop

Si no tienes una instalación nueva, actualizaste a Ubuntu 20.04 desde una versión anterior, debes reconfigurar.

```
sudo dpkg-reconfigure lightdm
```

Tal como te indica la flecha en la imagen anterior, si pulsas en el icono de Ubuntu podrás seleccionar el inicio de sesión con otro escritorio.

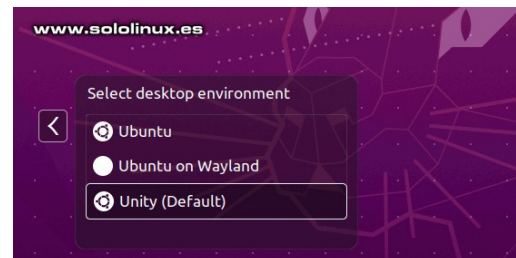


Imagen 9. Iniciar sesión en otros escritorios

Nuestro Ubuntu 20.04 con Unity Desktop instalado.



Imagen 10. Instalar Unity desktop en Ubuntu 20.04

Puedes verificar la versión de Unity con el siguiente comando.

```
unity --version
```

Si quieres personalizar tu nuevo escritorio te recomiendo instalar **Unity-tweak-tool**.

```
sudo apt install unity-tweak-tool
```

Como último aporte de este artículo desinstalamos Gnome. Este proceso no es obligatorio y es bajo tu responsabilidad.

```
sudo apt remove gnome-shell
sudo apt autoremove
```

El nuevo Tails 4.6 ya está listo para su descarga



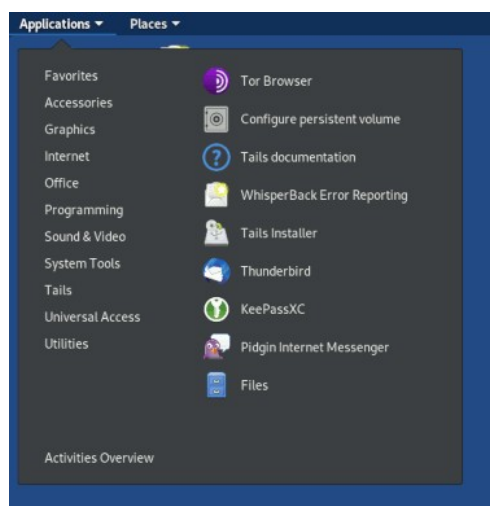
La **distribución linux** especializada en anonimización en la red **Tails 4.6 (The Amnesic Incognito Live System)**, acaba de lanzar su última versión, la 4.6.

Basada en **Debian** y diseñada para proporcionar acceso anónimo a la red. **Tor** ofrece la salida anónima de Tails en todas sus conexiones, exceptuando el tráfico que circula a través de la **red Tor**.

El nuevo Tails 4.6 ya está listo

La nueva versión de Tails, basada en **libu2f-udev**, admite la autenticación universal de dos factores (U2F) mediante llaves de seguridad USB. También se ha actualizado la composición del menú, incluyendo el configurador de la partición persistente, el instalador, su documentación y la herramienta con la que puedes enviar avisos sobre bugs o problemas.

El emulador de terminal se elimina de la lista. Se actualiza a Tor Browser 9.0.10, Thunderbird 68.7.0, Git 1: 2.11, Node.js 10.19.0, OpenLDAP 2.4.47, OpenSSL 1.1.1d, ReportLab 3.5.13, WebKitGTK 2.26.4, etc...



Otras actualizaciones a destacar son... Firefox 68.8.0 ESR, que ha corregido 14 vulnerabilidades, de las cuales 10 (CVE-2020-12387, CVE-2020-12388 y 8 bajo CVE-2020-12395) se marcaron como críticas y podrían permitir la ejecución de código maligno. NoScript también se actualizó a la versión 11.0.25, **Openssl** a la versión 1.1.1g.

Los enlaces de descarga oficiales, son:

- [Descargar Tails 4.6 por torrent](#)
- [Descarga directa de Tails 4.6](#)

Imagen 11. Menú de aplicaciones en Tails 4.6

Bashtop - El monitor de linux escrito en bash

Bashtop es un monitor de recursos de código abierto para Linux. Al igual que **top** o **htop**, nos permite monitorear el sistema y mostrar el uso y estadísticas del procesador, la red, los procesos, memoria y discos.

Tiene un menú fácil de usar además de teclas receptivas que facilitan el trabajo. Permite filtrar procesos y escalar la interfaz de usuario. Destacamos que es compatible con SIGTERM, SIGKILL y SIGINT. También nos permite cambiar entre sus opciones de manera sencilla.

Bashtop - El monitor de linux escrito en bash

Antes de continuar, debo indicar que es necesario tener bash 4.4 o superior. Puedes verificar la versión con el siguiente comando.

```
apt-cache policy bash
```

```
root@185:~# apt-cache policy bash
bash:
  Installed: 4.4-5
  Candidate: 4.4-5
  Version table:
 *** 4.4-5 500
    500 http://ftp.debian.org/debian
 stretch/main amd64 Packages
    100 /var/lib/dpkg/status
```

Si tienes una versión anterior, al ejecutar el script lanzará el siguiente error.

```
ERROR: Bash 4.4 or later is required (you are using
Bash 4.3).
```

Consider upgrading your distribution to get a more recent Bash version.

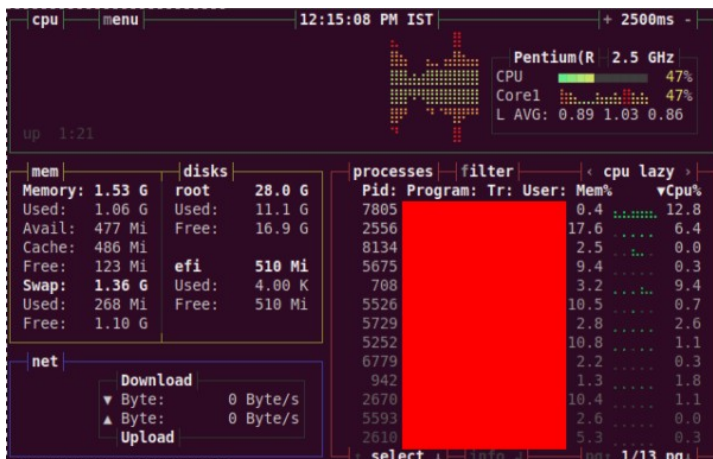


Imagen 12. Monitor de linux Bashtop

La instalación del monitor de linux **Bashtop** no tiene ninguna dificultad. Vemos como instalar la herramienta en diversas distribuciones linux.

En Ubuntu, Linux Mint y derivados:

```
sudo add-apt-repository ppa:bashtop-monitor/bashtop
sudo apt update
sudo apt install bashtop
```

Fedora:

```
sudo dnf install bashtop
```

BASHTOP

www.sololinux.es

BASHTOP

El monitor de linux escrito en bash

En Debian 10 / Debian 9:

```
### Debian 10
sudo echo "deb http://packages.azlux.fr/debian/
buster main" | sudo tee
/etc/apt/sources.list.d/azlux.list
sudo wget -qO - https://azlux.fr/repo.gpg.key | sudo
apt-key add -
sudo apt update
sudo apt install bashtop
### Debian 9
sudo echo "deb http://packages.azlux.fr/debian/
stretch main" | sudo tee
/etc/apt/sources.list.d/azlux.list
sudo wget -qO - https://azlux.fr/repo.gpg.key | sudo
apt-key add -
sudo apt update
sudo apt install bashtop
```

CentOS 8 y derivados:

CentOS 7 no tiene versión bash 4.4 disponible, por tanto no es compatible con esa versión.

```
sudo dnf config-manager --set-enabled PowerTools
sudo dnf install epel-release
sudo dnf install bashtop
```

Red Hat 8:

Al igual que con CentOS, el monitor de linux Bashtop tampoco es compatible con las versiones anteriores de Red Hat (RHEL).

```
ARCH=$( /bin/arch )
subscription-manager repos --enable
"codeready-builder-for-rhel-8-${ARCH}-rpms"
sudo dnf install epel-release
sudo dnf install bashtop
```

Ejecutar Bashtop

Una vez instalada la herramienta, tan solo tienes que ejecutar el siguiente comando.

```
bashtop
```

Script bash Bashtop

Si eres de los que prefiere tenerlo todo bajo control, seguro que prefieres crear tu propio script bash. Puedes copiar el código desde su [página oficial de Github](#).

Unir imágenes con el comando convert de Image Magik

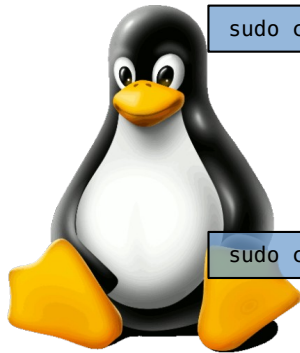
El comando **convert** forma parte de la mejor suite de manipulación de imágenes en linux, es evidente que hablamos de **Image Magick**.

Convert tienes muchas opciones (las veremos en un próximo post), pero una de las que más me llama la atención es generar imágenes en secuencia. En el artículo de hoy vemos como unir imágenes (dos) en una sola, con forma horizontal y vertical.

Unir imágenes con el comando convert

En nuestro caso de ejemplo tenemos dos imágenes, `corazon.png` y `tux.png`.

Si queremos unir dos imágenes horizontalmente, usamos la opción «**+append**».



```
sudo convert +append imagen1.png imagen2.png imagen-horizontal.png
```

EJEMPLO

```
sudo convert +append corazon.png tux.png salida-horizontal.png
```

Imagen 13. Imagen horizontal con el comando convert

Como puedes ver en la imagen anterior, las dos imágenes se han unido en una sola (horizontal). Ahora vemos como hacer lo mismo pero obteniendo un resultado vertical (una encima de otra), para ello utilizamos la opción «**-append**».

```
sudo convert -append imagen1.png imagen2.png imagen-vertical.png
```

EJEMPLO

```
sudo convert -append corazon.png tux.png salida-vertical.png
```

Como puedes ver, es un pequeño pero interesante tip que te puede ahorrar trabajo y tiempo.

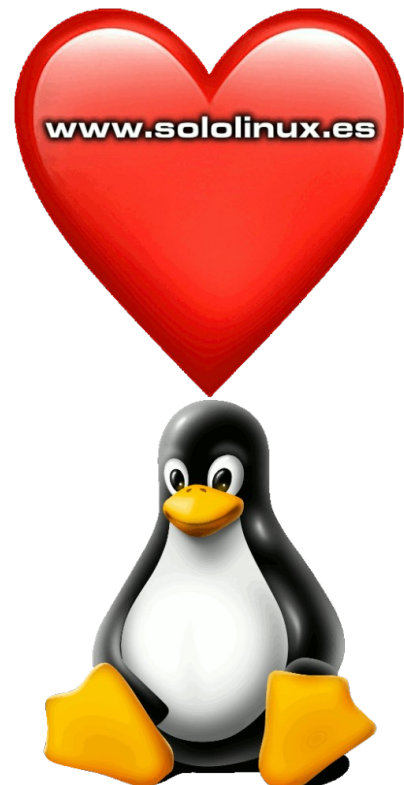


Imagen 14. Imagen vertical con el comando convert

Instalar LMMS Linux Multimedia Studio en Ubuntu 20.04 LTS

LMMS (Linux MultiMedia Studio) es una estación de trabajo de audio digital **open source**, que nos permite crear música con nuestra computadora.

Esta aplicación es de nivel profesional y está disponible de forma gratuita para **Linux, BSD, Microsoft Windows y Mac OS X**. Además de producir música, también puedes sintetizar audios y organizar muestras mediante su **teclado MIDI**. Destacamos que LMMS está disponible en 15 idiomas, incluido el Español.

Características principales de LMMS:

- *Crear tu música en Linux, Windows y macOS.*
- *Compón canciones, crea secuencias, mezcla y automatiza desde una interfaz fácil de usar.*
- *Toca las notas musicales con un controlador MIDI o desde el teclado de tu máquina.*
- *Consolida las pistas de instrumento usando el Editor de Ritmo+Bajo.*
- *Ajusta los patrones, notas, acordes y melodías con el Editor en Piano Roll.*
- *Automatización completa basada en pistas definidas por el usuario y fuentes de automatización controladas por computadora*
- *Importa archivos externos MIDI y otros proyectos de Hydrogen.*
- *Soporta complementos LADSPA.*
- *Soporta complementos de efectos VST®.*
- *Tiene su propio compresor incorporado, limitador, retardo, reverberación, distorsión e intensificador de bajos.*
- *Incluye ecualizadores gráficos y paramétricos.*
- *Visualizador/analizador de espectro.*

En este artículo vamos a **instalar LMMS en Ubuntu 20.04 y Ubuntu 18.04**, ya que nos centramos en la última versión del software, la 1.2.1 (las anteriores no tienen nueva versión vía ppa).



Imagen 15. Instalar LMMS en Ubuntu



Instalar LMMS en Ubuntu 20.04 LTS

Antes de comenzar debo avisarte que LMMS lo tenemos en los repositorios oficiales de Ubuntu 20.04, pero los de Ubuntu 18.04 no lo tienen, deberemos agregar su repositorio no oficial.

LMMS en Ubuntu 20.04

La instalación en Focal es tan simple como ejecutar los siguientes comandos.

```
sudo apt update
sudo apt install lmms
```

Puedes lanzar la herramienta desde tu menú de aplicaciones.

LMMS en Ubuntu 18.04

La versión LMMS que tenemos en los repositorios oficiales es la 1.1.3.7, como nosotros queremos la última agregamos el repositorio ubuntuhandbook1.

```
sudo add-apt-repository ppa:ubuntuhandbook1/lmms
```

Actualiza e instala.

```
sudo apt update
sudo apt install lmms
```

Busca la aplicación en el menú para ejecutarla.

Desinstalar LMMS en Ubuntu

Para desinstalar LMMS ejecutamos...

```
sudo apt remove --autoremove lmms
```

Si agregaste el repositorio necesario en Ubuntu 18.04 o derivados, también lo puedes quitar.

```
sudo add-apt-repository --remove
ppa:ubuntuhandbook1/lmms
```


Como instalar aMule en Ubuntu 20.04 LTS Focal Fossa

aMule es una aplicación de intercambio de archivos **p2p** libre y gratuito. Fue la herramienta de descarga más utilizada no hace muchos años atrás, además destacaba por ser compatible con las redes **eDonkey** como con **Kademlia**.

Hoy en día está en desuso. Como sucede con otros apartados tecnológicos, protocolos más modernos y seguros como el **BitTorrent** le han pasado por encima (y con motivo). En este mini artículo y ante la duda de un lector de **sololinux.es**, explicaremos **como instalar aMule en Ubuntu 20.04 LTS Focal Fossa**.

Como instalar aMule en Ubuntu 20.04

A pesar de que **aMule** podría haber seguido evolucionando y que aún mantiene [un foro bastante activo](#), su desarrollo parece estar abandonado. La última versión es de septiembre del 2016 y eso que muchos usuarios aún lo usan, sobre todo para descargar cosas viejas.

La nueva versión de Ubuntu ya no ofrece **aMule** en sus **ppa** y tampoco soporte, pero no te preocupes... en **sololinux** hemos verificado que la última versión (disponible en Ubuntu 19.10), sigue siendo funcional en **Ubuntu 20.04 LTS Focal Fossa**.

Así que ni corto ni perezoso, he subido los archivos necesarios a **sololinux** para que puedas instalar sin ningún problema aMule en la versión más reciente de Ubuntu. Más que nada, porque todos sabemos que el soporte de Ubuntu 19.10 termina en julio del 2020 y con él, también desaparecerán los archivos.

Sin más preámbulos descargamos los archivos y los instalamos, ya veras que simple.

```
wget https://www.sololinux.es/public/amule/amule-common_2.3.2-6_all.deb
wget https://www.sololinux.es/public/amule/amule_2.3.2-6_amd64.deb
wget https://www.sololinux.es/public/amule/libwxgtk3.0-0v5_3.0.4+dfsg-12_amd64.deb
```

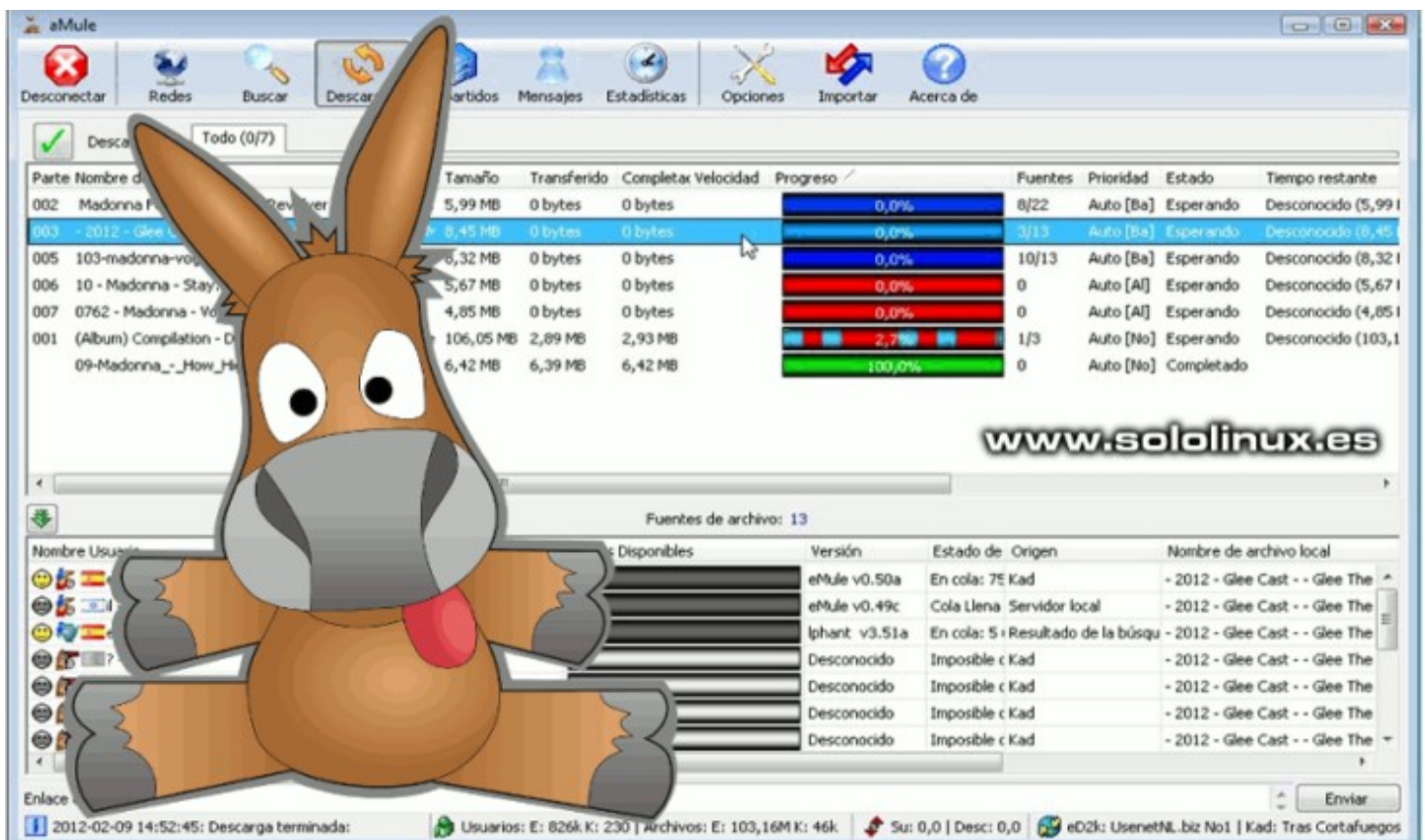
Una vez tenemos los archivos en nuestro sistema, los instalamos.

```
sudo apt install ./*.deb
```

También puedes instalarlos de manera gráfica haciendo click en las descargas, te recomiendo **gdebi**.

```
sudo apt install gdebi-core gdebi
```

Bien... ya tienes instalado tu aMule en Ubuntu 20.04, dale a descargar archivos.



Instalar los drivers de impresoras Canon en Ubuntu 20.04



Los drivers de las impresoras y multi-funciones, siempre han sido un quebradero de cabeza en linux. Sin embargo si tienes una máquina Canon, hay una manera fácil de instalar sus drivers, el software **ScanGear MP2**, **Cnijfilter2** y **UFR II**.

Con **Ubuntu 20.04 LTS** y **Canon** estamos de suerte, el usuario de **launchpad Thierry Ordissimo** mantiene un repositorio con los drivers oficiales de Canon, siempre actualizado. En este artículo vemos como instalar estos controladores para tu impresora o multi-función Canon.

Drivers de impresoras Canon en Ubuntu 20.04

Lo primero que hacemos es agregar el ppa y actualizar.

```
sudo add-apt-repository ppa:thierry-f/fork-michael-gruz
sudo apt update
```

Ahora dependiendo de tus necesidades instalas el driver que necesites, recuerda que dependiendo del dispositivo tendrás que instalar dos. Asegúrate en el listado que vimos anteriormente.

ScanGear MP2

```
sudo apt install scangearmp2
```

Cnijfilter2 (IJ printer driver)

```
sudo apt install cnijfilter2
```

UFR II (LIPSLX for Canon LBP/iR)

En este caso instala alguno de los drivers propuestos (solo uno).

```
sudo apt install cnrdvcups-lipslx
sudo apt install cnrdvcups-ufr2-uk
sudo apt install cnrdvcups-ufr2-us
```

Una vez instales los drivers apropiados se recomienda reiniciar el sistema.

Listado de máquinas compatibles a día de hoy (8-mayo-2020).

Scangearmp2

G6000 series, G6080 series, TS5300 series, TS5380 series, TS6300 series, TS6380 series, TS7330 series, TS8300 series, TS8380 series, TS8330 series, XK60 series, TS6330 series, TS3300 series, E3300 series

Cnijfilter2

E3300, G5000, G5080, G6000, G6080, GM2000, GM2080, TR703, TS3300, TS5300, TS5380, TS6300, TS6330, TS6380, TS700, TS708, TS7330, TR800, TS8330, TS8380, XK60

UFR II

MF540 Series, MF642C/643C/644C, IR-ADV C7565 III, LBP622C/623C, MF645C, IR-ADV C5550/5560 III, IR2625/2630, LBP227, IPR C165, LBP323/324/325, WG7000 Series, IR2635/2645, MF745C/746C, IR-ADV 4545/4551, LBP8750, LBP3980M, LBP253, D1300 Series, MF8300 Series, LBP7600C, IPR C1PLUS, IR2422, IR1018/1022/1023, IR-ADV 8505, MF8500C Series, LBP251, MF6100 Series, IR-ADV C3320, LBP8900, LBP310/311/312, MF520 Series, IR-ADV C2020/2030, MF4700 Series, IR-ADV 6255/6265, MF4500w Series, IR C5185, MF420 Series, MF720C Series, MF633C/635C, D530/D560, L90/L160/L230, IPR C850, MF240 Series, MF7200 Series, MF4500 Series, MF4500 Series, MF632C/634C, IR1020/1024/1025, LBP6670, MF240 Series, IPR C1, LC600 Series, IR5570/IR6570, MF3010, IR5570/IR6570, MF417, MF720C Series, IR-ADV C5045/5051, IR2420, IR7086-7105, MF4320-4350, MF631C, IR-ADV 6275, LBP841C, IR-ADV C2025, MF4800 Series, LBP443, IR C2880/C3380, MF8300 Series, IR-ADV 8585/8595, IR-ADV C5235/5240, LBP712C, LBP441e, LBP653C/654C, MF631C, MF8300C Series, MF510 Series, MF4400w Series, LBP252 CARPS2, IR2422, MF633C/635C, LBP842C, D400-450, IR-ADV C7280, IR-ADV 4225/4235, L90/L160/L230, MF8400, LBP8780, IR3570/IR4570, MF4360-4390, IR C5180, IR-ADV 4045/4051, IR2002/2202, MF410 Series, IR-ADV C9270/9280, MF4010 Series, IR C5880/C6880, LBP9510C, LBP9500, MF220 Series, IR-ADV C2220/2230, MF6500 Series, LBP8730, IR C1325/1335, IR C5870, IR C5180, MF4700 Series, MF9200 Series, MF8000 Series, LBP7750C/5460, IR-ADV C250/350, IR8070, IR3030, MF632C/634C, IR3235/IR3245, IR3235/IR3245, IR-ADV 6055/6065, IR-ADV 400/500, MF6800 Series, IPR C1, LBP8720/8710, IR1018/1022/1023, IR-ADV C3520, IPR C750, MF4700 Series, IPR C750, MF4010 Series, IR C1021/C1022, IR1730/1740/1750, IR1020/1024/1025, MF735C, IR C2110, D500 Series, LBP8630/8620/8610, IR C5870, IR-ADV C256/356, IR2020, IR-ADV 8205, MF8300C Series, MF220 Series, IR-ADV 4045, IR105PLUS, MF4800 Series, IR 3180C EUR, MF3010, MF240 Series, MF6100 Series, IR-ADV 6055/6065, D460-490, IR2002/2202, IR-ADV 6275, MF4360-4390, MF6500 Series, LBP253, MF620C Series, IR-ADV C3320, IR-ADV 8085/8095, L410 Series, LBP710C, MF8200C Series, MF110/910 Series, L1000, MF732C/734C/735C, IR-ADV C9060/9070, IR2230, MF4500 Series, D530/D560, IR-ADV C255/355, LBP312, IR-ADV 4025/4035, IR-ADV 8585/8595, IR2525/2530, IR-ADV C2225, IR85PLUS, imageRUNNER1133 Series, MF210 Series, LBP3370, MF5900 Series, IR-ADV 4245, LBP6710, IR2020, MF8400, MF731C/733C, MF4800 Series, MF210 Series, IR-ADV 4245/4251, MF8200C Series, IR-ADV 4545/4551, LBP5360, IR C1325/1335, MF110/910 Series, MF7200 Series, IR-ADV C3325/3330, IPR C600, MF731C/733C, IR C2550, LBP213, IR-ADV C7260/7270, MF6700 Series, IPR C700/800, MF260 Series, IR-ADV C7055/7065, MF9200 Series, IR-ADV C2218, IR-ADV 6555/6565, IR-ADV C5535/5540, MF230 Series, MF4400 Series, LBP5900, LBP841C, MF5800 Series, IR-ADV C5030/5035, MF732C/734C/735C, IR2525/2530, L90/L160/L230, IR-ADV 4025/4035, LBP9650C, IR-ADV 6075, LBP612C/613C, IR-ADV 8205, IR-ADV C9270/9280, IPR C1PLUS, LBP6650, LBP6680/3480, IR-ADV 4525/4535, MF8500C Series, IPR C850, IR2520, IR-ADV C351, IR2420, IPR C850, IR-ADV C3320L, IR-ADV C2020/2030, IR-ADV C7580/7570, IR-ADV C5535/5540, IR C1225, IR-ADV 8085/8095, L100/L150/L170, MF4100 Series, IR-ADV C9065, IR C3180, LBP5400, IR C5880/C6880, IR-ADV C9075, IR-ADV 4545, MF4200 Series, MF9100/9300, IR-ADV C2020L/2030L, MF4500w Series, IR2004/2204, IR C1028/C1030, IR8570N, LBP5970/5975, IR2520, LBP9950C/9900C, MF8300C Series, L190/L410 Series, LBP215, MF810/820, IR-ADV C255/355, LBP310/311/312, IR3025, LBP9520C, LBP6600, IR-ADV C5250/5255, LC800/L3000, LBP843C, D1500 Series, LBP9660C, MF4200 Series, IR-ADV C7580/7570, LBP7780C/5480, IR-ADV C3320/3330, MF4100 Series, IR3035/IR3045, MF8500C Series, LBP653C/654C, IR-ADV C2020/2030, LBP651C, IR2016, MF7400 Series, MF5900 Series, IR-ADV C7580/7570, MF210 Series, MF8000C Series, D1100, IR-ADV C355, LBP9600C/9500C, MF4400 Series, IR-ADV C5235/5240, LBP3800, MF4400 Series, LBP712C, LBP611C, MF720C Series, IR2022/2025/2030, LBP4500, LBP843C, LBP6700, IPR C650, LBP214, LBP161/162, MF8000C Series, LBP3920/3970, IR9070, MF5800 Series, MF260 Series, LBP7660C, IPR C750, LBP6780/3580, IR3230, IR C6870, IR2318/2320, LBP652C/654C, IR C1028/C1030, MF731C/733C, IR5075, IR-ADV C5045/5051, IR-ADV C2220/2230, IR2016, MF220 Series, D570, IR-ADV C256/356, IR-ADV C3525/3530, MF8400, IR C1021/C1022, IR2270/IR2870, LBP215, D400-450, MF520 Series, IR-ADV C356, LBP611C, LBP351/352, IR-ADV 6255/6265, IPR C700/800, IR C2580, IR-ADV 8505, D500 Series, LBP441, LBP3910/3930, IR-ADV 8505, IR C3000 Series, IR-ADV C9065/9075, IR-ADV 8285/8295, IR-ADV 6575, IR2535/2545, LBP612C/613C, IR2270/IR2870, LBP161/162, MF4600 Series, MF632C/634C, IR C4080/C4580, MF4360-4390, MF620C Series, MF620C Series, IR-ADV 4525/4535, LBP351/352, IR-ADV C5550/5560, IR-ADV 6575, IR-ADV C3325/3330, LBP214, IR3225, MF8200C Series, IR3530, IR-ADV 6555/6565, IR-ADV C7260/7270, LBP5910M, LBP3360, MF9100/9300, LBP3980, IR-ADV 8285/8295, IR3025, IR C5185, IR5075, LBP211/212, IR C2380/2550, L100/L150/L170, LBP251, IR-ADV 8105, MF8300 Series, IR-ADV 6555/6565, IR7086-7105, LBP7700C, IR1435, MF4600 Series, MF6500 Series, D1300 Series, IR-ADV 6560, MF230 Series, IR 5880C/6880C EUR, IR C4080, LBP3410, IR C3880/C4580, MF4100 Series, D1100, LBP3700, IR C3000 Series, IR5055/IR5065, IR2830, IR3035/IR3045, IR1435, MF8000C Series, MF420 Series, LBP251 CARPS2, IR2535/2545, IR1051, LBP253, IPR C650, MF9200 Series, D460-490, MF520 Series, IR-ADV C5030/5035, IR-ADV C3520/3530, IR-ADV 8585/8595, IR-ADV 4525/4535, IR-ADV C5250/5255, LBP712C, IR-ADV C5535/5540, IR2018, IR3570/IR4570, IR C3080/3480/3580, IPR C700/800, LBP6750/3560, MF7100 Series, IR2230, MF410 Series, IR C3080/C3580, MF8000 Series, IR-ADV C7565, MF4320-4350, IR-ADV C7565, LBP6670, LBP6650/3470, MF4200 Series, IR C6870, D400-450, IR C2880/C3380, MF810/820, IR5055/IR5065, IR-ADV C7565, L190/L410 Series, LBP3460, LBP612C, IR-ADV 8105, IR C3000 Series, IR2318/2320, LBP7660C, LBP252, IR7270N, LBP252, LC800/L3000, LBP5960, IR2006/2206, MF510 Series, L250, LBP442, IR1730/1740/1750, LBP5910, IR2018, IR-ADV C7055/7065, IR-ADV C5550/5560, IR-ADV 6575, MF510 Series, IR-ADV 4225/4235, D570, MF7400 Series, LBP7680C/5280, MF230 Series, IR3225, IR4530, IR-ADV C3525/3530, IR-ADV 6075, IR-ADV C350, IR-ADV C5550/5560, IR-ADV C475 III, LBP225, MF741C/743C, IR-ADV 4545/4551 III, LBP223, IR-ADV 715 III, IR-ADV C3520 III, LBP226, IR-ADV 4525/4535 III, LBP664C, IR-ADV 8505 III, IR-ADV C256/356 III, MF4400 Series, IR-ADV 8585/8595 III, IR-ADV 615, IR-ADV 6575 III, LBP228, IR-ADV C7580/7570 III, IR1643I, IR1643P, LBP621C, IR-ADV 525 III, IPR C710/810/910, LBP852C, IR-ADV C3525/3530 III, IR-ADV 525, LBP222, MF641C, D1600 Series, WG7000Z Series, IR-ADV 6555/6565 III, IR-ADV 715, LBP663C, IR-ADV C5535/5540 III, MF742C/744C, IR-ADV 615 III

Crear directorios y subdirectorios desde un archivo de texto

En linux existe un pequeño truco, con el cual puedes crear tantos directorios o subdirectorios como quieras. Muy útil cuando tienes que replicar varios sistemas, es muy rápido y sencillo de implantar.

Existen varios comandos capaces de realizar esta operación, en este artículo nos decantamos por el **comando xargs**, que es capaz de convertir entradas de argumentos estándar en un comando ejecutable.

Crear directorios y subdirectorios desde un archivo de texto

Lo que haremos será crear un archivo de texto, con todos los directorios y subdirectorios que necesitemos (los subdirectorios los separamos con una barra), algo similar a nuestro ejemplo.

- servidores
- cms
- comandos/linux
- comandos/bash
- comandos/bsd
- editores/vim
- editores/nano
- editores/ide/eclipse
- documentos/facturacion
- documentos/manuales
- documentos/varios
- linux
- unix
- bsd

Guardas el archivo. Nosotros lo nombramos «directorios.txt».

Este archivo de texto lo guardamos en la ruta donde queremos generar las nuevas carpetas y subcarpetas. Vale... pues ya lo tenemos todo preparado, ahora es tan simple como ejecutar el siguiente comando (con el nombre de tu archivo).

```
sudo xargs -I{} mkdir -p "{}" < directorios.txt
```

Al acceder a la ruta indicada desde nuestro entorno de escritorio, vemos que se han creado los directorios de forma correcta. Vemos la imagen de ejemplo.

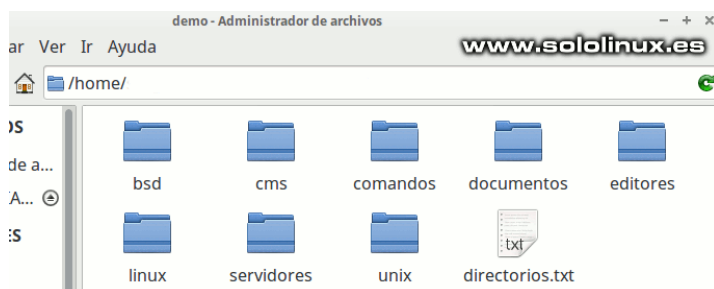


Imagen 16. Crear directorios desde un archivo de texto



También lo podemos verificar desde nuestra terminal linux.

```
sololinux ~/ $ tree -d
.
├── bsd
├── cms
├── comandos
│   ├── bash
│   ├── bsd
│   └── linux
├── documentos
│   ├── facturacion
│   ├── manuales
│   └── varios
├── editores
│   ├── ide
│   │   └── eclipse
│   ├── nano
│   └── vim
├── linux
├── servidores
├── unix
└── 18 directories
```

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



Más procesos de test de escritura aleatorio

Puedes forzar la maquina, metiendo más procesos I/O a la vez. No lo hagas en servidores en producción.

```
sudo fio --name=randread --ioengine=libaio --iodepth=16 --rw=randread --bs=4k --direct=0 --size=512M --
numjobs=4 --runtime=240 --group_reporting
```

```
[root@host ~]# sudo fio --name=randread --ioengine=libaio --iodepth=16 --rw=randread --bs=4k --direct=0 --size=512M --
numjobs=4 --runtime=240 --group_reporting
randread: (g=0): rw=randread, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T) 4096B-4096B, ioengine=libaio, iodepth=16
...
fio-3.7
Starting 4 processes
randread: Laying out IO file (1 file / 512MiB)
randread: Laying out IO file (1 file / 512MiB)
randread: Laying out IO file (1 file / 512MiB)
randread: Laying out IO file (1 file / 512MiB)
Jobs: 4 (f=4): [r(4)][100.0%][r=69.1MiB/s,w=0KiB/s][r=17.7k,w=0 IOPS][eta 00m:00s]
randread: (groupid=0, jobs=4): err= 0: pid=29950: Fri May 8 18:45:11 2020
read: IOPS=18.1k, BW=70.6MiB/s (74.1MB/s)(2048MiB/28989msec)
    slat (usec): min=95, max=2859, avg=219.22, stdev=40.94
    clat (usec): min=5, max=6125, avg=3316.56, stdev=316.30
    lat (usec): min=229, max=6349, avg=3535.92, stdev=335.07
    clat percentiles (usec):
    | 1.00th=[ 1926], 5.00th=[ 3097], 10.00th=[ 3261], 20.00th=[ 3326],
    | 30.00th=[ 3326], 40.00th=[ 3359], 50.00th=[ 3359], 60.00th=[ 3359],
    | 70.00th=[ 3392], 80.00th=[ 3392], 90.00th=[ 3490], 95.00th=[ 3523],
    | 99.00th=[ 3654], 99.50th=[ 4113], 99.90th=[ 5407], 99.95th=[ 5604],
    | 99.99th=[ 5866]
    bw ( KiB/s): min=17520, max=30000, per=25.02%, avg=18102.25, stdev=1772.39, samples=228
    iops        : min= 4380, max= 7500, avg=4525.56, stdev=443.10, samples=228
    lat (usec)   : 10=0.01%, 250=0.01%, 500=0.01%, 750=0.01%, 1000=0.01%
    lat (msec)   : 2=3.07%, 4=96.36%, 10=0.57%
    cpu          : usr=1.06%, sys=4.47%, ctx=524387, majf=0, minf=193
    IO depths    : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=100.0%, 32=0.0%, >=64=0.0%
    submit      : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
    complete    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.1%, 32=0.0%, 64=0.0%, >=64=0.0%
    issued rwts: total=524288,0,0,0 short=0,0,0,0 dropped=0,0,0,0
    latency     : target=0, window=0, percentile=100.00%, depth=16
Run status group 0 (all jobs):
  READ: bw=70.6MiB/s (74.1MB/s), 70.6MiB/s-70.6MiB/s (74.1MB/s-74.1MB/s), io=2048MiB (2147MB), run=28989-28989msec
Disk stats (read/write):
  sda: ios=520363/49, merge=0/78, ticks=110202/0, in_queue=110111, util=97.82%
```

En este post tan solo vimos un par de ejemplos básicos del uso de Fio, pero hemos medido la velocidad de nuestros discos en Linux. La herramienta tiene más opciones, revisa su Github oficial.

¿Te interesa colaborar
con Sololinux?

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

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

Por qué Linux es tan popular en alojamientos web

Por qué Linux es tan popular en alojamientos web. Aunque las estimaciones pueden variar dependiendo de la fuente, **GNU-Linux** es ampliamente reconocido por mantener una mayoría abrumadora de servidores, si lo comparamos con los ridículos (en cantidad) **servers de Windows**.

Esto no es por casualidad, no existe ninguna empresa que se precie que confíe sus datos en **Windows Server**. Ojo!!!, Google utiliza más de 15.000 servidores Linux para servir su contenido, Cloudflare quiso zanjar el asunto y no hace mucho afirmó lo siguiente...

En Cloudflare, solo ejecutamos Linux en nuestros servidores. Utilizamos las características más recientes de Linux, optimizamos el rendimiento y nos ocupamos en gran medida de la resiliencia del DoS.

Es curioso que incluso el mismísimo **Microsoft Windows** mantiene **servidores linux** en sus redes internas (y otras no tan internas).



Por qué Linux es tan popular en alojamientos web Linux vs Windows

Linux ha luchado por hacerse un hueco en el mercado de los ordenadores personales, pero a día de hoy tiene menos del 4% de la cuota de mercado.

Si bien cuando eres nuevo en administración de servidores te puedes sentir frustrado con Linux; es por que nunca lo has usado antes, te resulta completamente desconocido, sobre todo si utilizas Windows o Mac Os.

Sin embargo cuando hablamos de servidores, Linux tiene muchas más ventajas y es bueno conocerlas para entender el por qué.

Los servidores Linux son más asequibles (más baratos), Linux es un sistema operativo de código abierto que puede ser instalado para cualquier propósito de forma gratuita.

Lo mismo ocurre con las tres piezas clave necesarias para servir un sitio web, lo que conocemos como LAMP:

- **Apache:** principal aplicación de servidor web en Linux.
- **MySQL / MariaDB:** la base de datos más usada a nivel mundial.
- **PHP:** el lenguaje de parte del servidor más popular.

Las grandes empresas de alojamiento impulsan Linux porque no hay costos de licencia que cubrir, así que pueden bajar los precios y aspirar a mayores ganancias así como mejores ofertas a sus clientes. El único coste real, es la herramienta del panel de control web, si el anfitrión decide ofrecerla.

Con el hardware pasa lo mismo. Podemos afirmar que Linux es el mejor sistema operativo para máquinas lentas. Es fácil encontrar servidores a precios razonables con procesadores obsoletos,

pero que van muy rápido. Todo lo dicho hasta ahora, nos demuestra el motivo de que **Linux es tan popular**.

La preferencia de la industria

A medida que la industria del hospedaje madura, los estándares se desarrollan continuamente como puedes ver en **codigohosting**. La mayoría de las empresas utilizan hosting linux, ya que ofrecen buena velocidad y **uptime**.

Existen poderosas herramientas de gestión y administración para servidores Linux, lo que facilita enormemente su correcto funcionamiento. Los técnicos que quieren o querían trabajar en la industria del hospedaje tenían que manejar Linux, entonces se establecieron sistemas de certificación más conocidos como **LPIC**. Otras herramientas como cPanel o Plesk fueron desarrolladas para ocultar el sistema operativo de los usuarios, y facilitar su administración.

El uso de un sistema operativo estándar redujo el número de consultas al soporte, y estandarizan el conocimiento y las habilidades de los **administradores**.

Por qué existen servidores Windows

Si Linux es tan genial, ¿por qué algunos alojamientos siguen usando Windows?. Vemos tres buenas razones.

Linux no siempre puede hacer todo lo que los clientes de windows necesitan que haga. Las aplicaciones escritas en lenguajes exclusivos para Windows, como .NET o ASP deben ser ejecutadas en un servidor Windows, si o si. No tienes otra salida.

Como norma general las empresas prefieren **administradores linux**, el motivo es fácil de entender. Adaptarse a Windows es mucho más sencillo, que adaptarse a un servidor Linux.

El ahorro de costos en Linux es enorme, pero a veces existen beneficios prácticos en Windows que amortizan su precio. Es evidente que vale la pena pagar una licencia de **Windows Server**, antes que rediseñar toda una aplicación.

Estabilidad

Un estándar en la industria es el uso de **CageFS**, que es un sistema de archivos virtualizado para cada cliente. Esto quiere decir que si un vecino de hosting tiene un pico de tráfico, no afecta a tu web para nada. Además evita la visualización de archivos entre cada cuenta.

Con CageFS:

- Los usuarios solo tienen acceso a archivos seguros.
- Los usuarios no pueden ver a otros usuarios y ni detectarlos.
- A los usuarios se les prohíbe ver los archivos de configuración del servidor, por ejemplo los de Apache.
- La visión de los procesos está limitada.

Su tecnología:

- Los binarios seguros están disponibles para los usuarios.
- Elimina los accesos de usuario a los scripts SUID.
- Limita el acceso de cliente al sistema de archivos /proc.
- Previene ataques a los enlaces simbólicos.

Dropbear SSH la alternativa ligera a OpenSSH

De la mano de **Matt Johnston**, nos llega **Dropbear SSH**. Una aplicación ya veterana (creada en 2003), que nos proporciona un servidor y cliente compatible con **OpenSSH** en cualquier plataforma POSIX, como **GNU/Linux**, BSD y más.

Sus principales características son:

- Usa muy poca memoria, por tanto es ideal para sistemas muy antiguos.
- Implementa el reenvío X11 y el reenvío del agente de autenticación para clientes OpenSSH.
- Es totalmente compatible con la autenticación de clave pública OpenSSH ~ /.ssh /certified_keys.
- El servidor, el cliente y el generador / convertidor de claves se pueden compilar en un solo binario.
- El modo de salto múltiple, usa el reenvío TCP SSH para hacer un túnel a través de múltiples hosts SSH con un solo comando.

En el artículo de hoy vemos como **instalar Dropbear**, así como su uso.

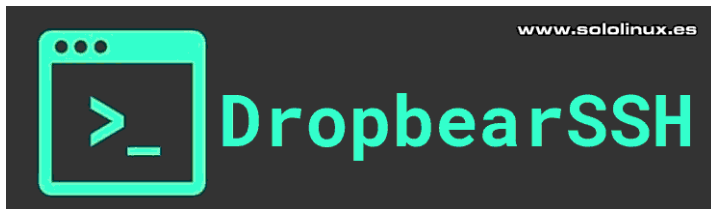


Imagen 17. Instalar Dropbear SSH

Dropbear SSH la alternativa ligera a OpenSSH

Comenzamos instalando la herramienta Dropbear.

Instalar Dropbear SSH

Debian, Ubuntu y derivados:

```
sudo apt install dropbear
```

CentOS y derivados:

```
sudo yum -y install epel-release
sudo yum -y install dropbear
```

Fedora y derivados:

```
sudo dnf -y install dropbear
```

Arch linux, Manjaro y derivados:

```
sudo pacman -S dropbear
# otra opcion
sudo pacman -Rs dropbear
```

FreeBSD, Alpine y derivados:

```
sudo apk add dropbear
```

Detener OpenSSH

Antes de iniciar el nuevo servidor, debemos detener OpenSSH con el siguiente comando.

```
sudo bash -c 'systemctl stop sshd && systemctl
disable sshd'
```

En sistemas basados en BSD...

```
sudo bash -c 'rc-service sshd stop && rc-update del
sshd'
```



Iniciar Dropbear

Una vez hayamos parado el servicio OpenSSH, podemos iniciar y habilitar el servicio Dropbear.

```
sudo systemctl start dropbear
sudo systemctl enable dropbear
```

En sistemas basados en BSD...

```
sudo rc-service dropbear start
sudo rc-update add dropbear
```

Como usar Dropbear

Dropbear no requiere de ningún aprendizaje alternativo, su funcionamiento y comandos son similares a OpenSSH. Por ejemplo para conectar...

```
ssh ipdelserver
ssh usuario@miservidor.midominio
dbclient usuario@miservidor.midominio
```

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



Verificar los password con John The Ripper en Ubuntu



John The Ripper Password check

John The Ripper es sin duda la herramienta para descifrar contraseñas (passwords), más conocida y utilizada. John funciona bien y es muy efectivo.

La herramienta **John The Ripper** es un **Metasploit**, que pertenece a la familia de aplicaciones **pentesting / hacking** de **Rapit7**. Es evidente que desde **sololinux** no fomentamos el pirateo, por eso no vamos a explicar las profundidades oscuras de John. Lo que si que vamos a ver, es como verificar los passwords (contraseñas) de los usuarios en un sistema linux. Fácil y rápido.



Imagen 18. Extraer password con John The Ripper

Verificar los password con John The Ripper

Como es evidente, lo primero que debemos hacer es instalar John The Ripper.

```
sudo apt update
sudo apt install john -y
```

La herramienta nos ofrece la opción, de instalar listados de palabras comunes definidas por idioma. Podemos ver las listas actuales con el siguiente comando.

```
sudo apt install wordlist
```

En nuestro caso nos ofrece las siguientes listas.

```
root@185:~# sudo apt install wordlist
Reading package lists... Done
Building dependency tree
Reading state information... Done
Package wordlist is a virtual package provided by:
 wgerman-medical 20160103-1
 wesperanto 2.1.2000.02.25-55
 wcanadian-small 2017.08.24-1
 wcanadian-large 2017.08.24-1
 wcanadian-insane 2017.08.24-1
 wcanadian-huge 2017.08.24-1
 wcanadian 2017.08.24-1
 wbritish-small 2017.08.24-1
 wbritish-large 2017.08.24-1
 wbritish-insane 2017.08.24-1
 wbritish-huge 2017.08.24-1
 wamerican-small 2017.08.24-1
 wamerican-large 2017.08.24-1
 wamerican-insane 2017.08.24-1
```

```
wamerican-huge 2017.08.24-1
wukrainian 1.7.1-2
wswiss 20161207-4
wswedish 1.4.5-2.2
wspanish 1.0.27
wportuguese 20171225-1
wpolish 20170707-1
wogerman 1:2-33
wnorwegian 2.2-3
wngerman 20161207-4
witalian 1.8
wgalician-minimos 0.5-43
wfrench 1.2.3-11build1
wfaroese 0.4.2-11
wdutch 1:2.10-6
wdanish 1.6.36-9
wcatalan 0.20111230b-10
wbulgarian 4.1-3ubuntu1
wbritish 2017.08.24-1
wbrazilian 3.0~beta4-20
wamerican 2017.08.24-1
miscfiles 1.5+dfsg-2
```

You should explicitly select one to install.

Puedes instalar todas las listas que quieras, no existe limite. En nuestro caso nos decantamos por la lista american-large y la spanish.

```
sudo apt install wspanish
sudo apt install wamerican-large
```

Necesitamos combinar **shadow** y **passwd** (este paso es temporal, si reinicias el sistema tendrás que ejecutarlo de nuevo).

```
sudo /usr/sbin/unshadow /etc/passwd /etc/shadow > /tmp/crack.password.db
```

Lanzamos John The Ripper, en busca de los usuarios y contraseñas existentes en nuestro sistema.

```
john /tmp/crack.password.db
```

Nosotros hemos creado un usuario (ademas del root), y unos passwords muy sencillos para que sea más rápido (3 segundos). Observa la salida de ejemplo.

```
root@185:~# john /tmp/crack.password.db
Loaded 2 password hashes with 2 different salts
(crypt, generic crypt(3) [?/64])
Press 'q' or Ctrl-C to abort, almost any other key
for status
12345 (sololinux) <----- Usuario
y pass
1234 (root) <----- Administrador
y pass
2g 0:00:00:03 100% 2/3 0.5167g/s 457.3p/s 482.1c/s
482.1c/s 123456..pepper
Use the "--show" option to display all of the
cracked passwords reliably
Session completed
```

Para mostrar las contraseñas descifradas de manera confiable, puedes agregar la opción «--show».

```
john --show /tmp/crack.password.db
```

```
root@185:~# john --show /tmp/crack.password.db
root:1234:0:0:root:/root:/bin/bash
sololinux:12345:1000:1000::/home/sololinux:/bin/sh
2 password hashes cracked, 0 left
```

Instalar las herramientas de Kali en Ubuntu 20.04 LTS



Katoolin es un script escrito en **Python**, que instala las herramientas de **Kali Linux** en Ubuntu y otros sistemas basados en **Debian**.

No importa si eres un analista de seguridad, o un aficionado entusiasta. **Katoolin** nos permite instalar cualquier herramienta de **Kali Linux** (o todas), en un abrir y cerrar de ojos. Los requisitos son sencillos:

- Python igual o superior a la versión 3.5
- Git es necesario
- sh, bash
- python3-apt

En el artículo de hoy veras que sencillo es el proceso de **instalar Katoolin**, así como su uso.

Instalar las herramientas de Kali en Ubuntu 20.04

Para crear el manual, hemos utilizado un Ubuntu 20.04 server. El proceso de instalación es el mismo en Ubuntu 18.04 y otros derivados.

Comenzamos actualizando e instalando GIT.

```
sudo apt update
sudo apt install git
```

Agregamos el **repositorio Universe**.

```
sudo add-apt-repository universe
```

Actualizamos de nuevo y clonamos desde Github la herramienta Katoolin3.

```
sudo apt update
sudo git clone
https://github.com/s-h-3-l-l/katoolin3
```

Una vez clonado, accedemos al directorio, concedemos permisos de ejecución e instalamos la herramienta.

```
cd katoolin3/
chmod +x ./install.sh
sudo ./install.sh
```

Comienza la instalación...

```
Get:892 http://mirror.neostada.nl/kali kali-rolling/main amd64 libatk1-0-data all 2.36.0-2 [149 kB]
Get:893 http://mirror.neostada.nl/kali kali-rolling/main amd64 libatk1-0 amd64 2.36.0-2 [22.2 kB]
Get:897 http://mirror.neostada.nl/kali kali-rolling/main amd64 libgtk2.0-common all 2.24.32-4 [270 kB]
Get:898 http://mirror.neostada.nl/kali kali-rolling/main amd64 libgtk2.0 amd64 2.24.32-4 [1010 kB]
Get:899 http://mirror.neostada.nl/kali kali-rolling/main amd64 libxcurser1 amd64 1.1.2-0.2 [37.3 kB]
Get:900 http://mirror.neostada.nl/kali kali-rolling/main amd64 libxinerama amd64 2:1.1.4-2 [17.7 kB]
Get:901 http://mirror.neostada.nl/kali kali-rolling/main amd64 libxkb2.0 amd64 2.24.32-4 [1010 kB]
Get:902 http://mirror.neostada.nl/kali kali-rolling/main amd64 python-cairo amd64 1.16.2-3 [59.5 kB]
Get:903 http://mirror.neostada.nl/kali kali-rolling/main amd64 python-gobject-2 amd64 2.28.0-1 [294 kB]
Get:904 http://mirror.neostada.nl/kali kali-rolling/main amd64 python-gtk2 amd64 2.24.0-6 [638 kB]
Get:905 http://mirror.neostada.nl/kali kali-rolling/main amd64 cherrytree all 0.10.0-0ubuntu1 [772 kB]
Get:906 http://mirror.neostada.nl/kali kali-rolling/main amd64 chrtwp amd64 1.0-1 [91.9 kB]
Get:907 http://mirror.neostada.nl/kali kali-rolling/main amd64 chromium-sandbox amd64 0.0.0-02.1 [142 kB]
Get:908 http://mirror.neostada.nl/kali kali-rolling/main amd64 libclang-cpp amd64 1:9.0.1-12 [8623 kB]
Get:909 http://mirror.neostada.nl/kali kali-rolling/main amd64 libpcc1 amd64 1:7.4.0-4 [124 kB]
Get:910 http://mirror.neostada.nl/kali kali-rolling/main amd64 libobjc4 amd64 10.20200418-1 [41.7 kB]
Get:911 http://mirror.neostada.nl/kali kali-rolling/main amd64 libobjc-dev amd64 0.0-11 [210 kB]
Get:912 http://mirror.neostada.nl/kali kali-rolling/main amd64 libcc-1 amd64 2.30-4 [1020 kB]
Get:913 http://mirror.neostada.nl/kali kali-rolling/main amd64 libzstdc1 amd64 10-20200418-1 [49.4 kB]
Get:914 http://mirror.neostada.nl/kali kali-rolling/main amd64 libzstdc1 amd64 1:10-20200418-1 [1100 kB]
Get:915 http://mirror.neostada.nl/kali kali-rolling/main amd64 libzstdc1 amd64 10-20200418-1 [510 kB]
Get:916 http://mirror.neostada.nl/kali kali-rolling/main amd64 libclang-common-9-dev amd64 1:9.0.1-12 [1729 kB]
```

Imagen 19. Instalar las herramientas de Kali en Ubuntu

Al terminar la instalación veras algo similar a lo siguiente.
 Executing: /tmp/apt-key-gpg/home.AFi1c4l47u/gpg.1.sh
 -qq --keyserver pool.sks-keyservers.net --recv-keys
 ED444FF07D8D0BF6
 Successfully installed.
 Run it with 'sudo katoolin3'.

Te recomiendo reiniciar el sistema.

```
sudo reboot
```

Una vez inicie el sistema de nuevo, lanzamos katoolin3 con el comando que nos indico en su proceso de instalación.

```
sudo katoolin3
```

Ahora nos aparece el menú general, puedes instalar la herramienta que necesites o incluso todas.

```
Main Menu
0) View Categories
1) Install All
2) Uninstall All
3) Search repository
4) List installed packages
5) List not installed packages
6) Install Kali Menu
7) Uninstall old katoolin
8) Help
9) Exit
```

Como ejemplo... si pulsas la tecla numérica «0», vemos las categorías de las herramientas a instalar. Selecciona según tu interés o necesidad.

```
Main Menu
0) View Categories
1) Install All
2) Uninstall All
3) Search repository
4) List installed packages
5) List not installed packages
6) Install Kali Menu
7) Uninstall old katoolin
8) Help
9) Exit
kat> 0
Select a Category
0) Exploitation Tools      8) Sniffing & Spoofing
1) Forensics Tools        9) Stress Testing
2) Hardware Hacking       10) Vulnerability Analysis
3) Information Gathering  11) Web Applications
4) Maintaining Access     12) Wireless Attacks
5) Password Attacks       13) HELP
6) Reporting Tools        14) BACK
7) Reverse Engineering
kat>
```

Si quieres desinstalar katoolin, es proceso es sencillo (incluye su propio script). Sigue los pasos indicados.

```
cd katoolin3/
chmod +x ./uninstall.sh
sudo ./uninstall.sh
```

Utiliza las herramientas de Kali en Ubuntu con prudencia y responsabilidad. Recuerda que puedes incurrir en graves delitos penados por ley. **No seas pirata.**

Como configurar una red con nmtui en linux

Seguro que más de una vez, has tenido problemas a la hora de configurar tu red con **NetworkManager**. Por suerte tenemos a **nmtui** para facilitarnos la tarea.

La herramienta **nmtui**, es una interfaz de usuario basada en texto desarrollada para administrar redes. No es tan potente ni completa como su hermana **nmcli**, pero la interfaz en consola que nos ofrece es muy sencilla, intuitiva y además no requiere de curva de aprendizaje.

Como configurar una red con nmtui

No es necesario ser **root**, ni usar **sudo**. Para comenzar a configurar o modificar la red, inserta el siguiente comando.

```
nmtui
```

Se inicia la aplicación. Con las flechas del teclado seleccionas la opción deseada, en nuestro caso editamos una conexión ya creada; con el tabulador (**tecla Tab**), seleccionas aceptar y pulsas Enter.

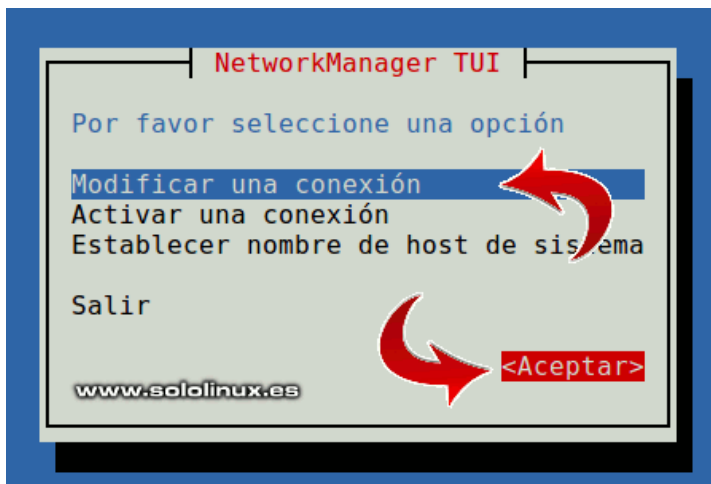


Imagen 20. NetworkManager TUI

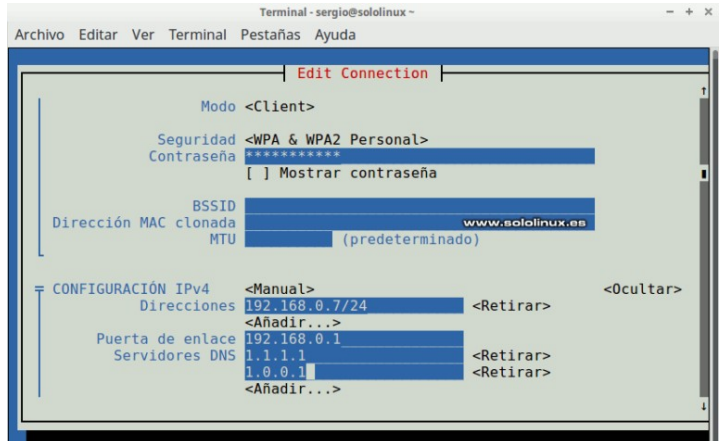


Imagen 22. Configurar una red con nmtui

Hemos retornado al menú de interfaces y redes, marca salir y pulsa Enter. Tu red ha sido creada o modificada, fácil, muy fácil.



Imagen 23. Salir de nmtui

En la nueva pantalla seleccionas la red con las flechas del teclado, con el tabulador puedes marcar las opciones de la derecha (pulsando repetidamente), en nuestro caso «Editar».

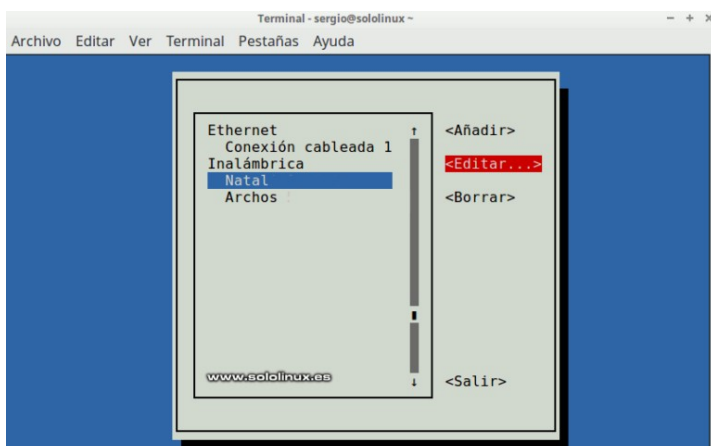
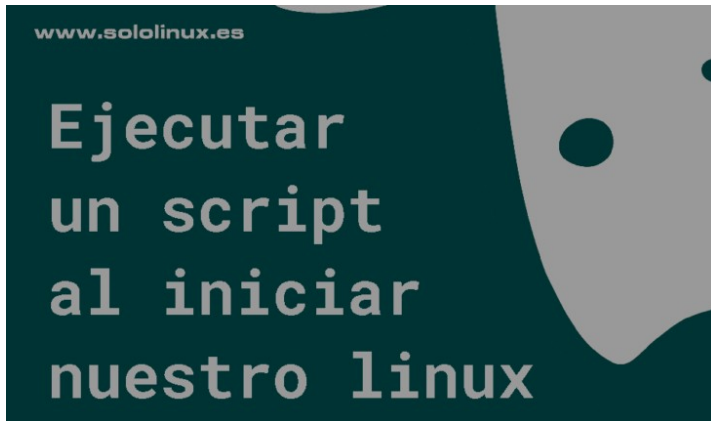


Imagen 21. Seleccionamos una red a editar

Ahora ya podemos modificar nuestra conexión de red, o crearla en caso que fuera tu elección. Puedes moverte por todos los apartados con las flechas del teclado. Para terminar marcas «Aceptar», y pulsas Enter.

Como ejecutar un script al iniciar nuestro sistema



Es algo común, que necesitemos **ejecutar un script al iniciar nuestro sistema linux**. Normalmente... scripts o comandos que verifican la integridad del sistema, otros que generan registros y nos avisan, o simplemente cualquier utilidad específica que necesites.

Existen diversos métodos para lograr nuestro objetivo, pero nosotros nos centramos en dos. Estos son validos para la inmensa mayoría de **sistemas basados en linux**. Hablamos de «**rc-local**» y de «**crontab**».

Como ejecutar un script al iniciar nuestro sistema

El método «**rc-local**», es el clásico de toda la vida. Su funcionamiento es simple. Todo el contenido del archivo se ejecutara al iniciar el sistema, concretamente antes de iniciar la sesión de usuario (por ejemplo, después de introducir nuestro usuario y password).

En la actualidad, el único requisito es que tengas el **servicio systemd** habilitado. Lo puedes verificar con el siguiente comando.

```
sudo systemctl status rc-local
```

```
root@185:~# sudo systemctl status rc-local
rc-local.service - /etc/rc.local Compatibility
Loaded: loaded (/lib/systemd/system/rc-local.service; enabled-runtime; vendor
Drop-In: /lib/systemd/system/rc-local.service.d
         └─debian.conf
Active: active (exited) since Wed 2020-05-13 10:05:42 BST; 1min 10s ago
Docs: man:systemd-rc-local-generator(8)
Process: 389 ExecStart=/etc/rc.local start (code=exited, status=0/SUCCESS)
```

Imagen 24. Ejecutar un script – Verificar el servicio

Bien... lo único que tienes que hacer es abrir el siguiente archivo, e insertar la ruta del script que quieres que se ejecute al iniciar el sistema. La ruta puede variar dependiendo de tu distribución.

```
sudo nano /etc/rc.local
# 0
sudo nano /etc/rc.d/rc.local
```

Veras algo similar a:

```
GNU nano 2.5.3 Archivo: /etc/rc.local

#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
exit 0
```

Ahora, después de las líneas comentadas y antes del **exit**, copias y pegas la ruta del script seleccionado. No te olvides de concederle permisos de ejecución al script. Por ejemplo.

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each
multiuser runlevel.
# Make sure that the script will "exit 0" on success
or any other
# value on error.
#
# In order to enable or disable this script just
change the execution
# bits.
#
# By default this script does nothing.
/home/sololinux/demoBash/libera.sh
exit 0
```

Guarda el archivo y cierra el editor. Como queremos verificar su correcto funcionamiento, reiniciamos el sistema.

```
sudo reboot
```

Una vez inicie nuestro linux:

```
sudo systemctl status rc-local
```

En el ejemplo vemos que nuestro script funciona correctamente (incluso nos informa de la hora de los diferentes comandos que contenía).

```
sergiogsololinux ~ $ sudo systemctl status rc-local
rc-local.service - /etc/rc.local Compatibility
Loaded: loaded (/lib/systemd/system/rc-local.service; static; vendor preset: enabled)
Drop-In: /lib/systemd/system/rc-local.service.d
         └─debian.conf
Active: active (exited) since mié 2020-05-13 11:55:40 EEST; 5min ago
Process: 1264 ExecStart=/etc/rc.local start (code=exited, status=0/SUCCESS)

may 13 11:55:37 sololinux rc.local[1264]: Created by SergioG.B.
may 13 11:55:37 sololinux rc.local[1264]: https://www.sololinux.es
may 13 11:55:40 sololinux rc.local[1264]: <----->
may 13 11:55:40 sololinux rc.local[1264]:
may 13 11:55:40 sololinux rc.local[1264]: Memory:      2936248      203884      free      shared      buff/cache      available
may 13 11:55:40 sololinux rc.local[1264]: Swap:        3084412         0      3084412
may 13 11:55:40 sololinux rc.local[1264]: .....TODD CORRECTO.....
may 13 11:55:40 sololinux rc.local[1264]: <----->
may 13 11:55:40 sololinux rc.local[1264]:
may 13 11:55:40 sololinux systemd[1]: Started /etc/rc.local Compatibility.
```

Imagen 25. Verificar la ejecución de script al iniciar el sistema

Con el método Crontab, es quizás más sencillo. Mira que fácil...

```
sudo crontab -e
```

Veras algo similar a...

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for
'any').#
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless
redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and
cron(8)
#
# m h dom mon dow  command
```

Al final del archivo, insertamos la orden de ejecutar un script en el inicio del sistema.

```
@reboot [ruta-del-script]
```

Por ejemplo...

```
@reboot /home/sololinux/demoBash/libera.sh
```

Guarda el archivo y cierra el editor.

Ya puedes reiniciar el sistema.

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



SolWordPress

Revista Bimestral

Manuales

Noticias

Temas

Plugins

SEO

Seguridad

Entra en: www.solowordpress.es



¿Te interesa colaborar con SoloLinux?

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

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

Extraer las claves wifi guardadas en linux

La primera vez que nos conectamos a una **red inalámbrica**, nos solicita la **clave wifi**. Estas **keys wifi** se van guardando en nuestro sistema linux, independientemente de nuestra **distribución**.

Como norma general... al cambiar a una **wifi** a la que hemos estado conectados, la conexión se produce de forma automática. Esto sucede porque existe un archivo asociado al dispositivo wifi (único), en el cual se van acumulando las contraseñas.

El problema puede surgir al cambiar el dispositivo, o simplemente añadir uno nuevo. Como ya dije anteriormente, cada adaptador de red tiene su propio archivo de **keys wifi**, por tanto tendrás que introducir otra vez el password.

Esto puede resultar un problema, no podemos memorizar todas las contraseñas. La solución es fácil... vamos a extraer las claves wifi que tenemos guardadas de una red en particular. Sencillo y divertido.

Extraer las claves wifi guardadas en linux

Es evidente que antes de extraer las claves, debemos identificar las redes wifi que tenemos guardadas en nuestro sistema. Ejecutamos el siguiente comando.

```
sudo ls /etc/NetworkManager/system-connections/
```

Un ejemplo de salida...

```
sololinux ~ # sudo ls /etc/NetworkManager/system-connections/
WifiHasH Chasnirkaya Sololinux Elochka
```

Bien... ya identificamos las redes guardadas. Como ejemplo vamos a **sacar la key wifi** de la red **Sololinux** (debes modificar el siguiente comando, recuerda que **Sololinux** es a modo de ejemplo).

```
sudo cat
/etc/NetworkManager/system-connections/Sololinux
```

```
sololinux ~ # sudo cat /etc/NetworkManager/system-connections/Natalia0805
[connection]
id=Sololinux
uuid=bf3d56d5-79c9-4epe-71af-vgt6756gt719
type=wifi
permissions=user:mint:;
secondaries=
[wifi]
mac-address=08:93:P8:G7:B0:D3
mac-address-blacklist=
mac-address-randomization=0
mode=infrastructure
seen-bssids=
ssid=Sololinux
[wifi-security]
auth-alg=open
group=
key-mgmt=wpa-psk
pairwise=
proto=
psk=hQ^_:!6}7RHXk!~tp{%F <<<<<-----KEY WIFI
[ipv4]
dns-search=
method=auto
[ipv6]
addr-gen-mode=stable-privacy
dns-search=
method=auto
```

Como puedes ver en el ejemplo, la key de la wifi Sololinux es:
hQ^_:!6}7RHXk!~tp{%F

Otra alternativa interesante es, hacer uso de la herramienta **Wifrest**, que además de linux también es compatible con Windows y Mac.

Si quieres probar **Wifrest**, debes clonar su repositorio.

```
sudo git clone
https://github.com/LionSec/wifrest.git
```

Copiamos el archivo al PATH y le concedemos permisos de ejecución.

```
sudo cd wifrest/
sudo cp wifrest.py /usr/bin/wifrest
chmod +x /usr/bin/wifrest
```

Lanzamos la aplicación con el siguiente comando.

```
sudo wifrest
```

Nos aparece una pantalla gráfica en comandos, muy sencilla y fácil de utilizar.



Imagen 26. Extraer las claves wifi con Wifrest

Please choose your operating system.

- 1) linux
- 2) Windows
- 3) Mac OS

> 1

All wireless networks :

WifiHasH
Doma
Chasnirkaya
Sololinux
Elochka

Insert the network name , or press (a) to see information about all networks.

> Sololinux

```
##### - Sololinux -
#####
```

key-mgmt=wpa-psk

--

psk=hQ^_:!6}7RHXk!~tp{%F <<<----- KEY WIFI

```
#####
#####
```

Administrar tarjetas ethernet con el comando ethtool

Ethtool es una sencilla herramienta, que nos permite visualizar y configurar los parámetros del hardware ethernet de nuestra máquina.

Este comando, nos proporciona mucha información sobre los dispositivos

Ethernet conectados. La configuración de la tarjeta Ethernet, abre un amplio abanico de posibilidades para optimizar la transferencia de datos en una red cableada.

En el artículo de hoy, vemos los ejemplos más comunes de uso de esta útil aplicación. Tanto para identificar, como configurar.

Administrar tarjetas ethernet con el comando ethtool

Normalmente, ethtool viene instalado en todas las distribuciones linux. Por si acaso... vemos como instalar ethtool.

En CentOS, Fedora y derivados:

```
sudo yum install -y ethtool
# 0
sudo dnf install -y ethtool
```

En Debian, Ubuntu, Linux Mint y derivados:

```
sudo apt install ethtool
```

En Arch Linux, Manjaro y derivados:

```
sudo pacman -S ethtool
```

En Suse, OpenSuse y derivados:

```
sudo zypper install -y ethtool
```

Una vez tengamos instalada la herramienta, debemos identificar nuestro dispositivo ethernet. Según nuestro linux usaremos el **comando ip**, o el **comando ifconfig**.

```
ip a
# 0
ifconfig
```

```
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: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
   link/ether 80:34:95:21:b3:6c brd ff:ff:ff:ff:ff:ff
   inet 61.210.151.251/24 brd 62.210.151.255 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
   inet6 fe80::8934:97ff:fe11:b91c/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
   link/ether 84:34:97:11:b3:6d brd ff:ff:ff:ff:ff:ff
```

Como podemos comprobar, nuestra maquina de ejemplo tiene dos tarjetas de red, eth0 y eth1. La que estamos utilizando es la eth0, así que revisamos su configuración básica con el siguiente comando.

```
ethtool eth0
```



```

Settings for eth0:
Supported ports: [ TP ]
Supported link modes:   10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Half 1000baseT/Full

Supported pause frame use: No
Supports auto-negotiation: Yes
Supported FEC modes: Not reported
Advertised link modes:   10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Half 1000baseT/Full

Advertised pause frame use: Symmetric
Advertised auto-negotiation: Yes
Advertised FEC modes: Not reported
Link partner advertised link modes: 10baseT/Half 10baseT/Full
                                    100baseT/Half 100baseT/Full
                                    1000baseT/Half 1000baseT/Full

Link partner advertised pause frame use: No
Link partner advertised auto-negotiation: Yes
Link partner advertised FEC modes: Not reported
Speed: 1000Mb/s
Duplex: Full
Port: Twisted Pair
PHYAD: 1
Transceiver: internal
Auto-negotiation: on
MDI-X: off
Supports Wake-on: g
Wake-on: g
Current message level: 0x000000ff (255)
drv probe link timer ifdown ifup rx_err tx_err
Link detected: yes

```

Ahora verificamos el controlador y el firmware de nuestro dispositivo.

```
ethtool -i eth0
```

```

driver: tg3
version: 3.137
firmware-version: 5720-v1.32 NCSI v1.1.15.0
expansion-rom-version:
bus-info: 0000:03:00.0
supports-statistics: yes
supports-test: yes
supports-eprom-access: yes
supports-register-dump: yes
supports-priv-flags: no

```

También podemos visualizar las estadísticas de uso.

```
ethtool -S eth0
```

Vemos la salida de ejemplo de un servidor en producción. Paquetes enviados, recibidos, errores, etc...

```

NIC statistics:
rx_packets: 32789017
tx_packets: 71122690
rx_bytes: 3038700048
tx_bytes: 99919029139
rx_broadcast: 365083
tx_broadcast: 4
rx_multicast: 1225215
tx_multicast: 1222
multicast: 1225215
collisions: 0
rx_crc_errors: 0
rx_no_buffer_count: 0
rx_missed_errors: 0
tx_aborted_errors: 0
tx_carrier_errors: 0
tx_window_errors: 0
tx_abort_late_coll: 0
tx_deferred_ok: 0
tx_single_coll_ok: 0
tx_multi_coll_ok: 0
tx_timeout_count: 0
rx_long_length_errors: 0
rx_short_length_errors: 0
rx_align_errors: 0
tx_tcp_seg_good: 25191059
tx_tcp_seg_failed: 0
rx_flow_control_xon: 0
rx_flow_control_xoff: 0
tx_flow_control_xon: 0
tx_flow_control_xoff: 0
rx_long_byte_count: 3038700048
tx_dma_out_of_sync: 0
tx_smbus: 954
rx_smbus: 1207818
dropped_smbus: 35
os2bmc_rx_by_bmc: 473
os2bmc_tx_by_bmc: 0

```

```

os2bmc_tx_by_host: 473
os2bmc_rx_by_host: 0
tx_hwtstamp_timeouts: 0
tx_hwtstamp_skipped: 0
rx_hwtstamp_cleared: 0
rx_errors: 0
tx_errors: 0
tx_dropped: 0
rx_length_errors: 0
rx_over_errors: 0
rx_frame_errors: 0
rx_fifo_errors: 94
tx_fifo_errors: 0
tx_heartbeat_errors: 0
tx_queue_0_packets: 8924582
tx_queue_0_bytes: 12508100173
tx_queue_0_restart: 0
tx_queue_1_packets: 8677383
tx_queue_1_bytes: 12138277106
tx_queue_1_restart: 6
tx_queue_2_packets: 8774592
tx_queue_2_bytes: 12278712285
tx_queue_2_restart: 4
tx_queue_3_packets: 9069991
tx_queue_3_bytes: 12731047984
tx_queue_3_restart: 4
tx_queue_4_packets: 9083140
tx_queue_4_bytes: 12724609732
tx_queue_4_restart: 2
tx_queue_5_packets: 8774137
tx_queue_5_bytes: 12292153412
tx_queue_5_restart: 10
tx_queue_6_packets: 9003116
tx_queue_6_bytes: 12593455265
tx_queue_6_restart: 7
tx_queue_7_packets: 8814795
tx_queue_7_bytes: 12363886239
tx_queue_7_restart: 11
rx_queue_0_packets: 4331120
rx_queue_0_bytes: 374862760
rx_queue_0_drops: 0
rx_queue_0_csum_err: 21
rx_queue_0_alloc_failed: 0
rx_queue_1_packets: 3916049
rx_queue_1_bytes: 345447419
rx_queue_1_drops: 94
rx_queue_1_csum_err: 1055
rx_queue_1_alloc_failed: 0
rx_queue_2_packets: 3940280
rx_queue_2_bytes: 349452804
rx_queue_2_drops: 0
rx_queue_2_csum_err: 42
rx_queue_2_alloc_failed: 0
rx_queue_3_packets: 4020698
rx_queue_3_bytes: 351359496
rx_queue_3_drops: 0
rx_queue_3_csum_err: 11
rx_queue_3_alloc_failed: 0
rx_queue_4_packets: 3962799
rx_queue_4_bytes: 344462151
rx_queue_4_drops: 0
rx_queue_4_csum_err: 46
rx_queue_4_alloc_failed: 0
rx_queue_5_packets: 3802558
rx_queue_5_bytes: 335758509
rx_queue_5_drops: 0
rx_queue_5_csum_err: 6
rx_queue_5_alloc_failed: 0
rx_queue_6_packets: 3822042
rx_queue_6_bytes: 334297324
rx_queue_6_drops: 0
rx_queue_6_csum_err: 34
rx_queue_6_alloc_failed: 0
rx_queue_7_packets: 3785559
rx_queue_7_bytes: 340749545
rx_queue_7_drops: 0
rx_queue_7_csum_err: 38
rx_queue_7_alloc_failed: 0

```

Si quieres saber el tipo de negociación (RX / TX), ejecutamos...

```
ethtool -a eth0
```

Veremos algo similar a...

```

Pause parameters for eth0:
Autonegotiate: on
RX: on
TX: off

```

Una vez hemos comprobado nuestra tarjeta, vemos algunos ejemplos de configuración realmente útiles.

Tal vez no sabías que puedes modificar la velocidad de tu tarjeta de red, normalmente 100 o 1000 (puedes insertar cualquier valor admitido por el dispositivo). Debes saber que al realizar esta modificación la interfaz queda en modo OFF, así que la tendrás que levantar otra vez con el comando `ip`, o `ifconfig`.

```
ethtool -s eth0 speed 1000
ip link set eth0 up
```

También puedes habilitar o deshabilitar la negociación automática de ethernet.

```
# Deshabilitar
ethtool -s eth0 autoneg off
# Habilitar
ethtool -s eth0 autoneg on
```

El comando **ethtool** nos permite modificar varios parámetros a la vez. La sintaxis es la siguiente.

```
ethtool -s [dispositivo] speed [10/100/1000] duplex [half/full] autoneg [on/off]
# ejemplo
ethtool -s eth0 speed 1000 duplex full autoneg off
```

Las modificaciones que hemos realizado son temporales, por tanto al reiniciar el sistema volverán a su estado original. Vemos como hacer que sean permanentes.

En sistemas basados en RHEL, CentOS, Fedora y derivados:

```
nano /etc/sysconfig/network-scripts/ifcfg-eth0
```

Insertamos nuestra configuración.

```
ETHTOOL_OPTS="speed 1000 duplex full autoneg off"
```

En sistemas basados en Debian, Ubuntu, Linux Mint y derivados:

```
nano /etc/network/interfaces
```

Pegamos nuestra configuración preferida.

```
post-up ethtool -s eth0 speed 1000 duplex full autoneg off
```

Guardamos el archivo, solo necesitas reiniciar tu sistema.

```
reboot
```

¿Te interesa colaborar
con SoloLinux?

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

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

Proteger un servidor contra ataques SYN y suplantación de IP

La herramienta **sysctl** nos permite ver, configurar y automatizar la configuración del kernel en el directorio `/proc/sys/`.

Todos sabemos, que una de las mayores lacras de los servidores (o vps) son los ataques, sobre todo los SYN o de IP. Aprovechando la potencia del **comando sysctl** para modificar el **kernel**, podemos mejorar de forma considerable nuestras defensas contra este tipo de ataques.

Para proteger un servidor de manera considerable, vamos a realizar los siguientes pasos.

- Deshabilitar el reenvío de IP.
- Desactivar el redireccionamiento de envío de paquetes.
- Prohibir el reenvío de paquetes ICMP.
- Rechazar los mensajes de error incorrectos.

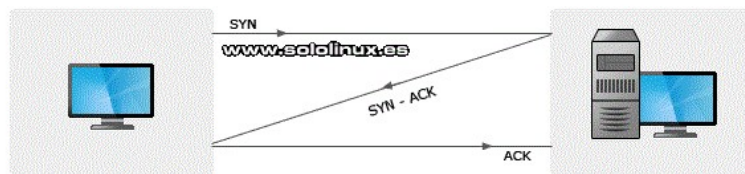


Imagen 26. Proteger un servidor contra ataques SYN

Proteger un servidor contra ataques SYN

Este tutorial será un poco diferente al resto, si quieres solucionar los defectos originales de un servidor o VPS, tan solo tienes que seguir todos los pasos que te indico. Solucionaremos los problemas mencionados anteriormente.

Todas las modificaciones las realizamos en el archivo de configuración **sysctl**, así que vamos a ello (si no existen las líneas, deben agregarse).

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

Modifica o inserta lo siguiente.

```
#net.ipv4.ip_forward=1
# cambiar o insertar por...
net.ipv4.ip_forward=0
```

```
#net.ipv4.conf.all.send_redirects = 0
# cambiar o insertar por...
net.ipv4.conf.all.send_redirects = 0
```

```
#net.ipv4.conf.all.accept_redirects = 0
# cambiar o insertar por...
net.ipv4.conf.all.accept_redirects = 0
# Agrega despues...
net.ipv4.conf.default.accept_redirects = 0
```

Al final del archivo copia y pega lo siguiente.

```
net.ipv4.icmp_ignore_bogus_error_responses = 1
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_max_syn_backlog = 2048
net.ipv4.tcp_synack_retries = 3
net.ipv4.netfilter.ip_conntrack_tcp_timeout_syn_recv=45
```

Guarda el archivo y cierra el editor.
Ahora recargamos el servicio.

```
sudo sysctl -p
```

Si aparece algún error del tipo «**el comando sysctl -p no cargó tcp_max_syn_backlog**», es porque no lee el valor 2048. La solución es ejecutar el siguiente comando...

```
sudo less /proc/sys/net/ipv4/tcp_max_syn_backlog
```

De todas formas mi recomendación es que reinicies el servidor.

```
sudo reboot
```

Ya tenemos nuestro servidor protegido. Felicidades, has aprendido a proteger un servidor Linux contra ataques SYN y suplantación de IP.

Solución al error: sudo: add-apt-repository: command not found

Vaya sorpresa me lleve el otro día, al intentar instalar el ppa de Thierry (impresoras Canon), en Ubuntu 20.04 LTS. Al intentar agregar el repo correspondiente.

```
sudo add-apt-repository ppa:thierry-f/fork-michael-gruz
```

Me salta el siguiente error:

sudo: add-apt-repository: command not found

El motivo es simple, el paquete add-apt-repository no está instalado. La solución parece simple... instalar la aplicación...

Pues noooo.... eso no funciona.

La solución al error «**sudo: add-apt-repository: command not found**», es tan sencilla como ejecutar los siguiente.

```
sudo apt update
sudo: add-apt-repository: command not found
```

Ejemplo de salida...

```
root@sololinux:~# sudo apt-get install software-properties-common
Reading package lists... Done
Building dependency tree... Done
The following additional packages will be installed:
  gir1.2-glib-2.0 iso-codes libgirepository-1.0-1 powermgmt-base
  python-apt-common python3-apt python3-dbus python3-gi
  python3-software-properties unattended-upgrades
Suggested packages:
  isoquery python3-apt-dbg python-apt-doc python-dbus-doc python3-dbus-dbg
  bsd-mailx default-mta | mail-transport-agent needrestart
The following NEW packages will be installed:
  gir1.2-glib-2.0 iso-codes libgirepository-1.0-1 powermgmt-base
  python-apt-common python3-apt python3-dbus python3-gi
  python3-software-properties software-properties-common unattended-upgrades
0 upgraded, 11 newly installed, 0 to remove and 35 not upgraded.
Need to get 3271 kB of archives.
After this operation, 22.8 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Una vez instalado lo anterior, actualizamos el sistema.

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

Ya puedes agregar cualquier repositorio ppa, no tendrás ningún problema.



¿Te interesa colaborar
con Sololinux?

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

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

Verificar la conexión SSH de un host remoto

Si administras servidores, es interesante verificar cada cierto tiempo que el **puerto ssh** permanece abierto. Normalmente el puerto 22.

En este artículo, nos aprovechamos de una serie de comandos para crear scripts bash. Los scripts nos facilitan la tarea, al permitirnos generar salidas impresas legibles para el humano.

Verificar la conexión SSH de un host remoto

Asumo que a estas alturas todos sabemos crear un **script bash**, así que nos limitamos a ofrecerlos. Vemos las opciones que tenemos disponibles más comunes (comandos o herramientas).

Timeout

La herramienta **timeout** viene instalada por defecto en la mayoría de **distribuciones linux**. Vemos un script.

```
#!/bin/bash
server=192.168.0.100      # IP del servidor
port=22                   # puerto
connect_timeout=5         # Tiempo de espera
timeout $connect_timeout bash -c "</dev/tcp/$server/$port"
if [ $? == 0 ];then
    echo "SSH Connection to $server over port $port
is possible"
else
    echo "SSH connection to $server over port $port
is not possible"
fi
```

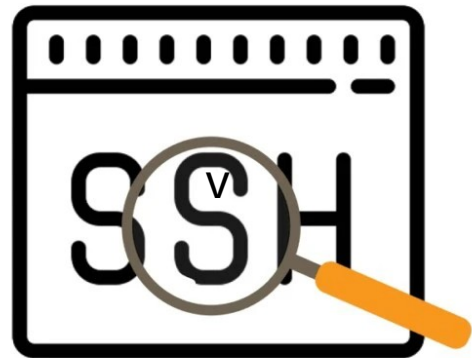
```
sololinux # bash timeout.sh
SSH Connection to 192.168.0.100 over port 22 is
possible
```

Nmap

Nmap es una de las mejores herramientas (para redes), que podemos encontrar. Al ser tan específica, no viene instalada por defecto, pero la tenemos en los repositorios oficiales. Instalamos **nmap**.

```
# Debian, Ubuntu, Linux Mint y derivados
sudo apt install -y nmap
# CentOS, Fedora y derivados
sudo yum -y install nmap
-o-
sudo dnf -y install nmap
# Arch Linux, Manjaro y derivados
sudo pacman -S nmap
-o-
sudo pacman -Rs nmap
# Suse, OpenSuse y derivados
zypper install nmap
# Paquete snap
sudo snap install nmap
# Desde código fuente
sudo wget https://nmap.org/dist/nmap-7.80.tar.bz2
sudo bzip2 -cd nmap-7.80.tar.bz2 | tar xvf -
./configure
make
sudo make install
```

www.sololinux.es



www.sololinux.es

El script bash.

```
#!/bin/bash
server=192.168.0.100      # IP del servidor
port=22                   # puerto
connect_timeout=5         # Tiempo de espera
status=`nmap $server -Pn -p $port | egrep -io 'open|
closed|filtered'`
if [ $status == "open" ];then
    echo "SSH Connection to $server over port $port
is possible"
elif [ $status == "filtered" ]; then
    echo "SSH Connection to $server over port $port
is possible but blocked by firewall"
elif [ $status == "closed" ]; then
    echo "SSH connection to $server over port $port
is not possible"
else
    echo "Unable to get port $port status from
$server"
fi
```

```
sololinux # bash nmap.sh
SSH Connection to 192.168.0.100 over port 22 is
possible
```

NC / NCAT

Ahora utilizaremos **«nc»**, para verificar si la conexión ssh está abierta o cerrada.

```
#!/bin/bash
server=192.168.0.109      # IP del servidor
port=22                   # puerto
connect_timeout=5         # Tiempo de espera
nc --wait $connect_timeout $server $port < /dev/null
&> /dev/null
if [ $? == 0 ];then
    echo "SSH Connection to $server over port $port
is possible"
else
    echo "SSH connection to $server over port $port
is not possible"
fi
```

Si observas el script, esta vez hemos insertado una ip diferente. La salida nos dirá que la conexión ssh no es posible.

```
sololinux # bash nc.sh
SSH connection to 192.168.0.109 over port 22 is not
possible
```

SSH

Mediante SSH también podemos lograr nuestro objetivo, eso si, nos ayudaremos de **ConnectTimeout** y de **StrictHostKeyChecking**.

```
#!/bin/bash
server=192.168.0.100      # IP del servidor
port=22                   # puerto
connect_timeout=5         # Tiempo de espera
ssh -q -o BatchMode=yes -o StrictHostKeyChecking=no
-o ConnectTimeout=$connect_timeout $server 'exit 0'
if [ $? == 0 ];then
    echo "SSH Connection to $server over port $port
is possible"
else
    echo "SSH connection to $server over port $port
is not possible"
fi
```

```
sololinux # bash ssh.sh
SSH Connection to 192.168.0.100 over port 22 is
possible
```

Telnet

Es evidente que no podía faltar la que posiblemente sea la mejor forma, está claro que nos faltaba verificar la **conexión SSH con telnet**.

```
#!/bin/bash
server=192.168.0.100      # IP del servidor
port=22                   # puerto
connect_timeout=5         # Tiempo de espera
echo quit | telnet $server $port 2>/dev/null | egrep
-qi "Connected"
if [ $? == 0 ];then
    echo "SSH Connection to $server over port $port
is possible"
else
    echo "SSH connection to $server over port $port
is not possible"
fi
```

```
sololinux # bash telnet.sh
SSH Connection to 192.168.0.100 over port 22 is
possible
```

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



¿Te interesa colaborar con Sololinux?

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

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

Como instalar Mono (.NET Framework) en Ubuntu 20.04



Mono es una implementación de código abierto, del privativo **.NET Framework de Microsoft**. Patrocinado por el mismo **Microsoft** y con la colaboración de una comunidad muy activa, Mono tiene como meta ser la opción preferida, para que los desarrolladores puedan crear aplicaciones multiplataforma.

Basada en los estándares **ECMA para C# y Common Language Runtime**, Mono es una plataforma de software diseñada específicamente, para crear aplicaciones de forma rápida y sencilla. En el artículo de hoy, vemos como la podemos instalar en Ubuntu 20.04 / 18.04 / 16.04, y todos sus derivados.

Como instalar Mono .NET Framework en Ubuntu 20.04

Mono cuenta con su propio repositorio, así que la instalación es rápida y sencilla. Aun así... antes de comenzar debemos actualizar e instalar otras herramientas necesarias.

```
sudo apt update
sudo apt install dirmngr gnupg apt-transport-https
ca-certificates
```

Una vez terminé el proceso anterior, agregamos la key.

```
sudo apt-key adv --keyserver
hkp://keyserver.ubuntu.com:80 --recv-keys
3FA7E0328081BFF6A14DA29AA6A19B38D3D831EF
```

Ahora agrega el repositorio que corresponda a tu versión de Ubuntu o derivado.

```
# Ubuntu 20.04 Focal
sudo sh -c 'echo "deb https://download.mono-
project.com/repo/ubuntu stable-focal main" >
/etc/apt/sources.list.d/mono-official-stable.list'

# Ubuntu 18.04 Bionic
sudo sh -c 'echo "deb https://download.mono-
project.com/repo/ubuntu stable-bionic main" >
/etc/apt/sources.list.d/mono-official-stable.list'

# Ubuntu 16.04 Xenial
sudo sh -c 'echo "deb https://download.mono-
project.com/repo/ubuntu stable-xenial main" >
/etc/apt/sources.list.d/mono-official-stable.list'
```

Para terminar la instalación ejecutamos los siguientes comandos.

```
sudo apt update
sudo apt install mono-complete
```

Para crear tus proyectos **.NET** tienes varias alternativas, pero sin duda la mejor es el **IDE MonoDevelop**. Lo puedes instalar con el siguiente comando:

```
sudo apt install monodevelop
```

Ya puedes comenzar a trabajar.

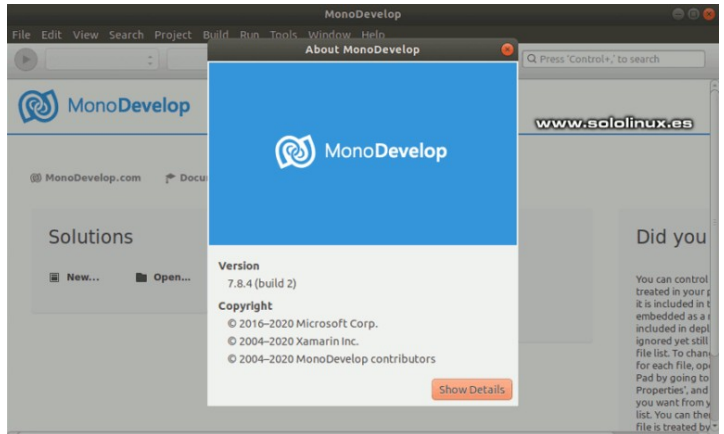


Imagen 27. MonoDevelop – Instalar Mono

Es recomendable que amplíes las funciones de **MonoDevelop**, por ejemplo el complemento para marcos de pruebas unitarias «**nunit**».

```
sudo apt install monodevelop-nunit
```

Como ultimo apunte del artículo «Como instalar Mono .NET Framework en Ubuntu 20.04», verificamos la instalación de Mono.

```
mono --version
```

```
root@sololinux:~# mono --version
Mono JIT compiler version 6.8.0.123 (tarball Tue May
12 15:11:57 UTC 2020)
Copyright (C) 2002-2014 Novell, Inc, Xamarin Inc and
Contributors. www.mono-project.com
TLS:          __thread
SIGSEGV:      altstack
Notifications: epoll
Architecture: amd64
Disabled:     none
Misc:         softdebug
Interpreter:  yes
LLVM:         yes(610)
Suspend:      hybrid
GC:           sgen (concurrent by default)
```


Instalar Samba en Ubuntu 20.04 LTS Focal y derivados

Samba es una herramienta que mediante la implantación del **protocolo SMB / CIFS**, permite a las **distribuciones linux** compartir archivos, impresoras y otros recursos, con sistemas basados en **Microsoft Windows**.

En este artículo veremos como instalar Samba en Ubuntu, y también como configurar Samba tanto en Ubuntu, como en Windows. Para que todo funcione correctamente, conviene que las maquinas estén en la misma subred y grupo de trabajo local. Nosotros usaremos estos datos de ejemplo...

IP de Windows =====> 192.168.100.2
IP de Ubuntu =====> 192.168.100.3
Grupo de trabajo =====> WORKGROUP

Instalar Samba en Ubuntu 20.04 LTS

Vamos paso por paso, así que primero identificamos el grupo de trabajo.

Identificar el Grupo de trabajo

Normalmente usaremos el grupo de trabajo de Windows. Lo podemos identificar desde la consola de Microsoft con el siguiente comando.

```
Software version           Windows 10 Pro
Workstation domain        WORKGROUP
Logon domain              DESKTOP
COM Open Timeout (sec)    0
COM Send Count (byte)     16
COM Send Timeout (msec)   250
The command completed successfully.
```

Imagen 28. Grupo de trabajo – Instalar Samba

Configurar el archivo host

En una pequeña red, no es común tener un sistema de DNS corriendo. Por tanto para referirnos a cada sistema por su nombre, lo ideal es agregarlos en el archivo de host local (en cada una de las máquinas conectadas). En Windows, abres el símbolo del sistema como administrador.

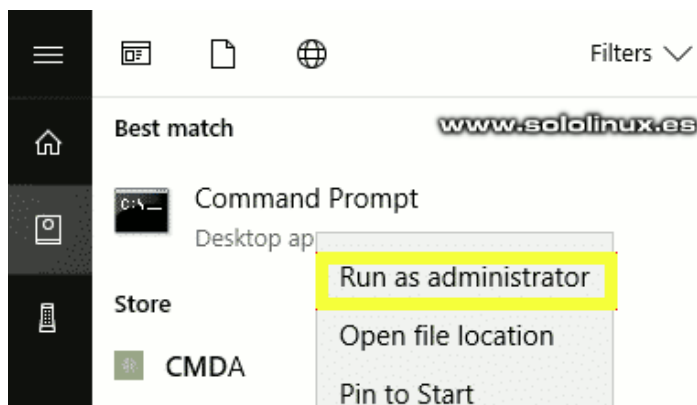


Imagen 29. Abrir consola como administrador – Instalar Samba

Ejecuta lo siguiente.

```
notepad C:\Windows\System32\drivers\etc\hosts
```

Introduce los datos de tu Ubuntu tal como te indico (con tus datos).

```
192.168.100.3  ubuntu-20-04.localhost  ubuntu-20-04
```

Guarda el archivo y cierra.

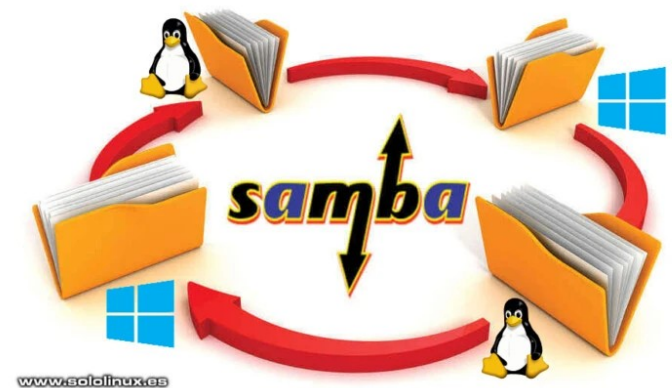
En Ubuntu haremos una operación similar, pero con la ip y el nombre identificador del sistema Windows.

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

En nuestro ejemplo introducimos lo siguiente.

```
192.168.100.102  MiWindows10
```

Guarda el archivo y cierra el editor.



Habilitar el uso compartido de archivos

En Windows, el uso compartido está deshabilitado por defecto. Para activarlo abres el símbolo del sistema como administrador, y ejecutas los siguientes comandos.

```
netsh advfirewall firewall set rule group="File and
Printer Sharing" new enable=Yes
netsh advfirewall firewall set rule group="Network
Discovery" new enable=Yes
```

Cierra la consola, volvemos a la máquina Ubuntu 20.04.

Instalar Samba en Ubuntu 20.04

Samba viene por defecto en los repositorios oficiales, para instalar la herramienta lanzamos el comando necesario.

```
sudo apt update
sudo apt install samba
```

Verificamos que Samba se instaló correctamente y que está corriendo.

```
sudo systemctl status smbd
```

```
Ubuntu-20-04@sololinux:~# sudo systemctl status smbd
● smbd.service - Samba SMB Daemon
   Loaded: loaded (/lib/systemd/system/smbd.service; enabled; vendor
  preset: ena
   Active: active (running) since Tue 2020-05-19 04:44:45 BST; 37s
  ago
     Docs: man:smbd(8)
           man:samba(7)
           man:smb.conf(5)
    Main PID: 2658 (smbd)
   Status: "smbd: ready to serve connections..."
     Tasks: 4 (limit: 4915)
    CGroup: /system.slice/smbd.service
            └─2658 /usr/sbin/smbd --foreground --no-process-group
              2660 /usr/sbin/smbd --foreground --no-process-group
              2661 /usr/sbin/smbd --foreground --no-process-group
              2663 /usr/sbin/smbd --foreground --no-process-group
May 19 04:44:44 sololinux systemd[1]: Starting Samba SMB Daemon...
May 19 04:44:45 sololinux systemd[1]: Started Samba SMB Daemon.
```

Listo, ya tenemos Samba instalado y activo.

Configurar Samba

Una vez hemos terminado de instalar Samba, vamos a configurarlo. Abrimos el archivo y lo modificamos con nuestros datos reales.

```
sudo nano /etc/samba/smb.conf
```

Revisa la configuración detenidamente, es importante que prestes mucha atención a las líneas del código del ejemplo que están marcadas con una flecha. Debes editar o añadir.

```
# Sample configuration file for the Samba suite for Debian
# GNU/Linux.
#
# This is the main Samba configuration file. You should
# read the
# smb.conf(5) manual page in order to understand the
# options listed
# here. Instalar Samba has a number of configurable options
# most of which
# are not shown in this example
#
# Some options that are often worth tuning have been
# included as
# commented-out examples in this file.
# - When such options are commented with ";", the proposed
# setting
# differs from the default Samba behaviour
# - When commented with "#", the proposed setting is the
# default
# behaviour of Samba but the option is considered
# important
# enough to be mentioned here
#
# NOTE: Whenever you modify this file you should run the
# command
# "testparm" to check that you have not made any basic
# syntactic
# errors.
===== Global Settings
=====
[global]
## Browsing/Identification ##
# Change this to the workgroup/NT-domain name your Samba
# server will part of
# workgroup = WORKGROUP <-----
# netbios name = ubuntu-20-04 <-----
# security = user <-----
# proxy = no <-----
# map to guest = bad user <-----
#### Networking ####
# The specific set of interfaces / networks to bind to
# This can be either the interface name or an IP
# address/netmask;
# interface names are normally preferred
# interfaces = 127.0.0.0/8 eth0 <-----
# Only bind to the named interfaces and/or networks; you
# must use the
# 'interfaces' option above to use this.
# bind interfaces only = yes <-----
# Most people will want "standalone server" or "member
# server".
# Running as "active directory domain controller" will
# require first
# server role = standalone server <-----
# obey pam restrictions = yes <-----
# This boolean parameter controls whether Samba attempts to
# sync the Unix
# [public]
# path = /samba/public <-----
# browseable = yes <-----
# guest ok = yes <-----
# guest only = yes <-----
# read only = no <-----
# force user = nobody <-----
# force create mode = 0777 <-----
# force directory mode = 0777 <-----
```

Una vez termines... guarda el archivo y cierra el editor. Es conveniente revisar la configuración de Samba por si tenemos algún error.

```
testparm
```

Nos falta reiniciar Samba (el servicio).

```
sudo systemctl restart smbd
```

Crear una carpeta publica

Hemos terminado de instalar Samba, también lo hemos configurado de manera correcta. Necesitamos crear una carpeta publica.

```
sudo mkdir -p /samba/public
```

Establecemos los permisos necesarios, para que todos los usuarios puedan leer y escribir en la misma.

```
sudo chown -R nobody:nogroup /samba/public
sudo chmod -R 0775 /samba/public
sudo chgrp sambashare /samba/public
```

Reiniciamos el servicio Samba.

```
sudo service smbd restart
```

Bien, ya lo tienes todo listo. Ahora puedes explorar la red desde tu administrador de archivos en Windows. Veras algo similar a...

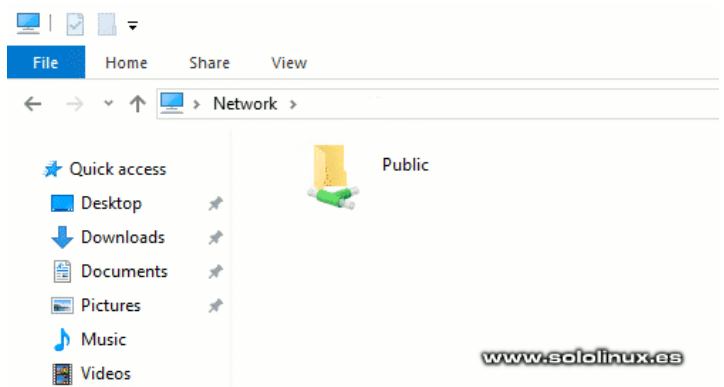


Imagen 30. Carpeta compartida en Windows – Instalar Samba

Instalar el tema MacOS Catalina en linux con XFCE

No importa la distribución Linux que estés utilizando, Xubuntu, Fedora Xfce, Manjaro, Linux Mint Xfce, Open Suse Xfce, etc...

Si te gusta el diseño del nuevo MacOS Catalina y utilizas el entorno gráfico XFCE, estas de suerte. Gracias al **theme MacOS de paullinuxthemer**, hoy vemos como modificar el aspecto de nuestro XFCE, de forma que se asemeje en todo lo posible a MacOS Catalina.

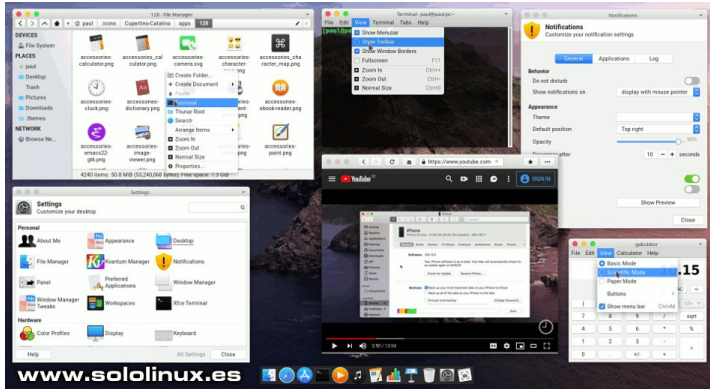


Imagen 31. Instalar el tema MacOS Catalina

Instalar el tema MacOS Catalina en linux con XFCE

Podemos instalar dos variantes, claro y oscuro. Aunque también puedes descargar los dos y decidir cual te gusta más.

Aviso: Este tipo de descargas suele desaparecer sin previo aviso, por ello, las subimos al servidor de **sololinux** para que siempre estén disponibles (son descargas originales).

Tema MacOS Catalina light:

```
wget
https://www.sololinux.es/public/MacOS-XFCE/McOS-CTLina-XFCE.tar.xz
```

Tema MacOS Catalina dark:

```
wget https://www.sololinux.es/public/MacOS-XFCE/McOS-CTLina-XFCE-Dark.tar.xz
```

Ahora lo/s descomprimes.

Descomprimir tema MacOS Catalina light:

```
tar xvf McOS-CTLina-XFCE.tar.xz
```

Descomprimir tema MacOS Catalina dark:

```
tar xvz MacOS-XFCE/McOS-CTLina-XFCE-Dark.tar.xz
```

Ahora accedes a tu administrador de archivos (normalmente Thunar), y en la ruta del usuario «/home/usuario» creas la carpeta «.themes».

Ejemplo...

```
/home/tu-usuario/.themes
```

Copia y pega en «.themes» los temas que hemos descomprimido en el paso anterior.

Bien, ahora haremos exactamente lo mismo con los iconos del tema OS-Catalina.

Descargar iconos Catalina light:

```
wget https://www.sololinux.es/public/MacOS-XFCE/0s-Catalina-icons.tar.xz
```



Imagen 32. Iconos del tema Catalina

Descargar iconos Catalina dark:

```
wget https://www.sololinux.es/public/MacOS-XFCE/0s-Catalina-Night.tar.xz
```

Ahora lo/s descomprimes.

Descomprimir iconos Catalina light:

```
tar xvz MacOS-XFCE/0s-Catalina-icons.tar.xz
```

Descomprimir iconos Catalina dark:

```
tar xvz 0s-Catalina-Night.tar.xz
```

De la misma manera que con el tema, accedemos al administrador de archivos (normalmente Thunar), y en la ruta del usuario «/home/usuario» creamos la carpeta «.icons» (es posible que ya la tengas).

Copia el contenido de las carpetas descomprimidas en «.icons».

Reinicia tu sistema.

```
sudo reboot
```

Una vez reiniciado el sistema, configuramos el nuevo tema.

- Configuración-> Apariencia-> Estilo = Selecciona el nuevo tema.
- Configuración-> Apariencia-> Iconos = Selecciona los nuevos iconos.
- Configuración-> Gestor de ventanas-> Estilo = Selecciona el nuevo tema.

Como ultimo paso, si lo quieres dejar prácticamente igual al original.

- Configuración-> Ajustes del gestor de ventanas-> Compositor = Sigue el ejemplo de la imagen siguiente...



Como ultimo apunte, puedes instalar el Plank for MacOS style Dock en Xfce.

```
sudo apt update
sudo apt install plank
```

Imagen 33. Configurar el Compositor de XFCE

Nuevo Hplib 3.20.5 con soporte para debian 10.3

Los nuevos controladores Hplib 3.20.5, son las utilidades de impresión, escaneo y fax que la compañía HP desarrolla específicamente para linux, algo que la comunidad agradece.

Hplib 3.20.5 agrega la compatibilidad para los siguientes dispositivos:

- HP DeskJet 1200, Ink Advantage 1200
- HP DeskJet 2300 Todo en uno, Ink Advantage 2300 Todo en uno
- HP ENVY 6000 series, Pro 6400 series.
- HP DeskJet Plus 6000, serie 6400
- Impresoras multifunción HP DeskJet 2700 y Ink Advantage 2700
- Impresora multifunción HP DeskJet Plus 4100, serie Ink Advantage 4100
- HP LaserJet Enterprise M610dn, M611dn, M611x, M612dn, M612x
- Impresora multifunción HP LaserJet Enterprise M634dn, M634z, M635h, M635fht, M636fh
- Impresora multifunción HP LaserJet Enterprise Flow M634h, M635z, M636z

HPLIP 3.20.5



www.sololinux.es

Además... esta nueva versión, añade soporte estable para Debian 10.3 y Manjaro 19.0.

Instalar Hplib 3.20.5

Instalar la nueva versión es bastante sencillo, tan solo debes ejecutar los siguientes comandos. Te advierto que al descargar directamente de sourceforge, puede demorar unos minutos (se paciente).

```
wget https://sourceforge.net/projects/hplip/files/hplip/3.20.5/hplip-3.20.5.run
```

Le damos permisos.

```
sudo chmod +x hplip-3.20.5.run
```

Instalamos el nuevo driver.

```
sudo ./hplip-3.20.5.run
```

Comienza la instalación...

```
root@sololinux:~# sudo ./hplip-3.20.5.run
Creating directory hplip-3.20.5
Verifying archive integrity... All good.
Uncompressing HPLIP 3.20.5 Self Extracting
Archive.....
.....
.....
HP Linux Imaging and Printing System (ver. 3.20.5)
HPLIP Installer ver. 5.1
Copyright (c) 2001-18 HP Development Company, LP
This software comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to distribute it
under certain conditions. See COPYING file for more details.
Installer log saved in: hplip-install_Wed-20-May-2020_17:10:13.log
```

Al terminar el proceso, te recomiendo reiniciar el sistema.

```
reboot
```

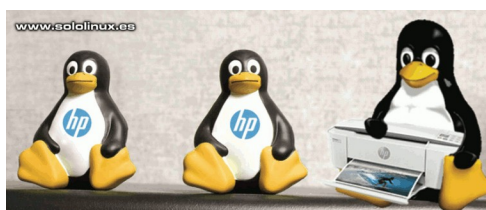


Imagen 34. Nuevo controlador Hplib 3.20.5

Instalar NetSurf en Linux – El navegador web ligero

NetSurf es un navegador web multiplataforma, diseñado para ser muy ligero y rápido. A diferencia de otros navegadores web modernos, NetSurf no tiene extensiones ni complementos que no sirven para nada.

Escrito en **ANSI C**, es de los pocos navegadores actuales que no toma como referencia a **Mozilla** o **Chromium**, tiene su propio motor y realmente vale la pena probarlo. En el artículo de hoy, veremos como instalar NetSurf en las distribuciones Linux más utilizadas por los usuarios.

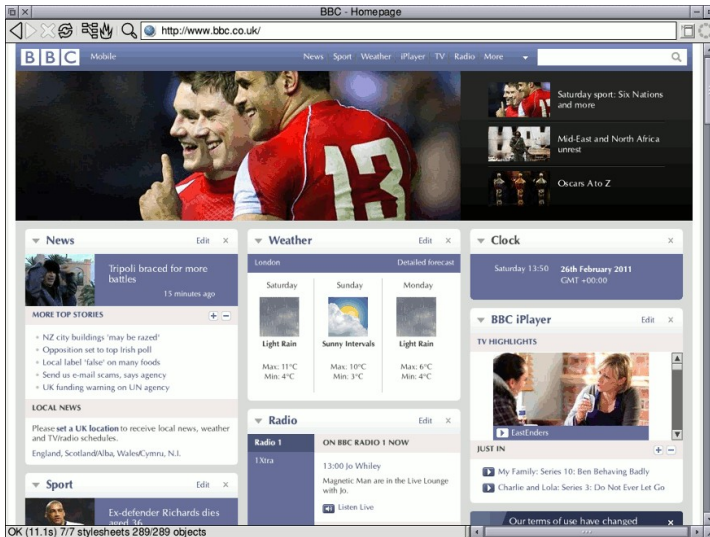


Imagen 35. Instalar NetSurf en Linux

Instalar NetSurf en Linux

La instalación del navegador web NetSurf, la realizamos desde nuestra terminal. Tan solo tienes que seguir las instrucciones que detallamos a continuación. Es muy fácil.

NetSurf en Ubuntu y derivados

```
sudo add-apt-repository universe
sudo apt update
sudo apt install netsurf
```

NetSurf en Debian

```
sudo apt install netsurf
```

NetSurf en Arch Linux, Manjaro y derivados

```
sudo pacman -S netsurf
```

NetSurf en Fedora

```
sudo dnf install netsurf
```

NetSurf en Suse, Open Suse y derivados

```
sudo zypper install netsurf
```

Ya tienes instalado tu nuevo navegador, disfrútalo.



SoloWordpress

Revista Bimestral

Manuales

Noticias

Temas

Plugins

SEO

Seguridad

Entra en: www.solowordpress.es



Conectar un Android o iPhone con Manjaro y Arch Linux

Si tienes un iPhone y quieres conectarlo con tu **Manjaro** o **Arch Linux**, es muy posible que tengas problemas debido a su privacidad y alta protección de datos.

Dependiendo de la versión instalada de **Manjaro**, también pueden producirse errores al conectar un **Android**. En este artículo veremos como hacer compatible los **sistemas Arch** (con sus derivados), con iPhone y Android.



Imagen 36. Conectar un Android o iPhone en Arch y derivados

Conectar un Android o iPhone con Manjaro

Te recomiendo que instales la compatibilidad para los dos sistemas, pero antes de comenzar... debes tener presente que en estos procesos no debes tener conectado ningún dispositivo en tu sistema, no lo olvides.

Para permitir la conexión de dispositivos Android, ejecuta los siguientes comandos:

```
sudo pacman -Syu
sudo pacman -S mtpfs
yay -S jmtfs
sudo pacman -S gvfs-mtp
sudo pacman -Sy gvfs-gphoto2
```

Para permitir la conexión de dispositivos iPhone, ejecuta el siguiente comando:

```
sudo pacman -Syyuu ifuse usbmuxd libplist libimobiledevice
```

Es así de fácil... ahora tan solo es necesario que reinicies tu sistema.

```
sudo reboot
```

Ya puedes conectar tu dispositivo móvil a tu Manjaro Linux, o Arch y derivados.



Obtener información del hardware con hwinfo



La **herramienta hwinfo** nos aporta una completa información del sistema, desde la línea de comandos. De código abierto bajo licencia GPL v2 +, se basa en la **biblioteca libhd**.

Desarrollada en un principio por **OpenSuse**, actualmente la puedes encontrar en los repositorios oficiales de la mayoría de distribuciones linux. Con ella podemos obtener información detallada de la tarjeta gráfica, monitor, módem, escáner, CPU, memoria RAM, disco duro, BIOS, cámara, unidades de CD / DVD, teclado, mouse, impresora, bluetooth, interfaz de red, USB y mucho más.

Obtener información del hardware con hwinfo

Vemos como instalar hwinfo en las distribuciones linux más populares.

Debian, Ubuntu, Linux Mint y derivados:

```
sudo apt install hwinfo
```

Open Suse, Suse y derivados:

```
sudo yast -i hwinfo
```

Fedora y derivados:

```
sudo dnf install hwinfo
```

CentOS y derivados:

```
sudo yum install epel-release
sudo yum update
sudo yum install hwinfo
```

Arch Linux, Manjaro y derivados:

```
sudo pacman -S hwinfo
# o
sudo pacman -Rs hwinfo
```

Su uso es bastante simple, por ejemplo... si quieres ver un listado completo y detallado de todo el hardware conectado, puedes ejecutar cualquiera de los siguientes comandos.

```
hwinfo
sudo hwinfo
hwinfo --all
```

El comando anterior puede imprimir un resultado excesivamente extenso, la verdad es que no es práctico. Para solucionar este problema podemos usar la opción **--short**, que acorta las salidas.

```
hwinfo --short
```

ejemplo de salida...

```
[root@miserver ~]# hwinfo --short
cpu:
Intel(R) Xeon(R) CPU E3-1230 v3 @ 3.30GHz, 800 MHz
Intel(R) Xeon(R) CPU E3-1230 v3 @ 3.30GHz, 875 MHz
Intel(R) Xeon(R) CPU E3-1230 v3 @ 3.30GHz, 843 MHz
Intel(R) Xeon(R) CPU E3-1230 v3 @ 3.30GHz, 830 MHz
Intel(R) Xeon(R) CPU E3-1230 v3 @ 3.30GHz, 836 MHz
Intel(R) Xeon(R) CPU E3-1230 v3 @ 3.30GHz, 830 MHz
Intel(R) Xeon(R) CPU E3-1230 v3 @ 3.30GHz, 1653 MHz
Intel(R) Xeon(R) CPU E3-1230 v3 @ 3.30GHz, 833 MHz

graphics card:
Hewlett-Packard Company iL04

storage:
Floppy disk controller
Intel 8 Series/C220 Series Chipset Family SATA Controller 1 [RAID mode]
Hewlett-Packard Company P222

network:
eth0 Broadcom NetXtreme BCM5720 Gigabit Ethernet PCIe
eth1 Broadcom NetXtreme BCM5720 Gigabit Ethernet PCIe

network interface:
lo Loopback network interface
eth0 Ethernet network interface
eth1 Ethernet network interface

disk:
/dev/sda HP LOGICAL VOLUME

partition:
/dev/sda1 Partition
/dev/sda2 Partition
/dev/sda3 Partition

usb controller:
Intel 8 Series/C220 Series Chipset Family USB xHCI
Intel 8 Series/C220 Series Chipset Family USB EHCI #2
Intel 8 Series/C220 Series Chipset Family USB EHCI #1
Hewlett-Packard Company iL04

bios:
BIOS

bridge:
Intel Xeon E3-1200 v3 Processor DRAM Controller
Intel Xeon E3-1200 v3/4th Gen Core Processor PCI Express x16 Controller
Intel Xeon E3-1200 v3/4th Gen Core Processor PCI Express x8 Controller
Intel Xeon E3-1200 v3/4th Gen Core Processor PCI Express x4 Controller
Intel 8 Series/C220 Series Chipset Family PCI Express Root Port #1
Intel 8 Series/C220 Series Chipset Family PCI Express Root Port #6
Intel 8 Series/C220 Series Chipset Family PCI Express Root Port #8
Intel C222 Series Chipset Family Server Essential SKU LPC Controller

hub:
Linux Foundation 1.1 root hub
Linux Foundation 2.0 root hub
Standard Microsystems Hub
Linux Foundation 3.0 root hub
Linux Foundation 2.0 root hub
Intel Hub
Linux Foundation 2.0 root hub
Intel Hub

memory:
Main Memory

unknown:
FPU
DMA controller
PIC
Timer
Keyboard controller
Hewlett-Packard Company iL04
Hewlett-Packard Company iL04
Hewlett-Packard Enterprise Memory controller
16550A
16550A
HP P222

/dev/ttyS0
/dev/ttyS1
/dev/sg0
```

También podemos especificar el hardware, sobre el que queremos obtener información. Las opciones disponibles son las siguientes:

arch, bios, block, bluetooth, braille, bridge, camera, cdrom, chipcard, cpu, disk, dsl, dvb, fingerprint, floppy, framebuffer, gfxcard, hub, ide, isapnp, isdn, joystick, keyboard, memory, modem, monitor, mouse, netcard, network, partition, pci, pcmcia, pcmcia-ctrl, pppoe, printer, redasd, reallyll, scanner, scsi, smp, sound, storage-ctrl, sys, tape, tv, uml, usb, usb-ctrl, vbe, wlan, xen, zip

Como ejemplo obtenemos información sobre la cpu.

```
hwinfo --cpu
```

Como el resultado es excesivamente extenso lo acortamos.

```
hwinfo --cpu --short
```

```
[root@miserver ~]# hwinfo --cpu --short
cpu:
      Intel(R) Xeon(R) CPU E3-1230 v3 @ 3.30GHz, 800 MHz
      Intel(R) Xeon(R) CPU E3-1230 v3 @ 3.30GHz, 804 MHz
      Intel(R) Xeon(R) CPU E3-1230 v3 @ 3.30GHz, 857 MHz
      Intel(R) Xeon(R) CPU E3-1230 v3 @ 3.30GHz, 880 MHz
      Intel(R) Xeon(R) CPU E3-1230 v3 @ 3.30GHz, 1000 MHz
      Intel(R) Xeon(R) CPU E3-1230 v3 @ 3.30GHz, 825 MHz
      Intel(R) Xeon(R) CPU E3-1230 v3 @ 3.30GHz, 833 MHz
      Intel(R) Xeon(R) CPU E3-1230 v3 @ 3.30GHz, 913 MHz
```

Como puedes observar, estamos ante una herramienta muy potente y fácil de usar. Para conocer el resto de opciones puedes revisar su propia ayuda.

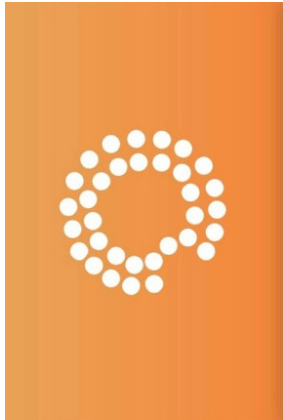
```
hwinfo --help
```

¿Te interesa colaborar
con SoloLinux?

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

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

VestaCP Plugin Manager – El administrador definitivo



VestaCP
Plugin Manager

www.sololinux.es

De la mano de **João Henrique**, nos llega un sorprendente script con la capacidad de agregar a **VestaCP**, una administración de complementos (plugins) realmente buena.

Los plugins pueden ser instalados directamente desde su repositorio de **Github**, o mediante archivos zip. Las principales características de **VestaCP Plugin Manager**, son:

- Agrega automáticamente los ejecutables de los plugins, al directorio bin de VestaCP.
- Inserta de forma predeterminada un nuevo tema de administración, que hace más agradable la experiencia de usuario.
- Instalación y desinstalación muy simple.
- Se permiten hooks para futuras actualizaciones de los complementos.
- Instalar en un dominio en particular las herramientas más conocidas (en estos momentos **WordPress** y **Moodle**).

En este artículo vemos como instalar **VestaCP Plugin Manager** en tu instalación VestaCP, además de unos ejemplos de uso que te sorprenderán.

VestaCP Plugin Manager – El administrador

Si quieres que el administrador de complementos funcione correctamente, necesitamos instalar la biblioteca js, Ejecuta los siguientes comandos dependiendo de tu distribución linux.

Debian y Ubuntu

```
sudo apt-get -y install jq curl
```

CentOS 7

```
sudo yum -y install jq curl
```

Al terminar, ejecutamos el instalador del administrador de complementos.

```
curl -sL https://raw.githubusercontent.com/jhmaverick/vestacp-plugin-manager/master/install.sh | bash -
```

Los cambios realizados afectan a los archivos...

- /usr/local/veda/web/inc/main.php
- /usr/local/veda/web/templates/header.html
- /usr/local/veda/web/templates/admin/panel.html
- /usr/local/veda/web/templates/user/panel.html
- /usr/local/veda/web/templates/admin/list_server_info.html
- /usr/local/veda/web/templates/footer.html

No te preocupes, si algo sale mal, puedes desinstalar la herramienta y retornar al estado original con el siguiente comando.

```
/usr/local/veda/plugin-manager/uninstall.sh
```

Vemos como era antes nuestro VestaCP y como es ahora. Recuerda que por defecto nos instala un nuevo y agradable tema (puedes desactivarlo).

Antes:

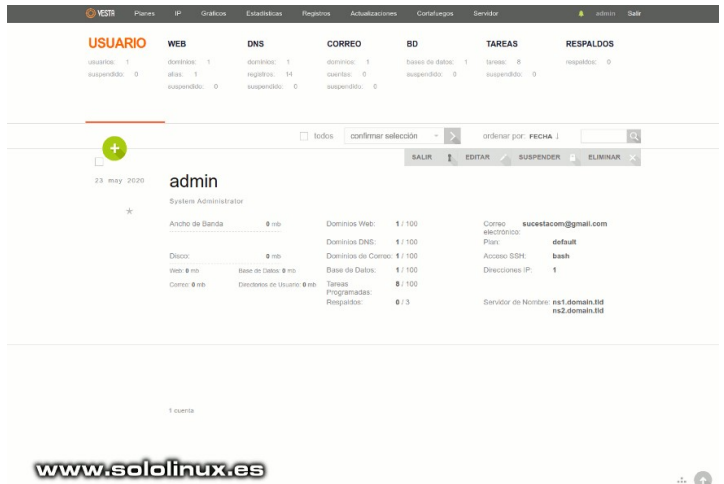


Imagen 37. VestaCP original

Ahora:

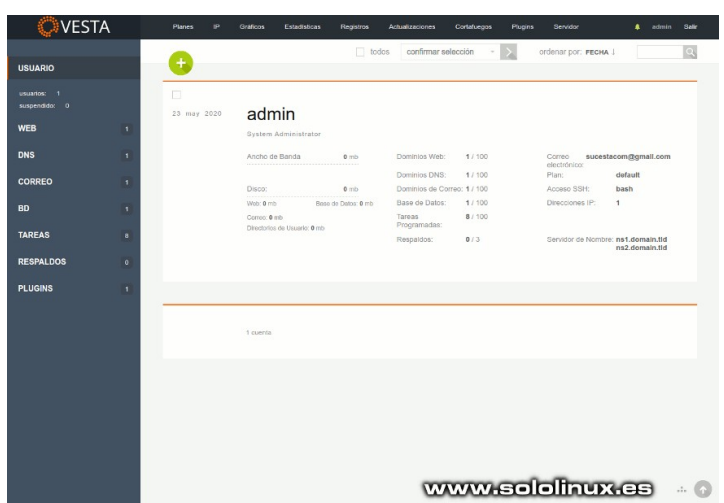


Imagen 38. VestaCP con VestaCP Plugin Manager

Como punto final al artículo, vemos unas imágenes de este fabuloso y sorprendente complemento de VestaCP.

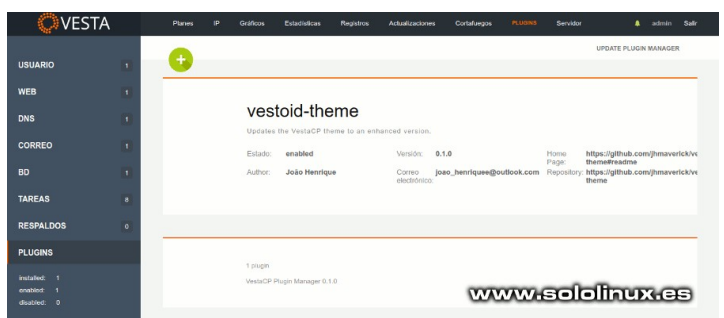


Imagen 39. Complementos instalados en VestaCP desde el plugin manager

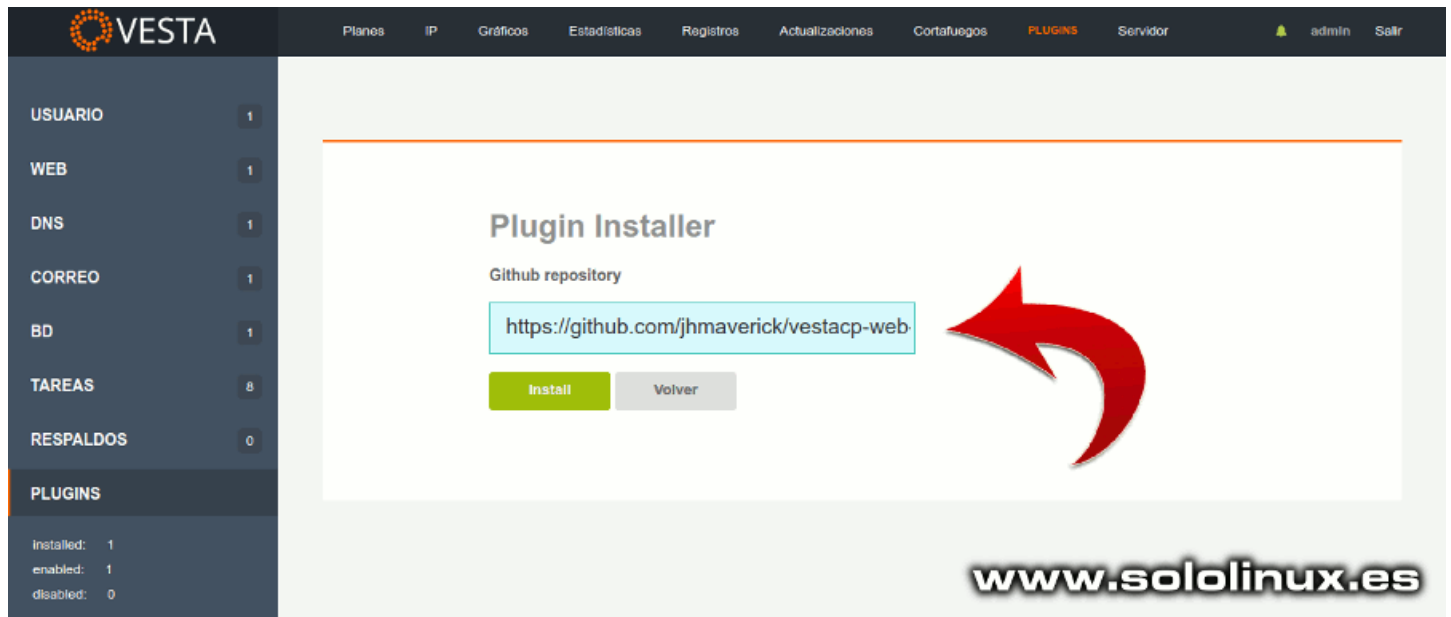


Imagen 40. Instalar plugins en VestaCP desde el administrador



Imagen 41. Instalar WordPress en VestaCP desde el administrador

Instalar Transmission 3.00 en cualquier distribución Linux

Acaban de lanzar, la última versión del cliente **BitTorrent** preferido por muchos usuarios. Hablamos de **Transmission 3.00**.

Transmission es el cliente predeterminada de muchas distribuciones (incluyendo Ubuntu), pero eso no quiere decir que sea el mejor. Existen otras opciones mucho más completas, por ejemplo **qBittorrent**.

A pesar de lo dicho... si buscas un cliente de **archivos torrent**, ligero, sencillo y muy rápido; Transmission es el mejor.

El nuevo **Transmission 3.00** viene con grandes novedades, vemos las más destacadas:

- Permite al servidor RPC escuchar en IPv6.
- Limita los intentos de autenticación incorrectos para evitar ataques de fuerza bruta.
- Agrega ID de pares para Xfplay, PicoTorrent, Free Download Manager, Folx y Baidu Netdisk torrent clients.
- Se añade el soporte para la extensión **TCP_FASTOPEN**, ahora las transmisiones son un poco más rápidas.
- Ahora también soporta mbedtls y wolfssl, LibreSSL
- Teclas de acceso rápido en la cola.
- Mejora considerablemente la interfaz de usuario Qt, en pantallas de alta resolución.
- Y muchas más.

En este artículo vemos como **instalar Transmission 3.00** en tu **distribución linux** favorita.

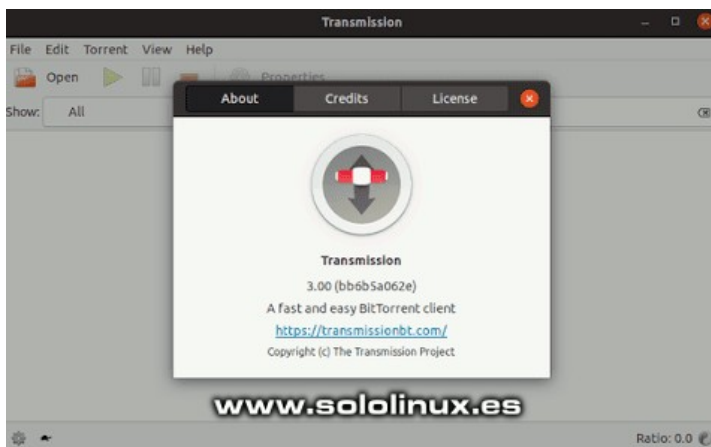


Imagen 42. Instalar Transmission 3.00 en Linux

Instalar Transmission 3.00 en linux

Comenzamos la instalación o actualización de **Transmission 3.00** (es posible que tu distro aún no haya agregado el nuevo transmission, si es tu caso ves al final del artículo para **instalar Transmission** de forma manual desde el código).

Ubuntu, Linux Mint y derivados

```
sudo add-apt-repository ppa:transmissionbt/ppa
sudo apt update
sudo apt full-upgrade
```

Fedora y derivados

```
sudo dnf install transmission
```

Open Suse y derivados

```
# Si es necesario, modifica por tu version de
OpenSuse
zypper addrepo
repositories/openSUSE:Leap:15.1/standard/openSUSE:Le
ap:15.1.repo
zypper refresh
# Version gtk
zypper install transmission-gtk
# Version Qt
zypper install transmission-qt
```

CentOS y derivados

```
sudo yum install epel-release
sudo yum update
sudo yum install transmission transmission-daemon
```

Debian

```
sudo apt-get install transmission
```

Arch Linux, Manjaro y derivados

```
# Version gtk
sudo pacman -S transmission-gtk
# Version Qt
sudo pacman -S transmission-qt
```

Gentoo

```
emerge -avq transmission
/etc/init.d/transmission-daemon start
```

Instalar Transmission 3.00 desde código fuente

Como ya comentamos anteriormente, es posible que aún no hayan insertado la nueva versión en los repositorios de tu distribución linux. La solución es sencilla, lo hacemos de manera manual.

```
wget
https://github.com/transmission/transmission/release
s/download/3.00/transmission-3.00.tar.xz
tar xf transmission-3.00.tar.xz
cd transmission-3.00
mkdir build
cd build
cmake
make
sudo make install
```

También tienes la opción de clonar su repositorio oficial.

```
git clone
https://github.com/transmission/transmission
Transmission
cd Transmission
git submodule update --init
mkdir build
cd build
cmake
make
sudo make install
```

Como encriptar un Pendrive con VeraCrypt en linux



VeraCrypt es una aplicación gratuita, con la que podrás encriptar cualquier dispositivo de almacenamiento conectado a tu **sistema linux**. En este artículo nos centramos en los pendrives (minipen).

No viene preinstalado de manera predeterminada en ninguna distribución linux, tampoco en sus repositorios oficiales. Aun así, **instalar VeraCrypt** para encriptar un pendrive es tarea bien sencilla. Oficialmente, **VeraCrypt** admite **Debian**, **Ubuntu**, **Fedora**, **OpenSUSE**, así como su código fuente.

Antes de ver **como encriptar un Pendrive con VeraCrypt en linux**, debes tener presente que se requiere borrar todas las particiones del dispositivo, si no es así... pueden producirse errores.

Como encriptar un Pendrive con VeraCrypt

Comenzamos con la instalación de la herramienta, en diferentes distribuciones linux.

Instalar VeraCrypt Ubuntu

```
# Ubuntu 20.04
wget https://launchpad.net/veracrypt/trunk/1.24-update4/+download/veracrypt-1.24-Update4-Ubuntu-20.04-amd64.deb -O /tmp/veracrypt-1.24-Update4-Ubuntu-20.04-amd64.deb
sudo apt install /tmp/veracrypt-1.24-Update4-Ubuntu-20.04-amd64.deb
# Ubuntu 18.04
wget https://launchpad.net/veracrypt/trunk/1.24-update4/+download/veracrypt-1.24-Update4-Ubuntu-18.04-amd64.deb -O /tmp/veracrypt-1.24-Update4-Ubuntu-18.04-amd64.deb
sudo apt install /tmp/veracrypt-1.24-Update4-Ubuntu-18.04-amd64.deb
# Ubuntu 16.04
wget https://launchpad.net/veracrypt/trunk/1.24-update4/+download/veracrypt-1.24-Update4-Ubuntu-16.04-amd64.deb -O /tmp/veracrypt-1.24-Update4-Ubuntu-16.04-amd64.deb
sudo apt install /tmp/veracrypt-1.24-Update4-Ubuntu-16.04-amd64.deb
```

Debian

```
# Debian 10
wget https://launchpad.net/veracrypt/trunk/1.24-update4/+download/veracrypt-1.24-Update4-Debian-10-amd64.deb -O /tmp/veracrypt-1.24-Update4-Debian-10-amd64.deb
sudo dpkg -i /tmp/veracrypt-1.24-Update4-Debian-10-amd64.deb
sudo apt-get install -f
```

```
# Debian 9
wget https://launchpad.net/veracrypt/trunk/1.24-update4/+download/veracrypt-1.24-Update4-Debian-9-amd64.deb -O /tmp/veracrypt-1.24-Update4-Debian-9-amd64.deb
sudo dpkg -i /tmp/veracrypt-1.24-Update4-Debian-9-amd64.deb
sudo apt-get install -f
```

Arch Linux / Manjaro

```
sudo pacman -S veracrypt
# o
sudo pacman -Rs veracrypt
```

CentOS / Fedora

```
# CentOS 8 / Fedora
wget https://launchpad.net/veracrypt/trunk/1.24-update4/+download/veracrypt-1.24-Update4-CentOS-8-x86_64.rpm -O /tmp/veracrypt-1.24-Update4-CentOS-8-x86_64.rpm
cd /tmp
sudo dnf install veracrypt-1.24-Update4-CentOS-8-x86_64.rpm
# CentOS 7
wget https://launchpad.net/veracrypt/trunk/1.24-update4/+download/veracrypt-1.24-Update4-CentOS-7-x86_64.rpm -O /tmp/veracrypt-1.24-Update4-CentOS-7-x86_64.rpm
cd /tmp
sudo yum install veracrypt-1.24-Update4-CentOS-7-x86_64.rpm
```

Open Suse

```
# Open Suse 15/15.1
wget https://launchpad.net/veracrypt/trunk/1.24-update4/+download/veracrypt-1.24-Update4-openSUSE-15-x86_64.rpm -O /tmp/veracrypt-1.24-Update4-openSUSE-15-x86_64.rpm
cd /tmp
sudo zypper install veracrypt-1.24-Update4-openSUSE-15-x86_64.rpm
```

Como usar VeraCrypt

Ahora vemos como **crear un pendrive encriptado**, además lo haremos de forma interactiva. Tan solo tienes que seguir la secuencia de imágenes.

Al abrir la herramienta, lo primero que debemos hacer es crear un nuevo volumen.

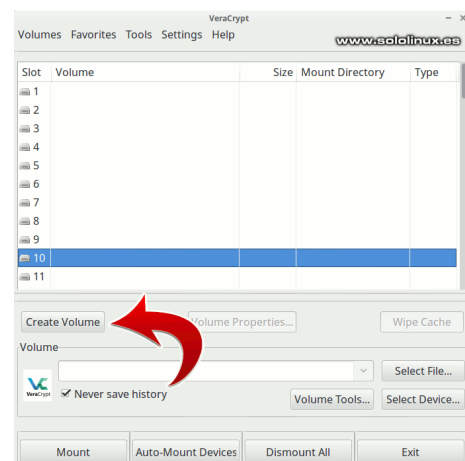


Imagen 43. Crear volumen en VeraCrypt

Marcamos la opción «Crear volumen en partición».



Imagen 44. Crear volumen en partición

Una vez tengamos seccionado el volumen en la partición, hacemos click en continuar.



Imagen 47. Pulsamos en continuar

En la siguiente pantalla, selecciona «Volumen VeraCrypt estándar».

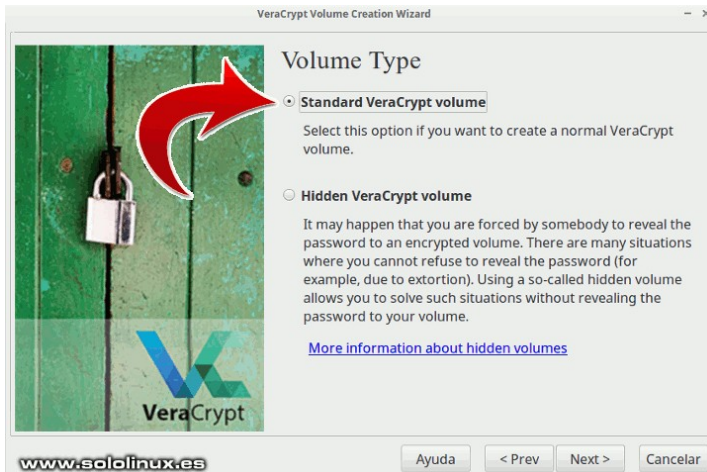


Imagen 45. Elegimos Volumen VeraCrypt estándar

Ahora, seleccionamos el algoritmo de encriptación deseado.



Imagen 48. Elegimos el algoritmo de encriptación

Seleccionamos el dispositivo.

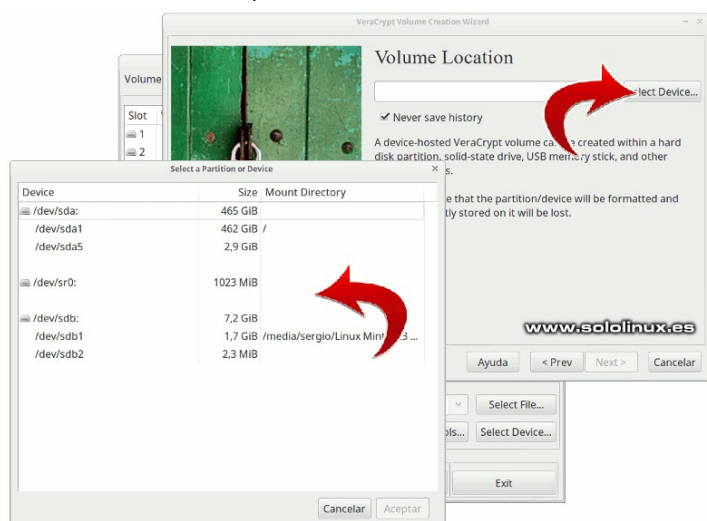


Imagen 46. Seleccionar el dispositivo a encriptar

Necesitamos una contraseña para poder desencriptar el dispositivo.



Imagen 49. Password

También es necesario indicar el tipo de formato de archivos, normalmente Fat o Ext4.

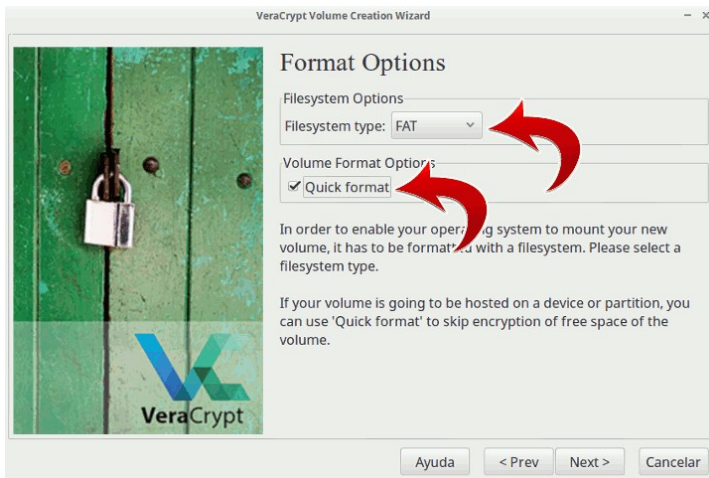


Imagen 50. Tipo de formato

Para concluir el proceso y el artículo «**Como encriptar un Pendrive con VeraCrypt en linux**», solo necesitas pulsar el botón «FORMAT». Felicidades, ya tienes tu pendrive encriptado.

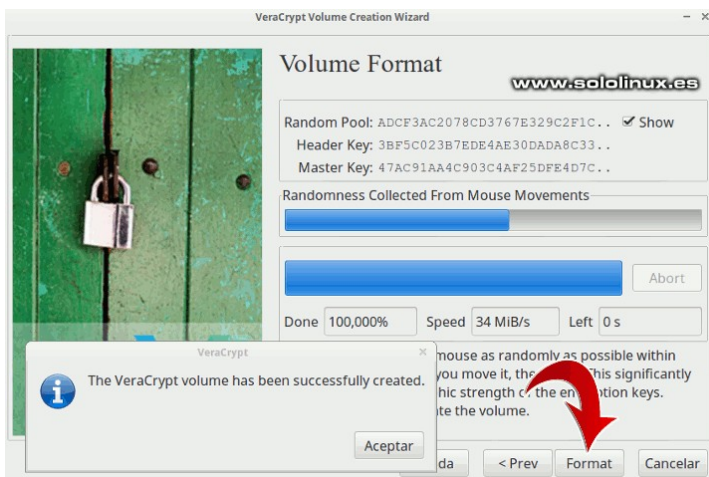


Imagen 51. Pulsa en formatear

SoloWordpress

- Revista Bimestral
- Manuales
- Noticias
- Temas
- Plugins
- SEO
- Seguridad
- Entra en: www.solowordpress.es



Limitar los recursos de procesos con systemd en linux

Por defecto, **systemd** establece unos límites predeterminados a los procesos que se inician. En sistemas con grandes cargas de trabajo, es posible que este límite sea demasiado pequeño.

Algunas aplicaciones o procesos específicos, pueden necesitar unos valores de recursos más altos que los asignados por el sistema. Por suerte este problema tiene solución, además bastante simple.

En el artículo de hoy veremos como aumentar este límite (a nivel global), o reducirlo si es necesario. También aprenderemos a realizar la misma operación, pero... a un proceso específico.

Limitar los recursos de procesos con systemd

Normalmente, la regla que permite aumentar el valor global viene comentada de forma predeterminada. Puedes verificarlo con el siguiente comando.

```
cat /etc/systemd/system.conf | grep
DefaultLimitMEMLOCK
```

ejemplo de salida...

```
sololinux # cat /etc/systemd/system.conf | grep
DefaultLimitMEMLOCK
#DefaultLimitMEMLOCK=
```

Ahora modificamos / agregamos un valor.

```
sudo nano /etc/systemd/system.conf
```

Busca la línea «**#DefaultLimitMEMLOCK=**».

```
#DefaultMemoryAccounting=no
#DefaultTasksAccounting=no
#DefaultTasksMax=
#DefaultLimitCPU=
#DefaultLimitFSIZE=
#DefaultLimitDATA=
#DefaultLimitSTACK=
#DefaultLimitCORE=
#DefaultLimitRSS=
#DefaultLimitNOFILE=
#DefaultLimitAS=
#DefaultLimitNPROC=
#DefaultLimitMEMLOCK=
#DefaultLimitLOCKS=
#DefaultLimitSIGPENDING=
#DefaultLimitMSGQUEUE=
#DefaultLimitNICE=
#DefaultLimitRTPRIO=
#DefaultLimitRTTIME=
```

www.sololinux.es

Imagen 52. Limitar los recursos de procesos en DefaultLimitMEMLOCK

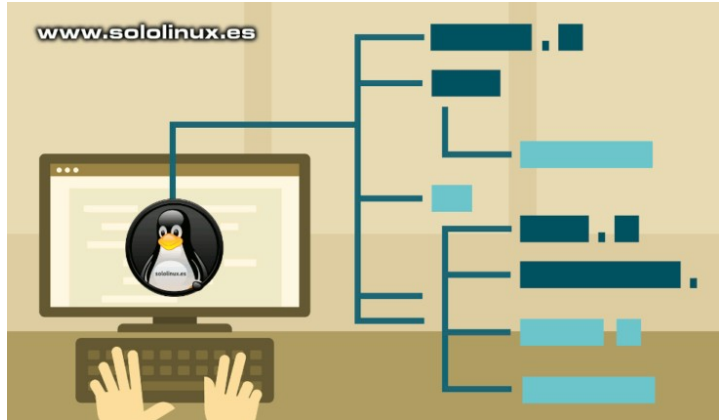
Descomenta la línea y agregas un valor, por ejemplo si quisiéramos poner una limitación general de 128k...

```
#DefaultLimitNOFILE=
#DefaultLimitAS=
#DefaultLimitNPROC=
DefaultLimitMEMLOCK=128000
#DefaultLimitLOCKS=
#DefaultLimitSIGPENDING=
#DefaultLimitMSGQUEUE=
```

Guarda el archivo, cierra el editor y reinicia el sistema.

```
sudo reboot
```

Si queremos limitar o aumentar un proceso específico, la forma de operar es un poco diferente. Primero lo identificamos, así que ejecuta el comando **systemctl**.



```
sudo systemctl
```

En nuestro ejemplo queremos modificar el **sshd.service**. Ejecutamos status para identificar su ruta.

```
sudo systemctl status sshd.service
```

Identificamos la ruta en: **/usr/lib/systemd/system/sshd.service** (puede variar dependiendo de tu distribución linux).

```
[root@62 ~]# systemctl status sshd.service
● sshd.service - OpenSSH server daemon
   Loaded: loaded
   ( /usr/lib/systemd/system/sshd.service; enabled;
   vendor preset: enabled)
   Active: active (running) since sáb 2020-05-23
   07:03:25 CEST; 3 days ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 1066 (sshd)
    CGroup: /system.slice/sshd.service
           └─1066 /usr/sbin/sshd -D
```

Editamos el archivo del servicio que nos interesa.

```
sudo nano /usr/lib/systemd/system/sshd.service
```

Nos aparece algo similar a...

```
[Unit]
Description=OpenSSH server daemon
Documentation=man:sshd(8) man:sshd_config(5)
After=network.target sshd-keygen.service
Wants=sshd-keygen.service
[Service]
Type=notify
EnvironmentFile=/etc/sysconfig/ssh
ExecStart=/usr/sbin/sshd -D $OPTIONS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartSec=42s
[Install]
WantedBy=multi-user.target
```

Lo que haremos es agregar la siguiente línea (a continuación de ExecStart).

```
LimitMEMLOCK=128000
```


El resultado será más o menos así.

```
[Unit]
Description=OpenSSH server daemon
Documentation=man:sshd(8) man:sshd_config(5)
After=network.target sshd-keygen.service
Wants=sshd-keygen.service
[Service]
Type=notify
EnvironmentFile=/etc/sysconfig/ssh
ExecStart=/usr/sbin/sshd -D $OPTIONS
LimitMEMLOCK=128000
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartSec=42s
[Install]
WantedBy=multi-user.target
```

Guarda el archivo, cierra el editor y reinicia el sistema.

```
reboot
```

Este manual ha sido verificado en: CentOS 7, Ubuntu 18.04 y Linux Mint 18.3. Limitar los recursos de procesos con systemd en linux.

¿Te interesa colaborar
con SoloLinux?

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

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

Eliminar directorios vacíos con un script bash

A base de instalar y desinstalar aplicaciones, es común que nuestro sistema se llene de carpetas (directorios) vacías. Es cierto que ocupan poco espacio de nuestro **disco**, pero también es cierto que son muy molestas en ciertos momentos, por ejemplo cuando buscamos un archivo en particular, o listamos los directorios de nuestro sistema.

En este artículo (de los que a mí me gustan), te presento un **script bash** con la capacidad de identificar los directorios vacíos y borrarlos (eliminar directorios vacíos), si tú se lo pides. Es bastante seguro, dado que aparte que tienes que indicar donde buscar, también te solicita permiso. Deberás responder YES o NO.

Eliminar directorios vacíos con un script bash
Creamos nuestro script para **eliminar directorios**.

```
nano borrard.sh
```

Copia y pega el siguiente código.

```
#!/bin/bash
#Check if user input parameter, if not ask to enter directory
if [ x"$1" = "x" ]; then
#Ask user to input directory where to start search for empty
directories.
echo -n "Introduce el directorio donde quieres eliminar las carpetas
vacías: "

#we read input
while read dir

do

#we check if input empty
test -z "$dir" && {

#if input empty - we ask once more to input directory
echo -n "Introduce el directorio: "

continue
}

#if entered no empty data - continue to do other things
break

done

#if user entered parameter do next:
else
#dirname will be passed parameter
dir=$1
fi
#this check if directory exist, exit if not
if [ ! -d $dir ]; then
echo "Este directorio no existe"
exit 1
fi

#We will store list of all directories in temporary file
DirList=/tmp/ditlist.tmp

# we search for all directories
find $dir -type d > $DirList

#writing all directories to variable
dirs=`cat $DirList`

#start checking every directory
for dir in $dirs

do

#we are checking if directory is empty
[ `ls $dir | wc -l` -lt 1 ] || continue
#this ask user if really delete directory
echo -n "Quieres eliminar el directorio vacio $dir: [No/yes] "
#reading users answer:
read answer
#Checing answer, if yes - we will delete folder, nothing in other
case:
if [ "$answer" = "yes" ]; then
rmdir "$dir"
fi
done

#Eliminar directorios vacios
```

Guarda el script y cierra el editor.

Para no tener problemas de permisos, es recomendable que siempre ejecutes este script bash con privilegios de administrador. Ahora vemos un ejemplo real en un servidor demo, no reproduzcas la ruta o te quedarás sin sistema operativo.

```
sudo bash borrard.sh
```

Nos pregunta el directorio del sistema (o cualquier otro), donde queremos buscar y borrar los directorios que están vacíos.

```
[root@62 ~]# sudo bash borrard.sh
Introduce el directorio donde quieres eliminar las
carpetas vacías:
```

Como es un demo y somos más chulos que un ocho, borramos los directorios vacíos de «/etc» (no lo hagas, solo es una demo). Ten cuidado al eliminar directorios vacíos.

Continuamos con nuestro ejemplo.

```
[root@62 ~]# bash borrard.sh
Introduce el directorio donde quieres eliminar las
carpetas vacías: /etc
```

Observa como borramos todos los directorios vacíos...

```
[root@62 ~]# sudo bash borrard.sh
Introduce el directorio donde quieres eliminar las carpetas vacías: /
etc
Quieres eliminar el directorio vacio
/etc/smartmontools/smartd_warning.d: [No/yes] yes
Quieres eliminar el directorio vacio /etc/tuned/recommend.d: [No/yes]
yes
Quieres eliminar el directorio vacio /etc/modules-load.d: [No/yes]
yes
Quieres eliminar el directorio vacio /etc/ppp/peers: [No/yes] y
Quieres eliminar el directorio vacio /etc/xdg/autostart: [No/yes] yes
Quieres eliminar el directorio vacio /etc/selinux/targeted/logins:
[No/yes] yes
Quieres eliminar el directorio vacio
/etc/selinux/targeted/active/modules/disabled: [No/yes] yes
Quieres eliminar el directorio vacio
/etc/selinux/targeted/modules/active/modules: [No/yes] yes
Quieres eliminar el directorio vacio /etc/selinux/tmp: [No/yes] yes
Quieres eliminar el directorio vacio /etc/selinux/final: [No/yes] yes
Quieres eliminar el directorio vacio /etc/multipath: [No/yes] yes
Quieres eliminar el directorio vacio /etc/gnupg: [No/yes] yes
Quieres eliminar el directorio vacio /etc/gss/mech.d: [No/yes] yes
Quieres eliminar el directorio vacio /etc/skel: [No/yes] yes
rmdir: fallo al borrar «/etc/skel»: El directorio no está vacío
Quieres eliminar el directorio vacio /etc/tmpfiles.d: [No/yes] yes
Quieres eliminar el directorio vacio /etc/systemd/user: [No/yes] yes
Quieres eliminar el directorio vacio /etc/udev/rules.d: [No/yes] yes
Quieres eliminar el directorio vacio /etc/krb5.conf.d: [No/yes] yes
Quieres eliminar el directorio vacio /etc/dracut.conf.d: [No/yes] yes
Quieres eliminar el directorio vacio /etc/lvm/archive: [No/yes] yes
Quieres eliminar el directorio vacio /etc/lvm/cache: [No/yes] yes
Quieres eliminar el directorio vacio /etc/lvm/backup: [No/yes] yes
Quieres eliminar el directorio vacio /etc/xinetd.d: [No/yes] yes
Quieres eliminar el directorio vacio /etc/glvnd/egl_vendor.d:
[No/yes] yes
Quieres eliminar el directorio vacio /etc/sysconfig/console: [No/yes]
yes
Quieres eliminar el directorio vacio /etc/sysconfig/modules: [No/yes]
yes
Quieres eliminar el directorio vacio /etc/chkconfig.d: [No/yes] yes
#
# Eliminar directorios vacios
```

Al recorrer todo el árbol de directorios indicado, el sistema volverá al usuario. Ten mucho cuidado al eliminar directorios, algunos pueden ser necesarios para un correcto funcionamiento del sistema.

Bloquear el acceso a los usuarios que utilizan la red Tor



Variados son los motivos, por los que algunos usuarios navegan a través de la **red TOR**. Uno de ellos, el más peligroso y poco ético, es proteger su anonimato para realizar tareas de tipo **hacking** sobre **servidores** web, u otros sistemas.

Si tú quieres curarte en salud, y bloquear todos los accesos que provengan de la **red TOR**, lo puedes lograr con un sencillo script fácil de implantar. Lo vemos.

Bloquear el acceso a usuarios que utilizan la red Tor
Para lograr nuestro objetivo tomaremos las **listas TOR**, que nos ofrece **dan-me-uk**. El funcionamiento es simple, descargamos la lista de IP y la agregamos a **iptables**.

Antes de comenzar, debes tener presente que estas **listas de Tor se actualizan cada 30 minutos**; y ese es el tiempo mínimo entre petición y petición del listado por ip. Si solicitas las **ip de Tor** antes de 30 minutos, el script no funcionara pues el archivo estará vacío. **NO LO OLVIDES**.

Creamos el script:

```
sudo nano blocktor.sh
```

Copia y pega lo siguiente...

```
#!/bin/bash
##script blocktor.sh
##by Sergio G.B. sololinux.es
cd /tmp/
echo -e "\n\tobtener lista de nodos TOR a bloquear\
n"
wget -q -O - "https://www.dan.me.uk/torlist/" -U
SXTorBlocker/1.0 > /tmp/tor.txt
sed -i 's|^#.*$||g' /tmp/tor.txt
iptables -F
for IP in `cat /tmp/tor.txt`
do
    echo $IP
    iptables -I INPUT -s $IP -j DROP
done
service iptables save
```

Guarda el archivo y cierra el editor.

Lo ejecutas:

```
sudo bash blocktor.sh
```

ejemplo de salida....

```
[root@sololinux ~]# bash blocktor.sh
1.171.37.19
1.43.247.217
1.9.116.33
100.14.156.2
100.14.23.208
100.15.139.123
100.15.249.55
100.18.5.33
100.2.182.112
100.33.64.89
100.8.164.39
etc.....
```

El proceso puede tardar unos segundos, la lista es grande. Al terminar verifica que todas las ip han sido añadidas y bloqueadas en iptables.

```
iptables -L
```

Ejemplo

```
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
DROP      all  --  186.120.250.217        anywhere
DROP      all  --  pt14.wesenddaily.com   anywhere
DROP      all  --  metaverse.metadomain.se anywhere
DROP      all  --  185.90.60.73           anywhere
DROP      all  --  185.86.151.168         anywhere
DROP      all  --  plus.com               anywhere
DROP      all  --  mx1.scaleclubrl.com    anywhere
DROP      all  --  185.86.150.133         anywhere
DROP      all  --  185.86.148.90          anywhere
DROP      all  --  185.86.148.79          anywhere
DROP      all  --  challenger240.mydhp.de anywhere
DROP      all  --  domain.com             anywhere
DROP      all  --  tornode2.vds-114833.com anywhere
DROP      all  --  canis2.uaservers.net   anywhere
DROP      all  --  dedi20311.hostsailor.com anywhere
DROP      all  --  185.82.126.30          anywhere
DROP      all  --  185.82.126.249         anywhere
DROP      all  --  185.82.126.215         anywhere
DROP      all  --  185.82.126.190         anywhere
DROP      all  --  185.82.126.137         anywhere
DROP      all  --  185.82.126.123         anywhere
DROP      all  --  se.cyberabwehrzentrum.at anywhere
DROP      all  --  vibhatsu.91jobsemails.com anywhere
DROP      all  --  ayodhya.91jobsemails.com anywhere
DROP      all  --  static-38.63.8.185.cloudhosting.lv anywhere

DROP      all  --  calnzombie.the-goober.net anywhere
```

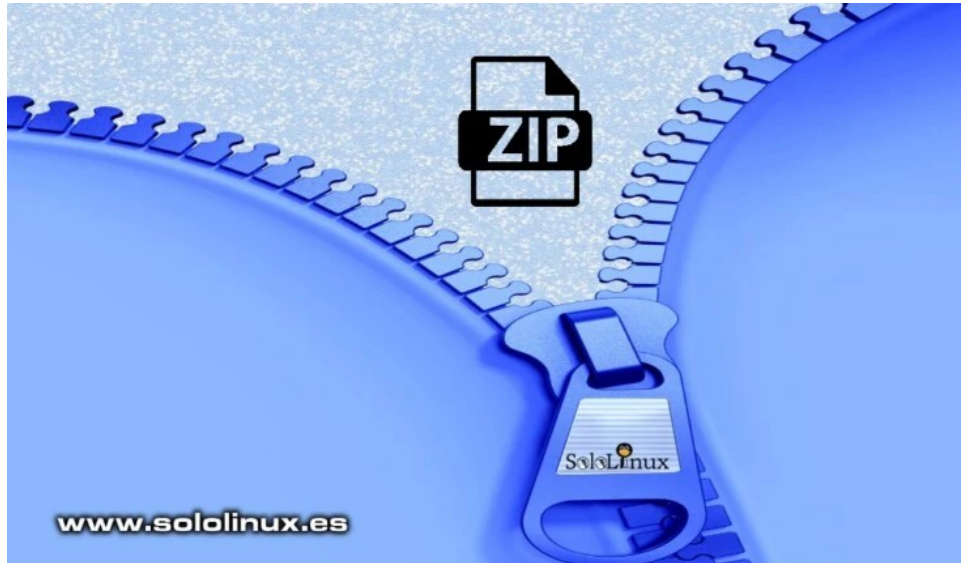
Lo ideal es que ejecutes el script mediante una tarea cron, te recomiendo una vez cada hora. Las ip van cambiando constantemente.

Comprimir archivos con zip y protegerlos con contraseña

Zip es una herramienta utilizada para empaquetar y comprimir datos. Está disponible para los sistemas operativos más usados, incluyendo **Linux**. Ampliamente usada para hacer **copias de seguridad**, tiene la capacidad de ahorrarnos espacio en nuestro disco.

Además, los **archivos zip** son fáciles de almacenar y compartir. Sin embargo, cualquier persona que tenga acceso a ellos puede descomprimirlos y ver su contenido. Para evitar este problema, es conveniente proteger la información contenida en el **archivo zip** mediante una contraseña.

En este artículo vemos como **comprimir y proteger archivos zip**, implantando una contraseña en los mismos.



Comprimir archivos con zip y protegerlos con contraseña

Crear un archivo comprimido protegido por contraseña es una tarea sencilla, pero primero recordamos como realizar una compresión simple, es así de fácil.

```
zip archivo-comprimido.zip archivo-original.jpg
```

El resultado puede ser visionado por cualquier usuario, para evitarlo insertamos una contraseña (en nuestro ejemplo 123456).

```
zip -P 123456 archivo-comprimido.zip archivo-original.jpg
```

ejemplo...

```
sergio@sololinux $ zip -P 123456 imagen.zip imagen.jpg
adding: imagen.jpg (deflated 10%)
```

Verdad que es sencillo.

Para descomprimir el archivo protegido ejecutamos el siguiente comando.

```
unzip imagen.zip
```

¿Te interesa colaborar
con SoloLinux?

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

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

Solución: Error Minimal BASH like line editing is supported



Independientemente de la **distribución linux** que utilices, es bastante común, que alguna vez tengas un **error Minimal BASH like line editing is supported**. No entres en pánico, tu sistema y archivos siguen estando en su sitio. Tranquilo... que tiene solución.

Por algún motivo desconocido, el **GRUB** está corrupto. Las causas pueden ser muchas, así que me resulta imposible ofrecer una explicación razonable, para ello deberías revisar los registros una vez hayamos reparado el **Grub**.

Tal vez la palabra **Grub** te pueda asustar, relájate, es muy fácil si sigues los pasos que te indico.

Solución: Error Minimal BASH like line editing is supported

Lo primero que debes conocer es, si tu sistema utiliza **Bios** o **Uefi**. Si tienes un sistema moderno, probablemente usas **Uefi**. De todas maneras expondremos la solución a los dos casos.

Iniciamos el sistema desde una **Live USB**, preferiblemente con la misma distro que tenemos instalada (no es obligatorio, pero si recomendable).

Es importante identificar las particiones del sistema, además es posible que localices la **partición UEFI**.

```
sudo fdisk -l
```

ejemplo...

Device	Start	End	Sectors	Size	Type
/dev/sda1	2048	976895	974848	476M	EFI System

Si usas **Uefi**, normalmente la partición raíz es la siguiente a **EFI**, en nuestro caso **/dev/sda2**. La montamos.

```
sudo mount /dev/sda2 /mnt
```

También montamos la partición de la **Uefi**.

```
sudo mount /dev/sda1 /mnt/boot/efi
```

Instalamos de nuevo el **Grub**.

```
sudo grub-install --boot-directory=/mnt/boot --
bootloader-id=ubuntu --target=x86_64-efi --efi-
directory=/mnt/boot/efi
```

Una vez termine, solo falta reiniciar el sistema.

```
sudo reboot
```

Si eres como yo, de los que prefieren usar **Bios** la cosa cambia. Mucho más simple. Después de identificar la partición raíz (en nuestro ejemplo **/dev/sda1**), montamos la partición e instalamos el **Grub**. Así de simple.

```
sudo mount /dev/sda1 /mnt
sudo grub-install /dev/sd1
```

Para terminar la reparación del error, reiniciamos el sistema.

```
sudo reboot
```

SoloWordpress

Revista Bimestral

Manuales

Noticias

Temas

Plugins

SEO

Seguridad

Entra en: www.solowordpress.es



Uso del comando uptime - Tiempo de actividad del sistema

Para un **administrador de sistemas**, cualquier dato que aporte el servidor es importante. Como no podía ser menos, conocer el tiempo de actividad (real), de un **sistema linux** es imprescindible; si quieres tener un control absoluto... es obligatorio.

El **comando uptime** es muy básico, aun así nos aportara importante información sobre el tiempo de actividad de nuestro **sistema linux**. Vemos como operar con él.

Ayúdanos a crecer, vota por SoloLinux en la siguiente encuesta.
Gracias por colaborar.



Uso del comando uptime - Tiempo de actividad

La sintaxis de este comando no puede ser más simple.

```
uptime [opcion]
```

Las opciones permitidas son pocas, pero potentes.

- -p / --pretty : Tiempo de actividad formateado.
- -s / --since : Fecha y hora de inicio del sistema.
- -V / --version : Versión instalada de uptime.
- -h / --help : Ayuda simple.

Ahora vemos algunos ejemplos de uso.

Comando uptime

```
uptime
```

salida de ejemplo...

```
[root@demo ~]# uptime
11:34:49 up 1 day, 22:44, 1 user, load average:
0,10, 0,10, 0,09
```

Comando uptime -p

```
uptime -p
```

salida de ejemplo...

```
[root@demo ~]# uptime -p
up 1 day, 22 hours, 47 minutes
```

Comando uptime -s

```
uptime -s
```

Impresión en pantalla demo

```
[root@demo ~]# uptime -s
2020-05-29 12:50:03
```

Comando uptime -V

```
uptime -V
```

salida de ejemplo...

```
[root@demo ~]# uptime -V
uptime from procs-ng 3.3.10
```

Comando uptime -h

```
uptime -h
```

ejemplo de salida...

```
[root@demo ~]# uptime -h
Usage:
uptime [options]
Options:
-p, --pretty    show uptime in pretty format
-h, --help      display this help and exit
-s, --since      system up since
-V, --version    output version information and exit
```

Como hemos podido comprobar, uptime es una herramienta muy simple pero de gran utilidad.

Habilitar el servicio ssh en Ubuntu 20.04 y derivados

Ubuntu tiene en sus repositorios **OpenSSH**. Este conjunto de utilidades de red segura, está basado en el protocolo **Secure Shell (SSH)**.

Como norma general el cliente viene preinstalado por defecto, pero no así es servidor. Tal vez te preguntes para que sirve el **servidor ssh**, la respuesta es simple... por ejemplo para iniciar sesión de forma remota y otras comunicaciones especiales en red. En este mini artículo, vemos como **habilitar el servicio ssh en Ubuntu y derivados**.

Ayúdanos a crecer, vota por SoloLinux en la siguiente encuesta.
Gracias por colaborar.

Habilitar el servicio ssh en Ubuntu 20.04

Es tan sencillo como ejecutar el siguiente comando.

```
sudo apt install ssh
```

Verás algo similar a...

```
sololinux # apt install ssh
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  openssh-server openssh-sftp-server
Paquetes sugeridos:
  rssh molly-guard monkeysphere
Paquetes recomendados:
  ncurses-term ssh-import-id
Se instalarán los siguientes paquetes NUEVOS:
  openssh-server openssh-sftp-server ssh
0 actualizados, 3 nuevos se instalarán, 0 para
eliminar y 0 no actualizados.
Se necesita descargar 381 kB de archivos.
Se utilizarán 1.113 kB de espacio de disco adicional
después de esta operación.
¿Desea continuar? [S/n]
```

Introduces «Y» para completar la instalación.
Automáticamente se instala y habilita el servicio. Lo verificamos.

```
systemctl status ssh.service
```

ejemplo de salida correcta...

```
root@sololinux:~# systemctl status ssh.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service;
   enabled; vendor preset: enab
   Active: active (running) since Sun 2020-05-31
13:31:34 BST; 18min ago
   Process: 523 ExecStartPre=/usr/sbin/sshd -t
(code=exited, status=0/SUCCESS)
   Main PID: 525 (sshd)
     Tasks: 5 (limit: 4915)
    CGroup: /system.slice/ssh.service
            └─525 /usr/sbin/sshd -D
              └─555 sshd: root@pts/0
                └─569 -bash
                  └─860 systemctl status ssh.service
                   └─861 systemctl status ssh.service
```



Recuerda que puedes modificar sus opciones, incluyendo el puerto por defecto (22).

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

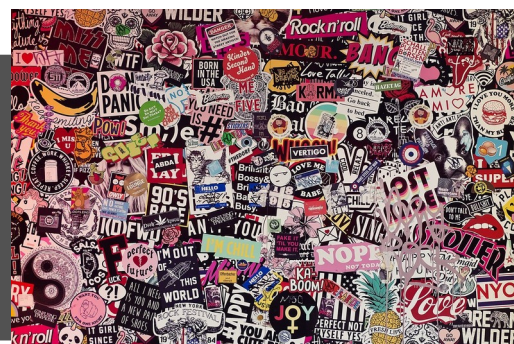
```
# $OpenBSD: sshd_config,v 1.101 2017/03/14
07:19:07 djm Exp $
# This is the sshd server system-wide configuration
file. See
# sshd_config(5) for more information.
# This sshd was compiled with
PATH=/usr/bin:/bin:/usr/sbin:/sbin
# The strategy used for options in the default
sshd_config shipped with
# OpenSSH is to specify options with their default
value where
# possible, but leave them commented. Uncommented
options override the
# default value.
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key
# Ciphers and keying
#RekeyLimit default none
```

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



?Te interesa colaborar con SoloLinux?



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

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

