Visita nuestro sitio SoloLinux.es

MAGAZINE SOLO LINUX

Nº 24

Tu revista, la revista de tod@s

ENERO 2021

Cómo actualizar **sudo** en Linux

Uso del comando **history** en Linux

Preguntas y respuestas sobre puertos en Linux

Linux Mint 20.1 Ulyssa
Listo para su descarga

Instalar **Luminance HDR 2.6.1.1** en Linux

Wifislax 2.4 64bits
El Linux forense español

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

EDITORIAL

BIENVENIDO A LA REVISTA SOLOLINUX

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

Os presentamos el número 24 de la **Revista SoloLinux**. Si con este número **CUMPLIMOS DOS AÑOS**.

Tras unos meses, casi un año un poco complicados para todos, seguimos al pie del cañón, intentando dar lo mejor de nosotros. Este numero lo quiero dedicar a todo el mundo, en especial a las familias y personas que han pasado y siguen pasando por el dichoso virus este que nos azota día a día Maldito COVID19.

Solo nos queda esperar un poco mas y ver si las vacunas hacen su efecto. Mucha fuerza y animo para todos. Respeten las distancias, usen mascarilla y sigan adelante.

Gracias a todos por seguir leyendo nuestros números de la revista y nuestras WEBS.

Para colaborar o poner publicidad en la revista solo envianos un Email a adrian@sololinux.es y te diremos como.

Equipo **SOLOLINUX**

www.sololinux.es

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

Tu revista, la revista de todos.

Esta revista es de **distribución gratuita**, si lo consideras oportuno puedes ponerle precio.
Tu también puedes ayudar, contamos con la posibilidad de hacer donaciones para la REVISTA, de manera muy simple a través de **PAYPAL**

AYUDANOS A SEGUIR CRECIENDO

PayPal
Apoyo con una donación

Síguenos en las Redes:

La revista SOLOLINUX esta realizada con **Libre Office Impress 7.0.0.3**

Editorial

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

Redacción

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

Diseño Portada

- **Karina Fernández**
  @karyfernandez.design

Agradecimientos

- **Erwin Andres,** Admin de Espacio Tecnológico por dedicarnos algo de su tiempo en la entrevista.

Publicidad

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

CON SOLOLINUX MULTIPLICARAS TUS CLIENTES

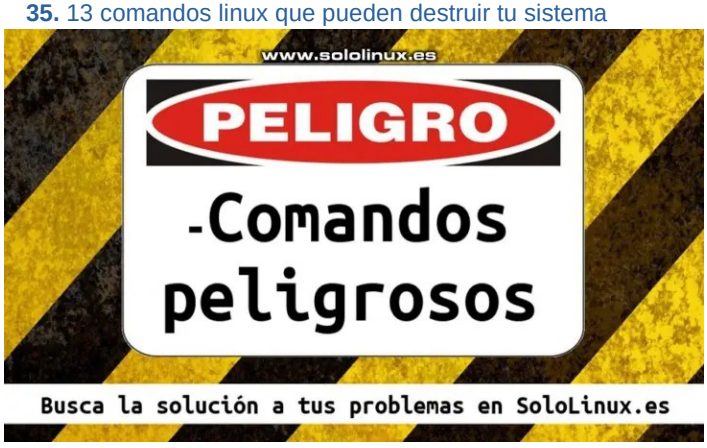
Para mayor información escribe un e-mail a: adrian@sololinux.es

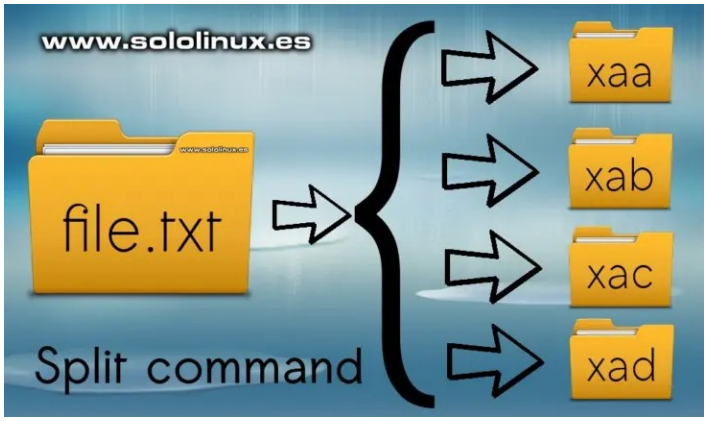
Contacto

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

- adrian@sololinux.es

MANUALES





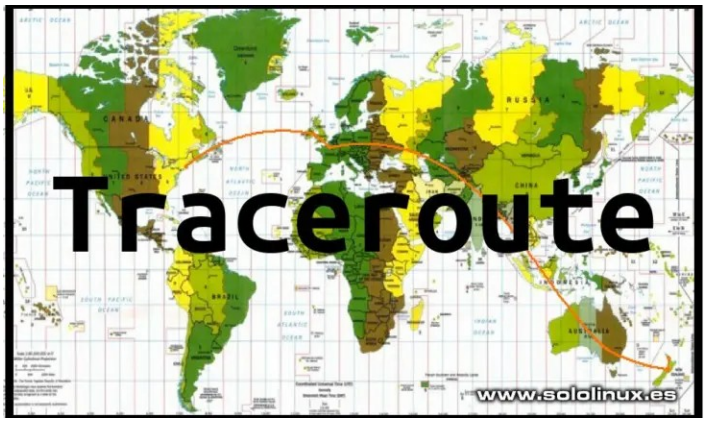
SOFTWARE

HARDWARE

DISTROS LINUX

ENTREVISTAS

REDES



Esta revista es de distribución gratuita, si lo consideras oportuno puedes ponerle precio.
Tu también puedes ayudar, contamos con la posibilidad de
hacer donaciones para la REVISTA, de manera muy simple
a través de PAYPAL

AYUDANOS A SEGUIR CRECIENDO


VANT
SOMOS LINUXEROS
intel
CORE
i5
intel
CORE
i7
intel
iRISxe
GRAPHICS
edge²
Nuestro utrabook más ligero y con mayor autonomía, ahora más potente
con Intel Core de 11ª generación y gráficos Intel Iris Xe
descúbrenos en www.vantpc.es
@vantpc
vant.pc
vantpc_es
t.me/vantpc

INSTITUTO LINUX

[UNIVERSIDAD TECNOLOGICA NACIONAL] FRD [FACULTAD REGIONAL DELTA]

Linux Professional Institute APPROVED TRAINING PARTNER

THE LINUX FOUNDATION AUTHORIZED TRAINING PARTNER

INSCRIPCIÓN ABIERTA 2021

TÉCNICO LINUX 2021

CURSO LINUX SYSTEM ADMINISTRATOR
+
CERTIFICACIÓN UTN-FRD
+
22 CLASES MAGISTRALES ON LINE ¡DE REGALO!
+
WORKSHOPS LPIC 1 (101-102) ¡DE REGALO!

Tutorías de Fabián Ampalio

+54 9 11 6969 9993

www.aprenderlinux.com

Debsecan – Actualizaciones de seguridad en Debian

Estar al día con las nuevas **actualizaciones de seguridad** que se lanzan continuamente, no es tarea difícil. En **Debian** disponemos de una herramienta que simplifica la tarea.

Esta utilidad nos ayuda a evaluar el estado de seguridad actual, sin tener que molestarnos en buscar las nuevas actualizaciones de seguridad de forma manual. También dispone de la función de informarnos sobre las actualizaciones faltantes, aunque es primordial conocer si existe alguna vulnerabilidad conocida en las herramientas ya instaladas.

Debsecan – Actualizaciones de seguridad en Debian

Instalamos Debsecan.

```
sudo apt update
sudo apt install debsecan
```

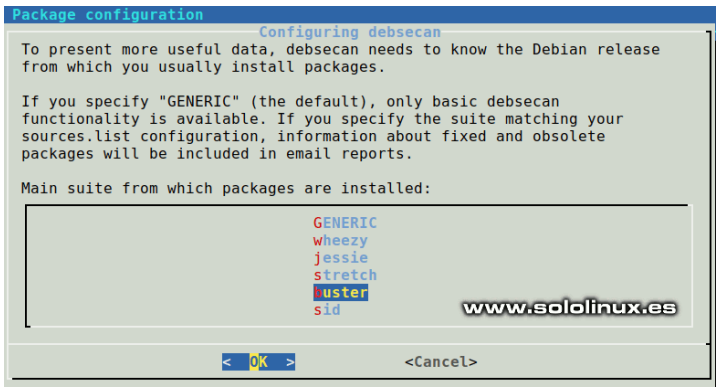
Ejemplo…

```
root@sololinux-demo:~# apt install debsecan
Reading package lists... Done
Building dependency tree... Done
The following additional packages will be installed:
  exim4 exim4-base exim4-config exim4-daemon-light guile-2.2-libs iso-
codes
  libevent-2.1-6 libfribidi0 libgc1c2 libgnutls-dane0 libgnutls30
libgsasl7
  libkyotocabinet16v5 liblzo2-2 libmailutils5 libntlm0 libpython2.7
  libunbound8 mailutils mailutils-common python-apt python-apt-common
Suggested packages:
  exim4-doc-html | exim4-doc-info eximon4 spf-tools-perl swaks
isoquery
  dns-root-data gnutls-bin mailutils-mh mailutils-doc python-apt-dbg
  python-apt-doc
The following NEW packages will be installed:
  debsecan exim4 exim4-base exim4-config exim4-daemon-light guile-2.2-
libs
  iso-codes libevent-2.1-6 libfribidi0 libgc1c2 libgnutls-dane0
libgsasl7
  libkyotocabinet16v5 liblzo2-2 libmailutils5 libntlm0 libpython2.7
  libunbound8 mailutils mailutils-common python-apt python-apt-common
The following packages will be upgraded:
  libgnutls30
1 upgraded, 22 newly installed, 0 to remove and 41 not upgraded.
Need to get 16.2 MB of archives.
After this operation, 83.6 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

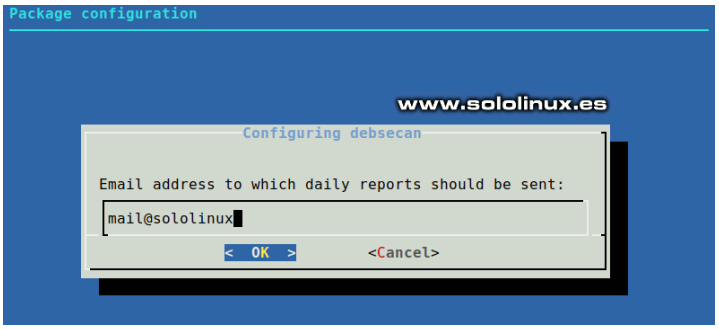
Con el siguiente comando puedes definir la versión instalada. No te olvides que para ejecutar estos comandos debes ser **root**.

```
dpkg-reconfigure debsecan
```

Imagen de ejemplo…



Nos pregunta si queremos que realice un escaneo diariamente y, nos envíe el reporte a un correo electrónico. Estos pasos no son obligatorios, pero si recomendables para un servidor Debian.



Ahora vemos algunos ejemplos de uso en manual. En el primer caso vemos todas las vulnerabilidades de nuestro sistema, incluyendo una pequeña descripción. **No te olvides de insertar tu distribución Debian, en nuestro caso «buster»**.

```
debsecan --suite buster
```

Ejemplo de salida…

```
root@sololinux-demo:~# debsecan --suite buster
CVE-2020-27350 apt (fixed)
CVE-2020-27350 apt-utils (fixed)
CVE-2020-14342 cifs-utils
CVE-2016-2781 coreutils (low urgency)
CVE-2019-14866 cpio (low urgency)
CVE-2020-8177 curl
CVE-2020-8231 curl
CVE-2020-8284 curl
CVE-2020-8285 curl
CVE-2020-8286 curl
CVE-2019-14855 dirmngr (low urgency)
CVE-2018-12886 gcc-8-base
    ........................................
            ........................................
```

Para visualizar más detalles de los paquetes…

```
debsecan --suite buster --format detail
```

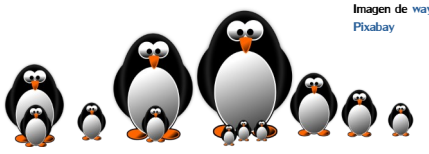


Imagen de way choose en Pixabay

```
root@sololinux-demo:~# debsecan --suite buster --format detail
CVE-2020-27350 (fixed)
  APT had several integer overflows and underflows while parsing .deb
pa ...
  installed: apt 1.8.2.1
             (built from apt 1.8.2.1)
  fixed in unstable: apt 2.1.13 (source package)
  fixed on branch:   apt 1.4.11 (source package)
  fixed on branch:   apt 1.8.2.2 (source package)
  fix is available for the selected suite (buster)
CVE-2020-27350 (fixed)
  APT had several integer overflows and underflows while parsing .deb
pa ...
  installed: apt-utils 1.8.2.1
             (built from apt 1.8.2.1)
  fixed in unstable: apt 2.1.13 (source package)
  fixed on branch:   apt 1.4.11 (source package)
  fixed on branch:   apt 1.8.2.2 (source package)
  fix is available for the selected suite (buster)
CVE-2020-14342
  It was found that cifs-utils' mount.cifs was invoking a shell when
req ...
  installed: cifs-utils 2:6.8-2
             (built from cifs-utils 2:6.8-2)
  fixed in unstable: cifs-utils 2:6.11-1 (source package)
CVE-2016-2781 (low urgency)
  chroot in GNU coreutils, when used with --userspec, allows local
users ...
  installed: coreutils 8.30-3
             (built from coreutils 8.30-3)
```

Con la siguiente opción, solo enumeramos las vulnerabilidades de seguridad faltantes.

```
debsecan --suite buster --only-fixed
```

Se imprime algo similar a…

```
root@sololinux-demo:~# debsecan --suite buster --only-fixed
CVE-2020-27350 apt (fixed)
CVE-2020-27350 apt-utils (fixed)
CVE-2020-25692 ldap-utils (fixed)
CVE-2020-25709 ldap-utils (fixed)
CVE-2020-25710 ldap-utils (fixed)
CVE-2020-27350 libapt-inst2.0 (fixed)
CVE-2020-27350 libapt-pkg5.0 (fixed)
CVE-2020-12049 libdbus-1-3 (fixed)
CVE-2020-15999 libfreetype6 (fixed)
CVE-2020-28196 libgssapi-krb5-2 (fixed)
          ..................................
                ....................................
```

Identificamos los paquetes que se verán afectados.

```
debsecan --suite buster --only-fixed
```

Imagen de ejemplo…

```
root@sololinux-demo:~# debsecan --suite buster --only-fixed --format packages
apt
apt-utils
ldap-utils
libapt-inst2.0
libapt-pkg5.0
libdbus-1-3
libfreetype6
libgssapi-krb5-2
libjson-c3
libk5crypto3
libkrb5-3
libkrb5support0
libldap-2.4-2
libldap-common
libmariadb3
libp11-kit0
libperl5.28
libpython3.7-minimal
libpython3.7-stdlib
libsnmp-base
libsnmp30
libsqlite3-0
libssl1.1
```


Con el siguiente comando, actualizamos nuestro sistema, sin olvidarnos de introducir nuestra versión (en nuestro caso buster).

```
sudo apt install $(debsecan --suite buster --only-fixed --
format packages)
```

```
root@sololinux-demo:~# sudo apt install $(debsecan --suite buster --only-fixed --format packages)
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  apt-doc aptitude | synaptic | wajig dpkg-dev powermgmt-base
  libsasl2-modules-gssapi-mit | libsasl2-modules-gssapi-heimdal krb5-doc
  krb5-user snmp-mibs-downloader perl-doc libterm-readline-gnu-perl
  | libterm-readline-perl-perl make libb-debug-perl liblocale-codes-perl
  python3.7-venv python3.7-doc binutils binfmt-support
Recommended packages:
  libsasl2-modules dbus krb5-locales
The following packages will be upgraded:
  apt apt-utils ldap-utils libapt-inst2.0 libapt-pkg5.0 libdbus-1-3
  libfreetype6 libgssapi-krb5-2 libjson-c3 libk5crypto3 libkrb5-3
  libkrb5support0 libldap-2.4-2 libldap-common libmariadb3 libp11-kit0
  libperl5.28 libpython3.7-minimal libpython3.7-stdlib libsnmp-base libsnmp30
  libsqlite3-0 libssl1.1 libxml2 mariadb-common openssl perl perl-base
  perl-modules-5.28 python3.7 python3.7-minimal snmp
32 upgraded, 0 newly installed, 0 to remove and 9 not upgraded.
Need to get 26.0 MB of archives.
After this operation, 3072 B of additional disk space will be used.
Get:1 http://security.debian.org buster/updates/main amd64 libapt-pkg5.0 amd64 1.8.2.2 [966 kB]
Get:2 http://security.debian.org buster/updates/main amd64 libapt-inst2.0 amd64 1.8.2.2 [204 kB]
Get:3 http://ftp.debian.org/debian buster/main amd64 libperl5.28 amd64 5.28.1-6+deb10u1 [3894 kB]
Get:4 http://security.debian.org buster/updates/main amd64 apt amd64 1.8.2.2 [1419 kB]
Get:5 http://security.debian.org buster/updates/main amd64 apt-utils amd64 1.8.2.2 [421 kB]
Get:6 http://security.debian.org buster/updates/main amd64 libssl1.1 amd64 1.1.1d-0+deb10u4 [1538 kB]
Get:7 http://security.debian.org buster/updates/main amd64 libp11-kit0 amd64 0.23.15-2+deb10u1 [316 kB]
Get:8 http://security.debian.org buster/updates/main amd64 openssl amd64 1.1.1d-0+deb10u4 [843 kB]
Get:9 http://ftp.debian.org/debian buster/main amd64 perl amd64 5.28.1-6+deb10u1 [204 kB]
Get:10 http://ftp.debian.org/debian buster/main amd64 perl-base amd64 5.28.1-6+deb10u1 [1514 kB]
Get:11 http://ftp.debian.org/debian buster/main amd64 perl-modules-5.28 all 5.28.1-6+deb10u1 [2873 kB]
```


Puedes visualizar su completo manual en el **sitio oficial** de Debian o, con este comando en **terminal linux**.

```
man debsecan
```

Si saltaste el paso de enviar reportes por mail, aún estas a tiempo.

```
debsecan --suite buster --format report --mailto root --
update-history
```

Es evidente que también admite tareas cron, revisa el manual oficial de cron con debsecan. Es muy fácil.

Imagen de OpenClipart-Vectors en Pixabay

Esta revista es de **distribución gratuita**, si lo consideras oportuno puedes ponerle precio. Y donarlo al proyecto. Tu también puedes ayudar, contamos con la posibilidad de hacer donaciones para la REVISTA, de manera muy simple a través de **PAYPAL**

AYUDANOS A SEGUIR CRECIENDO

PayPal™ Apoyo con una donación

Nuevo qBittorrent 4.3.2 Compatible con IDN

Hace pocos días fue lanzada la nueva versión del **cliente torrent, qBittorrent 4.3.2**. Además con una agradable sorpresa, pues la nueva versión nos sorprende con una característica sorprendente (entre otras). Ofrece soporte para nombres de dominio internacionalizados (IDN). Esto permite usar nombres de dominio en idiomas locales.

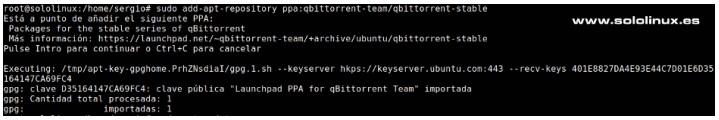
Con una interfaz similar a **µTorren**, **qBittorrent** es uno de los clientes preferidos por los usuarios de linux. No es para menos, es una excelente opción, aunque debo reconocer que si tienes muchos **torrents** puede ralentizar el sistema. Vemos sus nuevas características.

- Ahora se permite agregar una carpeta raíz al contenido del torrent·
- En plataformas con la última versión de **libtorrent** es posible validar **HTTPS**.
- Admite nombres de dominio internacionales (IDN).
- Se corrige el error de clasificación rota en algunas columnas.
- Se corrige el error sobre disponibilidad por valor de archivo.
- Reparado el estado de torrents sin metadatos.
- Ya no existe el error sobre límite superior de la opción «Máximo de conexiones HTTP simultáneas».
- Reparado el error en mover las opciones de «rastreador integrado» a la sección correspondiente, el cambio de extensión, guardar el estado del la pausa de un torrent, y muchos más.
- Ahora puedes usar «shift + delete» para eliminar torrents.
- Se permite adjuntar etiquetas cuando agregas nuevos torrents.
- Ya no existe la longitud máxima de entrada.
- Muchas más correcciones y novedades.

Nuevo qBittorrent 4.3.2 compatible con IDN

Los desarrolladores de qBittorrent mantienen un repositorio actualizado para Ubuntu (y derivados). Lo añadimos a nuestro sistema con este comando.

```
sudo add-apt-repository ppa:qbittorrent-team/qbittorrent-stable
```

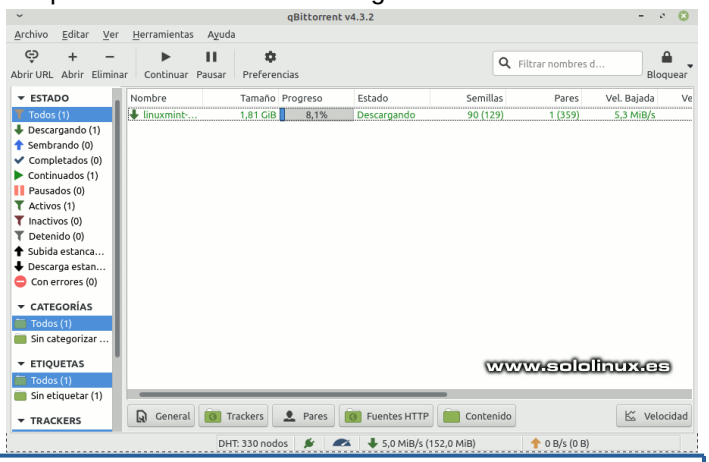


Ahora lo instalamos.

```
sudo apt update
sudo apt install qbittorrent
```

```
root@sololinux:/home/sergio# sudo apt install qbittorrent
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libqt5xml5 libtorrent-rasterbar10
Paquetes sugeridos:
  libtorrent-rasterbar-dbg qbittorrent-dbg
Se instalarán los siguientes paquetes NUEVOS:
  libqt5xml5 libtorrent-rasterbar10 qbittorrent
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 0 no
actualizados.
Se necesita descargar 6.853 kB de archivos.
Se utilizarán 12,7 MB de espacio de disco adicional después de esta
operación.
¿Desea continuar? [S/n] s
Des:1 http://mirror.datacenter.by/ubuntu bionic-updates/main amd64
libqt5xml5 amd64 5.9.5+dfsg-0ubuntu2.5 [99,5 kB]
Des:2
http://ppa.launchpad.net/qbittorrent-team/qbittorrent-stable/ubuntu
bionic/main amd64 libtorrent-rasterbar10 amd64
1.2.11+git20201124.afa406f890-1ppa1~18.04 [943 kB]
Des:3
http://ppa.launchpad.net/qbittorrent-team/qbittorrent-stable/ubuntu
bionic/main amd64 qbittorrent amd64 1:4.3.2.99~202012272006-7195-
abb854a1e~ubuntu18.04.1 [5.811 kB]
Descargados 6.853 kB en 4s (1.661 kB/s)
Seleccionando el paquete libtorrent-rasterbar10 previamente no
seleccionado.
(Leyendo la base de datos ... 344224 ficheros o directorios instalados
actualmente.)
Preparando para desempaquetar .../libtorrent-
rasterbar10_1.2.11+git20201124.afa406f890-1ppa1~18.04_amd64.deb ...
Desempaquetando libtorrent-rasterbar10 (1.2.11+git20201124.afa406f890-
1ppa1~18.04) ...
Seleccionando el paquete libqt5xml5:amd64 previamente no seleccionado.
Preparando para desempaquetar .../libqt5xml5_5.9.5+dfsg-
0ubuntu2.5_amd64.deb ...
Desempaquetando libqt5xml5:amd64 (5.9.5+dfsg-0ubuntu2.5) ...
Seleccionando el paquete qbittorrent previamente no seleccionado.
Preparando para desempaquetar
.../qbittorrent_1%3a4.3.2.99~202012272006-7195-
abb854a1e~ubuntu18.04.1_amd64.deb ...
Desempaquetando qbittorrent (1:4.3.2.99~202012272006-7195-
abb854a1e~ubuntu18.04.1) ...
Configurando libtorrent-rasterbar10 (1.2.11+git20201124.afa406f890-
1ppa1~18.04) ...
Configurando libqt5xml5:amd64 (5.9.5+dfsg-0ubuntu2.5) ...
Configurando qbittorrent (1:4.3.2.99~202012272006-7195-
abb854a1e~ubuntu18.04.1) ...
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...
Procesando disparadores para gnome-menus (3.13.3-11ubuntu1.1) ...
Procesando disparadores para hicolor-icon-theme (0.17-2) ...
Procesando disparadores para mime-support (3.60ubuntu1) ...
Procesando disparadores para desktop-file-utils (0.23+linuxmint8) ...
Procesando disparadores para libc-bin (2.27-3ubuntu1.4) ...
```

Ya puedes comenzar a descargar archivos torrent.



Desinstalar qBittorrent 4.3.2

Si quieres desinstalar el cliente torrent, la tarea es sencilla. Ejecuta los comandos que te indico.

```
sudo apt-get remove --autoremove qbittorrent
```

Para concluir el proceso borramos el repositorio.

```
sudo add-apt-repository --remove ppa:qbittorrent-team/qbittorrent-stable
```

```
root@sololinux:/home/sergio# sudo apt-get remove --autoremove qbittorrent
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los siguientes paquetes se ELIMINARÁN:
  libqt5xml5 libtorrent-rasterbar10 qbittorrent
0 actualizados, 0 nuevos se instalarán, 3 para eliminar y 0 no actualizados.
Se liberarán 12,7 MB después de esta operación.
¿Desea continuar? [S/n]
```


www.sololinux.es

Canales de Telegram: Canal SoloLinux – Canal SoloWordpress

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

Chat de SoloLinux en Telegram

Limitar el tiempo de sesión sudo en linux

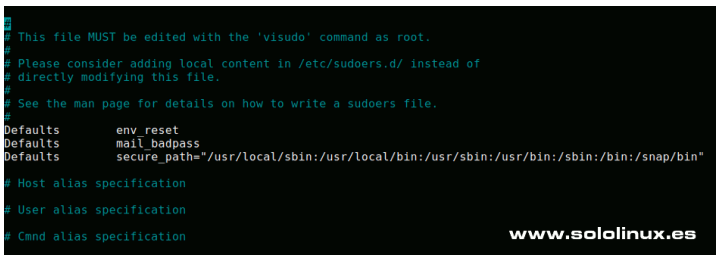
De forma predeterminada, el temporizador que limita el **tiempo de sesión sudo en linux**, está configurado en 5 minutos (algunas distribuciones lo tienen en 15 minutos). Esto quiere decir que cada vez que ejecutas «**sudo comando**», puedes ejecutar de nuevo «**sudo comando**» sin tener que ingresar otra vez la contraseña, siempre que no hayan pasado 5 minutos.

Esta protección es extremadamente útil, para que nadie pueda acceder a tu **sistema** en caso de ausencia. Por otro lado… si eres de los que trabaja continuamente en la **terminal linux**, puede ser un auténtico engorro. En este artículo vemos como modificar el tiempo de sesión sudo o, incluso deshabilitarlo.

Limitar el tiempo de sesión sudo en linux
Desactivar o modificar este valor, es tarea sencilla. Lo que haremos es agregar una orden en el **archivo sudoers**. Recuerda que esta modificación puede suponer un riesgo para la integridad de tu sistema, si operas junto a otras personas y sueles abandonar tu puesto de trabajo, no lo hagas.

Bueno, si estás decidido abre sudoers con **visudo**.

```
sudo visudo
```



En las opciones por defecto (defaults), agregamos esto…

```
Defaults      timestamp_timeout=
```

Si agregas el valor «0», se desactiva el temporizador y siempre solicitara la contraseña.

```
Defaults      timestamp_timeout=0
```

El valor se indica en minutos, por tanto si queremos demorar 20 minutos insertamos algo como esto.

```
Defaults      timestamp_timeout=20
```

```
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults
secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/
bin:/snap/bin"
Defaults        timestamp_timeout=20
# Limitar el tiempo de sesión sudo en linux / sololinux.es
# Host alias specification
# User alias specification
# Cmnd alias specification
```

Algunos dirán que esta manera de **limitar el tiempo de sesión sudo en linux**, se puede minimizar. Si claro, ya lo sé, por ejemplo agregando el valor en otra línea.

```
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset, timestamp_timeout = 20 # <<------ ejemplo
Defaults        mail_badpass
Defaults
secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/
bin:/snap/bin"
# Limitar el tiempo de sesión sudo en linux / sololinux.es
# Host alias specification
```

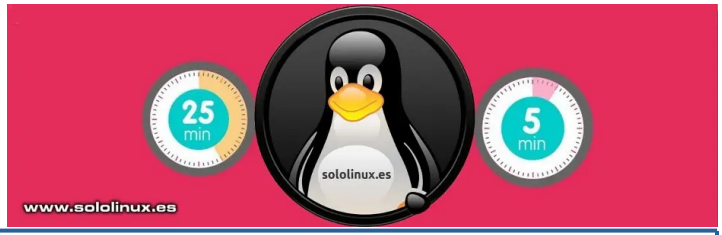
Creo que siempre es bueno tener las cosas bien ordenadas, por eso prefiero la primera opción. Decidas la que decidas, solo te falta guardar el archivo sudoers y cerrar el editor.

Si quieres terminar la sesión sudo antes de tiempo, ejecuta…

```
sudo -k
```

Si no eres partidario de modificar nada en tu sistema, puedes mantener la sesión sudo abierta (sin que pida password), con este comando. No debes cerrar la terminal / consola.

```
sudo -s
```



Cómo actualizar sudo en Linux

La **herramienta sudo**, es la utilidad más extendida en sistemas **Unix / Linux** si queremos ejecutar aplicaciones con privilegios de seguridad elevados, por ejemplo como **root**.

Como hablamos de una herramienta de seguridad importante, su desarrollo es continuo. Incomprensiblemente, las **distribuciones Linux** actuales no aplican las actualizaciones de la herramienta, a no ser que sea por un fallo grave de seguridad (salvo alguna excepción).

Por ejemplo… no es lógico que una distro como **Ubuntu 18.04** esté utilizando un desarrollo del año 2017, al cual… muy de vez en cuando le aplican algún parche, simplemente no es normal. En este artículo, vemos como actualizar sudo a su última versión estable.

Cómo actualizar sudo en linux

En nuestro caso actualizamos sudo en Ubuntu 20.04, pero estas instrucciones son válidas para otro tipo de distros. Lo primero que hacemos es verificar la versión que tenemos instalada de «sudo».

```
sudo -V
```

```
root@sololinux-demo:~$ sudo -V
Sudo versión 1.8.31
versión del complemento de políticas de sudoers 1.8.31
versión de gramática del archivo Sudoers 46
Sudoers I/O plugin version 1.8.31
```

Ahora accedemos a la **página oficial de descargas** y, busca en la tabla tu **distribución linux**. Cómo actualizar sudo en linux.

- **Sitio de descargas de sudo**

Ubuntu 16.04	sudo_1.9.4-3_ubu1604_amd64.deb	af51b9896c80106714b78112a3498529ec3741b9707841e2eed1d62e3aa543be
	sudo-ldap_1.9.4-3_ubu1604_amd64.deb	33065f8b747ee9a74efe2f5f6d0a90ab45b5607b7c6ad8b9a0b73fa36ab0535d
	sudo-logsrvd_1.9.4-3_ubu1604_amd64.deb	e7087804488876be4d47ab84a62edf28c6ea489f03711dadd7f142e7ddf33ce9
	sudo-python_1.9.4-3_ubu1604_amd64.deb	a0129d991632f8fd0def53a0801e21ece25e619eec2da97a30024bbb88956f88
	sudo_1.9.4-3_ubu1604_i386.deb	2fca72e611a6a560f3cf343e13ffbd89e9bc669423db80960de17f57baab920e
	sudo-ldap_1.9.4-3_ubu1604_i386.deb	7ce417a18ce1b38be6b7c90718bfee357e7812b466e228d5fa80e770f2a56650
	sudo-logsrvd_1.9.4-3_ubu1604_i386.deb	e0d90cbb92c4090a86a316dd67bc6a5b71b37ed102de80a32cd908ea1f8af3bd
	sudo-python_1.9.4-3_ubu1604_i386.deb	026b6461534938723ef7f5667cd462cfe1efa8a92e6d3639927a9b2af5c80a50
Ubuntu 18.04	sudo_1.9.4-3_ubu1804_amd64.deb	fbda1f3b751eab5a0dc6412a11ec5a72c982e2473056749b276c73b10b1d3236
	sudo-ldap_1.9.4-3_ubu1804_amd64.deb	2e8ff61bd89e43a5a056fe70f94f49d9d9bb8040ccc3160f59c8a7c8e38c7527
	sudo-logsrvd_1.9.4-3_ubu1804_amd64.deb	dc5030e1ae79d814c66b0c7fb49d6bfbb5028ad3933d3a2e50b642fa095283fd
	sudo-python_1.9.4-3_ubu1804_amd64.deb	03b688bf6bea45e8ec099b1190402f0e18cd0dc9d819c43c2b77e152c15e3813
	sudo_1.9.4-3_ubu1804_i386.deb	33160c833fbc03e0c8c8db7a05ff191121c85a5e303c1c4fd17b3cf55e2347a9
	sudo-ldap_1.9.4-3_ubu1804_i386.deb	e3feb04038c1fb2fb030f218386e5b3210864667724b68d71aa7f9e724738c5c
	sudo-logsrvd_1.9.4-3_ubu1804_i386.deb	9f4986e909f3abce118cfdd454d8f687f50a8e64e2eafa99b4d156d9a300fd65
	sudo-python_1.9.4-3_ubu1804_i386.deb	47a1025e61795756ca8a464412dc15beb2cb56b63fb32cc6b40a18fa82683300
Ubuntu 20.04	sudo_1.9.4-3_ubu2004_amd64.deb	628007db0a2b12e781b4fb392b6fa5e34534e344e1e84e5dff867a8729124a2b
	sudo-ldap_1.9.4-3_ubu2004_amd64.deb	9e061a3a023a69ea1709481f3ea8638c91bdefbff4724b5fd92758550979fdb0
	sudo-logsrvd_1.9.4-3_ubu2004_amd64.deb	ea423d681782c80ecd94a1d441c3519919b921d9fadaa43ab3d4c52c28c5c66d
	sudo-python_1.9.4-3_ubu2004_amd64.deb	4288ebf0980853c6aa3ed833ee8dd49a42171fc4b087831860b511c58010cd5a

Para nuestro Ubuntu 20.04, elegimos **«sudo_1.9.4-3_ubu2004_amd64.deb»**. Descargamos el paquete.

```
wget
https://www.sudo.ws/sudo/dist/packages/1.9.4p2/sudo_1.9.4-3_ubu2004_amd64.deb
```

Ahora lo instalamos. En el nuestro caso (al utilizar Ubuntu), ejecutaremos la herramienta **gdebi**, que instala por defecto todas las dependencias necesarias.

Es posible que no tengas gdebi por defecto en el sistema, así que lo instalamos.

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

Bien, ya lo tenemos. Instalamos la nueva versión de la herramienta sudo.

Cuando pregunte si estás seguro, responde «Y». La actualización es extremadamente rápida.

```
sudo gdebi sudo_1.9.4-3_ubu2004_amd64.deb
```

```
root@sololinux-demo:~# sudo gdebi sudo_1.9.4-3_ubu2004_amd64.deb
Reading package lists... Done
Building dependency tree
Reading state information... Done
Reading state information... Done
Provide limited super-user privileges to specific users
 Sudo is a program designed to allow a sysadmin to give limited root
 privileges to users and log root activity. The basic philosophy is to
give
 as few privileges as possible but still allow people to get their
work done.
Do you want to install the software package? [y/N]:y
/usr/bin/gdebi:113: FutureWarning: Possible nested set at position 1
  c = findall("[[(](\S+)/\S+[])]", msg)[0].lower()
(Reading database ... 24122 files and directories currently
installed.)
Preparing to unpack sudo_1.9.4-3_ubu2004_amd64.deb ...
Unpacking sudo (1.9.4-3) over (1.8.31-1ubuntu1) ...
Setting up sudo (1.9.4-3) ...
Installing new version of config file /etc/pam.d/sudo ...
Installing new version of config file /etc/sudoers ...
Processing triggers for man-db (2.9.1-1) ...
```

Listo, sudo ha sido actualizado. Debes recordar, que cada vez que ejecutas sudo se lee el archivo **sudoers**, por tanto no es necesario reiniciar el sistema.

Verificamos la versión instalada de «sudo».

```
sudo -V
```

Sudo ha sido actualizado correctamente.

```
sergio@sololinux:~$ sudo -V
Sudo versión 1.9.4p2
versión del complemento de políticas de sudoers 1.9.4p2
versión de gramática del archivo Sudoers 48
Sudoers I/O plugin version 1.9.4p2
```

Nota: **Gdebi** es una herramienta para Debian, Ubuntu y derivados; en otros sistemas, por ejemplo los basados en rpm el proceso puede ser diferente. **Instalar paquetes rpm**.

Uso del Comando history en Linux

En Linux, hay una herramienta que tiene la capacidad de mostrar todos los últimos comandos utilizados. Su propio nombre ya lo dice, «**history**». De forma predeterminada, el comando history nos imprime en pantalla los últimos quinientos comandos ingresados en nuestra consola / terminal.

En este artículo aprendemos a usarlo, incluyendo algunas opciones y una **variable** de entorno que mejora considerablemente la información aportada por el **comando history**.

Antes de comenzar y, para los más escépticos con la labor realizada desde **sololinux.es**, debo aclarar… que **history no es un comando linux** propiamente dicho; Realmente es una utilidad incluida en la mayoría de las **shell**, que puede variar de una a otra. Nosotros nos centramos en **bash**.

```
sergio@sololinux:~$ history
    1  history
    2  sudo apt update
    3  history
sergio@sololinux:~$ www.sololinux.es
```

Uso del comando history en linux

La utilidad se usa tal como suena, **history**.

```
history
```

```
root@sololinux-demo:~# history
    1  sudo -V
    2  root@sololinux-demo:~$ sudo -V
    3  Sudo versión 1.8.21p2
    4  versión del complemento de políticas de sudoers 1.8.21p2
    5  versión de gramática del archivo Sudoers 46
    6  Sudoers I/O plugin version 1.8.21p2root@sololinux-demo:~$ sudo
-V
    7  Sudo versión 1.8.21p2
    8  versión del complemento de políticas de sudoers 1.8.21p2
    9  versión de gramática del archivo Sudoers 46
   10  Sudoers I/O plugin version 1.8.21p2root@sololinux-demo:~$ sudo
-V
   11  Sudo versión 1.8.21p2
   12  versión del complemento de políticas de sudoers 1.8.21p2
   13  versión de gramática del archivo Sudoers 46
   14  Sudoers I/O plugin version 1.8.21p2root@sololinux-demo:~$ sudo
-V
   15  Sudo versión 1.8.21p2
   16  versión del complemento de políticas de sudoers 1.8.21p2
   17  versión de gramática del archivo Sudoers 46
   18  Sudoers I/O plugin version 1.8.21p2root@sololinux-demo:~$ sudo
-V
   19  Sudo versión 1.8.21p2
   20  versión del complemento de políticas de sudoers 1.8.21p2
   21  versión de gramática del archivo Sudoers 46
   22  Sudoers I/O plugin version 1.8.21p2root@sololinux-demo:~$ sudo
-V
   23  Sudo versión 1.8.21p2
   24  versión del complemento de políticas de sudoers 1.8.21p2
   25  versión de gramática del archivo Sudoers 46
   26  Sudoers I/O plugin version 1.8.21p2
   27  wget https://www.sudo.ws/sudo/dist/packages/1.9.4p2/sudo_1.9.4-
3_ubu2004_amd64.deb
   28  ls
   29  sudo gdebi sudo_1.9.4-3_ubu2004_amd64.deb
   30  sudo apt install gdebi-core
   31  apt update
   32  sudo apt install gdebi-core
   33  sudo gdebi sudo_1.9.4-3_ubu2004_amd64.deb
   34  sudo -V
   35  apt update
   36  apt list --upgradable
   37  apt full-upgrade
   38  history
root@sololinux-demo:~#
```

Como puedes ver en el anterior ejemplo, vemos el historial de comandos numerado, incluso el ejemplo de un artículo anterior donde vimos como actualizar la herramienta sudo.

Los comandos enumerados son muy útiles, si lo que deseas es ejecutar la misma herramienta. Por ejemplo… hemos listado como 37, **apt full-upgrade**. Para ejecutar de nuevo la orden de actualizar el sistema en su totalidad, es tan simple como insertar el símbolo de terminar exclamación, seguido del número de orden del listado.

```
!37
```

```
root@sololinux-demo:~# !37                                  www.sololinux.es
apt full-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@sololinux-demo:~#
```

History, permite definir el número de últimos **comandos** ejecutados en **nuestro linux**. Por ejemplo «ocho».

```
history 8
```

```
root@sololinux-demo:~# history 8
   34  sudo -V
   35  apt update
   36  apt list --upgradable
   37  apt full-upgrade
   38  history
   39  Sudo versión 1.8.21p2
   40  apt full-upgrade
   41  history 8
```

Otra forma posible es aprovecharnos de **tail**, que por defecto nos lista los 10 últimos comandos ejecutados.

```
history | tail
```

```
root@sololinux-demo:~# history | tail
   34  sudo -V
   35  apt update    www.sololinux.es
   36  apt list --upgradable
   37  apt full-upgrade
   38  history
   39  Sudo versión 1.8.21p2
   40  apt full-upgrade
   41  history 8
   42  history | less
   43  history | tail
root@sololinux-demo:~#
```

Si en vez de tails, utilizamos less, se listaran los últimos comandos línea por línea.

```
history | less
```

Salida de history detallada

Aquí el plato fuerte del artículo. La verdad es que la salida del **comando history** es un tanto insulsa, no aporta ningún detalle importante añadido que nos ayude a detectar interacciones no deseadas, o simplemente a recordar nuestro trabajo anterior.

La solución es… fácil, fácil, las **variables de entorno** ponen fin al problema. Copia y pega lo siguiente.

```
export HISTTIMEFORMAT='%F %T '
```

Ahora nos dice la hora y fecha, que el comando se ejecutó por última vez (con terminal abierto).

```
root@sololinux-demo:~# history
    1  2021-01-06 10:08:58 sudo -V
    2  2021-01-06 10:08:58 root@sololinux-demo:~$ sudo -V
    3  2021-01-06 10:08:58 Sudo versión 1.8.21p2
    4  2021-01-06 10:08:58 versión del complemento de políticas de
sudoers 1.8.21p2
    5  2021-01-06 10:08:58 versión de gramática del archivo Sudoers 46
    6  2021-01-06 10:08:58 Sudoers I/O plugin version
1.8.21p2root@sololinux-demo:~$ sudo -V
    7  2021-01-06 10:08:58 Sudo versión 1.8.21p2
    8  2021-01-06 10:08:58 versión del complemento de políticas de
sudoers 1.8.21p2
    9  2021-01-06 10:08:58 versión de gramática del archivo Sudoers 46
   10  2021-01-06 10:08:58 Sudoers I/O plugin version
1.8.21p2root@sololinux-demo:~$ sudo -V
   11  2021-01-06 10:08:58 Sudo versión 1.8.21p2
   12  2021-01-06 10:08:58 versión del complemento de políticas de
sudoers 1.8.21p2
   13  2021-01-06 10:08:58 versión de gramática del archivo Sudoers 46
   14  2021-01-06 10:08:58 Sudoers I/O plugin version
1.8.21p2root@sololinux-demo:~$ sudo -V
   15  2021-01-06 10:08:58 Sudo versión 1.8.21p2
   16  2021-01-06 10:08:58 versión del complemento de políticas de
sudoers 1.8.21p2
   17  2021-01-06 10:08:58 versión de gramática del archivo Sudoers 46
   18  2021-01-06 10:08:58 Sudoers I/O plugin version
1.8.21p2root@sololinux-demo:~$ sudo -V
   19  2021-01-06 10:08:58 Sudo versión 1.8.21p2
   20  2021-01-06 10:08:58 versión del complemento de políticas de
sudoers 1.8.21p2
   21  2021-01-06 10:08:58 versión de gramática del archivo Sudoers 46
   22  2021-01-06 10:08:58 Sudoers I/O plugin version
1.8.21p2root@sololinux-demo:~$ sudo -V
   23  2021-01-06 10:08:58 Sudo versión 1.8.21p2
   24  2021-01-06 10:08:58 versión del complemento de políticas de
sudoers 1.8.21p2
   25  2021-01-06 10:08:58 versión de gramática del archivo Sudoers 46
   26  2021-01-06 10:08:58 Sudoers I/O plugin version 1.8.21p2
   27  2021-01-06 10:08:58 wget
https://www.sudo.ws/sudo/dist/packages/1.9.4p2/sudo_1.9.4-
3_ubu2004_amd64.deb
   28  2021-01-06 10:08:58 ls
   29  2021-01-06 10:08:58 sudo gdebi sudo_1.9.4-3_ubu2004_amd64.deb
   30  2021-01-06 10:08:58 sudo apt install gdebi-core
   31  2021-01-06 10:08:58 apt update
   32  2021-01-06 10:08:58 sudo apt install gdebi-core
   33  2021-01-06 10:08:58 sudo gdebi sudo_1.9.4-3_ubu2004_amd64.deb
   34  2021-01-06 10:08:58 sudo -V
   35  2021-01-06 10:09:09 apt update
   36  2021-01-06 10:09:22 apt list --upgradable
   37  2021-01-06 10:09:31 apt full-upgrade
   38  2021-01-06 10:10:41 history
   39  2021-01-06 13:53:42 Sudo versión 1.8.21p2
   40  2021-01-06 13:54:07 apt full-upgrade
   41  2021-01-06 14:11:04 history 8
   42  2021-01-06 14:16:49 history | less
   43  2021-01-06 14:17:14 history | tail
   44  2021-01-06 14:24:55 history | less
   45  2021-01-06 14:29:38 export HISTTIMEFORMAT='%F %T '
   46  2021-01-06 14:29:46 history
```

Debes tener en cuenta, que la variable añadida es temporal. Para que sea permanente…

```
sudo echo "export HISTTIMEFORMAT='%F %T '" >>
~/.bash_profile
```

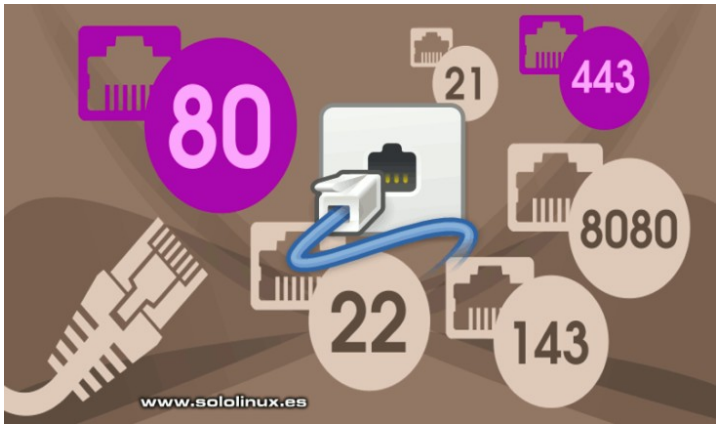
Como último apunte, es posible que tengas miradas indiscretas. En este caso borramos todo el historial.

```
history -c
```

```
root@sololinux-demo:~# sudo echo "export HISTTIMEFORMAT='%F %T '" >> ~/.bash_profile
root@sololinux-demo:~#  history -c
root@sololinux-demo:~# history
    1  2021-01-06 14:50:18 history
root@sololinux-demo:~#
```

www.sololinux.es

Preguntas y respuestas sobre puertos en Linux



Muchos son los artículos publicados en **sololinux** sobre los puertos y su manejo, pero jamás hemos dado una explicación a los usuarios más noveles, sobre que son en realidad los puertos; Por ello, hoy lanzamos el artículo «**Preguntas y respuestas sobre puertos en linux**», para novatos.

A pesar de que la denominamos para novatos, esta publicación es importante para cualquier tipo de usuario, si quieres comprender los puertos, sus detalles y numeración.

Preguntas y respuestas sobre puertos en linux

Respondemos alguna de las posibles dudas que te pueden surgir.

¿Qué es un puerto?

Básicamente, un puerto es un pequeño código que se utiliza como punto de acoplamiento en nuestra máquina, desde el cual podemos comunicarnos remotamente con otra máquina.

¿Qué es un puerto hardware?

El puerto hardware, es el punto de conexión en modo periférico físico a una máquina desde otro dispositivo.

¿Qué es un socket?

Denominamos socket a la combinación de puerto de software y **dirección IP**.

¿Cuántos puertos existen en Linux?

El rango de puertos va desde el 0 al 65535, por tanto tenemos 65536 puertos.

¿Por qué solo tenemos 65535 puertos?

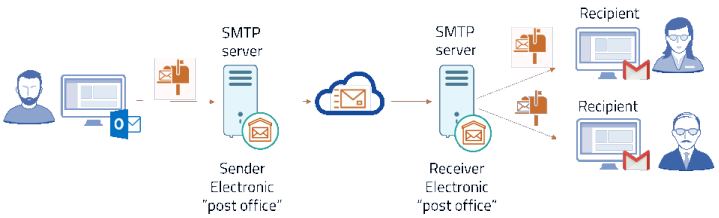
Esto se debe a la limitación TCP/IP, donde cada número de puerto tiene un tamaño de solo 16 bits. Esto equivale a 2^16 (2 elevado a la potencia 16).

¿Qué puertos son los predeterminados?

Los puertos predeterminados y más utilizados, van del 0 al 1023 (2^10 = 1024 puertos). Otras herramientas usan el resto de puertos.

¿Qué es un puerto predeterminado?

El puerto predeterminado es, un puerto designado para un servicio en particular, como **servidor web, servidor de correo, servidor ftp,** etc.



¿Es posible modificar un puerto predeterminado?

La respuesta es clara, si se puede. Tan solo debemos modificar el puerto de escucha, en el archivo de configuración del servicio que te interese.

¿Cuántos números de protocolo existen en TCP / UDP?

No confundas los protocolos con los números de puerto.

- **TCP** : 6
- **UDP** : 17

¿Dónde podemos ver información sobre los puertos?

Para lograr nuestro objetivo, ejecutamos el siguiente comando.

```
cat /etc/services
```

```
sergio@sololinux:~$ cat /etc/services
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-
known
# port number for both TCP and UDP; hence, officially ports have two
entries
# even if the protocol doesn't support UDP operations.
#
# Updated from http://www.iana.org/assignments/port-numbers and other
# sources like http://www.freebsd.org/cgi/cvsweb.cgi/src/etc/services .
# New ports will be added on request if they have been officially
assigned
# by IANA and used in the real-world or are needed by a debian package.
# If you need a huge list of used numbers please install the nmap
package.
tcpmux  1/tcp    # TCP port service multiplexer
echo  7/tcp
echo  7/udp
discard  9/tcp  sink null
discard  9/udp  sink null
systat  11/tcp  users
daytime  13/tcp
daytime  13/udp
netstat  15/tcp
qotd  17/tcp  quote
msp  18/tcp    # message send protocol
msp  18/udp
chargen  19/tcp  ttytst source
chargen  19/udp  ttytst source
ftp-data 20/tcp
ftp  21/tcp
fsp  21/udp  fspd
ssh  22/tcp    # SSH Remote Login Protocol
telnet  23/tcp
smtp  25/tcp  mail
time  37/tcp  timserver
time  37/udp  timserver
```

¿Cómo ver los puertos abiertos en linux?
Para identificar los **puertos abiertos en linux**, tenemos muchas herramientas; Por ejemplo «**NMAP**».

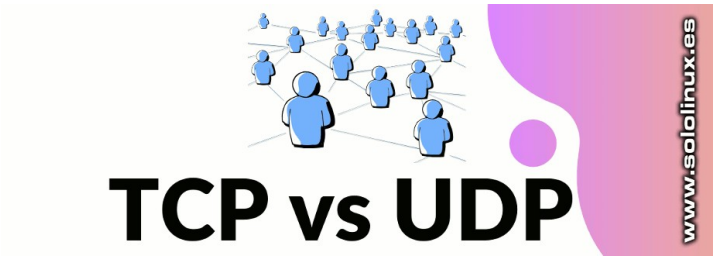
Puertos comunes en linux
Para concluir el artículo, vemos una tabla de los puertos más usados en linux.

Número de puerto	Uso del puerto
20	Transferencia de datos ftp
21	Conexión ftp
22	Puerto SSH
23	Administración remota con Telnet
25	Transferencia de correo SMTP
53	DNS
67	Bootp
68	DHCP
69	TFTP
80	Servicio Apache
88	Kerberos
110	Recepción de mails pop3
123	Servicio NTP (sincronización de horario)
137	NetBios
139	SMB (Samba)
143	IMAP
161	SNMP - Monitor de red
389	LDAP
443	HTTPS - HTTP + SSL para acceso web seguro
514	Syslogd
636	ldaps
873	Rsync
989	Transferencia de datos FTPS
990	FTPS
993	Correo IMAPS
995	POP3
1194	OpenVPN
1912	Radius
2049	NFS (nfsd, rpc.nfsd, rpc, mapa de puertos)
2401	Servidor CVS
3306	Base de datos MySql, MariaDB y más
3690	SVN
6000-6063	Conexión X11 desde remoto

Diferencias entre TCP y UDP

Cuando hablamos de protocolos de Internet en tráfico, los usuarios pueden elegir entre una configuración TCP o UDP. Las características y funciones de **TCP vs UDP** son diferentes, cada protocolo tiene sus ventajas, desventajas y posibles problemas.

Dicho esto, UDP es mucho más rápido, aun así muchos sistemas siguen dependiendo de TCP para descargar paquetes de datos. En este artículo echaremos un vistazo a los dos protocolos, pero recuerda que antes de decirte por uno u otro, debes conocer en profundidad tus necesidades.



Diferencias entre TCP y UDP

Protocolo TCP

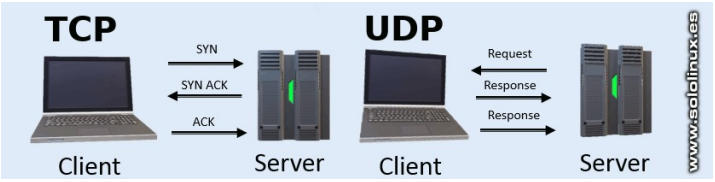
El **Protocolo de control de transmisión (TCP)** está orientado a la conexión, esto quiere decir que una vez que se establece la conexión, los datos se transmiten en dos direcciones. Este protocolo tiene la capacidad de verificar los posibles errores, esta fórmula nos garantiza que los datos se entregan en el orden enviado.

Dicho lo anterior, **TCP** es el protocolo perfecto para transferir información relacionada con páginas web, imágenes fijas y archivos de datos. Como punto negro, también debo indicar que los mecanismos de retroalimentación en TCP, generan una sobrecarga en la red que se traduce en un mayor consumo de ancho de banda.

Protocolo UDP

El **Protocolo de datagramas de usuario (UDP)**, es un protocolo de Internet mucho más simple. No requiere de servicios de recuperación y verificación de errores. Tampoco existe consumo extra al abrir una conexión, mantenerla abierta o terminarla; Los datos se envían de forma continua al destinatario, independientemente de si los recibe o no.

El protocolo UDP no es recomendable para el envío de correos electrónicos, tampoco para visitar **sitios web**, ni descargar archivos. Por otro lado, es la mejor decisión para comunicaciones en tiempo real de cualquier tipo, o realizar labores multitarea remota.



Comparamos TCP y UDP

Para una mejor comprensión, vemos una tabla comparativa.

Tabla comparativa entre TCP y UDP

Característica	UDP	TCP
Estado de la conexión	Protocolo sin conexión necesaria	Requiere una conexión establecida para transmitir datos
Garantía	No garantiza la entrega	Garantiza la entrega al enrutador de destino
Secuencia de datos	No secuencia datos	Si secuencia datos
Método de transferencia	Paquetes UDP con límites definidos; enviado y verificado en su integridad	Los datos son tratados como flujo de bytes; los mensajes se transmiten dependiendo de los límites establecidos
Retransmisión de datos	No retransmite los paquetes perdidos	Si retransmite los paquetes perdidos
Verificación de errores	Muy básica	Potente verificación de errores y reconocimiento de datos
Radiodifusión	Si	No
Velocidad	Rápido	Lento
Uso recomendado	Videoconferencia, streaming, DNS, VoIP, y más	HTTPS, HTTP, SMTP, POP, FTP, y más

Otros análisis

Velocidad TCP vs UDP

UDP admite el flujo de paquetes constante, esa es la gran diferencia sobre TCP. La conexión TCP, está obligada a reconocer un conjunto de paquetes (sea confiable o no), por tanto, se genera una retransmisión en cada reconocimiento cuyo resultado sea la perdida de paquetes.

El protocolo UDP evita estos consumos, por tanto, el efecto-resultado nos aporta una velocidad mucho más eficiente si hablamos de ancho de banda. No olvides que también es menos exigente en verificaciones.

Qué protocolo uso en videoconferencias

Los controles de flujo de TCP, aunque son confiables, no tienen la capacidad de recuperar datos faltantes muy rapido, por ello no es una buena elección en comunicaciones en tiempo real. La integridad de los datos es importante, pero debe estar equilibrada con la velocidad y, así garantizar una comunicación correcta.

Las aplicaciones web y de escritorio (de comunicación), priorizan UDP sobre TCP para el transporte de medios en tiempo real. En este caso, siempre debes usar UDP.

Linux Mint 20.1 Ulyssa – Listo para su descarga

Hace pocos días, se lanzó la esperada distribución Linux Mint 20.1 Ulyssa. Sus ediciones **Cinnamon, MATE y Xfce** (incluyendo la nueva **Cinnamon Edge**), ya están disponibles para su descarga final estable. Linux Mint 20.1 se basa en la versión actualizada de Ubuntu 20.04.1 LTS y, viene con el **kernel Linux 5.4 LTS**.

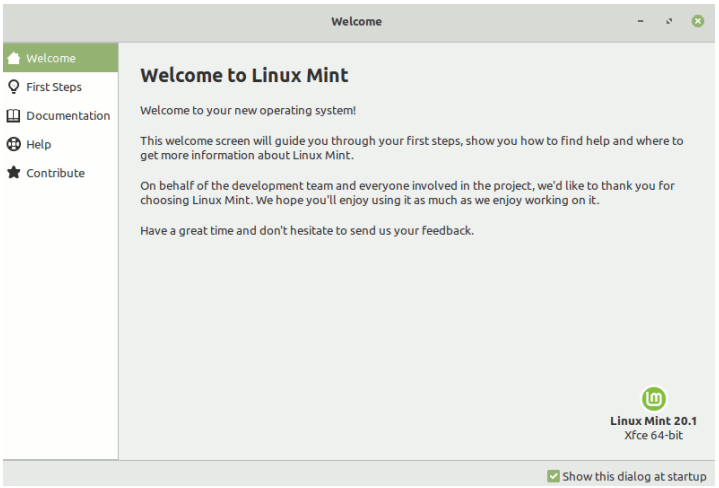
Como es lógico, Linux Mint 20.1 incluye muchas mejoras, paquetes actualizados, así como nuevas características que hacen de Linux Mint una de las mejores distribuciones linux que puedes encontrar. Si Linux Mint 20 es tu distribución actual, no es necesario instalar nada nuevo, tan solo actualizar el sistema como lo haces periódicamente.

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

Linux Mint 20.1 Ulyssa – Listo para su descarga
Uno de los cambios más interesante de esta versión es, la nueva aplicación **Web App Manager** que nos permite convertir un sitio web en una aplicación de escritorio. Todas las aplicaciones web se ejecutan en su propia ventana y, tienen sus iconos específicos que aparecen en el menú de aplicaciones, el panel y el selector Alt-Tab; Además se pueden anclar en el panel. Permite crear tantas aplicaciones web como necesites.

Por fin se incluye **Hypnotix** de manera predeterminada, hablamos de un fabuloso reproductor de **listas IPTV** para listas de reproducción M3U. Admite TV en vivo, películas y programas de TV; Además viene con un proveedor de IPTV gratuito conocido como **Free-IPTV,** que ofrece cientos de canales de TV online gratuitos.

La nueva versión de **Linux Mint**, nos brinda un mejorado soporte para impresoras y escáneres HP (gracias a los últimos **controladores HPLIP**). El reproductor **Celluloid**, ahora trabaja por defecto con video acelerado por hardware. Tampoco se olvidan de PackageKit como administrador de controladores predeterminado y la gran sorpresa… el **navegador web Chromium** es un paquete nativo.



Muchas de las aplicaciones propias de Linux Mint, también se actualizaron. Por ejemplo, el editor de texto Xed ahora permite cerrar corchetes automáticamente al editar el código fuente, el editor de imágenes Pix puede filtrar imágenes por calificación y, el visor de documentos **Xviewer** nos deja configurar los desplazamientos del ratón.

Otro detalle sorprendente es, que podemos configurar el formato del reloj en la pantalla de inicio de sesión. Por otra parte, **Linux Mint 20.1** crea un nuevo diseño unificado del sistema de archivos. Por ejemplo, se fusionan los directorios /bin, /sbin, /lib y /lib64 en /usr, ahora tendremos /usr/bin, /usr/sbin, /usr/lib y /usr/lib64.

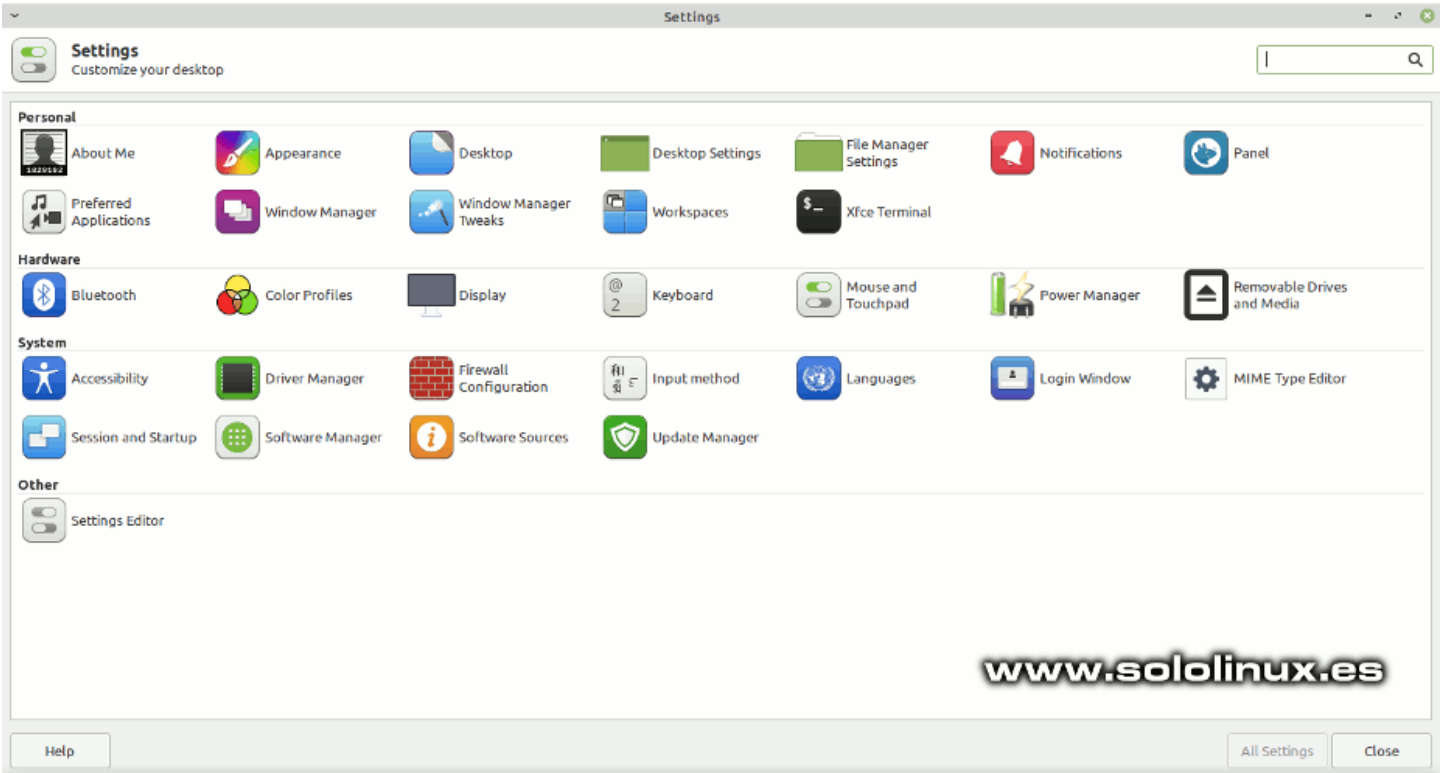
Todo lo dicho anteriormente, es suficiente para decantarnos por la nueva version de Linux Mint; Pero hay más sorpresas, se lanza una nueva versión denominada **Cinnamon Edge** que solucionara tus problemas con el hardware más moderno. Entre otras actualizaciones Edge viene con el Kernel 5.8. Observa la salida del **comando hostname**.

```
mint@mint:~$ hostnamectl
   Static hostname: mint
         Icon name: computer-laptop
           Chassis: laptop
        Machine ID: 986c72adc778411ab12715e3ed547f57
           Boot ID: b0d8700b851a4bc29836750e1939bb91
  Operating System: Linux Mint 20.1
            Kernel: Linux 5.8.0-33-generic  <<------ Kernel acualizado
      Architecture: x86-64
mint@mint:~$
```

Se actualizan los entornos de escritorio como… Cinnamon 4.8, que trae nuevas características y mejoras, por ejemplo la opción «Agregar a favoritos» en todas las aplicaciones de Linux Mint y GTK3. MATE también se actualiza a la versión 1.24. Lamentablemente, la edición Xfce trae la 4.14, parece ser que la 4.16 no ha llegado a tiempo.

Puedes **descargar Linux Mint 20.1 Ulyssa**, desde los enlaces torrent oficiales que te propongo.

- **Edición Cinnamon**
- **Edición Cinnamon Edge**
- **Edición Mate**
- **Edición XFCE**



Canales de Telegram: **Canal SoloLinux** – **Canal SoloWordpress**

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

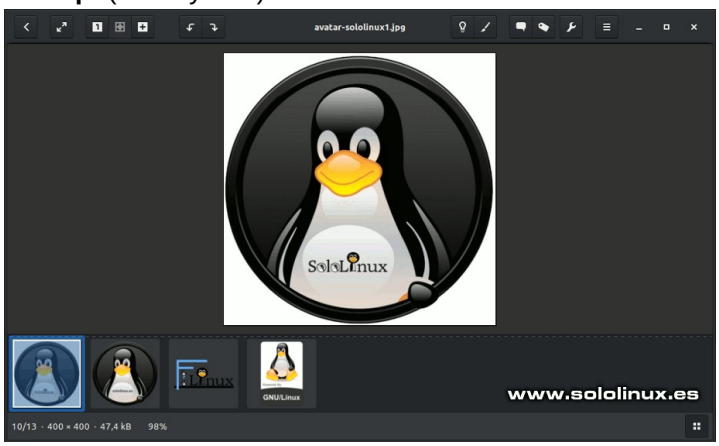
Chat de SoloLinux en Telegram

Instalar gThumb 3.11.2 en Ubuntu y derivados

El poderoso administrador de imágenes y fotografías **gThumb**, ha sido actualizado hace apenas unas horas. La nueva versión 3.11.2 viene con mejoras interesantes, que te ayudaran en maximizar tu productividad. Vemos las principales mejoras y novedades, de **gThumb 3.11.2**.

- Mantener el mismo píxel debajo del puntero al hacer zoom.
- Aumento del zoom proporcional en el visor de imágenes.
- Ahora lee correctamente los perfiles de color de archivos png correctamente.
- El visor de medios agrega la búsqueda precisa al hacer clic en la barra de progreso.
- El visor de medios indica el tiempo marcado al pasar el cursor sobre la barra de progreso.
- Se agrega soporte para los botones de ratón, hacia atrás y hacia delante en el navegador.
- Se agrega la opción mostrar y ocultar la barra de estado.
- Se agrega un botón para retornar a la última plantilla utilizada, cuando modificamos un nombre.
- Muchas correcciones de errores y, actualizaciones de los lenguajes.

En este artículo, vemos como instalar la herramienta en **Ubuntu, Linux Mint** y todos sus derivados, agregando un **apt** (es muy fácil).



Instalar gThumb 3.11.2 en Ubuntu y derivados

Agregamos el PPA adicional, con el siguiente comando.

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

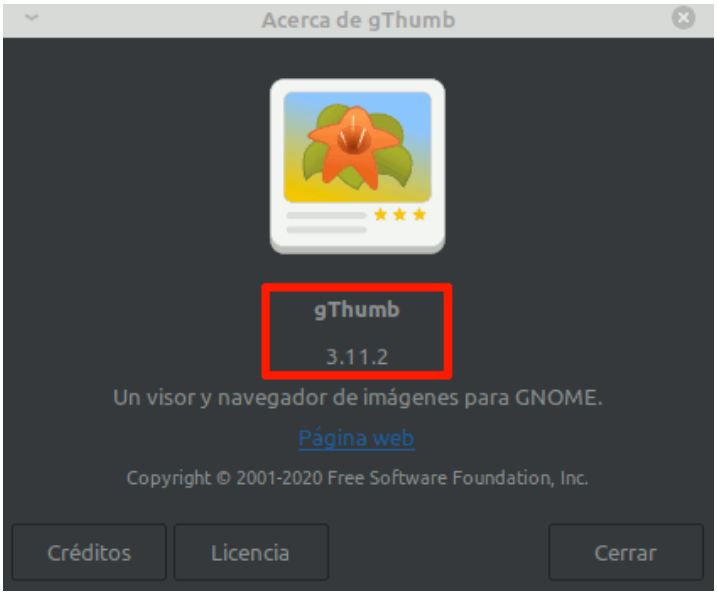
Ahora, solo falta actualizar e instalar gThumb 3.11.2 (se agregan librerías adicionales).

```
sudo apt update
sudo apt install gthumb
```

Ejemplo…

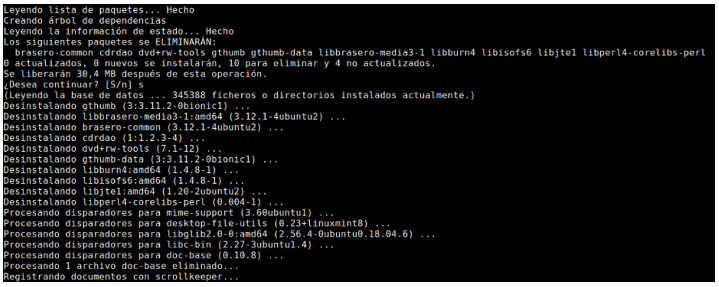
```
sergio@sololinux:# sudo apt install gthumb
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  brasero-common cdrdao dvd+rw-tools gthumb-data libbrasero-media3-1
libburn4
  libisofs6 libjte1 libperl4-corelibs-perl
Paquetes sugeridos:
  cdrskin gstreamer1.0-fluendo-mp3
Se instalarán los siguientes paquetes NUEVOS:
  brasero-common cdrdao dvd+rw-tools gthumb gthumb-data libbrasero-
media3-1
  libburn4 libisofs6 libjte1 libperl4-corelibs-perl
0 actualizados, 10 nuevos se instalarán, 0 para eliminar y 4 no
actualizados.
Se necesita descargar 8.319 kB de archivos.
Se utilizarán 30,4 MB de espacio de disco adicional después de esta
operación.
¿Desea continuar? [S/n] s
```

Desde tu menú de aplicaciones, abre la herramienta y verifica que tienes la última versión instalada.



Si por algún caso decides eliminar **gThumb 3.11.2**, la tarea es sencilla.

```
sudo apt remove --autoremove gthumb gthumb-data
```



Instalar Luminance HDR 2.6.1.1 en Linux

Luminance HDR, es un software de edición de **imágenes HDR** gratuito y de código abierto específico para **Linux**. Permite manipular imágenes de alto rango dinámico (HDR), y admite los siguientes formatos.

- **OpenEXR**
- **Radiance RGBE**
- **Tiff de 16 bits, 32 bits (flotante)**
- **LogLuv**
- **Raw**
- **PFS formato nativo (pfs)**
- **JPEG, PNG, PPM, PBM, TIFF, FITS y LDR**

Está basado en **Qt5** y se publica bajo la licencia GPL-2.0. Con **Luminance HDR**, podemos crear un archivo HDR a partir de un conjunto de imágenes de una misma escena, tomadas con diferentes ajustes de exposición. Destacamos que acepta guardar, cargar, rotar y redimensionar en archivos **Tonemap HDR**.

Instalar Luminance HDR 2.6.1.1 en Linux

Esta herramienta viene por defecto en la mayoría de distribuciones linux, pero como es habitual en Ubuntu y sus derivados nos ofrece versiones obsoletas.

Si queremos instalar la última versión en Ubuntu 20.04 o Ubuntu 20.10 (incluyendo todos sus derivados), debemos agregar el **siguiente ppa**.

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

Actualizamos e instalamos.

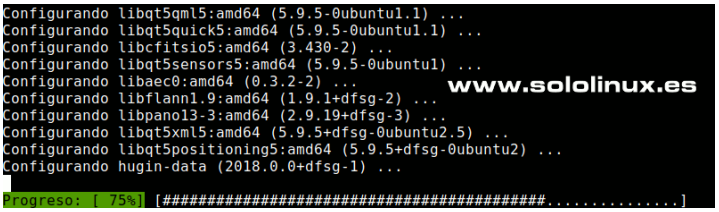
```
sudo apt update
sudo apt install luminance-hdr
```

Su usas Ubuntu 18.04, Linux Mint 19, o cualquiera de sus derivados, mediante ppa solo podrás instalar la versión 2.6.0 (mucho mejor que la 2.5.4 que viene por defecto). El ppa descrito para la versión 20.04, no es válido para Ubuntu 18.04, portando insertamos el repositorio que le corresponde.

```
sudo add-apt-repository ppa:dhor/myway
```

Actualizamos e instalamos.

```
sudo apt update
sudo apt install luminance-hdr
```



Si usas **Open Suse**, visita las descargas insertadas en la web oficialde Open Suse.

Si te decantas por Fedora, ejecuta el siguiente comando.

```
sudo dnf install luminance-hdr
```

Tal vez eres un fiel seguidor de Arch Linux o Manjaro.

```
sudo pacman -S luminancehdr
```

Para los que prefieren utilizar Flatpak, también existe la opción.

```
flatpak remote-add --if-not-exists flathub
https://flathub.org/repo/flathub.flatpakrepo
flatpak install flathub
net.sourceforge.qtpfsgui.LuminanceHDR
```

Una vez instalada la herramienta, puedes lanzarla desde el menú de aplicaciones de tu entorno de escritorio favorito. Si eres un profesional de la fotografía, tienes delante la herramienta perfecta.

Nota final: Dependiendo de tu **distribución linux**, es posible que aún no tengas disponible la versión 2.6.1.1 y se instale **Luminance 2.6.1**.

Deshabilitar IPv6 en Ubuntu 20.04 y otras distribuciones

En la mayoría de distribuciones, el protocolo **IPv6** viene activado por defecto directamente desde el **Kernel Linux 2.6**. Es cierto que para según que aplicaciones es necesario, pero no siempre es así; Otras muchas veces requieres deshabilitar IPv6, por ejemplo… por seguridad en momentos puntuales.

Habilitar o deshabilitar IPv6 en nuestro **sistema linux**, es tarea sencilla. En el artículo de hoy, vemos como deshabilitar o **habilitar IPv6** de forma temporal o permanente, esa decisión depende de tus necesidades.

Deshabilitar IPv6 en Ubuntu 20.04 y otros Linux

Verificar IPv6

Puedes verificar que IPv6 está habilitado, con el siguiente comando.

```
ip -6 addr
```

```
sergio@sololinux:~$ ip -6 addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 state UNKNOWN qlen
1000
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
3: wlo1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 state UP
qlen 1000
    inet6 fe80::e8f4:fa6f:11f1:74f8/64 scope link
noprefixroute
       valid_lft forever preferred_lft forever
sergio@sololinux:~$
```

Deshabilitar IPv6 temporalmente con sysctl

Para lograr nuestro objetivo, ejecuta los siguientes comandos.

```
sudo sysctl -w net.ipv6.conf.all.disable_ipv6=1
sudo sysctl -w net.ipv6.conf.default.disable_ipv6=1
sudo sysctl -w net.ipv6.conf.lo.disable_ipv6=1
```

Verificamos si se deshabilitó el protocolo.

```
ip -6 addr
```

En el siguiente ejemplo vemos que no hay respuesta, por tanto la operación ha sido un éxito.

```
sergio@sololinux:~$ ip -6 addr
sergio@sololinux:~$
```

Recuerda que la desactivación temporal desaparece al reiniciar el sistema.

Deshabilitar IPv6 permanentemente con sysctl

Para deshabilitar permanentemente el protocolo, editamos el archivo «**/etc/sysctl.conf**».

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

Añade las siguientes líneas.

```
net.ipv6.conf.all.disable_ipv6=1
net.ipv6.conf.default.disable_ipv6=1
net.ipv6.conf.lo.disable_ipv6 = 1
```

Guarda el archivo y cierra el editor. Vemos una imagen de ejemplo.

```
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#

#kernel.domainname = example.com

# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3

net.ipv6.conf.all.disable_ipv6=1
net.ipv6.conf.default.disable_ipv6=1
net.ipv6.conf.lo.disable_ipv6 = 1

###################################################################3
# Functions previously found in netbase
#

# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1
```


Solo falta aplicar los cambios.

```
sudo sysctl -p
```

```
root@sololinux-demo:~# sysctl -p
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
root@sololinux-demo:~# ip -6 addr
root@sololinux-demo:~#
```

Para habilitar de nuevo IPv6, tan solo debes borrar las líneas añadidas y aplicar otra vez los cambios.

```
sudo sysctl -p
```

Deshabilitar IPv6 desde el módulo Kernel

También tienes otra opción, es posible evitar la carga del **módulo IPv6** del Kernel en el **Grub**. Editamos su archivo de configuración. **AVISO:** Dependiendo de tu **distribución linux**, es posible que la ruta del archivo pueda ser diferente (nosotros realizamos este artículo con Ubuntu 18.04 LTS).

```
sudo nano /etc/default/grub
```

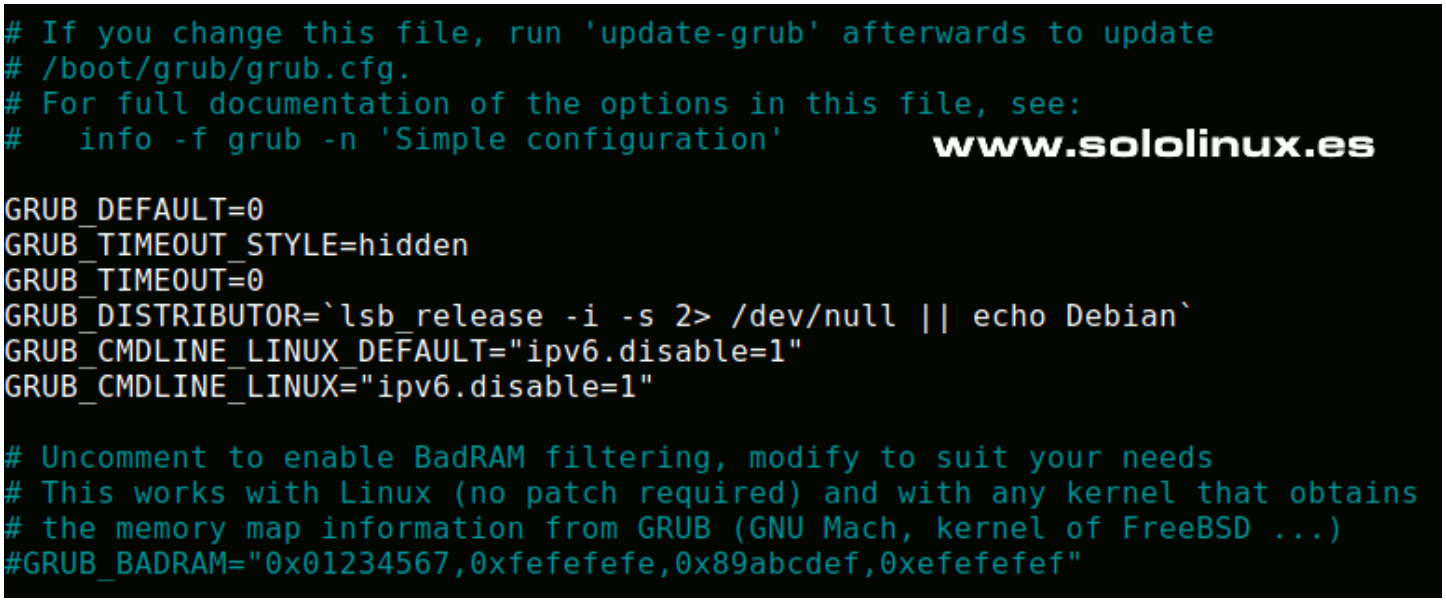
En las siguientes líneas…

```
GRUB_CMDLINE_LINUX_DEFAULT=""
GRUB_CMDLINE_LINUX=""
```

Añadimos lo siguiente.

```
ipv6.disable=1
ipv6.disable=1
```

Nuestro archivo debe ser como el ejemplo de la imagen.



Guarda el archivo y cierra el editor. Es necesario reiniciar el sistema.

```
sudo reboot
```

Canales de Telegram: **Canal SoloLinux** – **Canal SoloWordpress**

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

Chat de SoloLinux en Telegram

Instalar el driver wifi Realtek desde ppa en Ubuntu 20.04

Tal como hablamos en un **artículo anterior**, persisten los problemas con las **tarjetas Wifi Realtek** en nuestros sistemas linux. Lamentable, realmente es una situación lamentable.

Más pronto que tarde, seguro que algún fabricante de adaptadores wifi romperá el monopolio que **Realtek** mantiene con los grandes ensambladores de máquinas, como por ejemplo HP y, dejaremos de tener problemas con ellos.

Por suerte para los usuarios de Ubuntu, Linux Mint y derivados, el equipo de "**Linux Mint Türkiye**" mantiene un PPA actualizado, que contiene los controladores inalámbricos más recientes de los adaptadores Realtek rtlwifi. Vemos algunos de los adaptadores más comunes, que son compatibles con este ppa.

- **rtl8723bu**
- **rtl8822bu**
- **rtl8188eu**
- **rtl8188fu**
- **rtl8192cu**
- **rtl8192du**
- **rtl8192ee**
- **rtl8192eu**
- **rtl8192fu**
- **rtl8723au**
- **rtl8723bu**
- **rtl8723de**
- **rtl8723ds**
- **rtl8723du**
- **rtl8812au**
- **rtl8814au**
- **rtl8821ce**
- **rtl8821cu**
- **rtl8822bu**

Instalar el driver wifi Realtek desde ppa en Ubuntu 20.04

Agregamos el repositorio del **driver wifi Realtek**.

```
sudo add-apt-repository ppa:linuxmint-tr/wireless-ppa
```

```
sergio@sololinux:~$ sudo add-apt-repository ppa:linuxmint-tr/wireless-
ppa
Está a punto de añadir el siguiente PPA:
 ----------------------------------------
[TR] Kablosuz ağ sürücüleri için PPA deposu
[EN] PPA for wifi drivers
[TR] Eğer UEFI kipinde kurulum yaptıysanız, sürücü kurulumundan sonra
DKMS ile yüklenen sürücü modülleri için secureboot özelliğini
pasifleştirmelisiniz.
[EN] If you installed your system in UEFI mode, you must disable
secureboot for modules installed by dkms after installation.
[TR] Aşağıdaki komutla kurulum türünü tespit edebilirsiniz.
[EN] You can check installation mode with following command.
[ -d /sys/firmware/efi ] && echo "EFI" || echo "BIOS"
[EN] You can check secureboot status with following command.
[TR] Aşağıdaki komutla secureboot durumunu kontrol edebilirsiniz.
mokutil --sb-state
 ----------------------------------------
 Más información:
https://launchpad.net/~linuxmint-tr/+archive/ubuntu/wireless-ppa
Pulse Intro para continuar o Ctrl+C para cancelar
Executing: /tmp/apt-key-gpghome.8XimWIWwYk/gpg.1.sh --keyserver
hkps://keyserver.ubuntu.com:443 --recv-keys
59B93F2996D524757BFDF5E3C272D028F84AB7F9
gpg: clave C272D028F84AB7F9: clave pública "Launchpad PPA for Linux
Mint Türkiye" importada
gpg: Cantidad total procesada: 1
gpg:               importadas: 1
```

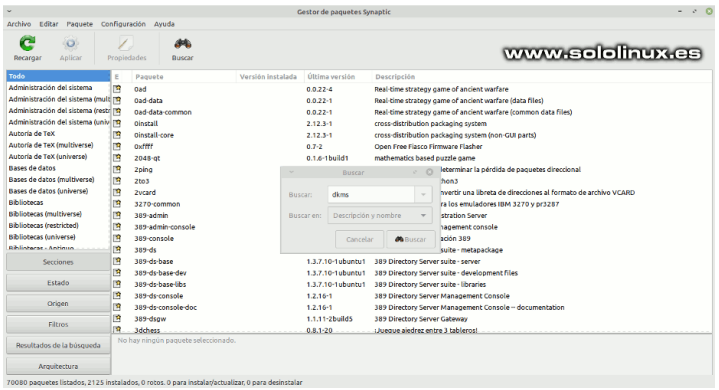
Actualizamos.

```
sudo apt update
```

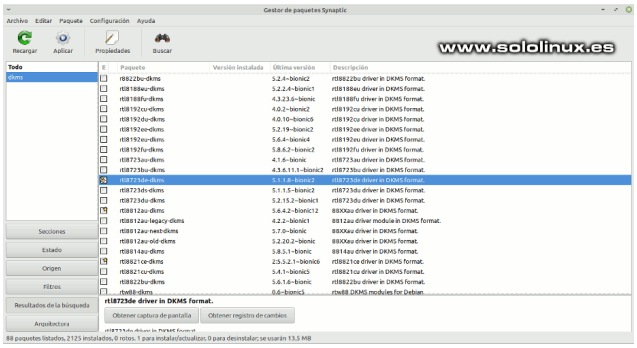
Ahora abrimos el gestor de **paquetes Synaptic**, si no lo tienes… lo instalas con este comando.

```
sudo apt install synaptic
```

Abrimos el gestor de paquetes desde nuestro menú de aplicaciones. En el buscador integrado de la herramienta insertamos la palabra «**dkms**» y, hacemos click en buscar.



Nos aparecen todos los modelos compatibles con el driver instalado, solo tienes que marcar para instalar el que necesitas y, pulsar en el botón «**aplicar**». La instalación comienza inmediatamente.



También lo puedes instalar mediante consola / terminal.

```
sudo apt install rtl8723de-dkms
```

Solo nos falta reiniciar el sistema para que cargue el nuevo módulo del kernel.

```
sudo reboot
```

Aviso importante

- Si tu sistema usa el **modo UEFI**, deberías deshabilitar el arranque seguro para los módulos instalados por dkms después de instalar el driver.
- Si al insertar «**dkms**» no localizas tu dispositivo en **Synaptic**, prueba con «r**tl**».

Canales de Telegram: Canal SoloLinux – Canal SoloWordpress

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

Chat de SoloLinux en Telegram

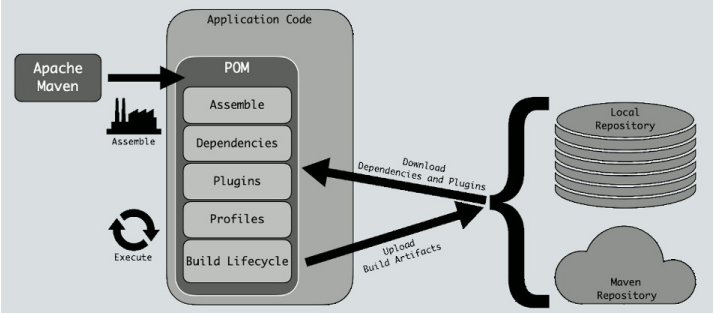
Instalar Apache Maven en Ubuntu 20.04

Apache Maven es una potente herramienta de gestión de proyectos, con un uso muy concreto, la creación, dependencias y documentación de proyectos. Basado en **POM** (modelo de objetos de proyecto), nos ayuda en la gestión del proceso de creación de un proyecto, incluyendo el almacenamiento de documentación, informes y más.

Maven viene con comandos integrados, que resultan extremadamente útiles para trabajar con los paquetes y el diseño del desarrollo. Está aplicación se creó para construir y administrar proyectos basados en Java, pues tiene la capacidad de documentar y reportar toda la información relacionada con el proyecto.

Antes de **instalar Apache Maven** en Ubuntu, vemos sus principales características.

- Sistema de gestión dependencias.
- Mecanismo distribuido de distribución de librerías, desde el repositorio local de Maven hacia los repositorios que están publicados en Internet o en la red corporativa.
- Mecanismos para ser extensible, con plugins customizables.
- Es multi-plataforma, puede funcionar tanto en entornos Linux como Windows.
- Es **opensource**.
- Fomenta la reutilización de código y librerías.
- Es compatible con la mayoría de **IDEs**.



Instalar Apache Maven en Ubuntu 20.04
Antes de comenzar, nos aseguramos de que todos los paquetes del sistema operativo Ubuntu instalados en el servidor, estén actualizados.

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

Continuamos instalando java.

```
sudo apt-get install default-jdk -y
```

Verificamos la versión instalada.

```
java -version
```

```
root@sololinux-demo:~# java -version
openjdk version "11.0.9.1" 2020-11-04
OpenJDK Runtime Environment (build 11.0.9.1+1-Ubuntu-
0ubuntu1.20.04)
OpenJDK 64-Bit Server VM (build 11.0.9.1+1-Ubuntu-
0ubuntu1.20.04, mixed mode, sharing)
```

Una vez tengamos java en nuestro sistema, necesitamos descargar e instalar Apache Maven en nuestro servidor Ubuntu 20.04.

```
cd /opt
wget
https://downloads.apache.org/maven/maven-3/3.6.3/binaries/ap
ache-maven-3.6.3-bin.tar.gz
```

```
root@sololinux-demo:~# cd /opt
root@sololinux-demo:/opt# wget https://downloads.apache.org/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.tar.gz
--2021-01-16 18:10:25--  https://downloads.apache.org/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.tar.gz
Resolving downloads.apache.org (downloads.apache.org)... 2a01:4f8:10a:201a::2, 88.99.95.219
Connecting to downloads.apache.org (downloads.apache.org)|2a01:4f8:10a:201a::2|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9506321 (9.1M) [application/x-gzip]
Saving to: 'apache-maven-3.6.3-bin.tar.gz'

apache-maven-3.6.3- 100%[===================>]   9.07M  7.28MB/s    in 1.2s

2021-01-16 18:10:27 (7.28 MB/s) - 'apache-maven-3.6.3-bin.tar.gz' saved [9506321/9506321]
```


Extraemos Apache Maven.

```
tar xzf apache-maven-3.6.3-bin.tar.gz
```

Es necesario renombrar el directorio extraído.

```
mv apache-maven-3.6.3 apachemaven
```

Ahora configuramos la variable de entorno necesaria, para definir la ruta de Java y Apache Maven. Creamos un nuevo archivo llamado «**apachemaven.sh**», en el directorio **/etc/profile.d/**.

```
nano /etc/profile.d/apachemaven.sh
```

Copia y pega lo siguiente en el archivo que estamos creando.

```
export JAVA_HOME=/usr/lib/jvm/default-java
export M2_HOME=/opt/apachemaven
export MAVEN_HOME=/opt/apachemaven
export PATH=${M2_HOME}/bin:${PATH}
```

Guarda el archivo y cierra el editor. Concedemos los permisos necesarios.

```
chmod +x /etc/profile.d/apachemaven.sh
```

Habilitamos la variable de entorno.

```
source /etc/profile.d/apachemaven.sh
```

```
root@sololinux-demo:~# nano /etc/profile.d/apachemaven.sh
root@sololinux-demo:~# chmod +x /etc/profile.d/apachemaven.sh
root@sololinux-demo:~# source /etc/profile.d/apachemaven.sh
```

Ya tenemos Apache Maven instalado, puedes verificar la versión instalada con el siguiente comando.

```
mvn -version
```

```
root@sololinux-demo:~# mvn -version
Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)
Maven home: /opt/apachemaven
Java version: 11.0.9.1, vendor: Ubuntu, runtime: /usr/lib/jvm/java-11-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "5.4.0", arch: "amd64", family: "unix"
```

Instalar Apache Maven en Ubuntu desde apt

Es recomendable instalar Apache Maven como explicamos anteriormente, es la mejor forma de estar actualizado, pues instalas la versión que tú quieres (descarga **oficial** de Apache Maven). Aun siendo así, también es posible instalar la herramienta desde **apt** (repositorios oficiales), verás que fácil.

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

Ahora… ejecutamos un comando que instala todo lo necesario, incluyendo la configuración de la variable de entorno.

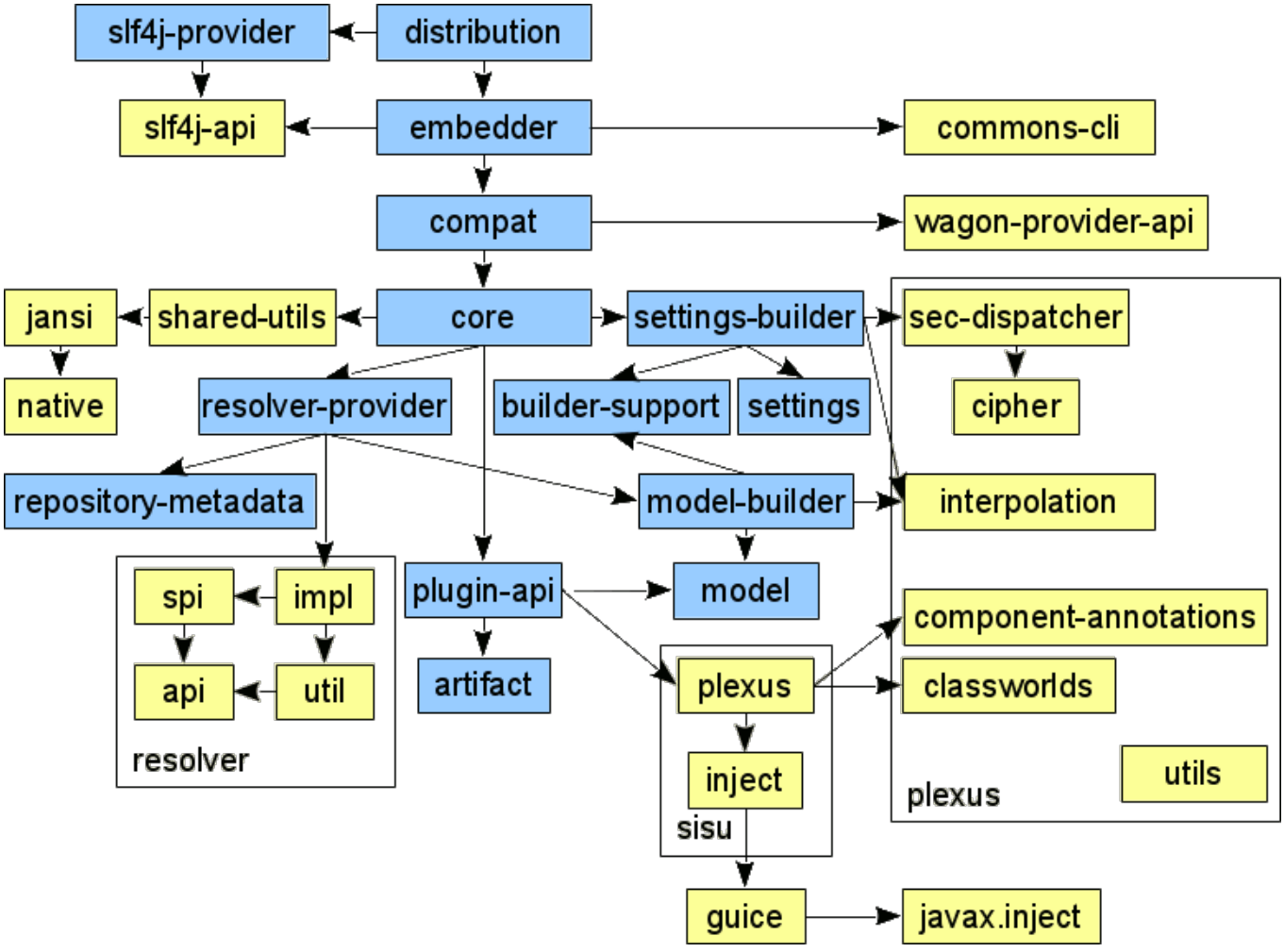
```
apt-get install maven -y
```

La instalación es rápida. Solo nos falta verificar que Apache Maven está instalado en nuestro sistema.

```
mvn -version
```

```
root@sololinux-demo:~# mvn -version
Apache Maven 3.6.3
Maven home: /usr/share/maven
Java version: 11.0.9.1, vendor: Ubuntu, runtime: /usr/lib/jvm/java-11-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "5.4.0", arch: "amd64", family: "unix"
```

Listo, ya tenemos la herramienta lista para producción.



Uso del Comando sar – Monitorizar los reCursos del sistema

La principal tarea de un **administrador de sistemas**, es asegurarse que los servidores que maneja sigan funcionando correctamente, pase lo que pase. Para lograr su objetivo, una importante ayuda… es monitorear continuamente el uso de recursos de las máquinas, como el uso de la memoria, de la CPU, etc.

SAR genera un informe, esta se usa para monitorear los recursos del sistema Linux. Informes relacionados con el rendimiento de un sistema, CPU, memoria, disco, etc. Todo es posible con el **comando sar**.

Uso del comando sar

Esta herramienta es bastante simple, tan solo debes aprender a usar sus tiempos y opciones. Observa la sintaxis.

```
sar option [intervalo-en-segundos] [numero-de-registros]
```

Un buen ejemplo (simple) es, la generación de un registro cada 2 segundos del contenido de las 5 últimas mediciones.

```
sar 2 5
```

```
root@sololinux-demo:~# sar 2 5
Linux 5.4.0 (sololinux-demo)    01/17/21        _x86_64_        (1 CPU)
                                                        www.sololinux.es
12:38:46        CPU     %user     %nice   %system   %iowait    %steal     %idle
12:38:48        all      0.00      0.00      0.00      0.00      0.00    100.00
12:38:50        all      0.00      0.00      0.00      0.00      0.00    100.00
12:38:52        all      0.00      0.00      0.00      0.00      0.00    100.00
12:38:54        all      0.00      0.00      0.00      0.00      0.00    100.00
12:38:56        all      0.00      0.00      0.00      0.00      0.00    100.00
Average:        all      0.00      0.00      0.00      0.00      0.00    100.00
```

Ejemplos de uso de la herramienta sar

Seguimos con el patrón anteriormente descrito, 5 informes cada dos segundos. En nuestro primer ejemplo analizamos la cpu de la maquina.

```
sar -u 2 5
```

Uso del comando sar – Monitorizar los recursos del sistema

Instalar sar en linux

La **herramienta sar,** pertenece al grupo de aplicaciones **sysstat**. No suele venir instalado en todas las **distribuciones linux**, así que debes instalarlo.

Instalar sar en Centos 7

```
sudo yum install sysstat -y
sudo systemctl start sysstat
sudo systemctl enable sysstat
```

Instalar sar en Fedora

```
sudo dnf install sysstat -y
sudo systemctl start sysstat
sudo systemctl enable sysstat
```

Instalar sar en Debian, Ubuntu y derivados

```
sudo apt install sysstat
sudo service sysstat restart
```

Instalar sar en Arch Linux, Manjaro y derivados

```
sudo pacman -S sysstat
```

Vemos algo similar a…

```
root@sololinux:~# sar -u 2 5
Linux 5.4.0-62-generic (sololinux)  17/01/21  _x86_64_ (2 CPU)
14:53:17        CPU     %user     %nice   %system   %iowait    %steal
   %idle
14:53:19        all     11,70      0,00      5,85      0,00      0,00
   82,44
14:53:21        all     10,65      0,00      4,94      0,00      0,00
   84,42
14:53:23        all     11,79      0,00      5,64      0,00      0,00
   82,56
14:53:25        all      9,09      0,00      3,12      0,00      0,00
   87,79
14:53:27        all      9,14      0,00      2,79      0,25      0,00
   87,82
Media:          all     10,48      0,00      4,47      0,05      0,00
   85,00
```

Ahora medimos el consumo de la ram.

```
root@sololinux:~# sar -r 2 5
Linux 5.4.0-62-generic (sololinux)  17/01/21  _x86_64_ (2 CPU)
14:54:16    kbmemfree   kbavail kbmemused  %memused kbbuffers
kbcached  kbcommit   %commit  kbactive   kbinact   kbdirty
14:54:18       236064   1563464   3697876     94,00     76920
1794392   7225164    119,79   1912904   1473976       300
14:54:20       237324   1565780   3696616     93,97     76920
1792708   7225164    119,79   1912496   1474744       316
14:54:22       239844   1569356   3694096     93,90     76920
1790660   7225164    119,79   1911888   1472680       328
14:54:24       240348   1569868   3693592     93,89     76928
1790088   7225164    119,79   1911540   1472680       392
14:54:26       240600   1570120   3693340     93,88     76928
1790088   7225164    119,79   1911900   1472680       408
Media:         238836   1567718   3695104     93,93     76923
1791587   7225164    119,79   1912146   1473352       349
```

Las estadísticas de consumo por bloques en dispositivo, también es interesante.

```
sar -d -p 2 5
```

```
root@sololinux:~# sar -d -p 2 5
Linux 5.4.0-62-generic (sololinux)  17/01/21  _x86_64_ (2
CPU)
14:56:32          DEV       tps     rkB/s     wkB/s   areq-
sz    aqu-sz     await     svctm     %util
14:56:34          sda      2,50    480,00    154,00
253,60      0,01      4,20      5,60      1,40
14:56:34          sdb      0,00      0,00      0,00
0,00      0,00      0,00      0,00      0,00
14:56:34          DEV       tps     rkB/s     wkB/s   areq-
sz    aqu-sz     await     svctm     %util
14:56:36          sda      0,00      0,00      0,00
0,00      0,00      0,00      0,00      0,00
14:56:36          sdb      0,00      0,00      0,00
0,00      0,00      0,00      0,00      0,00
14:56:36          DEV       tps     rkB/s     wkB/s   areq-
sz    aqu-sz     await     svctm     %util
14:56:38          sda      7,96      0,00     95,52
12,00      0,02      3,06      3,50      2,79
14:56:38          sdb      0,00      0,00      0,00
0,00      0,00      0,00      0,00      0,00
14:56:38          DEV       tps     rkB/s     wkB/s   areq-
sz    aqu-sz     await     svctm     %util
14:56:40          sda      0,00      0,00      0,00
0,00      0,00      0,00      0,00      0,00
14:56:40          sdb      0,00      0,00      0,00
0,00      0,00      0,00      0,00      0,00
14:56:40          DEV       tps     rkB/s     wkB/s   areq-
sz    aqu-sz     await     svctm     %util
14:56:42          sda      9,50     32,00    140,00
18,11      0,02      3,11      3,37      3,20
14:56:42          sdb      0,00      0,00      0,00
0,00      0,00      0,00      0,00      0,00
Media:            DEV       tps     rkB/s     wkB/s   areq-
sz    aqu-sz     await     svctm     %util
Media:            sda      4,00    102,30     77,92
45,10      0,01      3,23      3,70      1,48
Media:            sdb      0,00      0,00      0,00
0,00      0,00      0,00      0,00      0,00
```

Podemos generar informes, incluso de la actividad I/O (entrada/salida).

```
sar -b 2 5
```

```
root@sololinux:~# sar -b 2 5
Linux 5.4.0-62-generic (sololinux)      17/01/21        _x86_64_
                                                  www.sololinux.es
14:59:47          tps      rtps      wtps   bread/s   bwrtn/s
14:59:49         0,50      0,50      0,00     16,00      0,00
14:59:51         4,00      3,00      1,00   2032,00     28,00
14:59:53         2,00      1,00      1,00     32,00     32,00
14:59:55         1,50      0,00      1,50      0,00    136,00
14:59:57        14,00      0,00     14,00      0,00    128,00
Media:           4,40      0,90      3,50    416,00     64,80
```

Guardar registros de sar

Con la «**-o**», puedes guardar los registros en un archivo codificado. Vemos un ejemplo.

```
sar -r 2 5 -o /home/logs-sar
```

En este post hemos visto los ejemplos más habituales de uso de **sar**, si quieres puedes leer su manual completo (incluyendo todas sus opciones), «**aquí**».

Solución de problemas

En alguna versión de Debian o derivados, es posible que dispare el siguiente error:

```
Cannot open /var/log/sysstat/sdx: No such file or directory
Please check if data collecting is enabled in
/etc/default/sysstat
```

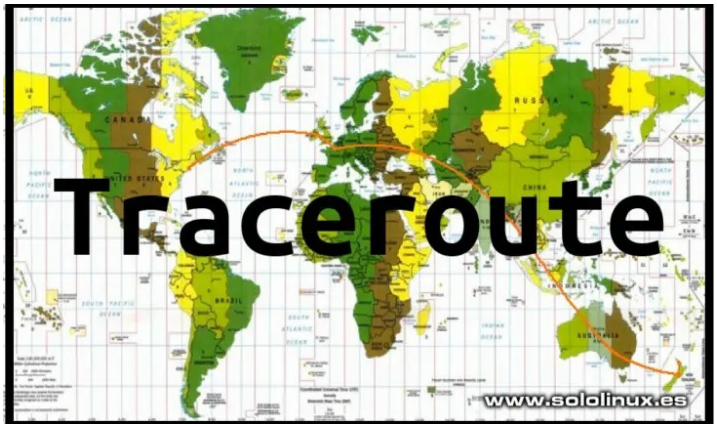
la soluciones sencilla, accede a…

```
sudo nano /etc/default/sysstat
```

Una vez en el archivo, asegúrate de como lo tienes, activado o desactivado.

```
# Should sadc collect system activity informations? Valid
values
# are "true" and "false". Please do not put other values,
they
# will be overwritten by debconf!
ENABLED="true"
```


Como usar traceroute en linux



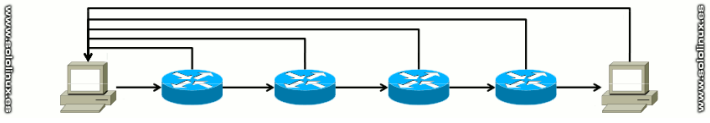
Traceroute es una herramienta para Linux, que nos permite realizar un seguimiento sobre las rutas de los paquetes de red. Al identificar el viaje de los paquetes, nos resulta de gran utilidad para detectar los problemas existentes en **conexiones de red** lentas.

La forma de operar de **Traceroute** es simple, básicamente se dedica al envío de paquetes de datos a un destino, ya sean computadoras, servidores o **sitios web**. En este proceso, va registrando todos los pasos y saltos intermedios a través de los cuales viajan los paquetes.

La salida del **comando traceroute**, serán las direcciones IP y los nombres de dominio por donde circulan los paquetes. Estas entradas, también nos informan sobre el tiempo que tardan los paquetes en llegar a su destino. Esto es extremadamente útil, para poder dar una explicación al porqué algunos sitios web tardan mucho en cargar.

Otra particularidad de usar **traceroute**, es mapear redes locales. Obtenemos información sobre la topología y las conexiones existentes en la red local.

Antes de aprender a **usar traceroute**, debes comprender que algunos dispositivos no interactúen correctamente. Este efecto se produce por errores en algunos modelos de enrutadores, por ISP que limitan la velocidad de los mensajes **ICMP**, o por seguridades añadidas que evitan el envío de **paquetes ICMP** (para evitar ataques **DoS**).



Como usar traceroute en linux

A pesar de las excelencias del comando traceroute, este no suele venir instalado en casi ninguna distribución linux. Por tanto, instalamos la herramienta.

Instalar traceroute en linux

Instalar traceroute en debian, Ubuntu y derivados

```
sudo apt install traceroute
```

Instalar traceroute en Fedora y derivados

```
sudo dnf install traceroute
```

Instalar traceroute en OpenSuse y derivados

```
sudo zypper in traceroute
```

Instalar traceroute en Arch Linux, Manjaro y derivados

```
sudo pacman -S traceroute
```

```
root@sololinux:~# sudo apt install traceroute
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  traceroute
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 45,4 kB de archivos.
Se utilizarán 152 kB de espacio de disco adicional después de esta operación.
Des:1 http://mirror.datacenter.by/ubuntu bionic/universe amd64 traceroute amd64 1:2.1.0-2 [45,4 kB]
Descargados 45,4 kB en 0s (332 kB/s)
Seleccionando el paquete traceroute previamente no seleccionado.
(Leyendo la base de datos ... 343392 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../traceroute_1%3a2.1.0-2_amd64.deb ...
Desempaquetando traceroute (1:2.1.0-2) ...
Configurando traceroute (1:2.1.0-2) ...
update-alternatives: utilizando /usr/bin/traceroute.db para proveer /usr/bin/traceroute (traceroute) en modo automático
update-alternatives: utilizando /usr/bin/lft.db para proveer /usr/bin/lft (lft) en modo automático
update-alternatives: utilizando /usr/bin/traceproto.db para proveer /usr/bin/traceproto (traceproto) en modo automático
update-alternatives: utilizando /usr/sbin/tcptraceroute.db para proveer /usr/sbin/tcptraceroute (tcptraceroute) en modo automático
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...
root@sololinux:~#
```


Usar traceroute en linux

En modo básico

La sintaxis de uso en modo básico es muy simple.

```
traceroute [dominio / ip]
```

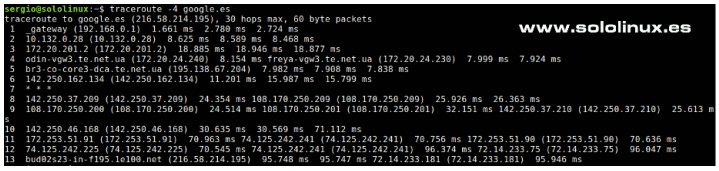
```
sergio@sololinux:~$ traceroute google.es
traceroute to google.es (216.58.214.195), 30 hops max, 60
byte packets
 1  _gateway (192.168.0.1)  3.356 ms  3.519 ms  4.280 ms
 2  10.132.0.28 (10.132.0.28)  5.801 ms  5.504 ms  4.779 ms
 3  172.20.201.2 (172.20.201.2)  24.962 ms  24.963 ms
24.931 ms
 4  freya-vgw3.te.net.ua (172.20.24.230)  4.327 ms  4.287 ms
 4.248 ms
 5  br3-co-ch2a-to-core4-dca.te.net.ua (195.138.67.206)
5.098 ms  5.074 ms  5.034 ms
 6  142.250.162.134 (142.250.162.134)  12.982 ms  12.116 ms
10.905 ms
 7  * * *
 8  142.250.238.0 (142.250.238.0)  23.717 ms  24.611 ms
142.250.37.209 (142.250.37.209)  24.563 ms
 9  108.170.250.201 (108.170.250.201)  25.170 ms
142.250.37.211 (142.250.37.211)  25.562 ms  26.892 ms
10  216.239.35.132 (216.239.35.132)  26.842 ms 72.14.237.108
(72.14.237.108)  32.139 ms  32.963 ms
11  74.125.242.241 (74.125.242.241)  32.796 ms  32.140 ms
172.253.51.91 (172.253.51.91)  39.476 ms
12  72.14.233.75 (72.14.233.75)  32.729 ms 74.125.242.225
(74.125.242.225)  30.883 ms 74.125.242.241 (74.125.242.241)
33.803 ms
13  72.14.233.181 (72.14.233.181)  32.346 ms 72.14.233.75
(72.14.233.75)  33.260 ms 72.14.233.181 (72.14.233.181)
32.556 ms
14  bud02s23-in-f3.1e100.net (216.58.214.195)  31.029 ms
30.823 ms  43.290 ms
sergio@sololinux:~$ traceroute 1.0.0.1
traceroute to 1.0.0.1 (1.0.0.1), 30 hops max, 60 byte
packets
 1  _gateway (192.168.0.1)  7.242 ms  9.601 ms  9.218 ms
 2  10.132.0.28 (10.132.0.28)  9.169 ms  9.093 ms  9.032 ms
 3  172.20.201.2 (172.20.201.2)  21.209 ms  21.172 ms
21.296 ms
 4  odin-vgw3.te.net.ua (172.20.24.240)  8.795 ms  9.185 ms
freya-vgw3.te.net.ua (172.20.24.230)  9.141 ms
 5  br4-dca-to-core3-co.te.net.ua (195.138.67.31)  9.223 ms
br4-dca-to-core4-dca.te.net.ua (195.138.67.21)  9.152 ms
br4-dca-to-core3-co.te.net.ua (195.138.67.31)  9.103 ms
 6  cloudflare-ix.giganet.ua (185.1.62.76)  14.467 ms
12.319 ms  10.769 ms
 7  one.one.one.one (1.0.0.1)  10.240 ms  11.223 ms  11.976
ms
```

Nota: Si aparecen asteriscos en algunas líneas, es porque existen medidas anti traceroute.

Traceroute con IPv4 o IPv6

Por defecto la herramienta utiliza el protocolo definido por el sistema, puedes modificarlo con las opciones «-4» y «-6».

```
traceroute -4 [dominio o ip]
traceroute -6 [dominio o ip]
```



Verificar un puerto específico

También es posible verificar la conexión de un puerto específico, para ello usamos el indicador «-p» seguido del número de puerto.

```
traceroute -p [numero de puerto] [dominio / ip]
```

Vemos un ejemplo con el **puerto 80**.

```
sergio@sololinux:~$ traceroute -p 80 google.es
traceroute to google.es (216.58.214.195), 30 hops max, 60 byte packets
 1  _gateway (192.168.0.1)  1.393 ms  2.073 ms  2.739 ms
 2  10.132.0.28 (10.132.0.28)  4.305 ms  4.236 ms  4.175 ms
 3  172.20.201.2 (172.20.201.2)  19.287 ms  19.256 ms  18.975 ms
 4  freya-vgw3.te.net.ua (172.20.24.230)  4.050 ms  3.992 ms  3.926 ms
 5  br3-co-ch2a-to-core4-dca.te.net.ua (195.138.67.206)  3.317 ms
3.534 ms br3-co-core3-dca.te.net.ua (195.138.67.204)  3.471 ms
 6  142.250.162.134 (142.250.162.134)  12.091 ms  11.527 ms  11.370 ms
 7  * * *
 8  * * *
 9  * * *
10  * * *
```

Quitar los nombres de dispositivos

Si observas las salidas anteriores del comando, se visualiza el nombre de los dispositivos. Esto puede confundir al usuario, ya que se mezclan un poco los datos. Usamos el indicador «**-n**».

```
traceroute -n [dominio o ip]
```

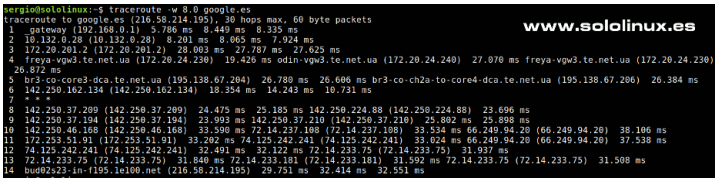
Ejemplo sin nombre de dispositivos.

```
sergio@sololinux:~$ traceroute -n google.es
traceroute to google.es (216.58.214.195), 30 hops max, 60
byte packets
 1  192.168.0.1  1.273 ms  2.367 ms  1.793 ms
 2  10.132.0.28  2.454 ms  2.339 ms  6.360 ms
 3  172.20.201.2  24.004 ms  24.040 ms  23.912 ms
 4  172.20.24.230  5.068 ms  6.280 ms 172.20.24.240  6.296
ms
 5  195.138.67.206  6.202 ms  6.165 ms  6.068 ms
 6  142.250.162.134  14.552 ms  14.637 ms  14.611 ms
 7  * * *
 8  142.250.37.209  24.930 ms 142.250.224.88  23.752 ms
108.170.250.209  25.789 ms
 9  108.170.250.200  24.146 ms 142.250.37.194  24.544 ms
108.170.250.200  24.464 ms
10  142.250.46.168  30.985 ms  30.126 ms 64.233.175.142
29.889 ms
11  74.125.242.225  30.907 ms 74.125.242.241  33.050 ms
32.948 ms
12  74.125.242.241  32.906 ms  32.138 ms 74.125.242.225
31.476 ms
13  72.14.233.181  32.716 ms 216.58.214.195  29.881 ms
72.14.233.181  32.151 ms
```

Limitar la espera del comando

Al usar traceroute, este tiene un tiempo de espera máximo predeterminado de 5 segundos para cada respuesta. Con la indicación «**-w**» es posible modificar el tiempo. Debes usar un carácter numérico en segundos, con punto flotante.

```
traceroute -w [segundos] [dominio o ip]
```



Especificar el máximo de saltos

Por defecto, al usar traceroute contamos con un máximo de 30 saltos hasta el destino. Es posible que necesites más, o por el contrario que solo quieras visualizar los primeros. La solución es el indicador «-m» y, el número entero del valor que necesites.

```
traceroute -m [valor numerico entero] [dominio / ip]
```

En el ejemplo, vemos los cinco primeros saltos.

```
sergio@sololinux:~$ traceroute -m 5 google.es
traceroute to google.es (216.58.214.195), 5 hops max, 60 byte packets
 1  _gateway (192.168.0.1)  1.578 ms  2.892 ms  2.754 ms
 2  10.132.0.28 (10.132.0.28)  3.637 ms  4.620 ms  4.535 ms
 3  172.20.201.2 (172.20.201.2)  20.991 ms  20.759 ms  20.695 ms
 4  odin-vgw3.te.net.ua (172.20.24.240)  3.258 ms freya-vgw3.te.net.ua
(172.20.24.230)  4.700 ms  4.432 ms
 5  br3-co-core3-dca.te.net.ua (195.138.67.204)  3.995 ms br3-co-ch2a-
to-core4-dca.te.net.ua (195.138.67.206)  4.130 ms br3-co-core3-
dca.te.net.ua (195.138.67.204)  3.981 ms
```

Usar una interfaz especifica

Como no podía ser menos, al usar traceroute también se nos permite definir el dispositivo de red deseado. En este caso utilizamos «**-i**». En este caso… sudo es obligatorio.

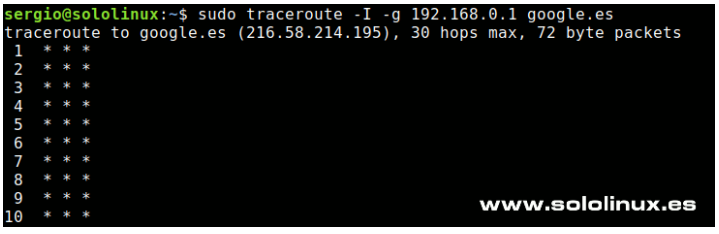
```
sudo traceroute -i [dispositivo] [dominio / ip]
```

```
sergio@sololinux:~$ sudo traceroute -i wlo1 1.0.0.1
[sudo] contraseña para sergio:
traceroute to 1.0.0.1 (1.0.0.1), 30 hops max, 60 byte packets
 1  _gateway (192.168.0.1)  2.907 ms  4.369 ms  4.352 ms
 2  10.132.0.28 (10.132.0.28)  4.307 ms  4.172 ms  4.126 ms
 3  172.20.201.2 (172.20.201.2)  22.921 ms  23.973 ms  23.930 ms
 4  freya-vgw3.te.net.ua (172.20.24.230)  5.312 ms  5.483 ms  5.443 ms
 5  br4-dca-to-core4-dca.te.net.ua (195.138.67.21)  5.400 ms  5.343 ms
 5.302 ms
 6  cloudflare-ix.giganet.ua (185.1.62.76)  34.842 ms  30.174 ms
26.955 ms
 7  one.one.one.one (1.0.0.1)  9.563 ms  10.219 ms  15.271 ms
```

Definir la puerta de enlace

Para enrutar los paquetes a través de una puerta de enlace definida, usa la opción «-g», seguida de la puerta de enlace. Debo aclarar, que esta función no siempre es efectiva cuando trazas servidores externos a tu red.

```
sudo traceroute -i [dispositivo] [dominio / ip]
```



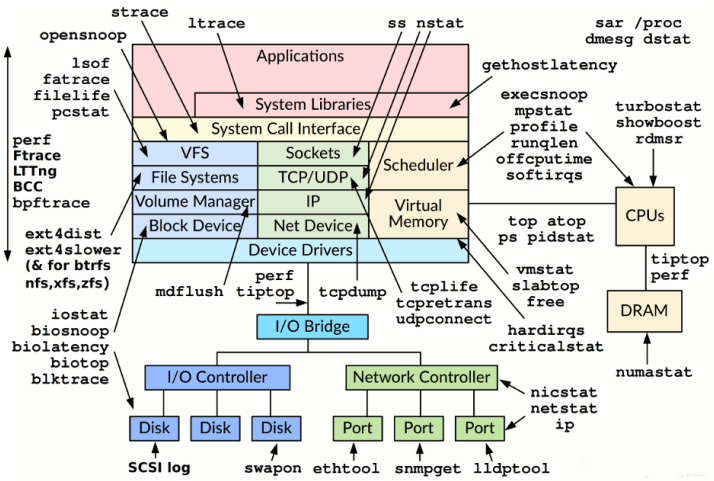
Ayuda de traceroute

En este artículo, hemos analizado los comandos más utilizados de traceroute. Puedes revisar el completo manual que integra la herramienta, con el siguiente comando.

```
traceroute --help
```

Uso del Comando straCe en Linux

Examinar y comprender las acciones hechas por los procesos y, las llamadas al sistema por ejecución, es una tarea que en algunos casos nos pueden ayudar a solventar problemas en nuestro sistema.
Así como existe **ptrace** que nos ayuda a depurar y diagnosticar procesos del kernel, tenemos el **comando strace** que tiene la capacidad de rastrear y monitorizar procesos, de forma que nos facilite el trabajo de depurar los procesos que interactúan con el Kernel.
En este artículo vemos algunos ejemplos de cómo usar la herramienta **strace**, para rastrear, **monitorear** y depurar procesos. Pero antes, debes tener claro… el significado de «**llamadas al sistema**».
Cada vez que se ejecuta una aplicación o herramienta, se lanza una solicitud de lectura, escritura, borrado, salida, enlace, etc., se realiza una llamada al sistema.
Las llamadas pueden ser producidas por cualquier motivo, y en un **sistema** que funciona correctamente son constantes y diversas. Comando strace.



Uso del comando strace en linux

La **herramienta strace**, no viene preinstalada en las distribuciones Linux corrientes, pero si la encontramos en sus repositorios oficiales; puedes instalarlo de manera sencilla desde el administrador de paquetes del sistema.

Instalar strace

Instalar strace en Debian, Ubuntu y derivados

```
sudo apt install strace
```

Instalar strace en CentOS, Fedora y derivados

```
sudo yum install strace
# o
sudo dnf install strace
```

Instalar strace en Arch Linux, Manjaro y derivados

```
[root@185 ~]# yum install strace
Complementos cargados:fastestmirror
Loading mirror speeds from cached hostfile
 * base: nl.mirrors.clouvider.net
 * epel: mirror.hostnet.nl
 * extras: mirror.ams1.nl.leaseweb.net
 * remi: mirrors.ukfast.co.uk
 * remi-php73: mirrors.ukfast.co.uk
 * remi-php74: mirrors.ukfast.co.uk
 * remi-safe: mirrors.ukfast.co.uk
 * remi-test: mirrors.ukfast.co.uk
 * updates: mirror.ams1.nl.leaseweb.net
Resolviendo dependencias
--> Ejecutando prueba de transacción
---> Paquete strace.x86_64 0:4.24-6.el7 debe ser instalado
--> Resolución de dependencias finalizada
Dependencias resueltas
================================================================================
==========
 Package          Arquitectura     Versión                Repositorio
   Tamaño
================================================================================
==========
Instalando:
 strace           x86_64           4.24-6.el7             base
   902 k
Resumen de la transacción
================================================================================
==========
Instalar  1 Paquete
Tamaño total de la descarga: 902 k
Tamaño instalado: 1.6 M
Is this ok [y/d/N]:
```

Uso del comando strace

Strace básico

En nuestro primer ejemplo, nos apoyamos en «ls». La salida es total, a lo mejor un poco liosa.

```
sudo strace ls
```

Si por ejemplo separamos la primera línea, identificamos tres partes bien diferenciadas.

- El nombre de la llamada al sistema.
- Argumentos que hacen la llamada al sistema (entre paréntesis).
- El valor de retorno de la llamada, en este caso «0».

También destacamos en la salida del **comando strace,** que la anotación entre barras inclinadas, son las variables agregadas para obtener el resultado. En este caso 31…

```
/* 31 vars */
```

Rastrear una llamada especifica

El comando strace nos ofrece mucha información sobre las llamadas al sistema, pero tal vez… nosotros solo estamos interesados en una en particular.

Aprovechando que en el ejemplo anterior usamos «ls», es la llamada que vamos a buscar. Nos ayudamos de «e» y «read».

```
sudo strace -e read ls
```

En nuestro caso nos aparece lo siguiente.

```
root@sololinux:~# strace -e read ls
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\20b\
0\0\0\0\0\0"..., 832) = 832
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\
0\1\0\0\0\20\35\2\0\0\0\0\0"..., 832) = 832
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0 \
25\0\0\0\0\0\0"..., 832) = 832
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\
16\0\0\0\0\0\0"..., 832) = 832
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0000b\
0\0\0\0\0\0"..., 832) = 832
read(3, "nodev\tsysfs\nnodev\ttmpfs\nnodev\tbd"..., 1024) = 400
read(3, "", 1024)                       = 0
+++ exited with 0 +++
```

La petición ejecutada, solo nos muestra las llamadas al sistema que fueron leídas, es lo que necesitamos en este caso.

Resumen de las llamadas al sistema

Obtener un resumen de llamadas, también es tarea simple con el comando strace. En este caso usaremos el argumento «**c**» o «**–summary-only**».

```
sudo strace -c ls
```

```
root@sololinux:~# strace -c ls
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 35.24    0.000562          33        17           mmap
 14.86    0.000237          20        12           mprotect
 13.73    0.000219          24         9           openat
 11.97    0.000191          24         8         8 access
  7.52    0.000120          13         9           fstat
  7.15    0.000114          16         7           read
  6.83    0.000109          10        11           close
  1.00    0.000016          16         1           execve
  0.88    0.000014          14         1           arch_prctl
  0.82    0.000013           4         3           brk
  0.00    0.000000           0         1           munmap
  0.00    0.000000           0         2           rt_sigaction
  0.00    0.000000           0         1           rt_sigprocmask
  0.00    0.000000           0         2           ioctl
  0.00    0.000000           0         2           getdents
  0.00    0.000000           0         2         2 statfs
  0.00    0.000000           0         1           set_tid_address
  0.00    0.000000           0         1           set_robust_list
  0.00    0.000000           0         1           prlimit64
------ ----------- ----------- --------- --------- ----------------
100.00    0.001595                    91        10 total
```

Otra opción interesante es, usar el argumento «**C**». Con este argumento, el comando strace nos imprime dos salidas, una normal y otra resumida.

```
sudo strace -C ls
```

```
access("/etc/selinux/config", F_OK)     = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=7129424, ...}) = 0
mmap(NULL, 7129424, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f47d45c7000
close(3)                                = 0
ioctl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0
ioctl(1, TIOCGWINSZ, {ws_row=36, ws_col=135, ws_xpixel=0, ws_ypixel=0}) = 0
openat(AT_FDCWD, ".", O_RDONLY|O_NONBLOCK|O_CLOEXEC|O_DIRECTORY) = 3
fstat(3, {st_mode=S_IFDIR|0700, st_size=4096, ...}) = 0
getdents(3, /* 16 entries */, 32768)    = 496
getdents(3, /* 0 entries */, 32768)     = 0
close(3)                                = 0
close(1)                                = 0
close(2)                                = 0
exit_group(0)                           = ?
+++ exited with 0 +++
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 25.22    0.000591          35        17           mmap
 17.20    0.000403          34        12           mprotect
 13.19    0.000309          34         9           openat
 11.61    0.000272          34         8         8 access
  7.04    0.000165          24         7           read
  6.74    0.000158          79         2           getdents
  5.16    0.000121          13         9           fstat
  4.48    0.000105          10        11           close
  2.52    0.000059          59         1           munmap
```


Llamadas de proceso en ejecución en tiempo real

Para esta acción debes conocer el **PID** del proceso, para ello utilizamos por ejemplo, **ps + grep**. En nuestro ejemplo buscamos el **PID** de **XORG**.

```
ps -ax | grep -i xorg
```

Nos encontramos con una salida similar a la siguiente.

```
sergio@sololinux:~$ ps -ax | grep -i xorg
  831 tty7     Ssl+   0:20 /usr/lib/xorg/Xorg -core :0 -seat seat0 -
auth /var/run/lightdm/root/:0 -nolisten tcp vt7 -novtswitch
 2350 pts/0    S+     0:00 grep --color=auto -i xorg
```

El **PID** es «831», por tanto para visualizarlo en tiempo real ejecuta el **comando strace** con la opción «**-p**».

```
sudo strace -p 831
```

AVISO: Ten cuidado con esta opción, puede llegar a saturar el sistema y provocar un bloqueo total.

Manual de strace

Para finalizar este artículo, te recomiendo que revises el manual oficial de strace.

```
sudo strace -h
```

```
root@sololinux:~# strace -h
usage: strace [-CdffhiqrtttTvVwxxy] [-I n] [-e expr]...
              [-a column] [-o file] [-s strsize] [-P path]...
              -p pid... / [-D] [-E var=val]... [-u username] PROG
[ARGS]
   or: strace -c[dfw] [-I n] [-e expr]... [-O overhead] [-S sortby]
              -p pid... / [-D] [-E var=val]... [-u username] PROG
[ARGS]
Output format:
  -a column      alignment COLUMN for printing syscall results
(default 40)
  -i             print instruction pointer at time of syscall
  -k             obtain stack trace between each syscall
(experimental)
  -o file        send trace output to FILE instead of stderr
  -q             suppress messages about attaching, detaching, etc.
  -r             print relative timestamp
  -s strsize     limit length of print strings to STRSIZE chars
(default 32)
  -t             print absolute timestamp
  -tt            print absolute timestamp with usecs
  -T             print time spent in each syscall
……...y mas
```

Wifislax 2.4 64bits – El linux forense español

Los chicos de seguridadwireless.net, acaban de lanzar la última versión de Wifislax, la 2.4 de 64 bits. Hablar de Wifislax son palabras mayores, es una de las distribuciones linux expertas en pruebas de seguridad, y análisis forenses más veteranas que puedes encontrar. Además… es española.
Este excelente live linux, es un tanto especial. A diferencia de otras (como puede ser **Kali**), no cuanta con un gran equipo de desarrollo a sus espaldas, todo se cocina en el foro de seguridadwireless. Este hecho no le quita valor a Wifislax, todo lo contrario; Es mucho más intuitiva y fácil de usar que otras como **Kali Linux**, sobre todo para lanzar pruebas sobre redes Wifi.

Altamente recomendada para los usuarios más noveles en linux

Wifislax 2.4 64bits – El linux forense español
Wifislax sigue utilizando como base **slackware64-14.2**, incluyendo todos parches de seguridad actualizados. Esta iso es un punto de control para la próxima versión estable, se han actualizado todos los paquetes del linux base, como firmwares, el kernel, el navegador y resto de aplicaciones.

Como no podía ser de otro modo, viene con el **kernel linux 5.4.91 LTS** con todos sus parches de seguridad y, reconocimiento de hardware (incluyendo los **dispositivos Realtek** problemáticos). Como navegador se monta el nuevo Chrome 88, que además de parches de seguridad, se elimina el soporte de **flash player**.

La **nueva versión**, también incluye todas las gemas **ruby** necesarias para utilizar hostbase. El usuario koala tiene abierto un hilo para pruebas en este **enlace**. Destacamos que se actualiza **airgeddon** a su nueva versión 10.40, ademas de otras mejoras en el rendimiento de la iso live y su funcionalidad.

Como es habitual en esta distribución especializada en análisis forense, y pruebas de penetración, puedes ampliar las funciones de **la live** con módulos extra. Visita el hilo del foro para obtenerlos.

Posiblemente, esta será la última versión de wifislax con kde4 y slackware64-14.2, lo más probable es que la siguiente versión ya venga con Slackware64 en su versión 15. Mientras tanto, puedes **descargar Wifislax64 2.4 64** desde el siguiente enlace.

- **Descargar Wifislax64 2.4**

13 Comandos linux que pueden destruir tu sistema

Los comandos linux que operan en la **shell de Linux** son muy poderosos, con solo un clic en la **tecla enter**… puedes bloquear el sistema, eliminar directorios imprescindibles, borrar archivos o incluso la carpeta raíz. Nuestro sistema ha sido destruido.

En algunos casos, Linux ni siquiera nos pide confirmación, tiene la capacidad de ejecutar el comando de inmediato. No será ni el primero, ni el último que destruye un sistema por no ser consciente de lo que estaba ejecutando. Algún día contaré la historia de un supuesto **sysadmin**, que acabó con los historiales (miles de pacientes) de un hospital privado en España, jajaj.

Ha llegado a mis oídos, que últimamente muchos **lamers** responden en foros o chats contenidos de supuesta ayuda, que más que ayudar, fastidian tu sistema. Parece motivo de risa, pero no lo es. No confíes en foros extraños y… muchos menos en sitios web con poca o ninguna reputación.

13 comandos linux que pueden destruir tu sistema

El siguiente listado de **comandos** son perfectamente útiles bajo entornos específicos, pero nada recomendados en sistemas estables y, aún menos si no dispones de suficientes conocimientos. No juegues con tu sistema, yo no soy el responsable de los posibles daños causados.

Formatear el sistema

El **administrador de Linux** usa este comando constantemente, cuando es necesario formatear o asignar un sistema de archivos a una partición. Si lo utilizas de forma errónea, también puedes formatear un disco con datos importantes. Por ejemplo.

```
mkfs.ext4 /dev/sda
```

Antes de ejecutar el anterior comando, asegúrate que seleccionaste el dispositivo o partición deseado.

Eliminar /etc y /boot

El directorio /etc, contiene los archivos de configuración del sistema. El directorio /boot, contiene archivos necesarios para el inicio del sistema, relacionados con el kernel, InitRD y **GRUB**. Jamás ejecutes los siguientes comandos.

```
rm -rf /etc
rm -rf /boot
```

Otra forma de destruir el sistema, es borrar los archivos de configuración. No lo hagas.

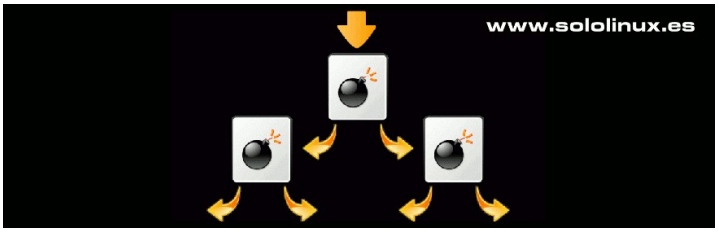
```
find / -iname "*.conf" -exec rm -rf {} \;
```

Eliminar el sistema de archivos

Esto elimina todo el sistema de archivos de tu servidor o desktop, cada byte de datos será borrado del disco. Ya sabes a lo que te expones al ejecutarlo.

```
rm -rf /
```

Bomba Fork



Este comando crea un sinfín de copias sobre sí mismo, hasta agotar los recursos del sistema provocando un bloqueo total de forma irremediable.

```
:(){ :|:& };:
```

En este **artículo anterior**, puedes aprender más sobre las **bombas fork**.

Llenar el disco de datos aleatorios

Estos comandos son útiles si quieres deshacerte de un disco y, que nadie pueda recuperar los datos. Si lo ejecutas en un disco por error, no podrás recuperar los datos contenidos.

```
dd if=/dev/urandom of=/dev/sda
```

Otra opción que sobrescribe el disco varias veces.

```
shred /dev/sda
```

13 comandos Linux que pueden destruir tu sistema

Desbarajuste en permisos de archivo
Todos los comandos mencionados anteriormente, eliminan o destruyen datos. Está claro que en el artículo **13 comandos linux que pueden destruir tu sistema**, debemos abordar otras opciones.

Los **permisos de archivos en linux**, son fundamentales para un correcto funcionamiento del sistema. Al ejecutar comandos erroneos sobre los permisos, puedes provocar un auténtico desbarajuste o caos sobre ellos, de forma que el sistema sea difícil de recuperar. Vemos algunos ejemplos.

El primer comando borra todos los permisos de los archivos y carpetas del sistema. El acceso resultará imposible.

```
chmod -Rv 000 /
```

Otro comando que logra el mismo objetivo que el anterior.

```
chown -R nobody:nobody /
```

Un comando opuesto a los anteriores, concederá permisos a todos y a todo. El riesgo es altísimo. Recuerdo con gracia, un usuario que tenía un **VPS** con varias web y, como tenía problemas con los permisos, ejecuto este comando. En pocas horas, decenas de intrusos le trastocaban el sistema, jajjaja.

```
chmod -R 777 /
```

Black Hole
El **Black Hole**, también conocido como agujero negro, nos indica que los datos han sido copiados o movidos correctamente, pero en realidad fueron descartados. Por ejemplo.

```
mv carpeta/dev/null
```

Algo similar…

```
Dev/null
```

El peligro de wget
Como punto final del artículo «**13 comandos linux que pueden destruir tu sistema**», vemos el **comando wget**.

Tal vez no seas consciente del peligro de este comando, tan utilizado por todos los **usuarios de linux**. El peligro de Wget no radica en el propio comando, sino en el usuario malintencionado que ofrece el enlace a un **script de shell** con auto-ejecución incluida. Este script puede contener código malicioso y, desconocemos sus posibles efectos.

El formato del comando es similar a lo siguiente.

```
wget https://sitioweb.com/script.sh -O- | sh
```

No ejecutes este tipo de comandos, a no ser que provengan de sitios o usuarios confiables.

Comparar archivos en Linux con el comando diff

El comando diff (diferencia), se usa para mostrar las diferencias entre archivos de linux, su funcionamiento es simple, compara los archivos línea por línea buscando caracteres que no sean iguales. A diferencia de otros comandos similares, **diff** tan solo compara e indica donde se encuentran las desigualdades localizadas.

Existen ciertos **símbolos e instrucciones** especiales que nos pueden ayudar a que los archivos sean idénticos, pero no es nuestro caso. Si quieres conocer esos argumentos especiales, revisa el manual del comando.

```
man diff
```

```
DIFF(1)                          User Commands                          DIFF(1)

NAME
       diff - compare files line by line

SYNOPSIS
       diff [OPTION]... FILES

DESCRIPTION
       Compare FILES line by line.

       Mandatory  arguments  to  long  options are mandatory for short options
       too.

       --normal
              output a normal diff (the default)

       -q, --brief
              report only when files differ

       -s, --report-identical-files
              report when two files are the same

       -c, -C NUM, --context[=NUM]
              output NUM (default 3) lines of copied context

       -u, -U NUM, --unified[=NUM]
              output NUM (default 3) lines of unified context

       -e, --ed
              output an ed script

       -n, --rcs                                        www.sololinux.es
              output an RCS format diff
```

Comparar archivos en linux con el comando diff
El **comando Diff** está disponible en casi todas las **distribuciones Linux** de forma predeterminada, por tanto no es necesario instalar nada. **Comparar archivos con diff** es tarea sencilla, observa su sintaxis.

```
diff [archivo1] [archivo2]
```

En nuestro ejemplo tenemos dos archivos, demo1.txt y demo2.txt. El primero contiene el siguiente texto.

Me
gusta
sololinux

El segundo…
sololinux
gusta
mucho

Ejecutamos nuestro comando de ejemplo.

```
diff demo1.txt demo2.txt
```

```
sergio@sololinux:~$ diff demo1.txt demo2.txt
1c1
< Me
---
> sololinux                                      www.sololinux.es
3c3
< sololinux
\ No hay ningún carácter de nueva línea al final del archivo
---
> mucho
\ No hay ningún carácter de nueva línea al final del archivo
sergio@sololinux:~$
```

Comando Vimdiff
El **comando Vimdiff** forma parte del editor **vim**, por lo que para poder usar **vimdiff** necesitamos tener **vim** instalado en nuestro sistema. **Vim** es uno de los editores preferidos por los profesionales del sector, a pesar de ello y debido a su curva de aprendizaje… poco a poco se está viendo relegado por otros más sencillos, como puede ser **nano**.

Si no tienes **Vim** instalado, tranquilo, seguro que lo tienes en los repositorios oficiales de tu sistema. Lo instalamos.

Debian, Ubuntu, Linux Mint y derivados:

```
sudo apt install vim
```

Rhel, Oracle, Centos, Fedora y derivados:

```
sudo yum install vim
# o
sudo dnf install vim
```

La sintaxis de **Vimdiff** es similar a la del comando **diff**.

```
vimdiff [archivo1] [archivo2]
```

```
vimdiff demo1.txt demo2.txt
```

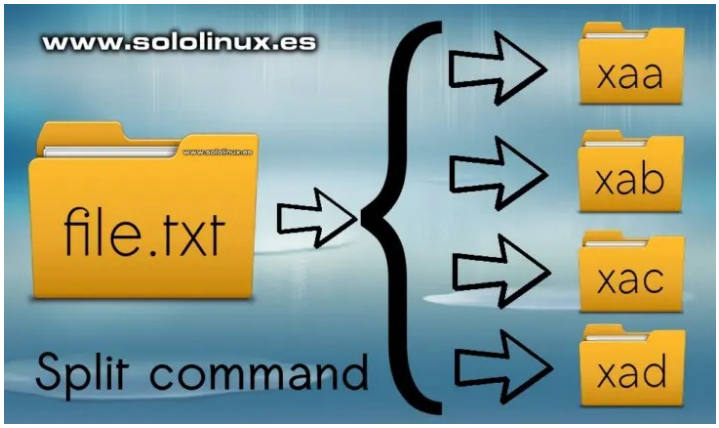
Como puedes observar, al comparar archivos con «**vimdiff**» la salida es mucho más visual que con el **comando diff**. Explicamos los colores.

- **Líneas en rojo:** Las líneas de color rojo nos indican que existen diferencias parciales, es decir, una parte de la línea marcada, no toda.
- **Líneas en azul y guiones:** El color azul nos dice que la línea entera no coincide.
- **Líneas en color de terminal:** El texto coincide.

Ya comentamos anteriormente, que el manejo del **editor Vim** no es fácil para los recién llegados a **Linux**. Como último apunte de este artículo, te propongo algunos atajos útiles de la herramienta «**vimdiff**».

- **]c :** Saltar a la siguiente diferencia.
- **[c :** Volver a la anterior diferencia.
- **dp :** Insertar la diferencia seleccionada en una nueva ventana.
- **do :** Obtener los cambios de otra ventana respecto a la actual.
- **CTRL-W + CTRL-W :** Pasar de una ventana a otra cuando están divididas.
- **zo :** Abre el contenido duplicado de los archivos.
- **zc :** Cierra el contenido duplicado de los archivos.
- **:diffupdate :** Busca de nuevo en los archivos localizando las modificaciones.

Uso del Comando split en linux



El **comando Split** en Linux, se usa para **dividir archivos grandes en archivos más pequeños**. Es algo normal que tengamos algunos archivos de gran tamaño, incluso algunos que vayan creciendo de forma incremental (normalmente registros).

Estos archivos tan grandes no son fáciles de leer y, aún menos de editar. Para solucionar estos problemas, **linux** nos ofrece un comando (desconocido por los usuarios noveles), con el cual podemos dividir un archivo en otros más pequeños (con mismo contenido), dependiendo de nuestras necesidades.

Uso del comando split en linux

Como ya comentamos anteriormente, **split** nos ayuda a dividir los archivos en otros más pequeños. Podemos realizar la división por números de línea, tamaño, longitud, y más. Es importante tener presente que de forma predeterminada, el **comando split** divide un archivo en archivos de 1000 líneas; por tanto, si tenemos un archivo con 2100 líneas, obtendremos tres archivos, dos con 1000, y uno con 100 líneas.

Su sintaxis básica es sencilla.

```
split [archivo]
```

Por ejemplo…

```
split script.txt
# o
split script.sh
```

Obtenemos tres archivos.

- **xaa**
- **xab**
- **xac**

Para nuestras necesidades, el archivo sigue siendo excesivamente grande. Con la opción «-l», podemos definir el número de líneas de los archivos. Mira que fácil es dividirlo en archivos de 150 líneas.

```
split -l 150 miscript.sh
```

Ejemplo…

```
split -l 150 libera.sh
```

Observa el resultado final.

También podemos aplicar un prefijo en particular.

```
split [archivo] [nombre de destino]
```

Vemos un ejemplo aplicando división por líneas.

```
split -l 10 script.sh miscript.sh
```

Listamos el resultado con el **comando ls**.

```
sergio@sololinux:~/demo$ ls
miscript.shaa  miscript.shac  miscript.shae  miscript.shag
miscript.shab  miscript.shad  miscript.shaf  script.sh
```

Como ultima opción interesante, te propongo dividir los archivos con el comando split por tamaño. Para ello usamos la opción «**-b**», seguido del tamaño máximo de cada archivo. Vemos un ejemplo en el cual generamos archivos de un mega.

```
split -b 1M script.sh
```

En el caso anterior, todos los archivos generados tienen un tamaño máximo de 1Mb.

Si quieres aprender más, puedes visualizar el manual de la herramienta split con el siguiente comando

```
man split
```

Manual del **comando split**.

```
SPLIT(1)                        User Commands
   SPLIT(1)
NAME          top
       split - split a file into pieces
SYNOPSIS          top
       split [OPTION]... [FILE [PREFIX]]
DESCRIPTION          top
       Output pieces of FILE to PREFIXaa, PREFIXab, ...;
default size is
       1000 lines, and default PREFIX is 'x'.
       With no FILE, or when FILE is -, read standard input.
       Mandatory arguments to long options are mandatory for
short
       options too.
       -a, --suffix-length=N
              generate suffixes of length N (default 2)
       --additional-suffix=SUFFIX
              append an additional SUFFIX to file names
       -b, --bytes=SIZE
              put SIZE bytes per output file
       -C, --line-bytes=SIZE
              put at most SIZE bytes of records per output
file
       -d     use numeric suffixes starting at 0, not
alphabetic
……...ABRE UN TERMINAL PARA VER MAS:…….
```

Modificar el límite de archivos abiertos en linux

Nuestro sistema linux, asigna de forma temporal un número denominado identificador cuando accedemos a un archivo. Por defecto, la memoria principal reserva un espacio para estos identificadores de archivos. La cantidad máxima de archivos abiertos, depende de la memoria que nuestro sistema asigne para tal efecto.

Los **procesos en Linux** tienen sus restricciones y, estás restricciones impiden ejecutar correctamente los procesos dependiendo de los límites. Debes recordar, que al ejecutar una herramienta, no solo se abre un archivo por aplicación, pueden ser decenas o incluso cientos.

Es evidente que el sistema necesita **memoria** para administrar cada archivo, y esto nos puede crear un problema de agotamiento de recursos en caso de tener que abrir más archivos de los recomendados, dado que a lo mejor no tienes límites o, estos son excesivamente altos para tu máquina. En este artículo, veremos como modificar el límite de archivos abiertos en nuestra **distribución linux**.

Modificar el límite de archivos abiertos en linux

En linux, tenemos dos tipos de límites definidos: **límite estricto** (o rígido) y el **límite flexible** (o suave).

- **El límite estricto** es un valor establecido estáticamente y, solo puede ser alterado por el usuario **root**.
- **El límite flexible** puede ser modificado por procesos de forma dinámica, es decir, en tiempo de ejecución, si el proceso necesita más archivos que el número permitido por el límite flexible actual.

Antes de comenzar, debes tener presente que los límites no es una ciencia exacta. Cada distribución linux los implanta como considera necesario.

Lo primero que hacemos es visualizar los límites establecidos.

Ver el número estricto de archivos

```
ulimit -Hn
```

```
sergio@sololinux:~$ ulimit -Hn
1048576
```

Ver el número flexible de archivos

```
ulimit -Sn
```

```
sergio@sololinux:~$ ulimit -Sn
1024
```

```
sergio@sololinux:~$ ulimit -Hn
1048576
sergio@sololinux:~$ ulimit -Sn
1024
www.sololinux.es
```

Modificar los límites de archivo

Podemos modificar los valores límite de forma simple. Vemos dos opciones, con permanencia y sin ella.

Límites sin permanencia

Modificar los límites sin permanencia (se perderán los cambios al reiniciar el sistema), es un atarea sencilla. Tan solo debes aplicar el número máximo de archivos abiertos, a los comandos que usamos anteriormente para verificar los valores.

Ejemplo.

```
ulimit -Hn 2000000
ulimit -Sn 3500
```

Límites con permanencia

Para establecer el límite de forma permanente, debes editar el archivo «**/etc/security/limits.conf**» con los permisos requeridos. Abrimos «**limits.conf**».

```
sudo nano /etc/security/limits.conf
```

```
# /etc/security/limits.conf                    www.sololinux.es
#
#Each line describes a limit for a user in the form:
#
#<domain>        <type>  <item>  <value>
#
#Where:
#<domain> can be:
#        - a user name
#        - a group name, with @group syntax
#        - the wildcard *, for default entry
#        - the wildcard %, can be also used with %group syntax,
#                 for maxlogin limit
#        - NOTE: group and wildcard limits are not applied to root.
#          To apply a limit to the root user, <domain> must be
#          the literal username root.
#
#<type> can have the two values:
#        - "soft" for enforcing the soft limits
```

Agrega al archivo la siguiente línea (con tu valor numérico requerido).

```
* hard nofile 2000000
```

Lo dicho anteriormente es válido para todos los usuarios del sistema, si quieres definir el valor por usuario, copia y pega lo siguiente (con el nombre de usuario).

```
nombre-de-usuario soft nofile 1200000
```

Nota final
Algunas **distribuciones linux** no ponen límites de archivos abiertos. Esto puede ser peligroso en máquinas con pocos recursos, son carne de cañón en trabajos exigentes. Puedes verificar si es tu caso, ejecutando ulimit.

```
ulimit
```

Ejemplo de sistema sin límites.

```
sergio@sololinux:~$ ulimit
unlimited
```

Verificar la suma de comprobación SHA256

La **suma de comprobación**, es una firma criptográfica específica de un archivo (en formato cadena), Esta firma representa de forma única el archivo, por lo que podemos deducir que al manipular el archivo, la firma cambia.

Existen varios algoritmos matemáticos para generar el **Checksum** de un archivo en Linux. Uno de esos algoritmos es **Secure Hash Algorithm 256**, de la **Agencia de Seguridad Nacional de los Estados Unidos;** Con eso está todo dicho.

Este algoritmo divide los datos del archivo en un tamaño menor y, combina y genera **valores hash** de cada parte; al sumarlos de nuevo crea el valor de suma de comprobación. La **suma de comprobación SHA256** genera un archivo de texto o una cadena anexa al archivo principal.

Verificar la suma de comprobación SHA256

Que mejor forma de aprender, que realizar un ejemplo práctico. En nuestro caso verificamos la descarga de **Ubuntu 20.04 LTS Focal Fossa**.

Descargamos la última versión estable.

```
wget https://releases.ubuntu.com/20.04/ubuntu-20.04.1-desktop-amd64.iso
```

También descargamos el archivo de suma de verificación.

```
wget http://releases.ubuntu.com/focal/SHA256SUMS
```

Como punto final, solo falta verificar la descarga. Para lograr nuestro objetivo ejecuta el siguiente comando.

```
sha256sum -c SHA256SUMS
```

Si la suma es correcta, obtendrás una respuesta similar a…

```
sergio@sololinux:~$ sha256sum -c SHA256SUMS
ubuntu-20.04.1-desktop-amd64.iso: La suma coincide
```

Ahora estamos seguros, de que el archivo **ISO** coincide con la **suma de comprobación original** y, por lo tanto, la descarga es original y no fue manipulada.

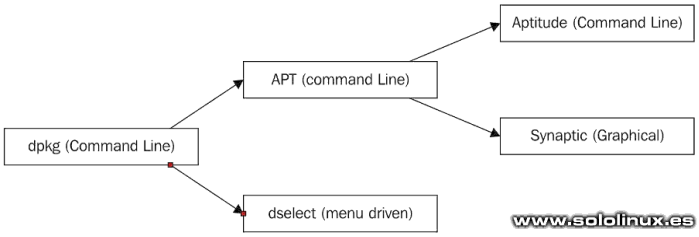
Esta revista es de distribución gratuita, si lo consideras oportuno puedes ponerle precio.
Tu también puedes ayudar, contamos con la posibilidad de
hacer donaciones para la REVISTA, de manera muy simple
a través de PAYPAL

AYUDANOS A SEGUIR CRECIENDO

Borrar la caché de Apt en Debian, Ubuntu y derivados

Lo que todos conocemos como **Apt (Advanced Packaging Tool)**, es la **herramienta de** instalación de paquetes y gestión de sus dependencias en Debian, Ubuntu y la práctica totalidad de distribuciones basadas en Debian. En contra de los que muchos piensan, la herramienta **apt** no instala nada por sí misma; en realidad es un avanzado **front-end** de la aplicación **dpkg**.

El **comando dpkg**, es una herramienta de bajo nivel; por tanto necesita de otra que opere en alto nivel, que sea capaz de descargar los paquetes remotos y resolver los posibles conflictos con las dependencias. En sistemas basados en Debian o Ubuntu, tenemos **apt**.



La forma en que **apt** instala los paquetes es la siguiente: descarga el paquete del software solicitado y, también descarga todas las dependencias que necesita la aplicación a instalar. Una vez descargados, se extraen los paquetes y se completa la instalación.

Estos paquetes descargados, una vez concluye la instalación se mueven a un directorio **caché** que podemos visualizar en: **/var/cache/apt/archives**. Muchas de las bibliotecas descargadas, también las encontraremos en el mismo directorio o, en otros similares. Enumeramos donde se guardan estos archivos.
/var/cache/apt/archives/

- ***/var/cache/apt/archives/partial/***
- ***/var/lib/apt/lists/partial/***
- ***/var/cache/apt/pkgcache.bin***
- ***/var/cache/apt/srcpkgcache.bin***

Estos paquetes se guardan para usarlos en otras instalaciones futuras. Sin embargo, a medida que el sistema acumula horas de uso e instalaciones, tenemos demasiados paquetes en la **caché**. Por lo tanto, es una buena práctica borrar la caché de Apt de vez en cuando, pues liberamos espacio ocupado y agilizamos nuestro sistema.

Borrar la caché de Apt en Debian, Ubuntu y derivados
Uno de los comandos más usados para borrar la **caché de apt**, es **clean**. Antes de borrar toda la caché, conviene saber que vamos a borrar, para ello ejecutamos el siguiente comando.

```
sudo apt clean --dry-run
```

Veremos algo similar a…

```
root@sololinux:~# sudo apt clean --dry-run
Del /var/cache/apt/archives/*
/var/cache/apt/archives/partial/*
Del /var/lib/apt/lists/partial/*
Del /var/cache/apt/pkgcache.bin
/var/cache/apt/srcpkgcache.bin
root@sololinux:~#
```

Bien, ahora borramos el contenido de la caché con el comando…

```
sudo apt clean
```

Ejemplo.

```
root@sololinux:~# sudo apt clean --dry-run
Del /var/cache/apt/archives/*
/var/cache/apt/archives/partial/*
Del /var/lib/apt/lists/partial/*
Del /var/cache/apt/pkgcache.bin
/var/cache/apt/srcpkgcache.bin
root@sololinux:~# sudo apt clean
root@sololinux:~#
```

En este momento, la **caché de apt** está borrada. Si lanzas un «update», verás que se descargan otra vez todos los repositorios instalados por completo.

```
sudo apt update
```

Similar al comando «**apt clean**», existe otro llamado «**apt autoclean**». Este último elimina los paquetes de **la caché**, en caso de que exista una versión más nueva en el repositorio. Lo lanzamos…

```
sudo apt autoclean
```

Recuerda que los paquetes en **caché** que permanecen, es porque aún no tienen una versión más reciente en el repositorio, **autoclean** no los elimina.

```
root@sololinux:~# sudo apt autoclean
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
root@sololinux:~#
```

Nota: **No confundas «autoclean» con «autoremove».**

Entrevista a Erwin Andres Espitia Torres, Admin de Espacio Tecnológico

Un número mas en la Revista **SOLOLINUX**, seguimos con las entrevistas a distintos sitios WEB relacionados con el mundo de **GNU/LINUX**. Unos de ellos son muy conocidos por la comunidad, otros no tanto.
Esta vez le toca a un BLOG el cual encontré por casualidad en la red. Se trata de Espacio Tecnológico Por el cual su **Admin Erwin Andres** no ha dudado ni un momento en respondernos a algunas preguntas sobre su creación.

Comenzamos con la Entrevista

SOLOLINUX: ¿Cuéntanos un poco sobre **espaciotecnologico**?
ERWIN ANDRES: *Espacio tecnológico es un blog que ofrece conocimiento tecnológico con prevalencia **FLOSS**, y más dirigido al público **GNU/Linux**.*

SOLOLINUX: *¿Cuando se empezó con este **BLOG**? Y ¿Como fue la idea para crearlo?*
ERWIN ANDRES: *Este proyecto nace en diciembre del año 2017. Surge por la necesidad personal de centralizar en un sitio los procedimientos, teorías, algunos desarrollos de software, asi como experiencias y anotaciones propias con la tecnología. En resumen, **espaciotecnologico** funciona como una bitácora profesional y laboral, la cual comparto en línea para cualquiera que la necesite.*

SOLOLINUX: *¿Hoy en día aproximadamente cuantas personas colaboran en el proyecto y cuales son sus tareas?*
ERWIN ANDRES: *Realmente soy la única persona que mantiene este proyecto. He realizado gestiones para que otros profesionales o aficionados aporten con su fuerza de trabajo, pero hasta ahora nadie resulta.*

SOLOLINUX: *¿A que perfil de usuarios estas destinados los artículos de **espaciotecnologico**?*
ERWIN ANDRES: *A usuarios con afinidad en la informática y la tecnología Como **espaciotecnologico** tiene cierta diversificación temática, ofrece conocimiento tanto para el público de **GNU/Linux**, como para **SEO managers**, también para **administradores de TI** y **estudiantes de ingeniería**, además de emprendedores y aficionados. Sin embargo, tiene algunas secciones comerciales en la que ofrezco servicios informáticos, para poder sufragar su existencia.*

SOLOLINUX: ¿Contáis con algún apoyo económico para mantener el proyecto?
ERWIN ANDRES: *Con ninguno. El blog* ***espaciotecnologico*** *nació como una bitácora personal para el apoyo profesional y laboral, y así lo he mantenido.*

SOLOLINUX: *¿Cuales son los fines principales de* ***espaciotecnologico****?*
ERWIN ANDRES: *Actualmente tiene por objeto servir como apoyo, guía y fuente de conocimiento técnico y tecnológico. En estos tres años que ha estado "****al aire****" buena parte de la comunidad estudiantil, tecnológica y emprendedora hispanohablante, ha encontrado en* ***espaciotecnologico*** *una respuesta directa a necesidades puntuales. Hace tres años comenzó recibiendo escasas 40 visitas al mes, y hoy asciende a una modesta cantidad 7000 visitas mensuales. De manera que éticamente hablando, este proyecto se convirtió en una responsabilidad social bien ganada, y seguiré asumiéndola mientras tenga los recursos.*

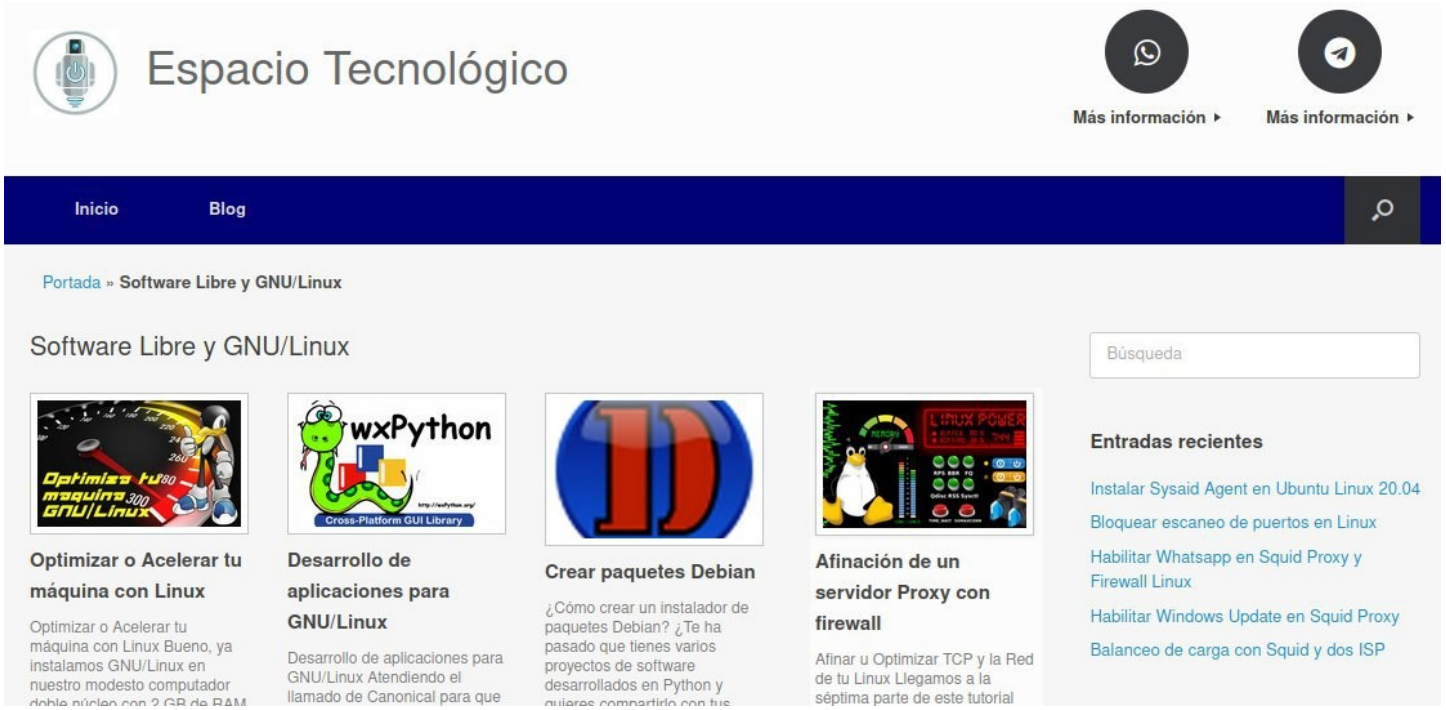
Buena parte del público consumidor de este recurso manifiesta sus agradecimientos en la sección de comentarios, y eso también me motiva a mantenerlo y hacerlo crecer.

SOLOLINUX: *¿Como podemos colaborar si es posible en la web?*
ERWIN ANDRES: *Claro que es posible colaborar, ya sea con contenido, con piezas gráficas o promocionandolo, y si es para el campo de la Educación, mucho mejor.*

SOLOLINUX: *¿Donde y como podemos ponernos en contacto con vosotros?*
ERWIN ANDRES: *Por medio de la dirección de correo* ***erwin.espitia.torres@gmail.com****, ese es el punto de partida.*

SOLOLINUX: *Y para terminar ¿Podrías darnos tu opinión sobre* ***GNU/LINUX y software libre****?*
ERWIN ANDRES: *Claro, para mí* ***GNU/Linux y el Software Libre*** *representan* ***recursos informáticos****,* ***éticos****,* ***legales*** *y* ***filosóficos*** *valiosísimos como garantes de nuestra libertad informática, como fuente de capital intelectual y como agentes en la reducción de la brecha digital.*

Bueno, por último quiero expresar mis agradecimientos a la revista ***SOLOLINUX*** *y a sus* ***colaborades****, por este espacio que me ofrecieron para compartir con sus lectores.*



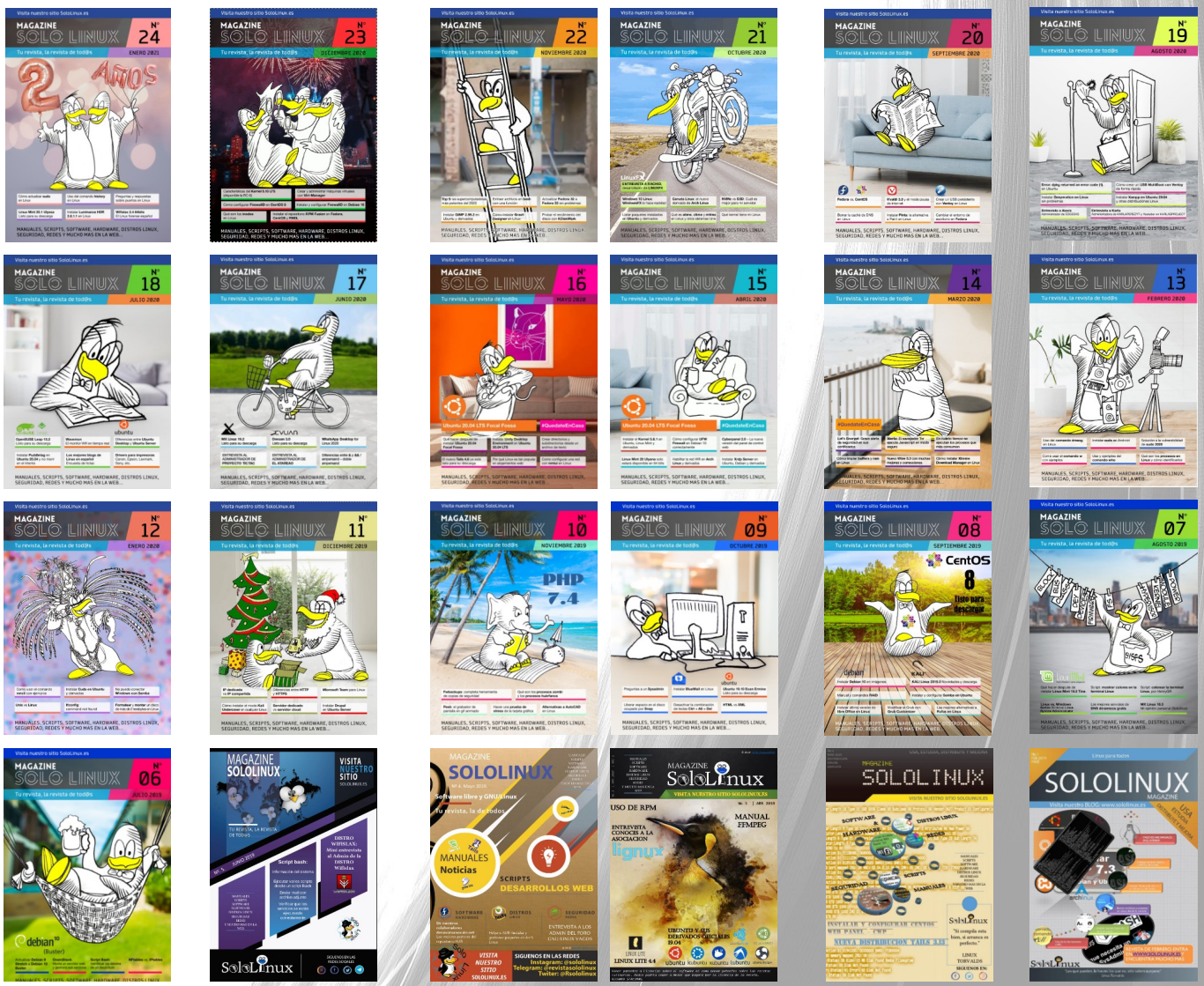
GRACIAS ERWIN ANDRES POR DEDICARNOS UN POCO DE TU TIEMPO PARA LA REALIZACION DE ESTA MINIENTREVISTA

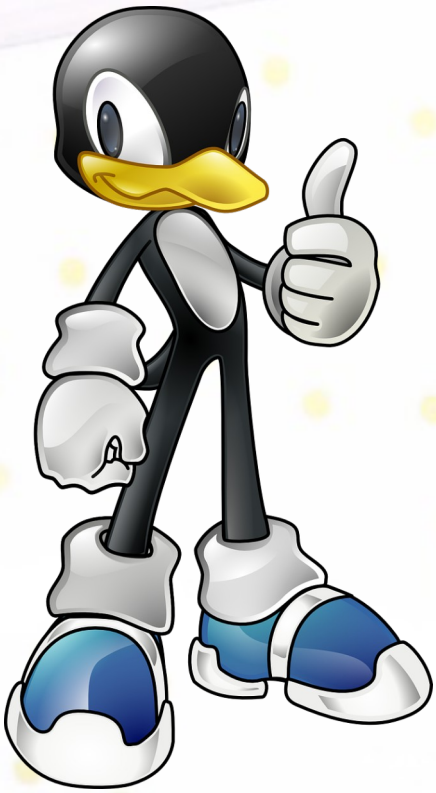
CELEBRAMOS DOS AÑOS DE LA **MAGAZINE SOLOLINUX**, REVISTA QUE NACIÓ UN 2 DE MARZO DEL AÑO 2019 SIENDO UNA COPIA EXACTA MENSUAL DEL SITIO WEB

WWW.SOLOLINUX.ES

EN PDF PARA LEER DONDE Y CUANDO QUIERA EL LECTOR SIN NECESIDAD DE DISPONER DE INTERNET EN EL MOMENTO DE SU LECTURA

SI TE PERDISTE ALGUNO DE NUESTROS DE NUESTROS NÚMEROS O QUIERES VOLVER A LEERLOS, TE INVITAMOS A VISITAR https://www.sololinux.es/revista-digital-magazine/





THIS IS YOUR YEAR

www.sololinux.es