

Visita nuestro sitio SoloLinux.es

MAGAZINE SOLO LINUX

Nº
29

Tu revista, la revista de tod@s

JUNIO 2021



Hacer copias de seguridad
con **rsnapshot**

OpenSUSE 15.3
Listo para su descarga

Cómo ver archivos **VHD**
o **VHDx** en **Ubuntu**

Instalar **Astra Linux**
La distribución rusa perfecta

Uso del comando **cal** y **ncal**
en Linux

Cambiar la **shell**
predeterminada con **chsh**

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

Dirección y maquetación

Adrián Almenar
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
Instagram: [@karyfernandez.design](https://www.instagram.com/karyfernandez.design)

Publicidad

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

CON SOLOLINUX
MULTIPLICARAS
TUS CLIENTES

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

Contacto

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

Agradecimientos

Gracias **Jose Luis** por tus palabras



La revista SOLOLINUX esta realizada con Libre Office Impress 7.1.4.2



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

Bienvenido a la Revista SOLOLINUX

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

Os presentamos el número 29 de la **Revista SoloLinux**.

Un numero mas que sumamos a nuestra colección, gracias por seguir confiando en nosotros. Nos gustaría tener alguna colaboración mas para la revista en forma de OPINIONES o con artículos

Sin mas **quiero agradecer a todos** los que hacéis posible que esta revista siga adelante.

Personalmente agradezco a Sergio todo su trabajo en la multitud de artículos que realiza a lo largo del mes para que esta revista pueda tener suficiente información mes a mes.

Gracias a TOD@S

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

Equipo SOLOLINUX

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

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

PayPal

Donar a Revistalinux

AYUDANOS A SEGUIR
CRECIENDO

PayPal

Donar a SoloLinux



designed by  freepik

SEGURIDAD

07. Hacer copias de seguridad con rsnasnapshot

Rsnapshot
Copias
de
seguridad



24. Buscar los correos de un dominio en Google

29. Buscar virus con un Live USB ClamAV Antivirus



DISTROS LINUX

09. OpenSUSE 15.3 – Listo para su descarga

12. Instalar Astra Linux – La distribución rusa perfecta



34. Migrar CentOS 8 a Rocky Linux 8.4

MANUALES

11. Como ver archivos VHD o VHDx en Ubuntu

15. Como poner Astra Linux en español – Castellano



17. Uso del comando cal y ncal en linux

19. Cambiar la shell predeterminada con chsh

21. bash: /home/user/.cargo/env: No existe el archivo



26. Iniciar el modo recovery en Ubuntu y otros linux

27. Instalar Python 3.9 en Debian 10 y derivados

31. 21 comandos linux que debes aprender

36. Instalar XFCE en Alma Linux y Rocky Linux 8.4

41. Iniciar Ubuntu en modo texto



42. Reinstalar el escritorio de Ubuntu

MANUALES

43. Comprimir archivos con ZStandard en Linux



48. 8 formas de contar el número de líneas
49. Matar procesos con kill, pkill o killall

SOFTWARE

22. Como instalar Brave en Linux



37. Instalar Ampache Music Streaming Server en Ubuntu 20.04

40. Whatsapp en linux con Whatsdesk - Junio 2021



NOTICIAS

45. 5 razones para volver a Linux



OPINIÓN DEL LECTOR

52. Opinión de Jose Luis desde Cuba.

SOLOLINUX

PayPal™ PayPal™
Donar a Revistalinux Donar a SoloLinux

Canales de Telegram:
Canal SoloLinux
Canal SoloWordpress

Chat de SoloLinux en Telegram

designed by freepik

edge²

estilo, ligereza, potencia, autonomía y libertad



1,1Kg de peso, más de 10 horas de autonomía, pantalla de 14" con tecnología WVA, procesadores i5 e i7 de 11ª generación, gráficos IrisXe, WIFI 6, Bluetooth5, Thunderbolt 4, USB-C compatible con PowerDelivery y DisplayPort, hasta 40GB de RAM y unidades SSDs PCIe4.0 de hasta 4TB y libertad para elegir tu sistema operativo, con diferentes distribuciones Linux para que elijas tu preferida,

y este verano, a partir de 777€

VANT
SOMOS LINUXEROS

www.vantpc.es

 @vantpc  vant.pc  vantpc_es  t.me/vantpc



**INSTITUTO
LINUX**



FABIÁN AMPALIO
Coordinador Gral.



GABRIEL CANEPA
Tutor



ALEJANDRO DRABENCHE
Tutor



JOAQUÍN AMPALIO
Tutor



FRANCISCO SANDALINAS
Tutor



DIEGO NOBILE
Tutor



INICIO 11 DE AGOSTO DE 2021

**INSCRIPCIÓN
ABIERTA**

CARRERA LINUX



+54 9 11 6969 9993



espania@carreralinux.com.ar



www.carreralinux.es

Hacer copias de seguridad con rsnapshot

La herramienta **rsnapshot**, tiene la capacidad de crear backups incrementales personalizados, tanto locales como remotos. Escrita en **Perl** sin dependencias de módulos, en realidad es un sencillo pero poderoso **back-end** del comando **rsync**.

Su configuración no es compleja, en unos pocos minutos lo puedes tener listo. Como es lógico, los archivos podrán ser restaurados por el **usuario root**, también por otros usuarios con permisos sin requerir la intervención del **root**. En este artículo de hoy, vemos como **instalar rsnapshot** en linux, así como su configuración básica.

Hacer copias de seguridad con rsnapshot

Procedemos a **instalar rsnapshot**, en nuestro sistema linux preferido.

Debian, Ubuntu y derivados:

```
sudo apt install rsnapshot
```

En CentOS, Fedora, Alma Linux y derivados:

```
sudo yum install rsnapshot
# 0
sudo dnf install rsnapshot
```

En Arch Linux, Manjaro y derivados:

```
sudo pacman -S rsnapshot
```

En OpenSUSE y derivados:

```
# Modifica la version si es necesario.
# En nuestro ejemplo OpenSUSE 15.2.
zypper addrepo
https://download.opensuse.org/repositories/openSUSE:Leap:15.2/standard/openSUSE:Leap:15.2.repo

zypper refresh

zypper install rsnapshot
```

```
root@Sololinux-demo:~# sudo apt install rsnapshot
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  liblchown-perl
The following NEW packages will be installed:
  liblchown-perl rsnapshot
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 123 kB of archives.
After this operation, 592 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu focal/universe amd64 liblchown-perl amd64 1.01-3build5 [8956 B]
Get:2 http://archive.ubuntu.com/ubuntu focal/universe amd64 rsnapshot all 1.4.3-2 [114 kB]
Fetched 123 kB in 0s (893 kB/s)
Selecting previously unselected package liblchown-perl.
(Reading database ... 22656 files and directories currently installed.)
Preparing to unpack .../liblchown-perl_1.01-3build5_amd64.deb ...
Unpacking liblchown-perl (1.01-3build5) ...
Selecting previously unselected package rsnapshot.
Preparing to unpack .../rsnapshot_1.4.3-2_all.deb ...
Unpacking rsnapshot (1.4.3-2) ...
Setting up liblchown-perl (1.01-3build5) ...
Setting up rsnapshot (1.4.3-2) ...
Processing triggers for man-db (2.9.1-1) ...
root@Sololinux-demo:~#
```

Una vez instalada la herramienta, vamos a configurarla. Para ello abrimos y editamos su archivo **«.conf»**.

```
sudo nano /etc/rsnapshot.conf
```

Rsnapshot

Copias de seguridad



www.sololinux.es

Lo primero que debes configurar, es la ubicación donde se guardan los backups. Busca la siguiente línea...

```
snapshot_root /var/cache/rsnapshot/
```

En nuestro ejemplo lo modificamos por **/backup**.

```
snapshot_root /backup/
```

Un poco más abajo, veremos los valores de rotación de backups.

```
#####
#   BACKUP LEVELS / INTERVALS   #
# Must be unique and in ascending order #
# e.g. alpha, beta, gamma, etc.   #
#####

retain alpha 6
retain beta 7
retain gamma 4
#retain delta 3
```

www.sololinux.es

Los explicamos...

- **retain alpha:** Diarias.
- **retain beta:** Semanales.
- **retain gamma:** Mensuales.
- **retain delta:** Anuales (por defecto deshabilitado).

Entonces, según lo visto anteriormente se guardarán 6 copias diarias, 7 semanales y 4 mensuales. En nuestro caso lo modificamos con otros valores más lógicos.

```
#####
#   BACKUP LEVELS / INTERVALS   #
# Must be unique and in ascending order #
# e.g. alpha, beta, gamma, etc.   #
#####

retain alpha 4
retain beta 3
retain gamma 1
#retain delta 3
```

Ahora debes descomentar dos comandos más.

```
cmd_ssh /usr/bin/ssh
# y
cmd_du /usr/bin/du
```

```
#####
# EXTERNAL PROGRAM DEPENDENCIES #
#####
# LINUX USERS: Be sure to uncomment "cmd_cp". This gives
you extra features.
# EVERYONE ELSE: Leave "cmd_cp" commented out for
compatibility.
#
# See the README file or the man page for more details.
#
cmd_cp          /bin/cp

# uncomment this to use the rm program instead of the built-
in perl routine.
#
cmd_rm          /bin/rm

# rsync must be enabled for anything to work. This is the
only command that
# must be enabled.
#
cmd_rsync       /usr/bin/rsync

# Uncomment this to enable remote ssh backups over rsync.
#
cmd_ssh         /usr/bin/ssh <-------- Descomentar

# Comment this out to disable syslog support.
#
cmd_logger      /usr/bin/logger

# Uncomment this to specify the path to "du" for disk usage
checks.
# If you have an older version of "du", you may also want to
check the
# "du_args" parameter below.
#
cmd_du          /usr/bin/du <-------- Descomentar

# Uncomment this to specify the path to rsnapshot-diff.
#
#cmd_rsnapshot_diff /usr/bin/rsnapshot-diff
# Specify the path to a script (and any optional arguments)
to run right
# before rsnapshot syncs files
#
#cmd_preexec      /path/to/preexec/script
```

Guarda el archivo y cierra el editor.

Verificamos que la configuración es correcta.

```
rsnapshot configtest
```

En nuestro ejemplo se produce un error, que nosotros mismos hemos provocado para que tengas clara la solución.

```
root@SoloLinux-demo:~# rsnapshot configtest
rsnapshot encountered an error! The program was invoked with these options:
/usr/bin/rsnapshot configtest
-----
ERROR: /etc/rsnapshot.conf on line 24:
ERROR: snapshot_root /backup - missing tabs to separate words - change \
spaces to tabs.
ERROR: /etc/rsnapshot.conf on line 227:
ERROR: backup /home/ localhost/ - snapshot_root needs to be defined before \
backup points
ERROR: /etc/rsnapshot.conf on line 228:
ERROR: backup /etc/ localhost/ - snapshot_root needs to be defined before \
backup points
ERROR: /etc/rsnapshot.conf on line 229:
ERROR: backup /usr/local/ localhost/ - snapshot_root needs to be defined \
before backup points
ERROR:
ERROR: Errors were found in /etc/rsnapshot.conf,
ERROR: rsnapshot can not continue. If you think an entry looks right, make
ERROR: sure you don't have spaces where only tabs should be.
```

Si observas el primer error (es el que produce el resto), nos dice que hemos separado «**snapshot_root** /**backup**» con la tecla espaciadora (espacio).

La herramienta **rsnapshot**, no tiene la capacidad de reproducir los espacios, debes usar la «**tecla Tab**». Entonces... borras la línea entera y escribes «**snapshot_root**», ahora pulsas la tecla tabuladora y escribes la ruta, «**/backup**».

```
rsnapshot configtest
```

Perfecto, ahora todo es válido.

```
root@SoloLinux-demo:~# rsnapshot configtest
Syntax OK
root@SoloLinux-demo:~#
```

También puedes hacer otra prueba, si quieres saber como se ejecutara la herramienta (en nuestro ejemplo, usamos la copia diaria).

```
rsnapshot -t alpha
```

```
root@SoloLinux-demo:~# rsnapshot -t alpha
echo 31420 > /var/run/rsnapshot.pid
mkdir -m 0700 -p /backup/
mkdir -m 0755 -p /backup/alpha.0/
/usr/bin/rsync -a --delete --numeric-ids --relative --
delete-excluded \
/home/ /backup/alpha.0/localhost/
mkdir -m 0755 -p /backup/alpha.0/
/usr/bin/rsync -a --delete --numeric-ids --relative --
delete-excluded /etc/ \
/backup/alpha.0/localhost/
mkdir -m 0755 -p /backup/alpha.0/
/usr/bin/rsync -a --delete --numeric-ids --relative --
delete-excluded \
/usr/local/ /backup/alpha.0/localhost/
touch /backup/alpha.0/
root@SoloLinux-demo:~#
```

Como último paso, solo necesitas crear las **tareas cron**. Debes tener presente, la nomenclatura asignada por la herramienta a los grupos de copias.

- **retain alpha:** Diarias.
- **retain beta:** Semanales.
- **retain gamma:** Mensuales.

Por ejemplo, si quieres ejecutar la copia de seguridad todos los días a las 00:00 horas, hora de servidor.

```
@ 0 * * * /usr/bin/rsnapshot alpha
```

Es así de simple.

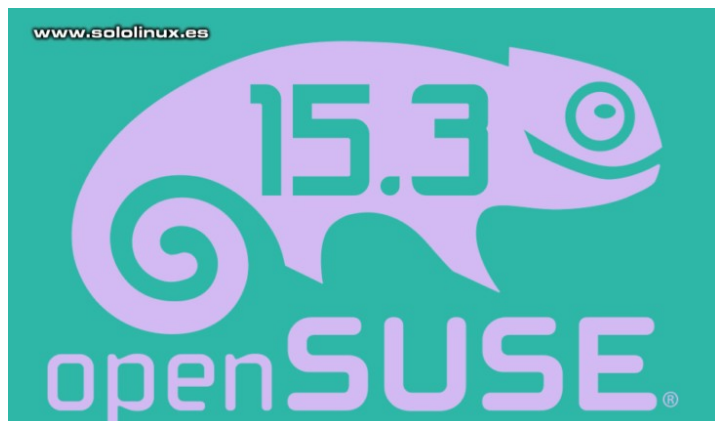
Nota final:

Las denominaciones «**alpha**, **beta** y **gamma**», pueden ser modificadas desde el archivo de configuración; por ejemplo... «**día**, **semana** y **mes**».

También te recomiendo revisar exhaustivamente el archivo de configuración, encontrarás múltiples e interesante funciones de las que no hablamos en este artículo.



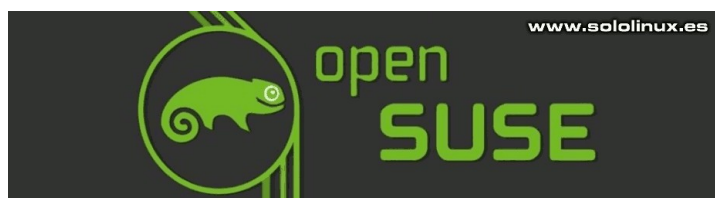
OpenSUSE 15.3 – Listo para su descarga



Después de casi un año de desarrollo, por fin ve la luz la distribución **openSUSE Leap 15.3** que como muchas veces hemos comentado... es una de las mejores **distribuciones linux** en la actualidad, además, sin duda alguna. Esta versión se basa en el paquete base de **SUSE Linux Enterprise**, pero también incluye aplicaciones personalizadas del repositorio de **openSUSE Tumbleweed**.

Una característica clave de esta nueva versión de **openSUSE Leap**, es el uso de los mismos paquetes binarios de **SUSE Linux Enterprise 15 SP 3**, en vez de hacer una reconstrucción de los **paquetes de SUSE** como se practicaba en versiones anteriores. Al utilizar los mismos paquetes **binarios de SUSE**, la migración de una distribución a otra se convertirá en una tarea mucho más sencilla.

Al compartir binarios, se ahorran recursos en la creación de nuevos paquetes, de actualizaciones y, también se unifican diferencias importantes entre las dos versiones. **En realidad y bajo mi punto de vista, esto parece un giro en la política de versiones por parte de SUSE Linux, con el único fin de apoderarse con parte del pastel que CentOS abandona, gracias al movimiento de RedHat.** De momento los usuarios salen beneficiados, eso es evidente, veremos lo que nos depara el futuro.



OpenSUSE 15.3 – Listo para su descarga

Vemos las principales novedades de la nueva versión:

- Se han actualizado la mayoría de componentes de la distribución, como por ejemplo **systemd** y el administrador de **paquetes DNF**.

- Los entornos de escritorio también se actualizan. Xfce 4.16, LXQt 0.16 y Cinnamon 4.6. Otros por estabilidad, mantienen versiones anteriores, KDE Plasma 5.18, GNOME 3.34, MATE 1.24, etc.
- Ahora se ofrecen de forma oficial nuevos paquetes, como TensorFlow Lite 2020.08.23, PyTorch 1.4.0, ONNX 1.6.0, Grafana 7.3.1, etc.
- También se actualizan las herramientas específicas de contenedores: Podman 2.1.1-4.28.1, CRI-O 1.17.3, containerd 1.3.9-5.29.3, kubeadm 1.18.4, etc.
- Para los desarrolladores, se ofrece de manera predeterminada los lenguajes Go 1.15, Perl 5.26.1, PHP 7.4.6, Python 3.6.12, Ruby 2.5, Rust 1.43.1, entre otros.
- La biblioteca **Berkeley DB** se ha eliminado de los paquetes apr-util, cyrus-sasl, iproute2, perl, php7, postfix y rpm, por las limitaciones impuestas por Oracle en la licencia. La rama Berkeley DB 6 se ha migrado a AGPLv3, cuyos requisitos también se aplican a las herramientas que utilizan BerkeleyDB en forma de biblioteca. Por ejemplo, RPM se envía bajo GPLv2 y AGPL que es incompatible con GPLv2.
- Se recupera la antigua ventana desde la cual puedes seleccionar el tipo de instalación o, de entorno de escritorio durante la instalación de openSUSE 15.3. En anteriores versiones, la selección por defecto se limitaba a Gnome y KDE.
- Se agrega soporte para sistemas IBM Z y LinuxONE (s390x).



Descargar openSUSE 15.3

Puedes descargar la nueva versión desde su página oficial, selecciona tu arquitectura, o si lo prefieres... una live específica con tu entorno de escritorio favorito.

- Descargar openSUSE 15.3**



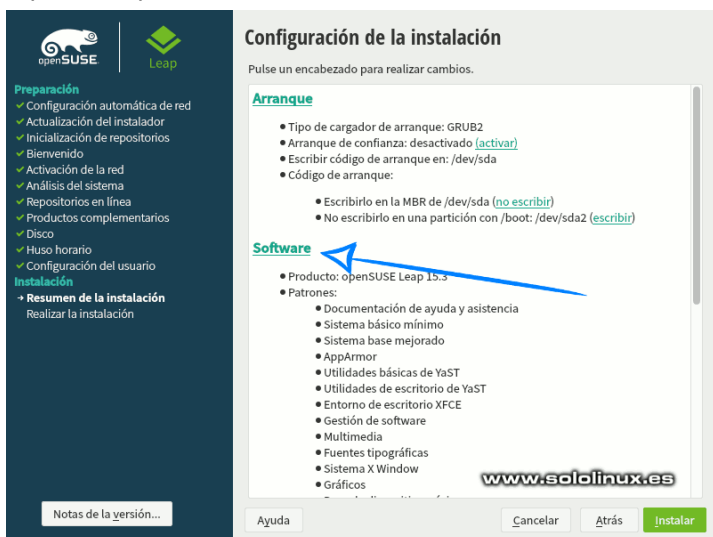
Imágenes de ejemplo

Como es habitual en **sololinux.es**, antes de escribir un artículo sobre una **distribución Linux**, la instalamos y analizamos. Con **openSUSE** no podía ser menos y más con la debilidad que siento por ella, por tanto vemos unas capturas de su instalación, así como del resultado final. Comenzamos.

Al iniciar pulsa la tecla F2 y selecciona tu idioma preferido. En el menú, marcas «Instalación» y pulsa enter.

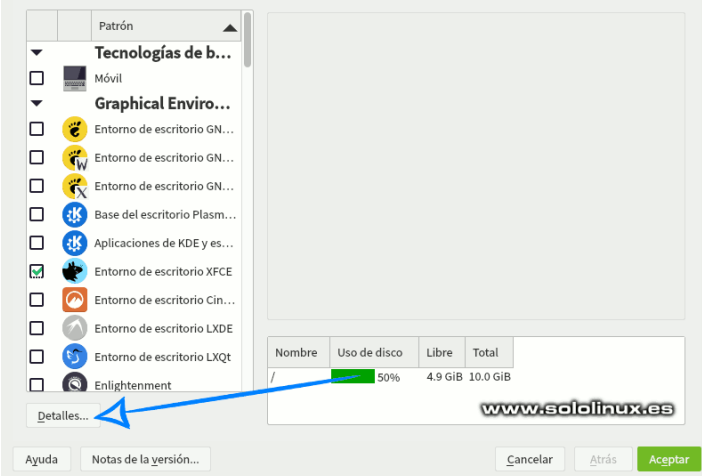


Después de configurar varios pasos, como particiones, repositorios, etc. Aparece una pantalla como la de la siguiente imagen, pulsa donde te indica la flecha (Software).



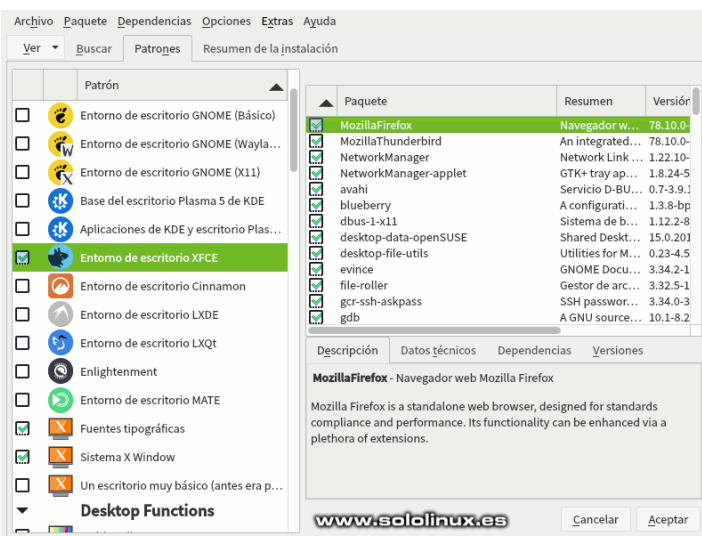
Nos aparece una pantalla (desconocida por los usuarios más noveles), donde puedes añadir o quitar conjuntos de paquetes. Esto no siempre es suficiente si quieres una buena personalización, pulsa en detalles (donde indica la imagen).

Selección de software y tareas del sistema

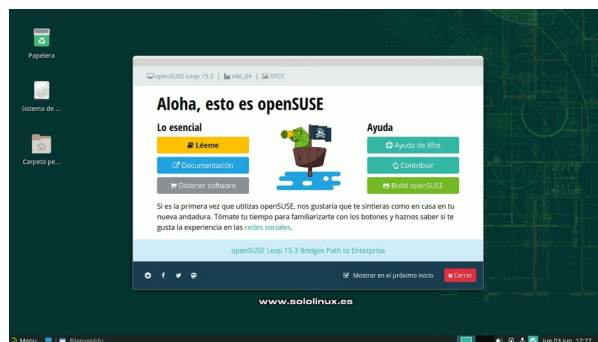


¿Sorprendido?, observa la nueva pantalla y revisa todo lo que te ofrece con mucha atención. Puedes agregar o quitar cualquier aplicación que se encuentre en sus repositorios, así como bibliotecas, paquetes, drivers, etc. La lista es infinita.

Una vez termines tu personalización, pulsas aceptar para continuar con la instalación. No te preocupes por las dependencias, el instalador las detecta y te solicita permiso para realizar cambios mediante correcciones varias.



El resultado final es simplemente espectacular. Tenemos un **Linux** estable como una roca, pero a la vez será válido para cualquier tipo de usuario.



Como ver archivos VHD o VHDx en Ubuntu



VHD (Virtual Hard Disk) o su versión más moderna **VHDx**, son unos formatos especiales de archivo, que representan unidades de discos duros virtuales. Su contenido es importante, pues en ellos podemos encontrar las particiones del disco, su sistema de archivos, así como todo el contenido (directorios, archivos, etc).

Utilizado en las virtualizaciones de **Hyper-V** de **Microsoft**, en el mini-artículo de hoy veremos como instalar el **el comando guestmount**, que es el que nos permitirá montar los archivos remotos de **Hyper-V** y verlos en nuestro linux.



Como ver archivos VHD o VHDx en Ubuntu

Lo primero que debemos hacer, es instalar todas las herramientas y librerías necesarias para lograr nuestro objetivo.

En Debian, Ubuntu y derivados:

```
sudo apt install libguestfs-tools
```

En Fedora, CentOS, Alma Linux y derivados:

```
sudo yum install libguestfs-tools
# o
sudo dnf install libguestfs-tools
```

```
root@SoloLinux-demo:~# sudo apt install libguestfs-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
www.sololinux.es
The following additional packages will be installed:
adwaita-icon-theme at-spi2-core attr Augeas-lenses binutils binutils-common
binutils-x86-64-linux-gnu btrfs-progs cpu-checker cryptsetup-bin db-util
db5.3-util dbus-user-session dconf-gsettings-backend dconf-service
```

Antes de continuar te recomiendo obtener **permisos de root**.

```
sudo su
```

Ahora puedes listar las particiones con el siguiente comando.

```
virt-list-partitions nombre.vhdx
```

Vemos un ejemplo...

```
root@SoloLinux-demo:/mnt# virt-list-partitions sololinux-storage.vhdx
```

La salida...

```
root@SoloLinux-demo:/mnt# virt-list-partitions sololinux-storage.vhdx
/dev/sda1
/dev/sda2
/dev/sda3
/dev/sda4
root@SoloLinux-demo:/mnt#
```

Bien, ya conocemos las particiones del dispositivo. Antes de montar creamos una nueva carpeta sin contenido.

```
mkdir /vhdx/
```

Solo nos falta montar la partición deseada. Vemos un ejemplo.

```
guestmount -a sololinux-storage.vhdx -m /dev/sda3 --ro /vhdx/
```

Listo, ya puedes acceder a la partición y realizar cualquier tarea.

AYUDANOS A SEGUIR
CRECIENDO

PayPal™ PayPal™

Donar a Revistalinux

Donar a SoloLinux



Canales de Telegram:
Canal SoloLinux
Canal SoloWordpress

Chat de SoloLinux en Telegram



designed by freepik

Instalar Astra Linux – La distribución rusa perfecta



Astra Linux es un sistema operativo ruso basado en Debian GNU / Linux, diseñado para **máquinas de escritorio y servidores**, que pone una especial énfasis en la seguridad de la información almacenada y procesada por los usuarios.

Usada por la mayoría de organismos gubernamentales rusos, incluyendo las áreas de defensa y los servicios de inteligencia. También ha sido adoptado por los ministerios de salud, ciencia, educación, finanzas e industria, además de poderosas industrias del ámbito privado.

Todos los nombrados anteriormente, usan la versión **Astra Linux Special Edition** que brinda una protección extrema sobre la información confidencial y, secretos de estado al nivel de «importancia especial». Para otras situaciones, como pueden ser comercios al por menor, usuarios domésticos, estudiantes, etc., existe la versión Astra Linux Common Edition, que también cuenta con excelentes medidas de protección aunque no tan duras como la versión Special.

Este linux toma las bases de **Debian**, usa el escritorio **Fly desktop** que es un **fork de KDE** pero bajo mi humilde opinión... mucho más logrado, además de ligero. Viene con las clásicas aplicaciones que traen otras distribuciones, además de otras muy interesantes desarrolladas por ellos mismos. Debo aclarar, que estas aplicaciones vienen en su lenguaje nativo (ruso) y no ofrecen traducción, así que aprendes ruso o simplemente me escribes tu consulta desde la **zona de contacto**, jaja.

En este artículo vemos como instalar este sistema operativo, que por cierto... como es lógico soporta los procesadores **Elbrús** de fabricación rusa. Puedes **descargar Astra Linux** desde su **página oficial**.

Instalar Astra Linux – La distribución rusa

Al iniciar la **ISO de Astra Linux**, por defecto nos aparece en ruso. Abajo a la izquierda, pulsa las **flechas del teclado** para seleccionar el idioma ingles. Si no aparece la pestaña de selección **pulsa F1**.



A partir de este punto, la instalación es muy similar a Debian, seleccionamos nuestro idioma y su localización.



Después de aceptar la licencia, selecciona tu teclado. Comienza la carga de los archivos de instalación. Nos solicita el nombre de la máquina, usuario, password, etc.



www.sololinux.es

Por defecto toma la **zona horaria** de tu conexión de internet, si no estas de acuerdo pulsa en retroceder y la seleccionas manualmente.



Ahora debes seleccionar el disco de destino, el tipo de partición/es, y aceptar la escritura de los cambios en el disco.



Una vez preparadas las particiones del disco, en la siguiente pantalla puedes seleccionar los paquetes de herramientas a instalar. Presta especial atención a la última opción, **Hardened Kernel**. Un kernel Linux hardened, está enfocado a mejorar la seguridad aplicando parches que mitigan la explotación del kernel y, los espacios del usuario. También se habilitan otras características de seguridad, como namespaces, audit y **SELinux**.



Una vez termine la instalación de los paquetes de software seleccionados, veremos una nueva pantalla. Te recomiendo que prestes atención y revises exhaustivamente todas las opciones. Por ejemplo, de manera predeterminada Astra instala dos kernels, el 4.15 y el 5.4. Siempre puedes seleccionar el que te interese en el **Grub**, pero si seleccionas la opción correspondiente iniciara automáticamente con el 5.4.



Una vez se apliquen los cambios seleccionados, Astra Linux te preguntará si quieres instalar el **cargador de arranque Grub** en el disco, o en otro dispositivo; como norma general debes seleccionar en el disco. La instalación ha terminado, pulsa en continuar para iniciar Astra Linux por primera vez.



En nuestro caso, no seleccionamos el Kernel 5.4 por defecto. El Grub nos permite utilizar uno u otro.



Vemos la pantalla de inicio de sesión. Te recomiendo que investigues los menús laterales de «**Session Type**» y «**Menu**». Te sorprenderás.



Observa la pantalla principal que viene por defecto, realmente impresionante. Limpia, clara y todo bien organizado, como tiene que ser una **distribución linux** con un destino final de uso tan marcado.



Astra es atractivo, pero siguiendo su propia línea de colores suaves.



Control center de Astra Linux.



Nota: No te molestes en buscar el entorno de escritorio **Fly desktop** o sus herramientas personalizadas para instalar en otra distribución derivada de Debian. No existen repositorios públicos sobre «**Fly**», vienen incluidos con la distro.

AYUDANOS A SEGUIR CRECIENDO

PayPal™ PayPal™

Donar a Revistalinux

Donar a SoloLinux

Canales de Telegram:
Canal SoloLinux
Canal SoloWordpress

Chat de SoloLinux en Telegram



designed by freepik

Como poner Astra Linux en español – Castellano



Este artículo de hoy, ya estaba preparado a raíz del lanzamiento del anterior denominado «**Instalar Astra Linux**». La verdad es que si tardé más de dos días en lanzarlo, es gracias a una serie de personajes que se han dedicado a lanzar improperios sobre mi persona, simplemente por enseñar a la comunidad hispana una tremenda distro Linux.

Esto me hizo replantear varias cosas, la primera es si vale la pena, la segunda es que me gustaría conocerlos en persona y..., en realidad no vale la pena perder el tiempo con ellos (saluda a tu comandante campeón). No concibo la idea de que alguien categorice, que para ser un buen linux, debe estar absolutamente todo en nuestro idioma nativo incluyendo la instalación. Bueno, sí que lo entiendo, se llama ignorancia culturalmente hablando.

En la actualidad, existen muchas **distribuciones linux** especiales sobre una temática en particular, por ejemplo las orientadas al pentest, por tanto no entiendo nada. Usas **Kali Linux** y las herramientas específicas en castellano son mínimas, por no decir ninguna. ¿Te parece mal?. Creo que deberías promover una denuncia internacional, contra los desarrolladores de Unix y, para no dejarnos a nadie... también contra **Torbals** por escribir su código en un lenguaje (ingles), no apto para tus conocimientos. Bueno, ya os dediqué demasiado tiempo que para mí es algo que tiene valor.

Astra Linux no es una distribución específica para detectar vulnerabilidades remotas, todo lo contrario. Sus funciones y desarrollos, te protegen de intrusos y en esto sí que es la mejor, le duela a quien le duela. Puedes pensar que ya existe Tails, yo te afirmo que no. **Tails** es para navegar y poco más. **Astra** es un linux completo, para un uso cotidiano.

Como poner Astra Linux en español – Castellano

Astra Linux es una distribución rusa, creada específicamente para proteger sus secretos de estado de miradas indiscretas y, te aseguro que en estos temas son buenos, muy buenos. Yo vivo con ellos, trabajo con

ellos y desarrollo con ellos, no es lógico hablar sin conocer.

Poner Astra Linux en español por completo es tarea imposible, a no ser que edites su código. **Astra** se distribuye en Ruso e Ingles, ahora me dices que porque en ingles... vale te lo explico. El ruso al igual que alguna de sus exreplicas, usan caracteres cirílicos en su escritura. Para uso interno esta bien, pero claro... a la hora de comunicarse con otro país, programar, etc., esto supone un problema.

La solución es sencilla... todos, absolutamente todos los sistemas operativos comercializados o distribuidos en estos países (incluyendo Windows), instalan por defecto el idioma nativo y dos teclados seleccionables, el inglés para poder escribir con latinas y el ruso, ucraniano, bielorruso, etc. Esto se amplía a los teclados como dispositivo físico.

Pongo una foto de mi teclado, así veréis que tiene varios caracteres por tecla, el inglés y el ruso.



Astra Linux se basa en Debian, por tanto existen paquetes para poder traducir la distribución en sí, además de las aplicaciones y software estándar que tienen paquetes en español. Sin embargo, **Astra linux** viene con el entorno de **escritorio Fly**, que es un excelente fork de **KDE** al cual solo le aplicaron el ruso y el inglés. No te preocupes, el inglés de **Fly** es muy **básico**, digamos al nivel de un niño de 6 u 8 años. Verás que fácil.

Lo primero que haremos será actualizar Astra, después procedemos a instalar y configurar el español en nuestro sistema.

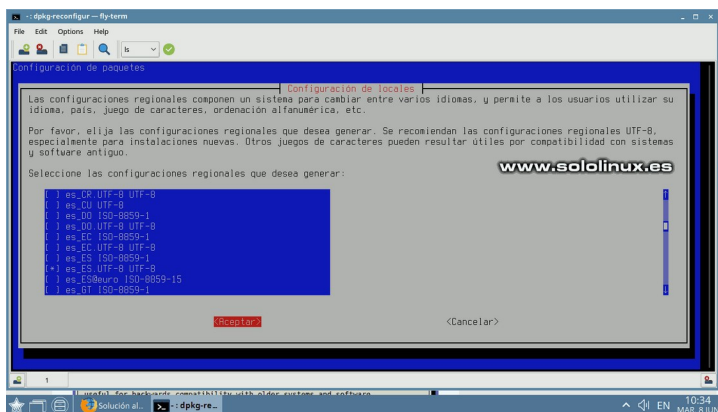
```
sudo su
apt full-upgrade
```

Ahora las locales.

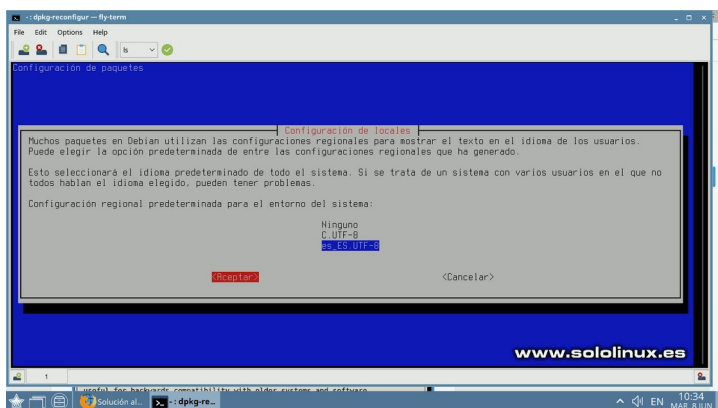
```
dpkg-reconfigure locales
```

En la pantalla, seleccionamos el español. Recomiendo el **es_ES**, ya que el teclado no ofrece otras opciones. Utiliza las flechas del teclado para identificar el idioma, lo seleccionas con la barra de espacio y con la tecla tabuladora (TAB) marcas en aceptar. Pulsa «Enter».

www.sololinux.es



En la nueva pantalla, te mueves hasta localizar tu lenguaje preferido, en nuestro caso «**es_ES.UTF-8**». Recuerda que para seleccionar debes utilizar la barra espaciadora.



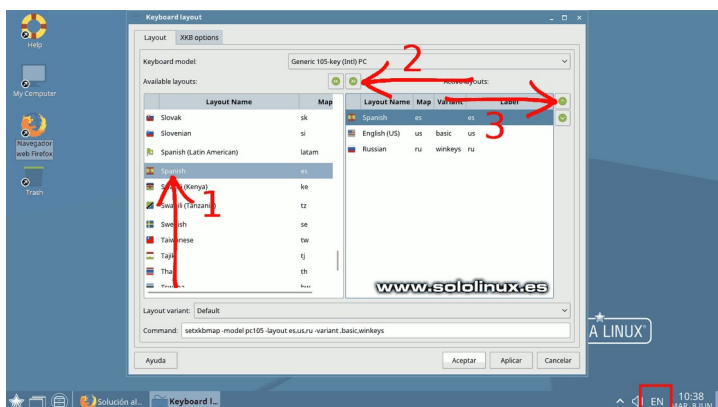
Una vez termine la instalación, reconfiguramos de nuevo y, reiniciamos el sistema.

```
sudo locale-gen es_ES.UTF-8
sudo reboot
```

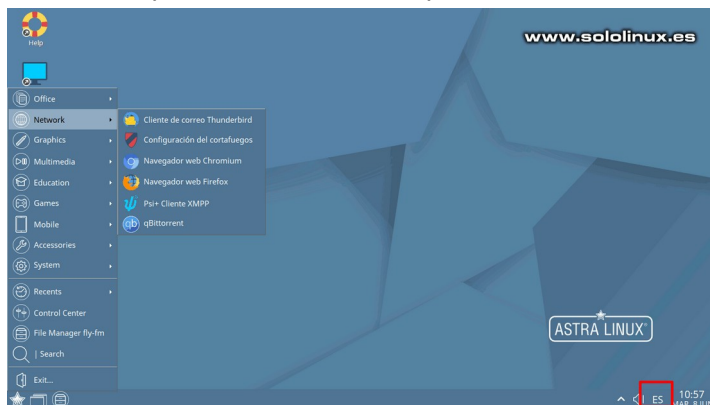
Al reiniciar el sistema, ya lo tenemos en español de España (Castellano). Ahora vamos a por el teclado. Pulsamos en inicio (estrella) y en «**Control Center**», ahora en «**keyboard layout**».

- 1) Seleccionas el idioma
- 2) Pulsas en la flecha para añadir el idioma
- 3) Lo pones el primero como default.

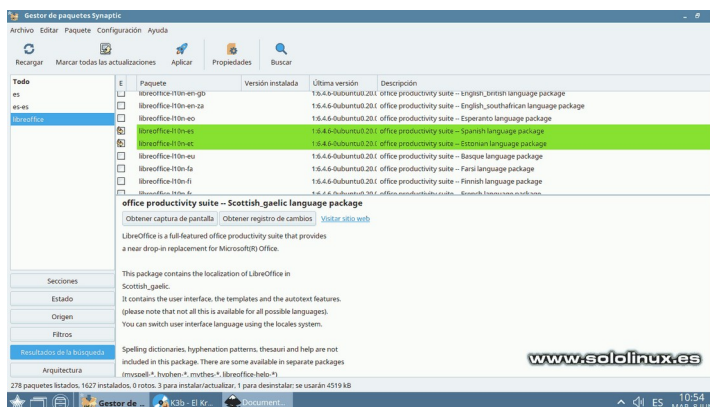
Aplicas y reinicias el sistema.



Al reiniciar el sistema, las aplicaciones y herramientas aparecen en castellano. Como ya comente anteriormente, el menú del **FLY Desktop** y herramientas propias permanecerán en ingles, pero vamos... es fácilmente compresible para cualquier usuario que sepa leer (solo eso, leer). Observa que en la parte inferior a la derecha, aparece el teclado en español.



Es posible que alguna aplicación permanezca en ingles. Tranquilo, no te ahogues en un vaso de agua. Esto puede suceder en software como **LibreOffice** que por su manera de distribuir los paquetes obligan a instalarlo aparte. La solución es tan simple como abrir el **Synaptic** e instalar el idioma que necesitas.



Nota del autor: No critiques lo que no conoces, por lo menos lee antes de comentar. Si consideras que no aprendes nada en **SoloLinux** te doy la enhorabuena, la solución es simple al igual que la oferta amplia (no molestes). **SoloLinux** es un proyecto sin ánimo de lucro, que incentiva y promueve el uso de Linux. Esto no da de comer, es un simple hobby que me cuesta dinero a final de mes. Mi negocio es otro y no depende de ti, eso te lo aseguro. Insisto, por favor no molestes.

Uso del comando cal y ncal en Linux

El comando `cal` y su símil `ncal`, son dos herramientas incluidas por defecto en la mayoría de **distribuciones Linux**, con una única misión, mostrar un calendario en la **terminal de linux**.

Ambos comandos permiten varias opciones, con las cuales puedes personalizar o determinar un tipo de impresión específica. En este artículo vemos algunos ejemplos prácticos de uso, pero antes observa la diferencia de formato de un comando a otro. Además del formato, algunos argumentos pueden variar por eso es importante que revises la ayuda del comando.

```
sergio@sololinux:~$ cal
    Junio 2021
do lu ma mi ju vi sa
1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30

sergio@sololinux:~$ ncal
    Junio 2021
lu 7 14 21 28
ma 1 8 15 22 29
mi 2 9 16 23 30
ju 3 10 17 24
vi 4 11 18 25
sa 5 12 19 26
do 6 13 20 27
```

Uso del comando cal y ncal en Linux

Su sintaxis básica es muy simple.

```
cal [mes] [año]
```

Con «**help**» puedes ampliar la información.

```
sergio@sololinux:~$ cal -help
Usage: cal [general options] [-jy] [[month] year]
       cal [general options] [-j] [-m month] [year]
       ncal -C [general options] [-jy] [[month] year]
       ncal -C [general options] [-j] [-m month] [year]
       ncal [general options] [-bhJjpwySM] [-H yyyy-mm-dd] [-s
country_code] [[month] year]
       ncal [general options] [-bhJeoSM] [year]
General options: [-31] [-A months] [-B months] [-d yyyy-mm]
```

Si ejecutamos el **comando cal o ncal** sin opciones ni parámetros, nos imprime el calendario de la fecha actual.

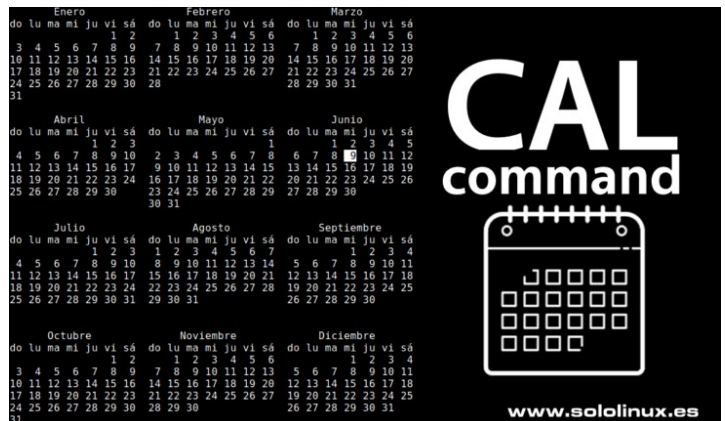
Ejemplo de `cal`. Al imprimir en pantalla, se marca el día de hoy.

```
sergio@sololinux:~$ cal
    Junio 2021
do lu ma mi ju vi sa
1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30
```

Para ver el calendario de un mes y año en particular, indicamos el número de mes y el año.

```
# Ejemplo de Marzo del 2021
cal 3 2021
```

```
sergio@sololinux:~$ cal 3 2021
    Marzo 2021
do lu ma mi ju vi sa
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```



Con la opción «**-m**», puedes escribir en nombre del mes. En nuestro caso marzo sin especificar año, por tanto se imprimirá del año en curso. Debes escribir el nombre en el idioma de tu sistema, en nuestro caso español de España.

```
cal -m marzo
```

```
sergio@sololinux:~$ cal -m marzo
    Marzo 2021
do lu ma mi ju vi sa
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

Ahora agregamos el año, por ejemplo marzo del 2020.

```
cal -m marzo 2020
```

```
sergio@sololinux:~$ cal -m marzo 2020
    Marzo 2020
do lu ma mi ju vi sa
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
```

Ahora vemos el **calendario anual** completo, con «**-y**».

```
cal -y
# o
ncal -y
```



Si observaste detenidamente la salida de «help», te habrás dado cuenta de que existen algunas diferencias entre cal y ncal. Ncal admite algunos argumentos más, por ejemplo la impresión del número de semana.

```
ncal -w 2021
```

```
sergio@sololinux:~$ ncal -w 2021
```

```

2021
Enero      Febrero    Marzo      Abril
lu  4 11 18 25 1 8 15 22 1 8 15 22 29 5 12 19 26
ma  5 12 19 26 2 9 16 23 2 9 16 23 30 6 13 20 27
mi  6 13 20 27 3 10 17 24 3 10 17 24 31 7 14 21 28
ju  7 14 21 28 4 11 18 25 4 11 18 25 1 8 15 22 29
vi  1 8 15 22 29 5 12 19 26 5 12 19 26 2 9 16 23 30
sá  2 9 16 23 30 6 13 20 27 6 13 20 27 3 10 17 24
do  3 10 17 24 31 7 14 21 28 7 14 21 28 4 11 18 25
    53 1 2 3 4 5 6 7 8 9 10 11 12 13 13 14 15 16 17
    Mayo      Junio      Julio      Agosto
lu  3 10 17 24 31 7 14 21 28 5 12 19 26 2 9 16 23 30
ma  4 11 18 25 1 8 15 22 29 6 13 20 27 3 10 17 24 31
mi  5 12 19 26 2 9 16 23 30 7 14 21 28 4 11 18 25
ju  6 13 20 27 3 10 17 24 1 8 15 22 29 5 12 19 26
vi  7 14 21 28 4 11 18 25 2 9 16 23 30 6 13 20 27
sá  1 8 15 22 29 5 12 19 26 3 10 17 24 31 7 14 21 28
do  2 9 16 23 30 6 13 20 27 4 11 18 25 1 8 15 22 29
    17 18 19 20 21 22 23 24 25 26 26 27 28 29 30 30 31 32 33 34 35
    Septiembre Octubre      Noviembre Diciembre
lu  6 13 20 27 4 11 18 25 1 8 15 22 29 6 13 20 27
ma  7 14 21 28 5 12 19 26 2 9 16 23 30 7 14 21 28
mi  1 8 15 22 29 6 13 20 27 3 10 17 24 1 8 15 22 29
ju  2 9 16 23 30 7 14 21 28 4 11 18 25 2 9 16 23 30
vi  3 10 17 24 1 8 15 22 29 5 12 19 26 3 10 17 24 31
sá  4 11 18 25 2 9 16 23 30 6 13 20 27 4 11 18 25
do  5 12 19 26 3 10 17 24 31 7 14 21 28 5 12 19 26
    35 36 37 38 39 39 40 41 42 43 44 45 46 47 48 48 49 50 51 52

```

Al añadir como opción tan solo un número, por ejemplo el 3, en la pantalla veremos una salida curiosa. El mes actual del año en curso, el mes anterior al actual y, el de la misma forma... también el mes siguiente.

```
cal -3
# 0
ncal -3
```

```
sergio@sololinux:~$ cal -3
```

```

Mayo 2021      Junio 2021      Julio 2021
do lu ma mi ju vi sá do lu ma mi ju vi sá do lu ma mi ju vi sá
2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 9 10 11 12 4 5 6 7 8 9 10
9 10 11 12 13 14 15 13 14 15 16 17 18 19 11 12 13 14 15 16 17
16 17 18 19 20 21 22 20 21 22 23 24 25 26 18 19 20 21 22 23 24
23 24 25 26 27 28 29 27 28 29 30 25 26 27 28 29 30 31
30 31

```

```
sergio@sololinux:~$ ncal -3
```

```

Mayo 2021      Junio 2021      Julio 2021
lu  3 10 17 24 31 7 14 21 28 5 12 19 26
ma  4 11 18 25 1 8 15 22 29 6 13 20 27
mi  5 12 19 26 2 9 16 23 30 7 14 21 28
ju  6 13 20 27 3 10 17 24 1 8 15 22 29
vi  7 14 21 28 4 11 18 25 2 9 16 23 30
sá  1 8 15 22 29 5 12 19 26 3 10 17 24 31
do  2 9 16 23 30 6 13 20 27 4 11 18 25

```

Estos sencillos comandos, nos puedes resultar muy útiles en situaciones específicas. Si quieres aprender más sobre ellos puedes consultar su manual integrado en consola.

```
man cal
# 0
man ncal
```

AYUDANOS A SEGUIR CRECIENDO

PayPal™ PayPal™

Donar a Revistalinux

Donar a SoloLinux



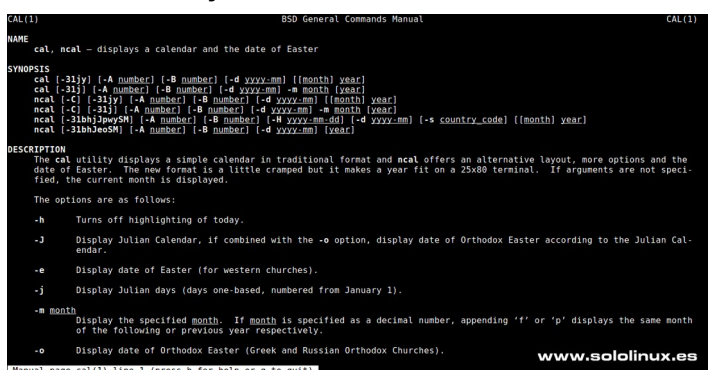
Canales de Telegram:
Canal SoloLinux
Canal SoloWordpress

Chat de SoloLinux en Telegram



designed by freepik

Manual de cal y ncal:



Cambiar la shell predeterminada con chsh

La **shell de linux**, es la capa más externa de un sistema operativo a través del cual podemos interactuar con sus servicios y herramientas, usando el intérprete de comandos definido por el sistema. La mayoría de las distribuciones Linux actuales, hacen uso de **bash** como **shell predeterminada**. Fácil de usar, además de necesitar una curva de aprendizaje mínima.

Sin embargo, existen otras opciones que también son excepcionalmente poderosas y, con una forma de operar muy similar a bash. Las opciones más conocidas podrían ser... **ksh**, **zsh**, **csch** y **fish**. Cada una de ellas, nos aporta alguna característica única que las distingue de las demás.

En el artículo de hoy, veremos como instalar un nuevo intérprete de comandos en nuestro linux y, que sea el que se ejecuta de forma predeterminada. En nuestro caso sustuiremos bash por zsh, pues lo podemos definir como su máximo competidor dada su potencia y, su desarrollo continuo insertando nuevas funciones.



Cambiar la shell predeterminada con chsh

Como ya comentamos anteriormente, **Zsh** viene con muchas características adicionales como autocompletado, autocorrección, cd automático, expansión de ruta recursiva, además de soporte con otros complementos. Esto hace de Zsh una excelente **alternativa a bash**, por

```
cat /etc/shells
```

Verificamos similar a...

```
root@SoloLinux-demo:~# cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/dash
/bin/bash
/bin/rbash
/usr/bin/screen
```

Normalmente Zsh no suele venir instalado en las **distribuciones** más comunes, lo instalamos fácilmente con los siguientes comandos.

Instalar ZSH en Debian, Ubuntu y derivados

```
sudo apt install zsh
```

Instalar ZSH en Fedora, CentOS, Alma y derivados

```
sudo dnf install zsh
# o
sudo yum install zsh
```

Instalar ZSH en Arch Linux, Manjaro y derivados

```
sudo pacman -Syu zsh
```

Verificamos de nuevo los intérpretes instalados, vemos que ya contamos con Zsh.

```
cat /etc/shells
```

```
root@SoloLinux-demo:~# sudo apt install zsh
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  zsh-common
Suggested packages:
  zsh-doc
The following NEW packages will be installed:
  zsh zsh-common
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 4450 kB of archives.
After this operation, 18.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 zsh-common all 5.8-3ubuntu1 [3744 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal/main amd64 zsh amd64 5.8-3ubuntu1 [707 kB]
Fetched 4450 kB in 0s (13.1 MB/s)
Selecting previously unselected package zsh-common.
(Reading database ... 22773 files and directories currently installed.)
Preparing to unpack .../zsh-common_5.8-3ubuntu1_all.deb ...
Unpacking zsh-common (5.8-3ubuntu1) ...
Selecting previously unselected package zsh.
Preparing to unpack .../zsh_5.8-3ubuntu1_amd64.deb ...
Unpacking zsh (5.8-3ubuntu1) ...
Setting up zsh-common (5.8-3ubuntu1) ...
Setting up zsh (5.8-3ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
root@SoloLinux-demo:~#
```

www.sololinux.es

Salida con Zsh instalado.

```
root@SoloLinux-demo:~# cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/dash
/bin/bash
/bin/rbash
/usr/bin/screen
/bin/zsh
/usr/bin/zsh
root@SoloLinux-demo:~#
```

Ahora comprobamos que funciona correctamente. Para cambiar el intérprete de comandos de **bash a zsh**, simplemente ejecuta el siguiente comando.

```
zsh
```

Para retornar a bash...

```
exit
```

```
root@SoloLinux-demo:~#
root@SoloLinux-demo:~# zsh
SoloLinux-demo#
SoloLinux-demo#
SoloLinux-demo# exit
root@SoloLinux-demo:~#
root@SoloLinux-demo:~# www.sololinux.es
```

Bueno, ahora vamos a **cambiar la shell** de manera definitiva. Nos ayudamos del **comando chsh**, que tiene una sintaxis muy simple.

```
chsh -s [ruta]
```

¿Recuerdas los resultados obtenidos al verificar las shell instaladas?, esa es la ruta que debemos insertar. Lo vemos con ejemplos para una mejor comprensión.

Lanzamos el comando chsh.

```
chsh -s /usr/bin/zsh
```

También puedes especificar otros usuarios.

```
chsh -s /usr/bin/zsh usuario
```

Inserta la contraseña para aplicar los cambios.

```
root@SoloLinux-demo:~# chsh -s /usr/bin/zsh
Password:
```

Ahora solo necesitas reiniciar la sesión, para tener Zsh por defecto.

```
sudo reboot
```

Verificas la shell predeterminada del sistema.

```
echo $SHELL
```

Nota: Estas instrucciones son válidas para cualquier intérprete de comandos instalado en nuestro sistema Linux.

AYUDANOS A SEGUIR CRECIENDO

PayPal™ PayPal™

Donar a Revistalinux

Donar a SoloLinux



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

[Chat de SoloLinux en Telegram](#)



designed by freepik

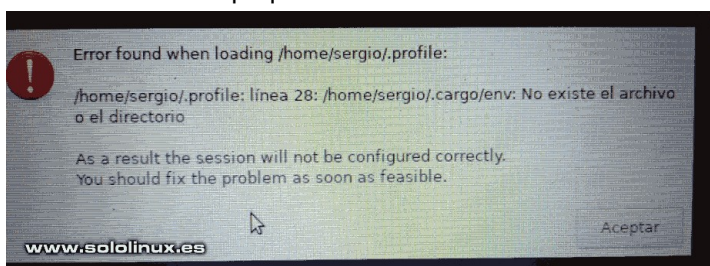
bash: /home/user/.cargo/env: No existe el archivo



No hace muchos días, alguien me lanzo una pregunta desesperada. En su afán por mantener limpia su **distribución linux**, ni corto ni perezoso comenzó a desinstalar herramientas que no utilizaba y, sus correspondientes directorios de su **/home**.

Todo parecía ir perfecto, hasta que reinicio el sistema. El sistema le avisa de que se ha producido un error, pulsa aceptar en la ventana del error e introduce su usuario y el **password**. Nada... no arranca, uff. Después de varios reinicios incluyendo la misma ventana de error, logra iniciar su linux. Da igual, parece que funciona, pero con extraños mensajes que no le permiten trabajar como es habitual.

Vamos a ver, que has borrado, que dice el mensaje de error le pregunto. Me responde... algo de «.cargo/env», que no sé lo que es, nunca instale una herramienta con ese nombre. Vaya, vaya la que has liado le respondo, tú nunca has instalado la herramienta cargo, pero si el **lenguaje de programación rust**. Cargo es su administrador de paquetes.



bash: /home/user/.cargo/env: No existe el archivo

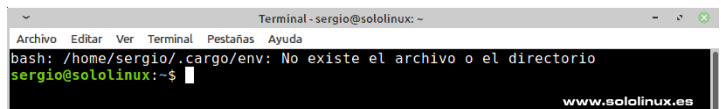
Nosotros hemos reproducido el error, para que veas lo fácil que es la solución. Nos dirigimos a la home de nuestro usuario y borramos el directorio oculto «.cargo».



Una vez borrada la carpeta, reiniciamos el sistema. Efectivamente se producen los errores que nuestro amigo nos comentó, cuatro intentos fueron necesarios hasta que logre iniciar el sistema. Todo se produce porque al instalar **Rust con Cargo**, se agrega una variable necesaria para que tu sistema funcione correctamente.

```
#!/bin/sh
# rustup shell setup
# affix colons on either side of $PATH to simplify matching
case ":{PATH}:" in
  *:"$HOME/.cargo/bin":*)
    ;;
  *)
    # Prepending path in case a system-installed rustc needs to be
    overridden
    export PATH="$HOME/.cargo/bin:$PATH"
    ;;
esac
```

El error también se reproduce al iniciar la **shell**.



Los fallos del sistema son constantes y bastantes molestos, esto pasa por tocar lo que no conoces. Por suerte la solución es bastante simple, ejecuta el siguiente comando sin **sudo**, (recuerda que el problema es con tu usuario).

```
curl https://sh.rustup.rs -sSf | sh -s -- --no-modify-path -y
```

Ignora los **warning**. Ejemplo de salida...

```
bash: /home/sergio/.cargo/env: No existe el archivo o el directorio
sergio@sololinux:~$ curl https://sh.rustup.rs -sSf | sh -s -- --no-modify-path -y
info: downloading installer
warning: it looks like you have an existing installation of Rust at:
warning: /usr/bin
warning: rustup should not be installed alongside Rust. Please
warning: uninstall your existing Rust first.
warning: Otherwise you may have confusion unless you are careful with
warning: your PATH
warning: If you are sure that you want both rustup and your already
warning: installed Rust
warning: then please reply 'y' or 'yes' or set
warning: RUSTUP_INIT_SKIP_PATH_CHECK to yes
warning: or pass '-y' to ignore all ignorable checks.
error: cannot install while Rust is installed
warning: continuing (because the -y flag is set and the error is
warning: ignorable)
info: profile set to 'default'
info: default host triple is x86_64-unknown-linux-gnu
warning: Updating existing toolchain, profile choice will be ignored
info: syncing channel updates for 'stable-x86_64-unknown-linux-gnu'
info: default toolchain set to 'stable-x86_64-unknown-linux-gnu'
info: stable-x86_64-unknown-linux-gnu unchanged - rustc 1.52.1 (9bc8c42bb
2021-05-09)
Rust is installed now. Great!
To get started you need Cargo's bin directory ($HOME/.cargo/bin) in
your PATH
environment variable.
To configure your current shell, run:
source $HOME/.cargo/env
sergio@sololinux:~$
```

El directorio borrado se ha generado de nuevo, reinicia el sistema para verificar que desapareció el error.

```
sudo reboot
```

Perfecto, el sistema ha vuelto a la normalidad.

Como instalar Brave en Linux

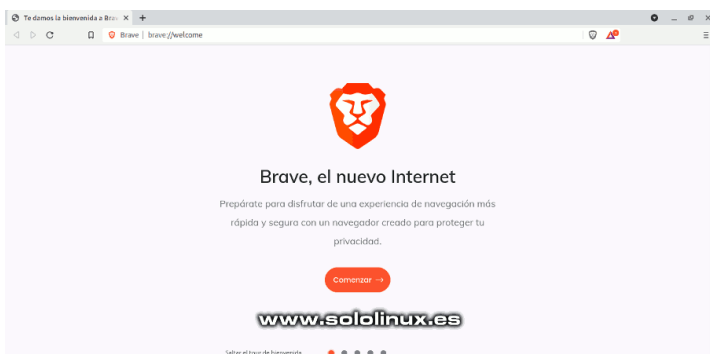


Basado en el proyecto de código abierto **Chromium**, Brave fue desarrollado por **Brian Bondy** y **Brendan Eich**, este último es conocido por ser el creador de **JavaScript** y cofundador de **Mozilla**. Como tal, hay un pedigrí centrado en la privacidad que atraerá a los usuarios conocedores de la tecnología.

Centrado en la privacidad, **Brave** elimina todos los anuncios de un sitio web y los reemplaza con sus propios anuncios. Esto le ha provocado varias denuncias, por beneficiarse de contenido ajeno y como no podía ser menos... el enfado de miles de editores de blogs diversos, como puede ser sololinux.

No es justo que se aprovechen del trabajo de otros, pues obtienen beneficios a coste cero, mientras los propietarios del contenido tienen que pagar un alto coste por mantener una infraestructura adaptada de forma óptima a su flujo de visitas. Por otro lado, Brave ofrece a los usuarios donar una especie de criptomonedas privadas a los editores del contenido visitado; pero para que esto sea efectivo, el editor o propietario del sitio web debe darse de alta en Brave como editor.

A pesar de lo dicho anteriormente, hoy creamos un artículo en el cual vemos como instalar este **navegador web**, en tu **distribución linux** preferida.



Como instalar Brave en Linux

Ahora vemos como instalar el navegador en diferentes distribuciones linux.

Brave en Fedora, CentOS, Alma y derivados

```
sudo dnf install dnf-plugins-core
sudo dnf config-manager --add-repo https://brave-browser-rpm-release.s3.brave.com/x86_64/
sudo rpm --import https://brave-browser-rpm-release.s3.brave.com/brave-core.asc
sudo dnf install brave-browser
```

Brave en OpenSUSE 15+ y derivados

```
sudo zypper install curl
sudo rpm --import https://brave-browser-rpm-release.s3.brave.com/brave-core.asc
sudo zypper addrepo https://brave-browser-rpm-release.s3.brave.com/x86_64/ brave-browser
sudo zypper install brave-browser
```

Brave en Debian, Ubuntu, Linux Mint y derivados

```
sudo apt install apt-transport-https curl
sudo curl -fsSL /usr/share/keyrings/brave-browser-archive-keyring.gpg https://brave-browser-apt-release.s3.brave.com/brave-browser-archive-keyring.gpg
echo "deb [signed-by=/usr/share/keyrings/brave-browser-archive-keyring.gpg arch=amd64] https://brave-browser-apt-release.s3.brave.com/ stable main" | sudo tee /etc/apt/sources.list.d/brave-browser-release.list
sudo apt update
sudo apt install brave-browser
```

```
$ sudo apt install brave-browser
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
brave-keyring libu2f-udev
Se instalarán los siguientes paquetes NUEVOS:
brave-browser brave-keyring libu2f-udev
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 85,0 MB de archivos.
Se utilizarán 276 MB de espacio de disco adicional después de esta operación.
[Se desea continuar?] [S/n]
Des1: http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libu2f-udev all 1.1.4-1ubuntu0.1 [3,884 B]
Des2: https://brave-browser-apt-release.s3.brave.com/stable/main amd64 brave-keyring all 1.12 [4,236 B]
Des3: https://brave-browser-apt-release.s3.brave.com/stable/main amd64 brave-browser amd64 1.25.72 [85,0 MB]
Descargados 85,0 MB en 31s (2,776 kB/s)
Seleccionando el paquete brave-keyring previamente no seleccionado.
(Leyendo la base de datos ... 378989 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../brave-keyring_1.12_all.deb ...
Desempaquetando brave-keyring (1.12) ...
Seleccionando el paquete brave-browser previamente no seleccionado.
Preparando para desempaquetar .../brave-browser_1.25.72_amd64.deb ...
Desempaquetando brave-browser (1.25.72) ...
Seleccionando el paquete libu2f-udev previamente no seleccionado.
Preparando para desempaquetar .../libu2f-udev_1.1.4-1ubuntu0.1_all.deb ...
Desempaquetando libu2f-udev (1.1.4-1ubuntu0.1) ...
Configurando brave-keyring (1.12) ...
Configurando libu2f-udev (1.1.4-1ubuntu0.1) ...
Configurando brave-browser (1.25.72) ...
update-alternatives: utilizando /usr/bin/brave-browser-stable para proveer /usr/bin/x-www-browser (x-www-browser) en modo automático
update-alternatives: utilizando /usr/bin/brave-browser-stable para proveer /usr/bin/gnome-www-browser (gnome-www-browser) en modo automático
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...
Procesando disparadores para gnome-menus (3.13.3-1ubuntu1) ...
```

Brave en Arch Linux, Manjaro y derivados

En Arch primero instalamos el asistente de AUR, yay.

```
sudo pacman -S --needed git base-devel
git clone https://aur.archlinux.org/yay.git
cd yay
makepkg -si
```

```
yay -S brave
```

También está disponible en pacman.

```
sudo pacman -S brave
```



designed by freepik

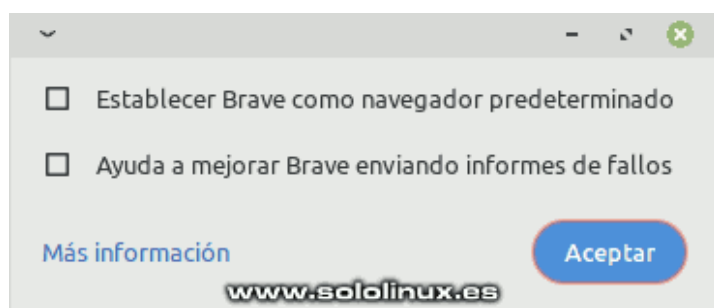
Brave en Elementary OS

```
sudo apt install apt-transport-https curl
curl -s https://brave-browser-apt-release.s3.brave.com/brave-core.asc
| sudo apt-key --keyring /etc/apt/trusted.gpg.d/brave-browser-
release.gpg add -
echo "deb [arch=amd64] https://brave-browser-apt-release.s3.brave.com/
stable main" | sudo tee /etc/apt/sources.list.d/brave-browser-
release.list
sudo apt update
sudo apt install brave-browser
```

Brave en Linux con Snap

```
sudo snap install brave
```

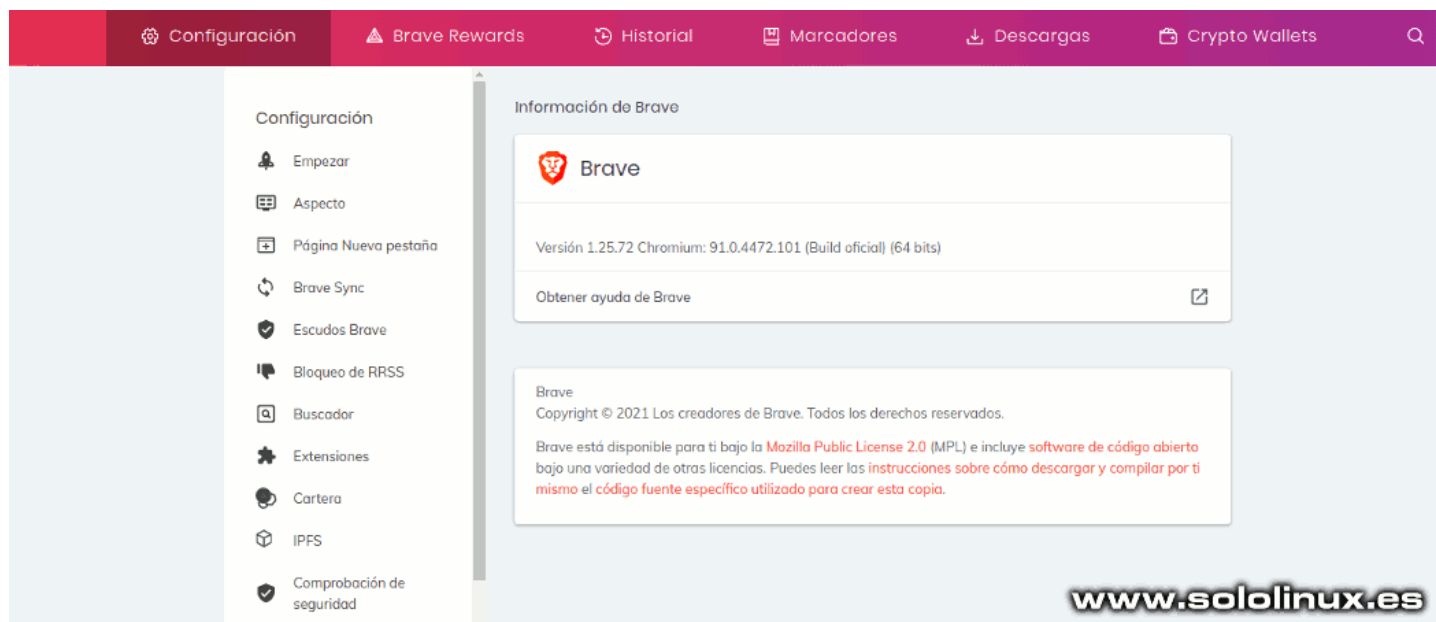
Una vez instalado Brave, lo puedes iniciar desde tu menú de aplicaciones, o ejecutando «brave» en tu terminal linux. Te pregunta si quieres que sea tu navegador predeterminado y, si quieres enviar informes de errores.



Antes de comenzar a usarlo, te recomiendo que revises la configuración por defecto del navegador web.

Nota del autor: Como comentamos en un [artículo anterior](#), Google permite el inicio de sesión de tu cuenta en navegadores derivados de Chromium, pero bloquea la api de los datos. Esto quiere decir que no puedes importar los marcadores, contraseñas, etc, desde otros navegadores.

Dicen los mentideros de internet, que en esta drástica decisión influyo mucho la proliferación de navegadores como Brave. Algo lógico, pues se aprovechan de la tecnología desarrollada por Google así como de su red, para después bloquear su AdSense e incrustar sus anuncios. De todas formas... si quieres que los blogs existan tal como los conoces hoy en día, no bloques **AdSense** dado que es su única forma de financiación para mantenerse online.

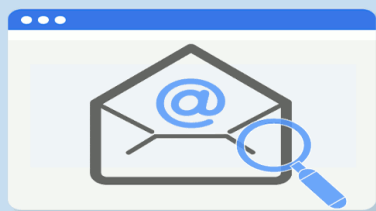


Buscar los correos de un dominio en Google

De todos es sabido, que si quieres evitar recibir correos no deseados más conocidos como **spam**, tus direcciones de correo electrónico no deben ser públicas y, aún menos que sean capturadas por buscadores de internet (serán publicadas sin remedio).

Salvo excepciones, en las cuales por temas de seo se requiere publicar un mail de contacto, las normas básicas de seguridad hacen hincapié en todo lo contrario. A pesar de ello, muchos despistados publican su dirección sin conocer las consecuencias que esto puede provocar.

A veces, un simple error humano puede llevar tu buzón de correo al colapso. Hoy quería presentarte un script programado en **Python**, con el cual podrás averiguar las direcciones de un dominio que han sido capturadas por los buscadores.



Buscar los correos de un dominio en Google

Para lograr nuestro objetivo, usaremos la herramienta **EmailFinder** escrita por **Josué Encinar**. El único requisito necesario para instalar la herramienta, es **pip3**. Si no recuerdas como instalar pip en tu distribución Linux predeterminada, revisa [este anterior artículo](#).

Ejemplo...

```
root@sololinux:~# sudo apt install python3-pip
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  dh-python libpython3-dev libpython3.6-dev python3-
asn1crypto
  python3-cffi-backend python3-cryptography python3-dev
python3-keyring
  python3-keyrings.alt python3-secretstorage python3-wheel
python3.6-dev
Paquetes sugeridos:
  python-cryptography-doc python3-cryptography-vectors
libkf5wallet-bin
  gir1.2-gnomekeyring-1.0 python-secretstorage-doc
Se instalarán los siguientes paquetes NUEVOS:
  dh-python libpython3-dev libpython3.6-dev python3-
asn1crypto python3-cffi-backend python3-cryptography
python3-dev python3-keyring
  python3-keyrings.alt python3-pip python3-secretstorage
python3-wheel python3.6-dev
0 actualizados, 13 nuevos se instalarán, 0 para eliminar y 0
no actualizados.
Se necesita descargar 46,0 MB de archivos.
Se utilizarán 80,4 MB de espacio de disco adicional después
de esta operación.
¿Desea continuar? [S/n]
```

Ahora instalamos **EmailFinder**, con el siguiente comando.

```
pip3 install emailfinder
```

Una vez instalada la herramienta, su uso es tan sencillo como ejecutar esta sintaxis acompañada del dominio.

```
emailfinder -d [dominio.es]
```

En nuestro ejemplo, buscamos las direcciones capturadas por los buscadores de una conocida emisora de radio española.

```
emailfinder -d cope.es
```

Comienza la búsqueda, pero en este caso **yandex** nos bloquea. Supongo que al realizar demasiadas búsquedas frecuentes, rechaza el rastreador del script.

```
root@SoloLinux-demo:~# emailfinder -d cope.es
  _ _ _ _ _
 _ _ _ _ _ \ _ _ _
 \ _ _ _ _ _ \ _ _ _
 \ _ _ _ _ _ \ _ _ _

|_ Author: @JosueEncinar
|_ Description: Search emails from a domain through search
engines.
|_ Version: 0.2.3b
|_ Usage: emailfinder -d domain.com
Searching in google...
Searching in bing...
Searching in baidu...
Searching in yandex...
[+] bing done!
[!] yandex error YandexDetection, Robot detected
[+] baidu done!
[+] google done!
Total emails: 18
-----
```

Como es lógico, no voy a publicar ni como texto ni como código el mail de nadie, pero si la imagen del ejemplo con el resultado final obtenido.

```
|_ Author: @JosueEncinar
|_ Description: Search emails from a domain through search engines.
|_ Version: 0.2.3b
|_ Usage: emailfinder -d domain.com

Searching in google...
Searching in bing...
Searching in baidu...
Searching in yandex...
[+] bing done!
[!] yandex error YandexDetection, Robot detected
[+] baidu done!
[+] google done!

Total emails: 18
-----
internet.usuarios@cope.es
informativos.larioja@cope.es
malaga@cope.es
janedoe@cope.es
gcomercial@cope.es
calahorra@cope.es
latarde@cope.es
direccion.extremadura@cope.es
programas.caceres@cope.es
jaca@cope.es
fallas@cope.es
direccion.valencia@cope.es
canal.denuncias@cope.es
informativos.caceres@cope.es
tiempodejuego@cope.es
soporteweb@cope.es
datospersonales@cope.es
linterna@cope.es
root@SoloLinux-demo:~#
```

www.sololinux.es

Si quieres actualizar la herramienta, ejecuta...

```
pip3 install emailfinder --upgrade
```

```
root@SoloLinux-demo:~# pip3 install emailfinder --upgrade
Requirement already up-to-date: emailfinder in /usr/local/lib/python3.8/dist-packages (0.2.3b0)
Requirement already satisfied, skipping upgrade: prompt-toolkit>=3.0.5 in /usr/local/lib/python3.8/dist-packages (from emailfinder) (3.0.18)
Requirement already satisfied, skipping upgrade: beautifulsoup4>=4.9.3 in /usr/local/lib/python3.8/dist-packages (from emailfinder) (4.9.3)
Requirement already satisfied, skipping upgrade: requests>=2.25.1 in /usr/local/lib/python3.8/dist-packages (from emailfinder) (2.25.1)
Requirement already satisfied, skipping upgrade: urllib3>=1.26.4 in /usr/local/lib/python3.8/dist-packages (from emailfinder) (1.26.5)
Requirement already satisfied, skipping upgrade: pyfiglet>=0.8.post1 in /usr/local/lib/python3.8/dist-packages (from emailfinder) (0.8.post1)
Requirement already satisfied, skipping upgrade: wcwidth in /usr/local/lib/python3.8/dist-packages (from prompt-toolkit>=3.0.5->emailfinder) (0.2.5)
Requirement already satisfied, skipping upgrade: soupsieve>1.2; python_version >= "3.0" in /usr/local/lib/python3.8/dist-packages (from beautifulsoup4>=4.9.3->emailfinder) (2.2.1)
Requirement already satisfied, skipping upgrade: certifi>=2017.4.17 in /usr/local/lib/python3.8/dist-packages (from requests>=2.25.1->emailfinder) (2021.5.30)
Requirement already satisfied, skipping upgrade: chardet<5,>=3.0.2 in /usr/local/lib/python3.8/dist-packages (from requests>=2.25.1->emailfinder) (4.0.0)
Requirement already satisfied, skipping upgrade: idna<3,>=2.5 in /usr/local/lib/python3.8/dist-packages (from requests>=2.25.1->emailfinder) (2.10)
root@SoloLinux-demo:~#
```

También puedes verificar la versión de EmailFinder instalada.

```
emailfinder -v
```

En nuestro caso...

```
root@SoloLinux-demo:~# emailfinder -v
0.2.3b
root@SoloLinux-demo:~#
```

AYUDANOS A SEGUIR
CRECIENDO

PayPal™ PayPal™

Donar a Revistalinux

Donar a SoloLinux

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

[Chat de SoloLinux en Telegram](#)



designed by freepik

Iniciar el modo recovery en Ubuntu y otros linux



Todas las **distribuciones linux** que utilizan **Grub**, tienen un entorno de restauración más conocido como «**Modo Recovery**». En este modo, algunas distros incluyen herramientas con la capacidad de borrar datos, instalar actualizaciones del sistema, restaurar datos, copias de seguridad o reiniciar el propio sistema.

El acceso al **modo recovery** no siempre está disponible, ya que actualmente muchas distribuciones se saltan el **menú de grub** durante el inicio del sistema, para lograr un arranque más rápido. En este artículo conoceremos las diferentes opciones del **modo recuperación**, pero antes vemos como **acceder al Grub**, en caso de que no aparezca por defecto.

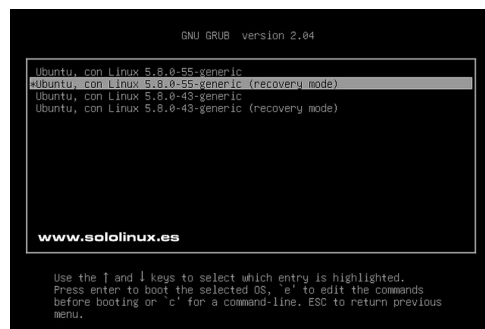
- **Con UEFI** – Pulsa la **tecla ESC** (escape), al iniciar el sistema.
- **Con BIOS** – Pulsa la **tecla Shift** (mayúsculas), al iniciar el sistema.



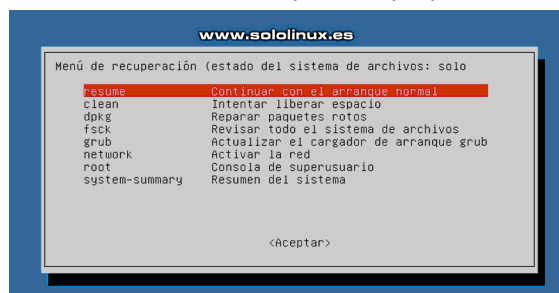
Iniciar el modo recovery en Ubuntu y otros linux

Recuerda que los pasos que vemos a continuación, son muy similares en las distros que usan **Grub** (especialmente derivados de Debian); en nuestro caso tomamos Ubuntu a modo de ejemplo, dado que es el más extendido entre los **usuarios de linux**.

Una vez estemos en la pantalla del menú del Grub, con el uso de las flechas marcas «**Opciones avanzadas para Ubuntu**» y pulsas enter. Aparece una pantalla con todos los kernel instalados en el sistema, selecciona el kernel en modo recuperación que más te interese. Pulsa la **tecla enter**.



Nos aparece otro menú, en el cual podemos elegir lo que necesitamos entre las opciones propuestas.



Explicamos las opciones del menú de recuperación.

- **resume** – Arranque normal con el kernel seleccionado.
- **clean** – Intenta liberar espacio en el disco, mediante el uso de comandos como pueden ser «autoremove».
- **dpkg** – Repara posibles paquetes rotos y actualiza el sistema si es necesario.
- **fsck** – Revisa la integridad del sistema de archivos.
- **grub** – Actualiza el Grub si has realizado algún cambio en el mismo.
- **network** – Habilita la red.
- **root** – Acceder a la terminal del usuario root.
- **system-summary** – Ofrece la siguiente información del sistema: Información general, Uso detallado del disco, Estado del software raid, Estado de LVM, Uso detallado de la memoria, Configuración detallada de la red, Base de datos del sistema (apt).

Como podemos ver en el menú, las opciones son muchas y variadas, seguro que alguna de ellas es la solución a tu problema. No olvides como acceder al modo recovery.

- **Con UEFI** – Pulsa la **tecla ESC** (escape), al iniciar el sistema.
- **Con BIOS** – Pulsa la **tecla Shift** (mayúsculas), al iniciar el sistema.

Bien, ya lo tenemos instalado. Puedes verificar con...

```
python3.9 --versión
```

Hemos verificado que lo tenemos instalado, pero no es la versión predeterminada del sistema. Si quieres que Python 3.9.5 sea la versión por defecto, ejecuta lo siguiente.

```
update-alternatives --install /usr/bin/python python /usr/local/bin/python3.9 1
```

Ahora sí que es la versión predeterminada, lo verificamos.

```
python --version
```

Salida de ejemplo completa.

```
root@SoloLinux-demo:~# python3.9 --version
Python 3.9.5
root@SoloLinux-demo:~# python --version
Python 2.7.16
root@SoloLinux-demo:~#
root@SoloLinux-demo:~# python3 --version
Python 3.7.3
root@SoloLinux-demo:~# update-alternatives --install /usr/bin/python python /usr/local/bin/python3.9 1
update-alternatives: using /usr/local/bin/python3.9 to provide /usr/bin/python (python) in auto mode
root@SoloLinux-demo:~#
root@SoloLinux-demo:~# python --version
Python 3.9.5
root@SoloLinux-demo:~#
```

AYUDANOS A SEGUIR CRECIENDO

PayPal™ PayPal™

[Donar a Revistalinux](#)

[Donar a SoloLinux](#)

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

[Chat de SoloLinux en Telegram](#)



designed by freepik

Buscar virus con un Live USB ClamAV Antivirus

Antivirus Live CD es un fork oficial de la **distribución linux 4MLinux**, en la cual se ha incluido el antivirus desarrollado para plataformas basadas en Unix, **ClamAV** operando en modo **Live USB**.

Esta pequeña ISO de apenas 280MB, está desarrollada especialmente para usuarios que necesitan un Live CD ligero, que cuente con la capacidad de analizar y reparar un sistema infectado por **virus**, **malwares** y resto de basura que circula por la red.

Antivirus Live CD con ClamAV Antivirus, soporta las conexiones Ethernet, WiFi, PPP y PPPoE, para poder actualizar la base de datos de las firmas ClamAV de forma sencilla y rápida. Al iniciar la ISO en modo Live, se montan automáticamente todas las particiones disponibles en la máquina, para poder escanearlas.

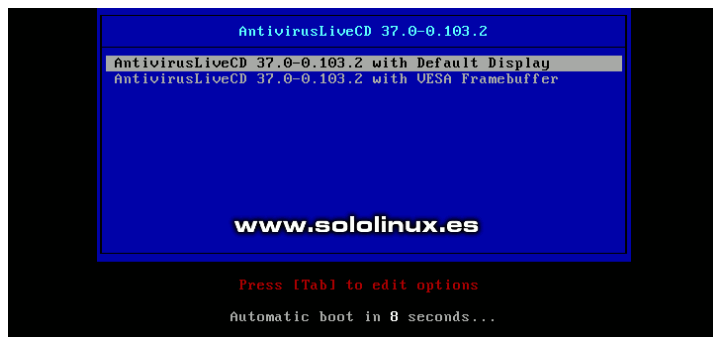
Las imágenes ISO son compatibles con la mayoría de herramientas de grabación, incluyendo a **Unetbootin** y **Ventoy**. La última versión de este fantástico Live Antivirus, es la 37.0-0.103.2, basado en la **distribución linux 4MLinux 37.0** al que le adjunta **ClamAV 0.103.2**.

Buscar virus con un Live USB ClamAV Antivirus

Puedes descargar la **última versión** desde su página oficial siguiendo este enlace, o con el comando **wget**.

```
wget
https://sourceforge.net/projects/antiviruslivecd/files/AntivirusLiveCD-37.0-0.103.2.iso
```

Una vez descargada la ISO, creas un **Live USB** con ella e, inicias la máquina desde el dispositivo con el antivirus instalado. La primera pantalla que nos aparece es la siguiente..., puedes pulsar enter o esperar a que termine la cuenta atrás.



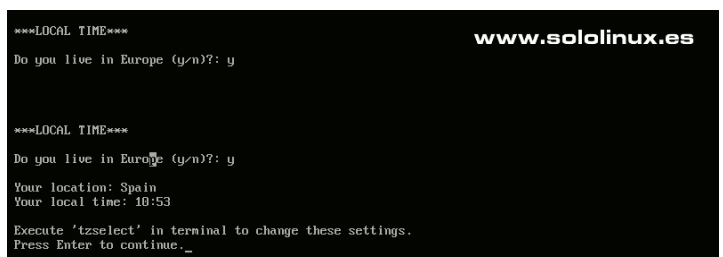
Una vez inicie la distribución, nos pregunta si queremos cambiar el idioma por defecto. Realmente no es necesario, además **ClamAV** viene en inglés por defecto. Pulsa «n» y la tecla enter.



En la pantalla que viene a continuación, te pregunta si quieres introducir una contraseña al **usuario root**. Vamos a ver... tan solo queremos limpiar nuestro sistema con una Live antivirus con ClamAV integrado. Pulsamos en «n» de NO, y la **tecla enter** para continuar.



Continuamos con la configuración. Por temas de horario nos pregunta nuestra localización; en realidad no es necesario a no ser que quieras guardar los reportes a modo informativo.



Al pulsar la tecla enter, nos pide el usuario y la contraseña (en caso de que la introdujeras). Por defecto, el **usuario de Antivirus Live CD** es «root», en nuestro caso la contraseña la dejamos en blanco y pulsamos de nuevo la **tecla enter**.



Para lanzar la herramienta ejecutamos el comando...

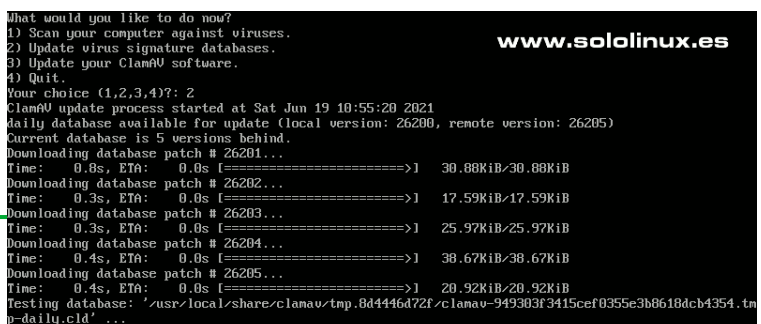
```
antivir
```

Nos aparece un menú, sencillo, práctico y sin posibilidad de error por nuestra parte.

- 1) Escanear todas las particiones de los dispositivos insertados en la máquina.
- 2) Actualizar las firmas de ClamAV.
- 3) Actualizar la herramienta ClamAV.
- 4) Salir.



Antes de analizar... seleccionamos la opción 2 para actualizar las firmas del antivirus.



Una vez termine la actualización, marcas la **opción 1** para analizar el sistema. Nos aparecen las opciones sobre como tratar los archivos infectados, te recomiendo la opción 3 (guardar en cuarentena). Pulsas enter para comenzar el escáner de archivos. El proceso puede ser lento, ten paciencia. Al terminar el escáner nos muestra el resultado del mismo.

Como **distribución linux** se comporta como cualquier otra, viene con herramientas como nano, top, etc.

```
Mem: 834068K used, 1469324K free, 312K shrd, 20K buff, 751524K cached
CPU:  0.2% usr  0.4% sys  0.0% nic 96.8% idle  0.0% io  0.0% irq  2.4% irq
Load average: 0.37 0.71 0.46 1/73 895
```

PID	PPID	USER	STAT	VSZ	%VSZ	CPU	%CPU	COMMAND
895	849	root	R	4640	0.2	0	0.4	top
802	1	root	S	3712	0.1	0	0.4	gpm -m /dev/input/mice -t ps2
798	1	root	S	4640	0.2	0	0.2	acpid
5	2	root	IW	0	0.0	0	0.2	[kworker/0:0-eve]
644	1	root	S	13268	0.5	0	0.0	udevd --daemon
1	0	root	S	4640	0.2	0	0.0	init
849	1	root	S	4640	0.2	0	0.0	-ash
229	1	root	S	4640	0.2	0	0.0	syslogd
231	1	root	S	4640	0.2	0	0.0	klogd -c 3
818	1	root	S	4640	0.2	0	0.0	udhcpd -i eth0 -n -p /var/run/dhcp.pid
850	1	root	S	4640	0.2	0	0.0	getty 38400 tty2
853	1	root	S	4640	0.2	0	0.0	getty 38400 tty3
858	1	root	S	4640	0.2	0	0.0	getty 38400 tty4
859	1	root	S	4640	0.2	0	0.0	getty 38400 tty5
860	1	root	S	4640	0.2	0	0.0	getty 38400 tty6
773	1	root	S	3476	0.1	0	0.0	/sbin/v86d
673	1	messageb	S	3340	0.1	0	0.0	dbus-daemon --system
13	2	root	IW	0	0.0	0	0.0	[rcu_sched]
12	2	root	SW	0	0.0	0	0.0	[ksoftirqd/0]
21	2	root	SW	0	0.0	0	0.0	[kcompactd0]
170	2	root	IW	0	0.0	0	0.0	[kworker/0:2-ata]
112	2	root	IW<	0	0.0	0	0.0	[kworker/0:1H-kb]
146	2	root	IW	0	0.0	0	0.0	[kworker/u2:1-ev]
2	0	root	SW	0	0.0	0	0.0	[kthreadd]
3	2	root	IW<	0	0.0	0	0.0	[rcu_gp]
4	2	root	IW<	0	0.0	0	0.0	[rcu_par_gp]
6	2	root	IW<	0	0.0	0	0.0	[kworker/0:0H-ev]
8	2	root	IW	0	0.0	0	0.0	[kworker/u2:0-ev]
9	2	root	IW<	0	0.0	0	0.0	[mm_percpu_wq]
10	2	root	SW	0	0.0	0	0.0	[rcu_tasks_rude_]
11	2	root	SW	0	0.0	0	0.0	[rcu_tasks_trace]
14	2	root	SW	0	0.0	0	0.0	[migration/0]
15	2	root	SW	0	0.0	0	0.0	[cpuhp/0]

www.sololinux.es

SOLO LINUX



designed by freepik

21 comandos linux que debes aprender



Los comandos de Linux para **terminal**, pueden dar una primera impresión de ser complejos de usar por un usuario principiante, pero a medida que van aprendiendo, se percatan de que son fáciles de utilizar además de poderosos y efectivos.

Muchas tareas que se realizan habitualmente a través de la terminal, también las podríamos hacer desde la **GUI**, pero seamos claros... desde terminal todo es más rápido y confiable. Además, si eres lector habitual de **sololinux** es porque quieres aprender.

A continuación, vemos los 21 comandos linux que todo principiante debe conocer y, que resultaran de mucha utilidad a la hora de administrar correctamente un sistema GNU/Linux.



21 comandos linux que debes aprender

Es evidente que para gustos existen los colores, seguro que echaras alguno en falta, pero también es seguro que los que están son indispensables. Trataremos de poner ejemplos en casi todos los comandos.

Comando ls

El comando **ls**, tiene la capacidad de listar archivos y carpetas de una ruta definida. Puedes leer su manual en el siguiente [enlace](#).

```
ls [-opciones] [ruta]
```

Ejemplos:

```
ls
ls -l
ls -ltr
```

Comando cd

El comando **cd** es útil para abrir, cambiar o moverte por diferentes directorios. Al escribir solo el comando **cd**, vuelves al directorio de inicio. Puedes leer su manual [aquí](#).

```
sergio@sololinux:~$
sergio@sololinux:~$ cd Descargas
sergio@sololinux:~/Descargas$ cd sololinux
sergio@sololinux:~/Descargas/sololinux$ cd ..
sergio@sololinux:~/Descargas$ cd
sergio@sololinux:~$
```

Comando mkdir

Este comando se usa para crear nuevos **directorios**. Es bastante sencillo de utilizar, pero si quieres puedes revisar [su manual](#).

```
mkdir micarpera
```

Comando man

La herramienta «**man**» es un interesante comando, con el cual puedes consultar el manual integrado de cualquier **comando linux**, incluyendo el propio comando. Por ejemplo...

```
man [comando/herramienta]
```

```
man man
man nano
```

Ejemplo de salida del manual de **nano**.

```
NANO(1)                                General Commands Manual                                NANO(1)
NAME
    nano - Nano's ANOther editor, an enhanced free Pico clone
SYNOPSIS
    nano [options] [[+line[,column]] file]...
DESCRIPTION
    nano is a small and friendly editor. It copies the look and feel of Pico, but is free software, and implements several features that Pico lacks, such as: opening multiple files, scrolling per line, undo/redo, syntax coloring, line numbering, and soft-wrapping overlong lines.

    When giving a filename on the command line, the cursor can be put on a specific line by adding the line number with a plus sign (+) before the filename, and even in a specific column by adding it with a comma.

    As a special case: if instead of a filename a dash (-) is given, nano will read data from standard input.
```

Comando pwd

El comando **pwd**, ejecuta una variable de entorno que nos muestra la ruta actual del usuario. Esta herramienta solo admite dos opciones.

- **-P** : Imprime la ruta real.
- **-L** : Imprime la ruta simbólica.

```
sergio@sololinux:~$ cd Descargas
sergio@sololinux:~/Descargas$ pwd
/home/sergio/Descargas
sergio@sololinux:~/Descargas$ pwd -P
/home/sergio/Descargas
sergio@sololinux:~/Descargas$ pwd -L
/home/sergio/Descargas
sergio@sololinux:~/Descargas$ cd
sergio@sololinux:~$
```

Comando echo

El «comando echo» pasa textos como argumento, para imprimirlos en pantalla. Puedes revisar [un anterior artículo](#), donde aprenderás a utilizarlo con ejemplos.

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

Comando locate

Si quieres buscar un archivo, directorio o una cadena de caracteres, locate es probablemente el comando más seguro y rápido. Su uso es simple, además hablamos de él hace muy poco [en este artículo](#). Vemos un ejemplo...

```
locate libera.sh
```

```
sergio@sololinux:~$ locate libera.sh
/home/sergio/libera.sh
/home/sergio/Descargas/libera.sh
/home/sergio/Escritorio/libera.sh
sergio@sololinux:~$
```

Comando history

History nos muestra el historial de las herramientas ejecutadas en terminal, desde que iniciaste la sesión. Puedes aprender más sobre él, [aquí](#).

```
history
```

```
sergio@sololinux:~$ history
1  ls -l | grep brave
2  cd ~/.config
3  ls -l | grep brave
4  cd
5  locale
6  locale text
7  locate
8  locate text
9  locate libera
10 locate libera.sh
11 sudo su
12 ls -ltr
13 cd home
14 cd sergio
15 cd descargas
16 cd /descargas
17 cd Descargas
```

www.sololinux.es

Comando sudo

El comando sudo nos permite ejecutar tareas que requieren de permisos administrativos. En sololinux hemos hablado mucho sobre el comando sudo, [pulsa aquí](#) para ver los artículos al respecto.

```
sudo su
```

```
sudo -V
```

```
sergio@sololinux:~$ sudo -V
Sudo versión 1.8.21p2
versión del complemento de políticas de sudoers 1.8.21p2
versión de gramática del archivo Sudoers 46
Sudoers I/O plugin version 1.8.21p2
```

Comando ping

Packet Internet Groper, más conocido como ping, es una herramienta con la capacidad de verificar y medir el estado de conectividad entre el host y un servidor, utilizando el [protocolo ICMP](#). Aprende más en [«Hacer ping con ejemplos»](#).

```
ping [-opciones] [URL / IP]
```

```
sergio@sololinux:~$ ping google.es
PING google.es (172.217.19.99) 56(84) bytes of data.
64 bytes from muc03s07-in-f99.1e100.net (172.217.19.99): icmp_seq=1
ttl=115 time=14.2 ms
64 bytes from muc03s07-in-f99.1e100.net (172.217.19.99): icmp_seq=2
ttl=115 time=37 ms
64 bytes from muc03s07-in-f99.1e100.net (172.217.19.99): icmp_seq=3
ttl=115 time=16.2 ms
```

Comando kill

El comando kill nos permite matar cualquier proceso activo del sistema, si no sabes como identificar los procesos... [visita este artículo](#); Para ampliar información sobre como matar procesos, también escribimos un post hace un tiempo, [artículo sobre kill](#). Ejemplo...

```
kill -9 3587
```

Comando wget

El comando wget es indispensable en tareas cotidianas, nos permite descargar archivos e incluso sitios web completos. Puedes aprender más en [este enlace](#). En el ejemplo descargamos la última [versión de WordPress](#), desde su página oficial.

```
wget https://wordpress.org/latest.zip
```

```
sergio@sololinux:~$ wget https://wordpress.org/latest.zip
--2021-06-21 19:55:48-- https://wordpress.org/latest.zip
Resolviendo wordpress.org (wordpress.org)... 198.143.164.252
Conectando con wordpress.org (wordpress.org)[198.143.164.252]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK www.sololinux.es
Longitud: 16866701 (16M) [application/zip]
Guardando como: "latest.zip.1"

latest.zip.1 100%[=====] 16,08M 2,21MB/s en 13s
2021-06-21 19:56:01 (1,24 MB/s) - "latest.zip.1" guardado [16866701/16866701]
```

Comando clear

El comando clear, limpia la pantalla de la terminal.

```
clear
```

Comando top

Con la [herramienta top](#), puedes verificar el estado del sistema. Usuarios, procesos, consumos, etc, desde una sencilla interfaz integrada en la [terminal](#).

```
top
```

```
top - 19:07:03 up 5 days, 6:41, 1 user, load average: 0.40, 0.18, 0.10
Tasks: 164 total, 1 running, 163 sleeping; 0 stopped, 0 zombie
Cpu(s): 1.7 us, 1.4 sy, 0.0 ni, 96.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
Mem: 7277988 total, 221772 free, 321776 used, 7224432 buff/cache
KiB Swap: 3145728 total, 3144696 free, 1024 used, 669116 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR   S  %CPU  %MEM     TIME+ COMMAND
350207 root        20   0 175048 24924 2220 S   0.7   0.3   1:56.71 lfd - sle
647063 root        20   0 85648 5296 3896 S   0.7   0.1   0:00.02 auth
9 root        20   0 0 0 0 S   0.0   0.0   5:47.41 rcu sched
394 root        20   0 63744 25108 24786 S   0.3   0.3   1:10.85 systemd-j
896 root        20   0 545988 20360 16696 S   0.3   0.3   1:49.57 rsyslogd
1251 mysql      20   0 207160 102664 15500 S   0.2   1.3   1:26.98 mariadb
1760 nobody     20   0 58684 21228 5182 S   0.3   0.3   8:06.54 litespeed
1761 nobody     20   0 63296 25824 5148 S   0.3   0.3   5:51.06 litespeed
822104 dovenull 20   0 46916 4392 3372 S   0.1   0.1   0:00.14 inap-login
647037 root        20   0 162100 2300 1604 R   0.3   0.0   0:00.03 top
647041 cyberpa+ 21   1 158004 35032 5480 S   0.3   0.4   0:00.43 lswgl
1 root        20   0 191328 4236 2616 S   0.0   0.1   1:05.40 systemd
2 root        20   0 0 0 0 S   0.0   0.0   0:00.05 kthreadd
4 root        20   0 0 0 0 S   0.0   0.0   0:00.00 kworker/0-
6 root        20   0 0 0 0 S   0.0   0.0   0:02.07 ksoftirqd+
www.sololinux.es 0 0 0 0 0 S   0.0   0.0   0:00.14 migration+
8 root        20   0 0 0 0 S   0.0   0.0   0:00.00 rcu bh
```

Comando head

El comando head muestra las 10 primeras líneas de un archivo. [Manual oficial de head.](#)

```
# Ejemplo
head libera.sh
```

```
sergio@sololinux:~$ head libera.sh
#!/bin/bash
echo "<----->"
echo "<----->"
echo "Comprobando estado de memoria"
free
echo "OK - Comprobacion completada"
echo "<----->"
sleep 2s; echo "Limpieza de memoria cache y swap";
echo "<----->"
```

Comando tail

El comando tail muestra las 10 últimas líneas de un archivo. [Manual oficial de tail.](#)

```
# Ejemplo
tail libera.sh
```

```
sergio@sololinux:~$ tail libera.sh
echo "Script liberador sh"
echo "Created by SergioG.B."
echo "https://www.sololinux.es"
echo "<----->"
sleep 2s; free
echo "....TODO CORRECTO...."
echo "<----->"
echo "<----->"
sergio@sololinux:~$
```

Comando rmdir

El comando **rmdir** borra directorios vacíos.

```
rmdir [carpeta / directorio]
```

Comando rm

El **comando rm**, tiene la capacidad de borrar cualquier archivo del sistema, incluyendo directorios sin contenido. Si el directorio tiene contenido y lo quieres borrar por completo, debes añadir la opción **-r**.

```
rm [archivo] [directorio]
# Borrar todo el contenido
rm -r [directorio]
```

Comando mv

Es evidente que en el artículo «21 comandos linux que debes aprender», no podía faltar el **comando mv**. Este comando tiene la capacidad de mover archivos de lugar, mientras cambia el nombre a la vez (o simplemente cambiar el nombre). Puedes aprender más sobre este comando, en un [artículo anterior](#).

Observa el siguiente ejemplo, en el cual cambiamos el nombre archivo1 por archivo2.

```
mv archivo1.txt archivo2.txt
```

```
sergio@sololinux:~/demo$ dir
archivo1.txt
sergio@sololinux:~/demo$ mv archivo1.txt archivo2.txt
sergio@sololinux:~/demo$ dir
archivo2.txt
sergio@sololinux:~/demo$ www.sololinux.es
```

Comando exit

Bueno... vamos llegando al final del artículo «**21 comandos linux que debes aprender**». No nos podíamos olvidar del comando exit, comando extremadamente útil para cerrar la **shell** o la ventana activa de la terminal.

```
exit
```

Comando reboot

Existen varios comandos para reiniciar tu **sistema linux**, pero no le des vueltas, **reboot** es el más utilizado además de ser compatible con la mayoría de sistemas GNU / Linux.

```
reboot
# o
sudo reboot
```

AYUDANOS A SEGUIR
CRECIENDO

PayPal™ PayPal™

Donar a Revistalinux

Donar a SoloLinux

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

[Chat de SoloLinux en Telegram](#)



designed by freepik

Migrar CentOS 8 a Rocky Linux 8.4

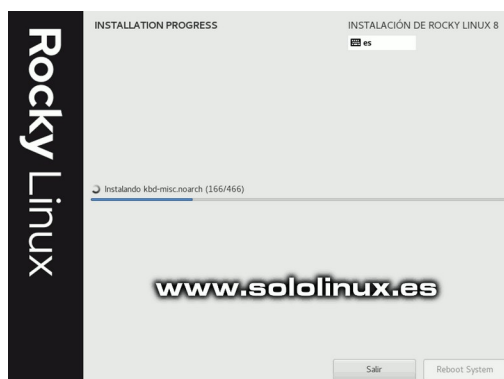


Derivado de **Red Hat Enterprise Linux 8.4**, Rocky Linux 8.4 ha lanzado su primera versión estable. Es realmente sorprendente, que se incluyan módulos actualizados con respecto a su distribución matriz (Red Hat), como **MariaDB 10.5**, OpenSCA 1.3.4, PostgreSQL 13, **Python 3.9**, Redis 6, scap-security-guide 0.1.54, Subversion 1.14 y SWIG 4.0, además de las utilidades de programación GCC 10, LLVM 11.0.0, Rust 1.49 y Go 1.15.7.

Nos comunican que también hay cambios importantes en aspectos de seguridad, en redes, en gestión de identidades, la agrupación en clústeres, el kernel, etc. Además se ofrece soporte para la encapsulación TCP y, etiquetas de seguridad para el protocolo **IKEv2** para la **VPN IPsec** (proporcionada por **Libreswan**), y varios añadidos más.

Cuando estallo la burbuja de CentOS (lo explicamos en [un anterior artículo](#)), Alma Linux y Rocky Linux saltaron a la palestra como los principales valedores, a la hora de desarrollar un derivado directo y 100% compatible con Red Hat Enterprise. Hace un tiempo **analizamos Rocky Linux en su versión beta**, el resultado fue un fiasco total; por el contrario... Alma Linux ya lo tenemos en producción.

Visto que por fin lanzaron una versión estable, hoy vemos como actualizar un servidor CentOS 8 a Rocky Linux 8.4. También instalaremos un Rocky Linux 8.4 Desktop, esperando que nos deje un mejor sabor de boca que su antecesor.



Migrar CentOS 8 a Rocky Linux 8.4

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

```
sudo dnf -y upgrade
```

Una vez se actualice el sistema, reiniciamos y posteriormente verificamos la **versión instalada de CentOS**.

```
sudo reboot
cat /etc/*-release
```

En el ejemplo vemos que disponemos de la última versión estable de **CentOS 8**.

```
[root@SoloLinux-demo ~]# cat /etc/*-release
CentOS Linux release 8.4.2105 <----- version instalada
NAME="CentOS Linux"
VERSION="8"
ID="centos"
ID_LIKE="rhel fedora"
VERSION_ID="8"
PLATFORM_ID="platform:el8"
PRETTY_NAME="CentOS Linux 8"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:centos:centos:8"
HOME_URL="https://centos.org/"
BUG_REPORT_URL="https://bugs.centos.org/"
CENTOS_MANTISBT_PROJECT="CentOS-8"
CENTOS_MANTISBT_PROJECT_VERSION="8"
CentOS Linux release 8.4.2105 <----- version instalada
CentOS Linux release 8.4.2105 <----- version instalada
[root@SoloLinux-demo ~]#
```

Si algo debo destacar de Rocky Linux, es el desarrollo de un script llamado **«migrate2rocky»**, que automatiza la tarea de migración entre Centos 8.4 y Rocky Linux 8.4. Lástima que la descarga oficial da error, así que lo creamos de forma manual.

```
sudo nano migrate2rocky.sh
```

Copia y pega lo siguiente.

Descarga el código de [Aquí](#)

Le concedemos los permisos necesarios.

```
chmod +x migrate2rocky.sh
```

Ahora lo ejecutamos con el siguiente comando.

```
sudo bash migrate2rocky.sh -r
```

El proceso es muy rápido, da la impresión que lo único que hace es actualizar nombres, agregar sus propios repositorios e instalar algún añadido. Observa la imagen de ejemplo.

```
[root@SoloLinux-demo ~]# sudo bash migrate2rocky.sh -r
Preparing to migrate CentOS Linux 8 to Rocky Linux 8.
Determining repository names for CentOS Linux 8.....
Found the following repositories which map from CentOS Linux 8 to Rocky Linux 8:
CentOS Linux 8  Rocky Linux 8
appstream      appstream
baseos         baseos
extras        extras
Getting system package names for CentOS Linux 8.....
Found the following system packages which map from CentOS Linux 8 to Rocky Linux 8:
CentOS Linux 8  Rocky Linux 8
centos-backgrounds  rocky-backgrounds
centos-gpg-keys     rocky-gpg-keys
centos-logos        rocky-logos
```

En nuestro caso solicita la instalación de config-manager.

```
dnf install 'dnf-command(config-manager)'
```

```
Complete!
Last metadata expiration check: 0:00:03 ago on mar 22 jun 2021 14:47:43 CEST.
> Leaving Shell

Removing dnf cache
Removing repos are enabled before the package swap
No such command: config-manager. Please use /bin/dnf --help
It could be a DNF plugin command, try: "dnf install 'dnf-command(config-manager)'"
Repo name missing?
[root@SoloLinux-demo ~]# dnf install 'dnf-command(config-manager)'
Rocky Linux 8 - AppStream                19 MB/s | 7.1 MB    00:00
Rocky Linux 8 - BaseOS                   3.1 MB/s | 2.5 MB    00:00
Rocky Linux 8 - Extras                   7.6 kB/s | 2.7 kB    00:00
Dependencias resueltas.

=====
Paquete                Arquitectura  Versión      Repositorio  Tan.
-----
Instalando:
dnf-plugins-core      noarch       4.0.18-4.el8 baseos       68 k
Instalando dependencias:
python3-dateutil      noarch       1:2.6.1-6.el8 baseos       250 k
python3-dnf-plugins-core noarch       4.0.18-4.el8 baseos       233 k
Resumen de la transacción
-----
Instalar 3 Paquetes
Tamaño total de la descarga: 551 k
Tamaño instalado: 1.3 M
¿Está de acuerdo [s/N]? :

www.sololinux.es
```

Actualizamos.

```
sudo dnf update
```

Para finalizar... reiniciamos el sistema.

```
sudo reboot
```

Verificamos que la operación de migrar CentOS 8 a Rocky Linux 8.4 es correcta.

```
cat /etc/*-release
```

```
[root@SoloLinux-demo ~]# cat /etc/*-release
NAME="Rocky Linux"
VERSION="8.4 (Green Obsidian)"
ID="rocky"
ID_LIKE="rhel fedora"
VERSION_ID="8.4"
PLATFORM_ID="platform:el8"
PRETTY_NAME="Rocky Linux 8.4 (Green Obsidian)"
ANSI_COLOR="0;32"
CPE_NAME="cpe:/o:rocky:rocky:8.4:GA"
HOME_URL="https://rockylinux.org/"
BUG_REPORT_URL="https://bugs.rockylinux.org/"
ROCKY_SUPPORT_PRODUCT="Rocky Linux"
ROCKY_SUPPORT_PRODUCT_VERSION="8"
Rocky Linux release 8.4 (Green Obsidian)
Rocky Linux release 8.4 (Green Obsidian)
Rocky Linux release 8.4 (Green Obsidian)
[root@SoloLinux-demo ~]#

www.sololinux.es
```

Hemos instalado LAMP, con un resultado satisfactorio.

HTTP Server Test Page

This page is used to test the proper operation of an HTTP server after it has been installed on a Rocky Linux system. If you can read this page, it means that the software is working correctly.

Just visiting?

This website you are visiting is either experiencing problems or could be going through maintenance.

If you would like the administrators of this website know that you've seen this page instead of the page you've expected, you should send them an email. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

The most common email address to send to is "webmaster@example.com".

Note:

The Rocky Linux distribution is a stable and reproducible platform based on the sources of Red Hat Enterprise Linux (RHEL). With this in mind, please understand that:

- Neither the Rocky Linux Project nor the Rocky Enterprise Software Foundation have anything to do with this website or its content.
- The Rocky Linux Project nor the RHEL Foundation "owns" this website. This test page is included with the distribution.

For more information about Rocky Linux, please visit the [Rocky Linux website](#).

I am the admin, what do I do?

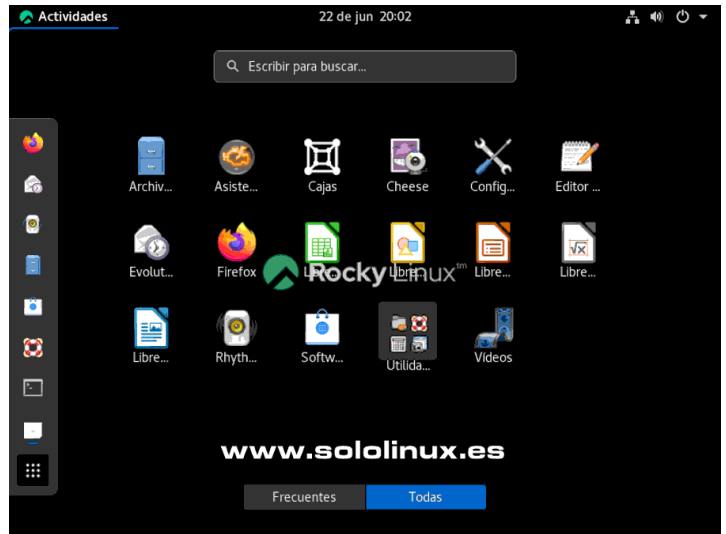
You may now add content to the webroot directory for your software.

For systems using the Apache Webserver: You can add content to the directory `/var/www/html/`. Until you do so, people visiting your website will see this page. If you would like this page to not be shown, follow the instructions in `/etc/httpd/conf.d/welcome.conf`.

For systems using Nginx: You can add your content in a location of your choice and tell the `root` configuration directive in `/etc/nginx/nginx.conf`.

www.sololinux.es

También hemos realizado una instalación limpia de **Rocky Linux 8.4 Desktop**. Su consumo es excesivo, mucho más que Alma Linux, además algunos de los problemas que observamos en la beta durante su instalación persisten, por tanto... de momento aparcaremos otra vez **Rocky Linux** esperando una nueva versión.



Nota del autor: Los módulos actualizados de Rocky Linux son a las últimas versiones disponibles, por tanto, no sigue la línea de «Red Hat Enterprise» y es más que seguro que se producirán algunas incompatibilidades con herramientas, aplicaciones, incluso con algún panel de control web. Aun siendo así, lo que hemos probado como servidor web Apache ha funcionado correctamente. Migrar CentOS 8 a Rocky Linux 8.4.

La versión desktop es un auténtico desastre, por lo menos en mi caso. Se repiten los errores verificados en la beta, a la hora de seleccionar el disco y alguno más. Las operaciones de selección y opciones, al igual que todo el proceso en general es extremadamente lento. Una vez se instala e inicia, su consumo es excesivo, muy excesivo.



designed by freepik

Canales de Telegram:

Canal SoloLinux

Canal SoloWordpress

Chat de SoloLinux en Telegram

Instalar XFCE en Alma Linux y Rocky Linux 8.4



Xfce es un entorno de escritorio no muy pesado (ligero-medio), para sistemas operativos basados en **UNIX / Linux**. Está diseñado para consumir pocos recursos mínimos del sistema, pero a diferencia de otros considerados livianos, este... es visualmente atractivo, muy completo y fácil de usar.

El escritorio Xfce se desarrolló en 1996 como una extensión del escritorio **Common Desktop Environment** (conocido como CDE), usando el kit de herramientas gráficas **XForms**. En la actualidad, ya no se utiliza código de XForms ni de CDE, pero igual se mantiene como un escritorio gráfico sólido y ligero.

Tanto Alma Linux como Rocky Linux, utilizan **Gnome** como escritorio por defecto con todo lo que ello conlleva. Si eres de los que prefieres aligerar tu máquina o **servidor con GUI**, mucho mejor con XFCE, así que vamos a ello.

Instalar XFCE en Alma Linux y Rocky Linux 8.4

El **repositorio epel** es necesario, por tanto actualizamos el sistema e instalamos el repositorio mencionado.

```
sudo dnf update
sudo dnf install epel-release
```

```
Rocky Linux 8.4 (Green Obsidian)
Kernel 4.18.0-305.3.1.el8_4.x86_64 on an x86_64
Activate the web console with: systemctl enable --now cockpit.socket

localhost login: root
Password:
root@localhost ~# dnf update
Rocky Linux 8 - AppStream                1.7 MB/s | 7.1 MB    00:04
Rocky Linux 8 - BaseOS                  688 kB/s | 2.3 MB    00:03
Rocky Linux 8 - Extras                   2.4 kB/s | 2.7 kB    00:01
Dependencias resueltas.
Paquetes por hacer:
- Listo!
root@localhost ~# dnf install epel-release
Última comprobación de caducidad de metadatos hecha hace 0:00:24, el mié 23 jun 2021 04:03:21 EDT.
Dependencias resueltas.

Paquete      Arquitectura  Versión      Repositorio  Tam.
-----
epel-release  noarch        8-10.el8     extras        22 k

Resumen de la transacción
Instalar: 1 Paquete
Tamaño total de la descarga: 22 k
Tamaño instalado: 32 k
¿Está de acuerdo [s/N]? : _
```

Podemos ver los grupos de paquetes instalados con el siguiente comando.

```
sudo dnf --enablerepo=epel group
```

Habilitamos el **powertools**.

```
sudo dnf config-manager --set-enabled powertools
```

Verificamos que tenemos XFCE en el grupo de paquetes.

```
sudo dnf group list
```

En la imagen posterior vemos que tenemos disponible XFCE.

```
root@localhost ~# dnf group list
Rocky Linux 8 - PowerTools                               628 kB/s | 2.1 MB    00:03
Última comprobación de caducidad de metadatos hecha hace 0:00:02, el mié 23 jun 2021 04:08:15 EDT.
Grupos de entorno disponibles:
  Servidor con GUI
  Server
  Estación de trabajo
  Espacios de trabajo KDE Plasma
  Host de virtualización
  Sistema operativo personalizado
Grupos de entorno instalados:
  Instalación mínima
Grupos disponibles:
  Gestión de Contenedores
  Desarrollo del núcleo de.NET
  Herramientas de desarrollo de RPM
  Herramientas de desarrollo
  Herramientas gráficas de Administración
  Gestión sin Cabeza
  Compatibilidad con legado de UNIX
  Servidores de red
  Soporte científico
  Herramientas de seguridad
  Soporte para tarjeta inteligente
  Herramientas del sistema
  Empaquetador de Fedora
  Xfce
root@localhost ~#
```

Ya puedes instalar el entorno de escritorio XFCE.

```
sudo dnf groupinstall "Xfce" "base-x"
```

Habilitamos el modo gráfico por defecto.

```
sudo systemctl set-default graphical
```

Si tu instalación de Alma Linux o Rocky Linux, era una minimal server, debes ejecutar...

```
echo "exec /usr/bin/xfce4-session" >> ~/.xinitrc
startx
```

Bien, solo necesitas reiniciar el sistema.

```
sudo systemctl reboot
```

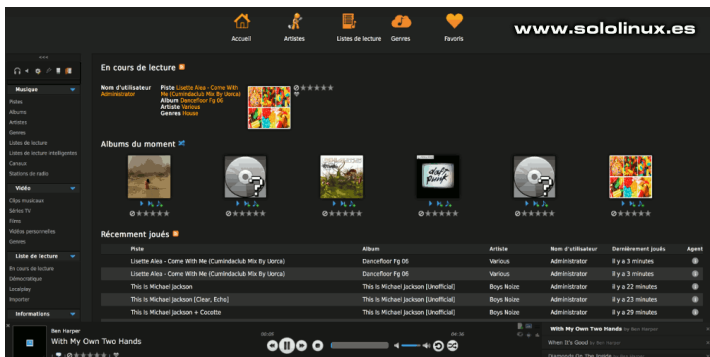


Instalar Ampache Music Streaming Server en Ubuntu 20.04



Ampache es un **servidor de música vía web** multiplataforma, además de **código abierto**. En sus inicios fue diseñado para ejecutarse en un **servidor web apache**, actualmente puede operar en otros sistemas, como por ejemplo Nginx.

Ampache puede transmitir música a pc, smartphone, tableta o TV inteligente, sin ningún tipo de limitación. En el artículo de hoy, instalaremos **Ampache en Ubuntu 20.04** a modo de ejemplo, pero es compatible con Debian 10 y todos sus derivados. En nuestro caso, usaremos Nginx como servidor web y MySQL como base de datos.



Instalar Ampache Music Streaming Server en Ubuntu 20.04

Como es habitual, primero actualizamos nuestro sistema.

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

Continuamos con la instalación de **Nginx**.

```
sudo apt install nginx
```

```
root@Sololinux-demo:~# sudo apt install nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libgd3 libjpeg-turbo8 libjpeg8 libnginx-mod-http-image-filter libnginx-
  mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream libtiff5 libwebp6
  libx11-6 libx11-data libxau6 libxcb1
  libxdmcp6 libxpm4 libxslt1.1 nginx-common nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc
The following NEW packages will be installed:
  libgd3 libjpeg-turbo8 libjpeg8 libnginx-mod-http-image-filter libnginx-
  mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream libtiff5 libwebp6
  libx11-6 libx11-data libxau6 libxcb1
  libxdmcp6 libxpm4 libxslt1.1 nginx nginx-common
  nginx-core
0 upgraded, 20 newly installed, 0 to remove and 0 not upgraded.
Need to get 2151 kB of archives.
After this operation, 8113 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Iniciamos Nginx, lo habilitamos y verificamos su estatus.

```
sudo systemctl start nginx
sudo systemctl enable nginx
sudo systemctl status nginx
```

```
root@Sololinux-demo:~# sudo systemctl start nginx
root@Sololinux-demo:~# sudo systemctl enable nginx
Synchronizing state of nginx.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable nginx
root@Sololinux-demo:~# sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2021-06-24 09:50:10 UTC; 2min 17s ago
     Docs: man:nginx(8)
   Main PID: 672 (nginx)
    Tasks: 2 (limit: 19660)
   Memory: 6.2M
   CGroup: /system.slice/nginx.service
           └─672 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
             └─673 nginx: worker process

Jun 24 09:50:10 Sololinux-demo systemd[1]: Starting A high performance web server and a reverse proxy server...
Jun 24 09:50:10 Sololinux-demo systemd[1]: Started A high performance web server and a reverse proxy server.
root@Sololinux-demo:~#
```

Ahora creamos el archivo de configuración.

```
sudo nano /etc/nginx/sites-available/ampache.conf
```

Copia y pega lo siguiente, ojo, inserta tu dominio.

```
nginxserver {
    listen 80;
    listen [::]:80;
    server_name ampache.midominio.com; #<<< dominio
    root /var/www/html/ampache;
    index index.php;

    access_log /var/log/nginx/ampache.midominio.com.access.log; #<<<
    error_log /var/log/nginx/ampache.midominio.com.error.log; #<<<
    client_max_body_size 100M;

    autoindex off;
    #Rewrite rule for Subsonic backend
    if ( !-d $request_filename ) {
        rewrite ^/rest/(.*)view$ /rest/index.php?action=$1 last;
        rewrite ^/rest/fake/(.*)$ /play/$1 last;
    }
    # Rewrite rule for Channels
    if ( !-d $request_filename ){
        rewrite ^/channel/([0-9]+)/(.*)$ /channel/index.php?
        channel=$1&target=$2 last;
    }
    #Rewrite rule for Channels
    if ( !-d $request_filename ){
        rewrite ^/channel/([0-9]+)/(.*)$ /channel/index.php?
        channel=$1&target=$2 last;
    }
    location /rest {
        limit_except GET POST {
            deny all;
        }
    }
    location ~ /bin/ {
        deny all;
        return 403;
    }
    location ~ /config/ {
        deny all;
        return 403;
    }
    location / {
        limit_except GET POST HEAD{
            deny all;
        }
    }
    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
        fastcgi_param SCRIPT_FILENAME
        $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}
```

Guarda el archivo y cierra el editor. Creamos el enlace absoluto.

```
sudo unlink /etc/nginx/sites-enabled/default
sudo ln -s /etc/nginx/sites-available/ampache.conf /etc/nginx/sites-
enabled/
sudo systemctl restart nginx
```

Instalamos properties y el repositorio universe.

```
sudo apt-get install software-properties-common
sudo add-apt-repository universe
```

Nota: Este paso no es obligatorio.

Aplicamos **Let's Encrypt SSL a Nginx**. Dependiendo de tu versión, deberás elegir un comando u otro.

```
sudo apt install -y nginx certbot python-certbot-nginx
# o
sudo apt install -y nginx certbot python3-certbot-nginx
```

Configura el certificado según tus necesidades.

```
sudo certbot --nginx -d ampache.dominio.com
```



Continuamos con la instalación de php.

```
sudo apt install php php-cli php-common php-gd php-xmllrpc php-fpm php-curl php-intl php-imagick php-mysql php-zip php-xml php-mbstring php-bcmath -y
```

Abrimos con un editor su archivo de configuración.

```
sudo nano /etc/php/7.4/fpm/php.ini
```

Busca las líneas indicadas a continuación y, modifícas los valores tal como en el ejemplo.

```
file_uploads = On
allow_url_fopen = On
short_open_tag = On
memory_limit = 256M
cgi.fix_pathinfo = 0
upload_max_filesize = 250M #Puedes modificar este valor si tienes
archivos muy grandes
max_execution_time = 360
```

Bien... es evidente que nos falta MySQL.

```
sudo apt install mysql-server
```

```
root@SoloLinux-demo:~# sudo apt install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl
  libevent-core-2.1-7 libevent-pthreads-2.1-7 libfcgi-perl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl
  liblwp-mediatypes-perl libmecab2 libnuma1 libtimedate-perl liburi-
  perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0
  mysql-client-core-8.0 mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libwww-perl mailx tinyca
The following NEW packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl
  libevent-core-2.1-7 libevent-pthreads-2.1-7 libfcgi-perl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl
  liblwp-mediatypes-perl libmecab2 libnuma1 libtimedate-perl liburi-
  perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0
  mysql-client-core-8.0 mysql-server mysql-server-8.0 mysql-server-
  core-8.0
0 upgraded, 26 newly installed, 0 to remove and 0 not upgraded.
Need to get 31.3 MB of archives.
After this operation, 259 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Asegura la instalación.

```
mysql_secure_installation
```

Es necesario crear una base de datos y su usuario.

```
sudo mysql
```

```
CREATE DATABASE ampache_db;
```

Ahora el usuario **«ampache»**. No te olvides de insertar un password.

```
CREATE USER 'ampache'@'localhost' IDENTIFIED WITH
mysql_native_password BY 'password';
GRANT ALL PRIVILEGES ON ampache_db.* TO 'ampache'@'localhost' WITH
GRANT OPTION;
FLUSH PRIVILEGES;
```

Una vez termines la configuración, escribe **«exit»** para salir de la **consola MySQL**.

```
mysql> exit
Bye
root@SoloLinux-demo:~#
```

Ya tenemos nuestro servidor preparado, llega la hora de instalar Ampache.

```
wget
https://github.com/ampache/ampache/releases/download/4.4.3/ampache-
4.4.3_all.zip
```

Ahora sigue todos los pasos que te indico.

```
unzip ampache-4.4.3_all.zip -d /var/www/html/ampache
```

```
sudo chown -R www-data:www-data /var/www/html/ampache
sudo chown -R 755 /var/www/html/ampache
```

Creamos el directorio donde cargaremos la música y, le concedemos permisos.

```
sudo mkdir -p /data/mi_musica
```

```
sudo chown -R www-data:www-data /data/mi_musica
```

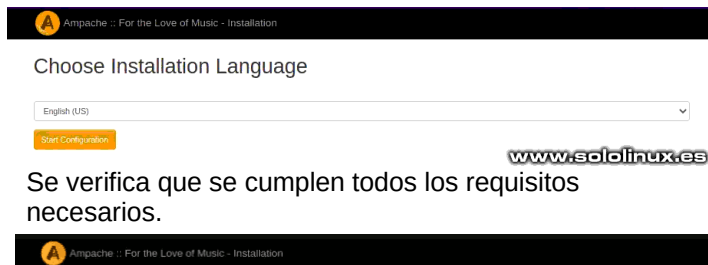
Ampache usa **ffmpeg** para convertir archivos, lo instalamos.

```
sudo apt install ffmpeg
```

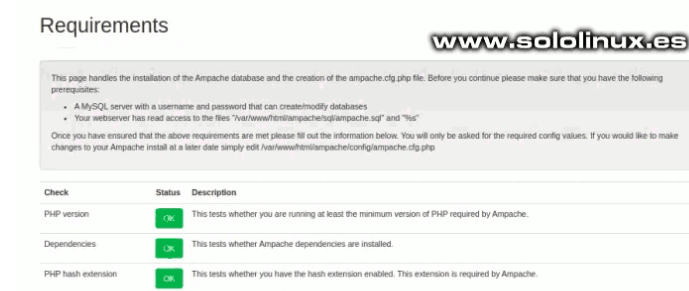
Para concluir la instalación, reiniciamos nuestro servidor **Ampache**.

```
sudo reboot
```

Una vez reinicie el sistema, tan solo tienes que insertar tu dominio elegido en el navegador web. Te aparece la ventana de selección de idioma...



Se verifica que se cumplen todos los requisitos necesarios.



Por último... inserta los datos que te solicita y continua. Al concluir todo el proceso, ya tendrás listo tu servidor de música **Ampache**, haces click en volver a la página inicial para acceder a la herramienta.



Ampache :: For the Love of Music - Installation

Install Progress

www.sololinux.es

33%

Step 1 - Create the Ampache database

This step creates and inserts the Ampache database, so please provide a MySQL account with database creation rights. This may take some time on slower computers.

Step 2 - Create configuration files (ampache.cfg.php ...)

Step 3 - Set up the initial account

Insert Ampache Database

Desired Database Name

MySQL Hostname

MySQL Port (optional)

MySQL Administrative Username

MySQL Administrative Password

Create Database

☐

MySQL 8 host? [Oracle MySQL FAQ](#)

Create Tables (ampache.sql)

☒

Create Database User

☐

Skip

Insert Database

AYUDANOS A SEGUIR
CRECIENDO

PayPal **PayPal**

Donar a Revistalinux

Donar a SoloLinux

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

Chat de SoloLinux en Telegram



designed by freepik

Whatsapp en linux con Whatsdesk – Junio 2021

Hace ya un tiempo que no actualizaba mi Whatsdesk. Si no conoces esta aplicación, estamos hablando de una herramienta **electron** desarrollada y mantenida por **Gustavo González**, que instala lo que se conoce como «Unofficial whatsapp client for linux».

Puedes pensar que no es necesaria, que te conectas o creas una aplicación que ataque al **navegador web** y listo. Vale... pues te digo que no, Whatsdesk no necesita que abras el navegador para nada, además de ser muy ligera y contener casi las mismas opciones de configuración junto con herramientas varias.

La herramienta dio sus primeros pasos poco a poco, pero mira por donde que en la actualidad... **Canonical** la tiene en su **tienda de aplicaciones snap**, así que con eso está todo dicho.



En [sololinux](https://sololinux.es) no usamos **snap**, preferimos instalar paquetes de forma manual. En el artículo de hoy, insertamos los enlaces directos para descargar las versiones basadas en deb y rpm. Por favor no abuses de ellos, puedes entorpecer la descarga de otros usuarios.



Whatsapp en linux con Whatsdesk – Junio 2021

Antes de nada quiero lanzar una crítica al aire. Señores de **Facebook**, Whatsapp o a quien corresponda, ya vale con la broma, su broma, su pXXX broma ya dura demasiados años. ¿Por qué no existe ninguna versión de su aplicación para escritorio, pero si para los de las ventanas y para Mac?. Es lamentable que además cierran su código, lamentable.

Bueno... mejor dejamos el tema y vemos las descargas directas que es lo que realmente nos interesa. Selecciona la que corresponda a tu **distribución linux**.

Ubuntu, Linux Mint, etc:

- [Whatsdesk 0.3.6_amd64.deb](#)
- [Whatsdesk 0.3.6_i386.deb](#)

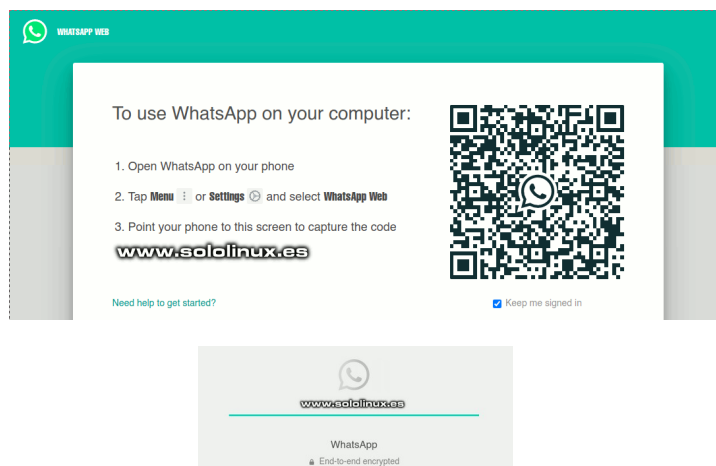
Fedora, OpenSuse, etc:

- [Whatsdesk 0.36.6_0.3.6.x86_64.rpm](#)

Si prefieres la instalación con **paquete snap**.

```
sudo snap install whatsdesk
```

Al iniciar por primera vez te solicitará que escanees con tu smartphone. Igual que con la aplicación oficial o la vía web.



Nota del autor: Soy consciente que cada día que pasa, más rechazo produce **Whatsapp**. Pero seamos sinceros... su uso es necesario ya sea por cuestiones laborales, o familiares. Es el más extendido y eso no tiene vuelta atrás, en mi caso... principalmente lo utilizo para hablar con mis seres queridos (que están a muchos kilómetros), para temas laborales cada día más **Telegram** o **Viber**.

Iniciar Ubuntu en modo texto



El uso del modo texto, es algo clásico entre administradores de sistemas y desarrolladores. Muchos de ellos, están días y días sin necesitar el escritorio, por tanto... es evidente que ahorraremos tiempo y recursos del sistema al iniciar directamente en modo texto, también conocido como **modo consola** o de **línea de comandos**.

En un [artículo](#) anterior, vimos como **iniciar linux en modo consola en sistemas basados en systemd**, hoy lo haremos desde el **Grub**. En nuestro caso, realizamos el proceso sobre un Ubuntu 20.04, pero es muy similar en otras **distribuciones linux**.



Iniciar Ubuntu en modo texto

Lo único que tenemos que hacer es... realizar una serie de modificaciones en el archivo de **configuración del Grub**. Son varios pasos, pero muy sencillos, sigue las indicaciones del artículo y no tendrás problemas.

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

Nos aparece el **Grub**.

```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=0
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command `vbeinfo`
#GRUB_GFXMODE=640x480

# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
#GRUB_DISABLE_RECOVERY="true"
```

Ahora observa las líneas indicadas.

```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'
GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=hidden <-----
GRUB_TIMEOUT=0
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash" <-----
GRUB_CMDLINE_LINUX="" <-----
# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that
obtains
# the memory map information from GRUB (GNU Mach, kernel of
FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"
# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console <-----
# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports
via VBE
# you can see them in real GRUB with the command `vbeinfo`
#GRUB_GFXMODE=640x480
```

Ahora editamos las siguientes líneas.

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
# Comentamos la línea, ejemplo...
#GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
```

```
GRUB_CMDLINE_LINUX=""
# Agregamos text entre comillas, ejemplo...
GRUB_CMDLINE_LINUX="text"
```

```
#GRUB_TERMINAL=console
# Descomentamos la línea, ejemplo...
GRUB_TERMINAL=console
```

Ejemplo del resultado.

```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'
GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=hidden <-----
GRUB_TIMEOUT=0
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash" <-----
GRUB_CMDLINE_LINUX="text" <-----
# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"
# Uncomment to disable graphical terminal (grub-pc only)
GRUB_TERMINAL=console <-----
# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command `vbeinfo`
#GRUB_GFXMODE=640x480
```

```
GRUB_TIMEOUT=0
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
#GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX="text"
# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"
# Uncomment to disable graphical terminal (grub-pc only)
GRUB_TERMINAL=console
# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command `vbeinfo`
#GRUB_GFXMODE=640x480
```

Ahora solo nos falta **actualizar el Grub** y, reiniciar el sistema.

```
sudo update-grub
# 0
sudo update-grub2
```

```
sudo reboot
```

Al reiniciar el sistema deberías acceder directamente a la interfaz cli, si por algún motivo no te funciona este método, sigue los pasos de [este anterior artículo](#).

Reinstalar el escritorio de Ubuntu



Te levantas por la mañana y enciendes tu máquina, vaya... algo sucede que se produce error al cargar el **entorno de escritorio**, no inicia. Esta situación descrita es algo común en usuarios noveles, que a veces borran o manipulan lo que no deben. Bueno, tranquilo que así se aprende.

Reparar o restaurar los archivos borrados o dañados no es tarea sencilla, aún menos para usuarios que están comenzando en el **mundo Linux**. Lo más rápido y menos complejo, es reinstalar de nuevo tu entorno de escritorio. En el artículo de hoy, aprenderemos a reinstalar el escritorio de Ubuntu, en caso de error fatal.



Reinstalar el escritorio de Ubuntu

Para realizar esta operación, es evidente que necesitas acceso a la **terminal**, bien y como accedo te preguntaras, jeje. Vale, pues muy fácil... justo en el momento es que tú consideres que debería empezar a cargar el escritorio, pulsas la combinación de teclas **Ctrl + Alt + F2**. Aparece la **shell** de Ubuntu, introduce tu usuario y contraseña.

Todas las operaciones a realizar, requieren **permisos sudo**. Para ir más rápido iniciamos como sudo directamente.

```
sudo su
```

Ahora configuramos todos los paquetes.

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

Si es necesario, se reinstalaran los paquetes configurados en el paso anterior. En nuestro caso usamos **«apt-get»**, simplemente porque es mucho más completo que **«apt»**, que puede producir errores.

```
apt-get -f install
```

Una vez terminé el proceso anterior, actualizamos.

```
apt-get update
apt-get dist-upgrade
```

```
Desempaquetando libgcc-s1:amd64 (10.3.0-1ubuntu1-20.04) sobre (10.2.0-5ubuntu1-20.04) ...
Configurando libgcc-s1:amd64 (10.3.0-1ubuntu1-20.04) ...
(Leyendo la base de datos ... 125172 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../libcc1-0_10.3.0-1ubuntu1-20.04_amd64.deb ...
Desempaquetando libcc1-0:amd64 (10.3.0-1ubuntu1-20.04) sobre (10.2.0-5ubuntu1-20.04) ...
Preparando para desempaquetar .../libstdc++6_10.3.0-1ubuntu1-20.04_amd64.deb ...
Desempaquetando libstdc++6:amd64 (10.3.0-1ubuntu1-20.04) sobre (10.2.0-5ubuntu1-20.04) ...
Configurando libstdc++6:amd64 (10.3.0-1ubuntu1-20.04) ...
(Leyendo la base de datos ... 125172 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../libz4_1.9.2-2ubuntu0.20.04.1_amd64.deb ...
Desempaquetando libz4 (1.9.2-2ubuntu0.20.04.1) sobre (1.9.2-2) ...
Preparando para desempaquetar .../liblz4-1_1.9.2-2ubuntu0.20.04.1_amd64.deb ...
Desempaquetando liblz4-1:amd64 (1.9.2-2ubuntu0.20.04.1) sobre (1.9.2-2) ...
Configurando liblz4-1:amd64 (1.9.2-2ubuntu0.20.04.1) ...
www.sololinux.es
```

Bueno, llego a la hora de la verdad, vamos a reinstalar el entorno de escritorio por defecto de Ubuntu. Como tenemos nuestra máquina en modo consola, este proceso es rapido.

```
apt-get install --reinstall ubuntu-desktop
```

Una vez concluya todo el proceso, borramos los paquetes no necesarios y lo que hemos descargado.

```
apt-get autoremove
apt-get clean
```

Reiniciamos...

```
reboot
```

Listo, Ubuntu ha recuperado su escritorio original.



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

[Chat de SoloLinux en Telegram](#)



designed by freepik

Comprimir archivos con ZStandard en Linux



Probablemente estés familiarizado con herramientas de compresión como **gzip**, **bzip** o **xz**. Pero talvez desconozcas que en 2015 un desarrollador de **Facebook** lanzó **ZStandard**, una herramienta más conocida como **zstd**, que logra una alta tasa de compresión en un tiempo menor al de la mayoría de sus competidores.

Reconozco y me incluyo yo mismo, que en el **mundo Linux** nos cuesta mucho adoptar oficialmente nuevos estándares (para que tocar lo que funciona bien). Aun siendo así, Zstandard está siendo adoptado por grandes empresas para la compresión de enormes cantidades de datos, además de recibir un apoyo total por parte de distribuciones como **Arch Linux**, que cambió la compresión de sus paquetes **xz** a Zstandard.



Comprimir archivos con ZStandard en Linux

En una prueba obtenida sobre la compresión de un archivo de 1350 Megabytes, con respecto a otros algoritmos se han obtenido los siguientes resultados.

Tipo de compresión	Tamaño de la salida	Tiempo de proceso
zstd	934 MB	43 segundos
zip	956 MB	85 segundos
gzip	955 MB	87 segundos
bzip2/bz2	943 MB	322 segundos
rar	877 MB	382 segundos
7zip	851 MB	635 segundos
xz	856 MB	964 segundos

Como puedes observar, la velocidad de la compresión es muy alta. Precisamente ese es su punto fuerte y, el que nosotros necesitamos. Lamentablemente, no todas las **distribuciones Linux** lo tienen incluido, pero sí que lo podemos encontrar en sus repositorios oficiales. Vamos a instalar ZStandard.

En Debian, Ubuntu, Mint y derivados:

```
sudo apt install zstd
```

En Fedora, CentOS, Alma y derivados:

```
sudo dnf install prename
```

En Arch Linux, Manjaro y derivados:

```
sudo pacman -S zstd
```

Vemos un ejemplo de instalación en Ubuntu 18.04.

```
root@sololinux:/home/sergio# cd
root@sololinux:~#
root@sololinux:~# sudo apt install zstd
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
zstd
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 278 kB de archivos.
Se utilizarán 1.141 kB de espacio de disco adicional después de esta operación.
Des:1 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 zstd amd64
1.3.3+dfsg-2ubuntu1.2 [278 kB]
Descargados 278 kB en 1s (356 kB/s)
Seleccionando el paquete zstd previamente no seleccionado.
(Leyendo la base de datos ... 376296 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../zstd_1.3.3+dfsg-2ubuntu1.2_amd64.deb ...
Desempaquetando zstd (1.3.3+dfsg-2ubuntu1.2) ...
Configurando zstd (1.3.3+dfsg-2ubuntu1.2) ...
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...
root@sololinux-demo:~#
```

El uso de la herramienta es muy simple, observa la compresión del archivo «libera.sh».

```
zstd libera.sh
```

```
sergio@sololinux:~$ zstd libera.sh
libera.sh : 29.96% ( 1986 => 595 bytes, libera.sh.zst)
sergio@sololinux:~$
```

Para descomprimir agregamos la «**opción -d**». No olvides que la extensión final es **zst**.

```
zstd -d libera.sh.zst
```

Verás algo similar a...

```
sergio@sololinux:~$ zstd -d libera.sh.zst
zstd: libera.sh already exists; overwrite (y/N) ? y
libera.sh.zst : 1986 bytes
sergio@sololinux:~$
```

Esta aplicación dispone de muchas opciones, que tal vez te puedan resultar de utilidad.

```
zstd -h
```

Canales de Telegram:
[Canal SoloLinux](#)
[Canal SoloWordPress](#)

[Chat de SoloLinux en Telegram](#)



designed by freepik

Opciones de zstd:

```
sergio@sololinux:~$ zstd -h
*** zstd command line interface 64-bits v1.3.3, by Yann Collet ***
Usage :
    zstd [args] [FILE(s)] [-o file]
FILE    : a filename
           with no FILE, or when FILE is -, read standard input
Arguments :
  -#      : # compression level (1-19, default:3)
  -d      : decompression
  -D file: use 'file' as Dictionary
  -o file: result stored into 'file' (only if 1 input file)
  -f      : overwrite output without prompting and (de)compress links
  --rm    : remove source file(s) after successful de/compression
  -k      : preserve source file(s) (default)
  -h/-H   : display help/long help and exit
Advanced arguments :
  -V      : display Version number and exit
  -v      : verbose mode; specify multiple times to increase verbosity
  -q      : suppress warnings; specify twice to suppress errors too
  -c      : force write to standard output, even if it is the console
  -l      : print information about zstd compressed files
  --ultra : enable levels beyond 19, up to 22 (requires more memory)
  --long[=#] : enable long distance matching with given window log (default : 27)
  -T#     : use # threads for compression (default:1)
  -B#     : select size of each job (default:0==automatic)
  --no-dictID : don't write dictID into header (dictionary compression)
  --[no-]check : integrity check (default:enabled)
  -r      : operate recursively on directories
  --format=gzip : compress files to the .gz format
  --test    : test compressed file integrity
  --[no-]sparse : sparse mode (default:enabled on file, disabled on stdout)
  -M#     : Set a memory usage limit for decompression
  --       : All arguments after "--" are treated as files
Dictionary builder :
  --train ## : create a dictionary from a training set of files
  --train-cover[=k=#,d=#,steps=#] : use the cover algorithm with optional args
  --train-legacy[=s=#] : use the legacy algorithm with selectivity (default: 9)
  -o file : 'file' is dictionary name (default: dictionary)
  --maxdict=# : limit dictionary to specified size (default : 112640)
  --dictID=# : force dictionary ID to specified value (default: random)
Benchmark arguments :
  -b#     : benchmark file(s), using # compression level (default : 1)
  -e#     : test all compression levels from -bX to # (default: 1)
  -i#     : minimum evaluation time in seconds (default : 3s)
  -B#     : cut file into independent blocks of size # (default: no block)
  --priority=rt : set process priority to real-time
sergio@sololinux:~$
```

AYUDANOS A SEGUIR
CRECIENDO

PayPal **PayPal**

Donar a Revistalinux

Donar a SoloLinux

Canales de Telegram:
Canal SoloLinux
Canal SoloWordpress

Chat de SoloLinux en Telegram



designed by freepik

5 razones para volver a linux

Distribuciones basadas en el **kernel Linux**, existen desde hace muchos años. Es posible que alguna vez te hayas propuesto migrar a **Linux**, pero al final dada su complejidad has decidido desistir en la idea y, continuar con el sistema que usas en la actualidad.

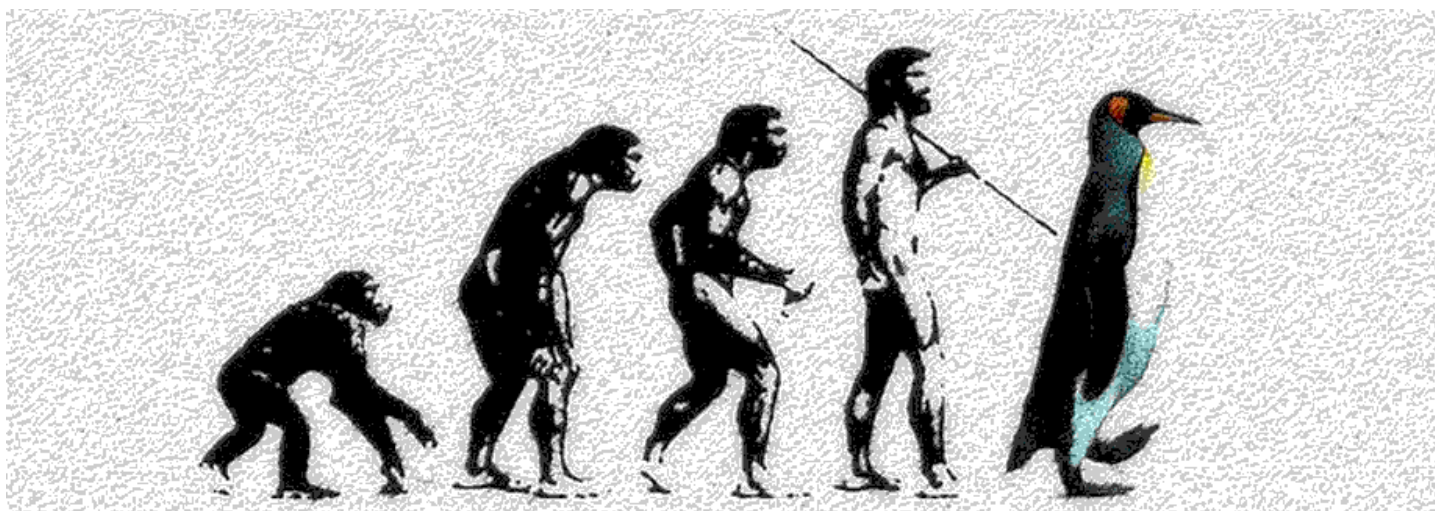
Durante muchos años, un sector de los desarrolladores y usuarios avanzados considerados como puristas, se negaban en rotundidad a facilitar el **acceso a Linux** a los usuarios que solo necesitaban un sistema que les ofreciera algo simple, útil, y adaptado a sus necesidades sin apenas curva de aprendizaje a la hora de migrar.

Te puedo asegurar que en algunas **comunidades linux**, hubo auténticas guerras dialécticas entre los puristas y, los que creían que la única forma de crecer, era **acercar Linux** a nuevos usuarios que mamaban **Windows** en sus casas y escuelas. Esto produjo un bum para Linux, se produjeron muchas excisiones y de ellas nacieron muchas de las fork que actualmente tienes a tu disposición. El **sistema Linux** dio un vuelco, que continúa tal como lo conocemos hoy en día. Es apto para todos los usuarios, evidentemente todo depende de la distribución elegida.



5 razones para volver a linux

Este artículo de hoy intenta motivar, que no convencer a aquellos que probaron hace 15 o 20 años linux y, lo dejaron por imposible al no obtener ayudas en su lenguaje nativo. Por suerte todo ha cambiado, bienvenido a **sololinux** amigo.



Linux y su facilidad de uso

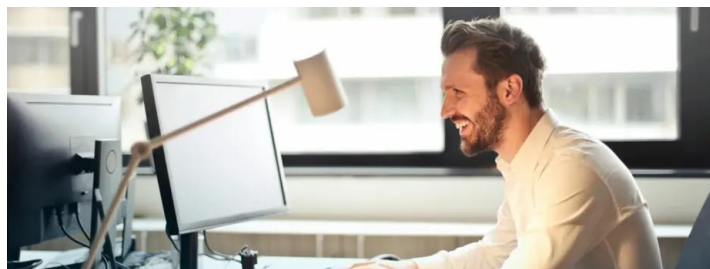
A lo largo de la historia, una de las principales quejas contra Linux era su complejidad de uso y, falta de intuición en general. Muchos desarrolladores han trabajado durante mucho tiempo, para solucionar estos problemas. En estos momentos podemos afirmar, que Linux se encuentra ahora mismo en otra dimensión si lo comparamos con tiempos pasados.

Está claro, que todo esto depende de la distribución específica que quieras usar. Pero para aquellos que se decanten por alguna muy popular como puede ser Ubuntu, les puedo adelantar que las mejoras y cambios son grandiosas respecto a las versiones más vetustas.

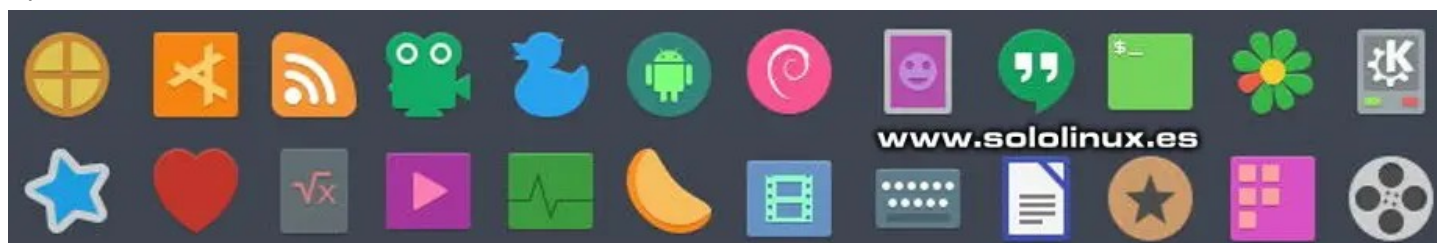
Atrás quedaron los días en los que tenías que buscar información en ingles, para pelear después con la línea de comandos. En estos momentos, puedes resolver la mayoría de los problemas surgidos desde una interfaz gráfica. Además, las **distribuciones Linux** actuales cuentan con muchas herramientas que te facilitaran la vida.

Aplicaciones con soporte para Linux

Si hace tiempo que probaste **Linux** por última vez, es posible que también se sorprenda la evolución de las aplicaciones nativas y, las que se han ido acercando a nuestro sistema del pingüino. Cada vez más aplicaciones ofrecen soporte para Linux, incluyendo algunas que parecía un imposible



Las principales plataformas de juegos, como **Steam**, también se mueven hacia linux. Los emuladores también han mejorado muchísimo, incluso puedes ejecutar aplicaciones que no tienen soporte nativo sin ningún problema. Solo quedan por llegar herramientas especializadas, que dependen en gran medida de la aceleración de gráficos. Paso a paso, cada día falta menos.



Uso de Linux a diario

Puedes utilizar Linux en tus tareas o entretenimientos habituales. No te olvides que es compatible con casi todos los navegadores web, además de con la misma fluidez que en **Windows**, o incluso más. En muchos casos, encontrarás las mismas aplicaciones o similares en Linux sin tener que usar un emulador como **Wine**.

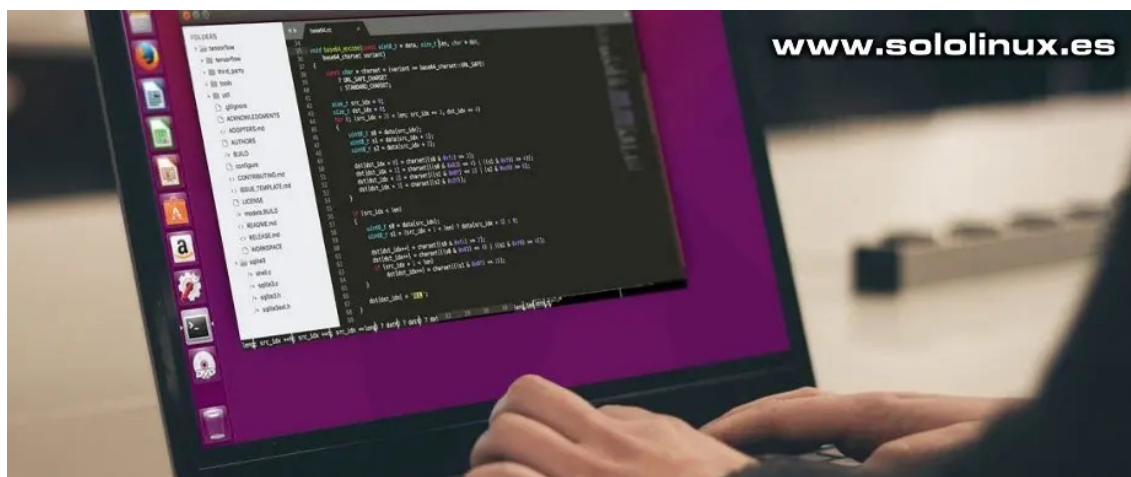
Crear, modificar y organizar tus archivos y documentos personales es sencillo, además puedes aprovechar la excelente integración nativa con la mayoría de plataformas de alojamiento en la nube. La transición de Windows a Linux es muy fácil, si hablamos de tus tareas diarias.

Trabajar con Linux

Si su trabajo diario trata sobre la edición de documentos, envío de correos electrónicos, etc (trabajo ofimático), Linux te sorprenderá con montones de herramientas y aplicaciones a tu alcance. En tiempos atrás, existía cierta incompatibilidad entre **Microsoft Office** y **LibreOffice (OpenOffice)**, hoy en día esa barrera ya está superada, **LibreOffice** es muy bueno, además de gratuito.

Por otro lado tenemos los medios audiovisuales, que dependiendo de tu labor específica Linux es espléndido. Si editas videos o grandes paquetes de imágenes, con Linux es mucho más fácil agilizar tu trabajo y automatizar partes del mismo usando scripts. Windows también ofrece este soporte, pero lo de Linux está a otro nivel.

Si te dedicas a programar o desarrollar aplicaciones, sitios web, etc., no te lo pienses más, Linux es para ti. Toneladas de herramientas y lenguajes de programación, harán de tu trabajo una pasión.



Jugar en Linux

Soy consciente de que esta es la razón principal, por la que muchos usuarios se alejan de Linux. Debes saber que todo está cambiando y, que muchos desarrolladores de juegos se han dado cuenta de que existe un gran mercado sin explotar, entre los usuarios de Linux. Cada día son más los que introducen sus productos en el **mundo de Linux**

Steam nos comunica, que ya disponen de cinco mil juegos nativos para Linux en su plataforma online, y la lista sigue aumentando a pasos agigantados. No te preocupes, en Linux podrás disfrutar de tus juegos preferidos, sin importar tus preferencias.

Nota del autor

Linux tiene una particularidad si lo comparamos con Windows, se ofrecen tantos «sabores» y distribuciones diferentes, que es difícil no encontrar una que se ajuste a tus necesidades. Algunas distribuciones están diseñadas para el trabajo diario en general, otras están dirigidas a expertos en seguridad, también podemos buscar las específicas para multimedia. En Linux tenemos de todo y para todos.

Migrar a Linux, es como cuando te mudas a otra vivienda más grande, debes acostumbrarte así que no desespere y dale una oportunidad, ten en cuenta que tu nueva casa tiene jardín y piscina, la anterior no.

No te arrepentirás de dedicar un poco de tu tiempo a aprender el sistema, cuanto más sepas más querrás saber. Como último apunte, te repito que es muy importante que busques y elijas la distribución Linux que más se adapte a tus necesidades y conocimientos, no te arrepentirás.

www.sololinux.es

AYUDANOS A SEGUIR
CRECIENDO

PayPal **PayPal**

Donar a Revistalinux

Donar a SoloLinux

Canales de Telegram:
Canal SoloLinux
Canal SoloWordpress

Chat de SoloLinux en Telegram



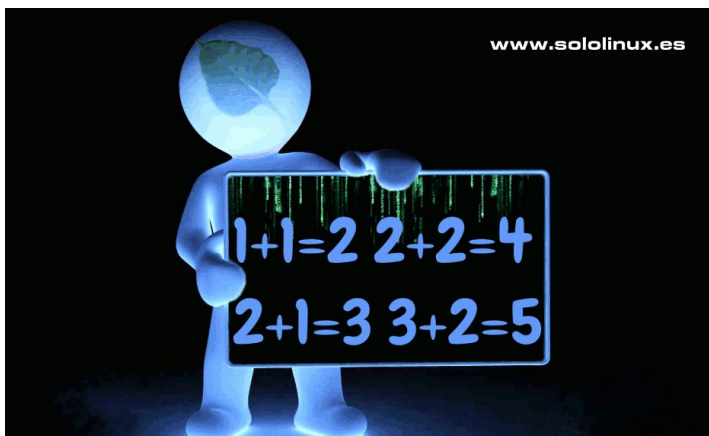
designed by freepik

8 formas de contar el número de líneas



Es algo típico de **Linux**, que podamos hacer una misma tarea de diferentes formas. Esto puede parecer una simple anécdota, pero te equivocas... pues cada herramienta o comando aún obteniendo el mismo resultado, pueden tener un fin totalmente diferente.

8 formas de contar el número de líneas



Para ver como contar el número de líneas, tomamos como ejemplo un **script bash** específico para limpiar servidores Apache, que tiene 42 líneas. Lo vemos.

```
#!/bin/bash
# by Sergio G.B. - SoloLinux.es
# scripts bash imprescindibles de un sysadmin linux
#
echo "<----->"
echo "<----->"
echo "Comprobando estado de memoria"
free
echo "OK - Comprobacion completada"
echo "<----->"
sleep 2s; echo "Limpieza de memoria cache y swap";
echo "<----->"
sleep 2s; echo "Deshabilitando HTTP"
killall -KILL httpd
echo "OK - Http deshabilitado"
echo "<----->"
sleep 5s; echo "Deshabilitando Swap"
swapoff -a
echo "OK - Swap deshabilitado"
echo "<----->"
sleep 5s; echo "Liberando page-caches, dentries e inodes"
sync;syncctl -w vm.drop_caches=3;sync
echo "OK - Server liberado"
echo "<----->"
sleep 2s; echo "Habilitando la Swap"
swapon -a
echo "OK - Swap habilitado"
echo "<----->"
sleep 2s; echo "Habilitando HTTP"
service httpd start
echo "OK - Http habilitado"
echo "<----->"
sleep 2s; free
echo "....MANTENIMIENTO CONCLUIDO...."
echo "<----->"
echo "<----->"
```

Al script lo hemos llamado «limpiar.sh», así que vamos a ver los ejemplos de uso.

Comando wc -l

```
wc -l limpiar.sh
```

En este caso, la salida nos indica el número de líneas y el nombre del archivo.

```
sergio@sololinux:~$ wc -l limpiar.sh
42 limpiar.sh
sergio@sololinux:~$
```

Comando wc -l <

```
wc -l < limpiar.sh
```

Tan solo el número de líneas del archivo.

```
sergio@sololinux:~$ wc -l < limpiar.sh
42
sergio@sololinux:~$
```

Comando cat archivo | wc -l

```
cat limpiar.sh | wc -l
```

Está claro que con el comando **cat** podemos hacer de todo.

```
sergio@sololinux:~$ cat limpiar.sh | wc -l
42
sergio@sololinux:~$
```

Comando awk 'END {print NR}' archivo

```
awk 'END {print NR}' limpiar.sh
```

```
sergio@sololinux:~$ awk 'END {print NR}' limpiar.sh
43
sergio@sololinux:~$
```

Comando sed -n '\$=' archivo

```
sed -n '$=' limpiar.sh
```

El comando **sed** también nos ayuda a lograr nuestro objetivo.

```
sergio@sololinux:~$ sed -n '$=' limpiar.sh
43
sergio@sololinux:~$
```

Comando grep -e «\$» -c archivo

```
grep -e "$" -c limpiar.sh
```

```
sergio@sololinux:~$ grep -e "$" -c limpiar.sh
43
sergio@sololinux:~$
```

Comando grep -e «^» -c archivo

```
grep -e "^" -c limpiar.sh
```

Otra opción del comando **grep**.

```
sergio@sololinux:~$ grep -e "^" -c limpiar.sh
43
sergio@sololinux:~$
```

Comando nl archivo | tail -n1

```
grep -e "^" -c limpiar.sh
```

Este comando nos indica el número total de líneas, incluyendo los espacios. También nos imprime los caracteres de la última línea.

```
sergio@sololinux:~$ nl limpiar.sh | tail -n1
36 echo "<----->"
sergio@sololinux:~$
```

Nota del autor: Podrás observar que algunos comandos pueden variar el resultado, este efecto es debido a la forma de tratar las líneas y los espacios en blanco.

Matar procesos con kill, pkill o killall

Los comandos **kill**, **pkill** y **killall** se utilizan para terminar o matar procesos en un sistema Linux. Esta serie de utilidades en línea de comandos, vienen de forma predeterminada en cualquier **distribución Linux**

Los tres comandos mencionados hacen prácticamente lo mismo, pero digamos que mediante diferentes formas. Dominar estas herramientas es indispensable, para cualquier **usuario de Linux** Nos permiten tener un control total sobre nuestro sistema, a la hora de manejar lo que debe estar funcionando y lo que no.

```
www.sololinux.es [~]$ killall
```

Matar procesos con kill, pkill o killall

Cuando ejecutas cualquier comando kill, lo que haces es enviar una señal al proceso indicándole lo que tiene que hacer. Puedes conocer todas las señales disponibles con el siguiente comando.

```
kill -l
```

Señales.

```
sergio@sololinux:~$ kill -l
1) SIGHUP 2) SIGINT 3) SIGQUIT 4) SIGILL 5) SIGTRAP
6) SIGABRT 7) SIGBUS 8) SIGFPE 9) SIGKILL 10) SIGUSR1
11) SIGSEGV 12) SIGUSR2 13) SIGPIPE 14) SIGALRM 15) SIGTERM
16) SIGSTKFLT 17) SIGCHLD 18) SIGCONT 19) SIGSTOP 20) SIGTSTP
21) SIGTTIN 22) SIGTTOU 23) SIGURG 24) SIGXCPU 25) SIGXFSZ
26) SIGTALRM 27) SIGPROF 28) SIGWINCH 29) SIGIO 30) SIGPWR
31) SIGSYS 34) SIGRTMIN 35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47)
SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7
58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX
sergio@sololinux:~$
```

La mejor forma de matar procesos en linux, es atacar directamente al/los ID de proceso. Identificar el número de proceso es tarea sencilla, para ello usamos el **comando pidof** seguido de la aplicación o herramienta a identificar. Por ejemplo...

```
pidof thunderbird
# 0
pidof firefox
```

Thunderbird solo tiene un proceso a abierto, pero **Firefox** varios.

```
sergio@sololinux:~$ pidof thunderbird
11838
sergio@sololinux:~$ www.sololinux.es
sergio@sololinux:~$
sergio@sololinux:~$ pidof firefox
12281 12255 12233 12193 12119 12090 12064 11997
sergio@sololinux:~$
```

Comando kill

Ahora matamos Thunderbird con la herramienta kill.

```
kill 11838
```



También puedes matar varios procesos a la vez.

```
kill 12281 12255 12233 12193 12119 12090 12064 11997
```

Es posible que la herramienta esté bloqueada y no se cierre con kill, agregando la **señal 9** seguros que lo mata.

```
kill -9 12281 12255 12233 12193 12119 12090 12064 11997
```

Comando pkill

El **comando pkill** tiene la misma función que **kill**, pero con una salvedad que puede parecer pequeña, pero que es muy importante. Con **pkill** puedes matar procesos sin introducir su ID, con el nombre o parte de él... es suficiente. Esto no siempre es bueno, ya que puedes matar procesos que no querías matar.

Imagínate que tienes los navegadores Chrome y Chromium abiertos, si escribes...

```
pkill chrome
```

Se cierra Chrome, pero si tecleas solo una parte de los caracteres se cerraran los dos navegadores, dado que existen coincidencias entre sus nombres.

```
pkill chro
```

Para matar lo que coincida al 100%, usa la opción -x.

```
pkill -x thunderbird
```

Pkill admite las señales del comando **kill**.

```
pkill -9 chrome
```



Comando killall

Este comando es similar a **kill**, pero les separa una diferencias sustancial. **Killall** requiere del nombre completo del proceso. Por ejemplo...

```
killall chromium
```

Puedes añadir las señales de kill.

```
killall -9 chromium
```

Es interesante el solicitar confirmación con la **opción -i**.

```
killall -i thunderbird
```

```
sergio@sololinux:~$  
sergio@sololinux:~$ killall -i thunderbird  
Matar thunderbird(11838) ? (y/N) y  
sergio@sololinux:~$
```

Manuales

Para ver los manuales de los tres comandos...

```
# kill  
man kill  
  
# pkill  
man pkill  
  
# killall  
man killall
```

www.sololinux.es

AYUDANOS A SEGUIR
CRECIENDO

PayPal **PayPal**

Donar a Revistalinux

Donar a SoloLinux

Canales de Telegram:
Canal SoloLinux
Canal SoloWordpress

Chat de SoloLinux en Telegram



designed by freepik

Curso GRATIS



Linux De 0 a root!



Victor S. Recio
Cloud Solution Architect at Microsoft

Jorge Ortego
Administrador de Sistemas at Banreservas



Jonathan Montero
Linux Engineer at Geophy



Gilberto Perez
CTO at PBO

Linux.d

La Comunidad de **Linux Dominicana** les invita a participar en un curso de básico a temas avanzados de **Linux**, impartidos en 20 semanas de 20 videos

Fecha de Inicio:
Julio 1 2021

Inscríbete a través de nuestro canal de Telegram:
<https://t.me/linuxdoescuela>



AYUDANOS A SEGUIR
CRECIENDO

PayPal **PayPal**

Donar a Revistalinux

Donar a SoloLinux

Canales de Telegram:
Canal SoloLinux
Canal SoloWordpress

Chat de SoloLinux en Telegram



designed by freepik

Publicidad:

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

CON SOLO LINUX
MULTIPLICARAS TUS
CLIENTES

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



SoloWordpress

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



Opinión de Jose Luis desde Cuba.

Al colectivo de la **Revista Sololinux**:

Mi nombre es José Luis y soy docente universitario en la carrera de turismo en la Universidad de Sancti Spíritus, Cuba.

Soy usuario de **GNU/Linux** hace solo seis meses. Aunque en los 20 años que trabajé con Windows utilicé mucho software libre mi salto al sistema operativo ocurrió como ya dije hace muy poco.

En Windows sabía hacer casi de todo. No había tarea demasiado difícil si ponía mi empeño. A pesar de no ser desarrollador ni **Sysadmin**, me consideraba un usuario por encima de la media. Quizá por eso demoré tanto mi salto a GNU/Linux. Pero, realmente si de algo me arrepiento es de no haberlo hecho antes, y si de algo estoy seguro es que no volveré nunca atrás.

Me gustaría agradecer enormemente a esta revista, gracias a la cual mi aprendizaje en GNU/Linux ha sido vertiginoso, según mi consideración. Desde que un amigo me compartiera la revista he leído y descargado todos sus números y sigo de cerca el sitio **sololinux.es** y todas las noticias y tutoriales que ahí se publican.

La revista **Sololinux** me ayudado en diferentes sentidos. Como por ejemplo:

- Me ha ayudado a crear un criterio para valorar tanto las distros como el software que más me conviene utilizar de acuerdo a mis necesidades que pueden ser bien distintas a las de otros pero los criterios para escoger pueden funcionar para todos.
- También gracias a esta revista he podido ejecutar múltiples tutoriales los cuales siempre funcionan que es mucho decir de por sí, ya que no todos los que encuentras en internet lo hacen, por lo menos no de la mejor manera. Entre estos se encuentran; instalar la última versión de **Libreoffice**, mejorar la calidad de sonido del sistema con **pulseaudio**, que hacer después de instalar **Linux Mint**, reparar un **pendrive dañado** así como formatearlo correctamente, instalar **ClamAV** y **ClamTK**, entre muchos otros. Todo esto para un usuario tan novel como yo es genial desde mi punto de vista.
- Decir además que no se si de manera directa e indirecta cambió mi enfoque, pues al llegar a Linux venía predispuesto a no utilizar la consola, y quería además utilizar extensivamente a **WINE**. Si bien desde mi punto de vista no tiene nada de malo utilizar **WINE** o **PlayOnLinux**, en realidad no lo utilizo de momento, y como ya he dicho más arriba he seguido varios manuales de la revista que enseñan como hacer las cosas mediante comandos.

Según palabras de **linuxeros** muchos más experimentados que yo y de la propia revista es normal que cada usuario recién llegado a Linux pase de una distro a otra hasta que encuentra la que se ajusta a sus necesidades o exigencias. Con esto no quiero criticar a ninguna distro, ningún entorno de escritorio, ni ningún software, ni ningún blog, porque lo único que siento por todas esas comunidades que sostienen tantos proyectos asombrosos es admiración y respeto. En mi caso comencé con muchos con Ubuntu 20.04, la cual me parece genial, pero el escritorio **GNOME** no funcionaba muy bien en mi Portátil la cual tiene ya 6 años, luego en un breve periodo de tiempo probé Ubuntu, Fedora, Linux Lite, hasta que aterricé en Linux Mint, con la cual me va genial. Algunos defensores a ultranza de GNOME me han dicho que Linux Mint es como un Windows. No voy a negar que un principio me sentí muy cómodo con una panel y un menú similares a la barra de tareas y el menú inicio de Windows. Pero esto es solo apariencia que se puede modificar muy fácil. Comparto algunas capturas de mi **CINNAMON** personalizado. Me encantan las múltiples opciones de personalización de Linux y la mayoría de aplicaciones. Pues abarcan desde lo meramente estético, hasta la automatización de tareas, pasando por la funcionalidad.

Quisiera dar mi opinión general sobre el **software libre y GNU/ Linux** ya que considero importante que cada cual sepa por que está donde está. Creo que lo más importante es la filosofía de colaboración en todos los aspectos. Creo que compartir los conocimientos que cada cual pueda albergar es lo más importante para que la especie humana pueda avanzar hacia un desarrollo sostenible. El software libre es prueba de ello y su modelo sirve y servirá a múltiples áreas del conocimiento. No importa si no tienes dinero que aportar o conocimientos de programación con compartir lo que sabes con los que sepan un poco menos que tu, promocionar o brindar tu opinión y experiencia de usuario, ya habrás aportado, y esto tiene más valor de lo que cualquiera puede imaginar.

Otra cosa que siempre ha llamado mi atención es que la mayoría tanto de aplicaciones como las propias distribuciones hacen un mejor uso de los recursos, casi siempre son más ligeras y pequeñas que sus contrapartes privativas/de pago. Sin embargo son tan o más poderosas. También creo de manera general (habrá sus excepciones) cada distribución según su funcionalidad viene con un software preinstalado que es realmente útil en cada caso. De ahí que este sea para mi uno de los puntos fuertes de Linux y uno de los parámetros esenciales a la hora de escoger una distro. Porque para que quiero un sistema del que tenga que quitar casi todo o simplemente no utilizar las herramientas que trae por defecto, y tener que instalarle hasta lo más insignificante buscando que funcione mejor y si estas palabras describen a Windows no es casualidad.

Quisiera probar muchas distribuciones Linux entre ellas **Debian**, **OpenSUSE**, **Manjaro**, **Kali**, así como todos los entornos de escritorio. Pero por ahora mi tiempo es limitado y el aprendizaje consume más del que tengo.

Por ultimo me quisiera agradecer nuevamente a la Revista **Sololinux** y toda la comunidad de software libre en general además de sugerir algunos tópicos sobre los cuales pudieran realizarse artículos interesantes dada la seriedad y profundidad con que se tratan los temas en esta publicación tan importante para los **linuxeros** de idioma español:

- Comparación de Navegadores, tanto de los más utilizados, como de los menos conocidos, pues en el mundo Linux hay muchos de estos y sería bueno que estos llamados navegadores alternativos fueran puestos debajo del lente critico y certero de la revista.
- Ofimática, conocer todas las suites que están disponibles en Linux y cual es su estado de desarrollo (aunque creo que LibreOffice va muy por delante no podemos obviar a las demás). También hay otras aplicaciones que si bien no integran ninguna de estas suites son imprescindibles para la mayoría de los usuarios como herramientas de PDF (soy consciente que de alguna manera esto se ha tratado en la revista) y otras utilidades que resultan imprescindibles en los entornos de oficina.
- Motores de búsqueda alternativos, por qué y cuando utilizarlos. Considero esto esencial pues si lo pensamos bien buscar información en internet es algo que hacemos constantemente y una las habilidades más importantes que todos sin importar cual es la tarea que realiza o su nivel de conocimientos informáticos debemos dominar.

Saludos a todas y todos.

