

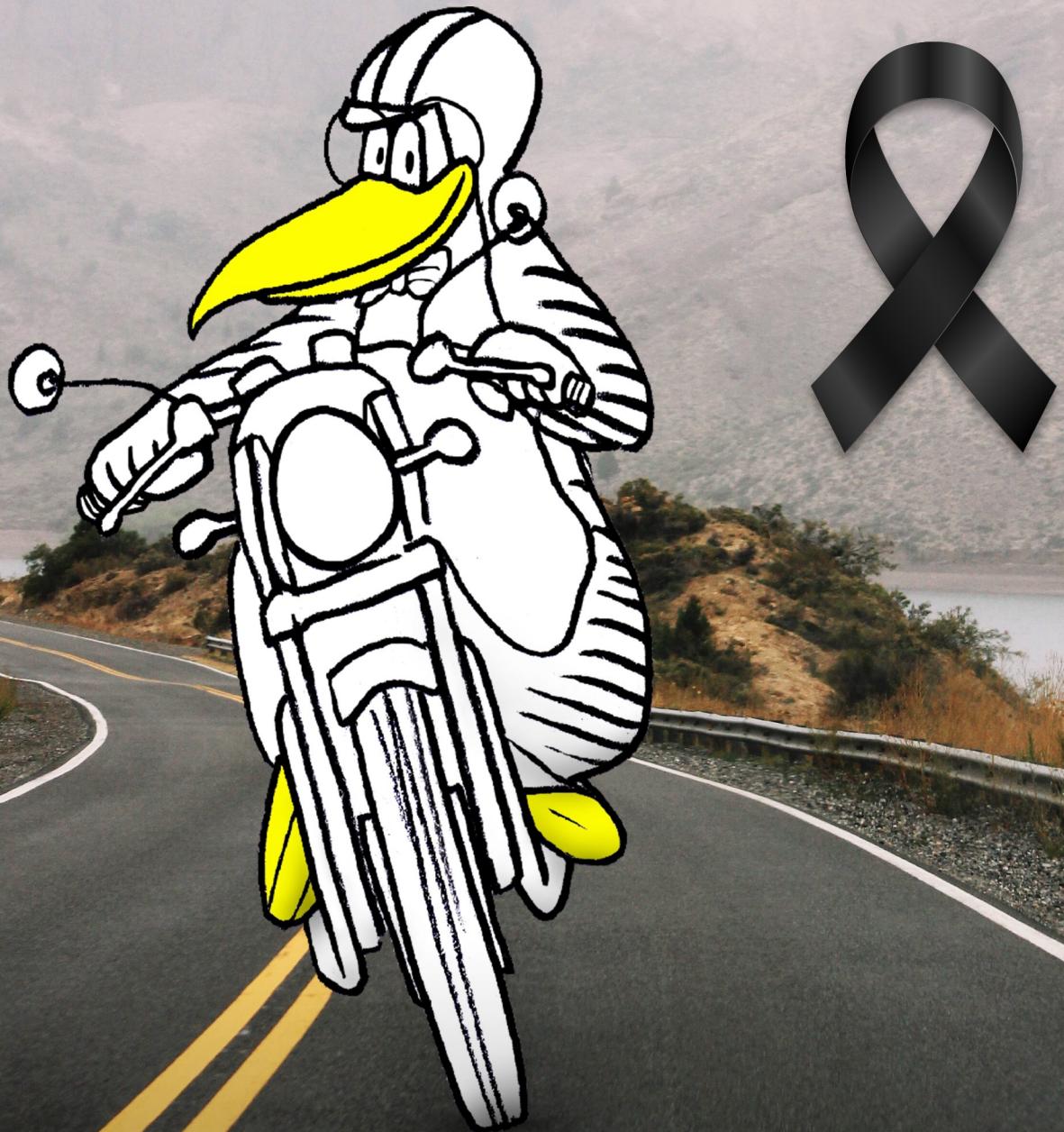
Visita nuestro sitio SoloLinux.es

MAGAZINE SOLO LINUX

Nº
32

Tu revista, la revista de tod@s

SEPTIEMBRE 2021



Uso del comando **passwd**
con ejemplos

Manjaro abandona
Mozilla Firefox

Instalar Rust
en Debian 11 Bullseye

Uso del comando **dmesg**
con ejemplos

Kali Linux 2021.3
Listo para su descarga

Instalar php 8
en Debian 11 Bullseye

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

Javier Salgado Serrano, por su donación a la revista

Aviso – Nota del autor:

Los sitios **SoloLinux.es**, **RevistaLinux.es** y la Revista **SoloLinux**, no mantienen ningún tipo de relación contractual con los propietarios de otros blogs, autores de opiniones publicadas o anunciantes de la revista.



La revista SOLOLINUX está realizada con **LibreOffice Impress 7.2.0.4**



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

Bienvenido a la Revista SOLOLINUX

Sergio G. B. nuestro compañero, mentor, Maestro.

Este mes la revista se ha quedado corta, no porque falten artículos (hay menos de los habituales) si no, porque no hemos sabido reemplazarle; Sergio nos ha dejado, aunque seguro que nos está observando desde su nuevo puesto.

El pasado 22 de septiembre, ha fallecido por culpa de un maldito virus por desgracia muy famoso.

Toda la comunidad está de luto y le echamos de menos. Era una inspiración para muchos, un guía incansable, rivalizaba en él su hambre de conocimientos y su basto saber acumulado pero, además, era un magnífico divulgador que nos sorprendía cada día, con un nuevo tema con su particular enfoque.

Sergio, los que te conocimos, intentaremos seguir con tu legado, sabemos que así lo habrías querido; quizás no consigamos igualarte, pero seguiremos adelante mientras podamos.

Gracias Sergio, hasta siempre.



Esta revista es de **distribución gratuita**.



designed by freepik

MANUALES

09. Uso del comando passwd con ejemplos
 13. Uso del comando dmesg con ejemplos



16. Combinar archivos pdf con PDFtk
 17. Cambiar de interfaz gráfica a línea de comandos con TTY



20. Crear un entorno virtual Python en Ubuntu 20.04
 23. Qué es el instalador calamares en Linux



24. Instalar Rust en Debian 11 Bullseye
 27. Instalar php 8 en Debian 11 Bullseye
 29. Uso del comando chage con ejemplos
 31. Instalar Opera en Debian 11 y todos sus derivados

SoloWordpress

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



CARTA AL LECTOR

07. CARTA AL LECTOR. GRACIAS
 08. Notas de los LECTORES

SOFTWARE

11. Comparar archivos y carpetas con meld



DISTROS LINUX

26. Kali Linux 2021.3 – Listo para su descarga



SEGURIDAD

15. Restringir la shell de usuario con rbash

HARDWARE

18. Conectar un iPhone como unidad externa en Ubuntu 20.04

www.sololinux.es

NOTICIAS

22. Manjaro abandona Mozilla Firefox





intel®

CORE™

i5

intel®

CORE™

i7

intel®

iRIS®^{Xe}

GRAPHICS

edge²

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

descúbrelos en www.vantpc.es



@vantpc



vant.pc



vantpc_es



t.me/vantpc

ABIERTA INSCRIPCIÓN



FABIÁN AMPALIO
CEO CLA IL



DIEGO NOBILE
Instructor



GABRIEL CANEPA
Tutor



ALEJANDRO DRABENČE
Tutor



JOAQUÍN AMPALIO
Staff

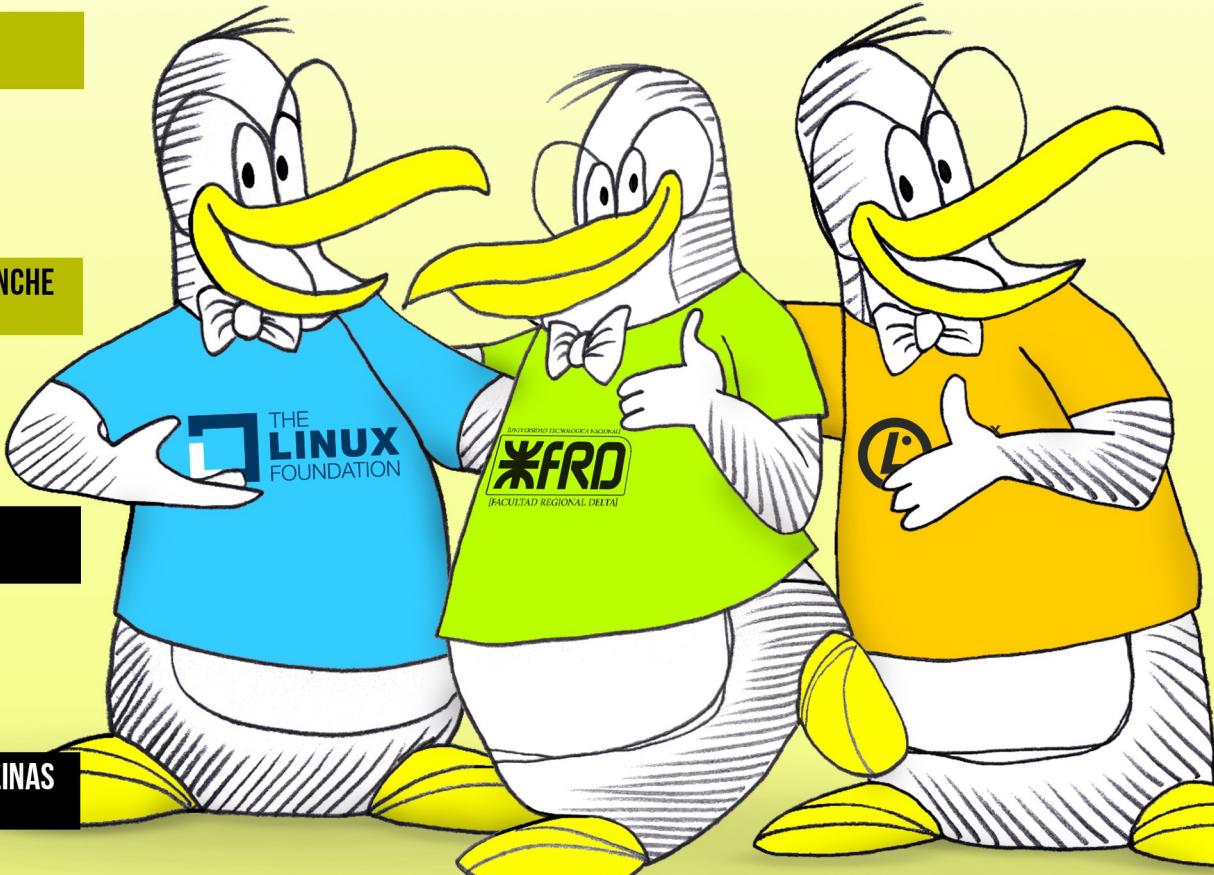


FRANCISCO SANDALINAS
Staff

Aprende y certifica Linux

LINUX SYSTEM ADMINISTRATOR & LINUX SYSTEM ENGINEER

CONSULTA PROMOCIONES ESPAÑA Y LATAM



+54 9 11 6969 9993



informes@carreralinux.com.ar



carreralinux.com.ar

CARTA AL LECTOR

Estamos de luto:

Estas líneas no solíamos escribirlas nosotros, siempre era Sergio G. B. Pero septiembre ha terminado con algo que no hubiéramos imaginado, y que nos ha golpeado en el corazón de nuestra comunidad.

El cerebro y creador de la comunidad, amigo, compañero, mentor y maestro **Sergio G. B.** ha fallecido. Estamos desolados, pero sabemos que desde donde esté nos seguirá apoyando y guiando.

Sergio G. B. fue y es una gran persona, que siempre ayudó sin mirar a quién, con una mentalidad libre como el software, y que supo dejar su huella marcada en esta nuestra comunidad, y en todo el panorama **Open-Source**.

Así ha quedado reflejado por la comunidad **Sololinux**, que se ha volcado en este momento de dolor, como del resto de comunidades, lectores, patrocinadores y empresas del sector que nos han hecho llegar una multitud de ánimos, apoyos y condolencias a la comunidad, familiares y amigos.

GRACIAS.

Nos deja un legado y un camino marcado, para continuar con su misión como él hubiera querido. Así perdurará en el tiempo lo que entre él y todos hemos creado.

Nuestro más sincero pésame a la familia, lamentando el dolor y reciban todo nuestro apoyo.

A Sergio G. B.:

Gracias por todo. Siempre estarás con nosotros.

www.sololinux.es



NOTAS DE LOS LECTORES. GRACIAS

Me he quedado muy apenado al enterarme hoy de la noticia, a pesar de conocer a Sergio únicamente a través del ciberespacio. Sus centenares de artículos destilan y atesoran muchísimo conocimiento, experiencia y pasión por lo que hacía. Nos deja todo un legado y es un ejemplo a seguir. Intercambié con él algunos correos y me demostró ser muy buena persona y me inspiró mucha confianza. A mí, como a otros compañeros que han intervenido, también me inspiró y animó a retornar a Linux y seguir aprendiendo. Que en paz descansse y que perviva su legado.

Firmado: Roberto G.

A pesar de ser nuevo en la comunidad, soy un gran lector de sus artículos de la revista Solo Linux, en mi país a los hombres de sistemas como Sergio, decimos que no ha muerto, solo a subido a la nube, una gran pérdida... descansse en paz maestro...

Firmado: Raul.

Triste noticia, pero toda semilla germina y crece y ese será su legado.

Que tenga un buen camino y mi sentido pésame para la familia.

Firmado: Julio Boniel.

Que mala noticia...

Mi más sentido pésame.

Tengo poco tiempo en el grupo (unos 7 u 8 meses) Y muy al principio, en uno de esos días de mensajes, Sergio borró una de mis respuestas.
Ya en lo privado, le aclaré que desde mi punto de vista, no había cometido ninguna falta.
Luego de aclarar, las diferencias de idioma entre España y México , llegamos a un sano empate.
Pero a partir de ese momento, comenzamos en un par de ocasiones a compartir historias, ideologías, temas del trabajo, etc.

En pocas palabras : todo un personaje.

Lástima que se nos adelantó...

Firmado: Héctor Anibal.

Un intelectual tecnologo... hasta el ultimo suspiro por el software libre...sea x covid o lo que sea...estarā en un plano espiritual donde compartirá un conocimiento mas absoluto de lo humano incluyendo lo infinito ..numeros, programacion, ciencia, etc...esa SABIDURIA informatica que ahora el alcanzará...en el otro plano lo conocerá todo...ojala su espíritu nos ayude en este plano...crear un mundo libre como ahora lo es el...

Firmado: D.

No me lo puedo creer. Descansa en paz Sergio. Gracias por tu obra.

Mi más sentido pésame a la familia y a esta tu familia linuxera.

Firmado: Luis

Este comunicado me deja tan...descolocado?sin palabras? desconcertado...?y muy triste...cuántas veces me habrá ayudado....cuántas respuestas a un recién llegado a este mundo Linux....que descansse en paz, mucha fuerza y ánimos a su familia,y compañeros. Gracias Sergio por todos tus artículos y sabiduría compartida....

Firmado: Jose Luis.

Rayos, yo me di cuenta quien era él en días pasados porque le hice unas preguntas directas. No soy muy viejo en este grupo. Cuantos buenos artículos escribió Que sus parientes reciban consuelo y paz a sus restos. Espero podamos salir de este virus lo más rápido posible.

Firmado: Drbn.

Que root lo acompañe en el bucle donde el desee pasar su eternidad.

Firmado: Nelson Sanchez.

Que noticia más triste y lamentable, mi más sentido pésame para sus familiares y amigos, me ayudo muchísimo con una antigua Pc que realmente gracias a él puse a funcionar nuevamente con una distro liviana que me recomendó, siempre leyendo sus artículos que descubrí en el poco tiempo que llevo en este canal, una gran pérdida su partida para todos nosotros

Firmado: Jose David.

Una noticia fatídica. Sergio era una persona que siempre estaba atento a cualquier duda o sugerencia y hasta te hacía una publicación si era necesario para explicar algo que desconocías, y de paso así ayudar a otras personas que se pudieran encontrar con la misma inquietud...

Mi más sentido pésame para la familia y todas las personas que de un modo u otro tuvimos la suerte de poder conocer un poquito (o un mucho!) a este gran amigo. Ojalá hubiera un CTRL+Z para esto...

Firmado: Benny Beat.

Uso del comando passwd con ejemplos



La herramienta **passwd**, tiene la particularidad de permitir la modificación de contraseñas, de todas las cuentas de usuario incluidas en un sistema Linux. Cuando trabajas con este comando, debes tener presente que si eres un usuario normal, solo podrás **cambiar la contraseña** de tu propia cuenta. Por otro lado tenemos el **superusuario o usuario root**, que si tiene permisos para modificar cualquier cuenta de usuario.

Muchos usuarios desconocen que con esta herramienta, se pueden hacer muchas más operaciones aparte de **modificar el password**. También podemos imprimir o modificar la información de la cuenta, la **shell de inicio** e incluso las fechas de caducidad, entre otras.



Uso del comando passwd con ejemplos

Su función principal es establecer una contraseña de usuario o, modificar la contraseña existente. La sintaxis es la siguiente:

```
passwd [usuario]
```

Si quieras modificar la contraseña del usuario actual, no es necesario definir ningún usuario. Vemos un ejemplo completo, en el cual podrás comprobar sus pasos.

1. Solicita la contraseña actual.
2. Introduces la nueva.
3. Repites la nueva.
4. La seguridad del sistema no admite la contraseña nueva por ser demasiado simple.
5. Tipeamos un password más complejo.
6. Lo repites.
7. Se acepta y actualiza a la nueva contraseña.

```
sergio@demo-sololinux:~$ passwd
Cambiando la contraseña de sergio.
(actual) contraseña de UNIX:
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
Bad: new password is too simple
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: contraseña actualizada correctamente
sergio@demo-sololinux:~$
```

Ahora vamos a bloquear la cuenta de un usuario haciendo que su contraseña caduque, después la desbloqueamos siguiendo el mismo proceso. Es evidente que para realizar este proceso debemos ser **root**.

```
# Bloquear cuenta
passwd -l sergio
# Desbloquear cuenta
passwd -u sergio
```

La imagen nos muestra el proceso completo.

```
root@demo-sololinux:~#
root@demo-sololinux:~# passwd -l sergio
passwd: información de caducidad de la contraseña cambiada.
root@demo-sololinux:~#
root@demo-sololinux:~# passwd -u sergio
passwd: información de caducidad de la contraseña cambiada.
root@demo-sololinux:~#
```

Si quieras, también puedes establecer la duración mínima y máxima en días, de la caducidad de la contraseña.

```
# Duración mínima
passwd -n 5 [usuario]
# Duración máxima
passwd -x 500 [usuario]
```

```
root@demo-sololinux:~#
root@demo-sololinux:~# passwd -n 5 sergio
passwd: información de caducidad de la contraseña cambiada.
root@demo-sololinux:~#
root@demo-sololinux:~# passwd -x 500 sergio
passwd: información de caducidad de la contraseña cambiada.
root@demo-sololinux:~#
```

Con la opción «**-S**», podemos visualizar el estado de la contraseña de usuario.

```
passwd -S [usuario]
```

```
root@demo-sololinux:~#
root@demo-sololinux:~# passwd -S sergio
sergio P 09/02/2021 5 500 7 -1
root@demo-sololinux:~#
```

Es interesante lanzar un aviso al usuario, comunicándole que su contraseña caducara pronto. Establecemos los días antes de que caduque con la opción «**-W**».

```
passwd -w [días] [usuario]
```

MANUALES

En el ejemplo también vemos como ha cambiado el status de la contraseña.

```
root@demo-sololinux:~#  
root@demo-sololinux:~# passwd -w 10 sergio  
passwd: información de caducidad de la contraseña cambiada.  
root@demo-sololinux:~#  
root@demo-sololinux:~# passwd -S sergio  
sergio P 09/02/2021 5 500 10 -1  
root@demo-sololinux:~#
```

Este último comando que te propongo no es recomendable, pero si es útil en caso de que olvides la contraseña de algún usuario. La opción «**-d**» borra el password del usuario indicado.

```
passwd -d [usuario]
```

```
root@demo-sololinux:~#  
root@demo-sololinux:~# passwd -d sergio  
passwd: información de caducidad de la contraseña cambiada.  
root@demo-sololinux:~#
```

www.sololinux.es

Con el argumento «**-h**», revisamos las opciones admitidas por el comando.

```
passwd -h
```

Se imprime la salida.

```
sergio@sololinux:~$ passwd -h  
Modo de uso: passwd [opciones] [USUARIO]  
Opciones:  
-a, --all           informa del estado de las contraseñas de  
                   todas las cuentas  
-d, --delete        borra la contraseña para la cuenta indicada  
-e, --expire        fuerza a que la contraseña de la cuenta  
                   caduque  
-h, --help          muestra este mensaje de ayuda y termina  
-k, --keep-tokens   cambia la contraseña sólo si ha caducado  
-i, --inactive INACTIVO establece la contraseña inactiva después de  
                   caducar a INACTIVO  
-l, --lock           bloquea la contraseña de la cuenta indicada  
-n, --mindays DÍAS_MIN establece el número mínimo de días antes  
                   de que se cambie la contraseña a DÍAS_MIN  
-q, --quiet          modo silencioso  
-r, --repository REP cambia la contraseña en el repositorio REP  
-R, --root CHROOT_DIR directorio en el que hacer chroot  
-S, --status         informa del estado de la contraseña la cuenta  
                   indicada  
-u, --unlock         desbloquea la contraseña de la cuenta indicada  
-w, --warndays DÍAS_AVISO establece el aviso de caducidad a DÍAS_AVISO  
-x, --maxdays DÍAS_MAX establece el número máximo de días antes de  
                   cambiar la contraseña a DÍAS_MAX
```

```
sergio@sololinux:~$
```

Si aún necesitas más información, este comando dispone de un completo manual integrado.

```
man passwd
```

www.sololinux.es

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

Chat de SoloLinux en Telegram

Comparar archivos y carpetas con meld



Comparar archivos y directorios es una tarea bastante común, en especial para los desarrolladores. Para este tipo de operaciones, en **linux** disponemos del fantástico **comando diff**, que busca diferencias por caracteres, líneas, tamaño y mucho más. Como es lógico, **diff** opera en terminal, y esto puede suponer un engorro para los usuarios más noveles, por ello se creó la **herramienta Meld**.

Meld es una interfaz gráfica del **comando diff**, considerada por la **comunidad Linux** como una de las mejores herramientas para realizar comparaciones entre archivos, directorios y proyectos de control de versiones. Es ampliamente utilizada por los usuarios normales y, también por desarrolladores de aplicaciones para verificar sus desarrollos diarios. Algunas de las características más destacadas, son:

- Ofrece una exhaustiva comparación de archivos / carpetas (bidireccional y tridireccional).
- Capacidad de combinar archivos y carpetas, gracias a la **función Auto Merge**.
- Permite el resaltado de sintaxis y la administración de archivo.
- Sencilla y agradable interfaz gráfica.
- Soporte para sistemas de control de versiones (**Git, Mercurial, Bazaar y SVN**).
- Permite ver el estado de las versiones en el sistema de control.
- Soporta la integración con herramientas en línea de comandos (git y mergetool).

En el artículo de hoy, vemos como instalar Meld en nuestra **distribución Linux** preferida, además de un sencillo ejemplo de uso que resultara de tu interés.



Comparar archivos y carpetas con meld

No vamos a enumerar los requisitos para el uso de la herramienta, porque son comunes en cualquier distribución moderna. Python 3.x, GTK 3.x, etc.

Selecciona el comando de instalación de la herramienta, dependiendo de tu sistema.

Debian, Ubuntu, Mint y derivados:

```
sudo apt-get install meld
```

Fedora, CentOS, Alma Linux y derivados:

```
sudo dnf install meld
# o
sudo yum install meld
```

Arch Linux, Manjaro y derivados:

```
sudo pacman -S meld
```

SUSE, OpenSUSE y derivados:

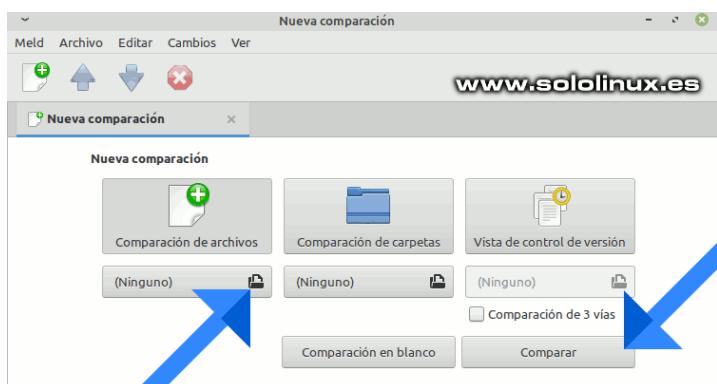
```
sudo zypper install meld
```

```
root@sololinux:#
root@sololinux: # sudo apt-get install meld
 Leyendo lista de paquetes... Hecho
 Creando árbol de dependencias
 Leyendo la información de estado... Hecho
 Se instalarán los siguientes paquetes NUEVOS:
   meld
 0 actualizados, 1 nuevos se instalarán. 0 para eliminar y 0 no actualizados.
 Se necesita descargar 494 kB de archivos.
 Se utilizarán 3.177 kB de espacio de disco adicional después de esta operación.
 Des:1 http://archive.ubuntu.com/ubuntu bionic/universe amd64 meld all 3.18.0-6 [494 kB]
 Descargados 494 kB en 0s (1.243 kB/s)
 Seleccionando el paquete meld previamente no seleccionado.
 (Leyendo la base de datos ... 376850 ficheros o directorios instalados actualmente.)
 Preparando para desempaquetar .../archives/meld_3.18.0-6_all.deb ...
 Desempaquetando meld (3.18.0-6) ...
 Configurando meld (3.18.0-6) ...
 Procesando disparadores para desktop-file-utils (0.23+linuxmint8) ...
 Procesando disparadores para libglib2.0-0:amd64 (2.56.4-0ubuntu0.18.04.8) ...
 Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...
 Procesando disparadores para shared-mime-info (1.9-2) ...
 Procesando disparadores para gnome-menus (3.13.3-11ubuntu1.1) ...
 Procesando disparadores para hicolor-icon-theme (0.17-2) ...
 Procesando disparadores para mime-support (3.60ubuntu1) ...
root@sololinux: #
```

www.sololinux.es

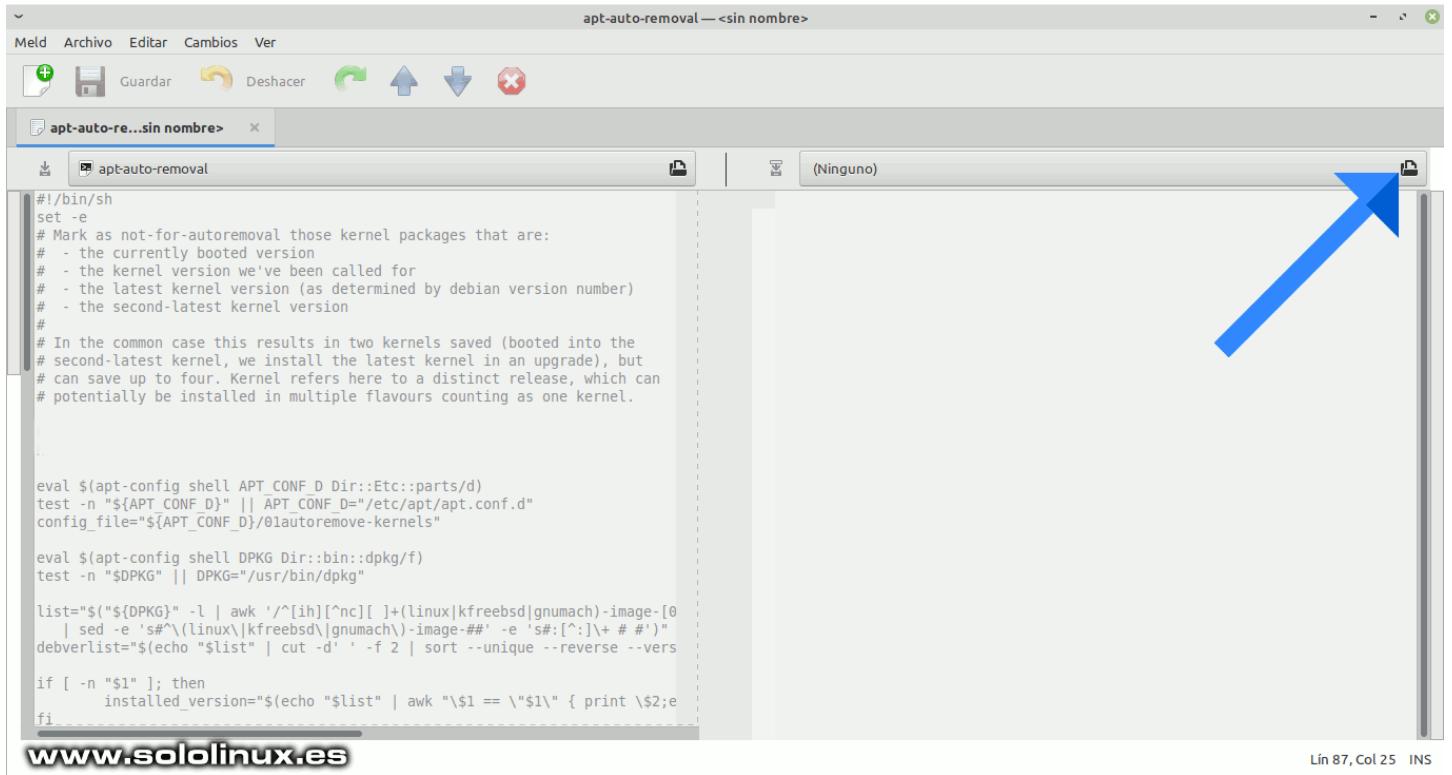
Para nuestra comparativa, tomamos como ejemplo el script del **artículo** «**Modificar la cantidad de Kernel que se guardan en Ubuntu**». Usando **meld**, compararemos el archivo original «**apt-auto-removal**» con el modificado «**nuevo-apt-auto-removal**». Verás que fácil.

Desde el menú de aplicaciones de tu **entorno de escritorio** preferido, podrás lanzar la **herramienta Meld**. De forma automática aparece una ventana interactiva, en la cual elegimos el archivo primario (principal). Una vez seleccionado, pulsa en «**Comparar**».



Se abre la herramienta con dos ventanas, en las cuales se compararan los archivos. Haz clic donde te indica la flecha de la siguiente imagen, para seleccionar el segundo archivo.

SOFTWARE



```
apt-auto-removal —<sin nombre>
Meld Archivo Editar Cambios Ver
Guarda Deshacer ↻ ↑ ↓ ✎
apt-auto-re...sin nombre> ×
apt-auto-removal (Ninguno)
#!/bin/sh
set -e
# Mark as not-for-autoremoval those kernel packages that are:
# - the currently booted version
# - the kernel version we've been called for
# - the latest kernel version (as determined by debian version number)
# - the second-latest kernel version
#
# In the common case this results in two kernels saved (booted into the
# second-latest kernel, we install the latest kernel in an upgrade), but
# can save up to four. Kernel refers here to a distinct release, which can
# potentially be installed in multiple flavours counting as one kernel.

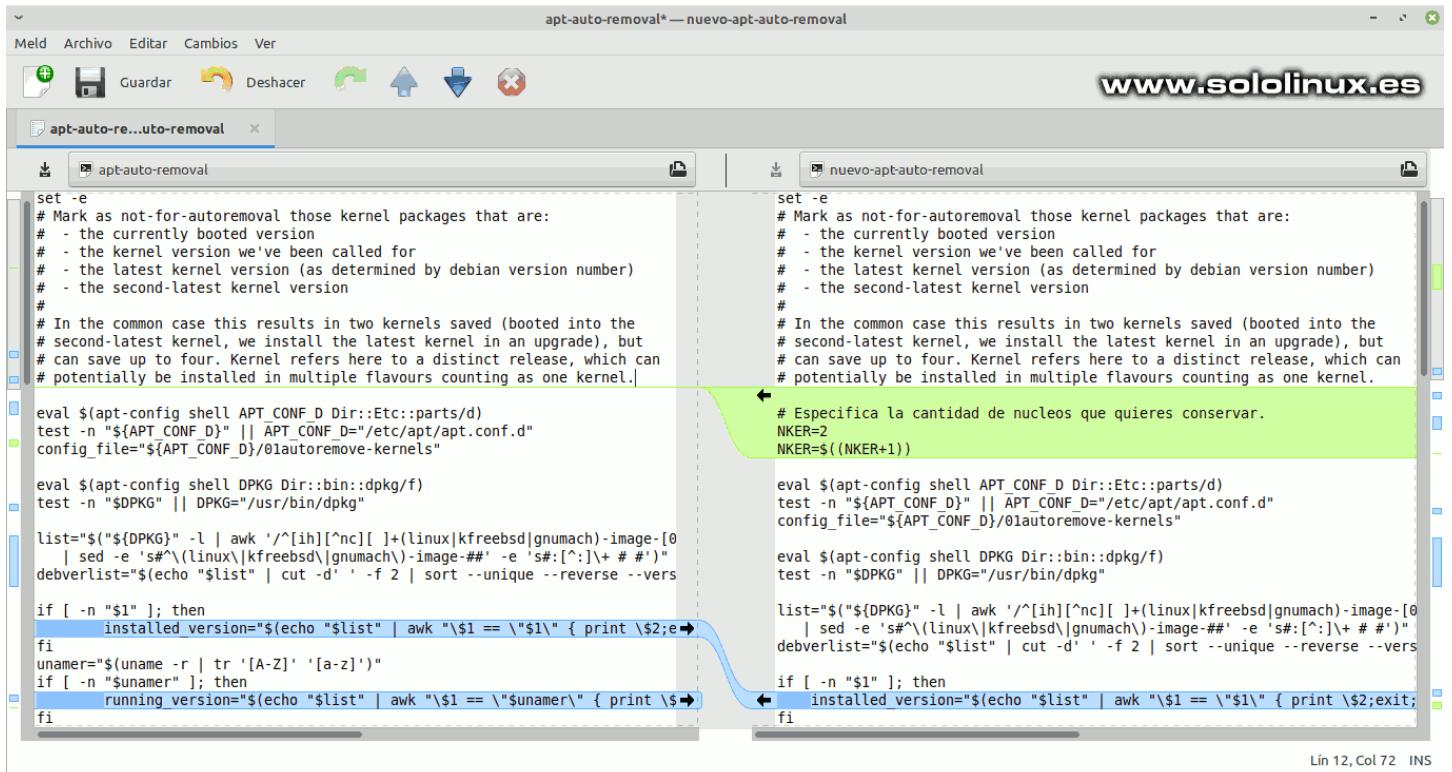
eval $(apt-config shell APT CONF_D Dir::Etc::parts/d)
test -n "${APT_CONF_D}" || APT_CONF_D="/etc/apt/apt.conf.d"
config_file="${APT_CONF_D}/01autoremove-kernels"

eval $(apt-config shell DPKG Dir::bin::dpkg/f)
test -n "$DPKG" || DPKG="/usr/bin/dpkg"

list=$(ls ${DPKG} -l | awk '/^([ih][^nc][ ])+(linux|kfreebsd|gnumach)-image-[0
| sed -e 's#\(^linux\|kfreebsd\|gnumach\)-image-##' -e 's#:[:]#+ # #''
debverlist=$(echo "$list" | cut -d' ' -f 2 | sort --unique --reverse --vers
if [ -n "$1" ]; then
    installed_version=$(echo "$list" | awk "\$1 == \"\$1\" { print \$2;e
fi
www.sololinux.es
Lin 87, Col 25 INS
```

Como puedes ver en la siguiente imagen, las diferencias están marcadas por colores. El tono verde claro, indica que esa zona al completo (en este caso caracteres) no se encuentra en el archivo primario. Al observar detenidamente, vemos que existe una flecha pequeñita en el segundo archivo. Si pulsas en ella... toda la zona marcada en verde se clonara al archivo primario.

Las líneas que tenemos en azul, son diferencias en espacios que en nuestro caso no influyen para nada en el buen funcionamiento del script. Una vez tengas todos los cambios realizados, puedes guardar el archivo. Es así de simple.



```
apt-auto-removal* — nuevo-apt-auto-removal
Meld Archivo Editar Cambios Ver
Guarda Deshacer ↻ ↑ ↓ ✎
apt-auto-re...uto-removal ×
apt-auto-removal nuevo-apt-auto-removal
#!/bin/sh
set -e
# Mark as not-for-autoremoval those kernel packages that are:
# - the currently booted version
# - the kernel version we've been called for
# - the latest kernel version (as determined by debian version number)
# - the second-latest kernel version
#
# In the common case this results in two kernels saved (booted into the
# second-latest kernel, we install the latest kernel in an upgrade), but
# can save up to four. Kernel refers here to a distinct release, which can
# potentially be installed in multiple flavours counting as one kernel.

eval $(apt-config shell APT CONF_D Dir::Etc::parts/d)
test -n "${APT_CONF_D}" || APT_CONF_D="/etc/apt/apt.conf.d"
config_file="${APT_CONF_D}/01autoremove-kernels"

eval $(apt-config shell DPKG Dir::bin::dpkg/f)
test -n "$DPKG" || DPKG="/usr/bin/dpkg"

list=$(ls ${DPKG} -l | awk '/^([ih][^nc][ ])+(linux|kfreebsd|gnumach)-image-[0
| sed -e 's#\(^linux\|kfreebsd\|gnumach\)-image-##' -e 's#:[:]#+ # #''
debverlist=$(echo "$list" | cut -d' ' -f 2 | sort --unique --reverse --vers
if [ -n "$1" ]; then
    installed_version=$(echo "$list" | awk "\$1 == \"\$1\" { print \$2;e
fi
unamer=$(uname -r | tr '[A-Z]' '[a-z]')
if [ -n "$unamer" ]; then
    running_version=$(echo "$list" | awk "\$1 == \"\$unamer\" { print \$2;e
fi
# Especifica la cantidad de nucleos que quieras conservar.
NKER=2
NKER=$((NKER+1))

eval $(apt-config shell APT CONF_D Dir::Etc::parts/d)
test -n "${APT_CONF_D}" || APT_CONF_D="/etc/apt/apt.conf.d"
config_file="${APT_CONF_D}/01autoremove-kernels"

eval $(apt-config shell DPKG Dir::bin::dpkg/f)
test -n "$DPKG" || DPKG="/usr/bin/dpkg"

list=$(ls ${DPKG} -l | awk '/^([ih][^nc][ ])+(linux|kfreebsd|gnumach)-image-[0
| sed -e 's#\(^linux\|kfreebsd\|gnumach\)-image-##' -e 's#:[:]#+ # #''
debverlist=$(echo "$list" | cut -d' ' -f 2 | sort --unique --reverse --vers
if [ -n "$1" ]; then
    installed_version=$(echo "$list" | awk "\$1 == \"\$1\" { print \$2;exit;
fi
www.sololinux.es
Lin 12, Col 72 INS
```

Uso del comando dmesg con ejemplos



Al iniciar un sistema Linux, el kernel identifica los dispositivos de hardware disponibles en el sistema. A medida que el kernel los va detectando e identificando, se generan unos registros que se escriben en un buffer (en anillo) del **Kernel**. Este **buffer en anillo**, también conocido como **buffer cíclico o circular**, es específico para manejar el registro interno del Kernel.

Saber leer este registro es importante, pues más de una vez puede salvar tu sistema. Recuerda que como en cualquier registro, en este también se guardan los errores. Con el **comando dmesg** no solo podrás ver todo el registro, sino que además podrás identificar de manera simple los errores por tipo, algo especialmente útil para nosotros. Vemos cómo se trabaja con este comando.

```
0.015831 On node 0 total_pages: 1024414  
0.015834 DMA zone: 64 pages used for memmap  
0.015836 DMA zone: 1142 pages reserved  
0.015838 DMA zone: 3982 pages, LIFO batch:0  
0.015990 DMA32 zone: 7753 pages used for memmap  
0.015992 DMA32 zone: 496144 pages, LIFO batch:63  
0.040605 Normal zone: 8192 pages used for memmap  
0.040608 Normal zone: 524288 pages, LIFO batch:63  
0.059082 ACPI: Local APIC address 0xf00e0000  
0.059151 ACPI: IRQ0 used by override.  
0.059153 ACPI: IRQ9 used by override.  
0.059180 e820: update [mem 0x78865000-0x7886efff] usable ==> reserved
```

Uso del comando dmesg con ejemplos

Su forma de uso simple, imprime el buffer al completo.

dmesg

```
sergio@sololinux:~$ dmesg
[    0.000000] Linux version 5.4.0-81-generic (buildd@lgw01-
amd64-051) (gcc version 7.5.0 (Ubuntu 7.5.0-3ubuntu1~18.04))
#91~18.04.1-Ubuntu SMP Fri Jul 23 13:36:29 UTC 2021 (Ubuntu
5.4.0-81.91~18.04.1-generic 5.4.128)
[    0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-5.4.0-
81-generic root=UUID=77674f63-7f05-4b96-a2aa-125cd616afcb ro
fsck.mode=force fsck.repair=yes
[    0.000000] KERNEL supported cpus:
[    0.000000]   Intel GenuineIntel
[    0.000000]   AMD AuthenticAMD
[    0.000000]   Hygon HygonGenuine
[    0.000000]   Centaur CentaurHauls
[    0.000000]   zhaixin Shanghai
[    0.000000] x86/fpu: x87 FPU will use FXSAVE
[    0.000000] BIOS-provided physical RAM map:
[    0.000000] BIOS-e820: [mem 0x0000000000000000-
0x000000000006efff] usable
[    0.000000] BIOS-e820: [mem 0x000000000006f000-
0x000000000006ffff] ACPI NVS
```

```
[      0x0000000] BIOS-e820: [mem 0x000000000000070000-  
0x00000000000008ffff] usable  
[      0x0000000] BIOS-e820: [mem 0x000000000000090000-  
0x00000000000009ffff] reserved  
[      0x0000000] BIOS-e820: [mem 0x000000000000100000-  
0x00000000001fffffff] usable  
[      0x0000000] BIOS-e820: [mem 0x0000000002000000-  
0x000000000201ffff] reserved  
[      0x0000000] BIOS-e820: [mem 0x0000000002020000-  
0x0000000007a40efff] usable  
[      0x0000000] BIOS-e820: [mem 0x0000000007a40f000-  
0x0000000007aebfff] reserved  
[      0x0000000] BIOS-e820: [mem 0x0000000007aebf000-  
0x0000000007bfbfff] ACPI NVS  
[      0x0000000] BIOS-e820: [mem 0x0000000007bfbf000-  
0x0000000007bfffffff] ACPI data  
[      0x0000000] BIOS-e820: [mem 0x0000000007bffff000-  
0x0000000007bfffffff] usable  
[      0x0000000] BIOS-e820: [mem 0x000000000e00f8000-  
0x000000000e00f8ffff] reserved  
[      0x0000000] BIOS-e820: [mem 0x000000000fed01000-  
0x000000000fed01ffff] reserved  
[      0x0000000] BIOS-e820: [mem 0x000000000ffa00000-  
0x000000000fffffff] reserved  
[      0x0000000] BIOS-e820: [mem 0x0000000010000000-  
0x0000000017fffffff] usable  
[      0x0000000] NX (Execute Disable) protection: active  
[      0x0000000] e820: update [mem 0x742f7018-0x74307057]  
usable ==> usable  
[      0x0000000] e820: update [mem 0x742f7018-0x74307057]  
usable ==> usable  
[      0x0000000] extended physical RAM map:  
[      0x0000000] reserve setup_data: [mem 0x0000000000000000-  
0x00000000000006efff] usable  
[      0x0000000] reserve setup_data: [mem 0x00000000000006f000-  
0x00000000000006ffff] ACPI NVS  
  
etc.....etc.....
```

Si prefieres imprimir sin formato...

Cuando tienes errores, lo recomendable es que apliques los filtros que vienen incorporados con el comando. Así podrás identificar el error, de forma sencilla y rápida. Los filtros disponibles son los siguientes:

- **emerg** – Error que impide el funcionamiento del sistema correctamente.
 - **alert** – Error que requiere intervención inmediata.
 - **crit** – Sistema bajo condiciones críticas.
 - **err** – Sistema bajo condiciones de error.
 - **warn** – Advertencias del sistema.
 - **debug** – Advertencias de depuración.
 - **notice** – Condiciones normales, pero debes prestar atención al aviso.
 - **info** – Información general.

MANUALES

Estos filtros de mensajes, los debes usar con esta sintaxis...

```
dmesg --level=alert,[filtro]
```

Ejemplos de uso.

```
dmesg --level=alert,emerg  
##  
dmesg --level=alert>alert  
##  
dmesg --level=alert,err  
##  
dmesg --level=alert,crit  
##  
dmesg --level=alert,warn  
##  
dmesg --level=alert,debug  
##  
dmesg --level=alert,notice  
##  
dmesg --level=alert,info
```

Imagen con ejemplos de uso, en el que podemos observar una serie de errores y alertas. Tendremos que revisar que sucede, aunque en realidad es una vieja **tarjeta SD** (no me preocupa mucho).

También tienes la opción de imprimir el buffer y, posteriormente borrar el ciclo del anillo.

dmesg -c

```
sergio@sololinux:~$ dmesg -c
[12501.582069] usb 1-4: USB disconnect, device number 10
[12503.462291] usb 1-4: new high-speed USB device number 11
using xhci_hcd
[12503.622454] usb 1-4: New USB device found, idVendor=0bda,
idProduct=0177, bcdDevice=77.11
[12503.622463] usb 1-4: New USB device strings: Mfr=1,
Product=2, SerialNumber=3
[12503.622469] usb 1-4: Product: USB2.0-CRW
[12503.622475] usb 1-4: Manufacturer: Generic
[12503.622479] usb 1-4: SerialNumber: 20121112761000000
[12503.630446] ums-realtek 1-4:1.0: USB Mass Storage device
detected
[12503.635461] scsi host2: usb-storage 1-4:1.0
[12504.652987] scsi 2:0:0:0: Direct-Access      Generic-
SD/MMC/MS PRO    1.00 PQ: 0 ANSI: 4
[12504.654874] scsi 2:0:0:0: Attached scsi generic sg1 type
0
[12505.743088] sd 2:0:0:0: [sdb] 1950720 512-byte logical
blocks: (999 MB/953 MiB)
[12505.743424] sd 2:0:0:0: [sdb] Write Protect is off
[12505.743427] sd 2:0:0:0: [sdb] Mode Sense: 2f 00 00 00
[12505.743759] sd 2:0:0:0: [sdb] Write cache: disabled, read
cache: enabled, doesn't support DPO or FUA
[12505.763833] sdb: sdb1
[12505.765578] sd 2:0:0:0: [sdb] Attached SCSI removable
disk
dmesg: fallo al leer el «buffer» del núcleo: Operación no
permitida
sergio@sololinux:~$
```

Ahora especificamos el dispositivo del cual quieres ver los mensajes, en nuestro caso USB.

```
dmesg | grep -i usb
```

```
sergio@sololinux:~$ dmesg | grep -i usb
[12501.582069] usb 1-4: USB disconnect, device number 10
[12503.462291] usb 1-4: new high-speed USB device number 11 using xhci_hcd
[12503.622454] usb 1-4: New USB device found, idVendor=0bda, idProduct=0177, bcdDevice=77.11
[12503.622463] usb 1-4: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[12503.622469] usb 1-4: Product: USB2.0-CRW
[12503.622475] usb 1-4: Manufacturer: Generic
[12503.622479] usb 1-4: SerialNumber: 20121112761000000
[12503.630446] ums-realtek 1-4:1.0: USB Mass Storage device detected
[12503.635461] scsi host2: usb-storage 1-4:1.0
sergio@sololinux:~$
```

Si tienes problemas con el anillo, puedes probar a reducir o aumentar su tamaño.

```
dmesg -s 1024
```

Es evidente, que este comando tiene muchas más opciones y filtros. Si quieres aprender más sobre él, revisa su manual integrado en **consola**.

man dmesg



Canales de Telegram:
Canal SoloLinux
Canal SoloWordpress

[Chat de SoloLinux en Telegram](#)



designed by freepik

Restringir la shell de usuario con rbash

En **Linux**, al igual que en otros sistemas derivados de **UNIX**, al crear un nuevo usuario se le asigna un **shell** predeterminado (normalmente **/bin/bash**). Esto quiere decir que un usuario con suficientes conocimientos, puede moverse libremente por otros directorios a los cuales no debería tener acceso, o incluso ejecutar alguna acción que dañe a otros usuarios.

Para evitar situaciones tan desagradables como las mencionadas anteriormente, existe **rbash**. **Rbash**, también conocido como **restricted shell**, tiene la capacidad de restringir las operaciones que un usuario (o varios), puede realizar en **línea de comandos**. En este artículo vemos como hacer esta operación, de manera sencilla y rápida.

Por ejemplo, si ejecutamos el modo simple **rbash**, o **bash** con la **opción -r** en la invocación, el **shell** se restringe automáticamente el usuario actual. Volverá a su estado normal al reiniciar el entorno. Observa la imagen de ejemplo.

```
sergio@demo-sololinux:~$  
sergio@demo-sololinux:~$ cd Descargas  
sergio@demo-sololinux:~/Descargas$  
sergio@demo-sololinux:~/Descargas$ rbash  
sergio@demo-sololinux:~/Descargas$ cd  
rbash: cd: restringido  
sergio@demo-sololinux:~/Descargas$
```

www.sololinux.es

Restringir la shell de usuario con rbash

Un modo restringido se usa para configurar un entorno más controlado que el estándar. Su comportamiento y forma de operar es idéntico a **bash**, con la excepción de lo que no está permitido. Vemos un listado de lo no permitido, más habitual.

- Cambiar de directorio con el **comando cd**.
- No se permite modificar los valores de **SHELL**, **PATH**, **ENV** o **BASH_ENV**.
- Comandos, secuencias o nombres que contengan «», como carácter o argumento.
- Archivos que contengan una barra como argumento para la opción **-p** del propio hash.
- Importar definiciones de funciones desde el entorno de inicio de shell.
- Analizar el valor de **SHELLOPTS** desde el entorno de inicio shell.
- Redirigir salidas usando los operadores de redirección **>**, **>|**, **<|**, **>&**, **&>** y **>>**.
- Usar **exec** para reemplazar el shell con otro comando.
- Agregar o eliminar comandos integrados con las opciones **-f** y **-d**, del **comando enable**.
- Uso del comando **enable builtin** para habilitar funciones shell, que están deshabilitadas.
- Desactivar cualquier modo de salir del modo restringido.
- Alguna más de menor importancia.

Si queremos restringir solo a un usuario, modificamos su entorno de inicio en **«\$HOME/.bash_profile»**, con un sencillo script. En el home del usuario, creamos el archivo correspondiente.

```
sudo nano .bash_profile
```

Copia y pega lo siguiente.

```
# .bash_profile  
# Get the aliases and functions  
if [ -f ~/.bashrc ]; then  
. ~/.bashrc  
fi  
# User specific environment and startup programs  
PATH=$PATH:$HOME/bin  
export PATH  
bash -r
```

Guarda el archivo, cierra el editor y reinicia el entorno de inicio del usuario en particular. Ahora probamos que funciona correctamente.

```
cd /etc/  
cd /tmp/
```

```
sergio@demo-sololinux:~$  
sergio@demo-sololinux:~$ cd /etc/  
bash: cd: restringido  
sergio@demo-sololinux:~$  
sergio@demo-sololinux:~$ cd /tmp/  
bash: cd: restringido  
sergio@demo-sololinux:~$
```

www.sololinux.es

Otra opción es modificar la **shell** predeterminada.

```
chsh -s /bin/rbash
```

También por usuario.

```
chsh -s /bin/rbash usuario
```

```
sergio@demo-sololinux:~$  
sergio@demo-sololinux:~$ sudo chsh -s /bin/rbash sergio  
sergio@demo-sololinux:~$
```

De todas formas, si tienes varios usuarios lo normal es imponer las mismas restricciones a todos. El superusuario jamás tendrá restricciones. En este caso (recomendable), creamos un **enlace simbólico** en **/bin de bash a rbash**, ningún usuario tendrá más permisos que los necesarios para su trabajo.

```
sudo cd /bin  
sudo ln -s bash rbash
```

Nota: En algunas **distribuciones linux** la URL puede variar, por ejemplo la ruta **/usr**.

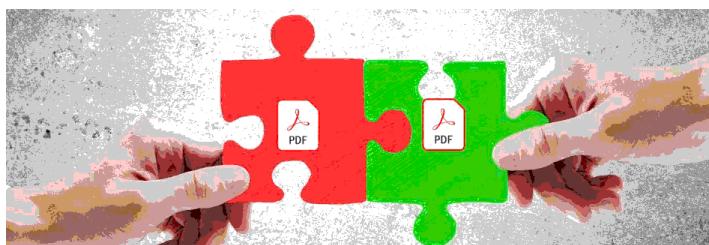
Combinar archivos pdf con PDFtk

Seguro que tienes multitud de documentos PDF almacenados en tu sistema. Deshacerte de ellos no es una opción, pues son manuales interesantes y las **revistas SoloLinux**, tal vez sea interesante hacer una copia de seguridad, pero tienes poco espacio y demasiados GB en pdf.

Una excelente solución es combinar varios archivos PDF en un solo PDF, así podrás administrar, visualizar o realizar copias de seguridad de manera más eficiente y rápida. **PDFtk Server** es una herramienta de línea de comandos, que entre otras muchas cosas nos permite combinar rápidamente **varios archivos PDF en uno solo**.

PDFtk, también conocido como **PDF Toolkit**, tiene una versión compatible con Linux conocida como **PDFtk Server** que nos permite fusionar documentos PDF, dividir PDF en nuevos documentos, rotar páginas, descifrar y cifrar, actualizar metadatos, aplicar marcas de agua y mucho más. Vemos sus características.

- Combinar documentos PDF o clasificar escaneos de páginas PDF.
- Dividir páginas PDF en un nuevo documento.
- Rotar documentos o páginas PDF.
- Cifrar o descifrar documentos con contraseña.
- Rellenar formularios PDF con datos X / FDF.
- Generar plantillas FDF desde formularios PDF.
- Aplicar marca de agua en fondo o un sello en primer plano.
- Generar informes de métricas, marcadores y metadatos del documento.
- Agregar o actualizar marcadores o metadatos PDF.
- Adjuntar archivos a páginas y documentos PDF.
- Desempaquetar archivos adjuntos.
- Descomprimir y recomprimir secuencias de páginas.
- Reparar documentos PDF dañados (si es posible).



Combinar archivos pdf con PDFtk en uno solo

Lamentablemente, la herramienta ha desaparecido de los repositorios oficiales de muchas **distribuciones linux**. Es una consecuencia del daño que está haciendo Canonical y sus **paquetes snap**, al intentar centralizar todo en una misma plataforma de la que ellos son propietarios (parecemos ovejas).

Es cierto que podrías instalarlo desde su código fuente, pero claro... cada distribución tiene su propio método y sus propias librerías, así que hacemos una excepción e instalamos la herramienta desde snap.

```
sudo snap install pdftk
```

```
sergio@demo-sololinux:~$ sudo snap install pdftk
Se ha instalado pdftk 2.02-4 por Scott Moser (smoser)
sergio@demo-sololinux:~$
```

La sintaxis para combinar varios documentos o archivos en uno, es muy simple. Observa.

```
pdftk archivo1.pdf archivo2.pdf archivo3.pdf cat output
nuevo-archivo.pdf
```

En nuestro ejemplo de uso, vamos a fusionar los cinco primeros números de la **revista SoloLinux**. En nuestro caso nos movemos a la carpeta descargas y, para una mayor comodidad (copiar y pegar) listamos el contenido.

```
sergio@demo-sololinux:~/Descargas$ ls
sergio@demo-sololinux:~/Descargas$ ls
REVISTA_SOLOLINUX_N01_FEBRERO_2019.pdf  REVISTA_SOLOLINUX_N04_MAYO_2019.pdf
REVISTA_SOLOLINUX_N02_MARZO_2019.pdf    REVISTA_SOLOLINUX_N05_JUNIO_2019.pdf
REVISTA_SOLOLINUX_N03_ABRIL_2019.pdf
sergio@demo-sololinux:~/Descargas$
```

Ahora ejecutamos el siguiente comando.

```
pdftk REVISTA_SOLOLINUX_N01_FEBRERO_2019.pdf
REVISTA_SOLOLINUX_N02_MARZO_2019.pdf
REVISTA_SOLOLINUX_N03_ABRIL_2019.pdf
REVISTA_SOLOLINUX_N04_MAYO_2019.pdf
REVISTA_SOLOLINUX_N05_JUNIO_2019.pdf cat output 5-numeros-
de-sololinux.pdf
```



Puedes aprender más sobre la herramienta, ejecutando el comando en modo simple.

```
pdftk
```

Verás algo similar a lo siguiente.

```
sergio@demo-sololinux:~$ pdftk
SYNOPSIS
pdftk <input PDF files | - | PROMPT>
[ input_pw <input PDF owner password | PROMPT> ]
[ <operation> <operation arguments> ]
[ output <output filename | - | PROMPT> ]
[ encrypt_40bit | encrypt_128bit ]
[ allow <permissions> ]
[ owner_pw <owner password | PROMPT> ]
[ user_pw <user password | PROMPT> ]
[ flatten ] [ need_appearances ]
[ compress | uncompress ]
[ keep_first_id | keep_final_id ] [ drop_xfa ] [ drop_xmp ]
[ verbose ] [ dont_ask | do_ask ]
Where:
<operation> may be empty, or:
[ cat | shuffle | burst | rotate |
generate_fdf | fill_form |
background | multibackground |
stamp | multistamp |
dump_data | dump_data_utf8 |
dump_data_fields | dump_data_fields_utf8 |
dump_data_annots |
update_info | update_info_utf8 |
attach_files | unpack_files ]
For Complete Help: pdftk --help
sergio@demo-sololinux:~$
```

Para más ayuda usa help.

```
pdftk --help
```

Cambiar de interfaz gráfica a línea de comandos con TTY

TTY es el acrónimo de **TeleTYpewriter**, y tomamos su nombre porque en los comienzos de UNIX, los terminales de usuario que conectaban a viejas computadoras eran en realidad... teletipos comunes o teletipos electromecánicos, también conocidos como máquinas tty.

Desde entonces, el nombre TTY se familiarizó con la línea de comandos hasta nuestros días, hasta tal punto que se sigue usando en consolas de texto. Actualmente, la cosa ha cambiado y todas las consolas basadas en texto son consolas virtuales, en vez de las vetustas consolas físicas. Por otro lado, tenemos el **comando tty** que imprime el nombre de archivo del terminal conectado a la entrada estándar.

Las **funciones TTY** son muchas y variadas, pero nosotros (en este artículo) nos vamos a centrar en como saltar de tu entorno de escritorio gráfico, a la línea de comandos de forma sencilla, rápida y reiniciando la sesión.

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

Escrito por David MacKenzie.
sergio@sololinux:~$
```



Ahora supongamos que usas el entorno de escritorio **Gnome**. No es algo común, pero puede suceder que en un momento dado el sistema Gnome se queda congelado y no hay nada que hacer. Bueno, TTY es la solución a tu problema.

Pulsas la combinación de teclas **Ctrl + Alt + F2 (tecla de función)**, automáticamente se abre la línea de comandos que como podrás comprobar es diferente a tu emulador de terminal. Si es necesario inserta tu nombre de usuario y contraseña.

En nuestro ejemplo de uso queremos reiniciar Gnome, por tanto deberás ejecutar el siguiente comando.

```
sudo killall -1 gnome-shell
```

Una vez ejecutado el comando, pulsas la combinación de teclas **Ctrl + Alt + F7 (tecla de función)** para retornar a la interfaz gráfica de Gnome. Problema solucionado.

```
Ubuntu 18.04.5 LTS demo-sololinux tty2
www.sololinux.es
demo-sololinux login: sergio
Password:
Last login: Wed Sep 8 12:41:28 CEST 2021 on tty2
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.14.0-051400-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

0 updates can be applied immediately.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
sergio@demo-sololinux:~$ sudo killall -1 gnome-shell
[sudo] contraseña para sergio:
sergio@demo-sololinux:~$
```

Cambiar de interfaz gráfica a línea de comandos con TTY

Puedes pensar que no tiene ningún sentido este artículo, que si quieras usar la **línea de comandos** ya tienes la terminal que te ofrece tu escritorio. Bueno... ya verás como ahora mismo comprenderás la utilidad de TTY y, como se ejecuta en la siguiente tabla.

TTY	Uso final	Secuencia de acceso
tty1	Línea de comandos	Ctrl + Alt + F1
tty2	Línea de comandos	Ctrl + Alt + F2
tty3	Línea de comandos	Ctrl + Alt + F3
tty4	Línea de comandos	Ctrl + Alt + F4
tty5	Línea de comandos	Ctrl + Alt + F5
tty6	Línea de comandos	Ctrl + Alt + F6
tty7	Interfaz gráfica (escritorio)	Ctrl + Alt + F7

Con el **comando who** puedes identificar las ejecuciones, así como el estado actual. En la siguiente imagen, verás que la actual es tty7.

```
sergio@sololinux:~$ who
sergio  tty1          2021-09-08 18:32
sergio  tty7          2021-09-08 10:02 (:0)
sergio  tty2          2021-09-08 10:47
sergio@sololinux:~$
```

Otro caso realmente interesante. El otro día estaba probando **VirtualBox** en un equipo bastante antiguo, claro... pasó lo que tenía que pasar. El consumo de la cpu aumento desmesuradamente, hasta proporciones de volver al sistema totalmente inútil a mis consultas.

La solución es similar a la anterior, pulsando la combinación de teclas **Ctrl + Alt + F2 (tecla de función)** accedí a la línea de comandos. Una vez en ella ejecuto la **herramienta htop**, para identificar el PID.

```
PID USER      PRT   NI  VIRT   RES   SHM   CPU% MEM% TIME% Command
1535 sergio    20   0 3489M 233M 130M S 96.4 6.1 1:59.62 /usr/lib/virtualbox/VirtualBoxVM --com
1568 sergio    21   1 3489M 233M 130M S 71.7 6.1 0:54.24 /usr/lib/virtualbox/VirtualBoxVM --com
```

Una vez identificado en número PID (1535), la solución resultó tan simple como ejecutar el siguiente comando.

```
sudo kill -1 1535
```

Pulsando la combinación de teclas **Ctrl + Alt + F7 (tecla de función)**, retornamos a nuestro entorno de escritorio con el problema solucionado.

Conec^rar un iPhone como unidad externa en Ubuntu 20.04

Hasta no hace mucho tiempo, conectar un iPhone como unidad externa en Ubuntu era una tarea complicada. El soporte que ofrecía Linux para los dispositivos iOS, como por ejemplo el iPhone, iPad e iPod, no fue de los mejores. Situación lógica, dado el trato con el que maneja sus controladores privativos.

Por suerte la situación ha cambiado, y hoy en día si es posible. En este artículo de hoy, vemos como montar el almacenamiento de tu iPhone (**iPad**, etc...) en **Ubuntu**, **Linux Mint** y derivados, de forma que se detecte como un nuevo dispositivo de almacenamiento externo.



Conectar un iPhone como unidad externa en Ubuntu 20.04

Para que el iPhone y otros dispositivos iOS sean reconocidos en Ubuntu, es necesaria la instalación de la biblioteca **libimobiledevice**. Debido a las medidas de seguridad de Apple, la librería Libimobiledevice es la única forma de obtener acceso a la información del dispositivo externo, así como a su almacenamiento interno.

Comenzamos instalando las librerías. Si tienes una versión actualizada, es posible que ya la tengas, aun siendo así también son necesarias sus utilidades.

```
sudo apt install libimobiledevice6 libimobiledevice-utils

sergio@demo-sololinux:~$ sudo sudo apt install
libimobiledevice6 libimobiledevice-utils
[sudo] contraseña para sergio:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
libimobiledevice6 ya está en su versión más reciente
(1.2.1~git20191129.9f79242~1build1).
fijado libimobiledevice6 como instalado manualmente.
Se instalarán los siguientes paquetes NUEVOS:
 libimobiledevice-utils
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0
no actualizados.
Se necesita descargar 88,1 kB de archivos.
Se utilizarán 509 kB de espacio de disco adicional después
de esta operación.
¿Desea continuar? [S/n]
```

Una vez que hayas instalado las librerías, conecta tu iPhone al sistema. Si el sistema de archivos del **iPhone** no se monta automáticamente, deberás emparejar el dispositivo **iPhone** con Ubuntu. Ejecuta lo siguiente en la terminal.

idevicepair pair

Si todo es correcto recibirás un mensaje de éxito, pero aún debes permitir las múltiples conexiones entre el dispositivo y la **instalación de Ubuntu**. Para lograr tu

objetivo, ejecuta el siguiente comando.

usbmuxd -f -v

Si por el contrario es imposible emparejar, y recibes error de dispositivo no encontrado como el de la siguiente imagen; mejor nos decantamos por la instalación de **iFuse**.

```
sergio@demo-sololinux:~$ sergio@demo-sololinux:~$ idevicepair pair  
No device found, is it plugged in?  
sergio@demo-sololinux:~$
```

El paquete iFuse también instala la librería Libimobiledevice, y como en el caso anterior permite montar un iPhone y que acceda al sistema de archivos Linux, entre otras cosas. IFuse viene incluido en los [repositorios oficiales de Ubuntu](#), por tanto su instalación es bastante simple.

```
sudo apt install ifuse
```

Una vez que hayas instalado las librerías, conecta tu **iPhone** al sistema. Si el sistema de archivos del iPhone no se monta automáticamente, deberás emparejar el dispositivo **iPhone** con Ubuntu. Ejecuta lo siguiente en la terminal.

```
idevicepair pair  
usbmuxd -f -v
```



En el extraño caso de que **iFuse** también de problemas, reinicia el sistema con el dispositivo conectado, creas su directorio manualmente y de igual forma lo montas.

```
sudo reboot
sudo mkdir /media/iphone
ifuse /media/iphone
```

Una vez montado, deberías poder ver el sistema de archivos como de cualquier otro dispositivo. Copiar, cortar, pegar, agregar, modificar, y mucho más.

Recuerda que de la misma forma que has montado la unidad manualmente, también lo puedes desmontar de manera segura ejecutando este comando en tu consola / terminal.

```
ifuse -u /media/iphone
```



designed by  freepik

www.sololinux.es

Crear un entorno virtual Python en Ubuntu 20.04



Un entorno virtual Python es un directorio que contiene la instalación completa de una versión de Python, en el cual se incluyen varios paquetes específicos así como todos los módulos adicionales que resulten necesarios.

Se permite crear varios entornos, de forma que nos permita tener un número ilimitado de versiones y **módulos de Python** diferentes.

Todo esto independientemente de la versión principal de Python instalada en el sistema. Su utilidad es grandiosa, sobre todo para desarrolladores, ya que permite instalar aplicaciones que requieren de diferentes versiones y módulos de Python, como por ejemplo... **Odoo, Django o Flask.**

La instalación y puesta en marcha del entorno virtual, es un proceso bastante rápido y sencillo que te tomara como mucho 10 minutos. En el artículo de hoy vemos lo sencillo que es crear, lo que también se conoce como Python virtual environment.



Crear un entorno virtual Python en Ubuntu 20.04

Como es habitual, lo primero que hacemos es actualizar nuestro sistema.

```
sudo apt update  
sudo apt upgrade
```

Ahora verificamos la versión actual de Python, instalada en nuestro sistema.

```
python3 -V  
  
sergio@demo-sololinux:~$ python3 -V  
Python 3.8.10  
sergio@demo-sololinux:~$
```

Para crear nuestro entorno necesitamos el paquete **python3-venv**, este paquete instalará el módulo **venv** junto con algunas dependencias necesarias. Recuerda que puedes usar cualquier versión Python que quieras, pero que esté disponible en el repositorio de Ubuntu.

```
sudo apt -y install python3-venv  
  
sergio@demo-sololinux:~$ sudo apt -y install python3-venv  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Se instalarán los siguientes paquetes adicionales:  
  python-pip-whl python3-distutils python3-lib2to3 python3.8-venv  
Se instalarán los siguientes paquetes NUEVOS:  
  python-pip-whl python3-distutils python3-lib2to3 python3-venv python3.8-venv  
0 actualizados, 5 nuevos se instalarán, 0 para eliminar y 0 no actualizados.  
Se necesita descargar 2.030 kB de archivos.  
Se utilizarán 4.465 kB de espacio de disco adicional después de esta operación.  
Des:1 http://es.archive.ubuntu.com/ubuntu focal-updates/universe amd64 python-pip-whl all 20.0.2-5ubuntu1.6 [1.805 kB]  
Des:2 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-lib2to3 all 3.8.10-0ubuntu1~20.04 [76,3 kB]  
Des:3 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-distutils all 3.8.10-0ubuntu1~20.04 [141 kB]  
Des:4 http://es.archive.ubuntu.com/ubuntu focal-updates/universe amd64 python3.8-venv amd64 3.8.10-0ubuntu1~20.04 [5.456 B]  
Des:5 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-venv amd64 3.8.2-0ubuntu2 [1.228 B]  
Descargados 2.030 kB en 1s (2.443 kB/s)  
Seleccionando el paquete python-pip-whl previamente no seleccionado.  
(Leyendo la base de datos ... 165252 ficheros o directorios instalados actualmente.)  
Preparando para desempaquetar .../python-pip-whl_20.0.2-5ubuntu1.6_all.deb ...  
Desempaquetando python-pip-whl (20.0.2-5ubuntu1.6) ...  
Seleccionando el paquete python3-lib2to3 previamente no seleccionado.  
Preparando para desempaquetar .../python3-lib2to3_3.8.10-0ubuntu1~20.04_all.deb ...  
Desempaquetando python3-lib2to3 (3.8.10-0ubuntu1~20.04) ...  
Seleccionando el paquete python3-distutils previamente no seleccionado.  
Preparando para desempaquetar .../python3-distutils_3.8.10-0ubuntu1~20.04_all.deb ...  
Desempaquetando python3-distutils (3.8.10-0ubuntu1~20.04) ...  
Seleccionando el paquete python3.8-venv previamente no seleccionado.  
Preparando para desempaquetar .../python3.8-venv_3.8.10-0ubuntu1~20.04_amd64.deb ...  
Desempaquetando python3.8-venv (3.8.10-0ubuntu1~20.04) ...  
Seleccionando el paquete python3-venv previamente no seleccionado.  
Preparando para desempaquetar .../python3-venv_3.8.2-0ubuntu2_amd64.deb ...  
Desempaquetando python3-venv (3.8.2-0ubuntu2) ...  
Configurando python-pip-whl (20.0.2-5ubuntu1.6) ...  
Configurando python3-lib2to3 (3.8.10-0ubuntu1~20.04) ...  
Configurando python3-distutils (3.8.10-0ubuntu1~20.04) ...  
Configurando python3.8-venv (3.8.10-0ubuntu1~20.04) ...  
Configurando python3-venv (3.8.2-0ubuntu2) ...  
Procesando disparadores para man-db (2.9.1-1) ...  
sergio@demo-sololinux:~$
```

```
window = Toplevel()  
  
window.title('Update Entry')  
window.geometry('650x475')  
  
self.define_grid(window, 10, 10)  
  
entry = db.return_entry(title)[0]
```



Para crear nuestro primer entorno virtual de Python, inserta un nuevo directorio de trabajo que se creara de forma automática. En nuestro caso, hemos elegido... «**my_python**». Ejecuta el siguiente comando.

```
sudo python3 -m venv /opt/my_python
```

Verifica que se han generado los archivos necesarios.

```
ls /opt/my_python
```

La carpeta debe incluir estos archivos...

```
sergio@demo-sololinux:~$ ls /opt/my_python
bin  include  lib  lib64  pyvenv.cfg  share
sergio@demo-sololinux:~$
```

Bien, solo falta habilitar el entorno, para ello accedemos al directorio y lo activamos. No olvide cambiar «my-python», por el nombre que tú hayas elegido.

```
sudo su
cd /opt/
source my_python/bin/activate
```

Automáticamente accedemos al nuevo entorno, observa que su nombre aparece al principio de cada línea de nuestra terminal / consola.

```
sergio@demo-sololinux:~$ sudo su
[sudo] contraseña para sergio:
root@demo-sololinux:/home/sergio# cd /opt/
root@demo-sololinux:/opt# source my_python/bin/activate
(my_python) root@demo-sololinux:/opt#
```

Una vez habilitado, ya puedes comenzar a trabajar en el nuestro proyecto. También puedes instalar y usar cualquier aplicación Python, incluyendo nuevos módulos. En el ejemplo instalamos el módulo IDNA, que ofrece soporte para el protocolo de nombres de dominio internacionalizados en aplicaciones (IDNA), como se especifica en [RFC 5891](#).

```
pip install idna
```

```
(my_python) root@demo-sololinux:~# pip install idna
Collecting idna
  Downloading idna-3.2-py3-none-any.whl (59 kB)
    |██████████| 59 kB 985 kB/s
Installing collected packages: idna
Successfully installed idna-3.2
```

```
sergio@demo-sololinux:~$ sudo su
[sudo] contraseña para sergio:
root@demo-sololinux:/home/sergio# cd /opt/
root@demo-sololinux:/opt# source my_python/bin/activate
(my_python) root@demo-sololinux:/opt# cd
(my_python) root@demo-sololinux:~#
(my_python) root@demo-sololinux:~# pip install idna
Collecting idna
  Downloading idna-3.2-py3-none-any.whl (59 kB)
    |██████████| 59 kB 985 kB/s
Installing collected packages: idna
Successfully installed idna-3.2
(my_python) root@demo-sololinux:~#
```

Para deshabilitar temporalmente el entorno de trabajo virtual (sin borrarlo), ejecuta...

```
deactivate
```

Opera tal como en el ejemplo, para no tener problemas.

```
(my_python) root@demo-sololinux:~# deactivate
root@demo-sololinux:~#
```

Manjaro abandona Mozilla Firefox

Esto se veía venir, y después de que varias distribuciones linux con poca repercusión entre los usuarios Linux, se atrevieran a quitar el **navegador web** por excelencia de la comunidad Linux. Llega **Manjaro** y hace lo mismo, aunque debemos puntualizar que de momento parece una toma de contacto, pues a modo de primera experiencia se incluye **Vivaldi** como navegador web predeterminado, tan solo en la nueva versión Manjaro Cinnamon 21.1.2.

La noticia fue recibida por los desarrolladores de Vivaldi, en medio de una gran algarabía. Es la primera vez que la versión completa de una de las distribuciones más importantes, dejaba fuera a Mozilla Firefox. Esta vez solo ha sido **Cinnamon**, pero ya verás como tan solo es la primera ficha del dominó que ha caído, vendrán más. Aquí el comunicado traducido del equipo Vivaldi.

¡Hoy tenemos una gran noticia! Estamos orgullosos de compartir con vosotros que Vivaldi es ahora el navegador predeterminado de la distribución Linux basada en Arch, Manjaro en su edición Cinnamon que ahora ofrece una experiencia de navegación rápida, confiable y rica en funciones.

Cinnamon, una de las ediciones más populares de Manjaro, pues es un entorno de escritorio que funciona sin ningún problema con las nuevas funciones y la filosofía de diseño que identifica a Vivaldi.

La popular distribución Linux Manjaro se basa en Arch Linux, un sistema de lanzamiento continuo, rápido y personalizable (rolling). Manjaro ha introducido una capa de estabilidad, con la capacidad de manejar las actualizaciones de paquetes con un proceso de evaluación y corrección de errores. Posee herramientas dedicadas que simplifican enormemente la instalación, el manejo de hardware y la administración de paquetes, actualizaciones y copias de seguridad. Esto hace que Manjaro sea muy fácil de usar, ahora con Vivaldi aún más.

Manjaro abandona Mozilla Firefox

En este artículo no vamos a discutir las características de Vivaldi, pero si es cierto que tomo la base de **Cromium** y la mejoró, además de manera considerable. Por otro lado tenemos a Firefox, que le duela a quien le duela tiene un comportamiento y una forma de operar lenta y tosca.

En realidad pienso que las enormes contribuciones a la comunidad, por parte de la **Fundación Mozilla** han mantenido a Firefox al pie del cañón. Hoy en día esto ya es inadmisible, Mozilla Firefox funcionaba muy bien en monopestañas y con múltiples sesiones. Esto ya no basta para estar a la altura de los grandes navegadores, **Gecko** se ha quedado atrás y, no vemos grandes esperanzas de mejorar su desarrollo a no ser que cambien o desarrollen un nuevo motor. En la siguiente imagen de ejemplo, podemos observar que ya no existe Firefox. El navegador web predeterminado es Vivaldi.



Ejemplo de uso del navegador web Vivaldi, en Manjaro Cinnamon.



Qué es el instalador calamares en Linux

Calamares es una utilidad de instalación para distribuciones GNU / Linux. Es liviano, fácil de usar, pragmático, y funciona en la mayoría de **distribuciones Linux** modernas. Permite una configuración flexible y personalizada, a la hora de instalar tu sistema operativo linux sin tener que interactuar directamente con la línea de comandos.

Calamares es un ejemplo vivo de la filosofía del proyecto GNU, pues los desarrolladores de diversas y opuestas distribuciones Linux, decidieron unir sus esfuerzos para crear un instalador conjunto que fuera eficiente, rápido y totalmente configurable.

Si bien es cierto que su comunidad de desarrolladores ha crecido, en sus comienzos fue un producto casi exclusivo para KDE. El motivo es evidente, los primeros en participar fueron en exclusiva, desarrolladores de KDE que aportaron sus contribuciones desde: **BBQLinux**, **Fedora**, **KaOS**, **Kubuntu**, **Manjaro**, Maui, Netrunner y **OpenMandriva**. Hoy en día es utilizado por muchas distribuciones, algunas tan conocidas como Debian.



Qué es el instalador calamares en Linux

La mayor parte del código de la herramienta, está escrita en C++ 17, usando **Qt 5.9 +** como interfaz de usuario. Ofrece un diseño modular y Python 3 es su principal lenguaje de programación. La interfaz de Python 3 se implanta con **Boost Python**, y toda la configuración de Calamares se realiza en YAML. Tienes a tu disposición muchísimos módulos listos para usar, aunque se permite crear los tuyos propios.

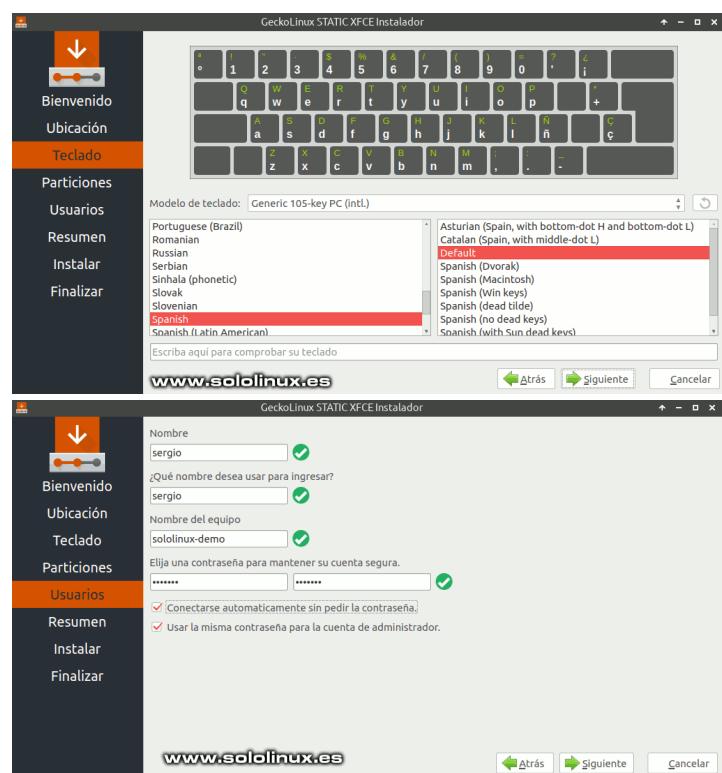
Debes saber que si tienes planeado crear tu propia distro, tienes la opción de modificar la apariencia del instalador, incluir ilustraciones y explicaciones del los pasos a seguir, el progreso, maximizar, etc. Todo ello añadiendo módulos, por ejemplo... en la distro **Arco Linux** tenemos un módulo que nos llama la atención, ya que aparece una pantalla en la cual puedes elegir el Kernel a instalar.

Elección del **kernel**, entornos de trabajo, herramientas de comunicación, multimedia, emuladores de terminal, gestores de archivos, etc. Todo depende del desarrollador, fabrica un instalador a tu gusto. Este instalador es totalmente gráfico, y se usa especialmente cuando se instala Linux desde una **Live**;

Es decir haciendo clic en el ícono del escritorio. Como no podía ser menos, admite multitud de idiomas, además de ofrecer la opción de configurarlo en la primera pantalla del instalador.

Zona horaria del sistema, marcar tu ciudad específica en el mapa, crear tablas de partición, discos de partición / SSD por megabyte, para cada sección, etiquetas, FS, claves y, mucho más. Una vez configurado con tus datos y personalizaciones, aparece la hora de resumen de la instalación, pulsas en aceptar y listo. Calamares es muy fácil de usar.

Vemos algunas imágenes de ejemplo.



Puedes descargar Calamares y sus módulos desde su repositorio oficial.

- [Descargar Calamares](#)
- [Descargar módulos de Calamares](#)

Instalar Rust en Debian 11 Bullseye



Rust es un lenguaje de programación de **código abierto**, centrado en la velocidad, la seguridad y el paralelismo. Este lenguaje es ampliamente utilizado por los desarrolladores, para crear una amplia gama de software como por ejemplo: motores de juegos, sistemas operativos, sistemas de archivos, componentes para un navegador y motores de simulación para realidad virtual.

Si lo analizamos sintácticamente vemos que es muy similar a **C++**, pero con la gran diferencia que Rust puede garantizar la seguridad de la memoria mediante el uso de un verificador que valida sus referencias. En el artículo de hoy vemos como instalar este impresionante lenguaje, en Debian 11 Bullseye y todos sus derivados.



Instalar Rust en Debian 11 Bullseye

Antes de comenzar, actualizamos nuestro sistema.

```
sudo apt update && sudo apt upgrade
```

Ahora las dependencias requeridas por Rust.

```
sudo apt install curl build-essential gcc make -y
```

```
Setting up libalgorithm-merge-perl (0.08-3) ...
Setting up libtsan0:amd64 (10.2.1-6) ...
Setting up libctf0:amd64 (2.35.2-2) ...
Setting up libxpm4:amd64 (1:3.5.12-2) ...
Setting up libgcc-10-dev:amd64 (10.2.1-6) ...
Setting up libgd3:amd64 (2.3.0-2) ...
Setting up cpp (4:10.2.1-1) ...
Setting up libc6-dev:amd64 (2.31-13) ...
Setting up libc-devtools (2.31-13) ...
Setting up binutils-x86_64-linux-gnu (2.35.2-2) ...
Setting up libstdc++-10-dev:amd64 (10.2.1-6) ...
Setting up binutils (2.35.2-2) ...
Setting up dpkg-dev (1.20.9) ...
Setting up gcc-10 (10.2.1-6) ...
Setting up g++-10 (10.2.1-6) ...
Setting up gcc (4:10.2.1-1) ...
Setting up g++ (4:10.2.1-1) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up build-essential (12.9) ...
Processing triggers for libc-bin (2.31-13) ...
Processing triggers for man-db (2.9.4-2) ...
Progress: [ 99%] [#########################################]
```

Una vez que haya completado la instalación requerida de paquetes, usamos la **herramienta curl** para descargar el **script de instalación de Rust**.

```
curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
```

Como puedes ver en el siguiente ejemplo, debes pulsar «1» para realizar una instalación estándar.

```
root@sololinux-demo:~# curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
info: downloading installer
Welcome to Rust!
This will download and install the official compiler for the Rust
programming language, and its package manager, Cargo.
Rustup metadata and toolchains will be installed into the Rustup
home directory, located at:
  /root/.rustup
This can be modified with the RUSTUP_HOME environment
variable.
The Cargo home directory located at:
  /root/.cargo
This can be modified with the CARGO_HOME environment
variable.
The cargo, rustc, rustup and other commands will be added to
Cargo's bin directory, located at:
  /root/.cargo/bin
This path will then be added to your PATH environment
variable by
modifying the profile files located at:
  /root/.profile
  /root/.bashrc
You can uninstall at any time with rustup self uninstall and
these changes will be reverted.
Current installation options:
  default host triple: x86_64-unknown-linux-gnu
  default toolchain: stable (default)
  profile: default
  modify PATH variable: yes
1) Proceed with installation (default)
2) Customize installation
3) Cancel installation
> 1
```

La instalación es rápida.

```
1) Proceed with installation (default)
2) Customize installation
3) Cancel installation
> 1

info: profile set to 'default'
info: default host triple is x86_64-unknown-linux-gnu
info: syncing channel updates for 'stable-x86_64-unknown-linux-gnu'
info: latest update on 2021-09-09, rust version 1.55.0 (c8dfcfe04 2021-09-06)
info: downloading component 'cargo'
info: downloading component 'clippy'
info: downloading component 'rust-docs'
info: downloading component 'rust-std'
info: downloading component 'rustc'
info: downloading component 'rustfmt'
info: installing component 'cargo'
info: installing component 'clippy'
info: installing component 'rust-docs'
17.0 MiB / 17.0 MiB (100%) 5.6 MiB/s in 2s ETA: 0s
info: installing component 'rust-std'
22.3 MiB / 22.3 MiB (100%) 12.7 MiB/s in 1s ETA: 0s
info: installing component 'rustc'
51.0 MiB / 51.0 MiB (100%) 13.9 MiB/s in 3s ETA: 0s
info: installing component 'rustfmt'
info: default toolchain set to 'stable-x86_64-unknown-linux-gnu'

stable-x86_64-unknown-linux-gnu installed - rustc 1.55.0 (c8dfcfe04 2021-09-06)

Rust is installed now. Great!
```

Antes de utilizar Rust, debes habilitar su entorno en la shell actual.

```
source ~/.profile
source ~/.cargo/env
```

Puedes buscar actualizaciones periódicamente con el siguiente comando.

```
rustup update
```

Ejemplo...

```
root@sololinux-demo:~# rustup update
info: syncing channel updates for 'stable-x86_64-unknown-linux-gnu'
info: checking for self-updates
  stable-x86_64-unknown-linux-gnu unchanged - rustc 1.55.0 (c8dfcfe04 2021-09-06)
info: cleaning up downloads & tmp directories
root@sololinux-demo:~#
```

Si por algún motivo, quieres desinstalar Rust...

```
rustup self uninstall
```

www.sololinux.es

Kali Linux 2021.3 - Listo para su descarga

Kali Linux 2021.3 es una **distribución Linux basada en Debian**, que viene con una tremenda colección de herramientas forenses y de seguridad. Siendo uno de los proyectos pentest más activos, cada año nos presentan 4 versiones para que siempre estemos a la última. Hace apenas unas horas, han lanzado la tercera versión de este año en curso 2021.

Como es habitual viene con muchas actualizaciones y grandes mejoras, entre las cuales debemos destacar el nuevo **OpenSSL** que se ha configurado para una compatibilidad más amplia. Esto permite el uso de protocolos heredados, lo que quiere decir que se permite la comunicación con sistemas más antiguos que los usados en la actualidad. Esta novedad es importante, ya que nos permite buscar superficies de ataque adicionales generadas por aplicaciones obsoletas, normalmente olvidadas por el usuario.



Kali Linux 2021.3 – Listo para su descarga

El uso de Kali (modo live) en entornos de virtualización (VMware, VirtualBox, Hyper-V y QEMU + Spice), se ha simplificado con una nueva configuración de Kali, en el modo de sesión mejorado de **Hyper-V**.

Los desarrolladores también lanzaron la primera versión de **Kali NetHunter** para un reloj inteligente: el TicHunter Pro (con **Wear OS**). Esto aún está en versión experimental, por tanto, sus características se limitan a ataques USB y algunas funciones básicas. Las mejoras futuras previstas incluyen soporte para Nexmon y, el uso interno de Bluetooth.

Kali Linux 2021.3 también viene con una serie de nuevas herramientas, realmente interesantes:

- Berate_ap – Orquestar puntos de acceso Wi-Fi no autorizados.
- CALDERA – Plataforma de emulación automatizada.
- EAPHammer – Ataques dirigidos contra redes Wi-Fi **WPA2-Enterprise**.
- HostHunter – Identificar y descubrir nombres de host usando técnicas OSINT.
- RouterKeygenPC – Generador de claves Wi-Fi WPA / WEP predeterminadas.
- Subjack – Adquisición de subdominios.
- WPA_Sycophant – Cliente para ataque de retransmisión EAP.

Los desarrolladores también han mejorado el soporte de imágenes de Kali ARM. El nuevo sitio >KALI Tools ha mejorado considerablemente respecto a su antecesor. Por otro lado, la nueva **distribución Linux KALI** también ha recibido mejoras interesantes a nivel visual, en su escritorio por defecto.

Puedes descargar Kali Linux 2021.3, desde su página oficial.

- Descargar Kali Linux 2021.3



Instalar php 8 en Debian 11 Bullseye



PHP es uno de los lenguajes de programación del lado de servidor, más populares y utilizados a nivel mundial. Según estadísticas oficiales, PHP está implantado en este año 2021 en el 79,1% de los sitios web más conocidos a nivel mundial. La serie 7.x sigue siendo la reina, con una cuota de mercado del 65,6%. Algo lógico, pues la mayoría de **CMS y frameworks**, como **WordPress, Magento** o Laravel, involucran el uso de PHP.

La última versión de **PHP** es la serie 8, que supone una actualización importante del lenguaje PHP. Nuevas y potentes características y optimizaciones, incluyendo los argumentos con nombre, tipos de unión, atributos, propiedades del constructor, expresiones de coincidencia, operador seguro null, JIT, mejoras en el sistema de tipos, además de un excelente manejo de errores y coherencias.

En el artículo de hoy veremos como instalar PHP 8.x, en Debian 11 Bullseye y sus derivados. Trataremos las tres opciones más habituales.

1. En módulo Apache
2. En módulo Apache con FPM
3. En Nginx



Instalar php 8 en Debian 11 Bullseye

Requisitos previos

Como es habitual, comenzamos actualizando el sistema.

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

También nos resultarán necesarias algunas dependencias, que procedemos a instalar inmediatamente.

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

```
sudo apt install apt-transport-https lsb-release ca-  
certificates
```

Agregar repositorio PHP 8

Antes de agregar el nuevo debemos importar e instalar la **clave GPG**.

```
sudo wget -O /etc/apt/trusted.gpg.d/php.gpg  
https://packages.sury.org/php/apt.gpg
```

```
root@sololinux-demo:~# sudo wget -O /etc/apt/trusted.gpg.d/php.gpg  
https://packages.sury.org/php/apt.gpg  
--2021-09-16 14:12:24-- https://packages.sury.org/php/apt.gpg  
Resolving packages.sury.org... 46.3.176.144  
Connecting to packages.sury.org (packages.sury.org)|46.3.176.144|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 1769 (1.7K) [application/octet-stream]  
Saving to: '/etc/apt/trusted.gpg.d/php.gpg'  
  
/etc/apt/trusted.gpg: 100%[=====] 1.73K --.KB/s in 0s  
2021-09-16 14:12:24 (46.3 MB/s) - '/etc/apt/trusted.gpg.d/php.gpg' saved [1769/1769]
```

www.sololinux.es

Ahora el repositorio.

```
sudo sh -c 'echo "deb https://packages.sury.org/php/ $  
(lsb_release -sc) main" > /etc/apt/sources.list.d/php.list'
```

```
root@sololinux-demo:~#  
root@sololinux-demo:~# sudo sh -c 'echo "deb  
https://packages.sury.org/php/ $(lsb_release -sc) main" >  
/etc/apt/sources.list.d/php.list'  
root@sololinux-demo:~#
```

Instalar PHP 8

En Apache

Para instalar PHP 8 como un módulo de Apache, ingresa el siguiente comando.

```
sudo apt update && sudo apt install php8.0 libapache2-mod-  
php8.0
```

Solo falta reiniciar el servicio, para que cargue el nuevo módulo.

```
sudo systemctl restart apache2
```

```
root@sololinux-demo:~# sudo apt update && sudo apt install php8.0 libapache2-mod-  
php8.0  
Hit:1 http://security.debian.org/debian-security bullseye-security InRelease  
Hit:2 http://ftp.debian.org/debian bullseye InRelease  
Hit:3 http://ftp.debian.org/debian bullseye-updates InRelease  
Get:4 https://packages.sury.org/php bullseye InRelease [6841 B]  
Get:5 https://packages.sury.org/php bullseye/main amd64 Packages [300 kB]  
Fetched 307 KB in 1s (598 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
3 packages can be upgraded. Run 'apt list --upgradable' to see them.  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following package was automatically installed and is no longer required:  
bsdmaintutils  
Use 'sudo apt autoremove' to remove it.  
The following additional packages will be installed:  
apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1  
libaprutil1-dbd-sqlite3 libaprutil1-ldap libjansson4 libluas5.3-0 libsodium23  
php-common php8.0-cli php8.0-common php8.0-opcache php8.0-readline  
Suggested packages:  
apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser  
php-pear  
The following NEW packages will be installed:  
apache2 apache2-bin apache2-data apache2-utils libapache2-mod-php8.0 libapr1  
libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libjansson4 libluas5.3-0  
libsodium23 php-common php8.0-cli php8.0-common php8.0-opcache  
php8.0-readline  
0 upgraded, 18 newly installed, 0 to remove and 3 not upgraded.  
Need to get 6678 kB of archives.  
After this operation, 28.5 MB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

MANUALES

En Apache con FPM

Para instalar PHP-FPM para Apache 2, debes aplicar los siguientes comandos. Recuerda que de forma predeterminada, PHP-FPM no está habilitado en Apache. Sigue todos los pasos que te indico, no tendrás ningún problema.

```
sudo apt update && sudo apt install php8.0-fpm libapache2-mod-fcgid  
sudo a2enmod proxy_fcgi setenvif && sudo a2enconf php8.0-fpm  
sudo systemctl restart apache2
```

Verificamos su estado.

```
sudo systemctl status php8.0-fpm  
  
root@sololinux-demo:~# sudo systemctl status php8.0-fpm  
● php8.0-fpm.service - The PHP 8.0 FastCGI Process Manager  
   Loaded: loaded (/lib/systemd/system/php8.0-fpm.service; enabled; vendor preset: enabled)  
   Active: active (running) since Thu 2021-09-16 14:31:45 CEST; 8min ago  
     Docs: man:php-fpm8.0(8)  
 Main PID: 18193 (php-fpm8.0)  
    Status: "Processes active: 0, idle: 2, Requests: 0, slow: 0, Traffic: 0req/sec"  
   Tasks: 3 (limit: 4915)  
  Memory: 10.6M  
  CGroup: /system.slice/php8.0-fpm.service  
          ├─18193 php-fpm: master process (/etc/php/8.0/fpm/php-fpm.conf)  
          ├─18194 php-fpm: pool www  
          └─18195 php-fpm: pool www  
  
www.sololinux.es
```

Sep 16 14:31:45 sololinux-demo systemd[1]: Starting The PHP 8.0 FastCGI Process...
Sep 16 14:31:45 sololinux-demo systemd[1]: Started The PHP 8.0 FastCGI Process...
[lines 1-15/15 (END)...skipping...

En Nginx

Nginx no ofrece soporte PHP de forma predeterminada, es necesario instalar PHP-FPM “administrador de procesos fastCGI” para manejar correctamente archivos PHP.

```
sudo apt install php8.0-fpm php8.0-cli -y  
sudo systemctl status php8.0-fpm
```

Resultado de la verificación del estado.

```
root@sololinux-demo:~# sudo systemctl status php8.0-fpm  
● php8.0-fpm.service - The PHP 8.0 FastCGI Process Manager  
   Loaded: loaded (/lib/systemd/system/php8.0-fpm.service; enabled; vendor preset: enabled)  
   Active: active (running) since Thu 2021-09-16 14:31:45  
           CEST; 19min ago  
     Docs: man:php-fpm8.0(8)  
 Main PID: 18193 (php-fpm8.0)  
    Status: "Processes active: 0, idle: 2, Requests: 0,  
           slow: 0, Traffic: 0req/sec"  
   Tasks: 3 (limit: 4915)  
  Memory: 10.6M  
  CGroup: /system.slice/php8.0-fpm.service  
          ├─18193 php-fpm: master process  
          │  (/etc/php/8.0/fpm/php-fpm.conf)  
          ├─18194 php-fpm: pool www  
          └─18195 php-fpm: pool www  
  
Sep 16 14:31:45 sololinux-demo systemd[1]: Starting The PHP 8.0 FastCGI Process Manager...  
Sep 16 14:31:45 sololinux-demo systemd[1]: Started The PHP 8.0 FastCGI Process Manager.  
root@sololinux-demo:~#
```

Nota: Si obtienes algún error al **instalar PHP 8 en Apache**, es porque no lo tienes totalmente instalado. Lo puedes solucionar ejecutando este comando.

```
sudo apt install apache2
```

```
root@sololinux-demo:~# apt install apache2  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  apache2-data apache2-utils  
Suggested packages:  
  apache2-suexec-pristine | apache2-suexec-custom www-browser  
The following NEW packages will be installed:  
  apache2 apache2-data apache2-utils  
0 upgraded, 3 newly installed, 0 to remove and 3 not upgraded.  
Need to get 0 B/252 kB of additional disk space.  
After this operation, 1986 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://security.debian.org/debian-security/bullseye-security/main amd64 apache2-data all 2.4.48-3.1+deb11u1 [160 kB]  
Get:2 http://security.debian.org/debian-security/bullseye-security/main amd64 apache2-utils amd64 2.4.48-3.1+deb11u1 [252 kB]  
20/23 0 B/s  
Fetched 678 kB in 0s (2487 kB/s)  
perl: warning: Setting locale failed.  
perl: warning: Please check that your locale settings:  
  LANGUAGE = (unset),  
  LC_ALL = (unset),  
  LC_TIME = "es_ES.UTF-8"  
  LC_MONETARY = "es_ES.UTF-8"  
  LC_ADDRESS = "es_ES.UTF-8"  
  LC_TELEPHONE = "es_ES.UTF-8"  
  LC_NAME = "es_ES.UTF-8"  
  LC_MEASUREMENT = "es_ES.UTF-8"  
  LC_IDENTIFICATION = "es_ES.UTF-8"  
  LC_NUMERIC = "es_ES.UTF-8"  
  LC_PAPER = "es_ES.UTF-8"  
  LANG = "es_ES.UTF-8"  
are supported and installed on your system.  
  
www.sololinux.es
```



designed by freepik

Uso del comando chage con ejemplos

El comando chage se utiliza para ver o modificar una cuenta de usuario, además de información variada como la caducidad de la contraseña, incluyendo el cambio de la fecha y más.

Ampliamente utilizado por los administradores de sistemas, en el artículo de hoy vemos sus formas de uso más comunes e interesantes. Incluyendo sus ejemplos prácticos que ayudan a una mejor comprensión.

chage command



Uso del comando chage con ejemplos

Antes de comenzar con los ejemplos de uso, debes tener presente que este comando requiere ser ejecutado por el **root**, o adquiriendo los permisos correspondientes escribiendo **sudo** antes del comando. La sintaxis es muy simple, observa.

```
sudo chage [opciones] usuario
```

Las opciones más habituales, son...

-d, --lastday ÚLTIMO_DÍA	establece el día del último cambio de la contraseña a ÚLTIMO_DÍA
-E, --expiredate FECHA_CAD	establece la fecha de caducidad a FECHA_CAD
-h, --help	muestra este mensaje de ayuda y termina
-i, --iso8601	use YYYY-MM-DD when printing dates
-I, --inactive INACTIVA	deshabilita la cuenta después de INACTIVA días de la fecha de caducidad
-l, --list	muestra la información de la edad de la cuenta
-m, --mindays DÍAS_MIN	establece el número mínimo de días antes de cambiar la contraseña a DÍAS_MIN
-M, --maxdays MAX_DAYS	set maximum number of days before password change to MAX_DAYS
-R, --root CHROOT_DIR	directory to chroot into
-W, --warndays DÍAS_AVISO	establece los días de aviso de expiración a DÍAS_AVISO

Muestra información relativa a la antigüedad de la cuenta

La opción «**-l**», nos imprime los datos requeridos.

```
sudo chage -l sergio
```

```
root@sololinux-demo:~# sudo chage -l sergio
Último cambio de contraseña :sep 14, 2021
La contraseña caduca : nunca
Contraseña inactiva : nunca
La cuenta caduca : nunca
Número de días mínimo entre cambio de contraseña : 0
Número de días máximo entre cambio de contraseña : 99999
Número de días de aviso antes de que caduque la contraseña : 7
root@sololinux-demo:~#
```

Modificar la fecha de modificación de contraseña

En este caso, utiliza esta sintaxis.

```
sudo chage -d YYYY-MM-DD usuario
```

```
root@sololinux-demo:~# sudo chage -d 2022-12-01 sergio
root@sololinux-demo:~# sudo chage -l sergio
Último cambio de contraseña :dic 01, 2022 <<<-----ejemplo
La contraseña caduca : nunca
Contraseña inactiva : nunca
La cuenta caduca : nunca
Número de días mínimo entre cambio de contraseña : 0
Número de días máximo entre cambio de contraseña : 99999
Número de días de aviso antes de que caduque la contraseña : 7
root@sololinux-demo:~#
```

MANUALES

Establecer fecha de caducidad de una cuenta

```
sudo chage -E YYYY-MM-DD usuario
```

```
root@sololinux:~# sudo chage -E 2024-12-31 sergio
root@sololinux:~#
Último cambio de contraseña : sep 02, 2020
La contraseña caduca : nunca
Contraseña inactiva : nunca
La cuenta caduca : dic 31, 2024
Número de días mínimo entre cambio de contraseña : 0
Número de días máximo entre cambio de contraseña : 99999
Número de días de aviso antes de que caduque la contraseña : 7
root@sololinux:~#
```

Mínimo número de días entre cambios de contraseña

Para no controlar el número de días entre cambios de password, puedes usar el valor 0.

```
sudo chage -m NUM_DAYS usuario
```

```
root@sololinux:~# sudo chage -m 0 sergio
root@sololinux:~#
root@sololinux:~# sudo chage -l sergio
Último cambio de contraseña : sep 02, 2020
La contraseña caduca : nunca
Contraseña inactiva : nunca
La cuenta caduca : dic 31, 2024
Número de días mínimo entre cambio de contraseña : 0
Número de días máximo entre cambio de contraseña : 99999
Número de días de aviso antes de que caduque la contraseña : 7
root@sololinux:~#
```

Máximo número de días entre cambios de contraseña

Para no controlar el número de días entre cambios de password, puedes usar el valor 0.

```
sudo chage -M NUM_DAYS usuario
```

```
root@sololinux:~# sudo chage -M 0 sergio
root@sololinux:~#
root@sololinux:~# sudo chage -l sergio
Último cambio de contraseña : sep 02, 2020
La contraseña caduca : sep 02, 2020
Contraseña inactiva : nunca
La cuenta caduca : dic 31, 2024
Número de días mínimo entre cambio de contraseña : 0
Número de días máximo entre cambio de contraseña : 0
Número de días de aviso antes de que caduque la contraseña : 7
root@sololinux:~#
root@sololinux:~#
```

Días de advertencia, antes de que caduque la contraseña

```
sudo chage -W NUM_DAYS usuario
```

En el ejemplo, insertamos tres días.

```
root@sololinux:~# sudo chage -W 3 sergio
root@sololinux:~#
root@sololinux:~# sudo chage -l sergio
Último cambio de contraseña : sep 02, 2020
La contraseña caduca : sep 02, 2020
Contraseña inactiva : nunca
La cuenta caduca : dic 31, 2024
Número de días mínimo entre cambio de contraseña : 0
Número de días máximo entre cambio de contraseña : 0
Número de días de aviso antes de que caduque la contraseña : 3
root@sololinux:~#
```

Uso de chage sin opciones

Si quieres configurar todos los valores, lo más práctico es usar el comando sin opciones. El sistema nos pregunta todos los valores uno por uno.

```
sudo chage [usuario]
```

```
root@sololinux-demo:~# sudo chage sergio
Cambiando la información de la edad para sergio
Introduzca el nuevo valor, o pulse INTRO para usar el valor predeterminado
Duración mínima de la contraseña [0]: 5
Duración máxima de la contraseña [99999]: 365
Último cambio de contraseña (AAAA-MM-DD) [2022-12-01]: 2021-12-31
Aviso de caducidad de la contraseña [-1]: 5
Contraseña inactiva [-1]: 3
Fecha de caducidad de la cuenta (AAAA-MM-DD) [-1]: 0
root@sololinux-demo:~#
```

Obtenemos este resultado.

```
root@sololinux-demo:~# sudo chage -l sergio
Último cambio de contraseña : dic 31, 2021
La contraseña caduca : dic 31, 2022
Contraseña inactiva : ene 03, 2023
La cuenta caduca : ene 01, 1970
Número de días mínimo entre cambio de contraseña : 5
Número de días máximo entre cambio de contraseña : 365
Número de días de aviso antes de que caduque la contraseña : 5
root@sololinux-demo:~#
```

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

Chat de SoloLinux en Telegram



designed by freepik

Instalar Opera en Debian 11 y todos sus derivados



En los últimos tiempos, estamos viviendo **drásticos cambios en el mundo de los navegadores web** para Linux y, otros sistemas operativos. Desde la decisión de Chrome de bloquear la API a Chromium, con el único fin de entorpecer el correcto funcionamiento de los clones del mismo Chromium, hasta la valiente iniciativa de los desarrolladores de **Manjaro Cinnamon al prescindir de Firefox**, usando como alternativa el navegador Vivaldi.

Las imposiciones con tufillo a mafia, por parte de **Canonical** tampoco se quedan atrás. Todos recordáis el revuelo que supuso la decisión, de forzar la instalación de **Chromium** mediante Snap. Pues ahora parece que vienen curvas otra vez, porque de nuevo Canonical con su producto estrella (Ubuntu), va a realizar la misma práctica pero con Mozilla Firefox.

Mientras esperamos a ver como termina todo esto y, también pendientes de una comparativa de navegadores que por motivos de salud aún no ha visto la luz. Vamos a ver como instalar en Debian 11, uno de esos navegadores que funciona muy bien, pero que no tiene un gran impacto en Linux. Hablamos de Opera, el **navegador web** que podríamos comparar con ese buen vecino, que siempre saluda y nunca tira las colillas al suelo.



Instalar Opera en Debian 11 y todos sus derivados

Antes de comenzar actualiza tu Debian 11 o Debian 10.

```
sudo apt-get update
sudo apt-get upgrade
```

Es posible que tu distribución no tenga la **herramienta curl** instalada por defecto, la instalas con este comando.

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

```
root@sololinux-demo:# root@sololinux-demo:# sudo apt install -y curl      www.sololinux.es
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
curl ya está en su versión más reciente (7.74.0-1.3+b1).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
root@sololinux-demo:#
```

Bien, ahora dependiendo de tu versión Debian (11 / 10), importa la key publica y agrega el repositorio que corresponda.

Debian 11

```
curl -fsSL https://deb.opera.com/archive.key | sudo gpg --dearmor -o /usr/share/keyrings/operabrowser-keyring.gpg
echo "deb [signed-by=/usr/share/keyrings/operabrowser-keyring.gpg] https://deb.opera.com/opera-stable/ stable non-free" | sudo tee /etc/apt/sources.list.d/opera_stable.list
```

Debian 10

```
curl -fsSL https://deb.opera.com/archive.key | sudo apt-key add -
echo "deb https://deb.opera.com/opera-stable/ stable non-free" | sudo tee /etc/apt/sources.list.d/opera-stable.list
```

Actualizamos de nuevo.

```
sudo apt-get update
```

```
root@sololinux-demo:~# sudo apt-get update
Obj:1 http://deb.debian.org/debian bullseye InRelease
Obj:2 http://deb.debian.org/debian bullseye-updates
InRelease
Obj:3 http://security.debian.org/debian-security bullseye-security InRelease
Des:4 https://deb.opera.com/opera-stable stable InRelease
[2.590 B]
Des:5 https://deb.opera.com/opera-stable stable/non-free
amd64 Packages [1.133 B]
Descargados 3.723 B en 7s (517 B/s)

Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Todos los paquetes están actualizados.
root@sololinux-demo:~#
```

Bien, el repositorio agregado nos permite elegir entre el navegador **Opera Estable**, y el **Opera Beta**. No cabe ninguna duda que si lo quieres usar en producción, debes elegir la versión estable. Vemos como **instalar Opera**.

```
# Instalar Opera Estable
sudo apt-get install -y opera-stable
# Instalar Opera Beta
sudo apt-get install -y opera-beta
```

```
root@sololinux-demo:~# sudo apt-get install -y opera-stable
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Paquetes recomendados:
  pepperflashplugin-nonfree chromium-codecs-ffmpeg-extra
Se instalarán los siguientes paquetes NUEVOS:
  opera-stable
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 77,6 MB de archivos.
Se utilizarán 235 MB de espacio de disco adicional después de esta operación.
.....etc, etc, .....
```

MANUALES

Encontrarás el lanzador del navegador en el menú de aplicaciones, de tu entorno de escritorio preferido. Normalmente en el apartado «**Internet**». Vemos un par de imágenes de ejemplo.

Observa el Modo Claro.



Finalizamos con una imagen de ejemplo del Modo Oscuro.



www.sololinux.es

