

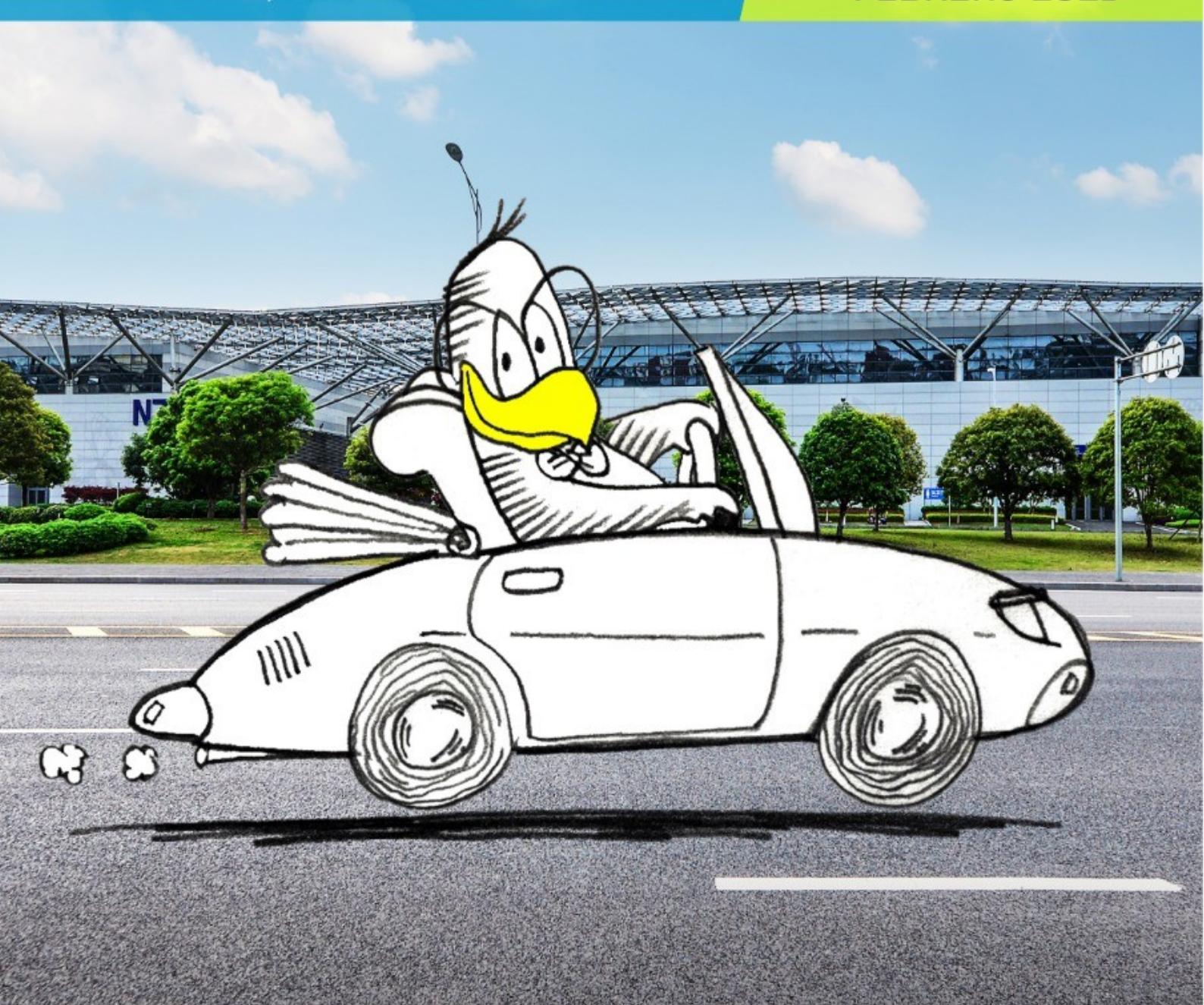
Visita nuestro sitio SoloLinux.es

MAGAZINE SÓLO LINUX

Nº
25

Tu revista, la revista de tod@s

FEBRERO 2021



Instalar **paquetes deb** en Ubuntu, Debian y derivados

Deshabilitar el Bluetooth en Linux con **systemctl**

8 formas de crear archivos en Linux

Uso del comando **printf** en Linux

Web Scraping con Python

Ocultar dispositivos montados en el dock de **Ubuntu 20.04**

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

EDITORIAL

BIENVENIDOS A LA REVISTA SOLOLINUX

Os presento el numero 25 de la revista SoLoLinux. Comenzamos con nuestro segundo año editorial. Desde SOLOLINUX esperamos que os guste este numero.

Igual que en números anteriores nos gustaría animar a todos nuestros lectores para que nos envíen sus **opiniones sobre el Software Libre o sobre GNU/Linux**, pueden enviarlo a adrian@sololinux.es, con ello queremos proponer que cada mes se publicada una o varias de esas opiniones sobre lo mencionado en la nueva sección de la revista *OPINIÓN DEL LECTOR*. **Queremos saber la opinión de todos.** Se intentara incluir el máximo de opiniones en cada numero, pero si no sale la tuya este mes no desesperes, al siguiente podría tener un hueco en la revista. **ANÍMENSE Y ENVÍEN SUS OPINIONES.** Gracias.

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

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

Gracias a TOD@s

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

Síguenos en las Redes:



La revista SOLOLINUX esta realizada con Libre Office Impress 7.0.0.3

www.sololinux.es

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

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

AYUDANOS A SEGUIR CRECIENDO



Editorial

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

Redacción

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

Diseño Portada

- Karina Fernández
[@karyfernandez.design](http://karyfernandez.design)

Agradecimientos

- Cayetano, Joseba por sus donaciones a la revista.
- Diego Caraballo, por su tiempo para la realización de la entrevista.

Publicidad

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

CON SOLOLINUX MULTIPLICARAS TUS CLIENTES

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

Contacto

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

- adrian@sololinux.es



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

MANUALES

- 7. Usar el comando apt-cache para buscar paquetes
- 10. Instalar paquetes deb en Ubuntu, Debian y derivados
- 12. Uso del comando Alias en linux
- 14. 8 formas de crear archivos en linux

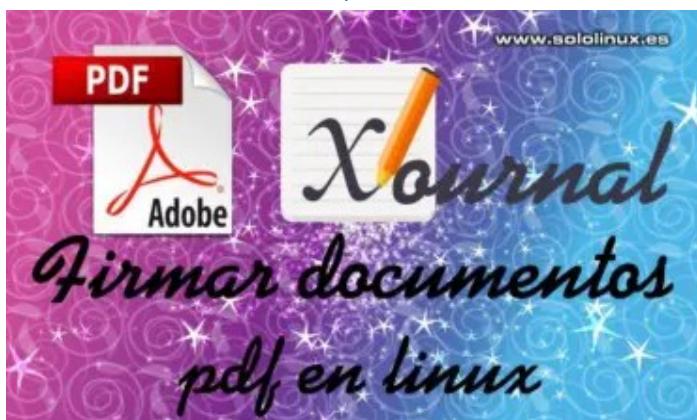


8 formas de crear archivos

- 16. Uso del comando printf en linux
- 18. Listar servicios con systemctl en linux
- 23. Ocultar dispositivos montados en el dock de Ubuntu 20.04
- 28. Instalar Etcher en Ubuntu 20.04 y otros linux



- 30. Como crear enlaces simbólicos en linux
- 32. Desinstalar MySQL Server en Ubuntu
- 38. Instalar Google Cloud SDK en Ubuntu 20.04
- 40. Como firmar documentos pdf en linux



- 46. Detectar cuellos de botella con el comando sysdig

HARDWARE

- 9. Deshabilitar el Bluetooth en Linux con systemctl



- 44. Raspberry Pi 3 vs Raspberry Pi 4

SCRIPTS

- 11. Colorear la salida del comando echo



- 21. Web Scraping con Python



CONTENIDO, NUMERO 25

DISTRIBUCIONES LINUX

13. Ubuntu 20.04.2 lanzado con el kernel 5.8
25. Instalar AlmaLinux Beta – El sucesor de CentOS 8

www.sololinux.es



33. Diferencias entre Ubuntu y Gentoo
35. Los 7 mejores derivados de Gentoo

www.sololinux.es



ENTREVISTA

48. Entrevista a Diego Caraballo, administrador del Blog Mi diario Python

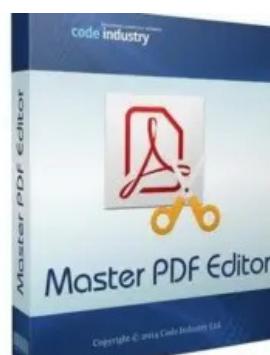
A screenshot of the Mi diario Python website. It features a header with the site's name and a navigation bar with links like 'Início' and 'Categorías'. Below the header are two main article snippets. The first snippet is titled 'Cómo ejecutar scripts Python automáticamente todos los días en Windows' and includes a progress bar graphic. The second snippet is titled 'Formateadores automáticos para utilizar en Python' and discusses BeautifulSoup. At the bottom of the page, there are two more sections: one on 'Barras de progreso en las aplicaciones Python con tqdm' and another on 'Web Scraping de datos COVID-19 usando BeautifulSoup en Python'.

SOFTWARE

20. Uso de Appliance en Ubuntu y otros linux
22. Instalar LeoCAD en linux



24. Ver listas IPTV online con TV-Lite
43. Master PDF Editor 4.3.89 – Gratis para linux



Master
PDF
Editor

OPINION DEL LECTOR

50. Opinion de Jose Luis desde México

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

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

AYUDANOS A SEGUIR
CRECIENDO



www.sololinux.es



CUMPLIMOS 10 AÑOS

Este año cumplimos 10 años. Gracias a todos los que venís confiando en nosotros para dar el salto a Linux (y a los que habéis confiando en nosotros, con el salto ya dado).

En 2011 creamos VANT y antes de acabar el año lanzamos nuestro primer portátil: El primer VANT MOOVE, con una CPU i3-2330m, hasta 8GB de RAM, 2.3Kg de peso y Ubuntu 11.04.

10 años hasta llegar a hoy, con una gama completa de sobremesas y portátiles para cubrir las necesidades de diferentes perfiles de usuario y con el ultrabook EDGE2 como tope de gama, con 1.1Kg de peso, procesadores i7-11065u, hasta 40GB de RAM y batería como para dejar tranquilamente el cargador en casa. Y con Ubuntu 20.04.2, o Linux Mint, o Debian, o KDE Neon, o...

Muchas cosas han cambiado en estos 10 años, salvo la ilusión con la que trabajamos, para convertir vuestra confianza en satisfacción, por muchos años.



la gama más completa de ordenadores linuxeros

descúbrelos en www.vantpc.es



@vantpc



vant.pc



vantpc_es



t.me/vantpc



INSCRIPCIÓN ABIERTA

2021

TÉCNICO LINUX 2021

CURSO LINUX SYSTEM ADMINISTRATOR

+
CERTIFICACIÓN UTN-FRD

+
22 CLASES MAGISTRALES ON LINE ¡DE REGALO!

+
WORKSHOPS LPIC 1 (101-102) ¡DE REGALO!



FABIÁN AMPALIO
Coordinador Gral.



GABRIEL CANEPA
Tutor



JOAQUÍN AMPALIO
Tutor



FRANCISCO SANDALINAS
Tutor



+34 669 584 870



adrian@sololinux.es



www.carreralinux.es


```
root@sololinux:~# sudo apt search mariadb
v libmariadb-client-lgpl-dev      -
v libmariadb-client-lgpl-dev:i386  -
v libmariadb-client-lgpl-dev-comp -
v libmariadb-client-lgpl-dev-comp -
p libmariadb-dev                 - MariaDB Connector/C, development files
p libmariadb-dev:i386              - MariaDB Connector/C, development files
p libmariadb-dev-compat           - MariaDB Connector/C, compatibility symlink
p libmariadb-dev-compat:i386       - MariaDB Connector/C, compatibility symlink
p libmariadb-java                - Java database driver for MariaDB and MySQL
p libmariadb3                     - MariaDB Connector/C
p libmariadb3:i386                - MariaDB Connector/C
p libmariadbclient               - MariaDB database development files
p libmariadbclient-dev:i386        - MariaDB database development files
p libmariadbclient-dev-compat     - MariaDB database development files (libmys
p libmariadbclient-dev-compat:i38 - MariaDB database development files (libmys
p libmariadbclient18              - MariaDB database client library
p libmariadbclient18:i386          - MariaDB database client library
p libmariadbdev                  - MariaDB embedded database, development fil
p libmariadbdev:i386              - MariaDB embedded database, development fil
p libmariadb18                   - MariaDB embedded database, shared library
p libmariadb18:i386              - MariaDB embedded database, shared library
p mariadb-client                 - MariaDB database client (metapackage depen
p mariadb-client-10.1             - MariaDB database client binaries
p mariadb-client-10.1:i386         - MariaDB database client binaries
p mariadb-client-core-10.1         - MariaDB database core client binaries
p mariadb-client-core-10.1:i386   - MariaDB database core client binaries
p mariadb-common                 - MariaDB common metapackage
v mariadb-common:i386            -
p mariadb-plugin-connect         - Connect storage engine for MariaDB
p mariadb-plugin-connect:i386     - Connect storage engine for MariaDB
p mariadb-plugin-cracklib-passwor - CrackLib Password Validation Plugin for Ma
p mariadb-plugin-cracklib-passwor - CrackLib Password Validation Plugin for Ma
p mariadb-plugin-gssapi-client    - GSSAPI authentication plugin for MariaDB c
p mariadb-plugin-gssapi-client:i3 - GSSAPI authentication plugin for MariaDB c
p mariadb-plugin-gssapi-server    - GSSAPI authentication plugin for MariaDB s
p mariadb-plugin-gssapi-server:i3 - GSSAPI authentication plugin for MariaDB s
p mariadb-plugin-mroonga          - Mroonga storage engine for MariaDB
p mariadb-plugin-mroonga:i386     - Mroonga storage engine for MariaDB
p mariadb-plugin-oqgraph          - OQGraph storage engine for MariaDB
p mariadb-plugin-oqgraph:i386     - OQGraph storage engine for MariaDB
p mariadb-plugin-spider          - Spider storage engine for MariaDB
p mariadb-plugin-spider:i386      - Spider storage engine for MariaDB
p mariadb-plugin-tokudb          - TokuDB storage engine for MariaDB
p mariadb-server                 - MariaDB database server <<<----- PAQUETE A INSTALAR
p mariadb-server-10.1             - MariaDB database server binaries
p mariadb-server-10.1:i386         - MariaDB database server binaries
p mariadb-server-core-10.1         - MariaDB database core server files
p mariadb-server-core-10.1:i386   - MariaDB database core server files
p mariadb-test                   - MariaDB database regression test suite
p mariadb-test:i386               - MariaDB database regression test suite
p mariadb-test-data              - MariaDB database regression test suite - d
v mariadb-test-data:i386          -
```

Como mi amigüete quiere instalar un **servidor MariaDB** (se incluyen el resto de paquetes), instalamos la aplicación que señalamos con una flecha en el ejemplo anterior.

```
sudo apt install mariadb-server
```

Imagen de ejemplo.

```
root@sololinux:~# sudo apt install mariadb-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
 galera-3 libaiol libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl libhtml-template-perl libjemalloc1 libterm-readkey-perl
 mariadb-client-10.1 mariadb-client-core-10.1 mariadb-common mariadb-server-10.1 mariadb-server-core-10.1 socat
Paquetes sugeridos:
 libmldb-perl libnet-daemon-perl libsql-statement-perl libipc-sharedcache-perl mariadb-test tinyca
Se instalarán los siguientes paquetes NUEVOS:
 galera-3 libaiol libconfig-initfiles-perl libdbd-mysql-perl libdbi-perl libhtml-template-perl libjemalloc1 libterm-readkey-perl
 mariadb-client-10.1 mariadb-client-core-10.1 mariadb-common mariadb-server mariadb-server-10.1 mariadb-server-core-10.1 socat
0 actualizados, 15 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 22,8 MB de archivos.
Se utilizarán 178 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] ■
```

www.sololinux.es

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

[Chat de SoloLinux en Telegram](#)

WWW.SOLOLINUX.ES

Deshabilitar el Bluetooth en Linux Con systemctl



Una particularidad de **Linux**, es que su **kernel** viene con toneladas de controladores para muchos dispositivos y tecnologías. Estos controladores se implantan como módulos del kernel predefinidos, por tanto, se instalan y habilitan en la gran mayoría de **distribuciones linux**.

Uno de estos módulos es el **Bluetooth**. La **tecnología Bluetooth** es popular, incluso viene habilitada por defecto en muchos de los dispositivos electrónicos que se comercializan actualmente. Pero... que quieras que te diga, a mí no me convence y no la quiero utilizar, además, por desconocimiento del usuario, es normal localizar dispositivos **Bluetooth** con una **key por defecto** que cualquiera puede identificar.

Este artículo de hoy, es para los que piensan o actúan como yo. Para qué quiero tener un servicio que no utilizo, si lo único que hace es consumir recursos. Vamos a deshabilitar el **Bluetooth** rápidamente y, tranquilo que si un día lo necesitas lo puedes habilitar de nuevo.



Deshabilitar el Bluetooth en Linux con systemctl

Los procesos del controlador Bluetooth, están habilitados de forma predeterminada y se inician automáticamente cada vez que encendemos nuestro sistema. Puedes verificarlo con el siguiente comando.

```
sudo systemctl status bluetooth.service
```

Si está habilitado por defecto, verás algo parecido a...

```
root@sololinux:~# sudo systemctl status bluetooth.service
● bluetooth.service - Bluetooth service
   Loaded: loaded (/lib/systemd/system/bluetooth.service; enabled;
   vendor preset: active (running) since Tue 2021-02-02 14:42:55 EET; 1h 7min ago
     Docs: man:bluetoothd(8)
   Main PID: 927 (bluetoothd)
     Status: "Running"
    Tasks: 1 (limit: 4521)
   CGroup: /system.slice/bluetooth.service
           └─927 /usr/lib/bluetooth/bluetoothd
feb 02 14:42:55 sololinux systemd[1]: Starting Bluetooth service...
feb 02 14:42:55 sololinux bluetoothd[927]: Bluetooth daemon 5.48
feb 02 14:42:55 sololinux bluetoothd[927]: Starting SDP server
feb 02 14:42:55 sololinux systemd[1]: Started Bluetooth service.
feb 02 14:42:55 sololinux bluetoothd[927]: Bluetooth management
interface 1.14 i
feb 02 16:21:31 sololinux bluetoothd[892]: Endpoint registered:
sender=:1.42 pat
feb 02 16:21:31 sololinux bluetoothd[892]: Endpoint registered:
sender=:1.42 pat
lines 1-17/17 (END)
```

Puedes deshabilitar el servicio rápidamente con:

```
sudo systemctl disable bluetooth.service
```

```
root@sololinux:~# sudo systemctl disable bluetooth.service
Synchronizing state of bluetooth.service with SysV service script with
/lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable bluetooth
Removed: /etc/systemd/system/dbus-org.bluez.service.
root@sololinux:~#
```

No te olvides de reiniciar linux, para que surta efecto.

```
sudo reboot
```

Una vez hayamos reiniciado nuestro sistema linux, verificamos que la operación «deshabilitar el Bluetooth», fue un éxito.

```
sudo systemctl status bluetooth.service
```

Como puedes ver en las siguientes líneas, hemos deshabilitado el Bluetooth con éxito.

```
root@sololinux:~# sudo systemctl status bluetooth.service
● bluetooth.service - Bluetooth service
  Loaded: loaded (/lib/systemd/system/bluetooth.service; disabled;
  vendor preset: active: inactive (dead)
    Docs: man:bluetoothd(8)
```

Si quieres habilitarlo de nuevo, tan solo debes ejecutar estos comandos (se reinicia el sistema).

```
root@sololinux:~# sudo systemctl status bluetooth.service
● bluetooth.service - Bluetooth service
  Loaded: loaded (/lib/systemd/system/bluetooth.service; disabled;
  vendor preset: active: inactive (dead)
    Docs: man:bluetoothd(8)
root@sololinux:~# sudo systemctl enable bluetooth.service
Synchronizing state of bluetooth.service with SysV service script with
/lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable bluetooth
Created symlink /etc/systemd/system/dbus-org.bluez.service → /lib/systemd/system/bluetooth.service.
root@sololinux:~#
```

www.sololinux.es

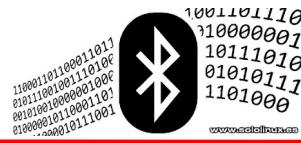
Para concluir, verifica de nuevo el servicio (debe estar habilitado de nuevo).

```
sudo systemctl status bluetooth.service
```

Se imprimirá en pantalla, algo similar a...

```
root@sololinux:~# sudo systemctl status bluetooth.service
● bluetooth.service - Bluetooth service
  Loaded: loaded (/lib/systemd/system/bluetooth.service; enabled;
  vendor preset: active: active (running) since Tue 2021-02-02 16:20:42 EET; 2min
54s ago
    Docs: man:bluetoothd(8)
   Main PID: 892 (bluetoothd)
     Status: "Running"
    Tasks: 1 (limit: 4521)
   CGroup: /system.slice/bluetooth.service
           └─892 /usr/lib/bluetooth/bluetoothd
feb 02 16:20:42 sololinux systemd[1]: Starting Bluetooth service...
feb 02 16:20:42 sololinux bluetoothd[892]: Bluetooth daemon 5.48
feb 02 16:20:42 sololinux systemd[1]: Started Bluetooth service.
feb 02 16:20:42 sololinux bluetoothd[892]: Starting SDP server
feb 02 16:20:42 sololinux bluetoothd[892]: Bluetooth management
interface 1.14 i
feb 02 16:21:31 sololinux bluetoothd[892]: Endpoint registered:
sender=:1.42 pat
feb 02 16:21:31 sololinux bluetoothd[892]: Endpoint registered:
sender=:1.42 pat
lines 1-17/17 (END)
```

Nota del autor: Si no haces un uso continuo del servicio de **Bluetooth**, es recomendable deshabilitarlo. Este tipo de servicios siempre están disponibles, por tanto su consumo es igual de alto como innecesario.



SOLOLINUX

Instalar paquetes deb en Ubuntu, Debian y derivados

Los **paquetes Deb**, son el formato de archivo oficial de un paquete de software instalable en **Debian**, **Ubuntu**, **Linux Mint** y todos sus derivados. Contienen todos los archivos necesarios, de la aplicación o herramienta a instalar, incluyendo los **metadatos**. Los **archivos de metadatos** contienen información específica, como el número de versión, dependencias, etc.

En realidad, ***.deb** es la extensión de estos archivos de paquete y **dpkg** es la herramienta específica que los instala. La utilidad «**dpkg**» es de bajo nivel, por tanto solemos usar algún **frontend** que trabaje sobre **dpkg**, como por ejemplo el conocido «**comando apt**».

En este artículo de hoy, veremos como instalar paquetes deb correctamente, en Ubuntu, Debian y derivados. Para ello usaremos el comando **dpkg**.



Instalar *.deb

Instalar paquetes deb en Ubuntu, Debian y derivados

Para instalar paquetes deb con la herramienta **dpkg**, ejecuta la siguiente sintaxis.

```
sudo dpkg -i paquete.deb
```

Para nuestro ejemplo, instalamos uno de los mejores editores de páginas web **WYSIWYG**, hablamos de **Bluegriffon**. Descargamos la herramienta y la instalamos con **dpkg**.

```
wget http://bluegriffon.org/freshmeat/3.1/bluegriffon-3.1.Ubuntu18.04-x86_64.deb
sudo dpkg -i bluegriffon-3.1.Ubuntu18.04-x86_64.deb
```

```
root@sololinux:~# wget http://bluegriffon.org/freshmeat/3.1/bluegriffon-3.1.Ubuntu18.04-x86_64.deb
--2021-02-03 14:09:53-
http://bluegriffon.org/freshmeat/3.1/bluegriffon-3.1.Ubuntu18.04-x86_64.deb
Resolviendo bluegriffon.org (bluegriffon.org)... 66.33.213.195
Conectando con bluegriffon.org (bluegriffon.org)[66.33.213.195]:80...
conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 283394478 (270M) [application/x-debian-package]
Guardando como: "bluegriffon-3.1.Ubuntu18.04-x86_64.deb"
bluegriffon-3.1.Ubu 100%[=====] 270,27M 3,96MB/s en 69s
2021-02-03 14:11:02 (3,94 MB/s) - "bluegriffon-3.1.Ubuntu18.04-x86_64.deb" guardado [283394478/283394478]
root@sololinux:~# sudo dpkg -i bluegriffon-3.1.Ubuntu18.04-x86_64.deb
Seleccionando el paquete bluegriffon previamente no seleccionado.
```

```
(Leyendo la base de datos ... 343548 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar bluegriffon-3.1.Ubuntu18.04-x86_64.deb ...
Añadiendo `desviación de /usr/bin/bluegriffon a /usr/bin/bluegriffon.ubuntu por bluegriffon'
Desempaquetando bluegriffon (3.1-2) ...
Configurando bluegriffon (3.1-2) ...
Procesando disparadores para gnome-menus (3.13.3-11ubuntu1.1) ...
Procesando disparadores para desktop-file-utils (0.23+linuxmint8) ...
Procesando disparadores para mime-support (3.60ubuntu1) ...
root@sololinux:~#
```

Es posible que al instalar otras herramientas específicas, nos aparezca un aviso comunicándonos que falta alguna dependencia. Para solucionar este problema, tan solo debes ejecutar el siguiente comando.

```
sudo apt -f install
```

Si es necesario, el **comando apt** instalará lo faltante. En el ejemplo siguiente, podemos observar que no es necesario instalar nada.

```
root@Sololinux:~# sudo apt -f install
Reading package lists... Done
Building dependency tree
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```



www.sololinux.es

Colorear la salida del comando echo



www.sololinux.es

El **comando echo** en Linux se usa para imprimir una **cadena** o, un conjunto de **cadenas** (textos) en la terminal de nuestro sistema linux. La cadena (suelen ser textos), se pasan como argumento.

Como es habitual en los **comandos linux**, echo también viene con variadas e interesantes opciones, que nos permiten manipular cómo se muestra la salida en pantalla. Puedes verlas en su manual integrado, con el siguiente comando.

```
man echo
```

Por otro lado, todos estamos de acuerdo que la terminal o consola de linux usa temas predeterminados, normalmente el texto es de color claro y el fondo oscuro. En un anterior artículo, vimos como personalizar la terminal de linux.

La salida del **comando echo**, también imprime el texto según el color del texto definido en el tema utilizado. Esto puede ser un poco aburrido y triste, a la hora de lanzar nuestros scripts. En este artículo de hoy, veremos como **colorear la salida del comando echo** dándole un toque más alegre, incluso divertido.

```
[~]$ echo
```

www.sololinux.es

Colorear la salida del comando echo

En **linux** tenemos soluciones para todo, en este caso, para modificar el color de la salida del **comando echo** usaremos secuencias de escape. Pero antes de comenzar, vemos la lista de colores más habitual.

Color	Color del texto	Color del fondo
Negro	30	40
Rojo	31	41
Verde	32	42
Amarillo	33	43
Azul	34	44
Magenta	35	45
Cian	36	46
Blanco	37	47

Las secuencias de escape pueden parecer un poco difíciles de manejar, pero todo lo contrario, son sencillas; tan solo debes acostumbrarte a usar cierto tipo de caracteres. Por ejemplo, la cadena de caracteres a imprimir debe ir precedida de esta secuencia de escape.

```
\e[1;m
```

Agregamos un color, en este caso el amarillo.

```
\e[1;33m
```

Nota: Con el «1» vemos colores vivos, también puedes cambiarlo por el «0» pero se imprimirán colores apagados.

Ahora vemos un ejemplo real. Observa que aparte de entre comillas insertamos «\e», no lo olvides nunca, ya que indica el comienzo de la secuencia de escape. Vemos un ejemplo insertado con el **comando echo**.

```
echo -e "\e[1;33mMe gusta SoloLinux"
```

```
sergio@sololinux:~$ echo -e "\e[1;33mMe gusta SoloLinux"
Me gusta SoloLinux
sergio@sololinux:~$
```

www.sololinux.es

Vemos otros ejemplos prácticos de uso, con la siguiente cadena de texto.

```
echo -e "\e[1;31mMe gusta SoloLinux"
```

En color rojo.
sergio@sololinux:~\$ echo -e «\e[1;31mMe gusta SoloLinux»
Me gusta SoloLinux

También es posible colorear la salida por partes, tan solo debes replicar la secuencia.

```
echo -e "\e[1;31mMe gusta \e[1;33mSoloLinux"
```

```
sergio@sololinux:~$ echo -e "\e[1;31mMe gusta \e[1;33mSoloLinux"
Me gusta SoloLinux
sergio@sololinux:~$
```

www.sololinux.es

Como punto final del artículo, no por ello menos interesante, vamos a colorear la salida, pero con una diferencia sustancial, usamos el código numérico de colores de fondo.

```
echo -e "\e[1;41mMe gusta SoloLinux"
```

Al revisar el resultado, vemos que el color de fondo no tiene fin.

```
sergio@sololinux:~$ echo -e "\e[1;41mMe gusta SoloLinux"
Me gusta SoloLinux
sergio@sololinux:~$
```

www.sololinux.es

Para evitar el error del color de fondo, es conveniente terminar con otro escape, pero... sin aplicar ningún código numérico de color.

```
echo -e "\e[1;41mMe gusta SoloLinux\e[1;m"
```

Ahora sí que tenemos un resultado correcto.

```
sergio@sololinux:~$ echo -e "\e[1;41mMe gusta SoloLinux\e[1;m"
Me gusta SoloLinux
sergio@sololinux:~$
```

www.sololinux.es

Uso del Comando Alias en linux



En sistemas operativos **Unix/Linux**, el comando **alias** indica a la **shell** que reemplace una cadena por otra, al ejecutar comandos predefinidos. Para que me entiendas mejor, permite sustituir una palabra o serie de palabras por otra.

Los alias son muy útiles para personalizar la interfaz de tu **terminal** o consola. Por ejemplo, los comandos o herramientas que más usas (incluyendo opciones e indicadores), los puedes invocar mediante el término que tu mismo elijas. Realmente, el comando alias puede ser de gran ayuda a la hora de mejorar tu productividad.

Debes tener presente, que los alias creados directamente en la shell no son permanentes, desaparecen al cerrar tu sesión. En este artículo, también aprenderemos a crear alias de forma permanente. Nosotros usaremos **bash** y **nano**, pero con otras opciones el proceso es similar.

[~]\$ alias

www.sololinux.es

Uso del comando Alias en linux

Nuestro sistema linux, ya viene con algunos alias predefinidos que pueden variar dependiendo de la **distribución linux** utilizada. Puedes verlos con el siguiente comando.

alias

Vemos dos ejemplos de salida, Ubuntu 20.04 Server y CentOS 7. Comenzamos por Ubuntu.

```
root@SoloLinux-demo:~# alias
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -alF'
alias ls='ls --color=auto'
root@SoloLinux:~#
```

En CentOS 7 difiere un poco.

```
[root@host ~]# alias
alias cp='cp -i'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l='ls -d . --color=auto'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
alias mv='mv -i'
alias rm='rm -i'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'
[root@host ~]#
```

Crear nuestro propio alias es realmente sencillo, tan solo debes usar esta síntesis. Es así de fácil.

```
alias mipalabra="comando"
```

Continuamos con un ejemplo básico, en el cual invocamos el **editor nano**. Si eres **administrador de sistemas**, seguro que usas algún editor de texto en consola, decenas de veces al día. Nosotros ponemos como ejemplo a nano, que como bien sabes lo puedes lanzar con un simple comando.

```
nano
nano archivo.txt
nano script.sh
```

Creamos un alias que nos evita tener que escribir nano, será suficiente la letra «**n**». Vamos a ello.

```
alias n="nano"
```

Listo, ves que fácil, ahora tan solo debes ejecutar la letra «**n**».

```
n
n archivo.txt
n script.sh
```

```
sergio@sololinux:~$ alias n="nano"
sergio@sololinux:~$ n
```

Borrar alias: Recuerda que este alias se mantiene activo mientras no cierres la sesión, si por cualquier motivo quieras borrarlo sin cerrar la sesión, ejecuta **unalias**.

```
unalias [palabra]
```

```
unalias n
```

Comando Alias de forma permanente

Para crear un alias permanente entre sesiones, debemos agregar los alias en el archivo de perfil de la configuración. Si usas bash, puedes acceder con el siguiente comando.

```
nano ~/.bashrc
```

En el caso que montes ZSH...

```
nano ~/.zshrc
```

Se abre el archivo de configuración, en el cual podrás identificar los alias predeterminados del sistema de forma rápida. Observa la imagen de ejemplo.

Solo necesitas insertar tu alias, en el bloque de alias o en uno nuevo. Por ejemplo...

```
alias n="nano"
```

Guardas el archivo, cierras el editor y la shell. Cuando inicies sesión de nuevo, el alias estará habilitado.

Ubuntu 20.04.2 lanzado con el kernel 5.8

Lanzada su primera versión el 23 de abril de 2020, **Ubuntu 20.04 LTS (Focal Fossa)** recibió ayer 5 de febrero de 2021 su segundo lanzamiento. Esta nueva versión escalable ofrece a los usuarios, un nuevo medio de instalación actualizado incorporando todas novedades y correcciones a las que nos tiene habituados **Canonical**.

El nuevo **Ubuntu 20.04.2 LTS**, como punto principal nos ofrece un nuevo kernel y la actualización de los controladores gráficos. La versión **20.04.2 LTS** se envía con el **kernel HWE** (Hardware Enablement) Linux 5.8 de forma predeterminada, así como el nuevo paquete de gráficos opensource, **Mesa 20.04.2**.



Ubuntu 20.04.2 LTS

Ubuntu 20.04.2 lanzado con el kernel 5.8

Si tenemos en cuenta que el Kernel 5.4, es un kernel LTS (Long Term Support), contando además con los beneficios de su soporte **upstream** a largo plazo (cuatro años), es posible que tengas la tentación de mantener tu Ubuntu 20.04.

Asegúrate bien, recuerda que la nueva versión viene con el **kernel de Linux 5.8 HWE**, que para más inri está diseñado para máquinas modernas, donde el hardware no es cien por cien compatible con el **kernel 5.4 LTS**.

Ubuntu Desktop 20.04.2 LTS, viene con el entorno de escritorio predeterminado **GNOME 3.36.8**. El resto de sabores oficiales, como Kubuntu, Lubuntu, Xubuntu, Ubuntu MATE, Ubuntu Budgie o Ubuntu Kylin, tambien se han actualizado.

Puedes descargar la nueva versión según tu escritorio preferido, desde los siguientes enlaces:

- [Descargar Ubuntu Desktop 20.04.2 LTS](#)
- [Descargar Ubuntu Server 20.04.2 LTS](#)
- [Descargar Kubuntu 20.04.2 LTS](#)
- [Descargar Xubuntu 20.04.2 LTS](#)
- [Descargar Lubuntu 20.04.2 LTS](#)
- [Descargar Ubuntu MATE 20.04.2 LTS](#)
- [Descargar Ubuntu Budgie 20.04.2 LTS](#)
- [Descargar Ubuntu Studio 20.04.2 LTS](#)
- [Descargar Ubuntu Kylin 20.04.2 LTS](#)



www.sololinux.es

8 formas de Crear archivos en linux



8 formas de crear archivos

Los **sistemas basados en Linux** y sus usuarios, somos conocidos por el uso intensivo que hacemos de la línea de comandos, para realizar cualquier tarea por muy compleja que llegue a ser.

Sin embargo, tenemos tantas **distribuciones Linux** diferentes (sobre todo de escritorio), que a veces nos pueden asaltar dudas sobre si los comandos son compatibles entre ellas. Para ello, los desarrolladores tienen la responsabilidad de hacer que todo sea 100% compatible y, que su entorno de escritorio sea lo más sencillo y fácil de usar para cualquier usuario.

En el artículo de hoy, veremos diferentes formas de **crear archivos en linux nuevos**. Tal vez no sepas, que una tarea tan sencilla y trivial, tiene múltiples formas de ejecutarla con un éxito asegurado.

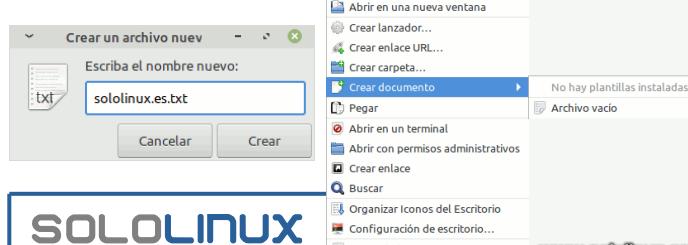


8 formas de crear archivos en linux

Tal vez alguna de las fórmulas propuestas en este artículo para crear un archivo, no sea muy práctica. Aun siendo así... en **sololinux.es** queremos aprender, por tanto debes conocer su existencia. Vemos las 8 formas de crear **archivos en linux** más comunes.

Desde la GUI

No cabe duda, que la forma más común en sistemas con entorno de escritorio instalado, es pulsando en el botón derecho del ratón y en crear nuevo archivo o documento. También desde tu administrador de archivos. Más fácil imposible.



Con el comando touch

El comando touch, es sin duda el más usado en estos menesteres. Es tan fácil como ejecutar el comando seguido del o los archivos a crear. Existen opciones que veremos en un próximo artículo.

```
touch sololinux.txt
# o
touch sololinux.txt sololinux1.txt sololinux2.txt sololinux3.txt
```

Con el comando cat

Ya comenté en un artículo anterior, que las posibilidades del comando cat son innumerables. Con el simple gesto de añadir un operador, podemos crear nuestro archivo y escribir en él, en tiempo real.

```
cat > sololinux.txt
```

El archivo se ha generado, puedes insertar los caracteres que quieras. Todo se guarda de forma automática.

```
sergio@sololinux:~$ cat > sololinux.es.txt
Ahora
puedes
insertar datos
```

Con el comando echo

El comando echo, también nos permite insertar un texto definido en un nuevo archivo. Usa la siguiente sintaxis y, no te olvides de usar comillas para definir el texto.

```
echo "caracteres"> archivo-nuevo
```

Por ejemplo...

```
echo "Me gusta sololinux"> sololinux.txt
```

Con heredoc

Heredoc no es un comando propiamente dicho, en realidad... es un tipo de redirección que permite pasar múltiples líneas de entrada a un comando. Nosotros nos ayudamos del comando cat (anteriormente descrito). Esta opción es muy interesante.

```
cat << EOF > archivo
```

Para guardar y salir de la edición debes ejecutar «EOF». Vemos un ejemplo...

```
cat << EOF > sololinux.txt
```

```
sergio@sololinux:~$ cat << EOF > sololinux.txt
> Hola
> soy
> SoloLinux
> EOF
sergio@sololinux:~$
```

www.sololinux.es

Con el comando printf

El comando printf, es un gran desconocido para los recién llegados a linux. Realmente es muy similar al comando echo. Tiene la particularidad de agregar formato, por ejemplo para agregar líneas debes usar el separador «\n».

Vemos unos ejemplos.

```
printf "Me gusta SoloLinux\n"> sololinux.txt
```

Ahora varias líneas.

```
printf "Me gusta soloLinux\n Yo sigo SoloLinux"> sololinux1.txt
```

Vemos una salida como...

```
sergio@sololinux:~$ printf "Me gusta soloLinux\n Yo sigo SoloLinux"> sololinux1.txt
sergio@sololinux:~$ cat sololinux1.txt
Me gusta soloLinux
Yo sigo SoloLinuxsergio@sololinux:~$
```

Crear archivos con un operador

Un operador de redirección, cambia el destino donde se mostraran los resultados, para ello usamos el carácter «>». Este símbolo le indica al sistema, que envíe los resultados al archivo especificado (si no existe lo genera). Realmente es una de las opciones más rápidas y eficaces, para lograr el objetivo de nuestro artículo de hoy.

```
> archivo
```

Un ejemplo simple.

```
> sololinux.txt
```

Con Nano o Vim

Como es evidente, también puedes crear archivos con cualquier editor. Los más utilizados son nano y Vim. Su uso es sencillo.

```
nano sololinux.txt
# o
vim sololinux.txt
```

```
:::
iLE88Dj. :jD88888Dj:
.LGitE888D.f8GjjjL8888E;
iE :8888Et. .G8888.
;i E888, ,8888,
D888, :8888:
D888, :8888:
D888, :8888:
D888, :8888:
888W, :8888:
W88W, :8888:
W88W: :8888:
DGGD: :8888:
:8888:
:W888:
:8888:
E888i
tW88D
```

www.sololinux.es



Uso del Comando printf en linux



El **comando printf** en Linux, se usa para mostrar cadenas formateadas, ya sea por número o por cualquier otro especificador de formato en la ventana de nuestra terminal. Funciona de la misma manera, que **printf** en el lenguaje de programación **C**, que es su base.

Las salidas impresas del comando, incluso su forma de operar nos recuerda al **comando echo**. Estamos ante una práctica herramienta que tiene la capacidad, de reproducir resultados con formato preciso a partir de argumentos numéricos o textuales.

Uso del comando printf en linux

La sintaxis de printf no es difícil, tan solo debes acostumbrarte a ella.

```
printf [-v var] formato [argumento]
```

Existen varios especificadores de formato, pero sin duda alguna los más utilizados son:

- **%s** – Especificador de cadena para la salida.
- **%b** – Nos permite interpretar secuencias de escape con un argumento.
- **%d** – Permite mostrar valores integrales.
- **%x** – Imprime valores hexadecimales en minúsculas con relleno de salida.
- **%f** – Permite mostrar valores con coma flotante.

También se permiten secuencias de escape e incluso, caracteres ordinarios. Una de las más conocidas y casi imprescindible es «**\n**», que indica salto de línea.

```
[~]$ printf
```

www.sololinux.es

Especificador %s

Cuando agregamos **%s**, indicamos una salida simple. Vemos un ejemplo.

```
printf "%s\n" "Hola, SoloLinux!"
```

El resultado.

```
sergio@sololinux:~$ printf "%s\n" "Hola, SoloLinux!"  
Hola, SoloLinux!  
sergio@sololinux:~$
```

Si no usáramos la secuencia «**\n**», el resultado sería visualmente horroroso.

```
printf "%s" "Hola, SoloLinux!"
```

Ejemplo de salida sin escape, por tanto sin salto de línea.

```
sergio@sololinux:~$ printf "%s" "Hola, SoloLinux!"  
Hola, SoloLinux!sergio@sololinux:~$
```

Especificador %b

Similar a «**%s**», pero con la diferencia que soporta varias secuencias de escape, de forma que nos permite imprimir varias líneas bien formateadas. Vemos un ejemplo.

```
printf "%b\n" "Hola, SoloLinux! \n" "Gracias por vuestra labor\n"
```

Vemos una salida como esta.

```
sergio@sololinux:~$ printf "%b\n" "Hola, SoloLinux! \n" "Gracias por  
vuestra labor\n"  
Hola, SoloLinux!  
Gracias por vuestra labor  
sergio@sololinux:~$
```

A modo de aprendizaje ejecutamos el mismo comando, pero sustituyendo **%b** por **%s**. Así podemos verificar que **%s**, no soporta varias secuencias de escape. Revisa la siguiente imagen, las secuencias se imprimen como si fueran texto.

```
sergio@sololinux:~$ printf "%s\n" "Hola, SoloLinux! \n" "Gracias por vuestra labor\n"  
Hola, SoloLinux! \n  
Gracias por vuestra labor\n  
sergio@sololinux:~$
```

www.sololinux.es

Especificador %d

Este especificador muestra los valores integrales de números enteros.

```
printf "%d\n" "2020" "2021"
```

```
sergio@sololinux:~$ printf "%d\n" "2020" "2021"  
2020  
2021  
sergio@sololinux:~$
```

Input: printf("Color %s, Number %d, Float %4.2f", "red", 123456, 3.14);

Output: Color red, Number 123456, Float 3.14

www.sololinux.es

Especificador %x

Imprime el hexadecimal de un valor con relleno de salida. Vemos un ejemplo.

```
printf "%08x\n" "2021"
```

Ejemplo de impresión en pantalla.

```
sergio@sololinux:~$ printf "%08x\n" "2021"
0000007e5
sergio@sololinux:~$
```

Especificador %f

Imprime valores decimales con coma flotante. También rellena de ceros el valor decimal.

```
printf "%f\n" "1,82" "2,16"
```

Observa la salida...

```
sergio@sololinux:~$ printf "%f\n" "1,82" "2,16"
1,820000
2,160000
sergio@sololinux:~$
```

Debes tener presente, que los valores decimales deben estar separados por una coma. Si usas como separador un punto, recibirás un error. Vemos un ejemplo del error.

```
printf "%f\n" "1.82" "2.16"
```

Imagen del error.

```
sergio@sololinux:~$ printf "%f\n" "1.82" "2.16"
bash: printf: 1.82: número no válido
0,000000
bash: printf: 2.16: número no válido
0,000000
sergio@sololinux:~$
```

Manual de printf

Puedes ver el manual integrado de la herramienta, con el siguiente comando.

```
man printf
```

```
Archivo Editar Ver Buscar Terminal Ayuda
PRINTF(1) User Commands PRINTF(1)

NAME
printf - format and print data

SYNOPSIS
printf FORMAT [ARGUMENT]...
printf OPTION

DESCRIPTION
Print ARGUMENT(s) according to FORMAT, or execute according to OPTION:
--help display this help and exit
--version output version information and exit
FORMAT controls the output as in C printf. Interpreted sequences are:
\"
double quote
\\
backslash
\a
alert (BEL)
\b
backspace
\c
produce no further output
\e
escape
\f
form feed
\n
new line
\r
carriage return
\t
horizontal tab
\v
vertical tab
\NNN byte with octal value NNN (1 to 3 digits)
\xHH byte with hexadecimal value HH (1 to 2 digits)
\uHHHH Unicode (ISO/IEC 10646) character with hex value HHHH (4 digits)
\UHHHHHHHH Unicode character with hex value HHHHHHHH (8 digits)
% a single %
\b ARGUMENT as a string with '\' escapes interpreted, except that
octal escapes are of the form \0 or \0NNN
```

Listar servicios con systemctl en linux

Un **servicio en Linux**, es una herramienta o aplicación que se ejecuta en segundo plano. Los servicios se pueden iniciar bajo demanda o en el momento de arrancar el sistema, todo depende de tus necesidades. Servicios comunes, pueden ser: las herramientas de un **servidor web**, **tareas cron** o el mismo **ssh**, entre otros muchos que se ejecutan en linux.

Aprender a **listar servicios en linux**, es una tarea importante; es la forma más rápida de identificar un problema, para posteriormente depurarlo. Le pese a quien le pese, las **distribuciones linux** más conocidas usan **systemd** como sistema de inicio y administrador de servicios, por tanto, en este artículo veremos como **listar servicios** con la **herramienta systemctl**.

Listar servicios con systemctl en linux

Systemd utiliza un concepto de módulos, que los podemos diferenciar por servicios, sockets, puntos de montaje, dispositivos, etc. Los módulos están definidos por archivos de texto con la extensión ini. Estos archivos contienen información y detalles, además de su configuración y los comandos que se deben ejecutar.

Las extensiones de archivo determinan su tipo. Por ejemplo, los archivos de los módulos de servicio del sistema tienen la extensión **.service**, y estos son los que nos interesan.

Podemos obtener un listado de todas las unidades de servicio cargadas y activas, con el siguiente comando.

```
sudo systemctl list-units --type service
```

El resultado está dividido en cinco columnas, además tiene un formato totalmente legible para humanos. Las columnas se identifican como...

- **UNIT** – Nombre de la unidad de servicio.
- **LOAD** – Nos indica si está cargado en la memoria.
- **ACTIVE** – Estado de activación del archivo del módulo de alto nivel.
- **SUB** – Estado de activación del archivo de unidad de bajo nivel.
- **DESCRIPTION** – Ofrece una breve descripción.

```
root@sololinux:~# sudo systemctl list-units --type service
UNIT                  LOAD   ACTIVE   SUB   DESCRIPTION
accounts-daemon.service    loaded active running Accounts Service
acpid.service           loaded active running ACPI event daemon
alsa-restore.service     loaded active exited  Save/Restore Sound Card St
apparmor.service         loaded active exited  AppArmor initialization
avahi-daemon.service     loaded active running Avahi mDNS/DNS-SD Stack
binfmt-support.service   loaded active exited  Enable support for addition
blk-availability.service loaded active exited  Availability of block devi
clean-mount-point@media-sergio-MEMORIA\x20USB.service loaded active running Clea
console-setup.service    loaded active exited  Set console font and keyma
cron.service            loaded active running Regular background program
cups-browsed.service     loaded active running Make remote CUPS printers
cups.service             loaded active running CUPS Scheduler
dbus.service             loaded active running D-Bus System Message Bus
exim4.service            loaded active running LSB: exim Mail Transport A
finalrd.service          loaded active exited  Create final runtime dir f
getty@tty1.service       loaded active running Getty on tty1
grub-common.service      loaded active exited  LSB: Record successful boo
hdtemp.service           loaded active exited  LSB: disk temperature moni
irqbalance.service       loaded active running irqbalance daemon
```

El comando anterior, solo lista los servicios cargados y activos. Si quieres verlos todos, ejecuta el comando siguiente.

```
sudo systemctl list-units --type service --all
```

UNIT	LOAD	ACTIVE	SUB	DESCRIPTION	www.sololinux.es
accounts-daemon.service	loaded	active	running	Accounts Service	
acpid.service	loaded	active	running	ACPI event daemon	
alsa-restore.service	loaded	active	exited	Save/Restore Sound Card State	
alsa-state.service	loaded	inactive	dead	Manage Sound Card State (restore and store)	
anacron.service	loaded	inactive	dead	Run anacron jobs	
apparmor.service	loaded	active	exited	AppArmor initialization	
apt-daily-upgrade.service	loaded	inactive	dead	Daily apt upgrade and clean activities	
apt-daily.service	loaded	inactive	dead	Daily apt download activities	
audio-restore.service	not-found	inactive	dead	audio-restore.service	
avahi-daemon.service	loaded	active	running	Avahi mDNS/DNS-SD Stack	
binfmt-support.service	loaded	active	exited	Enable support for additional executable binary form	
blk-availability.service	loaded	active	exited	Availability of block devices	
clamav-daemon.service	not-found	inactive	dead	clamav-daemon.service	
clean-mount-point@media-sergio-MEMORIA\x20USB.service	loaded	active	running	Clean up media/sergio/MEMORIA USB mount point	
command.service	not-found	inactive	dead	command.service	
console-screen.service	not-found	inactive	dead	console-screen.service	
console-setup.service	loaded	active	exited	Set console font and keymap	
cron.service	loaded	active	running	Regular background program processing daemon	
cups-blackservice	loaded	active	running	Make remote CUPS printers available locally	
cups-service	loaded	active	running	CUPS Scheduler	
dbus.service	loaded	active	running	D-Bus System Message Bus	
dm-event.service	loaded	inactive	dead	Device-mapper event daemon	
dns-clean.service	loaded	inactive	dead	Clean up any mess left by 0dns-up	
emergency.service	loaded	inactive	dead	Emergency Shell	
exim4.service	loaded	active	running	LSB: exim Mail Transport Agent	
fcof.service	not-found	inactive	dead	fcof.service	
festival.service	not-found	inactive	dead	festival.service	
finalrd.service	loaded	active	exited	Create final runtime dir for shutdown pivot root	
fstrim.service	loaded	inactive	dead	Discard unused blocks	
getty-static.service	loaded	active	running	getty on tty2-tty6 if dbus and logind are not available	
getty@tty1.service	loaded	active	running	Getty on tty1	

También podemos ver todos los archivos de módulo instalados.

```
sudo systemctl list-unit-files
```

En este caso también usamos una imagen como ejemplo. Es para que veas como se diferencian las salidas, mediante colores.

```
root@sololinux:~# sudo systemctl list-unit-files
UNIT           FILE                                     STATE
proc-sys-fs-binfmt_misc.automount          static
-.mount                                         generated
boot-efi.mount                                generated
dev-hugepages.mount                           static
dev-mqueue.mount                               static
proc-sys-fs-binfmt_misc.mount                 static
sys-fs-fuse-connections.mount                static
sys-kernel-config.mount                      static
sys-kernel-debug.mount                       static
acpid.path                                     enabled
cups.path                                       enabled
ntp-systemd-netif.path                        enabled
systemd-ask-password-console.path             static
systemd-ask-password-plymouth.path            static
systemd-ask-password-wall.path                static
session-c2.scope                             transient
accounts-daemon.service                      enabled
acpid.service                                 disabled
alsa-restore.service                         static
alsa-state.service                           static
alsa-utils.service                          masked
anacron.service                            enabled
apparmor.service                           enabled
apt-daily-upgrade.service                  static
apt-daily.service                           static
autovt@.service                           enabled
avahi-daemon.service                      enabled
binfmt-support.service                     enabled
blk-availability.service                  enabled
bluetooth.service                          disabled
bolt.service                                static
www.sololinux.es
```

Si lo que realmente te interesa, es conocer el estado de un servicio en particular. La sintaxis es sencilla.

```
sudo systemctl status [servicio].service
```

En nuestro ejemplo vemos el estado de «**cups**».

```
sudo systemctl status cups.service
```

Como puedes ver, el **servicio cups** está habilitado y corriendo.

```
root@sololinux:~# sudo systemctl status cups.service
● cups.service - CUPS Scheduler
   Loaded: loaded (/lib/systemd/system/cups.service; enabled; vendor
preset: enabled)
   Active: active (running) since Tue 2021-02-09 12:31:46 EET; 3h 9min
ago
     Docs: man:cupsd(8)
    Main PID: 641 (cupsd)
      Tasks: 1 (limit: 4521)
     CGroup: /system.slice/cups.service
             └─641 /usr/sbin/cupsd -l
feb 09 12:31:46 sololinux systemd[1]: Started CUPS Scheduler.
root@sololinux:~#
```

Explicamos los indicadores del ejemplo anterior:

- **Loaded** – Nos indica si está cargado y, la ruta del servicio.
- **Active** – Identifica si está activo y ejecutándose.
- **Docs** – Documentación.
- **Process** – Información sobre los procesos del servicio. En nuestro ejemplo no se utiliza, por eso no existe ningún proceso.

- **Main PID** – PID del servicio (identificador).
- **Tasks** – Número de tareas por unidad y su límite máximo.
- **CGroup** – Información relacionada sobre los grupos de control del servicio.

Como último aporte del artículo «Listar servicios con systemctl en linux», vemos una opción que tan solo nos indica si el servicio está activo.

```
systemctl is-active [servicio].service
```

Ejemplo...

```
systemctl is-active cups.service
```

```
root@sololinux:~# systemctl is-active cups.service
active
root@sololinux:~#
```

www.sololinux.es



Canales de Telegram:

[Canal SoloLinux](#) – [Canal SoloWordpress](#)

Espero que este artículo te sea de utilidad, puedes ayudarnos a mantener el servidor con una donación ([paypal](#)), o también Colaborar con el simple gesto de compartir nuestros artículos en tu sitio web, blog, foro o redes sociales

[Chat de SoloLinux en Telegram](#)



www.sololinux.es

Uso de AppImage en Ubuntu y otros linux

AppImage es una fabulosa herramienta, que opera como un paquete versátil de software en la mayoría de sistemas Linux. Es importante saber, que con **AppImage** el usuario actual no necesita permisos de superusuario para instalar aplicaciones.

AppImage en Ubuntu y otros linux, permite a los desarrolladores de aplicaciones implementar su software binario (independientemente de la distribución Linux utilizada); Esto se conoce como empaquetado ascendente.

Como indica su propio nombre, **AppImage** es algo similar a una imagen comprimida en la cual, se incluyen todas las bibliotecas y dependencias necesarias para ejecutar una aplicación (software), en Linux. Antes de aprender **cómo usar AppImage en Ubuntu y otros Linux**, conocemos sus principales características (que no son pocas):

- Es compatible con las principales distribuciones Linux.
- No se requiere de derechos root.
- Es portátil y tiene la capacidad de ejecutarse en una live.
- Permite desinstalar las aplicaciones borrando el archivo AppImage.



Uso de AppImage en Ubuntu y otros linux

AppImage es de amplia difusión, su listado de aplicaciones es tremendo. Además su crecimiento aumentó considerablemente gracias a la fastuosa **decisión tomada por Canonical con snap**. Como norma general, los desarrolladores incluyen en su zona de descargas las **aplicaciones AppImage**. Vemos un ejemplo.

Instalar AppImages

En nuestro artículo de ejemplo, vamos a instalar la **herramienta Krita** (por cierto, una excelente alternativa a **Photoshop**). Verás que simple es.

Es evidente que primero identificamos el AppImage de la herramienta seleccionada.

Descarga Krita 4.4.2
Disponible desde 19 enero, 2021 |

AppImage Ubuntu PPA Flatpak Gentoo

Δ Linux 64-bit Appimage (224MB)

Obtén la ultima versión independientemente de tu distribución. Descarga el controlador gmic-qt para que los filtros G'Mic funcionen.

Requisitos del sistema ▾ Todas las versiones disponibles ▾

Descargamos el AppImage.

```
wget https://download.kde.org/stable/krita/4.4.2/krita-4.4.2-x86_64.appimage
```

```
sergio@sololinux:~$ wget https://download.kde.org/stable/krita/4.4.2/krita-4.4.2-x86_64.appimage
--2021-02-10 17:10:35-- https://download.kde.org/stable/krita/4.4.2/krita-4.4.2-x86_64.appimage
Resolviendo download.kde.org (download.kde.org)... 168.119.32.158, 2a01:4f8:242:1ed5::3
Conectando con download.kde.org (download.kde.org) [168.119.32.158]:443... conectado.
Petición HTTP enviada, esperando respuesta... 302 Found
Ubicación: https://kde.mirrors.omnilance.com/stable/krita/4.4.2/krita-4.4.2-x86_64.appimage [siguiente]
--2021-02-10 17:10:35-- https://kde.mirrors.omnilance.com/stable/krita/4.4.2/krita-4.4.2-x86_64.appimage
Resolviendo kde.mirrors.omnilance.com (kde.mirrors.omnilance.com)... 185.126.112.9, 2001:67c:25dc:2::128:20
Conectando con kde.mirrors.omnilance.com (kde.mirrors.omnilance.com) [185.126.112.9]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 234688552 (224M) [application/octet-stream]
Guardando como: "krita-4.4.2-x86_64.appimage"
krita-4.4.2-x86_64.appimage 100%
[=====] 223,82M 4,89MB/s en 47s
2021-02-10 17:11:23 (4,72 MB/s) - "krita-4.4.2-x86_64.appimage" guardado [234688552/234688552]
```

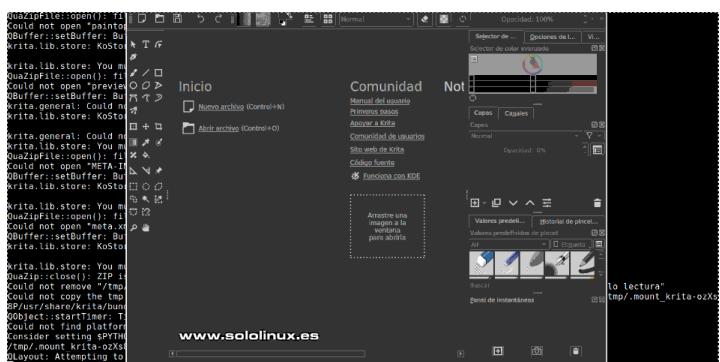
Como queremos instalar la herramienta sin permisos de root, concedemos permisos de usuario.

```
chmod u+x krita-4.4.2-x86_64.appimage
```

Ja. Bien. Ya lo tenemos.

Para ejecutar la herramienta tan solo debes hacer click en el AppImage o, lanzarla desde la consola con el siguiente comando.

```
./krita-4.4.2-x86_64.appimage
```



Para borrar completamente la aplicación, ejecutamos...

```
rm -f krita-4.4.2-x86_64.appimage
```



Web Scraping Con Python

Cuando escuchamos hablar de **web scraping** (también conocido como **raspado web**), se refiere a la extracción de datos de un sitio web simulando ser un humano. La información se recopila y, posteriormente se exporta a un formato en el cual el usuario se sienta cómodo, por ejemplo una **hoja de cálculo** o una API.

Lo que todos conocemos como **web scraping** se puede hacer manualmente, pero por comodidad y ahorro de tiempo, normalmente se usan herramientas automatizadas para extraer datos web, ya que suelen ser menos costosas y operan a un ritmo mucho más rápido.

Web Scraping con Python

Una de las herramientas más utilizadas para tal efecto, es **Python**. Hoy, **Diego Caraballo** de «[Mi diario Python](#)», nos enseña un pequeño código que simula el acceso a la página de GitHub y, por cada trabajo encontrado (para la posición Python) imprimirá en consola los datos que busca.

Observa que sencillo nos lo presenta Diego.

```
# Fuente: www.pythondiario.com
import requests
from bs4 import BeautifulSoup
URL = "https://jobs.github.com/positions?description=Python"
r = requests.get(URL)
s = BeautifulSoup(r.text, "html.parser")
for i in s.find_all("tr", class_="job"):
    print("Posición: ", i.td.h4.a.text)
    print("Descripción Url: ", i.td.h4.a.attrs['href'])
    print("Nombre Compañía: ", i.find("a", class_="company").text)
    print("Url Compañía: ", i.find("a", class_="company").attrs['href'])
    print("Tipo de trabajo: ", i.find("strong").text)
    print("Ubicación: ", i.find("span", class_="location").text, "\n")
```

```
# Fuente: www.pythondiario.com
# www.sololinux.es

import requests
from bs4 import BeautifulSoup

URL = "https://jobs.github.com/positions?description=Python"

r = requests.get(URL)
s = BeautifulSoup(r.text, "html.parser")

for i in s.find_all("tr", class_="job"):
    print("Posición: ", i.td.h4.a.text)
    print("Descripción Url: ", i.td.h4.a.attrs['href'])
    print("Nombre Compañía: ", i.find("a", class_="company").text)
    print("Url Compañía: ", i.find("a", class_="company").attrs['href'])
    print("Tipo de trabajo: ", i.find("strong").text)
    print("Ubicación: ", i.find("span", class_="location").text, "\n")
```

Si no estás familiarizado con el mundo de la programación en Python, es posible que las líneas de código que se muestran en el ejemplo te resulten difíciles de comprender, pero con un poco de práctica... rápidamente te moverás con soltura en él.

Python es fácil de usar, potente, rápido y versátil, lo que lo convierte en una excelente opción tanto para principiantes como para expertos. La legibilidad de Python lo convierte en un excelente primer lenguaje de programación, dado que te permite pensar como un programador y, no perder el tiempo con otras sintaxis confusas.

No cabe ninguna duda, que la mejor forma de aprender algo es mediante la práctica y las preguntas de ejercicio. Puedes revisar cientos de ejercicios en el siguiente link: <https://pythondiario.com/ejercicios-de-programacion-python>, como también los muchos tutoriales para principiantes que allí se brindan.

Autor: Diego Caraballo en colaboración con Sergio G.B.

Instalar LeoCAD en linux

LeoCAD es una herramienta de **diseño CAD**, gratuito y de código abierto para Linux y otras plataformas. Con este software CAD, puedes crear modelos de LEGO virtuales realmente impresionantes.

Se publica bajo **licencia pública GNU de código abierto** y, es totalmente compatible con el **LDraw Standard** y sus herramientas relacionadas. También admite la lectura y escritura de **archivos LDR y MPD**, además de permitirnos compartir y descargar modelos de Internet ya creados. La biblioteca de piezas **LDraw** actual de **LeoCAD**, cuenta con más de 10,000 piezas diferentes.

Novedades de la ultima versión:

- Ahora el tema predeterminado es de color oscuro.
- Se agrega una nueva ventana de vista previa 3D interactiva.
- Esta versión incluye una opción para atenuar los pasos anteriores y, admitir un nuevo modo de atenuación translúcido.
- Se añaden paletas para partes definidas.
- Permite personalizar los colores de la interfaz según tus necesidades.
- Guarda archivos con encabezados de estilo OMR.
- Se reemplaza la lista de piezas en HTML por una imagen estilo BOM.
- Usa el color de la parte en vez del color del borde, en el modo de estructura alámbrica.
- Traducciones al español y checo.
- Opciones de importación y exportación de plantillas de minifig.
- Herramienta de selección de color.
- Soporte para más partes del sintetizador.
- Nuevas fórmulas para cambiar el color de las piezas.
- Ahora se permite modificar el color de las líneas origen de la cuadrícula.
- Más atajos de acción.
- Deshabilitar la restauración de pestañas al cargar un modelo.
- Más opciones de línea de comandos para la cámara.
- Al asignar un método abreviado de teclado que ya está asignado, lanza un aviso.
- Mejorada la forma de ubicar piezas al arrastrarlas a un modelo.
- Zoom mejorado.
- Ahora no permite que los usuarios colapsen las listas de categorías o piezas.
- Puedes dividir la impresión de la consola en stdout y stderr al ejecutar en modo CLI.
- Renderizado con correcciones y mejoras.
- Se arreglan errores de la herramienta, al cargar algunos archivos MPD.
- Tamaño de vista fijo en pantallas retina.
- Vista fija rotando cuando se usa Look At.
- La ventana emergente se muestra en varios monitores si es necesario.
- Varias correcciones en el menú contextual.



Instalar LeoCAD en linux

Instalar LeoCad en linux es tarea sencilla. Comenzamos con la primera opción, que no podía ser otra que «flatpak». Tan solo debes ejecutar los siguientes comandos.

```
flatpak install flathub org.leocad.LeoCAD
```

```
sergio@sololinux:~$ flatpak install flathub org.leocad.LeoCAD
Required runtime for org.leocad.LeoCAD/x86_64/stable
(runtime/org.kde.Platform/x86_64/5.15) found in remote flathub
Do you want to install it? [y/n]: Do you want to install it? [y/n]: y
Installing in system:
org.kde.Platform/x86_64/5.15          flathub 35f0da737e7c
org.kde.Platform.Locale/x86_64/5.15   flathub bb6a61ee4707
org.leocad.LeoCAD/x86_64/stable       flathub 61d32820ab4
permissions: ipc, x11, dri
file access: home, xdg-config/kdeglobals:ro
dbus access: com.canonical.AppMenu.Registrar
Is this ok [y/n]:
```

Una vez concluya la instalación, la puedes lanzar directamente desde tu consola / terminal.

```
flatpak run org.leocad.LeoCAD
```

Si quieras estar seguro de tener la última versión, lo mejor es que te decantes por el paquete AppImage. En su Github oficial, puedes seleccionar la última versión y descargarla. Nosotros lo hacemos desde línea de comandos.

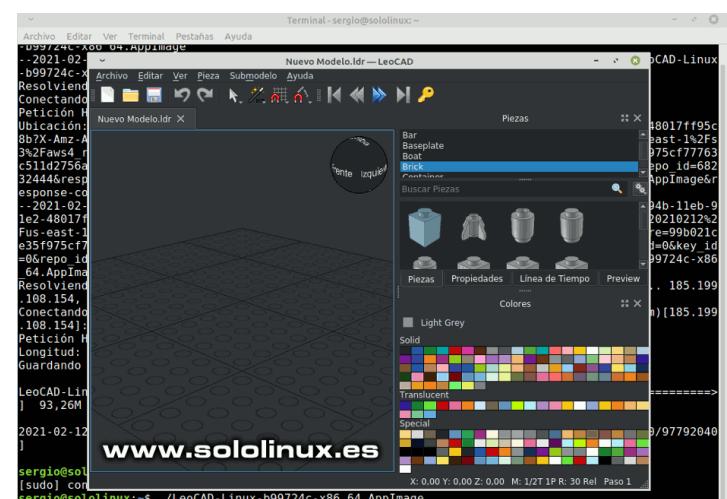
```
wget
https://github.com/leozide/leocad/releases/download/continuous/LeoCAD-Linux-b99724c-x86_64.AppImage
```

Le concedemos los permisos necesarios.

```
sudo chmod +x LeoCAD-Linux-b99724c-x86_64.AppImage
```

Ahora solo falta ejecutar la herramienta.

```
./LeoCAD-Linux-b99724c-x86_64.AppImage
```



Puedes borrar el AppImage de LeoCAD con:

```
rm LeoCAD-Linux-b99724c-x86_64.AppImage
```

Ocultar dispositivos montados en el dock de Ubuntu 20.04



De manera predeterminada, en **Ubuntu 20.04** y otras versiones anteriores y superiores se muestran los dispositivos montados en el dock, que suele aparecer a la izquierda del escritorio.

Particiones de disco montadas, **dispositivos USB externos** montados y otros volúmenes del sistema, aparecen en nuestro dock. Esta función no siempre es agradable, incluso a veces puede resultar molesta si tienes muchas aplicaciones abiertas en la barra lateral.

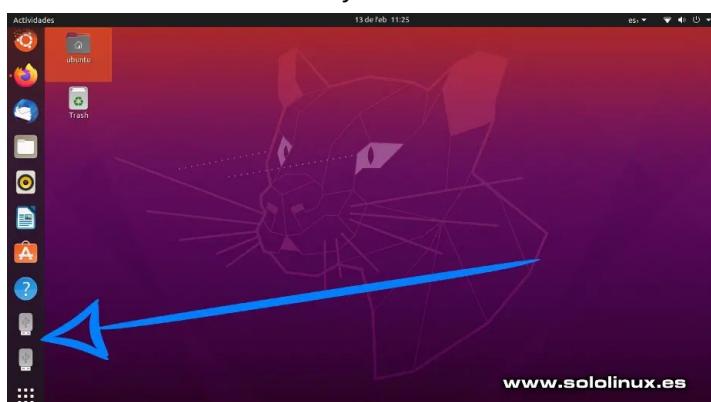
En este artículo veremos como ocultar dispositivos montados en el dock y, también los que aparecen en el **escritorio** de tu Ubuntu. Son diferentes comandos, así que puedes elegir entre uno, otro, los dos o ninguno.



Ocultar dispositivos montados en el dock de Ubuntu 20.04

Nunca me gusto este tipo de menús lanzadores rápidos, los considero más un estorbo que otra cosa. Además se van agregando aplicaciones abiertas y, no digo nada si tienes 4 o 6 dispositivos externos conectados y montados. Ya te pierdes!!!.

Lo que haremos con un simple comando, es ocultar cualquier dispositivo o partición montada que aparezca en el dock. Realmente es un alivio. Pero primero... vemos una imagen de ejemplo con dos **dispositivos USB** externos conectados y montados.



Para ocultar los dispositivos montados en el dock, ejecuta el siguiente comando.

```
gsettings set org.gnome.shell.extensions.dash-to-dock show-mounts false
```

Automáticamente los dispositivos desaparecieron del dock. Realmente siguen insertados y montados pero sin ocupar espacio en el menú de la izquierda. Observa la imagen de ejemplo...



Igual que puedes ocultar, también puedes recuperar los iconos. Es así de fácil.

```
gsettings set org.gnome.shell.extensions.dash-to-dock show-mounts true
```

De la misma forma que en el dock, dependiendo de tu configuración también es posible que aparezcan los iconos en el escritorio. No es tan molesto como con la opción anterior, pero en situaciones de escritorios con muchos iconos, también pueden resultar un problema.

Puedes ocultar dispositivos montados en el escritorio de Ubuntu 20.04 y, otras versiones con este comando.

```
gsettings set org.gnome.shell.extensions.desktop-icons show-mount false
```

Al igual que con el dock, puedes retornar el visionado de los **lanzadores (iconos)**, con...

```
gsettings set org.gnome.shell.extensions.desktop-icons show-mount true
```

Como puedes ver, ocultar dispositivos montados en tu distribución Ubuntu es tarea sencilla.



Ver listas IPTV online Con TV-Lite

TV-Lite es un reproductor de listas IPTV de código abierto, gratuito y con la capacidad de manejar **Sopcast** y **Acestream**, sobre linux y otras plataformas como Windows. En este artículo, veremos como instalar el reproductor en Ubuntu y sus derivados.

TV-Lite es el reemplazo natural del obsoleto **TV-Maxe**. Un tema importante que puede no agradar a muchos... usa **VLC** para la reproducción de medios y, necesita Acestream o Sopcast para que este software pueda manejar otros tipos de transmisión. Esto es importante, ya que es posible que resulten necesarias librerías específicas de otros escritorios diferentes al nuestro.



Ver listas IPTV online con TV-Lite

Para instalar TV-Lite, agregamos el repositorio válido para Ubuntu, Linux Mint y todos sus derivados.

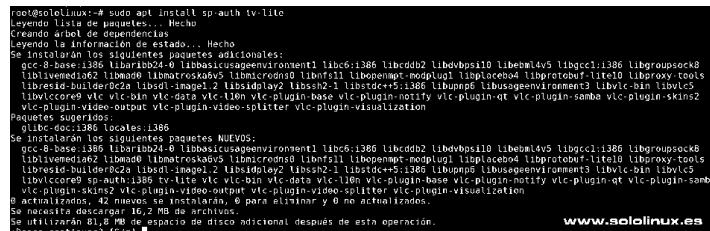
```
sudo add-apt-repository ppa:tv-lite/ppa-git
```

```
root@sololinux:/home/sergio# sudo add-apt-repository ppa:tv-lite/ppa-git
Está a punto de añadir el siguiente PPA:
PPA repository for TV-Lite.
WARNING: sp-auth can only be installed from terminal, because it's i386 architecture and installation on any other way gives dependencies issue error.
You can install it from terminal by typing command:
sudo apt install sp-auth
Más información: https://launchpad.net/~tv-lite/+archive/ubuntu/ppa-git
Pulse Intro para continuar o Ctrl+C para cancelar
Executing: /tmp/tmp-key-gpphone.5WljL45gGM/gpg.1.sh --keyserver
https://keyserver.ubuntu.com:443 --recv-keys
E37265209782236357AD4A9D4A5AAE3BE2B2601
gpg: clave 4A5AAE3BE2B2601: clave pública "Launchpad PPA for TV-Lite"
importada
gpg: Cantidad total procesada: 1
gpg: importadas: 1
```

Ahora actualizamos e instalamos la herramienta.

```
sudo apt update
sudo apt install sp-auth tv-lite
```

TV-Lite necesita de una serie de librerías y herramientas, que tal vez no tengas instaladas en tu sistema. Debes aceptar el aviso para que continúe la instalación. Imagen de ejemplo.



Puedes lanzar el reproductor IPTV, desde el menú de aplicaciones de tu escritorio por defecto. Si no tienes instalado **Acestream**, aparece el siguiente error.



Puedes instalar **Acestream** desde **snap** con el siguiente comando.

```
sudo snap install acestreamplayer
```

Si no lo tienes instalado...

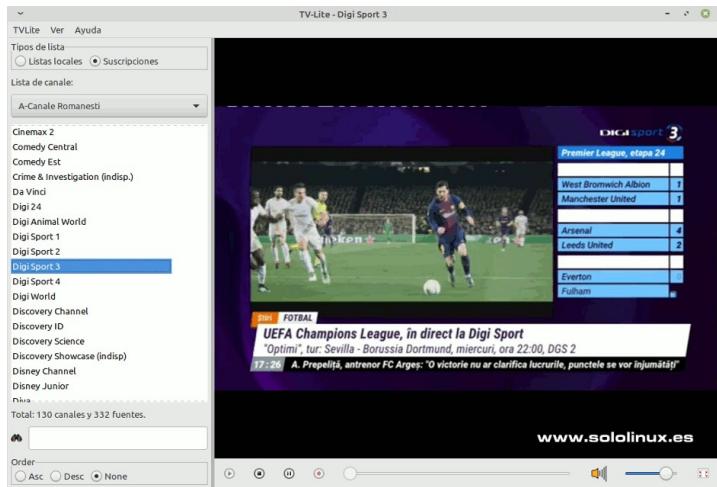
```
sudo apt install snapd
```

```
root@sololinux:~# sudo snap install acestreamplayer
2021-02-15T17:14:27+02:00 INFO Waiting for automatic snapd restart...
Se ha instalado acestreamplayer 3.1.49-snap3 por vasilisc
```

Reinicia el sistema.

```
sudo reboot
```

Ahora sí que puedes lanzar TV-Lite sin errores.



De forma predeterminada salen bastantes canales rumano, pero puedes agregar las listas que tú quieras. Recuerda que desde **sololinux.es** no fomentamos el pirateo (todo lo contrario). El uso malintencionado de este tipo de herramientas, puede estar penado por la ley de tu país con fuerte multas, incluso prisión. NO SEAS PIRATA.

En administrar suscripciones, puedes agregar la siguiente url: «<https://is.gd/freeiptvall>». Contiene muchos canales libres de diferentes países, siempre actualizado.

Para desinstalar el reproductor de listas IPTV, ejecuta estos comandos.

```
sudo apt remove sp-auth tv-lite
sudo add-apt-repository --remove ppa:tv-lite/ppa-git
```

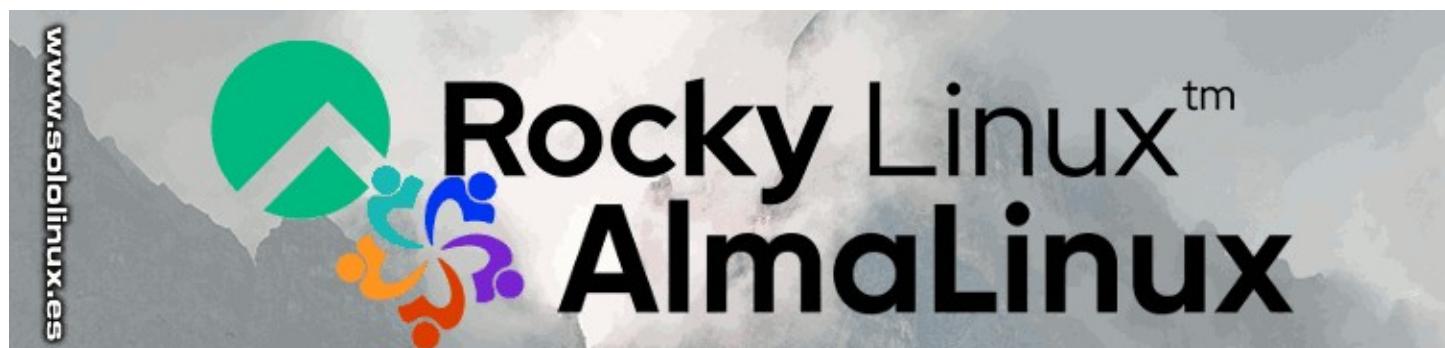
Instalar AlmaLinux Beta – El sucesor de CentOS 8

Desde que Red Hat, la empresa matriz de CentOS, anunciara que dejaba de apoyar a CentOS tal como lo conocíamos, para convertirse en una especie de **Fedora**. Muchas fueron las reacciones, casi ninguna buena.

Es evidente que **CentOS** dejará de ser una de las distribuciones de servidor referencia, para pasar a un segundo plano y convertirse en una distro de escritorio destinada a desaparecer con el tiempo, ya que no tiene potencial suficiente para competir con otras ya consolidadas. Buena jugada la de Red Hat, ya que acabar con su derivado gratuito es una buena estrategia comercial, si solo les preocupa el beneficio económico.

Si hablamos de **servidores web** y otras ramas, Debian y CentOS se llevan la palma, pero eso se acabó. Miles de administradores de sistemas, incluso las empresas de alojamiento pusieron el grito en el cielo, nadie quiere que su servidor estable se convierta en un banco de pruebas para Red Hat, nadie.

Por suerte, la comunidad se puso manos a la obra y muchas son las iniciativas, pero viables al 100% (por lo menos de momento), solo tenemos dos. Rocky Linux y AlmaLinux.



Gregory Kurtzer, el fundador del CentOS original junto con una comunidad en la que me incluyo, desarrolla **Rocky Linux** de la cual aún se trabaja en el código (no hay beta). Por otro lado tenemos a **CloudLinux**, que de un plumazo vio como su impresionante plataforma se venía abajo. Su reacción fue rápida y firme, para comenzar se invierten 1.000.000 de dólares, sus programadores y recursos necesarios, con el fin de crear un nuevo derivado de **Red Hat Enterprise** lo más urgente posible.

Ahora... este nuevo derivado de **RHEL** ya tiene nombre definitivo y versión beta lista para que la pruebes. **Instalar AlmaLinux Beta**, es una tarea sencilla y muy similar a CentOS. En este artículo vemos como hacerlo paso a paso con imágenes.

Instalar AlmaLinux Beta – El sucesor de CentOS 8

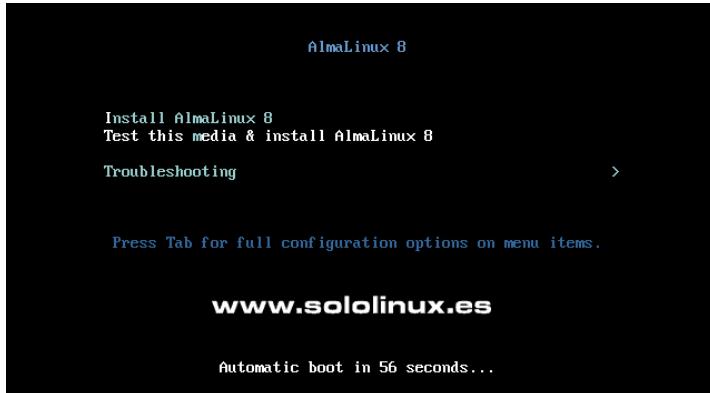
La beta de **AlmaLinux** viene con su propio **rastreador de errores**. Los desarrolladores de CloudLinux trabajan en estrecha colaboración con los tickets de error que tu mismo puedes reportar, de forma que entre todos consigamos una versión estable. Los usuarios registrados en [esta página](#), pueden presentar los informes de errores que serán revisados y priorizados por el equipo de ingeniería de CloudLinux.

Para instalar **AlmaLinux**, nos dirigimos a su zona de [descargas oficial](#) y, seleccionamos la que nos interese. En nuestro caso **AlmaLinux-8.3-beta-1-x86_64-dvd1.iso**.

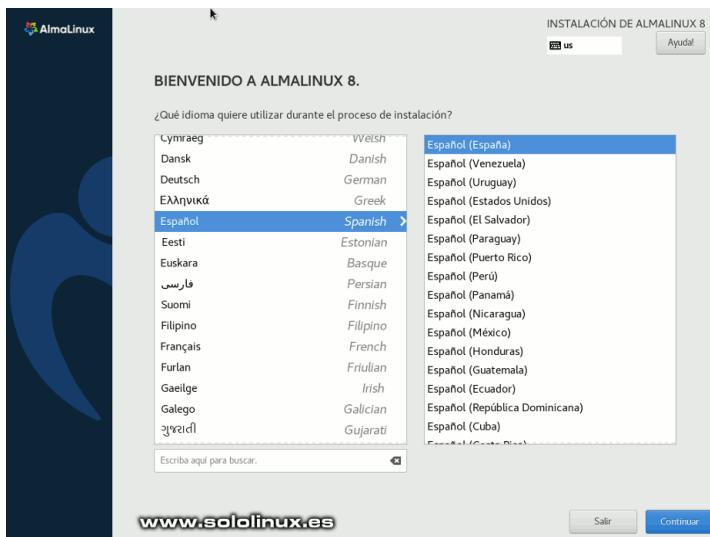
Index of /almalinux/8.3-beta/isos/x86_64/

		www.sololinux.es
.. /		
Almalinux-8.3-beta-1-x86_64-boot.iso	28-Jan-2021 13:45	680525824
Almalinux-8.3-beta-1-x86_64-boot.iso.manifest	28-Jan-2021 13:46	626
Almalinux-8.3-beta-1-x86_64-dvd1.iso	28-Jan-2021 13:47	8710520832
Almalinux-8.3-beta-1-x86_64-dvd1.iso.manifest	28-Jan-2021 13:47	470733
Almalinux-8.3-beta-1-x86_64-minimal.iso	28-Jan-2021 13:46	1892679680
Almalinux-8.3-beta-1-x86_64-minimal.iso.manifest	28-Jan-2021 13:46	102488
CHECKSUM	28-Jan-2021 20:49	1211

Bien, grabamos la iso en un dispositivo USB y comenzamos la instalación.



En la primera pantalla que nos aparece, marcamos nuestro lenguaje por defecto.

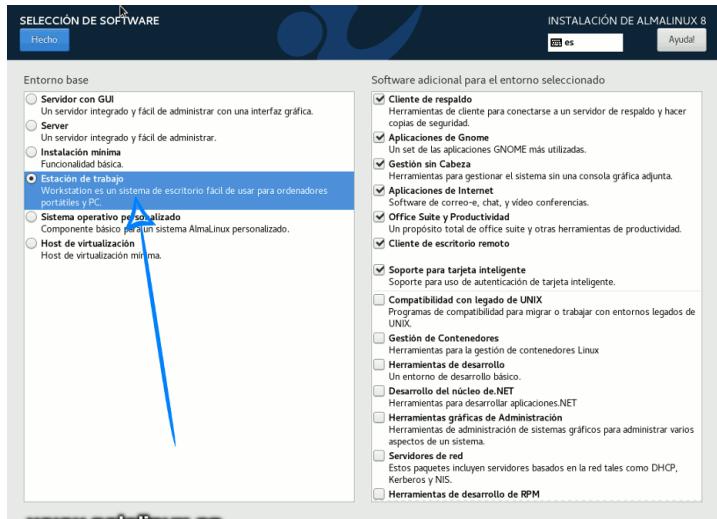


A continuación nos aparece la pantalla de ajustes. Debes prestar especial atención al destino de la instalación, a crear una **contraseña de root** (si no lo haces estará deshabilitado), a la selección del software, también debes habilitar la red (deshabilitada por defecto) y a crear un nuevo usuario. Observa que aparecen varios errores.

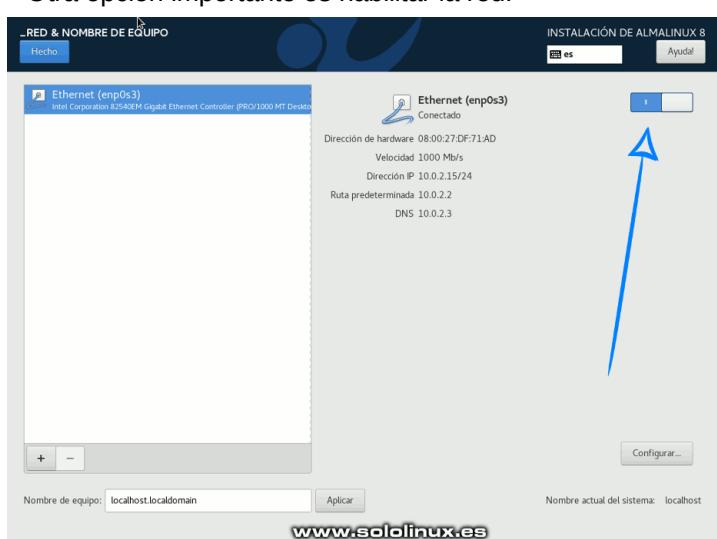


Ahora vemos un par de imágenes de ejemplo de configuración. La primera y... posiblemente decisiva para un correcto funcionamiento del sistema, es que elijas el fin por el que quieras **instalar AlmaLinux** en tu máquina. En nuestro caso seleccionamos **estación de trabajo**, pero se ofrecen seis opciones diferentes.

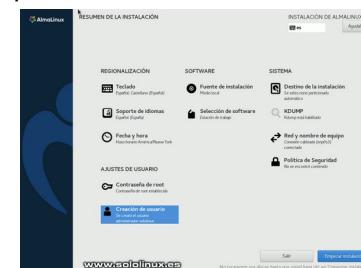
- **Servidor con GUI**
- **Server**
- **Instalación mínima**
- **Estación de trabajo**
- **Sistema personalizado**
- **Host de virtualización**



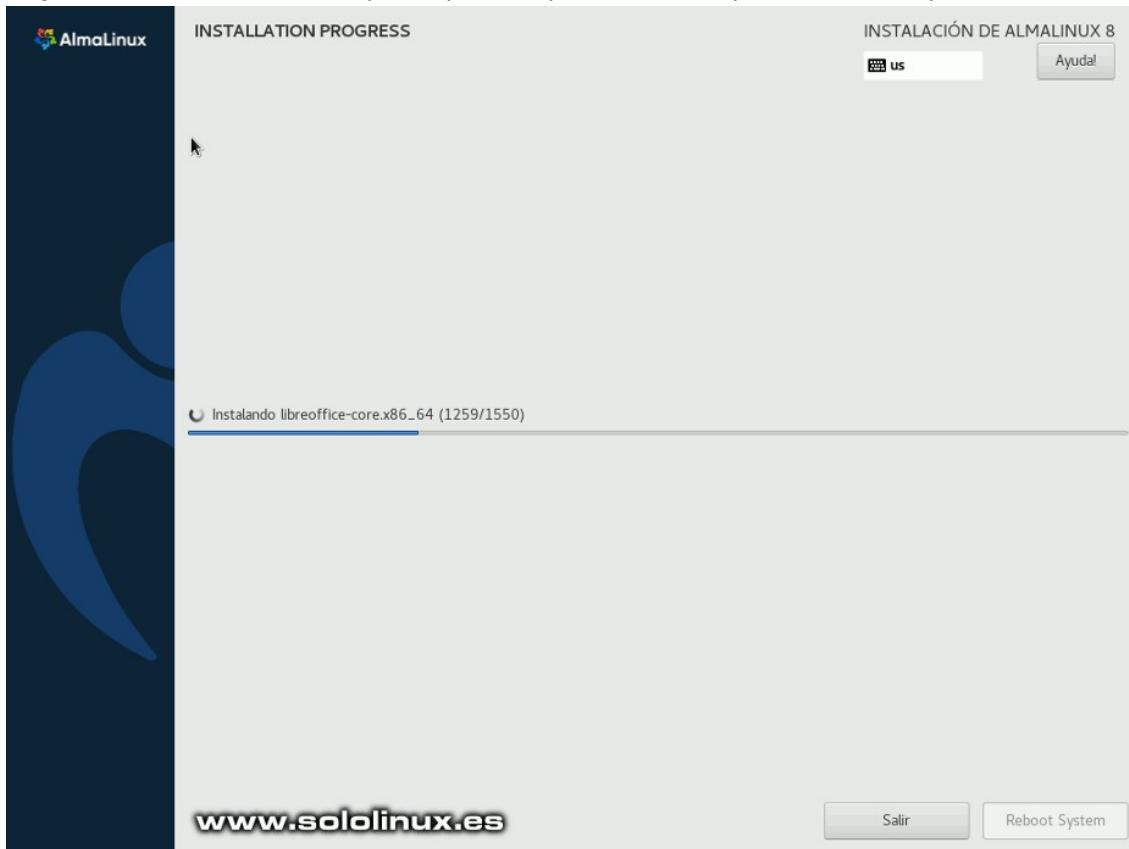
Otra opción importante es habilitar la red.



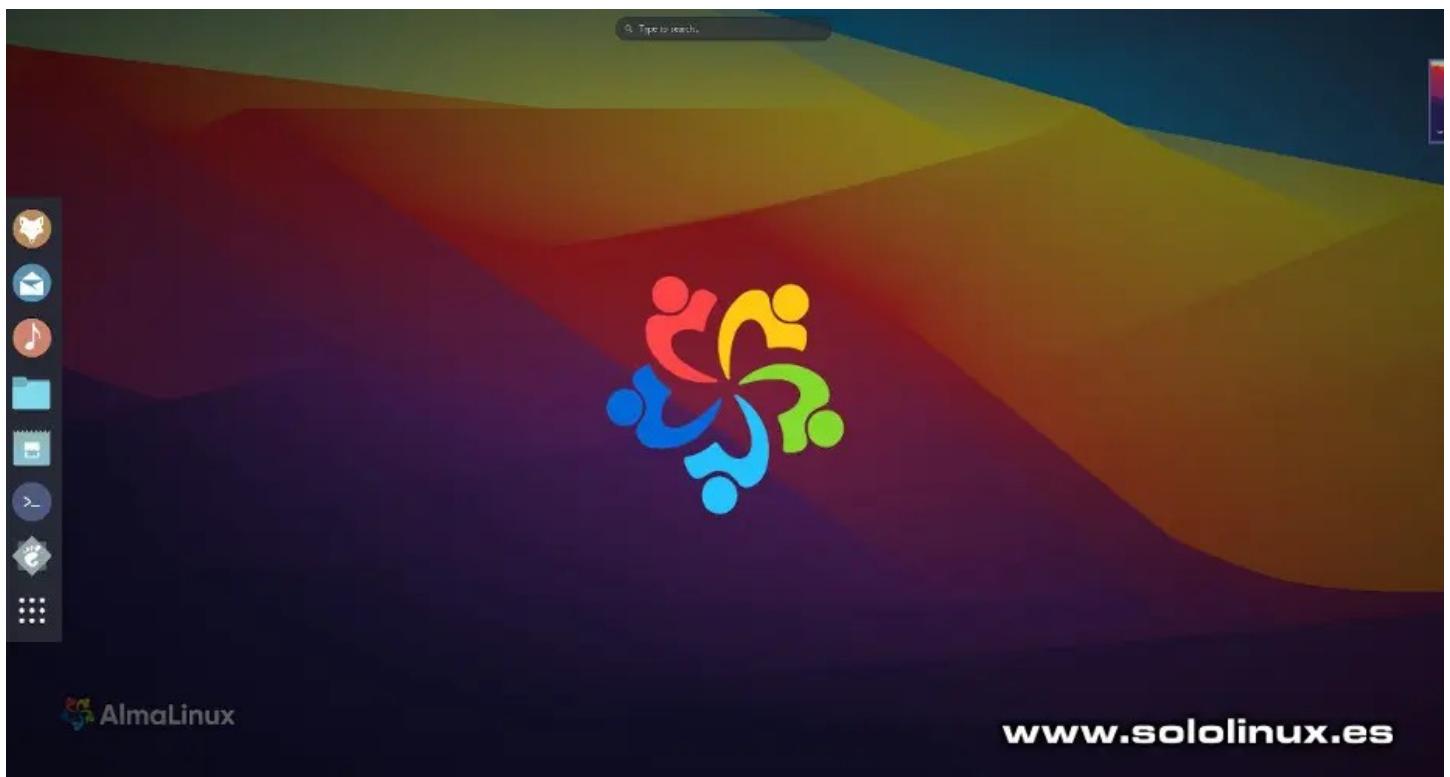
Una vez finalizada la configuración, ya no aparecen errores en la pantalla de instalación.



Pulsa en «Empezar instalación». El tiempo del proceso puede variar dependiendo de la potencia de la máquina.



Al concluir el proceso, pulsa en «Reboot System». Se reiniciará el sistema con un trabajado entorno de escritorio Gnome.



Nota final: Recuerda que AlmaLinux aún continua en proceso de desarrollo, esta versión no es apta para que la uses en producción.

Instalar Etcher en Ubuntu 20.04 y otros linux



Etcher es una excelente herramienta de escritura de imágenes en USB, de código abierto y multiplataforma. Disponible para sistemas operativos como Linux, Windows o Mac, destaca por su facilidad de uso y potencial, al estilo de otra potente herramienta conocida como **Ventoy**.

Su interfaz es igual independientemente del **sistema operativo** de origen, sencilla y muy fácil de usar, ya que con solo 3 pasos puedes pasar tu **archivo.iso** al dispositivo USB. Una vez lanzada la aplicación, debes elegir una imagen y la unidad de destino, una vez hecho esto, haces clic en el botón Flash y ya lo tienes. Así de fácil.

Existen opciones que pueden ralentizar el proceso de grabación, como por ejemplo la de validar la grabación, muy útil si quieres verificar la integridad y confirmar que la operación resultó un éxito. En este artículo, vemos cómo **instalar Etcher** en Ubuntu 20.04, Ubuntu 18.04, Linux Mint, Debian 10 y resto de derivados. También en Manjaro, Arch, CentOS, Fedora, Open Suse, etc.



Instalar Etcher en Ubuntu 20.04

Para instalar Etcher agregamos su repositorio estable oficial.

```
echo "deb https://deb.etcher.io stable etcher" | sudo tee /etc/apt/sources.list.d/balena-etcher.list
```

```
sololinux@sololinux:~$ sudo echo "deb https://deb.etcher.io stable etcher" | sudo tee /etc/apt/sources.list.d/balena-etcher.list
[sudo] contraseña para sololinux:
deb https://deb.etcher.io stable etcher
sololinux@sololinux:~$
```

Ubuntu deshabilita por defecto los repositorios sin firmar, por tanto agregamos la key necesaria.

```
sudo apt-key adv --keyserver https://keyserver.ubuntu.com:443 --recv-keys 379CE192D401AB61
```

Verás algo similar a la siguiente imagen.

```
sololinux@sololinux:~$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 379CE192D401AB61
Executing: /tmp/tmp.1MdnrhkN8dx/gpg1.sh --keyserver keyserver.ubuntu.com --recv-keys 379CE192D401AB61
gpg: clave 379CE192D401AB61: clave pública 'Bintray (by JFrog) <bintray@bintray.com>' importada
gpg: Cantidad total procesada: 1
gpg: importadas: 1
sololinux@sololinux:~$
```

www.sololinux.es

Bien, ahora ya puedes actualizar e instalar Etcher.

```
sudo apt update
sudo apt-get install balena-etcher-electron
```

Comienza la instalación.

```
sololinux@sololinux:~$ sudo apt install balena-etcher-electron
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  balena-etcher-electron
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 82,6 MB de archivos.
Se utilizarán 235 MB de espacio de disco adicional después de esta operación.
Des:1 https://dl.bintray.com/etcher/debian stable/etcher amd64 balena-etcher-electron amd64 1.5.100 [82,6 MB]
Descargados 82,6 MB en 21s (3.965 kB/s)
```

```
Seleccionando el paquete balena-etcher-electron previamente no seleccionado.
(Leyendo la base de datos ... 343479 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../balena-etcher-electron_1.5.100_amd64.deb ...
Desempaquetando balena-etcher-electron (1.5.100) ...
Configurando balena-etcher-electron (1.5.100) ...
Procesando disparadores para hicolor-icon-theme (0.17-2) ...
Procesando disparadores para mime-support (3.60ubuntu1) ...
Procesando disparadores para desktop-file-utils (0.23+linuxmint8) ...
Procesando disparadores para gnome-menus (3.13.3-11ubuntu1.1) ...
```

Una vez instalada la herramienta, la puedes lanzar desde el menú de aplicaciones de tu entorno de escritorio. En la imagen de ejemplo vemos lo fácil que es de utilizar.



Para desinstalar Etcher, ejecutamos...

```
sudo apt remove balena-etcher-electron
sudo rm /etc/apt/sources.list.d/balena-etcher.list
sudo apt update
```

Instalar Etcher en otros linux En CentOS, Fedora y derivados

```
sudo wget https://balena.io/etcher/static/etcher-rpm.repo -O /etc/yum.repos.d/etcher-rpm.repo
sudo yum install -y balena-etcher-electron
# o
sudo dnf install -y balena-etcher-electron
```

Si lo quieres desinstalar.

```
sudo yum remove -y balena-etcher-electron
sudo rm /etc/yum.repos.d/etcher-rpm.repo
sudo yum clean all
sudo yum makecache fast
# o
sudo dnf remove -y balena-etcher-electron
sudo rm /etc/yum.repos.d/etcher-rpm.repo
sudo dnf clean all
sudo dnf makecache
```

En OpenSuse y derivados

```
sudo zypper ar https://balena.io/etcher/static/etcher-rpm.repo
sudo zypper ref
sudo zypper in balena-etcher-electron
```

Lo borramos con...

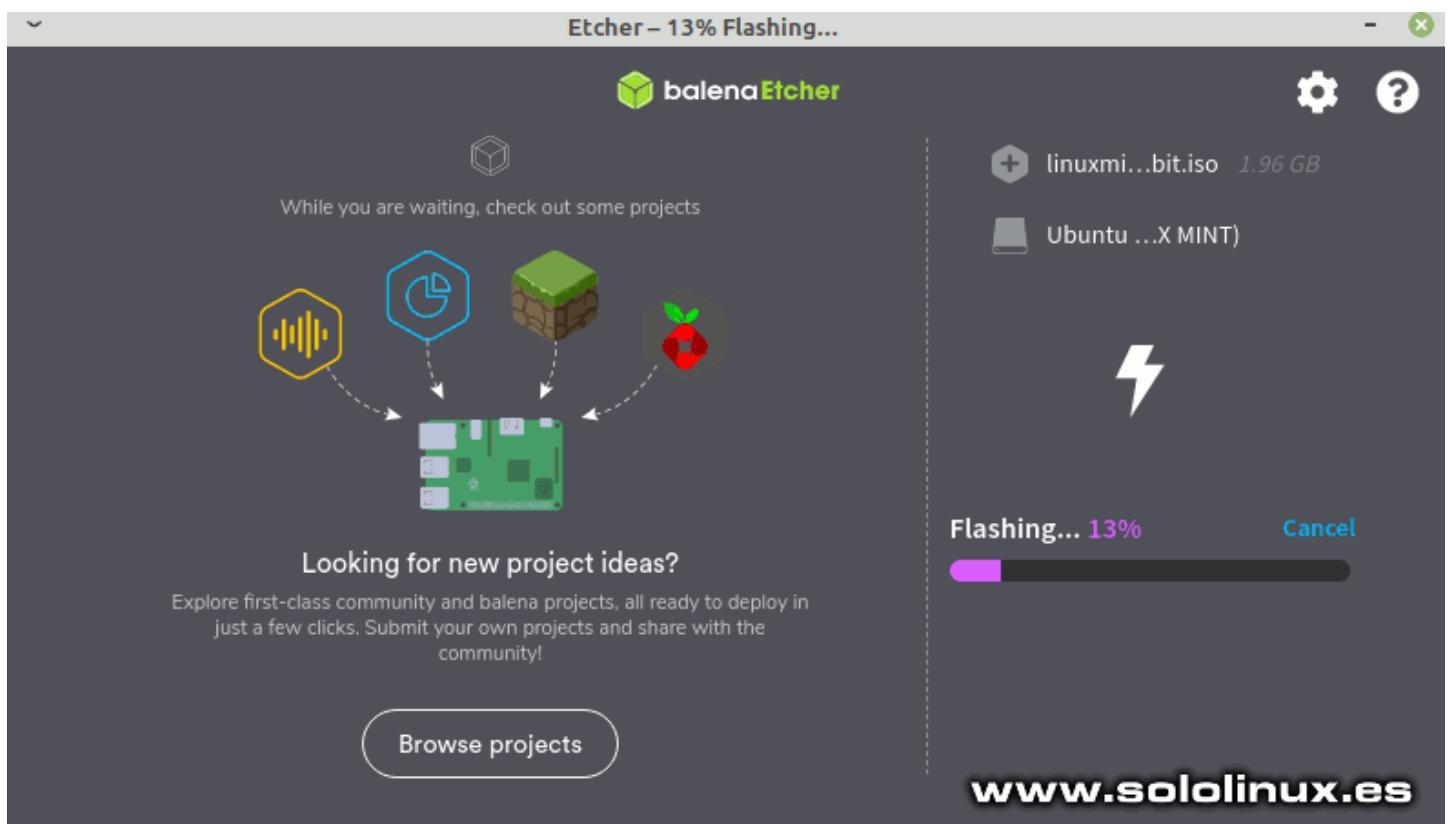
```
sudo zypper rm balena-etcher-electron
```

En Arch Linux, Manjaro y derivados

```
yay -S balena-etcher
```

En caso de que lo quieras desinstalar.

```
yay -R balena-etcher
```



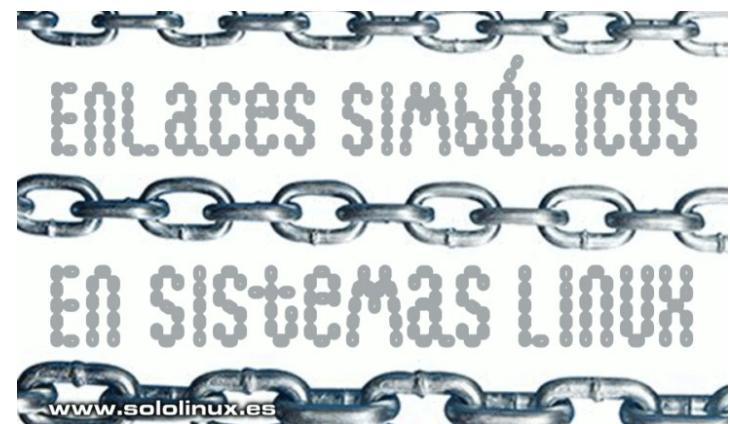
Como Crear enlaces simbólicos en linux

Un **enlace simbólico** también conocido como «**Symlink**», es un tipo de archivo especial de **Linux**, que se utiliza con el propósito de apuntar o enlazar a otro archivo. Antes de continuar con la explicación, debes tener presente que al crear enlaces simbólicos no se modifica el contenido del archivo original, su única misión es enlazar por tanto no tienen contenido.

Los **enlaces simbólicos** son particularmente útiles como archivos de acceso directo; enlace simbólico de una herramienta, acceso a una aplicación desde el **escritorio**, a la carpeta de inicio, archivos y bibliotecas de programas, etc, etc.

Otro uso importante de los **Symlinks** o enlaces simbólicos en **linux**, es ofrecer una mayor seguridad de los archivos ejecutables. Crear un enlace simbólico y permitir que los usuarios lo ejecuten, no les otorga a los usuarios ningún permiso sobre el archivo original. No existe ningún riesgo de que el original sea explotado, directa o indirectamente. Si accedes a tu **sistema**, observaras que está lleno de ellos.

Hoy, vemos cómo crear enlaces simbólicos de archivos en Linux, también como borrarlos si llegado el caso ya no nos interesan.



Como crear enlaces simbólicos en linux

En **linux** utilizamos dos tipos de enlaces, los simbólicos y los fijos, también conocidos como **enlaces duros**. La función básica de los dos es la misma, pero en realidad son muy diferentes en su forma de operar. Vemos las dos opciones.

Crear enlaces simbólicos

Un enlace simbólico es simplemente un puntero al nombre del archivo principal, por tanto si modificas el destino del principal, el **enlace simbólico** ya no es funcional. Aun así... resultan de tremenda utilidad para nuestro sistema **linux**. Vemos como crear un enlace simbólico, es algo muy simple, tan solo necesitamos el **comando ln** y su argumento «**-s**» o «**--symbolic**».

En nuestro caso de ejemplo tenemos el archivo **demo.txt**, cuyo contenido lo verificamos con el **comando cat**.

```
cat demo.txt
sergio@sololinux:~$ cat demo.txt
Hola
me gusta
SoloLinux
```

Ahora creamos un archivo simbólico, que apunte a **demo.txt**, lo llamaremos **demo_symbolic**.

```
ln -s demo.txt demo_symbolic
```

Habráis observado que primero insertamos el archivo original y después el enlace. Independientemente del tipo de enlace, esto siempre es así. Verificamos que tenemos nuestro **enlace simbólico** creado correctamente.

```
cat demo_symbolic
```

En la imagen inferior, vemos el resultado completo de la operación.

```
sergio@sololinux:~$ ln -s demo.txt demo_symbolic
sergio@sololinux:~$ cat demo_symbolic
Hola
me gusta
SoloLinux
```

www.sololinux.es

Crear enlaces duros o fijos

Un **enlace duro** también conocido como **enlace fijo**, apunta directamente al **inode** del archivo principal. Esto quiere decir que el enlace será válido, incluso si mueves el archivo principal. Para los enlaces duros o fijos, usamos el comando **ln** sin ningún tipo de argumento. Vemos un ejemplo.

```
ln demo.txt demo_duro
```

Verificamos el resultado.

```
sergio@sololinux:~$ ln demo.txt demo_duro
sergio@sololinux:~$ cat demo_duro
Hola
me gusta
SoloLinux
```

Ahora movemos el enlace duro a la carpeta demo y, comprobamos que efectivamente el enlace sigue siendo operativo, incluso al haberlo cambiado de sitio.

```
sergio@sololinux:~$ cd demo
sergio@sololinux:~/demo$ cat demo_duro
Hola
me gusta
SoloLinux
```

Borrar enlaces simbólicos o duros

Los enlaces simbólicos o duros, los puedes borrar como haces habitualmente con otros archivos, usando el comando **rm**. Vemos unos ejemplos.

```
rm demo_symbolic
```

Si tienes varios...

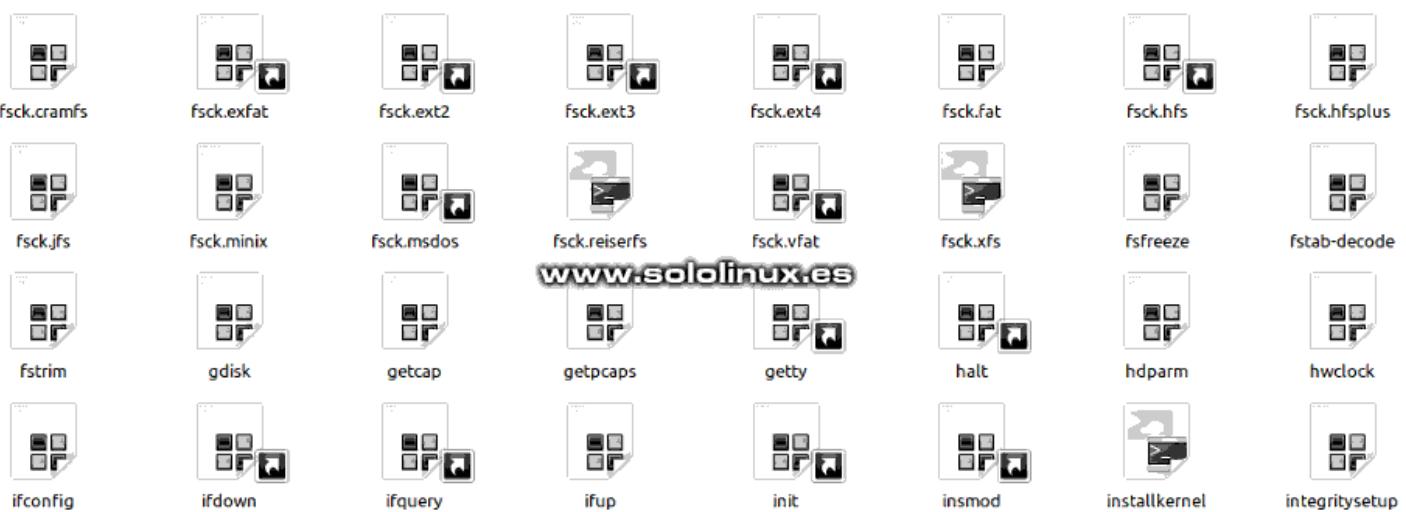
```
rm demo_symbolic demo_symbolic1 demo_symbolic2
```

Si por el contrario y para finalizar el artículo queremos borrar un enlace simbólico o duro, que se encuentra en una ruta, añadimos el argumento **-f**. Por ejemplo...

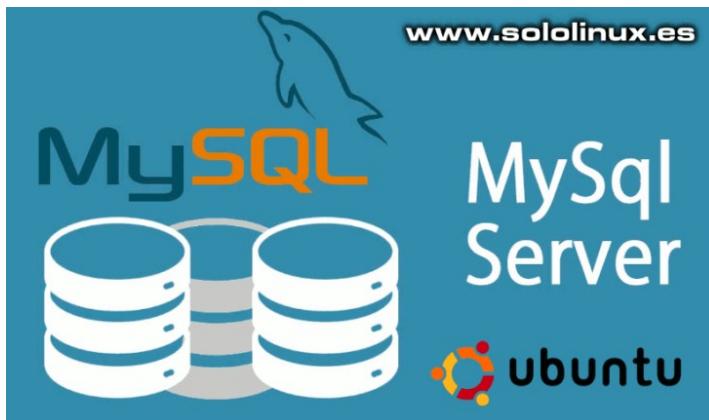
```
rm -f /demo/demo_duro
```

Salida...

```
sergio@sololinux:~$ rm -f /demo/demo_duro
sergio@sololinux:~$
```



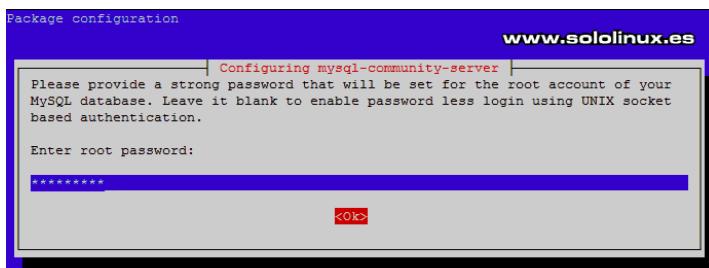
Desinstalar MySQL Server en Ubuntu



Como la mayoría de los sistemas operativos Linux actuales, **Debian** o **Ubuntu** no podían ser menos. La facilidad que supone instalar o desinstalar aplicaciones y herramientas es apabullante.

Aun siendo así, no todas las aplicaciones son tan sencillas de eliminar completamente (sobre todo en servidores). Es posible que sean más complejos en su integración, o su propia naturaleza hacen que dependan de múltiples carpetas de configuración y librerías propias, no fáciles de localizar.

Uno de estos software complejos es MySQL Server. **Desinstalar MySQL Server** no equivale a ejecutar «**apt purge**» o «**apt remove**», como se hace de forma común. Existen pequeños detalles, que te explicamos en este artículo con los que si lograras desinstalar MySQL totalmente y sin morir en el intento.



Desinstalar MySQL Server en Ubuntu

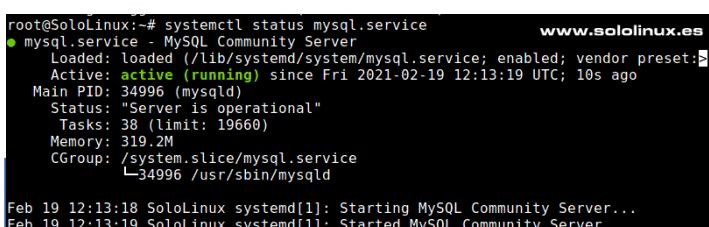
Primero puedes asegurarte de que realmente trabajas con MySQL, y no con otro motor similar. Para ello te propongo dos opciones. Puedes verificar la versión instalada con este comando.

```
mysql -V
```

```
root@SoloLinux:~# mysql -V
mysql Ver 8.0.23-0ubuntu0.20.04.1 for Linux on x86_64 ((Ubuntu))
root@SoloLinux:~#
```

O bien... verificar su estado.

```
systemctl status mysql.service
```



Bueno, es evidente que tenemos MySQL instalado, no perdamos más tiempo vamos a desinstalarlo totalmente sin dejar rastro en nuestro sistema.

```
sudo apt purge mysql-server*
```

```
root@SoloLinux:~# sudo apt purge mysql-server*
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'mysql-server-5.5' for glob 'mysql-server*'
Note, selecting 'mysql-server-5.6' for glob 'mysql-server*'
Note, selecting 'mysql-server-5.7' for glob 'mysql-server*'
Note, selecting 'mysql-server' for glob 'mysql-server*'
Note, selecting 'mysql-server-8.0' for glob 'mysql-server*'
Note, selecting 'mysql-server-core-5.5' for glob 'mysql-server*'
Note, selecting 'mysql-server-core-5.6' for glob 'mysql-server*'
Note, selecting 'mysql-server-core-5.7' for glob 'mysql-server*'
Note, selecting 'mysql-server-core-8.0' for glob 'mysql-server*'
Package 'mysql-server-5.5' is not installed, so not removed
Package 'mysql-server-5.7' is not installed, so not removed
Package 'mysql-server-core-5.7' is not installed, so not removed
Package 'mysql-server-core-5.5' is not installed, so not removed
Package 'mysql-server-core-5.6' is not installed, so not removed
Package 'mysql-server-8.0' is not installed, so not removed
The following packages were automatically installed and are no longer
required:
  libatoi1 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 libnumua1 libtimschedule-perl liburi-perl
  mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0
  mysql-client-core-8.0
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
  mysql-server* mysql-server-8.0* mysql-server-core-8.0*
0 upgraded, 0 newly installed, 3 to remove and 0 not upgraded.
After this operation, 126 MB disk space will be freed.
Do you want to continue? [Y/n]
```

Con «**apt purge**» hemos eliminado los binarios y los archivos de configuración, sin embargo, otros archivos de configuración de MySQL y, de la propia base de datos siguen intactos en nuestro sistema. Puedes verlos con estos comandos.

```
ls /etc/mysql
sudo ls /var/lib/mysql
```

```
root@SoloLinux:~# ls /etc/mysql
conf.d  debian.cnf  my.cnf  my.cnf.fallback
root@SoloLinux:~# sudo ls /var/lib/mysql
#ib_16384_0.dblwr  binlog.000001  ca-key.pem          debian-
5.7.flag  ibdata1  private_key.pem  sys
#ib_16384_1.dblwr  binlog.000002  ca.pem            ib_buffer_pool
mysql    public_key.pem  undo_001
#innodb_temp    binlog.000003  client-cert.pem  ib_logfile0
mysql.ibd    server-cert.pem  undo_002
auto.cnf    binlog.index  client-key.pem  ib_logfile1
performance_schema  server-key.pem
```

Demasiados archivos quedaron en el sistema. Los eliminamos para no dejar rastro de nada.

```
sudo rm -r /etc/mysql /var/lib/mysql
```

```
sudo rm -r /var/log/mysql
```

Para concluir lanzamos un autoremove.

```
sudo apt autoremove
```

```
root@SoloLinux:~# sudo apt autoremove
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be REMOVED:
  libatoi1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl libevent-core-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 libnumua1 libtimschedule-perl liburi-perl mecab-ipadic-utf8 mecab-utils mysql-client-8.0
  mysql-client-core-8.0
0 upgraded, 0 newly installed, 23 to remove and 0 not upgraded.
After this operation, 123 MB disk space will be freed.
Do you want to continue? [Y/n] ■
```

Una vez ejecutado el anterior comando, ya no quedará ni rastro de MySQL Server en Ubuntu y derivados.

Diferencias entre Ubuntu y Gentoo

Como dicen en mi tierra, el hábito no hace al monje, pero sí que es el mayor enemigo del cambio. El ser humano se mueve por costumbres y, los usuarios de Linux no podíamos ser una excepción. Independientemente de cuál fue tu primera **distribución linux**, es normal que estés acostumbrada a ella u otras con un motor similar.

Cambiar no es fácil, pero a medida que tus conocimientos o necesidades aumentan, a veces es necesario. Si es tu caso, tal vez estás considerando dicha posibilidad, soy consciente de que no siempre es fácil aprender un nuevo sistema. Si eres usuario de **Ubuntu**, **Linux Mint** o cualquiera de sus derivados, lo más sencillo es subir a **Arch Linux**; Un excelente sistema, pero que a la larga te darás cuenta de que viene a ser lo mismo que Ubuntu, con la única diferencia de sus comandos particulares.



Si lo que buscas es crear un sistema ligero y totalmente personalizado, debes subir el listón y pensar en **Gentoo**. En este artículo no vamos a convencer a nadie, ni mucho menos intentar que te lances al vacío. **Gentoo** son palabras mayores, nosotros en este artículo simplemente trataremos las principales **diferencias entre Ubuntu y Gentoo**, después... tú decides.



Diferencias entre Ubuntu y Gentoo

Ubuntu vs Gentoo, comparamos dos excelentes sistemas basados en linux.

Ubuntu



Ubuntu es una versión pulida de Debian y, sin dudarlo la **distribucion linux** más utilizada en la actualidad. Si buscas información sobre linux, segura que te aparecerán muchas referencias sobre ella.

Independientemente de la versión seleccionada, Ubuntu se presenta con un trabajado instalador gráfico. Por defecto, Ubuntu viene con el entorno de **escritorio GNOME**, pero puedes optar por otras versiones con XFCE, KDE, etc. Existe una variedad de preferencias impresionante, incluyendo con ajustes preestablecidos totalmente diferentes.

La forma estándar de agregar software, es mediante sus propios repositorios, estos distribuyen archivos en formato **deb de Debian** que son bastante fiables. También se ofrecen otras fórmulas como el **discutido snap** (impuesto por Canonical), o los recomendados **Flatpak** y **AppImage**.

Gentoo

Gentoo es de otra galaxia, está dirigido a usuarios experimentados, por lo que no está recomendado para usuarios inexpertos. La primera en la frente... nada más iniciar Gentoo, descubrimos que su instalador es en línea de comandos, lo que quiere decir que necesitas compilar el software a instalar.

Cuando eliges **Gentoo** como sistema, debes tener conocimientos amplios sobre el manejo de la línea de comandos. Aparte de eso, si eres avanzado tampoco necesitaras un escritorio predeterminado; pero si te resulta necesario, lo puedes instalar sin problemas.

Puedes pensar que un instalador en línea de comandos es un inconveniente, yo te afirmo que es todo lo contrario, es una gran ventaja una vez que te acostumbras a él. El administrador de paquetes (también en línea de comandos), es muy particular y, tiene una cantidad de características que sorprende a propios y extraños, ¡¡¡incluso noticias!!!.

Si tienes dudas de su instalador, puedes agregar soporte para **Flatpak**. También admite **AppImage**, pero necesitas instalar **libfuse** que se entrega como **sys-fs /fuse**.



Diferentes filosofías

Ubuntu usa el administrador de paquetes Debian, principalmente con paquetes binarios. Normalmente el código fuente también está disponible para la mayoría de los paquetes, por tanto la instalación es más rápida. Además, saltar entre diferentes administradores de escritorio no supone ningún problema.

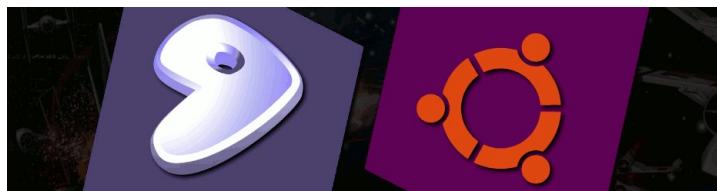
Por otro lado, **Gentoo** tiene como objetivo ofrecer el código fuente y que el instalador lo compile de una forma u otra, dependiendo de en qué plataforma se instale. Esto crea un sistema extremadamente optimizado para tu sistema en particular. Incluso puedes elegir tu modelo de CPU si quieras. Esto quiere decir que una aplicación que en Ubuntu pesa 100MB, en Gentoo pesa 30MB. La diferencia es considerable.

Otra de las principales diferencias entre Ubuntu y Gentoo, es la variable USE. Con ella puedes forzar que el binario solo admita un escritorio específico. Aparte de eso, también puedes compilar cada paquete e instalarlo en ambos sistemas de forma independiente.

Diferencias entre paquetes

Los paquetes en Gentoo contienen enlaces **upstream** y, normalmente no incluyen su código fuente. Su sistema de paquetes, te ayuda a configurar las opciones de compilación y a manejar los parches. No es tan difícil.

Por el contrario, los paquetes en Ubuntu contienen todo el código binario o fuente. Las dependencias están controladas por el sistema, pero ojo... **Gentoo** puede usar los parámetros llamados slots, que permiten tener varias versiones instaladas de una misma aplicación; En Ubuntu debes usar aplicaciones externas para lograr este efecto, o versiones portables.



Diferencias de uso e instalación

El **objetivo de Gentoo** es optimizar cada instalación, dependiendo del sistema anfitrión. El proceso es largo y tedioso, pero esto conducirá a un sistema mucho más rápido, eficaz y estable. Esto puede ahuyentar a muchos usuarios, aun siendo así, contando que tus requerimientos no son excesivos, tal vez te puedas decantar por alguno de sus derivados que incluyen instalador gráfico.

En un próximo artículo veremos los más importantes. Si no puedes esperar... empieza por **Sabayon**, un gran derivado italiano de Gentoo que vale la pena conocer. Puedes **descargar Sabayon** con tu escritorio favorito, pulsando en la siguiente imagen.



En Ubuntu, puedes elegir e instalar imágenes desde su **sitio web**. Encontraras un sistema operativo, listo para operar como servidor o sistema de escritorio pre-configureado desde el momento de su instalación. También se ofrecen los paquetes fuente y compilarlo tu mismo, pero no tiene nada que ver con Gentoo.

En su **cdimage** puedes descargar todas las versiones disponibles.

- [Descargar Ubuntu](#)

Resultado final

Como era de esperar, Ubuntu es mucho mejor en facilidad de instalación, tanto del propio Ubuntu como de

paquetes adicionales. **Gentoo tiene** la gran ventaja, de ser mucho mejor cuando hablamos de rendimiento, ese es su objetivo principal. **Diferencias entre Ubuntu y Gentoo**.

Debes elegir según tus prioridades y necesidades, sin miedo y al toro (que se dice en España). Es posible que te decantes por Gentoo, pero no te olvides que hablamos de un sistema para usuarios avanzados, no medios, ni mucho menos noveles.



Los 7 mejores derivados de Gentoo



A diferencia de otras **distribuciones Linux** como Ubuntu, Fedora, etc., **Gentoo** es bastante desconocido. Por ese motivo creamos un anterior artículo, en el cual **comparábamos a Ubuntu con Gentoo** (una forma de darlo a conocer).

La exclusividad de **Gentoo Linux** es, porque necesita construirse a partir del código fuente. Esto requiere muchas horas de dedicación, a veces incluso interminables días. Esta singularidad de Gentoo, hace de ella una espada de doble filo. Por un lado obtenemos un rendimiento y una estabilidad realmente increíble (hemos construido según nuestra máquina); Y por el otro, el tiempo, la molestia y, los altos conocimientos necesarios conducen a la frustración del usuario, que finalmente abandona el proyecto.

Por suerte para los usuarios menos avanzados, existen derivados personalizados de **Gentoo Linux** que eliminan la dificultad de la instalación y, acercan la distribución a usuarios no profesionales. Estos **7 mejores derivados de Gentoo Linux** que hoy proponemos, no son oficiales. Son creados por entusiastas y desarrolladores de Gentoo, que pretenden acercar este Linux a todo el mundo.

Los 7 mejores derivados de Gentoo

Calculate Linux



Hasta hace pocos meses, no era común el uso de **Calculate Linux** en sistemas personales o de escritorio. Sin embargo, tal como comentamos en un [anterior artículo](#), su nueva versión (20.6) se ha ganado el respeto de toda la comunidad. Actualmente, sus instalaciones aumentan a un ritmo vertiginoso.

Disponible en diferentes entornos de escritorio, cualquiera de las variantes de Calculate Linux se puede usar en modo Live CD/USB. Este **derivado de Gentoo** es muy estable, además funciona sin problemas en hardware un tanto obsoleto. Vemos sus principales características.

- Compatible con versiones anteriores de **Gentoo** Linux original.
- Puedes elegir entre varios entornos de escritorio, como: KDE, Xfce, Cinnamon, MATE, etc.
- La versión de servidor ofrece herramientas de gestión remota y, funciones de autenticación de usuarios.
- Es ideal para redes con máquinas Linux y Windows, a la vez.

Puedes **descargar Calculate Linux** desde su página oficial:

- [Descargar Calculate Linux](#)

Sabayon Linux

Sabayon Linux era el derivado de Gentoo más popular, hasta que Calculate le disputa el trono. Este proyecto iniciado por un desarrollador italiano, tiene una filosofía principal muy marcada, ofrecer a los usuarios una experiencia directa y sencilla.



Sabayon viene precargado con todas las aplicaciones necesarias, para instalar y trabajar (esto hace que su **Live CD/USB** sea un poco pesada). Usa el conocido sistema de gestión de paquetes **Portage de Gentoo**, pero también cuenta con otro administrador de paquetes llamado **Entropy**. Vemos sus principales características.

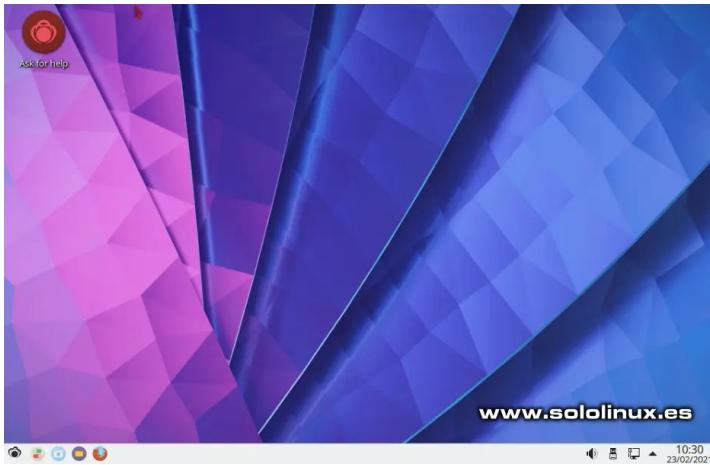
- Versiones con varios entornos de escritorio, KDE, XFCE y GNOME.
- Un instalador fácil de usar, sin la necesidad de construir el flasheable desde el código fuente.
- Excelente soporte con máquinas del tipo **Banana Pi**, **Raspberry Pi**, etc.
- Sabayon usa un sistema de lanzamiento continuo, muy fácil de actualizar.

- El soporte para este sistema es excelente, blog oficial, foro, wiki e incluso chat en vivo.

Puedes **descargar Sabayon Linux** desde su página oficial:

- [Descargar Sabayon Linux](#)

Redcore Linux



Redcore Linux, tal vez sea la distribución más equilibrada entre Gentoo y las distros tradicionales a las que estamos acostumbrados. A diferencia de otros derivados de Gentoo, Redcore se permite el lujo de instalar paquetes sin construirlos desde cero, además nos ofrece un repositorio de software separado, que contiene una base de código prediseñada.

Vemos sus principales características.

- Es muy amigable con usuarios novatos, ya que se eliminaron muchas características complejas de Gentoo.
- Tiene muchas herramientas preinstaladas para producción, juegos y entretenimiento.
- Su repositorio exclusivo no es demasiado grande. Pero sus paquetes son de alta calidad y, no tienen un consumo excesivo.
- Sigue un modelo de lanzamiento continuo que es capaz de ofrecer actualizaciones fluidas durante toda la vida útil de su máquina.
- Solo tiene versión **KDE Plasma**, por tanto, todo el desarrollo está integrado a la perfección.

Puedes descargar Redcore Linux desde su página oficial:

- [Descargar Redcore Linux](#)

Gentoo Studio



Gentoo Studio es un mod no oficial, que convierte **Gentoo** en un estudio de audio digital o DAW basado en Gentoo Linux. Existe **Ubuntu Studio** con funciones muy similares, pero si quieres el poder de Gentoo... **Gentoo Studio** debe ser tu primera elección.

Vemos sus principales características.

- Muy fácil de instalar y de configurar.
- No tiene **bloatware** (software inflado), aparte de algunas herramientas útiles para ingeniería de audio.
- Es liviano y personalizable, para lograrlo usa el escritorio XFCE.
- Contiene un kernel de Linux parcheado, que da prioridad a los equipos de audio conectados.
- Viene con una herramienta, que detecta componentes de audio conectados por USB, desde la propia terminal.
- Necesita de una instalación minimal de Gentoo.

Puedes descargar Gentoo Studio desde su página oficial:

- [Instalar Gentoo Studio](#)

Funtoo

Posee unas características similares a las de Gentoo, pero con algunas mejoras significativas. Esta es una distribución Linux muy fácil de usar. Cualquiera con pocos conocimientos sobre Linux, puede empezar a utilizar **Funtoo Linux** y trabajar sin problemas.

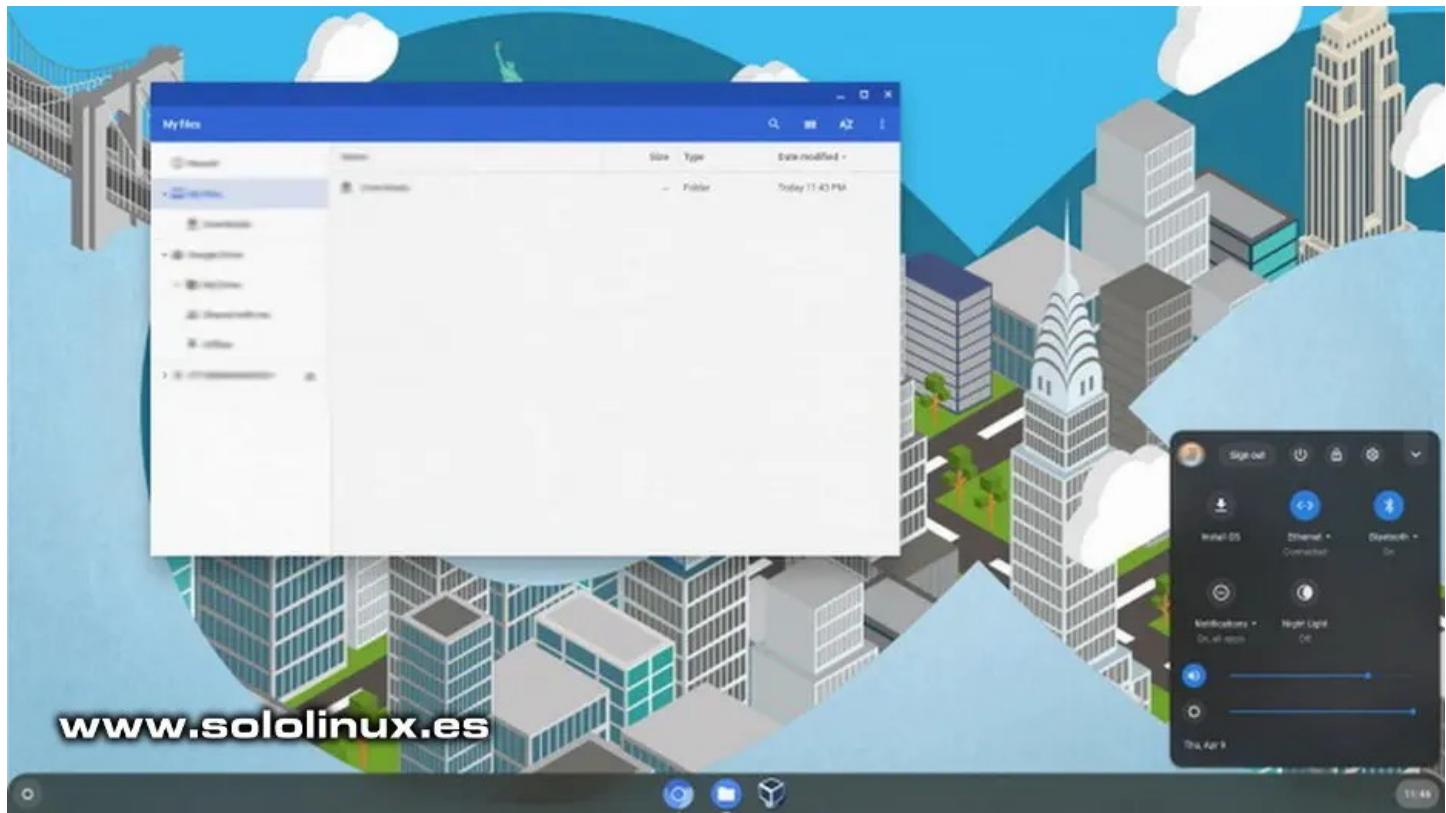
- [Instalar Funtoo Linux](#)

Pentoo

Esta distribución derivada de Gentoo, está orientada a las pruebas de penetración y seguridad de la red. Viene con muchas herramientas test, similares a las que trae por defecto **Kali Linux**. Esta distribución no está diseñada para instalarse en un disco duro, puedes usar Pentoo desde un dispositivo USB.

- [Descargar Pentoo desde su página oficial](#)

Los 7 mejores derivados de Gentoo

CloudReady

CloudReady es una buena alternativa a **Chrome OS** (con hardware personalizado). Este sistema operativo se creó tomando la base del proyecto Chromium. Tal vez te sorprenda, pero Chromium OS es un derivado de Gentoo Linux.

Neverware desarrolla tres versiones, la Home es completamente gratuita. Puedes revisar su página oficial, e instalar este derivado.

- [Descargar CloudReady desde su página oficial](#)

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

AYUDANOS A SEGUIR
CREENDO

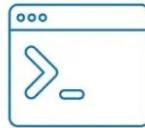


Los 7 mejores derivados de Gentoo

www.sololinux.es

Instalar Google Cloud SDK en Ubuntu 20.04

www.sololinux.es



Google Cloud Platform

Google Cloud SDK es un conjunto de herramientas y aplicaciones, que ayudan a los desarrolladores y usuarios en la tarea de administrar los servicios en la nube, que tengan alojados en [Google Cloud Platform](#).

El conjunto de herramientas empaquetado, contiene entre otros los **comandos gcloud, gsutil**, además de **bq**. Está disponible en formato de paquete, para su instalación en Ubuntu y otros sistemas Linux. Nosotros en este artículo nos centraremos en Ubuntu.

En Ubuntu disponemos de varias formas de **instalar Google Cloud SDK**, una de ellas es mediante **Snap**. Como es habitual en **sololinux** preferimos evitar el tipo de paquetes mencionado anteriormente, nosotros lo instalamos desde su [«repositorio oficial»](#).



Google Cloud SDK

www.sololinux.es

Instalar Google Cloud SDK en Ubuntu 20.04

Lo primero que debes hacer es actualizar el sistema.

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

Instalamos algunas herramientas y librerías necesarias.

```
sudo apt install curl apt-transport-https ca-certificates gnupg

root@SoloLinux:~# sudo apt install curl apt-transport-https ca-certificates gnupg
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20210119~20.04.1).
curl is already the newest version (7.68.0-1ubuntu2.4).
gnupg is already the newest version (2.2.19-3ubuntu2.1).
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 1704 B of archives.
After this operation, 161 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Una vez termine el proceso anterior, agregamos el repositorio oficial que Google pone a nuestra disposición.

```
echo "deb [signed-by=/usr/share/keyrings/cloud.google.gpg]
https://packages.cloud.google.com/apt cloud-sdk main" | sudo tee -a
/etc/apt/sources.list.d/google-cloud-sdk.list
```

```
curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key --keyring /usr/share/keyrings/cloud.google.gpg add -
```

Procedemos a instalar Google Cloud SDK.

```
sudo apt update
sudo apt install google-cloud-sdk
```

```
google-cloud-sdk-pubsub-emulator google-cloud-sdk-bigtable-emulator
google-cloud-sdk-datastore-emulator kubectl
The following NEW packages will be installed:
  google-cloud-sdk python3-crcmod
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 92.4 MB of archives.
After this operation, 495 MB 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 python3-crcmod amd64 1.7+dfsg-2build2 [18.8 kB]
Get:2 https://packages.cloud.google.com/apt cloud-sdk/main amd64 google-cloud-sdk all 329.0.0-0 [92.4 MB]
Fetched 92.4 MB in 2s (46.9 MB/s)
Selecting previously unselected package python3-crcmod.
(Reading database ... 22654 files and directories currently installed.)
Preparing to unpack .../python3-crcmod_1.7+dfsg-2build2_amd64.deb ...
Unpacking python3-crcmod (1.7+dfsg-2build2) ...
Selecting previously unselected package google-cloud-sdk.
Preparing to unpack .../google-cloud-sdk_329.0.0-0_all.deb ...
Unpacking google-cloud-sdk (329.0.0-0) ...
Setting up python3-crcmod (1.7+dfsg-2build2) ...
Setting up google-cloud-sdk (329.0.0-0) ...

[Progress: 1/73% [=====]]
```

Bien..., ahora lanzamos la herramienta con el siguiente comando.

```
sudo gcloud init
```

Veremos una serie de avisos, pulsa «Y» para continuar.

```
root@SoloLinux:~# sudo gcloud init
Welcome! This command will take you through the configuration of
gcloud.
Your current configuration has been set to: [default]
You can skip diagnostics next time by using the following flag:
  gcloud init --skip-diagnostics
Network diagnostic detects and fixes local network connection issues.
Checking network connection...done.

Reachability Check passed.
Network diagnostic passed (1/1 checks passed).
You must log in to continue. Would you like to log in (Y/n)?
```

Google Cloud SDK nos ofrece una url, que debes copiar y pegar en tu **navegador web** favorito para obtener la clave de seguridad. Un ejemplo.

```
Go to the following link in your browser:
https://accounts.google.com/o/oauth2/auth?
response_type=code&client_id=255255255255.apps.googleusercontent.co
m&redirect_uri=urn%3Aietf%3Awg%3Aoauth%3V2%3A0ob&scope=openid+https%
3A%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%0A%7C%
2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%
2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%
2Fwww.googleapis.com%2Fauth%2Fcompute+https%5Y%6T%
%2Fwww.googleapis.com%2Fauth%
%2Faccounts.reauth&state=aEMPr48U1qTUhfytftFDfTyf6r6RFr6r6rfYFyfYf&
i&prompt=consent&access_type=offline&code_challenge=8Kwe8ds3hakgt6T&&r
6r6r6ggyggV/vtGFYft6gfYFyf&code_challenge_method=S256
Enter verification code:
```

Iniciamos sesión en Google Cloud SDK.



Obtenemos la key.



Solo falta introducir en la terminal, la key que nos concede **Cloud SDK** y acceder / seleccionar el proyecto en el que necesitamos trabajar.

```
Enter verification code: [REDACTED]
You are logged in as: [REDACTED]

Pick cloud project to use:
[1] [REDACTED]
[2] [REDACTED]
[3] [REDACTED]
[4] Create a new project
Please enter numeric choice or text value (must exactly match list
item): [REDACTED]
```

www.sololinux.es

Como puede comprobar, **instalar Google Cloud SDK** en Ubuntu 20.04 y operar con él, es tarea sencilla.

Canales de Telegram:

[Canal SoloLinux – Canal SoloWordpress](#)

Espero que este artículo te sea de utilidad, puedes ayudarnos a mantener el servidor con una donación ([paypal](#)), o también colaborar con el simple gesto de compartir nuestros artículos en tu sitio web, blog, foro o redes sociales. Instalar Google Cloud SDK en Ubuntu 20.04.

[Chat de SoloLinux en Telegram](#)

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

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

AYUDANOS A SEGUIR
CRECIENDO



www.sololinux.es

Como firmar documentos pdf en linux



A la hora de crear documentos o publicaciones, sin duda alguna, PDF (**Portable Document Format**) es el formato más extendido, ya que permite compartir cualquier escrito en prácticamente la totalidad de plataformas existentes. Nuestra [revista digital SoloLinux](#), también la puedes encontrar en **formato pdf**.

La mayoría de **navegadores web** actuales, incorporan su propio **visor de documentos PDF**, por tanto, puedes abrir y ver archivos PDF sin problemas, en prácticamente cualquier dispositivo moderno con conexión a internet. Sin embargo, modificar archivos de documentos PDF ya creados, no siempre es tan sencillo.

En el artículo de hoy, vemos cómo agregar una firma a un documento PDF. Recuerda que **Adobe Acrobat Reader** en su versión oficial, sí que permite firmar un documento ya sea mediante escritura, realizando un dibujo o insertando una imagen de tu firma manuscrita. Pero... como es habitual en la **empresa Adobe** (fiel servidor de su amo Microsoft, siempre arrodillado), Acrobat Reader ya no es compatible con Linux.

Tranquilo, la **herramienta Xournal** nos ofrece la solución perfecta.



Como firmar documentos pdf en linux

Xournal es una aplicación con interfaz gráfica, desarrollada principalmente para tomar de notas y crear bocetos. Muy útil si trabajas con pantalla táctil, también nos permite editar archivos PDF ya creados y, exportarlos a pdf una vez han sido modificados. Como norma general, Xournal no viene instalado por defecto en casi ninguna **distribución linux**. La instalamos.

Instalar Xournal en Ubuntu, Debian y derivados

```
sudo apt install xournal
```

Instalar Xournal en Fedora, CentOS 8 y derivados

```
sudo dnf install xournalpp
```

Instalar Xournal en CentOS 7,Rhel 7 y derivados

```
yum install epel-release  
yum install xournalpp
```

Instalar Xournal en Arch Linux, Manjaro y derivados

```
sudo pacman -S xournalpp
```

Ejemplo de instalación en Ubuntu,

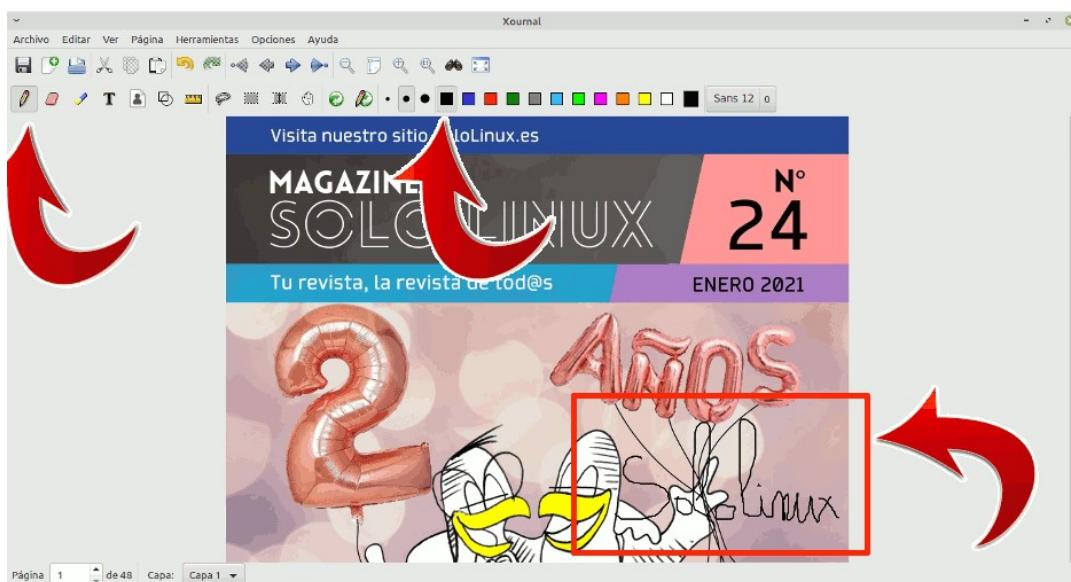
```
root@sololinux:~# sudo apt install xournal
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
 libart-2.0-2 libgnomecanvas2-0 libgnomecanvas2-common
Se instalarán los siguientes paquetes NUEVOS:
 libart-2.0-2 libgnomecanvas2-0 libgnomecanvas2-common xournal
0 actualizados, 4 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 413 kB de archivos.
Se utilizarán 1.403 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

**Firmar documentos pdf con Xournal**

Xournal nos permite tres formas de firmar documentos PDF. Puedes dibujar directamente una firma con lápiz de la herramienta, escribir la firma desde tu teclado, o importar una imagen de firma e integrarla en el documento. Vemos las dos opciones.

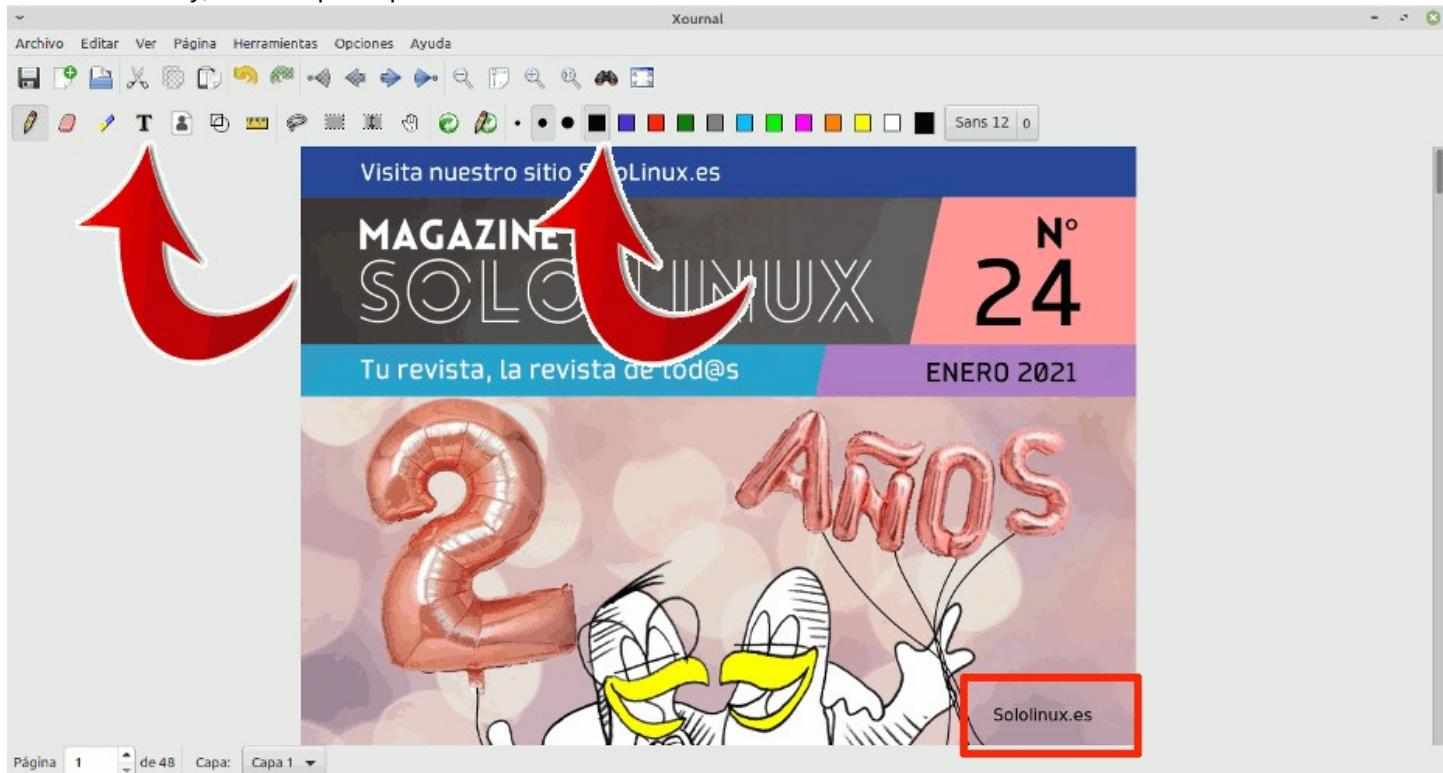
Firmar pdf con dibujo

Es tan fácil como abrir el **archivo pdf**, seleccionar el lápiz, el tamaño y el color.



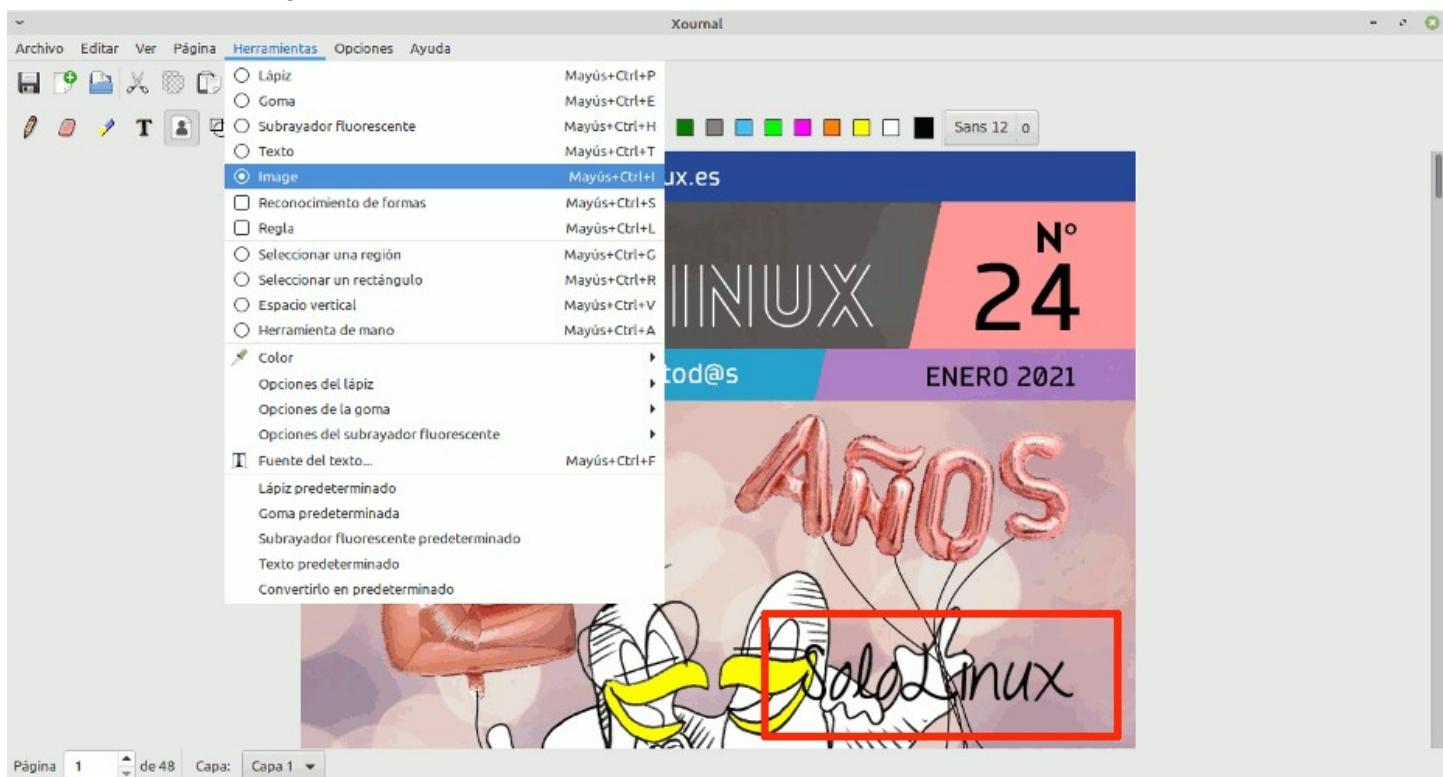
Firmar pdf con escritura

La herramienta también nos permite insertar una firma directamente desde el teclado físico de tu sistema. Seleccionas el icono texto y, el color que tú prefieras.

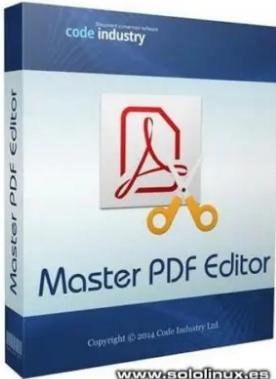


Firmar pdf insertando una imagen

Por último vemos como insertar una imagen con tu firma. Para ello seleccionas el icono de imagen, o bien haces clic en herramientas / imagen.



Master PDF Editor 4.3.89 – Gratis para linux



Master PDF Editor

Master PDF Editor, es una herramienta comercial increíblemente poderosa a la hora de **editar documentos PDF**. Desarrollada por la empresa «**Code Industry**», es Multiplataforma, por tanto es funcional en linux, Windows y MacOS.

La aplicación permite crear, editar o insertar texto e imágenes, anotar, visualizar, cifrar, firmar documentos PDF y, un mucho más. Bueno... lo dicho anteriormente no es del todo cierto.

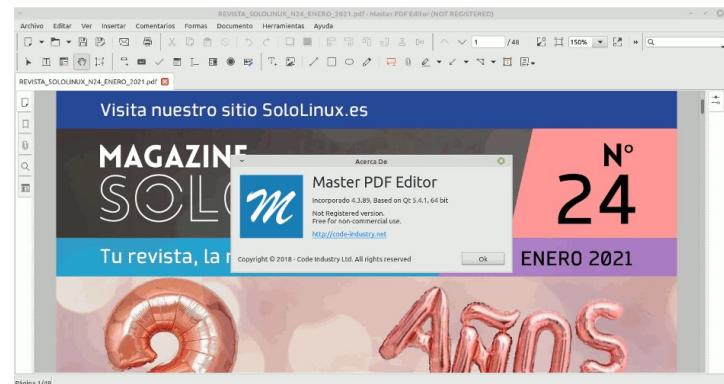
Con el lanzamiento de la versión 5.x, los desarrolladores de **Master PDF Editor** eliminaron algunas funciones que antes eran gratuitas para los usuarios de linux, con uso no comercial. Ahora... al editar o agregar texto, imágenes, firmas etc., se inserta una marca de agua en el documento y, por si fuera poco no solo inserta la marca de agua en la **página pdf** editada, noooo, se inserta en todas las páginas del documento (la marca de agua es bastante grande).



Master PDF Editor 4.3.89 – Gratis para linux

Por si te interesa, **Master PDF Editor 4.3.89** sigue siendo una excelente herramienta totalmente funcional, además de ser gratuita para uso no comercial y sin restricciones para los usuarios de linux. Vemos sus principales características.

- Permite manipular texto y objetos en **documentos PDF**; Editar texto o agregar uno nuevo, insertar imágenes, mover y redimensionar objetos y, guardar las imágenes de un documento PDF.
- Anotaciones en PDF: resaltar, tachar o subrayar texto, añadir comentario, notas adhesivas, insertar flechas, líneas, rectángulos, sellos, etc.
- Insertar nuevos formularios en PDF.
- Todo tipo de operaciones sobre las páginas del documento PDF (rotar, mover, eliminar, añadir, etc.).
- Edición de los marcadores PDF.
- Firmar documentos PDF con certificado digital, incluyendo las operaciones de verificado y creación.
- Capacidad de optimización de los PDF.
- Encriptar archivos PDF (solo en versión **Qt5**).



Descargar Master PDF Editor

Es evidente que en este artículo ofrecemos la versión funcional, soy consciente que ya tiene unos días, pero sus resultados siguen siendo óptimos. Existen versiones para Qt5 que solo soportan 64 bits y, las de Qt4 con versiones de 64 y 32 bits.

Qt5 64 bits

Para Debian, Ubuntu y derivados

- [Descargar Master PDF Editor en deb – Qt5](#)

Para Fedora, Centos y derivados

- [Descargar Master PDF Editor en rpm – Qt5](#)

Autoejecutable portable

- [Descargar Master PDF Editor portable – Qt5](#)

Qt4 64 y 32 bits

Para Debian, Ubuntu y derivados de 32 bits

- [Descargar Master PDF Editor en deb 32 bits – Qt4](#)

Para Fedora, Centos y derivados de 32 bits

- [Descargar Master PDF Editor en rpm 32 bits – Qt4](#)

Autoejecutable portable de 32 bits

- [Descargar Master PDF Editor portable 32 bits – Qt4](#)

Autoejecutable portable de 64 bits

- [Descargar Master PDF Editor portable 64 bits – Qt4](#)

Todas las descargas son directas desde nuestros propios servidores, no abuses de ellas o entorpecerás al resto de usuarios. Gracias por vuestra comprensión.

Nombre del archivo	Size (KB)
master-pdf-editor-4.3.89_i386.deb	10299
master-pdf-editor-4.3.89_i386.rpm	14425
master-pdf-editor-4.3.89_i386.tar.gz	14559
master-pdf-editor-4.3.89_qt4.amd64.tar.gz	14504
master-pdf-editor-4.3.89_qt5.amd64.deb	10364
master-pdf-editor-4.3.89_qt5.amd64.tar.gz	14650
master-pdf-editor-4.3.89_qt5.x86_64.rpm	14523

www.sololinux.es



www.sololinux.es

Raspberry Pi 3 vs Raspberry Pi 4



Lanzado en 2012, **Raspberry Pi** tenía la intención de ser un dispositivo destinado a la enseñanza e investigación por parte de los estudiantes. Su diminuto tamaño, bajo costo y alta modularidad, lo convirtió rápidamente en un éxito. La gran aceptación no solo atrajo a los aficionados de la **tecnología**, también a grandes empresas que revolucionaron el **IoT** (Internet de las cosas), la robótica, proyectos de electrónica y, otros a nivel industrial.

Realmente, estamos hablando de una pequeña computadora que ya va por su cuarta generación. Salvo contadas excepciones, cada generación cuenta con dos versiones, el **A** y el **B**, pero sucesivas revisiones y mejoras derivan en otros modelos como el **A+**, o el **B+**.

Como era de esperar, los dispositivos de tercera y cuarta generación son los más habituales en la actualidad. También es evidente, que el de cuarta generación tiene un coste mayor (es más poderoso). En el artículo de hoy intentaremos conocer un poco mejor cada **dispositivo**, y si vale la pena el sobre coste de uno sobre otro.

Raspberry Pi 3 vs Raspberry Pi 4

Raspberry Pi 3 al igual que Raspberry Pi 4, ofrece las mismas funciones que una computadora básica pero con una gran diferencia, todo lo necesario viene integrado en una sola placa. Procesadores con una **arquitectura ARM**, memoria **RAM**, puerto Ethernet, salidas a monitor o pantalla, puertos USB, **Wi-Fi**, Bluetooth, y un **conector GPIO** de 40 pines.

Estos dos dispositivos difieren en rendimiento, conexiones y otras capacidades. Con el fin que puedas decidir cuál te conviene más, en la comparativa **Raspberry Pi 3 vs Raspberry Pi 4**, tomaremos un **Raspberry Pi 3 B+** y un **Raspberry Pi 4 B**.

Que dispositivo rinde más

Al hablar de rendimiento, está claro que **Raspberry 4** es muy superior. Un procesador **Broadcom** de cuatro núcleos a 1,5 GHz, una tecnología de memoria más moderna que puedes elegir entre 1 GB y 8 GB, junto a una GPU **Broadcom Videocore VI**, hacen que estemos ante una buena máquina (si hablamos de Raspberry Pi).

Por otro lado tenemos a la Raspberry Pi 3 B+, que aunque no cuenta con tantas opciones de **memoria ram**, ni con una cpu tan potente, también nos puede ofrecer un rendimiento aceptable al combinar 1GB de RAM, con una GPU Broadcom Videocore IV.

Vemos una tabla de componentes que nos lo aclara mejor. **Raspberry Pi 3 vs Raspberry Pi 4**.

	Procesador	Memoria RAM	GPU
Raspberry Pi 3 B+	Broadcom BCM2837B0, Quad-core Cortex-A53 (ARMv8) 64-bit SoC, 1.4GHz	SDRAM LPDDR2 de 1 GB	Broadcom Videocore VI
Raspberry Pi 4 B	Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) 64-bit SoC, 1.5GHz	SDRAM LPDDR4 de 1GB/2GB/4GB/8GB	Broadcom Videocore VI



Raspberry Pi 3 Model B+

The final revision of our third-generation single-board computer

1.4GHz 64-bit quad-core processor, dual-band wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and Power-over-Ethernet support (with separate PoE HAT)

www.sololinux.es

Raspberry Pi 3 vs Raspberry Pi 4 – Cuál es mejor

Audio y Visión

El **Raspberry Pi 3 B+** monta un **puerto HDMI** y reproduce videos hasta 1920×1080p. La calidad del video es satisfactoria, y tiene la particularidad de montar un HDMI estándar (no necesita adaptador). También viene con un conector de audio y video analógico de 3,5 mm.

Por otro lado, **Raspberry Pi 4 B** integra dos conectores micro-HDMI en su placa, esto permite una salida en dos



pantallas con una calidad de reproducción de hasta 4K. El problema... necesitas adquirir dos adaptadores de micro-HDMI a HDMI. Al igual que con **Pi 3 B+**, también viene con un conector de audio y video analógico de 3,5 mm.

Redes y comunicaciones

Las dos **Raspberry Pi** vienen preparadas para conexiones de red cableadas, inalámbricas (Wi-Fi) y Bluetooth. La conexión Gigabit Ethernet del **Pi 3**, se ve obstaculizada por la interfaz USB (están interconectadas) que reduce su rendimiento hasta unos 315 Mbps. En el **Raspberry Pi 4 B**, se corrigió este fallo y ahora ofrece un rendimiento óptimo.

Los **dispositivos inalámbricos** de las dos Raspberry, admiten bandas de 2,4 GHz y 5 GHz. El Bluetooth del **Raspberry Pi 4 B** es la versión 5.0, el **Raspberry Pi 3 B+** viene con la 4.2.

www.sololinux.es

Raspberry Pi 4

Your tiny, dual-display, desktop computer
...and robot brains, smart home hub, media centre,
networked AI core, factory controller, and much more

Almacenamiento y otros puertos

Además de los puertos mencionados anteriormente, estas placas vienen con alguno más, por ejemplo, cuatro puertos USB independientemente de tu Raspberry. Los puertos USB de la **Raspberry Pi 3 B+** son USB 2.0, en la **Raspberry Pi 4 B**, dos son USB 3.0 y los restantes USB 2.0.

La **conexión GPIO** (entrada / salida de uso general) de 40 pines, permite conectar otros módulos adicionales válidos para múltiples funciones. Sus posibilidades son prácticamente infinitas, incluso tus propios experimentos relacionados con la electrónica. Se permite usar como puerto de alimentación.

La alimentación eléctrica del **Raspberry Pi 4 B** es, mediante un puerto USB tipo C, por el contrario, la del **Raspberry Pi 3 B+** en mediante un micro-USB.

Los dos dispositivos tienen una ranura **MicroSD**, que está reservada para una tarjeta del mismo formato cargada con un sistema operativo basado en Linux. También se utiliza como **disco duro**.

Conclusión

La velocidad del **Raspberry Pi 4 B**, te hará decantarte por el de cuarta generación, recuerda que a más velocidad, más consumo y más calor. Dependiendo del uso que le vayas a dar, es posible que debas adquirir un sistema de refrigeración independiente.

Raspberry Pi B 3+ también es una buena opción, además de más económica. Su rendimiento está a la altura de tareas básicas y, exceptuando el doble monitor tampoco notarás grandes diferencias en un uso normal y cotidiano. Tú eliges, Raspberry Pi 3 vs Raspberry Pi 4.

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

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

AYUDANOS A SEGUIR
CREENDO



www.sololinux.es

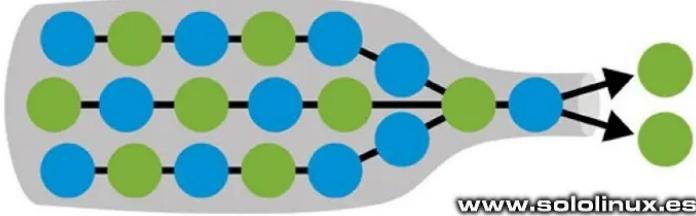
Detectar cuellos de botella con el comando sysdig



El comando sysdig, es una herramienta que te ayuda a resolver problemas del sistema, mediante análisis y exploración. Se usa para capturar, filtrar y decodificar llamadas al sistema y otros eventos.

Sysdig es útil tanto para inspeccionar sistemas activos, como para generar **registros** de seguimiento que puedes analizar posteriormente. Se incluye un potente lenguaje de filtrado, con el cual puedes personalizar la salida o incluirlo en un script.

En este artículo nos centramos en como detectar **cuellos de botella** en nuestro sistema. Sí, hablamos de esos momentos puntuales en los que nuestro sistema se queda medio muerto, normalmente debido a los excesivos procesos que intentan ejecutarse al mismo tiempo.



Detectar cuellos de botella con el comando sysdig

Esta herramienta no viene instalada por defecto en ninguna **distribución linux** (salvo contadas excepciones). Procedemos a su instalación, pero antes quiero avisarte que puede generar problema en sistemas que no tengan instalado **apt** o **yum**.

Vamos a instalar **sysdig** (como **usuario root**, o con permisos).

```
sudo su
curl -s https://s3.amazonaws.com/download.draios.com/stable/install-sysdig | bash
```

```
root@sololinux:# curl -s https://s3.amazonaws.com/download.draios.com/stable/install-sysdig | bash
* Detecting operating system
* Installing Sysdig public key
OK
* Installing sysdig repository
* Installing kernel headers
```

```
* Installing sysdig
Seleccionando el paquete sysdig previamente no seleccionado.
(Leyendo la base de datos ... 343182 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../sysdig_0.27.1_amd64.deb ...
Desempaquetando sysdig (0.27.1) ...
Configurando sysdig (0.27.1) ...
Loading new sysdig-0.27.1 DKMS files...
Building for 5.4.0-66-generic
Building initial module for 5.4.0-66-generic
Secure Boot not enabled on this system.
Done.
sysdig-probe:
Running module version sanity check.
- Original module
  - No original module exists within this kernel
- Installation
  - Installing to /lib/modules/5.4.0-66-generic/kernel/extrudepmode...
DKMS: install completed.
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...
```

Al ejecutar el comando en su formato simple, nos aparece un sinfín de todo lo que sucede en nuestro sistema. No lo recomiendo, pero puedes probar.

```
sudo sysdig
```

Observa el interminable ejemplo...

```
www.sololinux.es
[...]
329953 13:27:15 165223262 1 sysdig (12829) > switch next=12829(sysdig) pgft_m=0 pgft_min=0 vn_size=0 vn rss=0 vn swap=0
329953 13:27:15 165223262 1 <switch next=12829(sysdig) pgft_m=0 pgft_min=0 vn_size=0 vn rss=0 vn swap=0
329948 13:27:15 165233417 1 sysdig (12829) > switch next=6427(pgft_m=0 pgft_min=0 vn_size=1596 vn rss=5344 vn swap=0
329948 13:27:15 165233417 1 <switch next=6427(pgft_m=0 pgft_min=0 vn_size=1596 vn rss=5344 vn swap=0
329943 13:27:15 165239676 1 xfc4-terminal (9153) > write fd=8<--> size=8
329943 13:27:15 165239676 1 xfc4-terminal (9153) > write fd=8<--> size=8
329943 13:27:15 165239676 1 xfc4-terminal (9153) > write fd=8<--> size=155
329946 13:27:15 165286678 1 xfc4-terminal (9153) > read rd=68 data=0 vn_size=155
329946 13:27:15 165286678 1 <read rd=68 data=0 vn_size=155
329947 13:27:15 165291522 1 xfc4-terminal (9153) > read fd=13<--> size=3886
329947 13:27:15 165291522 1 <read fd=13<--> size=3886
329949 13:27:15 165306726 1 xfc4-terminal (9153) > write fd=4<--> size=8
329950 13:27:15 165307483 1 xfc4-terminal (9153) > write res_A data=.....
329950 13:27:15 165307483 1 <write res_A data=.....
329951 13:27:15 165312058 1 xfc4-terminal (9153) > recvmsg fd=10<-->
329951 13:27:15 165312058 1 <recvmsg fd=10<-->
329953 13:27:15 165320686 1 xfc4-terminal (9153) > poll res=5 fd=4<-->
329953 13:27:15 165320686 1 <poll res=5 fd=4<-->
329954 13:27:15 165324731 1 xfc4-terminal (9153) > poll res=1 fd=15<-->
329954 13:27:15 165324731 1 <poll res=1 fd=15<-->
329956 13:27:15 165329159 1 xfc4-terminal (9153) > read res=0 data=.....
329957 13:27:15 165334791 1 xfc4-terminal (9153) > read res=0 data=.....
329957 13:27:15 165334791 1 <read res=0 data=.....
329958 13:27:15 165341684 1 xfc4-terminal (9153) > poll fd=5<--> 5:u1 11:u3 13:f3 timeout=6
329966 13:27:15 165356699 1 sysdig (12829) > switch next=12829(sysdig) pgft_m=4 pgft_min=561 vn_size=651912 vn rss=38968 vn swap=0
329966 13:27:15 165356699 1 <switch next=12829(sysdig) pgft_m=4 pgft_min=561 vn_size=651912 vn rss=38968 vn swap=0
329967 13:27:15 165364747 1 sysdig (12829) > switch next=6427 pgft_m=0 pgft_min=0 vn_size=1596 vn rss=5344 vn swap=0
329967 13:27:15 165364747 1 <switch next=6427 pgft_m=0 pgft_min=0 vn_size=1596 vn rss=5344 vn swap=0
329968 13:27:15 165364749 1 sysdig (12829) > switch next=12829(sysdig) pgft_m=0 pgft_min=0 vn_size=1596 vn rss=5344 vn swap=0
329968 13:27:15 165364749 1 <switch next=12829(sysdig) pgft_m=0 pgft_min=0 vn_size=1596 vn rss=5344 vn swap=0
329973 13:27:15 1653624413 1 xfc4 (6427) > switch next=12829(sysdig) pgft_m=0 pgft_min=0 vn_size=0 vn rss=0 vn swap=0
329973 13:27:15 1653624413 1 <switch next=12829(sysdig) pgft_m=0 pgft_min=0 vn_size=0 vn rss=0 vn swap=0
329976 13:27:15 165465664 1 sysdig (12829) > switch next=6427 pgft_m=0 pgft_min=0 vn_size=1596 vn rss=5344 vn swap=0
329976 13:27:15 165465664 1 <switch next=6427 pgft_m=0 pgft_min=0 vn_size=1596 vn rss=5344 vn swap=0
```

Este comando nos ofrece muchas opciones, pero hoy nos centramos en los cuellos de botella o mejor... los llamamos «**bottlenecks**». Lo puedes ejecutar con este comando.

```
sudo su
sysdig -c bottlenecks
```

Ahora espera unos segundos, por ejemplo 30 o 60 y pulsas la tecla **Ctrl + c**. Verás un resultado similar a...

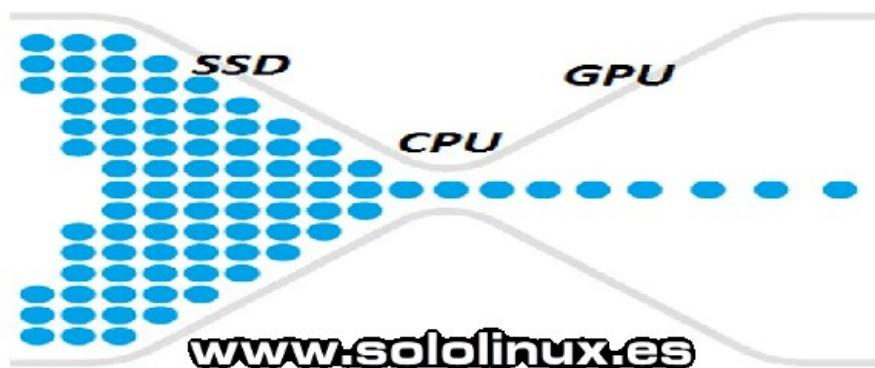
```
root@sololinux:~# sysdig -c bottlenecks
^C231237 0.000000000 ThreadPoolServ (3624) > epoll_wait maxevents=32
1840514 60.060168752 ThreadPoolServ (3624) < epoll_wait res=0
117710 0.000000000 cron (666) > nanosleep interval=60000000000(60s)
1633954 60.000624799 cron (666) < nanosleep res=0
41833 0.000000000 Chrome_ChildIOT (3180) > epoll_wait maxevents=32
1055508 45.263730434 Chrome_ChildIOT (3180) < epoll_wait res=1
41906 0.000000000 chromium-browser (3177) > futex addr=7FFE71E98838
op=128(FUTEX_PRIVATE_FLAG) val=0
1055521 45.262984064 chromium-browser (3177) < futex res=0
286745 0.000000000 ThreadPoolsingl (3030) > futex addr=7F83B873D908
op=128(FUTEX_PRIVATE_FLAG) val=0
1409717 43.552938505 ThreadPoolSingl (3030) < futex res=0
320210 0.000000000 xfc4-terminal (13082) > poll fds=4:e1 5:u1 7:u1 11:u3
12: f3 timeout=4294967295
1335686 37.343041500 xfc4-terminal (13082) < poll res=1 fds=5:u1
611836 0.000000000 ThreadPoolForeg (13007) > futex addr=7F83B873D908
op=128(FUTEX_PRIVATE_FLAG) val=0
1545240 33.001265102 ThreadPoolForeg (13007) < futex res=-110(ETIMEDOUT)
1030559 0.000000000 ThreadPoolForeg (12865) > futex addr=7F83B87FD908
op=128(FUTEX_PRIVATE_FLAG) val=0
2023659 33.000756708 ThreadPoolForeg (12865) < futex res=-110(ETIMEDOUT)
973047 0.000000000 ThreadPoolForeg (9116) > futex addr=7F83B51DC908
op=128(FUTEX_PRIVATE_FLAG) val=0
1951436 33.000576390 ThreadPoolForeg (9116) < futex res=-110(ETIMEDOUT)
239425 0.000000000 ThreadPoolForeg (12865) > futex addr=7F83B87FD908
op=128(FUTEX_PRIVATE_FLAG) val=0
1030565 33.000458604 ThreadPoolForeg (12865) < futex res=-110(ETIMEDOUT)
```

Presta atención a las columnas 2, 3 y 4.

- **Columna 2** : Tiempo de ejecución.
- **Columna 3** : Proceso.
- **Columna 4** : PID del proceso.

Detectar cuellos de botella con el comando sysdig

Bien, una vez identificado el proceso que nos provoca el cuello de botella, es tan sencillo como matarlo con el **comando kill**. Si no recuerdas como, o simplemente no estás seguro de como operar, puedes revisar este [anterior articulo](#).



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

Espero que este artículo te sea de utilidad, puedes ayudarnos a mantener el servidor con una donación ([paypal](#)), o también colaborar con el simple gesto de compartir nuestros artículos en tu sitio web, blog, foro o redes sociales.

Detectar cuellos de botella con el comando sysdig.

[Chat de SoloLinux en Telegram](#)

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

**AYUDANOS A SEGUIR
CREENDO**



ENTREVISTA

Entrevista a Diego Caraballo, administrador del Blog Mi diario Python

The screenshot shows the homepage of the Mi Diario Python blog. At the top, there's a navigation bar with 'Mi Diario Python' on the left and 'Inicio' and 'Categorías' on the right. Below the header, there are several article cards:

- Cómo ejecutar scripts Python automáticamente todos los días en Windows**: Includes a snippet of code for Windows Task Scheduler.
- Formateadores automáticos para utilizar en Python**: Discusses linters like PyLint.
- En esta publicación les mostraré cómo programar la ejecución de un script Python usando el Programador de tareas (Windows Task...)**: Includes a link to 'Leer más...'.
- Leer más...**
- 75%** and **50%** progress bars.
- COVID-19**: An article about web scraping COVID-19 data using BeautifulSoup, featuring a red COVID-19 virus illustration.
- Web Scraping de datos COVID-19 usando BeautifulSoup en Python**: Includes a snippet of Python code for BeautifulSoup.
- Introducción El coronavirus (COVID-19), un tema que nos ha tocado a todos de alguna manera y que hoy es una...**: Includes a link to 'Leer más...'.
- Barras de progreso en las aplicaciones Python con tqdm**: Includes a snippet of Python code for tqdm.
- Si su programa tarda un tiempo en mostrar resultados, evite frustrar a los usuarios mostrando el progreso que está haciendo...**: Includes a link to 'Leer más...'.
- Ejercicios con for en python | Aprende a programar**: Shows a screenshot of a terminal window with a for-loop example.
- Login en Instagram con Python y Selenium**: Shows two cartoon illustrations of a person at a desk and a person working on a laptop.

Un número mas en la Revista SOOLINUX, seguimos con las entrevistas a distintos sitios WEB relacionados con el mundo de GNU/LINUX. Esta vez lo dedicamos a un Blog de nuestro amigo **Diego Caraballo** “MI DIARIO PYTHON” donde con su ayuda puedes aprender de este gran lenguaje de programación

Comenzamos con la entrevista:

Su nombre es **Diego Caraballo**, vive en Uruguay y es Analista Programador.

Si bien su lenguaje de cabecera es **Python**, actualmente trabaja como desarrollador **Full Stack Developer** en el sector local y también de forma remota como **Freelancer**.

SOOLINUX: ¿Cuéntanos un poco sobre tu blog personal <https://pythondiario.com/>?

DIEGO CARABALLO: Mi Diario Python es un blog, que como su nombre lo dice, comenzó como un diario de apuntes personal y hoy lo visitan +- 1500 usuarios únicos de forma diaria.

Se caracteriza por tener muchos ejercicios con soluciones para que principiantes e intermedios puedan practicar y ver diferentes formas de resolver los problemas planteados. También hay un varios tutoriales desde 0 y artículos alusivos a Python.

SOOLINUX: ¿Cuando se empezó con este BLOG? ¿Como fue la idea para crearlo?

DIEGO CARABALLO: El Blog nace en el 2012, cuando comienzo a interesarme por el Desarrollo de Software.

Cuando uno arranca este maravilloso mundo del Desarrollo de Software, es mucha la información que va procesando, sumado a la dificultad que puede presentar en un principio, la idea siempre fue ir publicando lo que aprendía, primero con el objetivo de poder repasar mis apuntes y luego compartirlo para que otros principiantes como yo se vieran beneficiados encontrando soluciones a sus problemas de forma más rápida.

Otra motivación fue aumentar el caudal de información sobre éste lenguaje en español, al corroborar que la misma resultaba escasa.



SOOLINUX: ¿Nos podrías decir que es para ti el lenguaje de programación PYTHON? "Intenta convencernos de que es el mejor lenguaje de programación :)"

DIEGO CARABALLO: Una de las mayores ventajas de Python es que es fácil de leer porque es un lenguaje de programación de alto nivel: se lee bastante parecido al inglés. Incluso los principiantes encontrarán que el código comienza a tener sentido para ellos con bastante rapidez y pueden comprender los conceptos básicos, incluso si no lo saben todo. Esto es útil tanto cuando está comenzando a escribir su propio código como cuando está mirando código escrito por otros programadores.

En general, Python es mucho más fácil de aprender que la mayoría de los lenguajes de programación y es un excelente primer lenguaje de programación.

SOOLINUX: ¿Hoy en día aproximadamente cuantas personas colaboran en el BLOG?

DIEGO CARABALLO: Si bien en la actualidad solo yo estoy publicando en el blog, han pasado varias personas como por ejemplo (Luis Salcedo) que han dejado maravillosos artículos y que hoy en día son de referencia en búsquedas en Google. En este sentido me encuentro inmensamente agradecido con estas personas que han colaborado en blog, compartiendo sus conocimientos.

Próximamente abriré nuevamente la sección de los colaboradores.

SOOLINUX: ¿A que perfil de usuarios estas destinados los artículos del BLOG?

DIEGO CARABALLO: El blog está dedicado a toda persona que le interese Python.

Si bien hay algunas secciones específicas como Tutoriales y Ejercicios, hay muchas publicaciones sobre problemas reales que puede llegar a afrontar un programador con éste lenguaje.

SOOLINUX: ¿Dónde y como podemos ponernos en contacto con vosotros? Publicitáte lo mejor que puedas :)

DIEGO CARABALLO: Pueden ponerse en contacto al correo diegocaraballo84@gmail.com o escribiendo un comentario en el blog.

SOOLINUX: Y para terminar ¿Podrías darnos tu opinión sobre GNU/LINUX y software libre?

DIEGO CARABALLO: El código abierto es increíble, y hay muchas razones por las que podría considerar consumir, publicar, colaborar o dar soporte al código abierto.

Sus cuatro libertades principales para mi lo dicen todo: la libertad de ejecutar el software, la libertad de estudiar y modificar el software, la libertad de redistribuir el software y la libertad de distribuir sus modificaciones. Pueden visitar mi cuenta de GitHub para ver mis repositorios y en los que colaboro:

- <https://github.com/DiegoCaraballo>

REDES SOCIALES MI DIARIO DE PYTHON

- FACEBOOK
- TWITTER
- INSTAGRAM
- YOUTUBE



OPINIÓN DEL LECTOR

Jose Luis desde México nos envió su opinión sobre **GNU/LINUX** y su breve historia desde que comenzó a utilizar **GNU/LINUX**.

Gnu/Linux es un sistema operativo muy estable, sobre todo me gusta lo personalizable que es ya que existen una gran variedad de distribuciones para todos los gustos y necesidades, además de que se acopla al equipo del usuario, por que no hace falta tener un equipo costoso para poder tener un sistema operativo moderno, con soporte y bonito.

La Historia de Jose Luis en GNU/LINUX.

Por el año 2001-2003 un vecino me invitó a su casa y vi que tenía un sistema operativo diferente, con ventanas transparentes y con una consola de comandos. Le pregunté acerca del sistema operativo y me dijo que era Linux (Mandrake) y le dije que si me ayudaba a instalaelo en mi pc, a lo cual accedió.

En revistas ya había visto al S.O. del pingüino en Cd, pero al ser gratis me supuse que debía ser un sistema operativo de baja calidad y una copia barata de mi Windows 98. Y además que no tenía idea de como instalarlo.

Así que me vecino me lo instaló y como tenía Winmodem no podía conectarme a internet. Sin internet no le puse mucho interés y menos al ver que TODO era comandos: insertar pendrive, cd, descomprimir archivos, etc. Así que lo dejé en el olvido.

Volví a Linux en el año 2020 debido a la pandemia, con más tiempo que nunca me acordé del pingüino y su sistema, y decidí investigar sobre qué distribución me vendría bien para principiante y opté por Linux Mint.

Poco a poco voy aprendiendo más sobre gnu/Linux, y desde aquel año que conocí al pingüino, ha cambiado mucho Linux, me gusta la terminal de comandos, pero se agradece que también se pueda hacer mediante interfaz gráfica.

Canales de Telegram:

Canal SoloLinux – Canal SoloWordpress

Espero que este artículo te sea de utilidad, puedes ayudarnos a mantener el servidor con una donación ([paypal](#)), o también colaborar con el simple gesto de compartir nuestros artículos en tu sitio web, blog, foro o redes sociales. Instalar Google Cloud SDK en Ubuntu 20.04.

Chat de SoloLinux en Telegram

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

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

**AYUDANOS A SEGUIR
CRECIENDO**

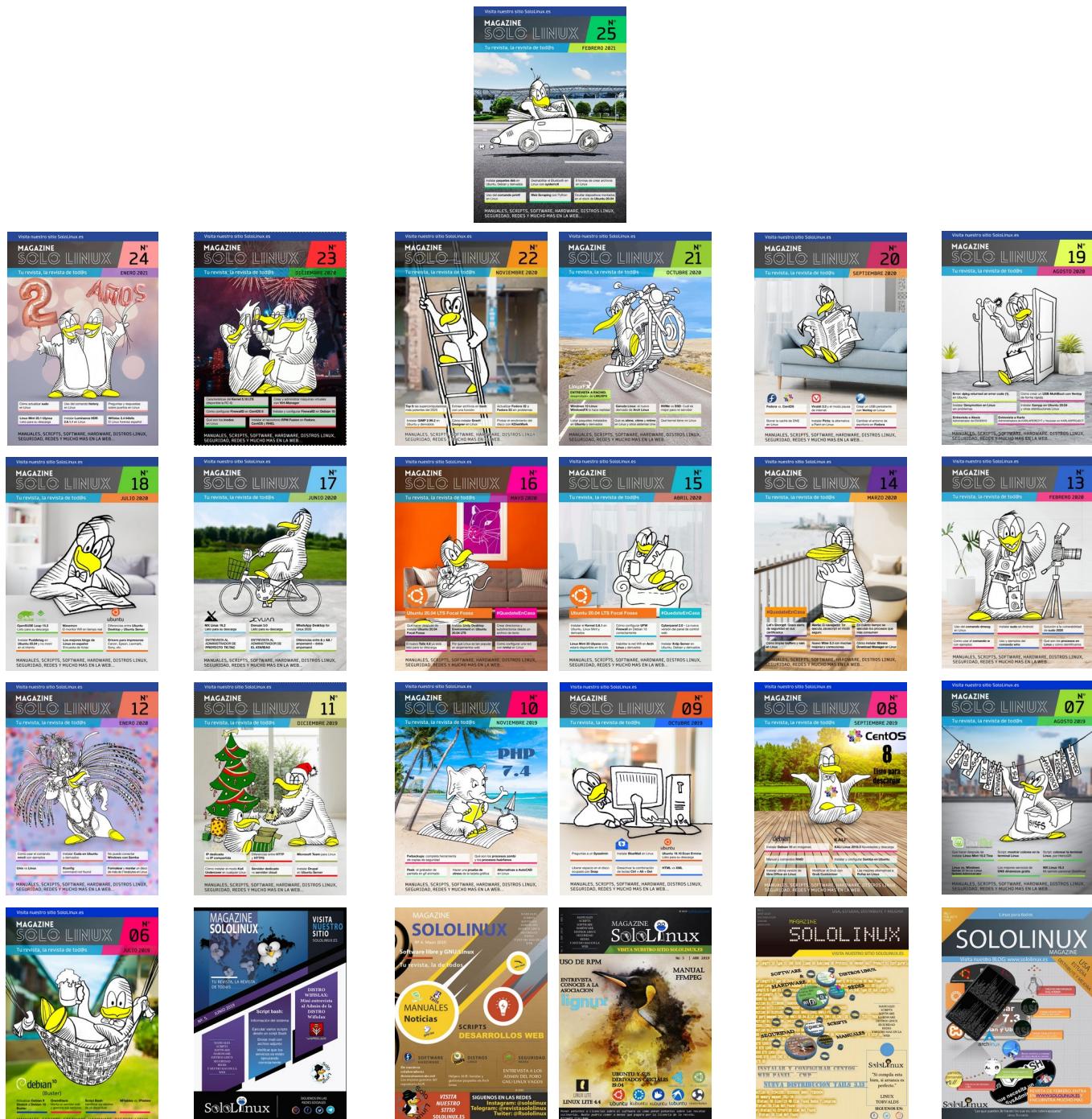


www.sololinux.es

MAGAZINE SOLOLINUX, REVISTA QUE NACIÓ UN 2 DE MARZO DEL AÑO 2019 SIENDO UNA COPIA EXACTA MENSUAL DEL SITIO WEB [WWW.SOLOLINUX.ES](http://www.sololinux.es)

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

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





www.sololinux.es