

Visita nuestro sitio SoloLinux.es

MAGAZINE SOLO LINUX

Nº
31

Tu revista, la revista de tod@s

AGOSTO 2021



Debian 11 Bullseye - Listo para su
descarga

Cómo Instalar Debian 11 de forma
sencilla

El Linux Kernel 5.14 ha sido liberado

Cómo instalar KALI Linux en
VirtualBox 6.1

Modificar la Swap con bash script

Diferencias entre apt y apt-get en
Ubuntu y derivados

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

Dirección y maquetación

Adrián Almenar
e-mail: adrian@sololinux.es

Redacción

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

Diseño Portada

Karina Fernández
Instagram: [@karyfernandez.design](https://www.instagram.com/karyfernandez.design)

Publicidad

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

CON SOLOLINUX
MULTIPLICARAS
TUS CLIENTES

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

Contacto

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

Aviso – Nota del autor:

Los sitios SoloLinux.es, RevistaLinux.es y la [Revista SoloLinux](http://RevistaSoloLinux.es), no mantienen ningún tipo de relación contractual con los propietarios de otros blogs, autores de opiniones publicadas o anunciantes de la revista.



La revista SOLOLINUX esta realizada con **Libre Office Impress 7.2.0.4**



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

Bienvenido a la Revista SOLOLINUX

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

Os presentamos el número 31 de la **Revista SoloLinux**.

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

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

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

Gracias a TOD@S

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

Equipo SOLOLINUX

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

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

PayPal

Donar a Revistalinux

AYUDANOS A SEGUIR
CRECIENDO

PayPal

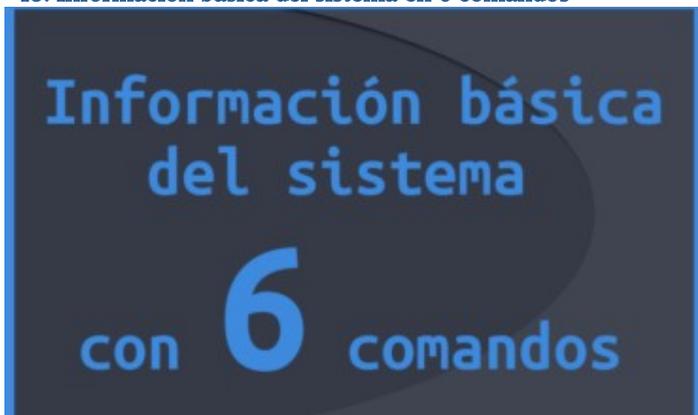
Donar a SoloLinux



designed by  freepik

MANUALES

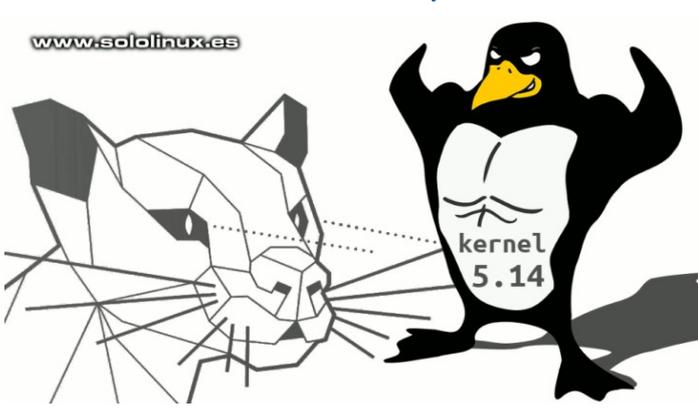
- 07. Instalar un servidor shortener con YOURLS en Ubuntu 20.04
- 11. Conectar a una máquina remota con Chrome Remote Desktop
- 13. Información básica del sistema en 6 comandos



- 15. Error: Sudo Command Not Found en Debian 10
- 22. Qué son los niveles de ejecución en Linux
- 32. Como instalar Debian 11 de forma sencilla



- 36. Probar el rendimiento de un servidor con JMeter
- 40. Error en Debian 11 al ejecutar sudo
- 44. Diferencias entre apt upgrade y apt full-upgrade
- 48. Diferencias entre apt y apt-get en Ubuntu y derivados
- 54. Modificar los kernel que se guardan en Ubuntu
- 57. Instalar el Kernel 5.14 en Ubuntu y derivados



SOFTWARE

- 10. Instalar Microsoft Teams en Rocky Linux 8.4



- 24. Instalar Netdata Monitoring en Ubuntu 20.04
- 28. Instalar Apache JMeter en Linux
- 41. Novedad: LibreOffice 7.2 listo para su descarga



Canales de Telegram:
Canal SoloLinux
Canal SoloWordpress

Chat de SoloLinux en Telegram

PayPal

Donar a SoloLinux

PayPal

Donar a Revistalinux



designed by freepik

DISTROS LINUX

- 17. Poner KALI Linux 2021 en español
- 26. Como instalar KALI Linux en VirtualBox 6.1
- 30. Debian 11 Bullseye – Listo para su descarga



SEGURIDAD

- 20. Instalar DVWA en KALI Linux 2021
- 50. Como instalar VirusTotal Uploader en Linux

SCRIPTS

- 45. Modificar la swap con bash script

www.sololinux.es

NOTICIAS

- 52. 30 años de Kernel Linux entre nosotros
- 56. El Linux Kernel 5.14 ha sido liberado



VANT

SOMOS LINUXEROS

www.vantpc.es

intel
CORE
i5

intel
iRISxe
GRAPHICS



HDMI
HIGH-DEFINITION MULTIMEDIA INTERFACE

**¿PREPARADO
PARA VOLVER
A CLASE?**



@vantpc



vant.pc



vantpc_es



t.me/vantpc





**INSTITUTO
LINUX**



FABIÁN AMPALIO
CEO CLA IL



DIEGO NOBILE
Instructor



GABRIEL CANEPA
Tutor



ALEJANDRO DRABENCHE
Tutor



JOAQUÍN AMPALIO
Staff



FRANCISCO SANDALINAS
Staff



LINUX SYSTEM ADMINISTRATOR + KALI LINUX

- 6 MESES DE DURACIÓN -
CLASES EN VIVO O ACCESO INMEDIATO



INICIO 11 DE AGOSTO DE 2021

**1 PAGO DE
€ 197
Ó 3 CUOTAS DE
€ 97**



+54 9 11 6969 9993



espania@carreralinux.com.ar



www.carreralinux.es

Instalar un servidor shortener con YOURLS en Ubuntu 20.04

YOURLS que en realidad significa **Your Own URL Shortener** (Su propio acortador de URL). Es una herramienta compuesta de varios scripts PHP, que nos permite tener nuestro propio **servicio de acortamiento de URL**. Para que me entiendas mejor, lo mismo que **TinyURL**, **Bitly** y otros tantos servicios existentes, pero bajo tu control.

Ejecutar tu propio servicio acortador de URL es útil si haces uso de ellos continuamente. Podrás agregar tu propio dominio a los acortadores, además de no depender de nadie, tus datos los manejas tú solo tú. En este artículo veremos como montar un **servidor** de estas características, ayudándonos del increíble paquete **YOURLS**.

<https://www.sololinux.es> >>> <https://tinyurl.com/2hm4uuvj>

Instalar un servidor shortener con YOURLS en Ubuntu 20.04

Como siempre... comenzamos actualizando nuestro sistema.

```

sudo apt update
sudo apt full-upgrade

```

```

root@Sololinux-demo:~# sudo apt full-upgrade
Reading package lists... Done
Building dependency tree... Done
Calculating upgrade... Done
The following packages will be upgraded:
  apt apt-utils base-files bash bsdtails busybox-initramfs ca-certificates
  cifs-utils curl dirmngr distro-info-data gcc-10-base gnupg gnupg-l10n
  gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-server gpgconf gpgsm gpgv
  initramfs-tools initramfs-tools-bin initramfs-tools-core ldap-utils less
  libapt-pkg6.0 libasound2 libasound2-data libblkid1 libbrotli1 libc-bin libc6
  libcryptsetup12 libcurl4 libfdisk1 libfreetype6 libgcc-s1 libgcc1
  libglb2.0-0 libgnutls30 libgomp1 libgssapi-krb5-2 libhogweed5 libk5crypto3
  libkrb5-3 libkrb5support0 libldap-2.4-2 libldap-common liblz4-1 liblzma5
  libmount1 libmysqldbclient21 libnettle7 libp11-kit0 libpam-modules
  libpam-modules-bin libpam-runtime libpam0g libpci3 libperl5.30 libprocps8
  libpython2.7-minimal libpython2.7-stdlib libpython3.8 libpython3.8-minimal
  libpython3.8-stdlib libseccomp2 libsmartcols1 libsnmp-base libsnmp35
  libsquid3-0 libssh-4 libssl1.1 libstdc++6 libsystemd0 libudev1 libuid1
  libwbclient0 libxml2 libzstd1 linux-base locales lsof lz4 mount
  openssh-client openssh-server openssh-sftp-server openssl perl perl-base
  perl-modules-5.30 procps python2.7 python2.7-minimal python3.8
  python3.8-minimal rsyslog screen snmp sudo systemd systemd-sysv
  systemd-timesyncd tar tzdata ubuntu-keyring udev util-linux uuid-runtime

```



Ahora sigue todos los pasos indicados, para instalar el **paquete LEMP** (servidor).

```

sudo apt-get install nginx
sudo systemctl start nginx
sudo systemctl enable nginx

```

```

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

```

Llega el turno de instalar MariaDB.

```

sudo apt-get install nginx mariadb-server
sudo systemctl start mariadb
sudo systemctl enable mariadb

```

Aseguramos la instalación.

```
sudo mysql_secure_installation
```

Es necesario crear una base de datos con usuario y password. Introduce tus datos reales.

```
sudo mysql -u root -p
```

Los datos son:

- **MOMBREDB** – Nombre de la nueva base de datos.
- **MIUSUARIO** – Usuario de la base de datos.
- **PASSWORD** – Tu contraseña elegida.

Ejecuta lo siguiente con tus datos (línea por línea).

```
CREATE DATABASE MOMBREDB DEFAULT CHARACTER SET UTF8 COLLATE utf8_unicode_ci;
CREATE USER 'MIUSUARIO'@'localhost' IDENTIFIED BY 'PASSWORD';
GRANT ALL PRIVILEGES ON MOMBREDB.* TO 'MIUSUARIO'@'localhost' IDENTIFIED BY 'PASSWORD' WITH GRANT OPTION;
FLUSH PRIVILEGES;
EXIT;
```

Observa la imagen de ejemplo real.

```
root@SoloLinux-demo:~# sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 57
Server version: 10.3.30-MariaDB-0ubuntu0.20.04.1 Ubuntu 20.04
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> CREATE DATABASE MOMBREDB DEFAULT CHARACTER SET UTF8 COLLATE utf8_unicode_ci;
Query OK, 1 row affected (0.000 sec)
MariaDB [(none)]> CREATE USER 'MIUSUARIO'@'localhost' IDENTIFIED BY 'PASSWORD';
Query OK, 0 rows affected (0.000 sec)
MariaDB [(none)]> GRANT ALL PRIVILEGES ON MOMBREDB.* TO 'MIUSUARIO'@'localhost' IDENTIFIED BY 'PASSWORD' WITH GRANT OPTION;
Query OK, 0 rows affected (0.000 sec)
MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)
MariaDB [(none)]> EXIT;
Bye
root@SoloLinux-demo:~#
```

Instalamos PHP.

```
sudo apt-get install php-fpm php-xml php-pear php-cli php-zip php-json
php-mysql php-gd php-mbstring php-curl php-bcmath -y
```

Es necesario reiniciar Nginx.

```
sudo systemctl restart nginx
```

Llego el momento de descargar YOURLS, pero para ello necesitamos **instalar GIT** y acceder a **/var/www/html**.

```
sudo apt install git
```

```
cd /var/www/html
```

```
sudo git clone https://github.com/YOURLS/YOURLS.git
```

```
root@SoloLinux-demo:/var/www/html# sudo git clone
https://github.com/YOURLS/YOURLS.git
Cloning into 'YOURLS'...
remote: Enumerating objects: 9686, done.
remote: Counting objects: 100% (91/91), done.
remote: Compressing objects: 100% (74/74), done.
remote: Total 9686 (delta 39), reused 40 (delta 16), pack-reused 9595
Receiving objects: 100% (9686/9686), 14.13 MiB | 18.89 MiB/s, done.
Resolving deltas: 100% (5847/5847), done.
root@SoloLinux-demo:/var/www/html#
```

Borramos dos archivos innecesarios.

```
sudo rm index.html
sudo rm index.nginx-debian.html
```

Accedemos a la carpeta user (recuerda que ya estamos en **/var/www/html**).

```
cd YOURLS/user/
```

Creamos el archivo de configuración (aprovechamos el que viene de ejemplo).

```
cp config-sample.php config.php
```

Ejemplo del proceso.

```
root@SoloLinux-demo:/var/www/html# cd YOURLS/user/
root@SoloLinux-demo:/var/www/html/YOURLS/user# cp config-sample.php
config.php
root@SoloLinux-demo:/var/www/html/YOURLS/user#
```

Accede al archivo y lo modificas, con los datos de la base de datos que usaste anteriormente.

```
sudo nano config.php
```

Las líneas a editar son:

```
/** MySQL database username */
define( 'YOURLS_DB_USER', 'your db user name' ); <<<--- Usuario
/** MySQL database password */
define( 'YOURLS_DB_PASS', 'your db password' ); <<<<--- Password
/** The name of the database for YOURLS
** Use lower case letters [a-z], digits [0-9] and underscores [_]
only */
define( 'YOURLS_DB_NAME', 'yours' ); <<<--- Nombre de la base de
datos
/** YOURLS installation URL
** All lowercase, no trailing slash at the end.
** If you define it to "http://sho.rt", don't use "http://www.sho.rt"
in your >
** To use an IDN domain (eg http://héhé.com), write its ascii form
here (eg ht
define( 'YOURLS_SITE', 'http://your-own-domain-here.com' ); <<<<---
Url de acceso
** A random secret hash used to encrypt cookies. You don't have to
remember it>
** Hint: copy from http://yours.org/cookie */
define( 'YOURLS_COOKIEKEY', 'modify this text with something
random' ); <<<--- Inserta una cadena de caracteres aleatoria
/** Username(s) and password(s) allowed to access the site. Passwords
either in>
** YOURLS will auto encrypt plain text passwords in this file
** Read http://yours.org/userpassword for more information */
$yours_user_passwords = [
    'username' => 'password', <<<--- password
    // 'username2' => 'password2',
    // You can have one or more 'login'=>'password' lines
];
```

```
#!/PHP
/* This is a sample config file.
* Edit this file with your own settings and save it as "config.php"
*
* IMPORTANT: edit and save this file as plain ASCII text, using a text editor, for instance TextEdit on Mac OS or
* Notepad on Windows. Make sure there is no character before the opening <?php at the beginning of this file.
*/
/**
** MySQL settings - You can get this info from your web host
*/
/** MySQL database username */
define( 'YOURLS_DB_USER', 'your db user name' );
/** MySQL database password */
define( 'YOURLS_DB_PASS', 'your db password' );
/** The name of the database for YOURLS
** Use lower case letters [a-z], digits [0-9] and underscores [_] only */
define( 'YOURLS_DB_NAME', 'yours' );
/** MySQL hostname.
** If using a non standard port, specify it like 'hostname:port', e.g. 'localhost:9999' or '127.0.0.1:666' */
define( 'YOURLS_DB_HOST', 'localhost' );
/** MySQL tables prefix
** YOURLS will create tables using this prefix (eg 'yours_url', 'yours_options', ...)
** Use lower case letters [a-z], digits [0-9] and underscores [_] only */
define( 'YOURLS_DB_PREFIX', 'yours_' );
*/
```

Una vez hayas concluido la configuración del archivo, lo guardas y cierras el editor. Es necesario conceder los permisos correspondientes.

```
cd
chown -R www-data:www-data /var/www/html
chmod -R 775 /var/www/html
```

```
root@SoloLinux-demo:~# cd
root@SoloLinux-demo:~# chown -R www-data:www-data /var/www/html
root@SoloLinux-demo:~# chmod -R 775 /var/www/html
root@SoloLinux-demo:~#
```

Si tienes el firewall habilitado, abres los puertos necesarios.

```
ufw allow http
ufw allow https
```

Configuramos NGINX.

```
nano /etc/nginx/sites-available/yours.conf
```

Copia y pega la configuración del archivo, sin olvidarte de modificar la url.

```
server {
    listen 80;
    listen [::]:80;
    server_name midominio.com;
    root /var/www/html;
    location / {
        try_files $uri $uri/ /yourls-loader.php$is_args$args;
    }
    location ~ /\.php$ {
        include fastcgi.conf;
        fastcgi_index index.php;
        fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
    }
}
```

Guarda el archivo y cierra el editor.

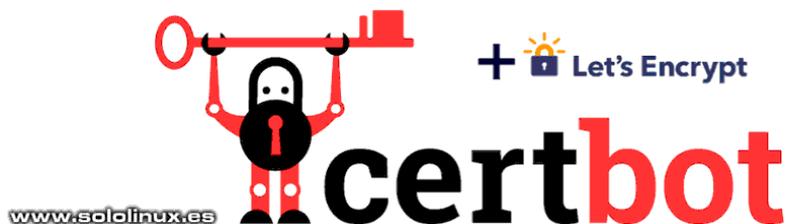
Habilitamos el archivo.

```
ln -s /etc/nginx/sites-available/yourls.conf /etc/nginx/sites-enabled/
```

Para concluir, es necesario reiniciar NGINX.

```
systemctl restart nginx
```

En realidad nuestro servidor ya está listo para producción, pero mucho mejor si aseguramos con SSL instalando **Let's Encrypt** con **Certbot**.



Instalamos la herramienta y las bibliotecas necesarias.

```
sudo apt-get install certbot python3-certbot-nginx
```

Puedes agregar SSL fácilmente a tu dominio, ejecutando este comando.

```
sudo certbot --nginx -d midominio.com
```

Para concluir el proceso, también debes hacer unas pequeñas modificaciones en el archivo de configuración.

```
sudo nano /var/www/html/YOURLS/user/config.php
```

En la línea donde insertamos la url del sitio.

```
/** YOURLS installation URL
 * All lowercase, no trailing slash at the end.
 * If you define it to "http://sho.rt", don't use "http://www.sho.rt" in your >
 * To use an IDN domain (eg http://héhé.com), write its ascii form here (eg ht>
 define( 'YOURLS_SITE', 'http://your-own-domain-here.com' );
```

Debes modificar http por https.

```
/** YOURLS installation URL
 * All lowercase, no trailing slash at the end.
 * If you define it to "http://sho.rt", don't use "http://www.sho.rt" in your >
 * To use an IDN domain (eg http://héhé.com), write its ascii form here (eg ht>
 define( 'YOURLS_SITE', 'https://your-own-domain-here.com' );
```

Después de la línea anterior, añade esta otra.

```
define( 'YOURLS_ADMIN_SSL', true);
```

```
/**
 * Site options
 */
/** YOURLS installation URL
 * All lowercase, no trailing slash at the end.
 * If you define it to "http://sho.rt", don't use "http://www.sho.rt" in your browser (and vice-versa)
 * To use an IDN domain (eg http://héhé.com), write its ascii form here (eg http://xn--hh-bjab.com) */
define( 'YOURLS_SITE', 'https://your-own-domain-here.com' );
define( 'YOURLS_ADMIN_SSL', true);
```

Guarda el archivo y cierra el editor, solo falta reiniciar el **servidor**.

```
sudo reboot
```

Ya tenemos listo nuestro servidor, ahora solo falta instalar y configurar la herramienta, según tus necesidades. Desde tu navegador web favorito, accede a la siguiente url.

<https://midominio.com/admin/install.php>

Sigue los pasos que te indica la **herramienta YOURLS**. Usa tu nombre de usuario y contraseña de administrador para iniciar sesión en...

<https://midominio.com/admin/index.php>

Listo, ya tenemos nuestro servidor acortador de url listo para producción.

SoloWordPress

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



Instalar Microsoft Teams en Rocky Linux 8.4



Microsoft Teams es una herramienta diseñada por Microsoft, para facilitar las comunicaciones entre usuarios, permitiendo chats, el intercambio de archivos y enlaces, planificación de reuniones, videoconferencias, grabar audios, tareas pendientes y varias cosas más.

Microsoft afirma, que su base principal se centra en ofrecer una comunicación rápida y moderna. También nos dicen que no tiene ningún coste, ya que viene incluido de forma predeterminada en **Windows 10**, **Windows 11** y en **Microsoft 365**. No te fastidia, los **windowseros** ya se tuvieron que rasgar el bolsillo antes, jaja.

Bueno basta de bromas. En el afán de colaboración por parte de Microsoft con la comunidad **opensource**, sí que se ofrece una versión para **Linux** totalmente gratuita, que **no libre**. Así que como soy de culo inquieto y siempre estoy cacharreando, he decidido probar la instalación de esta herramienta en una distribución como **Rocky Linux**. Como el resultado fue satisfactorio, decido compartir la operación por si algún atrevido quiere usar **Microsoft Teams**.



Instalar Microsoft Teams en Rocky Linux 8.4

Antes de comenzar, debo haceros saber que existen varias formas de instalar Microsoft Teams, pero como aquí nos gusta lo tradicional, agregaremos los **repositorios oficiales** proporcionados por Microsoft e instalaremos la aplicación.

Tu mismo puedes verificar que operamos sobre un Rocky Linux 8.4.

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

Creamos el archivo del nuevo repositorio.

```
sudo nano /etc/yum.repos.d/ms-teams.repo
```

Copia y pega lo siguiente.

```
[Teams]
name=teams
baseurl=https://packages.microsoft.com/yumrepos/ms-teams
enabled=1
gpgcheck=1
gpgkey=https://packages.microsoft.com/keys/microsoft.asc
```

Guarda el archivo y cierra el editor. Ahora actualizamos.

```
sudo dnf update
```

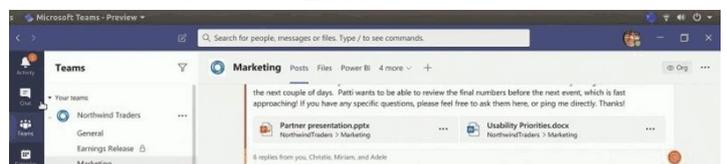
```
[root@SoloLinux-demo ~]# sudo dnf update
teams                               254 kB/s | 42 kB    00:00
Dependencias resueltas.
Nada por hacer.
¡Listo!
[root@SoloLinux-demo ~]#
```

Llego el momento de instalar Microsoft Teams.

```
sudo dnf install teams
```

```
[root@SoloLinux-demo ~]# sudo dnf install teams
Última comprobación de caducidad de metadatos hecha hace 0:04:09, el mar 03 ago 2021
18:34:15 CEST.
Dependencias resueltas.
=====
Paquete      Arq.  Versión      Repositorio  Tam.
=====
Instalando:
teams        x86_64  1.4.00.13653-1  Teams        109 M
Instalando dependencias:
adwaita-cursor-theme  noarch  3.28.0-2.el8  appstream    646 k
adwaita-icon-theme    noarch  3.28.0-2.el8  appstream    11 M
-----etc, etc-----
libsoup        x86_64  2.62.3-2.el8  baseos       423 k
libthai        x86_64  0.1.27-2.el8  appstream    202 k
libtiff        x86_64  4.0.9-18.el8  appstream    187 k
nspr           x86_64  4.25.0-2.el8_2  appstream    141 k
nss            x86_64  3.53.1-17.el8_3  appstream    722 k
nss-softokn    x86_64  3.53.1-17.el8_3  appstream    483 k
nss-softokn-freebl  x86_64  3.53.1-17.el8_3  appstream    375 k
nss-sysnss     x86_64  3.53.1-17.el8_3  appstream    71 k
nss-util       x86_64  3.53.1-17.el8_3  appstream    135 k
pango          x86_64  1.42.4-6.el8  appstream    296 k
pixman         x86_64  0.38.4-1.el8  appstream    256 k
rest           x86_64  0.8.1-2.el8   appstream    69 k
Instalando dependencias débiles:
abattis-cantarell-fonts  noarch  0.0.25-6.el8  appstream    154 k
dconf          x86_64  0.28.0-4.el8  appstream    107 k
Resumen de la transacción
=====
Instalar 76 Paquetes
Tamaño total de la descarga: 137 M
Tamaño instalado: 362 M
¿Está de acuerdo [s/N]?:
```

El tamaño de la instalación, dependerá en gran medida del estado de tu sistema. Pero una vez concluya, abre tu **menú de aplicaciones** y allí encontrarás el icono de **Microsoft Teams**, haces clic en él para ejecutar el software en tu sistema Linux.



Conectar a una máquina remota con Chrome Remote Desktop

Chrome Remote Desktop es una excelente, pero no muy conocida herramienta, desarrollada por el gigante Google que nos facilita la tarea de conectarnos a un escritorio remoto de forma rápida y segura. Utiliza un protocolo propietario desarrollado por ellos mismos, el cual se conoce como **Chromoting**.

Este método que hoy te propongo, es válido para cualquier sistema desktop basado en Debian. Pero para complicar un poco más la cosa, además que pienso que es de máxima utilidad, nosotros haremos este trabajo sobre un **Ubuntu Server LTS 20.04** sin **desktop**, es evidente que lo tendremos que instalar. Ya verás que fácil.

Conectar a una máquina remota con Chrome Remote Desktop

Para realizar este proceso debes ser **root** y, comenzamos a operar en la **máquina remota** (en nuestro caso un Ubuntu Server 20.04).

```
sudo su
```

Continuamos actualizando nuestro sistema.

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

```
root@SoloLinux-demo:~# apt update
Get:1 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [970 kB]
Get:5 http://archive.canonical.com/ubuntu focal InRelease [12.1 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal/main Translation-en [506 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal/restricted amd64 Packages [22.0 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal/restricted Translation-en [6212 B]
Get:9 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
Get:11 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1131 kB]
Get:12 http://archive.ubuntu.com/ubuntu focal-updates/main Translation-en [247 kB]
Get:13 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [367 kB]
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [52.9 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [843 kB]
Get:16 http://archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [177 kB]
Get:17 http://archive.canonical.com/ubuntu focal/partner amd64 Packages [856 B]
Get:18 http://archive.canonical.com/ubuntu focal/partner Translation-en [384 B]
Get:19 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [792 kB]
Get:20 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [156 kB]
Get:21 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [330 kB]
Get:22 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [47.6 kB]
Get:23 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [635 kB]
Get:24 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [98.4 kB]
Get:25 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [22.3 kB]
Get:26 http://security.ubuntu.com/ubuntu focal-security/multiverse Translation-en [5072 B]
Fetched 20.7 MB in 3s (6553 kB/s)
Reading package lists... Done
Building dependency tree... Done
116 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@SoloLinux-demo:~# apt full-upgrade
```

Instalar Chrome Remote Desktop en Ubuntu 20.04

Ahora descargamos el paquete Chrome Remote Desktop y, lo instalamos.

```
wget https://dl.google.com/linux/direct/chrome-remote-desktop_current_amd64.deb
dpkg --install chrome-remote-desktop_current_amd64.deb
apt install -y --fix-broken
```

```
root@SoloLinux-demo:~# wget https://dl.google.com/linux/direct/chrome-remote-desktop_current_amd64.deb
--2021-08-04 08:41:16-- https://dl.google.com/linux/direct/chrome-remote-desktop_current_amd64.deb
Resolving dl.google.com (dl.google.com)... 2a00:1450:4013:c14::5d, 2a00:1450:4013:c14::5b, 2a00:1450:4013:c14::be, ...
Connecting to dl.google.com (dl.google.com)|2a00:1450:4013:c14::5d|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 37417740 (36M) [application/x-debian-package]
-----etc, etc-----n
```

Instalar el entorno de escritorio XFCE

Como en nuestro caso queremos operar con un servidor remoto, instalamos un escritorio ligero y funcional, **XFCE**.

```
sudo DEBIAN_FRONTEND=noninteractive apt install -y xfce4 desktop-base
```

Configuramos el inicio predeterminado con **XFCE**.

```
sudo bash -c 'echo "exec /etc/X11/Xsession /usr/bin/xfce4-session" > /etc/chrome-remote-desktop-session'
```

Light Locker, no funciona bien con Chrome Remote Desktop, la pantalla se queda en blanco y no se puede desbloquear. Mejor instalamos **XScreenSaver**.

```
apt install -y xscreensaver
```

Opcional – La instalación que hemos realizado de XFCE es mínima, si la quieres más completa ejecuta el siguiente comando.

```
apt install -y task-xfce-desktop
```

Es importante que no te olvides agregar el **USER**, al grupo de **Google Remote Desktop** y que después reinicies el sistema.

```
sudo usermod -a -G chrome-remote-desktop $USER
```

```
reboot
```

Configurar el escritorio remoto en Chrome

Bueno, ya lo tienes instalado. Llego la hora de conectarte, pero no olvides que debes haber iniciado sesión en tu cuenta de Google, si no es así, no funciona. Abre la siguiente url en tu máquina local.

<https://remotedesktop.google.com/headless>

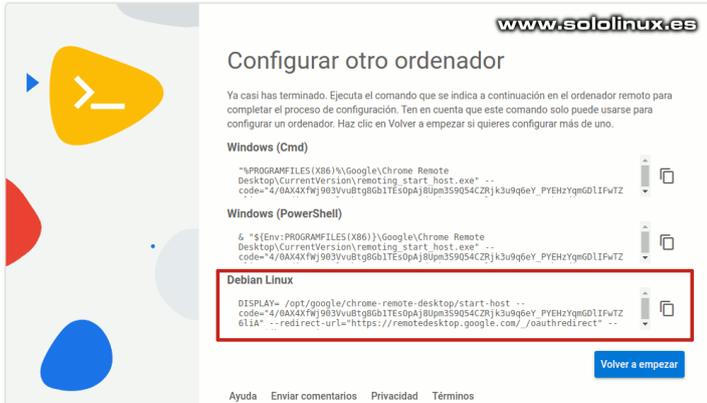
Verás la siguiente pantalla o similar. Pulsa en empezar.



Aparece el script que debes instalar en la máquina remota, como nosotros ya lo tenemos, pulsa en siguiente.



En la nueva pantalla, debes autorizar al **Escritorio Remoto de Chrome** a configurar un nuevo ordenador. Al pulsar autorizar se abre otra ventana, en la cual podrás obtener un código que debes insertar en la línea de comandos de la máquina remota.



Ejemplo de código que debes copiar y pegar.

```
DISPLAY= /opt/google/chrome-remote-desktop/start-host --
code="4/0AX4XfWj903vubt98b1TE50pA]8Upm359054CZr-
POSF1M2JiqAd0tcPQ"
--redirect-url="https://remotedesktop.google.com/_/oauthredirect" --
name=$(hostname)
```

Al ejecutar la cadena nos pide insertar un pin, no lo olvides porque lo necesitaras para conectar remotamente.

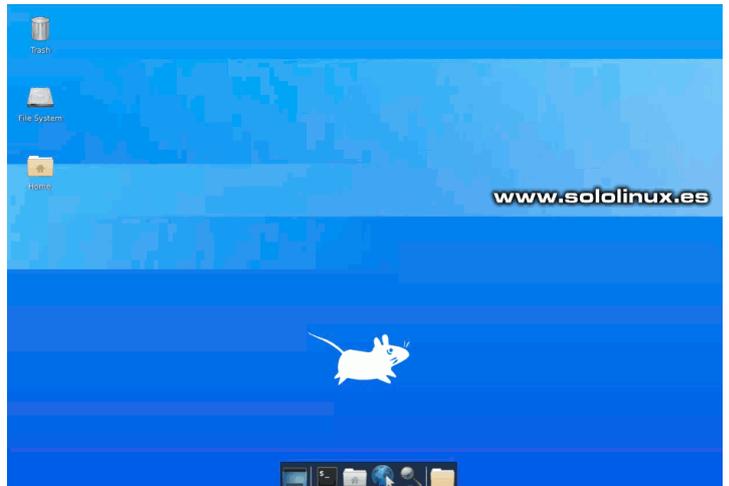
```
Enter a PIN of at least six digits:
Enter the same PIN again:
[0318/171746.983120:ERROR:cert_verify_proc_builtin.cc(559)] No
net_fetcher for performing AIA chasing.
[0318/171747.062520:INFO:host_stopper.cc(38)] Stopping existing host:
610ad6c6-9078-424d-8938-b099d965c5ac. This may take a few seconds.
[0318/171747.075063:ERROR:cert_verify_proc_builtin.cc(559)] No
net_fetcher for performing AIA chasing.
[0318/171747.110841:ERROR:service_client.cc(132)] Received error code:
7, message: The caller does not have
permission[0318/171747.820448:INFO:daemon_controller_delegate_linux.cc
(99)] Killing process 18032
[0318/171748.612246:INFO:daemon_controller_delegate_linux.cc(99)]
2021-03-18 17:17:48,603:INFO:User 'edxd' is already a member of
'chrome-remote-desktop'.
[0318/171750.087701:INFO:daemon_controller_delegate_linux.cc(99)]
[0318/171748.719199:INFO:remoting_user_session.cc(754)] Daemon process
started in the background, logging to
'/tmp/chrome_remote_desktop_20210318_171748_coqBUz'
WARNING: This system uses GDM. Some GDM versions have a bug that
prevents local login while Chrome Remote Desktop is running. If you
run into this issue, you can stop Chrome Remote Desktop by visiting
https://remotedesktop.google.com/access on another machine and
clicking the delete icon next to this machine. It may take up to five
minutes for the Chrome Remote Desktop to exit on this machine and for
local login to start working again.
Using host id: 1198f230-a125-4836-84ee-9af63fd22b8d
Launching X server and X session.
Starting Xvfb on display :20
X server is active.
Launching X session: ['/bin/sh', '/etc/chrome-remote-desktop-session']
Launching host process
['/opt/google/chrome-remote-desktop/chrome-remote-desktop-host', '--
host-config=', '--audio-pipe-name=/home/edxd/.config/chrome-remote-
desktop/pulseaudio#d68353f7c2/fifo_output', '--server-supports-exact-
resize', '--ssh-auth-sockname=/tmp/chromoting.edxd.ssh_auth_sock', '--
signal-parent']
Host ready to receive connections.
Log file: /tmp/chrome_remote_desktop_20210318_171748_coqBUz
```

Conectar a la máquina remota

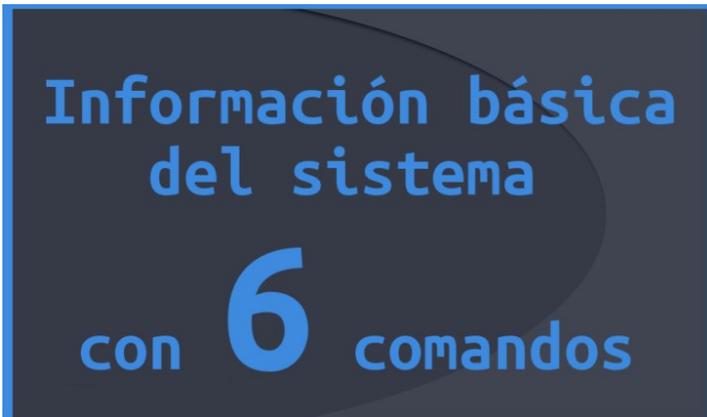
Si todo salió bien, tan solo tienes que acceder a la siguiente url, seleccionar la máquina remota e introducir el código pin insertado.

<https://remotedesktop.google.com/>

Conectadoss!!!, pero si tienes problemas puedes revisar su [ayuda oficial](#).



Información básica del sistema en 6 comandos



Si bien es cierto que existen **herramientas GUI**, que ofrecen la capacidad de ofrecernos información completa del sistema. Yo siempre recomiendo extraer, e imprimir en consola estos datos mediante **comandos**.

Debes saber, que todas esas herramientas GUI ofertadas, no dejan de ser meras interfaces gráficas de comandos Linux que interactúan con el sistema para imprimir los valores solicitados. En este sencillo artículo de hoy, veremos los 6 comandos más utilizados para extraer información básica de un sistema basado en **Linux/GNU**.

#>6 comandos Linux

Información básica del sistema en 6 comandos

Vemos los principales comandos, con ejemplo de uso incluido. Como norma general, ninguno de ellos requiere de «**sudo**».

Comando uptime

Este comando nos indica una información limitada, pero extremadamente útil.

```
uptime
```

Primero nos indica la hora actual del sistema, después el tiempo que lleva encendido (en nuestro ejemplo dos días), ahora indica el tiempo que llevan los usuarios conectados, el número de usuarios y finaliza ofreciendo la carga del sistema (**load average**).

```
root@SoloLinux-demo:~# uptime
09:30:32 up 2 days, 11 min, 1 user, load average: 0.01, 0.03, 0.05
root@SoloLinux-demo:~#
```

El **comando uptime** también ofrece interesantes opciones, por ejemplo si queremos saber la fecha y hora que inicio el sistema, ejecutamos...

```
Uptime -s
```

```
root@SoloLinux-demo:~# uptime -s
2021-08-04 09:18:34
root@SoloLinux-demo:~#
```

Podemos ver todas las opciones en la ayuda del comando.

```
uptime --h
```

```
sergio@sololinux:~$ uptime -h
www.sololinux.es

Uso:
uptime [opciones]

Opciones:
-p, --pretty    mostrar tiempo de ejecución en formato legible
-h, --help     mostrar esta ayuda y salir
-s, --since    sistema levantado desde
-V, --version  mostrar la información de la versión y salir

Para más detalles vea uptime(1).
sergio@sololinux:~$
```

Comando uname

Al utilizar el **comando uname** sin ningún tipo de argumento, nos indica el tipo de sistema que tenemos. En nuestro caso Linux.

```
uname
```

```
root@SoloLinux-demo:~# uname
Linux
root@SoloLinux-demo:~#
```

Al agregar «**-a**», la información se amplía de forma considerable.

```
uname -a
```

Veremos los siguientes datos.

- **Kernel del sistema**
- **Nombre del host**
- **Fecha de lanzamiento del kernel**
- **Versión del núcleo**
- **Arquitectura del hardware**
- **Tipo de sistema operativo**

```
root@SoloLinux-demo:~# uname -a
Linux SoloLinux-demo 5.4.0 #1 SMP Fri Nov 20 21:47:55 MSK 2020 x86_64
x86_64 x86_64 GNU/Linux
root@SoloLinux-demo:~#
```

Es interesante revisar la ayuda de este comando.

```
uname --help
```

```
root@SoloLinux-demo:~$ uname --help
Modo de empleo: uname [OPCIÓN]...
Muestra cierta información del sistema. Sin ninguna OPCIÓN, igual que -s.
-a, --all muestra toda la información, en el siguiente orden,
excepto que se omite -p y -i si son desconocidos:
-s, --kernel-name muestra el nombre del núcleo
-n, --nodename muestra el nombre de 'host' del nodo de red
-r, --kernel-release muestra la versión del núcleo
-v, --kernel-version print the kernel version
-m, --machine print the machine hardware name
-p, --processor print the processor type (non-portable)
-i, --hardware-platform print the hardware platform (non-portable)
-o, --operating-system print the operating system
--help muestra esta ayuda y finaliza
--version informa de la versión y finaliza
ayuda en línea sobre GNU coreutils: <http://www.gnu.org/software/coreutils/>
Informe de errores de traducción en uname a
<http://translationproject.org/team/es.html>
Full documentation at: <http://www.gnu.org/software/coreutils/uname>
or available locally via: info '(coreutils) uname invocation'
root@SoloLinux-demo:~$
```

Comando date

Si quieres saber el día de la semana, la fecha, la hora, y la zona horaria donde se encuentra el sistema, puedes usar el **comando date**.

```
date
```

Dos ejemplos diferentes.

```
root@SoloLinux-demo:~# date
Fri Aug 6 10:02:16 UTC 2021
root@SoloLinux-demo:~#
```

```
sergio@sololinux:~$ date
vie ago 6 13:01:51 EEST 2021
sergio@sololinux:~$
```

También puedes identificar datos variados de una fecha en particular, usando este formato **MM/dd/YYYY**.

```
date --date="03/09/1971"
```

Ejemplo de salida.

```
sergio@sololinux:~$ date --date="03/09/1971"
mar mar 9 00:00:00 MSK 1971
sergio@sololinux:~$
```

La herramienta date, ofrece muchas opciones y argumentos diversos, los vemos con **help**.

```
date --help
```

```
sergio@sololinux:~$ date --help
Modo de empleo: date [OPCIÓN]... [+FORMATO]
o bien: date [-u|--utc|--universal] [MMDDhhmm[[SS]AA][.ss]]
Muestra la hora actual en el FORMATO dado, o establece la fecha del sistema.

Los argumentos obligatorios para las opciones largas son también obligatorios para las opciones cortas.
-d, --date=STRING      display time described by STRING, not 'now'
                        and warn about questionable usage to stderr
--debug
-f, --file=DATEFILE   like --date; once for each line of DATEFILE
-I[FMT], --iso-8601[=FMT] output date/time in ISO 8601 format.
                        FMT='date' for date only (the default),
                        'hours', 'minutes', 'seconds', or 'ns'
                        for date and time to the indicated precision.
                        Example: 2006-08-14T02:34:56-06:00
-R, --rfc-email       output date and time in RFC 5322 format.
                        Example: Mon, 14 Aug 2006 02:34:56 -0600
--rfc-3339=FMT       output date/time in RFC 3339 format.
                        FMT='date', 'seconds', or 'ns'
                        for date and time to the indicated precision.
                        Example: 2006-08-14 02:34:56-06:00
-r, --reference=FILE  display the last modification time of FILE
-s, --set=STRING      set time described by STRING
-u, --utc, --universal print or set Coordinated Universal Time (UTC)
--help               muestra esta ayuda y finaliza
--version            informa de la versión y finaliza

FORMATO controla la salida. Las secuencias que se interpretan son:
%% un % literal
%a el nombre local abreviado de la semana (p. ej., dom)
%A el nombre local completo de la semana (p. ej., Domingo)
%b el nombre local abreviado del mes (p. ej., ene)
%B el nombre local completo del mes (p. ej. Enero)
www.sololinux.es
```

Comando hostname

Con **hostname** visualizamos el nombre del host.

```
hostname
```

```
root@SoloLinux-demo:~# hostname
SoloLinux-demo
root@SoloLinux-demo:~#
```

Hostname también tiene la capacidad de imprimir las IP del sistema, por ejemplo si queremos visualizar la **IPv4** usamos la opción «i».

```
hostname -i
```

```
root@SoloLinux-demo:~# hostname -i
185.183.20.171
root@SoloLinux-demo:~#
```

Si prefieres ampliar la información con la IP local, la IPv4 y la IPv6, agregamos «b».

```
hostname -I
```

El resultado se amplía considerablemente.

```
root@SoloLinux-demo:~# hostname -I
127.0.0.1 185.183.20.171 2a0b:7080:10::1:670e
root@SoloLinux-demo:~#
```

Comando whoami

Este comando nos muestra el usuario conectado actualmente.

```
whoami
```

Vemos dos ejemplos de resultado.

```
root@SoloLinux-demo:~# whoami
root
root@SoloLinux-demo:~#
```

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

Este comando no tiene funciones adicionales, aun siendo así puedes revisar su mínima ayuda integrada con este comando.

```
whoami --help
```

```
sergio@sololinux:~$ whoami --help
Modo de empleo: whoami [OPCIÓN]...
Muestra el nombre de usuario asociado con el ID efectivo del usuario actual.
Equivalente a id -un.

--help muestra esta ayuda y finaliza
--version informa de la versión y finaliza

ayuda en línea sobre GNU coreutils: <http://www.gnu.org/software/coreutils/>
Informe de errores de traducción en whoami a <http://translationproject.org/team/es.html>
Full documentation at: <http://www.gnu.org/software/coreutils/whoami>
or available locally via: info '(coreutils) whoami invocation'
sergio@sololinux:~$
www.sololinux.es
```

Comando w

Como último apunte de este artículo, el **comando w** es extremadamente útil para los **administradores de sistemas**, a la hora de identificar posibles usuarios no autorizados. Por ejemplo, con el comando básico listamos todos los usuarios conectados al sistema, así como otros datos importantes como su IP de procedencia.

```
w
```

```
root@SoloLinux-demo:~# w
10:43:14 up 2 days, 1:24, 3 user, load average: 0.00, 0.01, 0.05
USER TTY FROM LOGIN@ IDLE JCPU PCPU
WHAT
root pts/0 62.16.11.187 09:27 2.00s 0.08s 0.00s w
sergio pts/0 62.16.11.187 06:12 4.10s 0.12s 0.00s w
sololinux pts/0 62.16.11.187 04:08 1.05s 0.04s 0.00s w
root@SoloLinux-demo:~#
```

Si agregas el nombre, tan solo aportara los datos del usuario marcado. En nuestro ejemplo el **root**.

```
w root
```

Datos del root.

```
root@SoloLinux-demo:~# w root
10:50:38 up 2 days, 1:32, 1 user, load average: 0.00, 0.01, 0.05
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
root pts/0 62.16.11.187 09:27 6.00s 0.09s 0.00s w root
root@SoloLinux-demo:~#
```

Error: Sudo Command Not Found en Debian 10

Sudo, también llamado «**super user do**», es un comando para Linux que nos permite ejecutar comandos con privilegios de **usuario root**. Al ingresar tu contraseña personal, se confirman tus solicitudes verificando los permisos insertados en el archivo de **sudoers**.

Pero tal vez tengas una desagradable sorpresa, pues dependiendo de la versión, al realizar una nueva instalación de Debian 10, no podrás ejecutar tareas con privilegios utilizando el **comando sudo**; obtendrás el error «**Sudo Command Not Found en Debian 10**». La razón de este error es sencilla a la vez que incomprendible, el comando sudo no está incluido por defecto, en algunas instalaciones de Debian 10.

También se han detectado instalaciones, en las cuales sí existe la herramienta sudo, pero no se agregó ningún usuario. Si es tu caso, te saltas la parte de la instalación y directamente añades el usuario.

Error: Sudo Command Not Found en Debian 10

Está molesta situación, por suerte tiene fácil solución. Debes instalar sudo y agregar el usuario, para tener los permisos necesarios. Lo único que debes tener en cuenta es, iniciar sesión como root, ya que como un usuario normal resultara imposible hacer esta operación.

Cerramos la sesión del usuario actual.



Ahora iniciamos como **root**. En las imágenes de ejemplo operamos sobre **Debian 10 Desktop**, en Debian modo consola (servidor) el proceso sería el mismo.



Si te resulta imposible iniciar sesión como **root**, puedes obtener acceso con el comando «**su -**».

```
su -
```

Aunque nuestra distribución no tenga instalado sudo, sí que lo podemos encontrar en los repositorios oficiales. Instalamos la herramienta.

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

```
root@SoloLinux-demo:~# apt-get install sudo -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
sudo
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 514 kB of archives.
After this operation, 2257 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 sudo
amd64 1.8.31-1ubuntu1.2 [514 kB]
Fetched 514 kB in 0s (6435 kB/s)
Selecting previously unselected package sudo.
(Reading database ... 80060 files and directories currently
installed.)
Preparing to unpack .../sudo_1.8.31-1ubuntu1.2_amd64.deb ...
Unpacking sudo (1.8.31-1ubuntu1.2) ...
Setting up sudo (1.8.31-1ubuntu1.2) ...
Processing triggers for man-db (2.9.1-1) ...
root@SoloLinux-demo:~#
```

Busca el usuario que quieres usar como sudo, o como en nuestro caso... creas uno nuevo llamado **sololinux**.

```
adduser sololinux
```

Rellena los datos solicitados.

```
root@SoloLinux-demo:~# adduser sololinux
Adding user `sololinux' ...
Adding new group `sololinux' (1000) ...
Adding new user `sololinux' (1000) with group `sololinux' ...
Creating home directory `/home/sololinux' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for sololinux
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n]
root@SoloLinux-demo:~#
```

Ahora agregamos el usuario al **grupo sudo**, con el **comando usermod**.

```
usermod -aG sudo sololinux
```

Ahora verificamos que el **usuario sololinux**, también está en el **grupo sudo**.

```
id sololinux
```

```
root@SoloLinux-demo:~# id sololinux
uid=1000(sololinux) gid=1000(sololinux) groups=1000(sololinux),27(sudo)
root@SoloLinux-demo:~#
```

Llegado este momento deberías ejecutar sudo sin ningún problema, pero te lo puedes creer o no... no sería el primer caso que ni aun así, hemos logrado el acceso sudo. Tranquilo, tiene solución. Vamos a editar el archivo sudoers de forma manual, sin deshacer todo lo realizado anteriormente.

```
nano /etc/sudoers
```

Te encontrarás algo similar a esto...

```
GNU nano 4.8 /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults    env_reset
Defaults    mail_badpass
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"
# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
root    ALL=(ALL:ALL) ALL
# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
# See sudoers(5) for more information on "#include" directives:
#include   /etc/sudoers.d
```

Después de la última línea.

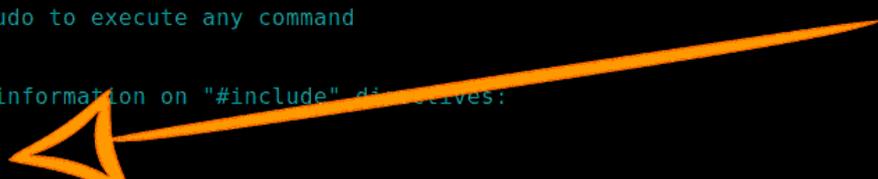
```
#includedir /etc/sudoers.d
```

Debes agregar el usuario que quieras (en nuestro caso **sololinux**), con los permisos correspondientes.

```
USUARIO ALL=(ALL:ALL) ALL
```

Debe quedar como en la imagen de ejemplo.

```
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults    env_reset
Defaults    mail_badpass
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"
# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
root    ALL=(ALL:ALL) ALL
# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
# See sudoers(5) for more information on "#include" directives:
#include   /etc/sudoers.d
sololinux ALL=(ALL:ALL) ALL
```



www.sololinux.es

Guarda el archivo y cierra el editor. Para concluir se recomienda reiniciar el sistema.

```
reboot
```

www.sololinux.es

Poner KALI Linux 2021 en español



KALI Linux es una de las mejores distribuciones linux, orientadas al **pentesting**, análisis y ataques de **fuerza bruta** u otros tipos más complejos.

Leo en muchos foros y chats, usuarios desesperados que no saben como poner **KALI Linux** en español. La verdad es que es una tarea bastante sencilla y simple, dejar nuestro **KALI** íntegramente en nuestro idioma nativo (excepto algunas herramientas específicas), incluyendo también el teclado y la zona geografica. Verás que tarea más fácil y atractiva tenemos hoy.

Poner KALI Linux 2021 en español

Te recomiendo obtener **permisos de root**, antes de comenzar. Para ello ejecuta el siguiente comando.

```
sudo -i
```

```
(kali@kali) - [~]
└─$ sudo -i

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

[sudo] password for kali:
(Message from Kali developers)

We have kept /usr/bin/python pointing to Python 2 for backwards
compatibility. Learn how to change this and avoid this message:
=> https://www.kali.org/docs/general-use/python3-transition/

(Run: "touch ~/.hushlogin" to hide this message)
└─(root@kali) - [~]
www.sololinux.es
```

Ahora debes **actualizar el sistema**. Si tienes un sistema recién instalado, puede demorar un buen rato.

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

Ejemplo de actualización de Kali Linux 2021.2.

```
(root kali)-[~]
└─# apt update && apt full-upgrade
Get:1 http://kali.koyanet.lv/kali kali-rolling InRelease [30,5 kB]
Get:2 http://kali.koyanet.lv/kali kali-rolling/main amd64 Packages [17,7 MB]
Get:3 http://kali.koyanet.lv/kali kali-rolling/main amd64 Contents (deb) [39,8 MB]
Get:4 http://kali.koyanet.lv/kali kali-rolling/contrib amd64 Packages [108 kB]
Get:5 http://kali.koyanet.lv/kali kali-rolling/contrib amd64 Contents (deb) [123 kB]
Get:6 http://kali.koyanet.lv/kali kali-rolling/non-free amd64 Packages [199 kB]
Get:7 http://kali.koyanet.lv/kali kali-rolling/non-free amd64 Contents (deb) [954 kB]
Fetched 58,9 MB in 32s (1,816 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
308 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  gstreamer1.0-pulseaudio python3-event python3-gevent-websocket
  python3-greenlet python3-jupyter-core python3-mcrypt python3-nbformat
  python3-parameterized python3-plotly python3-zope.event
Use 'apt autoremove' to remove them.
The following NEW packages will be installed:
  aspnecore-runtime-3.1 aspnecore-targeting-pack-3.1 dotnet-apphost-pack-3.1
  dotnet-host dotnet-hostfxr-3.1 dotnet-runtime-3.1 dotnet-runtime-deps-3.1
  dotnet-sdk-3.1 dotnet-targeting-pack-3.1 gnome-desktop3-data
  gstreamer1.0-plugins-bad libcd1394-25 libdca0 libdvdnv4 libdvddread8 libfaad2
  libfluidsynth2 libgnome-desktop-3-19 libgssdp-1.2-0
  libgstreamer-plugins-bad1.0-0 libguppnp-1.2-0 libguppnp-igd-1.0-4 liblmbase25
  libinstpatch-1.0-2 libkate1 libltd1 libmanette-0.2-0 libmjpegutils-2.1-0
  libmms0 libmodplug1 libmpcdec6 libmpeg2encpp-2.1-0 libmplex2-2.1-0 libnice10
  libofa0 libopenexr25 libopenni2-0 libpipewire-0.3-0 libpipewire-0.3-modules
  libSDL2-2.0-0 libsoundtouch1 libspa-0.2-modules libstrtp2-1 libvo-aacenc0
  libvo-amrwbenc0 libwildmidi2 libxkbregistry0 libzbar0
  linux-image-5.10.0-kali9-amd64 netstandard-targeting-pack-2.1 pipewire
  pipewire-bin python3-aiofiles python3-bidict python3-cocept python3-engineio
  python3-faraday-agent-parameters-types python3-fastapi python3-flask-socketio
  python3-orjson python3-pydantic python3-slowapi python3-socketio
  python3-starlette python3-uvicorn timgm6mb-soundfont xdg-desktop-portal
  xdg-desktop-portal-gtk
The following packages will be upgraded:
  amass amass-common apache2 apache2-bin apache2-data apache2-utils apt apt-utils
  aspell atftp base-passwd bash bluez bluez-hcidump bluez-obexd bsdxtrautils
  bsduutils busybox cifs-utils clang clang-9 console-setup console-setup-linux
  curl debconf debconf-i18n distro-info-data dmidecode e2fsprogs eject exploitdb
  faraday faraday-client fdisk fern-wifi-cracker firefox-esr
  firmware-amd-graphics firmware-atheros firmware-brcm80211 firmware-intel-sound
  firmware-iwlwifi firmware-libertas firmware-linux firmware-linux-nonfree
  firmware-misc-nonfree firmware-realtek firmware-sof-signed
  firmware-ti-connectivity freerdp2-x11 fuse3 galera-4 gdal-data
  gir1.2-gstreamer-1.0 gir1.2-javascriptcoregtk-4.0 gir1.2-webkit2-4.0
  grub-common grub-pc grub-pc-bin grub2-common gtk2-engines-pixbuf hotpatch
  ieee-data intel-microcode lpp-usb isc-dhcp-client isc-dhcp-common kali-defaults
  kali-defaults-desktop kali-desktop-core kali-desktop-xfce kali-linux-core
  kali-linux-default kali-linux-headless kali-tools-top10 keyboard-configuration
  kismet kismet-capture-common kismet-capture-linux-bluetooth
  kismet-capture-linux-wifi kismet-capture-nrf-51822 kismet-capture-nrf-52840
  kismet-capture-nrf-mousejack kismet-capture-nxp-kw41z
  kismet-capture-rz-killerbee kismet-capture-ti-cc-2531 kismet-capture-ti-cc-2540
  kismet-capture-ubertooth-one kismet-core kismet-logtools klibc-utils legion
  libapache2-mod-php7.4 libapt-pkg6.0 libarmadillo10 libaspell15 libass9
  libavcodec58 libavfilter7 libavformat58 libavresample4 libavutil56 libblkid1
  libbluetooth3 libbson-1.0-0 libc-bin libc-dev-bin libc-l10n libc6 libc6-dev
  libc6-386 libclang-common-9-dev libclang-cpp9 libcom-err2 libcupst2
  libcurl3-gnutls libcurl4 libdebconfclient0 libdns-export110 libegl mesa0
  libext2fs2 libfdisk1 libfreerdp-client2-2 libfreerdp2-2 libfuse3-3 libgpm1
  libgrypt20 libgdal20 libgl1-mesa-dri libglapi-mesa libglx-mesa0 libgnutls30
  libgssapi-krb5-2 libgstreamer1.0-0 libgtk2.0-0 libgtk2.0-common libhd5-103-1
  libhd5-hl-100 libhogweed6 libicu67 libisc-export1105
  libjavascripcoregtk-4.0-18 libk5crypto3 libklibc libkrb5-3 libkrb5support0
  liblvm9 libmariadb3 libmm-glib0 libmongoc-1.0-0 libmount1 libnettle8
  libnftables1 libnginx-mod-http-geoip libnginx-mod-http-image-filter
  libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  libnginx-mod-stream-geoip libnss-systemd libogg1.4 libpam-modules
  libpam-modules-bin libpam-runtime libpam-systemd libpam0g libpfm4
  libpolkit-agent-1-0 libpolkit-gobject-1-0 libpostproc55 libpq5
  libpython2-stdlib libpython2.7-minimal libpython2.7-stdlib libqt5core5a
  libqt5dbus5 libqt5gui5 libqt5network5 libqt5sprintsupport5 libqt5qml5
  libqt5qmlmodels5 libqt5quick5 libqt5sql5 libqt5sql5-sqlite libqt5test5
  libqt5widgets5 libqt5xml5 libruby2.7 libsane-common libsane1 libserd-0-0
  libsmartcols1 libsndfile1 libssl2 libswresample3 libswscale5 libsystemd0
  libudev1 libuid1 libuv1 libuv1-dev libwebkit2gtk-4.0-37 libwebp6 libwebpdemux2
  libwebpmux3 libwinpr2-2 libwp6-1.0-1 libx11-6 libx11-data libx11-xcb1
  libx264-160 libxatracker2 libxml2 libxml2-utils libxvctrl0 linux-image-amd64
  linux-libc-dev llvm-9 llvm-9-dev llvm-9-runtime llvm-9-tools locales logsave
  mariadb-client-10.5 mariadb-client-core-10.5 mariadb-common mariadb-server-10.5
  mariadb-server-core-10.5 mesa-va-drivers mesa-vdpau-drivers mesa-vulkan-drivers
  metasploit-framework mtkat2 minicom modemmanager mount netdiscover nfs-common
  nftables nginx nginx-common nginx-core openjdk-11-jre openjdk-11-jre-headless
  os-prober passing-the-hash php7.4 php7.4-cli php7.4-common php7.4-jison
  php7.4-mysql php7.4-opcache php7.4-readline policykit-1 postresql-13
  postgresql-client-13 powershell-empire python-apt-common python2
  python2-minimal python2.7 python2.7-minimal python3-aiohttp python3-apt
  python3-autobahn python3-dateutil python3-django python3-faraday-plugins
  python3-gdal python3-impacket python3-kismetcapturefreaklabszigbee
  python3-kismetcaptureurl1433 python3-kismetcaptureurladsb
  python3-kismetcaptureurlamr python3-lxml python3-pil python3-pkg-resources
  python3-pygments python3-setuptools python3-six python3-urllib3 python3-yaml
  qt5-gtk-platformtheme qtbase5-dev-tools rebind rfid ruby-activestorage
  ruby-addressable ruby-cms-scanner ruby2.7 ruby2.7-dev sane-utils spoonstooth
  sqlmap ssl-cert starkiller systemd systemd-sysv theharvester udev uids
  util-linux virtualbox-guest-utils virtualbox-guest-x11 voipopper whois pcapn
  zaproxy zsh
308 upgraded, 68 newly installed, 0 to remove and 0 not upgraded.
Need to get 1,018 MB of archives.
After this operation, 814 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Una vez actualizado el **sistema KALI**, te recomiendo reiniciar el sistema.

```
reboot
```

Una vez reiniciado el sistema, accede de nuevo a la **terminal de Kali Linux** y ejecuta...

```
sudo -i
```

Bien, ahora identificamos el idioma instalado en el sistema. Por defecto Kali viene en ingles.

```
locale
```

```
(root@kali) - [~]
# locale
LANG=en_US.UTF-8
LANGUAGE=
LC_CTYPE="en_US.UTF-8"
LC_NUMERIC="en_US.UTF-8"
LC_TIME="en_US.UTF-8"
LC_COLLATE="en_US.UTF-8"
LC_MONETARY="en_US.UTF-8"
LC_MESSAGES="en_US.UTF-8"
LC_PAPER="en_US.UTF-8"
LC_NAME="en_US.UTF-8"
LC_ADDRESS="en_US.UTF-8"
LC_TELEPHONE="en_US.UTF-8"
LC_MEASUREMENT="en_US.UTF-8"
LC_IDENTIFICATION="en_US.UTF-8"
LC_ALL=

www.sololinux.es
```

Esta distribución forense viene por defecto con muchos idiomas incluidos. Lamentablemente, solo trae habilitado el inglés, pero tranquilo que esto tiene fácil solución. Tan solo debe ejecutar este comando.

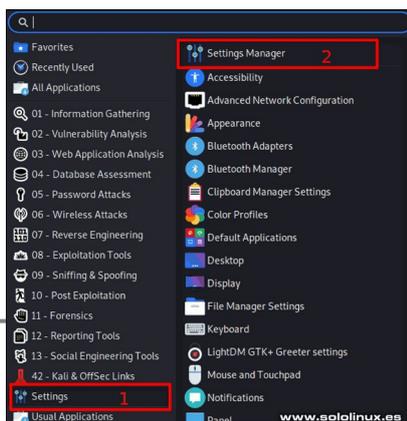
```
localectl set-locale LANG=es_ES.UTF-8
```

```
(root@kali) - [~]
# localectl set-locale LANG=es_ES.UTF-8
127 x

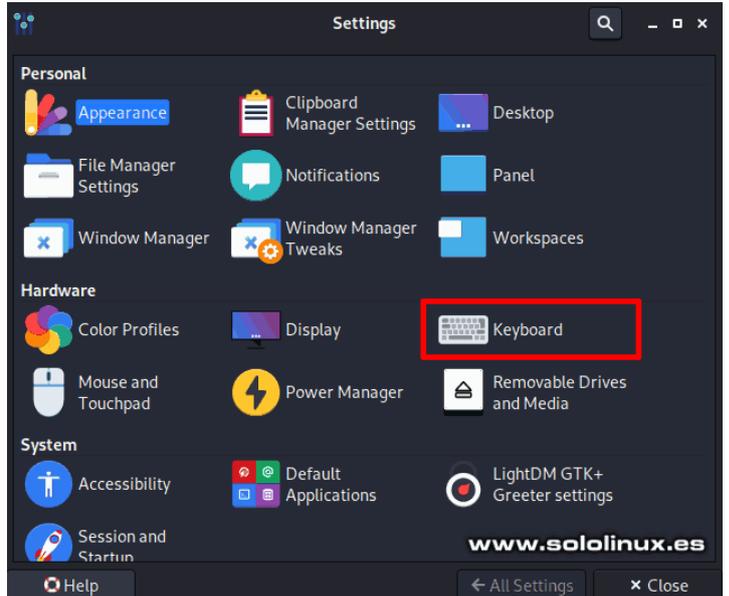
www.sololinux.es
```

En estos momentos, aunque no veas ningún cambio ya tienes instalado y habilitado el idioma español. Para aplicar los cambios debes reiniciar el sistema, pero espera un poco, aún no hemos terminado todo el proceso.

En el panel de aplicaciones de **KALI**, buscas y haces click sobre **«Settings»**, en el desplegable pulsas sobre **«Settings Manager»**. Observa la imagen de ejemplo.

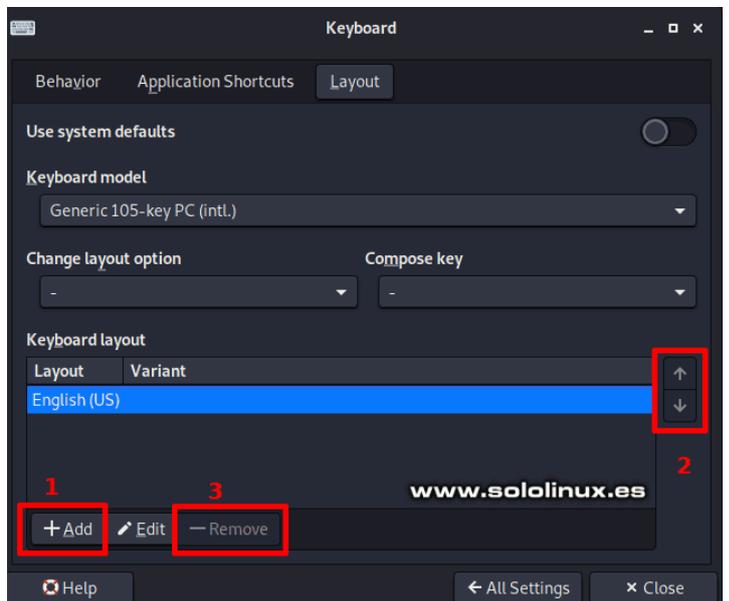


Vemos varias herramientas de configuración, abrimos **«keyboard»**.



Vas a la sección Layout y sigue los pasos que te indico.

- **Add** – Agrega el idioma español en cualquiera de sus variantes.
- **Flechas** – Marca el español y lo colocas el primero.
- **Remove** – Selecciona el idioma ingles y pulsas en Remove.



Pulsas en **Close** y sales de la herramienta. Para terminar el proceso, es obligatorio ejecutar el siguiente comando.

```
setxkbmap es sundeadkeys
```

Ya casi terminamos, nos falta cambiar la zona horaria. Te recomiendo listar las zonas, para cerciorarte que eliges el correcto.

```
sudo -i
timedatectl list-timezones
```

Recorre el listado al completo, hasta que localices tu zona más próxima.

```
(kali kali)-[~]
└─$ sudo -i
(Message from Kali developers)

We have kept /usr/bin/python pointing to Python 2 for backwards
compatibility. Learn how to change this and avoid this message:
→ https://www.kali.org/docs/general-use/python3-transition/

(Run: "touch ~/.hushlogin" to hide this message)
└─(root kali)-[~]
└─# timedatectl list-timezones
Africa/Abidjan
Africa/Accra
Africa/Algiers
Africa/Bissau
Africa/Cairo
Africa/Casablanca
Africa/Ceuta
Africa/El_Aaiun
Africa/Johannesburg
Africa/Juba
////////////////////////////////////
America/Belize
America/Blanc-Sablon
America/Boa_Vista
America/Bogota
America/Boise
America/Cambridge_Bay
America/Campo_Grande
America/Cancun
America/Caracas
America/Cayenne
America/Chicago
America/Chihuahua
America/Costa_Rica
America/Creston
America/Cuiaba
America/Curacao
America/Danmarkshavn
America/Dawson
America/Dawson_Creek
America/Denver
America/Detroit
America/Edmonton
America/Eirunepe
America/El_Salvador
lines 42-65
////////////////////////////////////
Europe/Copenhagen
Europe/Dublin
Europe/Gibraltar
Europe/Helsinki
Europe/Istanbul
Europe/Kaliningrad
Europe/Kiev
Europe/Kirov
Europe/Lisbon
Europe/London
Europe/Luxembourg
Europe/Madrid
Europe/Malta
Europe/Minsk
Europe/Monaco
Europe/Moscow
Europe/Oslo
Europe/Paris
Europe/Prague
Europe/Riga
Europe/Rome
Europe/Samara
Europe/Saratov
Europe/Simferopol
lines 268-291
```

En nuestro caso la zona más próxima es Europa/Madrid, por tanto ejecutamos lo siguiente.

```
timedatectl set-timezone Europe/Madrid
```

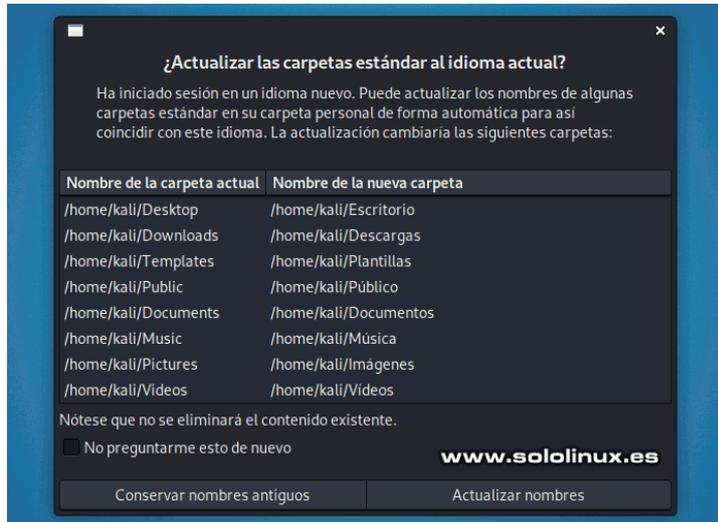
En pocos segundos veras como se cambia la hora, de manera automatica.



Reiniciamos el sistema.

```
reboot
```

Una vez el sistema haya iniciado de forma correcta, aparece una ventana en la cual nos indica si queremos actualizar el nombre de las carpetas. Es evidente que debes decir que si quieres actualizar, si no lo haces... los nombres continuaran estando en ingles.



Terminamos actualizando de nuevo nuestro KALI y, borrando los archivos inútiles.

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

Concluida la actualización, ejecutamos remove.

```
apt autoremove
```

```
(root kali)-[~]
└─# apt autoremove
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los siguientes paquetes se ELIMINARÁN:
  gstreamer1.0-pulseaudio python3-gevent python3-gevent-websocket
  python3-greenlet python3-jupyter-core python3-m2crypto python3-
  nbformat
  python3-parameterized python3-plotly python3-zope.event
0 actualizados, 0 nuevos se instalarán, 10 para eliminar y 0 no
actualizados.
Se liberarán 47,0 MB después de esta operación.
¿Desea continuar? [S/n]
```

NOTA: Dependiendo de la versión de KALI instalada, pueden fallar o no ser estable las locales del idioma o del teclado. Si es tu caso revisa este anterior artículo.

Instalar DVWA en KALI Linux 2021



Rompiendo se aprende y la práctica hace al maestro, pero tenemos una duda, ¿dónde podemos practicar nuestras habilidades de **hacking ético**? La respuesta es sencilla, en **DVWA (Damn Vulnerable Web Application)**. Hablamos de una **aplicación web vulnerable** desarrollada en PHP/MySQL, cuyo principal objetivo es permitir a los profesionales de la seguridad evaluar sus conocimientos, en un entorno completamente legal.

Estas prácticas ayudan a los desarrolladores web a comprender mejor el proceso de seguridad de sus aplicaciones. También resulta extremadamente útil para estudiantes, profesores y aficionados, en el proceso de aprendizaje en temas de seguridad sobre aplicaciones web.

Cualquier usuario que quiera empezar con la piratería ética, debe realizar sus pruebas, pero no olvides que cualquier experimento con intentos de penetración en sistemas ajenos sin permiso, es ilegal y puede acarrear graves problemas, incluyendo cuantiosas multas o hasta prisión. Mejor no te la juegues y por ello, en el artículo de hoy vemos como instalar esta aplicación como base de pruebas, en tu instalación de **KALI Linux en español**.

Instalar DVWA en KALI Linux 2021

KALI Linux ya viene con su servidor local instalado, accedemos directamente a la **carpeta html**.

```
cd /var/www/html
```

Ahora descargamos DVWA.

```
sudo git clone https://github.com/ethicalhack3r/DVWA
```

```
(kali) kali-[/var/www/html]
└─$ cd /var/www/html

(kali) kali-[/var/www/html]
└─$ sudo git clone https://github.com/ethicalhack3r/DVWA
[sudo] password for kali:
Clonando en 'DVWA'...
remote: Enumerating objects: 3398, done.
remote: Counting objects: 100% (49/49), done.
remote: Compressing objects: 100% (32/32), done.
remote: Total 3398 (delta 20), reused 35 (delta 16), pack-reused 3349
Recibiendo objetos: 100% (3398/3398), 1.65 MiB | 4.46 MiB/s, listo.
Resolviendo deltas: 100% (1510/1510), listo.
```

```
(kali) kali-[/var/www/html]
└─$
```

Sin salir de **/var/www/html**, listamos el contenido descargado.

```
ls
```

Vemos el contenido de **/html**.

```
(kali) kali-[/var/www/html] www.sololinux.es
└─$ ls
DVWA  index.html  index.nginx-debian.html
```

En la imagen anterior, puedes observar que se encuentran dos archivos y un directorio llamado **DVWA**. Le concedemos los permisos necesarios.

```
sudo chmod -R 777 DVWA
```

DVWA contiene una carpeta llamada **config**, accedemos a ella.

```
cd DVWA/config
```

En **config** nos encontramos con el archivo de configuración predeterminado, es mejor dejarlo como copia de seguridad y generar otro.

```
sudo cp config.inc.php.dist config.inc.php
```

Abrimos el archivo con el **editor nano**.

```
sudo nano config.inc.php
```

Si quieres puedes modificar el usuario y la contraseña, pero no te olvides de los datos porque los vas a necesitar después. Una vez editado el archivo (si es tu caso), lo guardas y cierras el editor. **No cambies el nombre de la base de datos**.

```
# Database management system to use
$dbMS = 'MySQL';
# $dbMS = 'PGSQL'; // Currently disabled

# Database variables
# WARNING: The database specified under db_database WILL BE ENTIRELY DELETED during setup.
# Please use a database dedicated to DVWA.
#
# If you are using MariaDB then you cannot use root, you must use create a dedicated DVWA user.
# See README.md for more information on this.
$DVWA = array();
$DVWA['db_server'] = '127.0.0.1';
$DVWA['db_database'] = 'dvwa';
$DVWA['db_user'] = 'dvwa';
$DVWA['db_password'] = 'p@ssw0rd';
$DVWA['db_port'] = '3306';
www.sololinux.es
```

KALI ya viene con el motor de base de datos instalado, lo iniciamos.

```
sudo systemctl start mysql
```

Verificamos que todo es correcto.

```
systemctl status mysql
```

```
(kali) kali-[/var/www/html/DVWA/config]
└─$ sudo systemctl start mysql

(kali) kali-[/var/www/html/DVWA/config]
└─$ systemctl status mysql
● mariadb.service - MariaDB 10.5.11 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; disabled; vendor preset: disabled)
   Active: active (running) since Sun 2021-08-08 11:08:35 CEST; 21s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 2885 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysql (code=exited, status=0/SUCCESS)
   Process: 2886 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
   Process: 2888 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= [ VAR=cd /usr/bin/..; /usr/bin/galera_recover>
   Process: 2949 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
   Process: 2951 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
  Main PID: 2935 (mariabdb)
    Status: "Taking your SQL requests now..."
     Tasks: 14 (limit: 2295)
    Memory: 111.2M
       CPU: 3.388s
    CGroup: /system.slice/mariadb.service
           └─2935 /usr/sbin/mariabdb
```

Accedemos a la **consola de MariaDB** y creamos el usuario con su password. Recuerda que estos datos los proporcionamos anteriormente, en el archivo config.inc.php; en nuestro caso utilizamos los que venían por defecto, porque al ser un entorno de pruebas local, no existe ningún riesgo. Sigue los pasos que te indico a continuación.

```
sudo mysql -u root -p
create user 'userDVWA'@'127.0.0.1' identified by "dvwa";
grant all privileges on dvwa.* to 'userDVWA'@'127.0.0.1' identified by
'dvwa';
exit
```

Ejemplo completo del proceso con respuestas incluidas.

```
(kali)~# sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 44
Server version: 10.5.11-MariaDB-1 Debian 11
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.
MariaDB [(none)]> create user 'userDVWA'@'127.0.0.1' identified by
'dvwa';
Query OK, 0 rows affected (0.008 sec)
MariaDB [(none)]> grant all privileges on dvwa.* to
'userDVWA'@'127.0.0.1' identified by 'dvwa';
Query OK, 0 rows affected (0.008 sec)
MariaDB [(none)]> exit
Bye
```

Para instalar DVWA en KALI Linux 2021 correctamente, debemos hacer unas modificaciones en el archivo **php.ini**. La versión actual de php en KALI es la 7.4, si cambia la versión abres el archivo que corresponda.

```
sudo nano /etc/php/7.4/apache2/php.ini
```

Busca las líneas **«allow_url_fopen»** y **«allow_url_include»**. Asegúrate que las dos están en **«On»**.

```
;;;;;;;;;;;;;;;;;;;;;;;;;
; Fopen wrappers ;
;;;;;;;;;;;;;;;;;;;;;;;;;
; Whether to allow the treatment of URLs (like http:// or ftp://) as
files.
; http://php.net/allow-url-fopen
allow_url_fopen = On
; Whether to allow include/require to open URLs (like http:// or
ftp://) as files.
; http://php.net/allow-url-include
allow_url_include = Off
```

Una vez tengas las dos en **«On»**, guarda el archivo y cierra el editor.

```
;;;;;;;;;;;;;;;;;;;;;;;;;
; Fopen wrappers ;
;;;;;;;;;;;;;;;;;;;;;;;;;
; Whether to allow the treatment of URLs (like http:// or ftp://) as
files.
; http://php.net/allow-url-fopen
allow_url_fopen = On
; Whether to allow include/require to open URLs (like http:// or
ftp://) as files.
; http://php.net/allow-url-include
allow_url_include = On
; Define the anonymous ftp password (your email address). PHP's default setting
; for this is empty.
; http://php.net/from
from="john@doe.com" www.sololinux.es
```

Para dejar la herramienta lista para su uso, solo falta **iniciar Apache**.

```
sudo systemctl start apache2
```

Verificamos que el servicio está corriendo.

```
systemctl status apache2
```

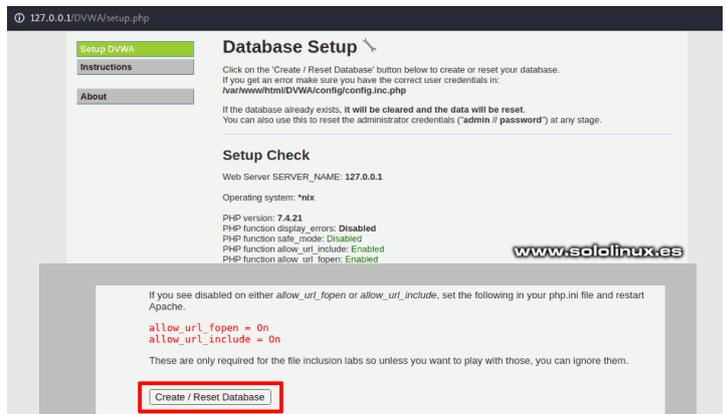
```
(kali)~# sudo systemctl start apache2

(kali)~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; disabled;
   vendor preset: disabled)
   Active: active (running) since Sun 2021-08-08 11:23:34 CEST; 40s
   ago
     Docs: https://httpd.apache.org/docs/2.4/
    Process: 3176 ExecStart=/usr/sbin/apachectl start (code=exited,
   status=0/SUCCESS)
   Main PID: 3187 (apache2)
     Tasks: 6 (limit: 2295)
    Memory: 18.5M
       CPU: 292ms
    CGroup: /system.slice/apache2.service
           └─3187 /usr/sbin/apache2 -k start
           └─3189 /usr/sbin/apache2 -k start
           └─3190 /usr/sbin/apache2 -k start
           └─3191 /usr/sbin/apache2 -k start
           └─3192 /usr/sbin/apache2 -k start
           └─3193 /usr/sbin/apache2 -k start
```

Una vez hemos terminado de instalar DVWA en KALI Linux 2021, iniciamos la herramienta insertando esta URL en tu navegador web preferido.

<http://127.0.0.1/DVWA/setup.php>

Verás una pantalla con diversos errores, no te preocupes es algo normal, ya que falta generar la base de datos. Lo único que nos falta es pulsar en **«Crear database»**, para que se genere y nos redirija a la página de inicio de sesión.



Puedes iniciar sesión con los datos de acceso predeterminados, que son...

- **Username:** admin
- **Password:** password



Username
admin www.sololinux.es

Password
●●●●●●

Login

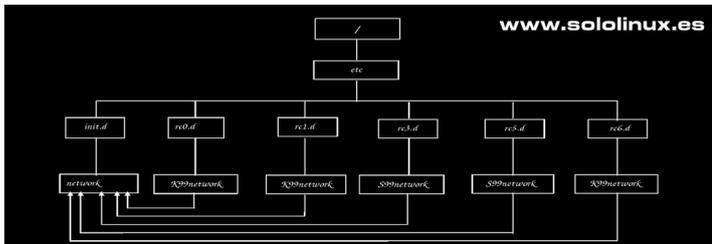
Qué son los niveles de ejecución en Linux

Cuando arrancamos un sistema Linux, este inicia con unas condiciones predefinidas, que definen su estado. Por ejemplo, podemos iniciar un sistema Linux en línea de comandos y trabajar desde allí. Normalmente, esto esta indicado para usuarios expertos y **sysadmin**.

Por otro lado, tenemos el **inicio de sesión con GUI**, que es el más común y práctico para la inmensa mayoría de usuarios. Estos son estados diferentes, que comúnmente los denominamos niveles de ejecución. Determinar estos niveles es fundamental, pues de ellos dependen el uso que le daremos al sistema.

Especifica las herramientas y aplicaciones, que deben ejecutarse o estar disponibles al iniciar nuestro Linux.

Fijar el estado del sistema cuando arranca, por ejemplo si usara entorno de escritorio (GUI) o no.



Qué son los niveles de ejecución en Linux

Antes de comenzar con el artículo, puedes identificar el nivel de tu sistema con el siguiente comando.

```
runlevel
```

La salida del comando nos imprime un número, que en este caso es 5. Este es el nivel predeterminado en la mayoría de **distribuciones Linux**.

```
root@SoloLinux-demo:~# runlevel
N 5
root@SoloLinux-demo:~#
```

Puedes ver el manual de runlevel, con el **comando man**.

```
man runlevel
```

```
RUNLEVEL(8)                                runlevel
NAME
runlevel - Print previous and current SysV runlevel
SYNOPSIS
runlevel [options...]
OVERVIEW
"Runlevels" are an obsolete way to start and stop groups of services used in SysV init. systemd provides a compatibility layer that maps runlevels to targets, and associated binaries like runlevel. Nevertheless, only one runlevel can be "active" at a given time, while systemd can activate multiple targets concurrently, so the mapping to runlevels is confusing and only approximate. Runlevels should not be used in new code, and are mostly useful as a shorthand way to refer the matching systemd targets in kernel boot parameters.
Table 1. Mapping between runlevels and systemd targets
Runlevel Target
0          poweroff.target
1          rescue.target
2, 3, 4   multi-user.target
5          graphical.target
6          reboot.target
DESCRIPTION
runlevel prints the previous and current SysV runlevel if they are known. The two runlevel characters are separated by a single space character. If a runlevel cannot be determined, N is printed instead. If neither can be determined, the word "unknown" is printed.
Manual page runlevel(8) line 1/62 67% (press h for help or q to quit)
```

Como puedes comprobar, en el **manual de runlevel** nos indica que en realidad tenemos 7 niveles. Tal vez puedan parecer un poco complejos, pero te aseguro que no lo son. En la siguiente tabla lo verás todo mucho más claro.

Niveles	Objetivo	Descripción
0	poweroff.target	Apagado del sistema
1	rescue.target	Ejecuta un modo de usuario único utilizado por el root. No se permiten otros usuarios y tampoco se inician demonios. Se utiliza para reparar sistemas.
2	multi-user.target	Se ejecuta en modo multiusuario, pero sin iniciar demonios, ni interfaces de red.
3	multi-user.target	Lanza el sistema en modo multiusuario, pero sin interfaz gráfica de usuario. Solo se permite la interfaz de línea de comandos
4	multi-user.target	No está definido por tanto no se utiliza. Se permite que el usuario determine su uso.
5	graphical.target	Inicia el sistema en modo multiusuario interfaz gráfica de usuario. Este nivel es el más utilizado por las distribuciones linux actuales.
6	reboot.target	Se reinicia el sistema.

También puedes listar los niveles de tu sistema, con este comando.

```
ls -l /lib/systemd/system/runlevel*
sergio@sololinux:~$ ls -l /lib/systemd/system/runlevel*
lrwxrwxrwx 1 root root 15 jul 21 21:51 /lib/systemd/system/runlevel0.target -> poweroff.target
lrwxrwxrwx 1 root root 13 jul 21 21:51 /lib/systemd/system/runlevel1.target -> rescue.target
lrwxrwxrwx 1 root root 17 jul 21 21:51 /lib/systemd/system/runlevel2.target -> multi-user.target
lrwxrwxrwx 1 root root 17 jul 21 21:51 /lib/systemd/system/runlevel3.target -> multi-user.target
lrwxrwxrwx 1 root root 17 jul 21 21:51 /lib/systemd/system/runlevel4.target -> multi-user.target
lrwxrwxrwx 1 root root 16 jul 21 21:51 /lib/systemd/system/runlevel5.target -> graphical.target
lrwxrwxrwx 1 root root 13 jul 21 21:51 /lib/systemd/system/runlevel6.target -> reboot.target

/lib/systemd/system/runlevel1.target.wants:
total 0

/lib/systemd/system/runlevel2.target.wants:
total 0

/lib/systemd/system/runlevel3.target.wants:
total 0

/lib/systemd/system/runlevel4.target.wants:
total 0

/lib/systemd/system/runlevel5.target.wants:
total 0
sergio@sololinux:~$
```

Cada nivel de ejecución, es un entorno específico y particular que se ejecuta a través de un grupo de scripts. Estos scripts se guardan en directorios únicos, que a su vez se almacenan en el directorio /etc. Estos son los directorios que contienen los archivos específicos y sus **scripts**.

- rc0.d
- rc1.d
- rc2.d
- rc3.d
- rc4.d
- rc5.d
- rc6.d

En nuestro primer ejemplo, vemos el contenido de «rc0.d».

```
ls /etc/rc0.d
```

Se imprime lo siguiente, pero ojo... esto puede variar dependiendo de tu sistema y configuración.

```
sergio@sololinux:~$ ls /etc/rc0.d
K01alsa-utils K01cryptdisks-early K01hdtemp K01lightdm K01networking K01rsyslog K01uuidd
K01avahi-daemon K01cups-browsed K01irqbalance K01lvm2-lvmetad K01openvpn K01saned
K01cryptdisks K01exim4 K01kerneloops K01lvm2-lvmpolld K01plymouth K01speech-dispatcher
sergio@sololinux:~$
```

También puedes ver el contenido de todos los archivos, incluyendo otro directorio rc que contiene otro tipo de scripts que inician con el sistema.

```
ls /etc/rc*.*

sergio@sololinux:~$ ls /etc/rc*.*
/etc/rc0.d:
K01alsa-utils      K01cryptdisks-early K01hddtemp      K01lightdm
K01networking     K01rsyslog          K01uuidd
K01avahi-daemon   K01cups-browsed    K01irqbalance  K01lvm2-lvmetad
K01openvpn        K01saned
K01cryptdisks     K01exim4           K01kerneloops  K01lvm2-lvmpolld
K01plymouth       K01speech-dispatcher
/etc/rc1.d:
K01alsa-utils     K01cups-browsed    K01irqbalance  K01lvm2-lvmetad
K01rsyslog        K01ufw
K01avahi-daemon  K01exim4          K01kerneloops  K01lvm2-lvmpolld
K01saned          K01uuidd
K01cups           K01hddtemp        K01lightdm     K01openvpn
K01speech-dispatcher
/etc/rc2.d:
K01cups           S01binfmt-support S01dbus         S01irqbalance
S01lvm2-lvmpolld S01plymouth       S01speech-dispatcher
S01acpid          S01console-setup.sh S01exim4       S01kerneloops
S01mintsytem     S01rsync          S01sysstat
S01anacron       S01cron           S01grub-common S01lightdm
S01ntp           S01rsyslog        S01uuidd
S01avahi-daemon  S01cups-browsed   S01hddtemp     S01lvm2-lvmetad
S01openvpn       S01saned
/etc/rc3.d:
K01cups           S01binfmt-support S01dbus         S01irqbalance
S01lvm2-lvmpolld S01plymouth       S01speech-dispatcher
S01acpid          S01console-setup.sh S01exim4       S01kerneloops
S01mintsytem     S01rsync          S01sysstat
S01anacron       S01cron           S01grub-common S01lightdm
S01ntp           S01rsyslog        S01uuidd
S01avahi-daemon  S01cups-browsed   S01hddtemp     S01lvm2-lvmetad
S01openvpn       S01saned
/etc/rc4.d:
K01cups           S01binfmt-support S01dbus         S01irqbalance
S01lvm2-lvmpolld S01plymouth       S01speech-dispatcher
S01acpid          S01console-setup.sh S01exim4       S01kerneloops
S01mintsytem     S01rsync          S01sysstat
S01anacron       S01cron           S01grub-common S01lightdm
S01ntp           S01rsyslog        S01uuidd
S01avahi-daemon  S01cups-browsed   S01hddtemp     S01lvm2-lvmetad
S01openvpn       S01saned
/etc/rc5.d:
K01cups           S01binfmt-support S01dbus         S01irqbalance
S01lvm2-lvmpolld S01plymouth       S01speech-dispatcher
S01acpid          S01console-setup.sh S01exim4       S01kerneloops
S01mintsytem     S01rsync          S01sysstat
S01anacron       S01cron           S01grub-common S01lightdm
S01ntp           S01rsyslog        S01uuidd
S01avahi-daemon  S01cups-browsed   S01hddtemp     S01lvm2-lvmetad
S01openvpn       S01saned
/etc/rc6.d:
K01alsa-utils     K01cryptdisks-early K01hddtemp      K01lightdm
K01networking     K01rsyslog          K01uuidd
K01avahi-daemon   K01cups-browsed    K01irqbalance  K01lvm2-lvmetad
K01openvpn        K01saned
K01cryptdisks     K01exim4           K01kerneloops  K01lvm2-lvmpolld
K01plymouth       K01speech-dispatcher
/etc/rcS.d:
S01alsa-utils     S01cryptdisks      S01dns-clean    S01kmod
S01lvm2          S01plymouth-log   S01procps       S01ufw
S01apparmor      S01cryptdisks-early S01keyboard-setup.sh S01lmsensors
S01networking   S01pppd-dns       S01udev         S01x11-common
sergio@sololinux:~$
```

Bueno, ahora ya sabemos qué es un nivel de ejecución y, los directorios donde se almacenan sus archivos de configuración. Ahora vemos como ejecutar y cambiar el nivel de ejecución. Para esta operación disponemos de dos comandos, el init y el telinit. En nuestro caso usamos el init para reiniciar el sistema.

```
sudo init 6
```

El sistema reinicia instantáneamente, al iniciar volverá a su nivel anterior, el 5. Con el comando **systemctl**, también nos ayuda a cambiar de nivel, pero de una forma un poco diferente. En vez de usar el carácter numérico del nivel, opera con el nombre del objetivo (revisa la tabla superior). Identificamos el nivel actual.

```
systemctl get-default
```

La respuesta es **«graphical.target»**, que corresponde al nivel de ejecución 5.

```
sergio@sololinux:~$
sergio@sololinux:~$ systemctl get-default
graphical.target
sergio@sololinux:~$
```

Si quieres cambiar al nivel 3 (modo línea de comandos), ejecuta este comando.

```
sudo systemctl isolate multi-user.target
```

Si ejecutas **runlevel** de nuevo, te dirá el nivel actual y el nivel anterior.

```
root@SoloLinux-demo:~# runlevel
5 3
root@SoloLinux-demo:~#
```

Para volver al sistema con GUI, cambiamos de nivel otra vez.

```
sudo systemctl isolate graphical.target
```

Como último apunte de este artículo, ejecutamos de nuevo el comando **runlevel**. Vemos que el valor de los niveles se ha invertido, ya que solo tiene la capacidad de imprimir dos niveles, el actual y el anterior.

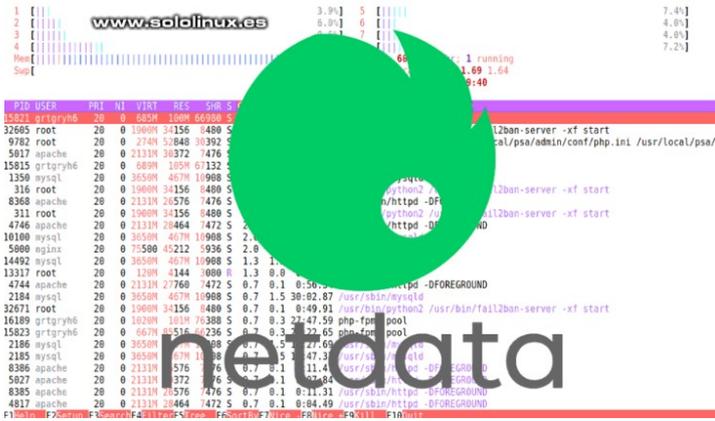
```
root@SoloLinux-demo:~#
root@SoloLinux-demo:~# runlevel
3 5
root@SoloLinux-demo:~#
```

SoloWordPress

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

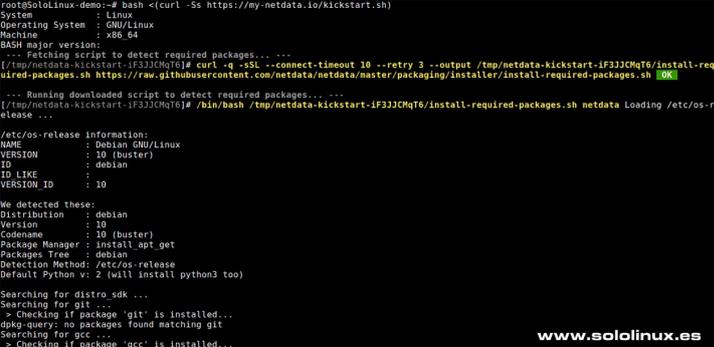


Instalar Netdata Monitoring en Ubuntu 20.04



```
sudo bash <(curl -Ss https://my-netdata.io/kickstart.sh)
```

Una vez se descarga el script, la instalación comienza al instante.



Una correcta supervisión del servidor, es fundamental a la hora de **administrar un servidor Linux**. Independientemente de la complejidad de nuestra infraestructura, debemos monitorizar constantemente muchos aspectos de nuestra de red y hardware. Es la única forma de asegurar un adecuado funcionamiento y, evitar posibles problemas antes de que sucedan.

No todos los usuarios tienes el conocimiento para instalar, configurar y utilizar herramientas de monitorización a través de **línea de comandos**. Todas o por lo menos la mayoría de estas herramientas, las podemos sustituir por una excelente aplicación que podrás controlar vía navegador web de manera gráfica. Hablamos de **Netdata Monitoring**.

Netdata es una de las mejores y más sencillas herramientas de supervisión, que permite a los usuarios vigilar las partes fundamentales de un **servidor o VPS**. Esta herramienta gratuita y de **código abierto**, recopila miles de métricas de hardware y software para organizar los datos en una interfaz fácil de usar. Todas las métricas se visualizan y muestran en un panel interactivo e intuitivo, en forma de cuadros gráficos.



Instalar Netdata Monitoring en Linux

En el artículo de hoy, vemos como **instalar Netdata en tu servidor o VPS**, en cualquier **distribución Linux** moderna. Nosotros la hemos probado en Ubuntu 20.04, Debian 10, CentOS 8 y Alma Linux 8.4, con un resultado satisfactorio en todos los casos. Fácil de instalar y **fácil de usar, así de simple es Netdata**.

Antes de comenzar, asegúrate de tener tu sistema actualizado. Después descargamos un script proporcionado por el desarrollador oficial, que automatiza casi todo el proceso.

Varias veces te solicitará pulsar la **tecla enter** para continuar, otras que pulses **«Y»**.



Como instalar KALI Linux en VirtualBox 6.1



Kali Linux es una de las pocas distribuciones Linux existentes en la actualidad, que son prácticamente exclusivas para profesionales de la seguridad, aunque también admite usuarios aficionados. KALI es una fusión entre el discontinuado WHAX y Auditor Security Collection.

La principal diferencia entre Kali Linux y otras distribuciones, es que KALI incluye de serie un montón de utilidades específicas que han sido desarrolladas para operar en áreas de seguridad. Debes tener claro, que este sistema no está destinado para un uso principal, ya que su forma de operar es muy específica.

El objetivo principal de este artículo, es explicar de forma simple cómo instalar Kali Linux en VirtualBox, así como algunos pequeños matices con los que nos hemos encontrado. Para este post, hemos descargado la versión específica para VirtualBox, que puedes encontrar aquí.



Como instalar KALI Linux en VirtualBox 6.1

Es altamente recomendable instalar la última versión de VirtualBox y, su correspondiente Extensión Pack. Instala VirtualBox y después la Extensión Pack. Una vez tengas la herramienta de virtualización y su pack instalado, la inicias desde tu menú de aplicaciones.

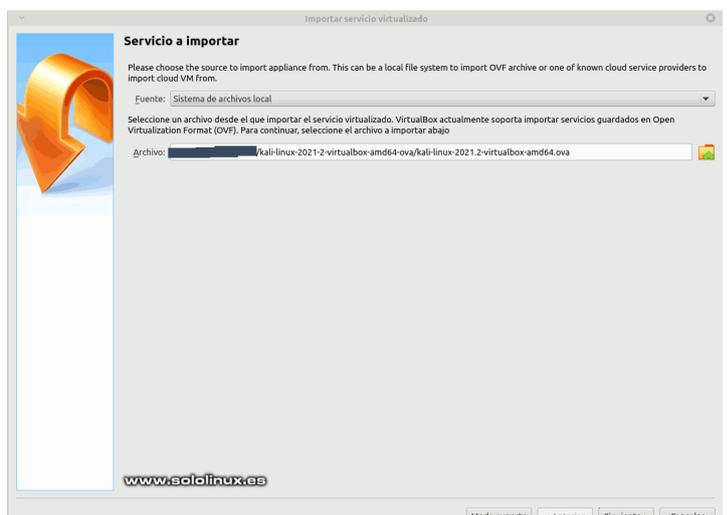
Puedes leer muchos tutoriales sobre como instalar KALI en VirtualBox, sorprendentemente la gran mayoría te recomiendan crear un espacio de disco de 80 o 100 GB. Para mí esto es un grandísimo error, pues hablamos de una distribución linux que es para lo que es, no es

válida para descargar películas piratas, ni grandes archivos.

Además uno de los pocos puntos en los cuales puedes necesitar GB adicionales, es al generar archivos de passwords para **bruteforce**. Precisamente para ello, instalamos el pack de VirtualBox; así puedes guardarlos en un pendrive y liberar trabajo del disco duro anfitrión. Tal como yo lo veo, en las pruebas realizadas nos crea un disco de 10,6 GB. Esto es algo más que suficiente, así que comenzamos haciendo clic en Archivo, en el menú que se abre... en **Importar servicio virtualizado**.



Seleccionas el sistema de archivos local y, el archivo que descargamos anteriormente. Pulsa en siguiente.



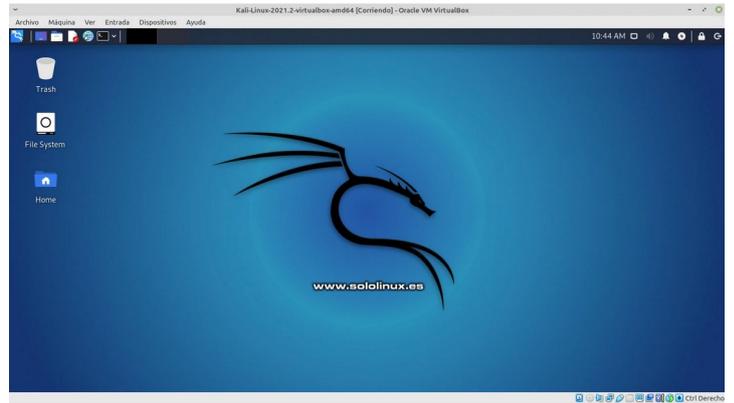
Nos aparece una descripción, en la cual nos indica la configuración por defecto de la máquina virtual, impuesta por el propio **KALI Linux 2021.2**. Revisa los valores y haces clic en **Importar**.



Debes aceptar la licencia de **KALI LINUX 2021.2**. Pulsa en **Acepto**.



Arranca **KALI**.

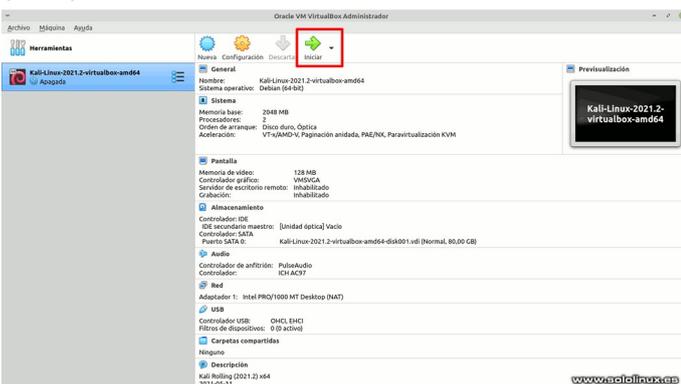


Nota del autor: Te recomiendo que pongas tu nueva instalación de KALI LINUX en español, para ello revisa este anterior artículo.

La importación del servicio puede tardar entre 5 y 15 minutos, todo depende de la potencia de tu máquina anfitrión.



Al concluir el proceso, la herramienta retorna a la página principal. Pulsa en **INICIAR**.



SoloWordPress

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

Instalar Apache JMeter en Linux

Apache JMeter es una herramienta de **código abierto**, específica para ejecutar pruebas de carga. En un artículo anterior, hablamos sobre el **monitor de servidores Netdata**, Apache JMeter es el complemento perfecto para probar el rendimiento de un sitio web, incluyendo diversos niveles de carga.

Principales características de JMeter:

- Interfaz gráfica sencilla e intuitiva.
- Permite realizar pruebas de rendimiento, de carga y de estrés para servidores HTTP, **HTTPS**, SOAP, base de datos, LDAP, SMTP, IMAP, **POP3**, y muchos más.
- Puedes generar planes de pruebas con un simple editor de texto, ya que se guardan en formato XML.
- Multi-hilo y extensible.
- Pruebas automatizadas y manuales.
- Validador de las respuestas del servidor.
- Permite visualizar los resultados mediante tablas o gráficos.

En realidad hemos partido este artículo en dos partes, más que nada por separar el trabajo de instalar con el de hacer uso de la herramienta. En el post de hoy la instalamos, en el próximo aprenderemos a utilizarla.



Instalar Apache JMeter en Linux

Apache JMeter es una aplicación **Java**, por tanto es necesario instalar **jdk**.

En Debian, Ubuntu, Mint y derivados:

```
sudo apt install default-jre-headless default-jdk-headless
```

En Alma Linux, Rocky, CentOS 8, Fedora y derivados:

```
sudo dnf install java-11-openjdk
```

En OpenSUSE, SUSE y derivados:

```
sudo zypper install java-11-openjdk
```

```
root@SoloLinux-demo:~# sudo apt install default-jre-headless default-jdk-headless
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
ca-certificates-java java-common libavahi-client3 libavahi-common-data
libavahi-common3 libcups2 libgraphite2-3 libharfbuzz0b libjpeg-turbo8
libjpeg8 liblcms2-2 libnspr4 libnss3 libpcsclite1 openjdk-11-jdk-headless
openjdk-11-jre-headless
Suggested packages:
default-jre cups-common liblcms2-utils pcsd openjdk-11-demo
openjdk-11-source libnss-mdns fonts-dejavu-extra fonts-ipafont-gothic
fonts-ipafont-mincho fonts-wqy-microhei fonts-wqy-zenhei fonts-indic
The following NEW packages will be installed:
ca-certificates-java default-jdk-headless default-jre-headless java-common
libavahi-client3 libavahi-common-data libavahi-common3 libcups2
libgraphite2-3 libharfbuzz0b libjpeg-turbo8 libjpeg8 liblcms2-2 libnspr4
libnss3 libpcsclite1 openjdk-11-jdk-headless openjdk-11-jre-headless
0 upgraded, 18 newly installed, 0 to remove and 0 not upgraded.
Need to get 263 MB of archives.
After this operation, 412 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

www.sololinux.es



Verificamos la versión instalada.

```
java -version
```

```
root@SoloLinux-demo:~# java -version
openjdk version "11.0.11" 2021-04-20
OpenJDK Runtime Environment (build 11.0.11+9-Ubuntu-0ubuntu2.20.04)
OpenJDK 64-Bit Server VM (build 11.0.11+9-Ubuntu-0ubuntu2.20.04, mixed
mode, sharing)
root@SoloLinux-demo:~#
```

Ahora descargamos con el **comando wget** la última versión estable de la herramienta, a día de hoy la 5.4.1. De todas formas te recomiendo revisar su **página de descargas oficial**, de forma que puedas obtener la versión más actual.

```
wget https://apache.paket.ua//jmeter/binaries/apache-jmeter-5.4.1.tgz
```

```
root@SoloLinux-demo:~# wget
https://apache.paket.ua//jmeter/binaries/apache-jmeter-5.4.1.tgz
--2021-08-14 13:14:05--
https://apache.paket.ua//jmeter/binaries/apache-jmeter-5.4.1.tgz
Resolving apache.paket.ua (apache.paket.ua)... 213.231.5.55
Connecting to apache.paket.ua (apache.paket.ua)|213.231.5.55|:443...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 70704620 (67M) [application/x-gzip]
Saving to: 'apache-jmeter-5.4.1.tgz'
apache-jmeter-5.4.1.tgz 100%
[=====]
67.43M 1.36MB/s in 2m 22s
2021-08-14 13:16:27 (485 KB/s) - 'apache-jmeter-5.4.1.tgz' saved
[70704620/70704620]
root@SoloLinux-demo:~#
```

Apache JMeter 5.4.1 (Requiere Java 8+)

Binaries

[apache-jmeter-5.4.1.tgz sha512 pgp](#)
[apache-jmeter-5.4.1.zip sha512 pgp](#)

Source

[apache-jmeter-5.4.1_src.tgz sha512 pgp](#)
[apache-jmeter-5.4.1_src.zip sha512 pgp](#)

www.sololinux.es

Una vez concluya la descarga, extraemos el archivo.

```
tar xvf apache-jmeter-5.*.tgz
```

Ahora movemos el directorio a /opt.

```
sudo mv apache-jmeter-5.4.1 /opt/apache-jmeter
```

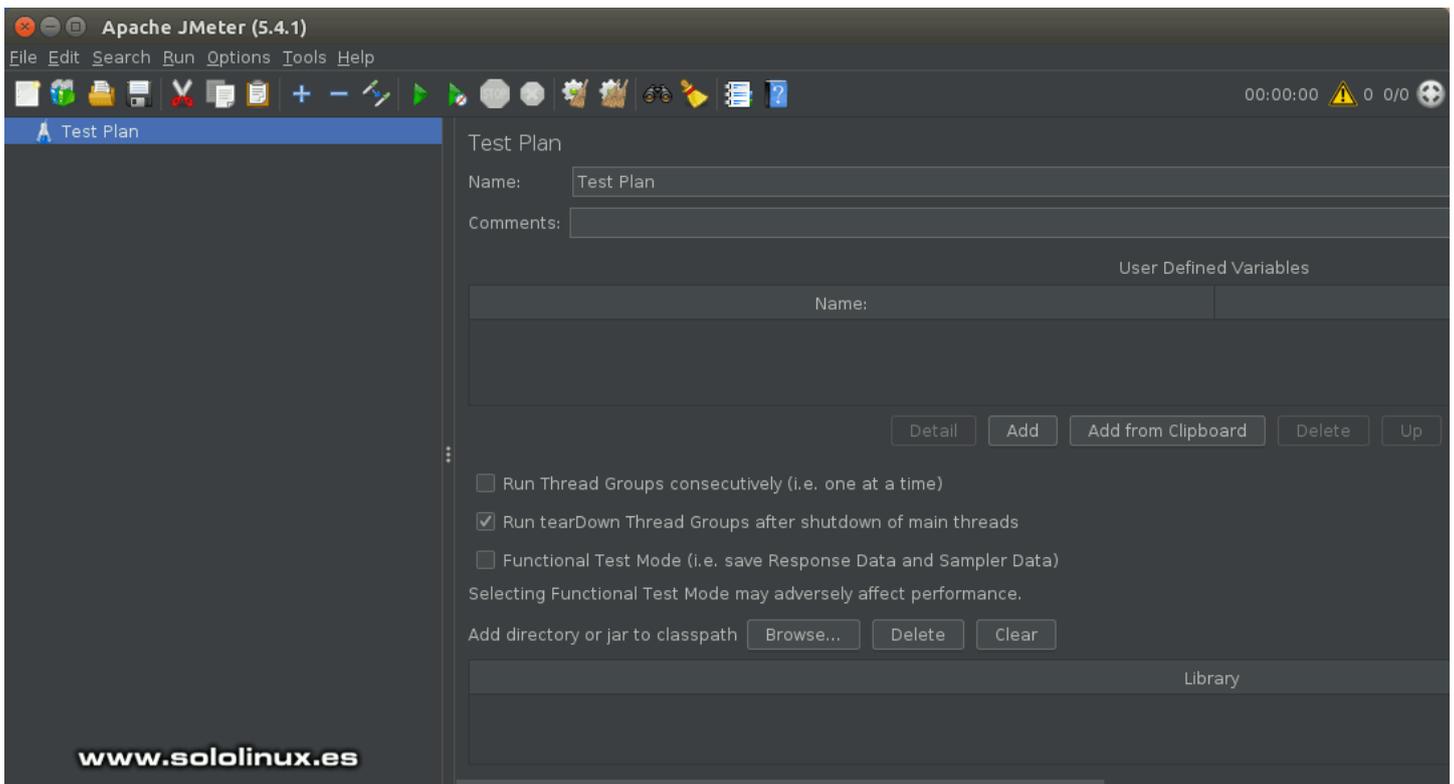
Listo, ya podemos lanzar la herramienta en modo GUI con el siguiente comando. El modo interfaz gráfica no se recomienda para pruebas de carga reales.

```
sudo /opt/apache-jmeter/bin/jmeter.sh
```

```
root@SoloLinux-demo:~# sudo /opt/apache-jmeter/bin/jmeter.sh
```

```
=====
Don't use GUI mode for load testing !, only for Test creation and Test debugging.
For load testing, use CLI Mode (was NON GUI):
  jmeter -n -t [jmx file] -l [results file] -e -o [Path to web report folder]
& increase Java Heap to meet your test requirements:
  Modify current env variable HEAP="-Xms1g -Xmx1g -XX:MaxMetaspaceSize=256m" in the jmeter batch file
Check : https://jmeter.apache.org/usermanual/best-practices.html
=====
```

Aparece la interfaz gráfica de JMeter. En un próximo artículo **aprenderemos a usar la aplicación.**



www.sololinux.es

Debian 11 Bullseye - Listo para su descarga

Después de dos años de trabajo, Debian acaba de lanzar de forma oficial su última versión estable, hablamos de **Debian 11 Bullseye**. Este nuevo lanzamiento viene con muchos componentes actualizados y, nuevas tecnologías GNU / Linux imprescindibles hoy en día.

Impulsado por el **kernel 5.10 LTS** series, mantendrá su soporte activo hasta diciembre de 2026. Recuerda que la mejora en soporte de hardware de esta versión es brutal. Además, **Bullseye** se convierte en la primera versión que ofrece un kernel con soporte para el sistema de archivos **exFAT**.



Debian 11 Bullseye – Listo para su descarga

Todos los entornos de escritorio compatibles, han sido actualizados a las versiones más recientes, incluyendo **GNOME 3.38**, **KDE Plasma 5.20**, **Xfce 4.16**, **LXQt 0.16**, **LXDE 11** y **MATE 1.24**. **Debian 11 Linux**, usa el compilador del sistema **GCC 10.2**, así como **GNU C Library 2.31**, **LLVM 11.0.1** y, otras tecnologías actuales demandadas por los usuarios.

Otras interesantes y nuevas características, incluyen la impresión extendida sin controladores a **dispositivos USB** con el nuevo paquete **ipp-usb**, el escaneo sin controladores con el nuevo paquete **sane-airscan**, así como un mejorado **comando open** que permite abrir archivos automáticamente, desde la línea de comandos para una determinada aplicación (GUI o CLI).

Actualizar Debian 10 a Debian 11

Para ofrecer una jerarquía de control de recursos unificada, **systemd** usa de forma predeterminada los **grupos de control v2 (cgroupv2)**, pero si es necesario puedes volver a habilitar los obsoletos **cgroups** heredados modificando unos parámetros en el kernel.

El nuevo **Bullseye**, incluye un conjunto de herramientas específicas para la investigación del **virus COVID-19** en el ámbito de secuencia, para combatir la pandemia. También se han actualizado otros muchos paquetes, la mayoría habituales en nuestro trabajo diario entre los que podemos destacar los siguientes:

- **Apache 2.4.48**
- **BIND DNS Server 9.16**
- **Calligra 3.2**
- **Cryptsetup 2.3**
- **Emacs 27.1**
- **GIMP 2.10.22**
- **Colección de compiladores GNU 10.2**
- **GnuPG 2.2.20**
- **Inkscape 1.0.2**
- **LibreOffice 7.0**
- **Kernel de Linux serie 5.10**
- **MariaDB 10.5**
- **OpenSSH 8.4p1**
- **Perl 5.32**
- **PHP 7.4**
- **PostgreSQL 13**
- **Python 3, 3.9.1**
- **Óxido 1.48**
- **Samba 4.13**
- **Vim 8.2**

Si no tienes **Debian 10** instalado, o simplemente quieres hacer una instalación limpia, puedes descargar Bullseye desde su página oficial (se recomienda **Latest stable Debian release**).

- [Descargar Debian 11.o.o](#)

Si prefieres descargar las **live de Debian 11** por torrent:

- [Debian-live-11.0.0-amd64-cinnamon.iso.torrent](#)
- [Debian-live-11.0.0-amd64-gnome.iso.torrent](#)
- [Debian-live-11.0.0-amd64-kde.iso.torrent](#)
- [Debian-live-11.0.0-amd64-xfce.iso.torrent](#)
- [Debian-live-11.0.0-amd64-lxde.iso.torrent](#)
- [Debian-live-11.0.0-amd64-lxqt.iso.torrent](#)
- [Debian-live-11.0.0-amd64-mate.iso.torrent](#)
- [Debian-live-11.0.0-amd64-standard.iso.torrent](#)



Si por el contrario eres usuario de **Debian 10** y, prefieres actualizar de Debian 10 a Debian 11, sigue los pasos indicados en el siguiente artículo de **SoloLinux**. No olvides hacer una copia de seguridad antes de lanzar el proceso.

- **Actualizar Debian 10 a Debian 11**



Como instalar Debian 11 de forma sencilla

En el artículo de hoy vamos a tratar las bondades y propiedades, que nos presenta Debian Bullseye (ya las comentamos en el post anterior). Nos vamos a limitar a instalar Debian a través de imágenes, de manera amena y sencilla. Si tú eres un usuario experto o avanzado, esta lectura no es para ti.

En nuestro caso nos hemos decantado por la versión **XFCE**, pues según mi opinión personal, es el escritorio perfecto para realizar labores diarias ofreciendo un óptimo rendimiento. Debian Live 11 amd64 Xfce ha sido la opción elegida, aunque si eres de los que prefiere **versiones non-free**, también está disponible. Dejo los enlaces de descarga por **torrent**.

- [Descargar Debian Live 11 amd64 Xfce ISO](#)
- [Descargar Debian Live 11 amd64 Xfce NonFree ISO](#)

Como instalar Debian 11 de forma sencilla

Al iniciar la máquina con la ISO de Debian Bullseye, nos aparece una pantalla como la siguiente imagen. Si seleccionamos alguna de las dos primeras opciones, iniciara la live de Debian en modo escritorio XFCE. En este escritorio tienes un icono para instalar Debian.

La instalación mediante el icono del escritorio es diferente al tradicional, por ejemplo llega un momento que puedes elegir otros entornos de escritorio, o un modo servidor. Por contra... no habilita el **usuario root** y eso es algo que no me gusta. Yo prefiero la tercera opción que es la tradicional, pero en modo gráfico interactivo. La seleccionas y pulsas la **tecla enter**.

Main Menu

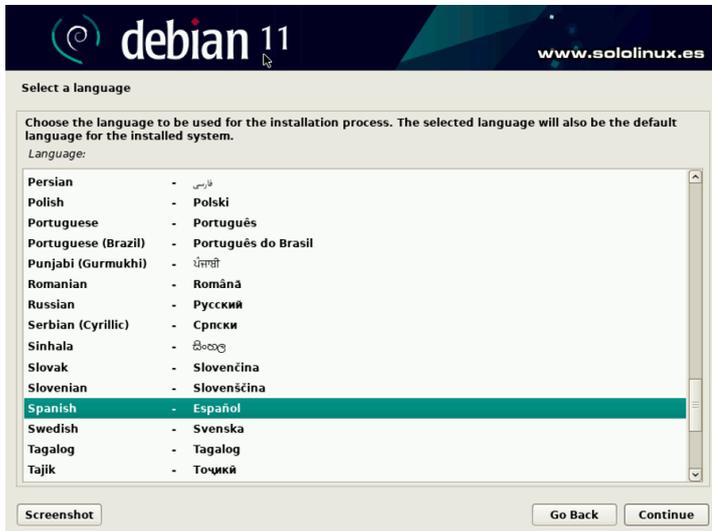
```
Debian GNU/Linux Live (kernel 5.10.0-8-amd64)
Debian Live with Localisation Support >
Graphical Debian Installer
Debian Installer
Debian Installer with Speech Synthesis
```



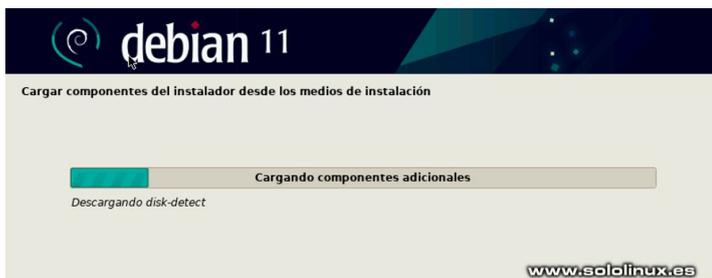
www.sololinux.es

La instalación comienza inmediatamente. En las tres pantallas siguientes debemos elegir nuestro idioma, nuestra región y el tipo de teclado. En nuestro caso...

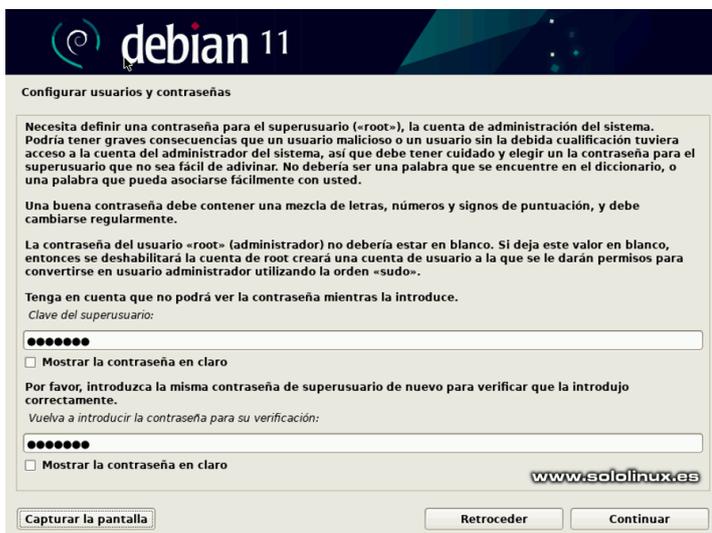
- Spanish – Español
- España
- Español



Comienza la carga del instalador, para continuar con la configuración de componentes básicos como la red.



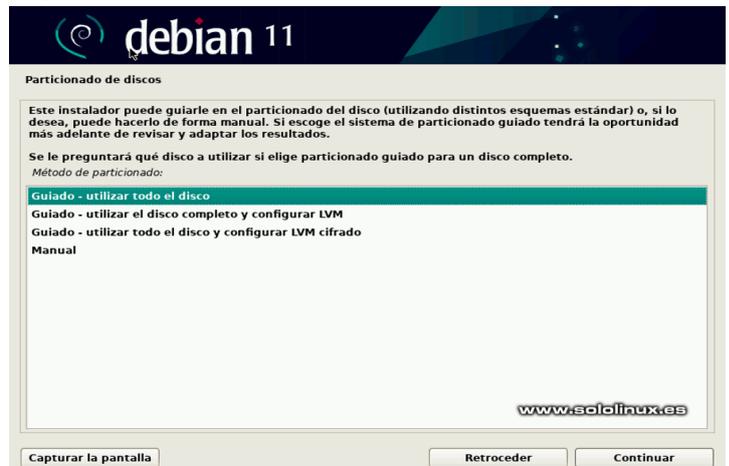
A continuación solicita una contraseña para el superusuario root. Una vez lo tengas, nos dice que también debemos crear un usuario. Nombre completo, nombre de usuario y password.



Tomando como referencia los datos de ubicación que proporcionamos anteriormente (**España**), ahora nos pregunta la zona horaria de nuestra ubicación.



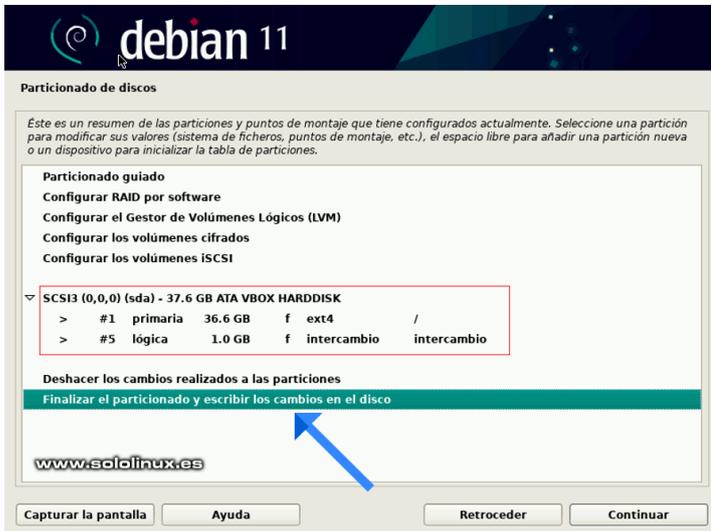
Cargan los componentes de manipulación de discos. Vemos una pantalla donde nos aparecen cuatro opciones de particionado, las tres primeras son guiadas, la cuarta es en modo manual solo apto para usuarios avanzados. En nuestro caso seleccionamos la primera opción.



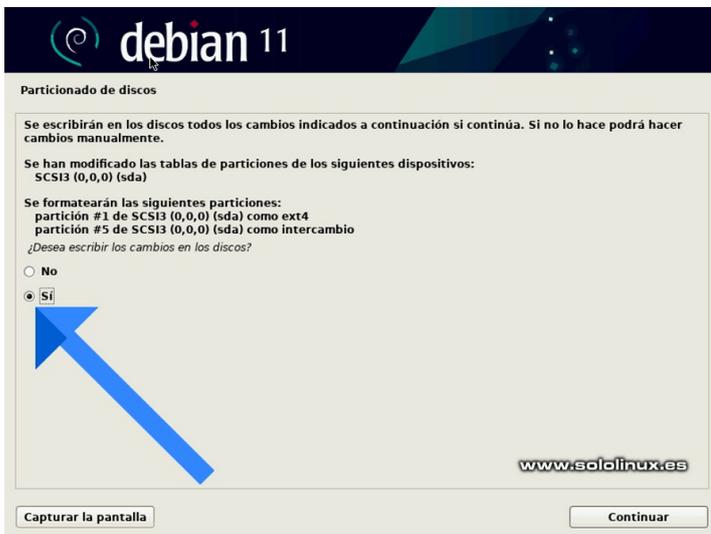
Marcamos el disco que vamos a utilizar. En la siguiente pantalla, te pregunta si quieres tener todos los archivos en una partición, si quieres crear una partición /home, o si por otro lado te decantas por separar /home, /var y /tmp. Separar la /home es buena idea, pero en nuestro caso usamos la opción recomendada para novatos.



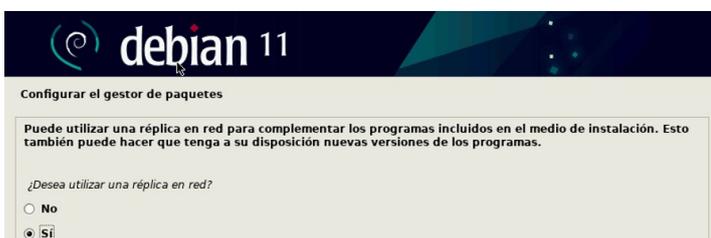
Seguro que en el paso anterior has echado de menos la partición de intercambio (**swap**), no te preocupes, independientemente de la opción que hayas elegido, la **partición swap** se genera de forma automática. Antes de pulsar en continuar, asegúrate de tener marcada la línea que indica la flecha de la imagen de ejemplo.



La siguiente pantalla es importante. Debes marcar la opción, donde permites escribir los cambios en los discos.



Comienza la instalación. Tras unos segundos, el instalador nos pregunta si queremos usar una réplica en red. No es obligatorio, pero al seleccionar que sí que quieres, se ofrece la opción de seleccionar tu región más cercana con paquetes del instalador. Te recomiendo que sigas el proceso, la velocidad de las descargas aumentarán de forma considerable.



También aparece la opción de configurar un **proxy**, si no es tu caso pulsa en continuar para omitir el proceso. Una vez termine el proceso de instalación, Debian 11 nos pregunta si queremos instalar el cargador de arranque **GRUB**, el disco duro. Selecciona «Sí» y pulsa en continuar. En la nueva pantalla que te aparece, seleccionas el disco donde se instalara el GRUB (el mismo donde instalas Debian), y pulsas de nuevo en continuar.



Una vez termine la instalación del GRUB, el sistema te pide reiniciar la máquina. Cuando pulsas en continuar para reiniciar, antes de la operación y de forma automática se borran archivos de instalación que ya no se van a utilizar.



Como instalar Debian 11 de forma sencilla

Una vez termine la instalación del GRUB, el sistema te pide reiniciar la máquina. Cuando pulsas en continuar para reiniciar, antes de la operación y de forma automática se borran archivos de instalación que ya no se van a utilizar.



Al iniciar el sistema vemos el GRUB.



Después de insertar nuestro usuario y contraseña, aparece nuestro flamante **Debian Bullseye** que tiene un renovado fondo de escritorio, realmente espectacular.



Lo primero que debes hacer es actualizar el sistema, pero ojo que esta versión no acepta sudo por defecto, por tanto debes operar tal como te indico. En un próximo artículo veremos como solucionar este problema.

```
su
apt update
apt full-upgrade
```

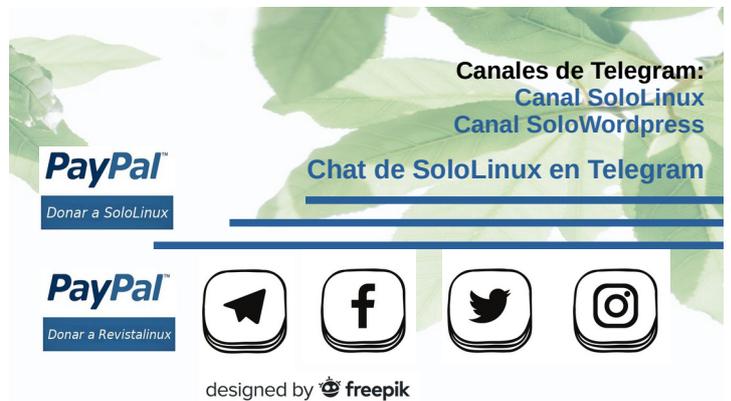
```
sergio@sololinux:~$ su
Contraseña:
root@sololinux:/home/sergio# apt update
Obj:1 http://deb.debian.org/debian bullseye InRelease
Obj:2 http://deb.debian.org/debian bullseye-updates InRelease
Obj:3 http://security.debian.org/debian-security bullseye-security InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se pueden actualizar 72 paquetes. Ejecute «apt list --upgradable» para verlos.
root@sololinux:/home/sergio# apt full-upgrade
```

Recuerda que puedes verificar la versión instalada con el comando **cat**.

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

Salida de **Debian 11 Bullseye**.

```
root@sololinux:/home/sergio# cat /etc/*release
PRETTY_NAME="Debian GNU/Linux 11 (bullseye)"
NAME="Debian GNU/Linux"
VERSION_ID="11"
VERSION="11 (bullseye)"
VERSION_CODENAME=bullseye
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
root@sololinux:/home/sergio#
```



SoloWordpress

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

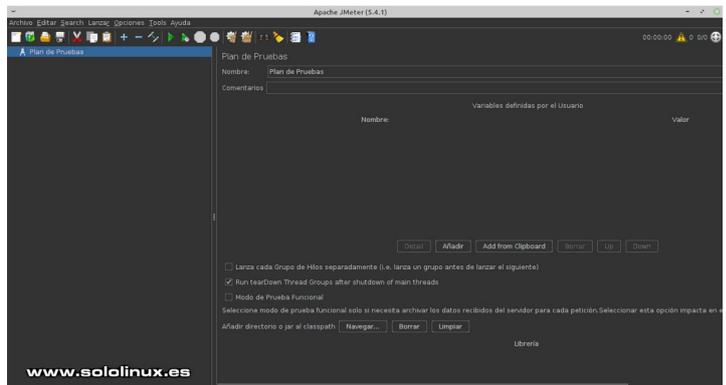


Probar el rendimiento de un servidor con JMeter



```
sergio@sololinux:~$ sudo /opt/apache-jmeter/bin/jmeter.sh
ago. 18, 2021 2:30:01 P. M. java.util.prefs.FileSystemPreferences$1
run
INFORMACIÓN: Created user preferences directory.
=====
Don't use GUI mode for load testing !, only for Test creation and Test
debugging.
For load testing, use CLI Mode (was NON GUI):
jmeter -n -t [jmx file] -l [results file] -e -o [Path to web report
folder]
& increase Java Heap to meet your test requirements:
  Modify current env variable HEAP="-Xms1g -Xmx1g -
  XX:MaxMetaspaceSize=256m" in the jmeter batch file
Check : https://jmeter.apache.org/usermanual/best-practices.html
=====
```

Aparece la interfaz gráfica de la herramienta.



Instalar Apache JMeter en Linux, ese es el título de un anterior artículo, en el cual explicábamos como instalar una herramienta específica para lanzar pruebas de carga, Apache JMeter. Como ya comentamos en aquel post, tan solo era la primera parte de dos, primero instalamos y posteriormente configuramos.

El lanzamiento de Debian 11, nos obligó a frenar la salida del segundo artículo; bueno... ya hemos respirado profundamente dos o tres días, se nos ha pasado el sofoco y la emoción, así que lo prometido es deuda. Hoy vamos a ver con un simple ejemplo, como configurar JMeter para ejecutar estrés a un servidor de principio a fin.

Puede parecer un poco liosa, pero no es tan difícil si sigues los pasos que te indico. Lo único que me gustaría recomendarte es prudencia con los valores, en nuestro caso los usamos bajos a modo de ejemplo. También debes tener presente que con esta herramienta, puedes llegar a colapsar un servidor o VPS, haz pruebas solamente en tus propias máquinas, quedan restos y te van a pillar si lo usas como forma de ataque. Su misión es la siguiente:

- Verificar los criterios de rendimiento especificados.
- Validar la escalabilidad, fiabilidad y, uso de los recursos del sistema.
- Confrontar dos o más sistemas para comparar rendimientos.
- Identificar los errores del sistema con mucha carga de trabajo.

Probar el rendimiento de un servidor con JMeter

Si seguiste los [pasos explicados en el post anterior](#), ya tienes JMeter instalado en tu sistema Linux. Abrimos la herramienta con el siguiente comando.

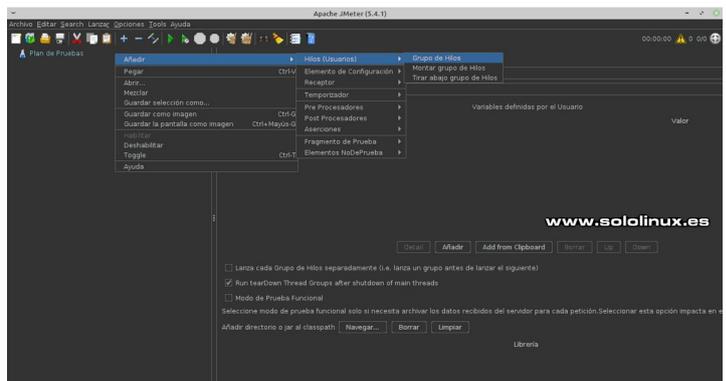
```
sudo /opt/apache-jmeter/bin/jmeter.sh
```

JMeter ya viene con un plan creado, pero sin datos ni valores, de todas formas lo aprovechamos. Los pasos a seguir para configurar JMeter correctamente, son...

- **Grupo de Hilos (subprocesos).**
- **Valores predeterminados de las solicitudes HTTP.**
- **Administrador de cookies.**
- **Peticiones HTTP.**
- **Escuchas HTTP.**

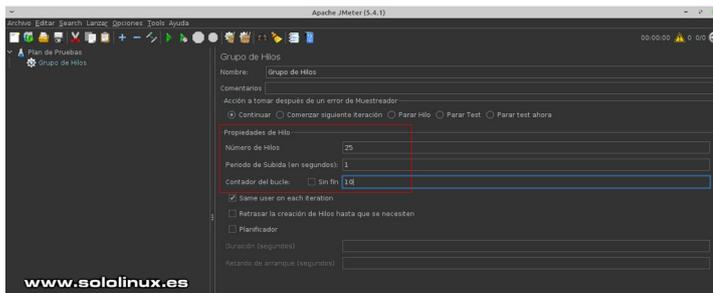
Grupo de Hilos

Sigue todo lo que te indico paso a paso. Posiciona el puntero del ratón sobre Plan de pruebas, pulsa el botón derecho del morse para abrir el menú. Selecciona tal como la siguiente imagen: **Añadir / Hilos (usuarios) / Grupo de Hilos**.



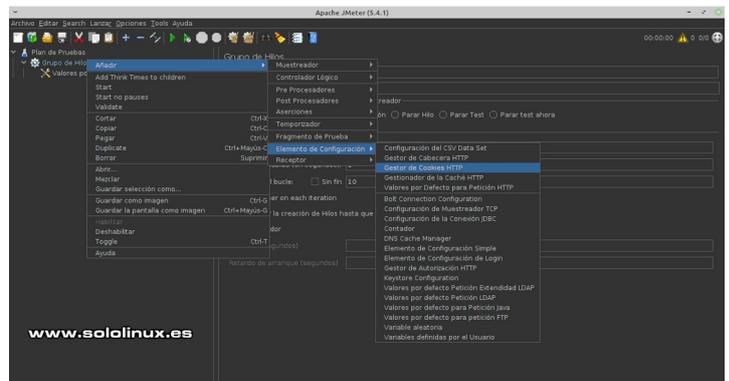
Aquí debes configurar las propiedades de los hilos. Ten cuidado con los valores que insertas, no vayas a tirar tu vps o servidor. En nuestro caso usamos los valores 25 / 1 / 10.

- **Número de hilos:** Número de usuarios que quieres simular. Cada hilo representa un usuario simulado.
- **Periodo de subida:** Tiempo entre el inicio del primer hilo y el último hilo.
- **Contador del bucle:** Cuantas peticiones debe enviar cada usuario simulado.



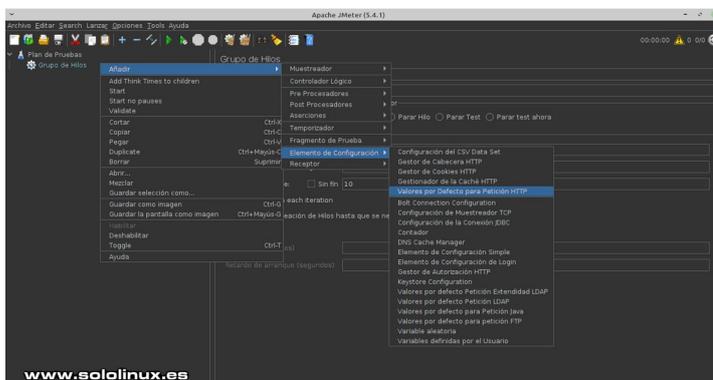
Administrador de cookies

Si tu sitio web genera cookies para los visitantes, agregamos su compatibilidad posicionando el ratón sobre «Grupo de Hilos», ahora pulsa el botón derecho del mouse y la secuencia: **Añadir / Elementos de configuración / Gestor de Cookies HTTP**.



Valores predeterminados de las solicitudes HTTP

Ahora posiciona el ratón sobre «Grupo de Hilos», pulsa el botón derecho del mouse y la secuencia: **Añadir / Elementos de configuración / Valores por defecto para peticiones HTTP**.



En valores por defecto asegúrate que estas en la opción Basic. Ajusta los valores indicados.

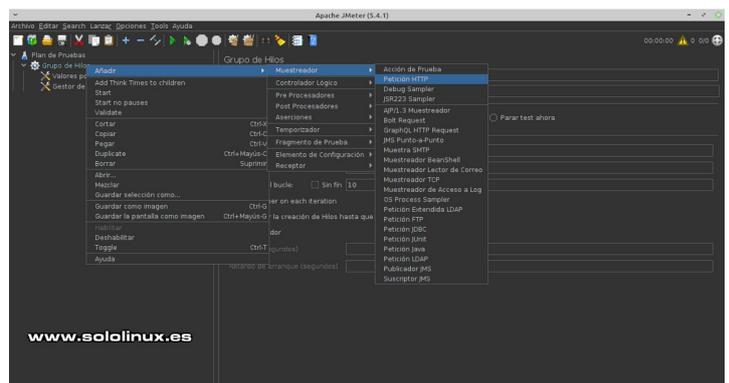
- **Protocolo:** https
- **Nombre o IP:** Nombre del host que resuelva, Dominio o dirección IP.
- **Puerto:** 443 (por defecto de https, si es necesario lo cambias).

No añadas ningún valor, déjalo por defecto tal como la siguiente imagen de ejemplo.

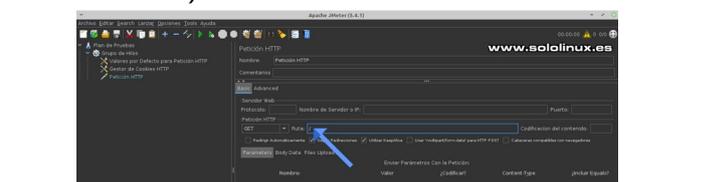
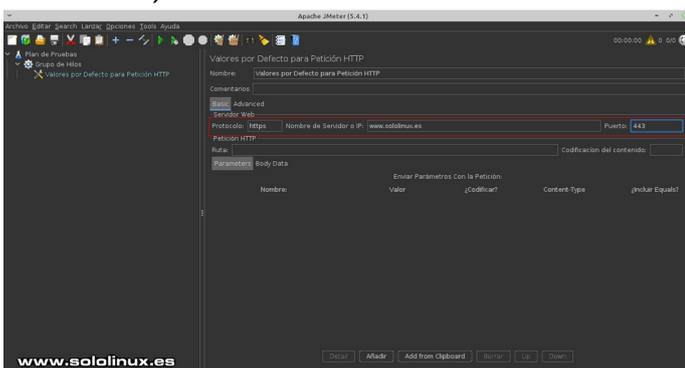


Peticiones HTTP

Posiciona el puntero del mouse sobre «Grupo de Hilos», ahora pulsa el botón derecho del ratón y la secuencia: **Añadir / Muestreador / Petición HTTP**.

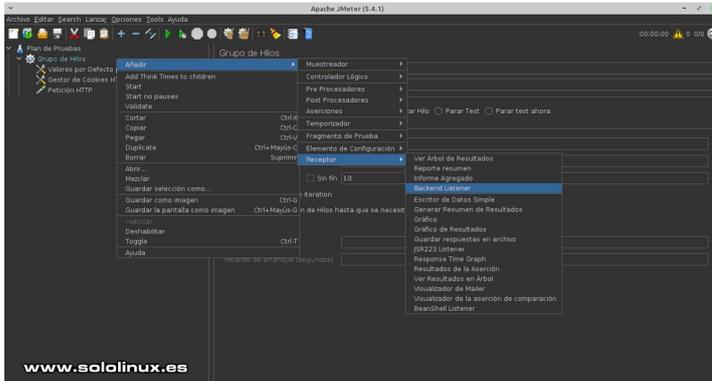


En nuestro artículo de ejemplo, ya insertamos en valores predeterminados el protocolo, dirección y puerto, por tanto lo dejamos en blanco. Lo que si debes configurar es la ruta donde lanzar las peticiones, si lo dejas en </>, atacara sobre el index o página principal (puedes cambiar la ruta).

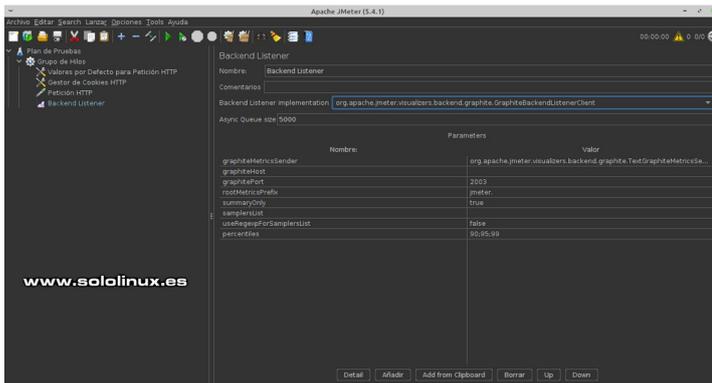


Escuchas HTTP

Para conocer los resultados necesitamos escuchar las respuestas, agregamos lo que se conoce como Receptor. De nuevo colocamos el ratón sobre «Grupo de Hilos», pulsa el botón derecho del mouse y la secuencia: **Añadir / Receptor / Backend Listener**.

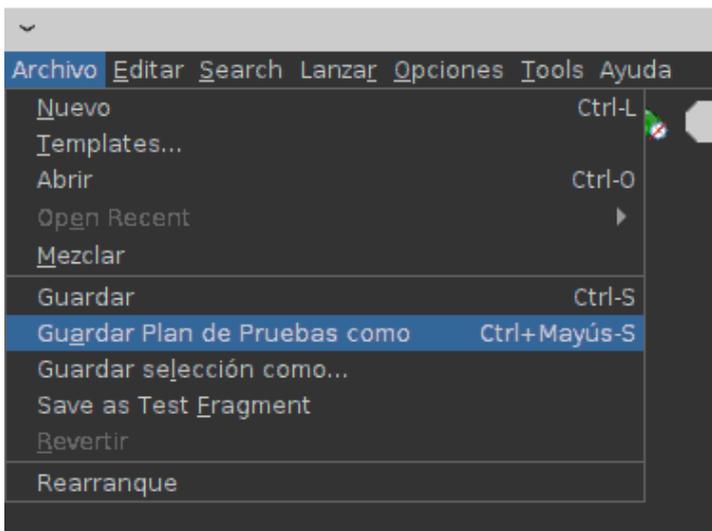


No es necesario modificar nada, déjalo como esta por defecto.

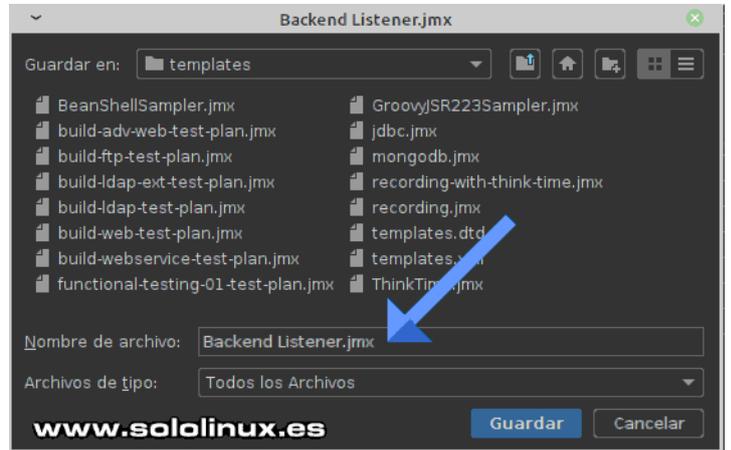


Guardar Plan

Para guardar el plan, pulsa **Archivo** en el menú superior y, en **Guardar Plan de Pruebas como**.



Te recomiendo que guardes los planes creados o modificados en Templates, así que busca la siguiente ruta: **/opt/apache-jmeter/bin/templates**. Cambia el nombre por defecto, en nuestro caso borramos el Backend y lo dejamos como «**Listener.jmx**». Pulsa en guardar.



Ejecutar prueba de carga

Creamos una carpeta para guardar los informes generados.

```
sudo mkdir /opt/apache-jmeter/reports/
```

Los propios desarrolladores desaconsejan provocar estrés a un servidor desde la propia GUI, debes usar la terminal de tu linux. Para lanzar la prueba según los criterios del artículo, ejecutamos el siguiente comando.

```
sudo /opt/apache-jmeter/bin/jmeter.sh -n -t /opt/apache-jmeter/bin/templates/Listener.jmx -l /opt/apache-jmeter/reports/testresult.jtl -e -o /opt/apache-jmeter/reports/
```

Explicamos los indicadores.

- **-n**: Modo sin interfaz gráfica.
- **-t**: Ruta del archivo del plan.
- **-l**: Ruta del archivo de registro.
- **-e -o**: Genera un archivo en formato HTML.

Ejemplo de uso.

```
sergio@sololinux:~$ sudo /opt/apache-jmeter/bin/jmeter.sh -n -t /opt/apache-jmeter/bin/templates/Listener.jmx -l /opt/apache-jmeter/reports/testresult.jtl -e -o /opt/apache-jmeter/reports/ [sudo] contraseña para sergio: Creating summariser <summary> Created the tree successfully using /opt/apache-jmeter/bin/templates/Listener.jmx Starting standalone test @ Wed Aug 18 15:23:42 EEST 2021 (1629289422555) Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445 Warning: Nashorn engine is planned to be removed from a future JDK release summary = 250 in 00:00:11 = 23,4/s Avg: 943 Min: 314 Max: 5011 Err: 0 (0,00%) Tidying up ... @ Wed Aug 18 15:23:55 EEST 2021 (1629289435266) ... end of run sergio@sololinux:~$ cd /opt/apache-jmeter/reports/ sergio@sololinux:/opt/apache-jmeter/reports$ dir content index.html sbadmin2-1.0.7 statistics.json testresult.jtl sergio@sololinux:/opt/apache-jmeter/reports$
```

Dependiendo de las solicitudes de la prueba, el proceso tardará más o menos. Una vez concluya el proceso, abres tu navegador web favorito e insertas la siguiente URL.

/opt/apache-jmeter/reports/index.html

Imagen de ejemplo.

Apache JMeter Dashboard

Dashboard

Charts

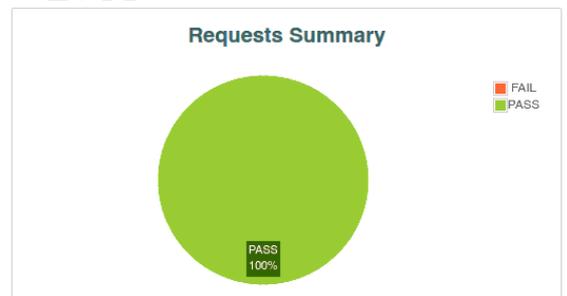
Customs Graphs

Test and Report information	
Source file	"testresult.jtl"
Start Time	"18/8/21 15:23"
End Time	"18/8/21 15:23"
Filter for display	""

www.sololinux.es

APDEX (Application Performance Index)

Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
0.668	500 ms	1 sec 500 ms	Total
0.668	500 ms	1 sec 500 ms	Petición HTTP



www.sololinux.es

Canales de Telegram:
 Canal SoloLinux
 Canal SoloWordpress

Chat de SoloLinux en Telegram

PayPal
 Donar a SoloLinux

PayPal
 Donar a Revistalinux

designed by freepik

SoloWordpress

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



Error en Debian 11 al ejecutar sudo

Al igual que en la versión anterior, en **Debian 11** también existen algunas versiones donde al ejecutar **sudo** nos aparece un error. Lo llamamos error por denominarlo de alguna forma reconocible, porque en realidad no es un error, es una medida de seguridad bajo mi punto de vista un tanto desproporcionada. Aparece un mensaje similar al siguiente.

User is not in the sudoers file. This incident will be reported.

Por muy extraño que parezca, **sudo** no funciona de forma predeterminada en algunas instalaciones nuevas de **Debian 11**. La razón es fácil de explicar, el nombre de usuario no se agrega automáticamente al grupo sudo, nos dicen que por medidas de seguridad. Este efecto también se produce al agregar un nuevo usuario, si no lo añades intencionadamente al grupo sudo.

Error en Debian 11 al ejecutar sudo

Por suerte la solución al problema es bastante sencilla, aunque de todas formas si prefieres mantenerlo como está ahora por el motivo que sea, puedes acceder con permisos con este comando.

```
su -
sergio@sololinux:~$ su -
Contraseña:
root@sololinux:~#
```

Esto anterior puede ser un engorro, ya que por ejemplo si quieres actualizar el sistema, o ejecutar cualquier comando que requiera permisos, tienes que ejecutar dos comandos.

```
sergio@sololinux:~$ su -
Contraseña:
root@sololinux:~# apt update
Obj:1 http://deb.debian.org/debian bullseye InRelease
Des:2 http://deb.debian.org/debian bullseye-updates InRelease [36,8 kB]
Obj:3 http://security.debian.org/debian-security bullseye-security InRelease
Descargados 36,8 kB en 2s (19,0 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Todos los paquetes están actualizados.
root@sololinux:~#
```

Lo más cómodo es operar de la manera tradicional, y para ello lo mejor es agregar nuestro **usuario al grupo sudo** (o varios usuarios). Antes de hacer nada, verificamos que no exista el usuario en el grupo sudo (desde la sesión del usuario).

```
groups
```

Como puedes ver en el siguiente ejemplo, el **usuario sergio** no pertenece al **grupo sudo**.

```
sergio@sololinux:~$ groups
sergio cdrom floppy audio dip video plugdev netdev bluetooth lpadmin scanner
sergio@sololinux:~$
```

Está claro que tenemos que añadir el **usuario sergio** al grupo. Sigue los pasos que te indico, con el nombre de tu usuario. Comenzamos obteniendo los permisos requeridos.

```
su -
```

Ejecuta este comando.

```
# Inserta tu usuario
usermod -aG sudo sergio
```

Bien, ya lo tienes, pero tranquilo que aún no está operativo. Es necesario reiniciar la sesión de usuario o el propio sistema. En nuestro caso reiniciamos el sistema.

```
reboot
```

Ejemplo de secuencia completa.

```
sergio@sololinux:~$ su -
Contraseña:
root@sololinux:~# usermod -aG sudo sergio
root@sololinux:~#
root@sololinux:~# reboot
.....
```

Una vez reinicie el sistema, verificamos que nuestro usuario pertenece al **grupo sudo**.

```
groups
```

En la salida de ejemplo, vemos que sergio está incluido en **sudo**.

```
sergio@sololinux:~$ groups
sergio cdrom floppy sudo audio dip video plugdev netdev bluetooth lpadmin scanner
sergio@sololinux:~$
```

Para terminar intentamos actualizar de la forma tradicional.

```
sudo apt update
```

Como puedes ver en la imagen de ejemplo, ahora sí que funciona tal como esperamos. Bravo.

```
sergio@sololinux:~$ sudo apt update
[sudo] password for sergio:
Obj:1 http://security.debian.org/debian-security bullseye-security InRelease
Obj:2 http://deb.debian.org/debian bullseye InRelease
Obj:3 http://deb.debian.org/debian bullseye-updates InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Todos los paquetes están actualizados.
sergio@sololinux:~$
```



Novedad: LibreOffice 7.2 listo para su descarga

Poco queda por decir del fabuloso **paquete ofimático LibreOffice**, excepto que acaba de ver la luz una nueva y novedosa versión, que **Henry G.R.** (de **SoloWordpress**) ha estado analizando para explicarnos a todos sus principales novedades y características mejoradas. Comenzamos el artículo.

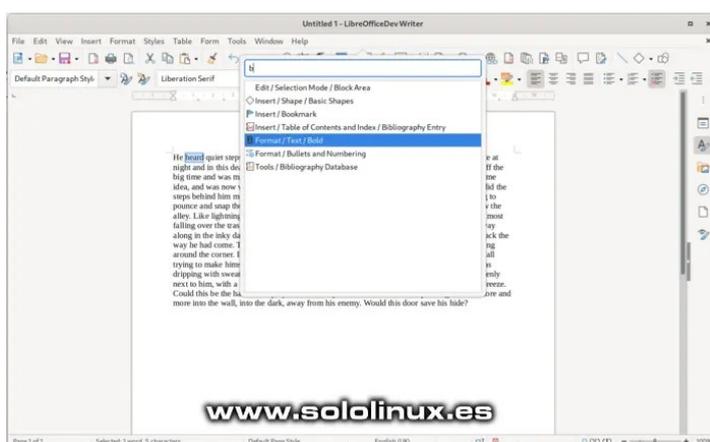
Ya está disponible para su descarga, la nueva y mejorada versión del paquete ofimático LibreOffice, la versión 7.2 llega cargada de mejoras y modificaciones, algunas no muy visibles pero no por ello menos importantes. También veremos como instalar la última versión en tu **distribución Linux** preferida.

Novedad LibreOffice 7.2 listo para su descarga

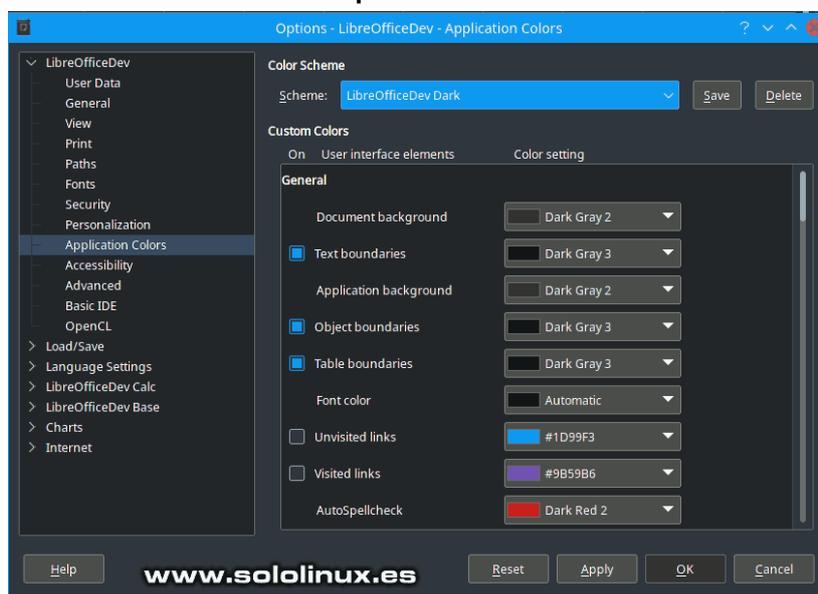
En general, se ha mejorado en velocidad del tratamiento (por parte de cada uno de los componentes de la suite) de los archivos y, entre estas mejoras destaca el tratamiento del módulo **«Calc»** de los archivos grandes en formato **«.xlsx»**. Muchos otros cambios no tienen que ver con la interoperabilidad, sino con mejoras en la productividad y la calidad de los documentos. Entre los que cabe destacar:

El nuevo **comando HUD** (“Heads-up display”).

Esta herramienta está disponible en cada uno de los módulos de LibreOffice 7.2 y, te permite buscar y ejecutar programas, de forma muy parecida a como lo hace el escritorio Unity de Ubuntu. Para acceder a esta herramienta en LibreOffice 7.2, debes ir al menú **«Ayuda»** → **«Buscar Comandos»** o usar la combinación de teclado: **«Mayus+Esc»**



Otra incorporación muy solicitada por los usuarios, es el «modo oscuro» que está tan de moda hoy en día, lo puedes seleccionar a través del menú **«Herramientas»** → **«Opciones»** → **«LibreOffice»** → **«Colores de la Aplicación»**



Claro que si eres amante de la terminal, deberás seguir otros pasos:

- **Abre una terminal (usualmente, Ctrl+Alt+T)**
- **Cambia al directorio donde quieres realizar la descarga (\$ cd ~/Descargas)**
- **Para sistemas basados en paquetes .deb (Debian, Ubuntu, Mint, etc.).**

```
wget https://es.libreoffice.org/colabora/dl/deb-x86_64/7.2.0/es/LibreOffice_7.2.0_Linux_x86-64_deb.tar.gz
```

- **Para sistemas basados en paquetes .rpm: (Fedora, Mandriva, OpenSuse, etc.).**

```
wget https://es.libreoffice.org/colabora/dl/rpm-x86_64/7.2.0/es/LibreOffice_7.2.0_Linux_x86-64_rpm.tar.gz
```

- **Una vez concluya la descarga seleccionada, desempaquetamos el archivo comprimido.**

```
tar -zxvf LibreOffice_7.2.0_Linux_x86-64*.tar.gz
```

Esto generará un directorio con el nombre de: «**LibreOffice_7.2.0_Linux_x86-64_deb**», que a su vez contiene los directorios «**DEBS**» o «**RPMS**» y «**readmes**», conteniendo respectivamente los paquetes de instalación (según tu sistema esté basado en .deb o en .rpm) y la documentación.

La instalación se realiza de forma completa, si intentas instalar los paquetes uno por uno, puedes encontrarte con problemas de dependencias no satisfechas. Sigue las indicaciones que te ofrezco y no tendrás problemas.

En Ubuntu, Debian, Linux Mint y derivados:

```
sudo dpkg -i LibreOffice_*/DEBS/*.deb
```

En Fedora, Alma Linux, CentOS y derivados:

```
sudo dnf install LibreOffice_*/RPMS/*.rpm
# o
sudo yum install LibreOffice_*/RPMS/*.rpm
```

En SUSE, OpenSUSE y derivados:

```
sudo zypper install LibreOffice_*/RPMS/*.rpm
```

```
sergio@sololinux:~/Descargas$ sudo dpkg -i LibreOffice_*/DEBS/*.deb
[sudo] password for sergio:
Seleccionando el paquete libobasis7.2-base previamente no seleccionado.
(Leyendo la base de datos ... 132127 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../libobasis7.2-base_7.2.0.4-4_amd64.deb ...
Desempaquetando libobasis7.2-base (7.2.0.4-4) ...
Seleccionando el paquete libobasis7.2-calc previamente no seleccionado.
Preparando para desempaquetar .../libobasis7.2-calc_7.2.0.4-4_amd64.deb ...
Desempaquetando libobasis7.2-calc (7.2.0.4-4) ...
Seleccionando el paquete libobasis7.2-core previamente no seleccionado.
Preparando para desempaquetar .../libobasis7.2-core_7.2.0.4-4_amd64.deb ...
Desempaquetando libobasis7.2-core (7.2.0.4-4) ...
www.sololinux.es
```

Como puedes comprobar, tenemos instalada la última versión estable de LibreOffice, hablamos de la suite **7.2.0.4**.

Acerca de LibreOffice
^ x



LibreOffice

The Document Foundation

LibreOffice es un paquete de productividad moderno, fácil de usar y de código abierto para procesar texto, hojas de cálculo, presentaciones y más.

Versión suministrada por The Document Foundation.
© 2000-2021 de los colaboradores de LibreOffice.
LibreOffice se basó en OpenOffice.org.

[Créditos](#)
[Sitio web](#)
[Novedades](#)

Información de la versión

Versión:	7.2.0.4 / LibreOffice Community
Montaje:	9a9c6381e3f7a62afc1329bd3...
Entorno:	Subprocs. CPU: 2; SO: Linux 5.10
Interfaz de usuario:	Repres. IU: predet.; VCL: gtk3
Config. regional:	es-ES (es_ES.UTF-8); Interfaz: es-ES
Otros:	Calc: threaded



www.sololinux.es
Cerrar

Autor: **Henry G.R.**

Diferencias entre apt upgrade y apt full-upgrade

¿Cuál es la diferencia entre el comando **apt upgrade** y **apt full-upgrade**?. Pueden parecer lo mismo por la similitud de sus funciones, pero te aseguro que tienen diferencias sustanciales sobre el como y cuando operar con ellos.

Sus principales diferencias son: El comando **apt upgrade**, se usa para descargar y aplicar cualquier actualización disponible de sus paquetes de una manera segura. Esto se logra **NO eliminando** previamente, los paquetes que están instalados en el sistema Linux. Por el contrario, el **comando apt full-upgrade** tiene la misma función exceptuando que si fuera necesario (para evitar conflictos), borra los paquetes previamente instalados antes de instalar los nuevos.

```
sergio@sololinux:~$ sudo apt update
[sudo] password for sergio:
Obj:1 http://deb.debian.org/debian bullseye InRelease
Des:2 http://security.debian.org/debian-security bullseye-security InRelease [44,1 kB]
Des:3 http://deb.debian.org/debian bullseye-updates InRelease [36,8 kB]
Descargados 80,9 kB en 3s (32,0 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Todos los paquetes están actualizados.
sergio@sololinux:~$ sudo apt upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
sergio@sololinux:~$ sudo apt full-upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
sergio@sololinux:~$
```

Diferencias entre apt upgrade y apt full-upgrade

Pongamos un sencillo ejemplo. Tenemos una herramienta llamada X cuya versión instalada es la 1.8, que al ejecutar los siguientes comandos...

```
sudo apt update
sudo apt list --upgradable
```

Nos dice que la herramienta X tiene una nueva versión, la 2.1. Si ejecutamos...

```
sudo apt upgrade
```

Se mantendrán la inmensa mayoría de los archivos de la versión 1.8, por si te resultan de utilidad en un futuro no muy lejano. Por otro lado, si en vez de **upgrade** usamos...

```
sudo apt full-upgrade
```

Se borrará por completo la anterior versión (en el ejemplo la 1.8), y se instalará la 2.1. Esto generalmente es necesario al actualizar de una versión a otra, ya que es algo frecuente que los paquetes más nuevos tengan problemas con algunas dependencias de los paquetes anteriores.

Por otro lado, en situaciones específicas puede hacer más daño que beneficios ejecutar «**full-upgrade**». Ten en cuenta que a veces, puedes tener algún software personalizado en un **sistema Linux** que depende de aquellos paquetes más antiguos, Este software personalizado que tú has desarrollado, puede no ser reconocido correctamente por el **comando apt**, por lo que al ejecutar «**full-upgrade**» se borren bibliotecas que resultan necesarias para el buen funcionamiento de la aplicación.

Como puedes ver, las **diferencias entre upgrade y full-upgrade** son pocas pero importantes. Lo resumimos en la siguiente tabla.

apt upgrade	apt full-upgrade
No elimina paquetes y omite la actualización si es necesario eliminar.	Elimina paquetes antiguos para actualizar a una nueva versión.
Se suele usar durante el ciclo de vida de una versión en particular.	Se suele usar para actualizar a una versión más moderna.
Opción muy segura	Usar si sabes lo que haces y con precaución

Modificar la swap con bash script

Cuando instalamos una **distribución Linux**, como norma general se habilita una partición en el disco duro para la **memoria Swap**, también conocida como espacio de intercambio (si no es tu caso te recomiendo crearla manualmente). Esta memoria es específica, para no saturar la memoria RAM física del sistema.

Cuando los procesos en ejecución de Linux llenan la RAM hasta un valor predeterminado, nuestro sistema recurre a esta partición del disco (Swap) buscando más espacio. Esto está pensado para no saturar la **RAM** y, que nuestro sistema no sufra interrupciones o cuellos de botella hasta que logre liberar carga de la RAM.

Ver, liberar o configurar los valores de la Swap mediante comandos es tarea sencilla, pero te imaginas un script bash que lo haga todo por nosotros. Mi paisano Javier de [Weblinus](#) lo hace realidad y, como buen linuxero que es... lo ofrece a [SoloLinux.es](#) para masificar su difusión. Muchas gracias Javier, realmente interesante este **script bash**.

Modificar la Swap con bash script

El interesante script que hoy proponemos, nos ofrece un menú desde el cual podemos realizar importantes tareas relacionadas con la Swap. Pero antes de comenzar, debes recordar que el **swappiness** es un valor que nos permite ajustar el equilibrio entre la RAM y la Swap.

Presentamos el menú del script:

- **Ver el estado de la RAM y la Swap**
- **Imprimir el swappiness actual**
- **Deshabilitar la Swap**
- **Habilitar la Swap**
- **Mover datos de la Swap a la RAM**
- **Modificar el swappiness para uso actual**
- **Modificar el swappiness para uso permanente**

Imagen de ejemplo.

```

mar ago 24 20:13:03 EEST 2021
-----
                Menu Principal
-----

1. Ver estado de uso de la RAM y Swap
2. Ver valor de swappiness actual
3. Deshabilitar el uso de la swap
4. Habilitar el uso de la swap
5. Trasladar datos de la swap a la RAM
6. Modificar el valor de swappiness para la sesión
7. Modificar valor de swappiness de forma persistente
8. Salir

Elige una opción [ 1 - 8 ] █          www.sololinux.es

```

Como ves en la imagen de ejemplo, su utilidad está fuera de toda duda. Así que no perdamos más tiempo, creamos el script.

```
sudo nano scriptswap.sh
```

Copia y pega el código creado por Javier.

```
#!/bin/bash
# scriptswap.sh
# Script para gestionar el uso de la memoria swap del sistema.
# Creado por Francisco Javier Izquierdo Hidalgo.
# Licencia Pública General de GNU v2.
# Se distribuye SIN NINGUNA GARANTÍA, tampoco la implícita de -
# - COMERCIALIZACIÓN o APTITUD PARA UN PROPÓSITO EN PARTICULAR.
# Más detalles en GNU General Public License.
# Variables
estado=$(free -h)
valor=$(cat /proc/sys/vm/swappiness)
# Funciones
# Indicación de pausa de visualización
function pause(){
    local message="$@"
    echo
    [ -z $message ] && message="Presiona [Enter] para continuar..."
    read -p "$message" readEnterKey
}
# Mostrar un menú en pantalla
function show_menu(){
    date
    echo "-----"
    echo "                Menu Principal                "
    echo "-----"
    echo
    echo "1. Ver estado de uso de la RAM y Swap"
    echo "2. Ver valor de swappiness actual"
    echo "3. Deshabilitar el uso de la swap"
    echo "4. Habilitar el uso de la swap"
    echo "5. Trasladar datos de la swap a la RAM"
    echo "6. Modificar el valor de swappiness para la sesión"
    echo "7. Modificar valor de swappiness de forma persistente"
    echo "8. Salir"
}
# Mostrar mensaje de cabecera
function write_header(){
    local h="$@"
    echo
    "-----"
    echo "    ${h}"
    echo
    "-----"
}
# Ver estado de uso de RAM y Swap
function uso_memoria(){
    write_header " Información del uso de la memoria"
    echo "El uso de RAM y swap es:"
    echo
    echo "$estado"
    pause
}
# Ver valor de swappiness actual
function val_swappiness(){
    write_header " Valor actual de swappiness"
    echo "El valor actual de swappiness es:"
    cat /proc/sys/vm/swappiness
    pause
}
# Deshabilitar el uso de la swap
function swap_off(){
    write_header " Deshabilitando el uso de la memoria swap"
    sudo swapoff -a
    sleep 3
    echo
    echo "Deshabilitado el uso de la swap"
    uso_memoria
}
# Habilitar el uso de la swap
function swap_up(){
    write_header " Habilitando el uso de la memoria swap"
    sudo swapon -a
    sleep 3
    echo
    echo "Habilitado el uso de la swap"
    uso_memoria
}
# Trasladar datos de la Swap a RAM
function swap_on(){
    write_header " Trasladando los datos de la swap a la RAM"
    sudo swapoff -a ; sudo swapon -a
    sleep 3
    echo
    echo "Trasladados los datos de la swap a la RAM con éxito"
    uso_memoria
}
}
```

CONTINUA EN EL SIGUIENTE CODIGO...

```
# Modificar el valor swappiness en la sesión actual
function swappiness_ses(){
    echo "
"
    write_header "Vamos a modificar el valor swappiness para la sesión
actual"
    echo "Cuanto mayor sea este valor, más tardará el sistema en usar la
swap"
    echo "Introduce el valor que te interese (entre 0 y 100)"
    read n
    numero=$(echo $n | grep "^[0-9]*$")
    if [ $numero ]
    then
        sudo sysctl vm.swappiness=$n
        echo "El nuevo valor de swappiness es :"
        cat /proc/sys/vm/swappiness
    else
        echo "El valor debe ser numérico y estar en el rango 0-100"
        exit 1
    fi
}
# pause
}
# Modificar el valor swappiness de forma persistente
function swappiness_per(){
    write_header " Vamos a modificar el valor swappiness de forma
persistente"
    echo "Se requieren privilegios de sudo y los cambios se aplicarán
tras un reinicio"
    echo "Cuanto mayor sea este valor, más tardará el sistema en usar la
swap"
    echo "Introduce el valor que te interese (entre 0 y 100)"
    read n
    numero=$(echo $n | grep "^[0-9]*$")
    if [ $numero ]
    then
        sudo write "vm.swappiness=$n" >> /etc/sysctl.conf
        val_swappiness
    else
        echo "El valor debe ser numérico y estar en el rango 0-100"
    fi
}
# pause
}
# Obtener entrada por teclado y elegir opción
function read_input(){
    local c
    echo
    read -p "Elige una opción [ 1 - 8 ] " c
    case $c in
        1) uso_memoria ;;
        2) val_swappiness ;;
        3) swap_off ;;
        4) swap_up ;;
        5) swap_on ;;
        6) swappiness_ses ;;
        7) swappiness_per ;;
        8) echo ; echo " FINALIZADO EL PROGRAMA." ; echo ; echo " Vigila
la estabilidad del sistema!" ; exit 0 ;;
        *)
            echo "Seleccione una sola opción entre 1 to 8."
            pause
        esac
}
# Programa principal
while true
do
    clear
    show_menu # mostrar menu
    read_input # esperar la entrada del usuario
done
```

Una vez lo tengas listo, guarda el archivo y cierra el editor.

Puedes ejecutar el script con el siguiente comando.

```
sudo bash scriptswap.sh
```

Si lo prefieres, también puedes concederle permisos de ejecución y, lanzar la herramienta.

```
sudo chmod +x scriptswap.sh
./scriptswap.sh
```

Vemos un par de imágenes de ejemplo, para que te hagas una idea de lo sencillo que es usar el script. En la primera imagen vemos el valor actual **Swappiness**.

```
mar ago 24 20:17:45 EEST 2021
-----
                        Menu Principal
-----

1. Ver estado de uso de la RAM y Swap
2. Ver valor de swappiness actual
3. Deshabilitar el uso de la swap
4. Habilitar el uso de la swap
5. Trasladar datos de la swap a la RAM
6. Modificar el valor de swappiness para la sesión
7. Modificar valor de swappiness de forma persistente
8. Salir

Elige una opción [ 1 - 8 ] 2
-----
                        Valor actual de swappiness
-----
El valor actual de swappines es:
60

Presiona [Enter] para continuar...  www.sololinux.es
```

Nosotros lo bajamos a 20, pero solo para la sesión actual. Al reiniciar el sistema, el valor volverá a su estado predeterminado (en este caso 60).

```
mar ago 24 20:50:41 EEST 2021
-----
                        Menu Principal www.sololinux.es
-----

1. Ver estado de uso de la RAM y Swap
2. Ver valor de swappiness actual
3. Deshabilitar el uso de la swap
4. Habilitar el uso de la swap
5. Trasladar datos de la swap a la RAM
6. Modificar el valor de swappiness para la sesión
7. Modificar valor de swappiness de forma persistente
8. Salir

Elige una opción [ 1 - 8 ] 6
-----
                        Vamos a modificar el valor swappiness para la sesión actual
-----
Cuanto mayor sea este valor, más tardará el sistema en usar la swap
Introduce el valor que te interese (entre 0 y 100)
20 
```

Diferencias entre apt y apt-get en Ubuntu y derivados



No hace mucho tiempo, alguien me recriminó en un foro que había utilizado «**apt-get**». Ese comando ya no existe, **APT** es su sucesor me espeto el tipo todo orgulloso. Ok, ok respondí, reconozco que soy tonto y tú en el biberón tomabas **pingüinos tux** triturados. Chao amigo, ya aprenderás cuando seas mayor.

Pasados unos días, ya más tranquilo (a veces este genio **aragonés** me desboca) quería explicar sus diferencias, particularidades y más; Pues detecto que existe un gran desconocimiento en este aspecto, incluso en personal docente que lo explica (por suerte son pocos), pero lo explica mal. Así que vamos a ello, a ver cuantas controversias genera este artículo de hoy, jeje.



Diferencias entre apt y apt-get en Ubuntu y derivados

Partimos de la base que APT (Advanced Packaging Tool) no es ningún comando, APT es un administrador de paquetes en línea de comandos desarrollado por **Debian** en 1998. Si no recuerdo mal, fue el primer administrador de paquetes desarrollado para Linux, incluso anterior al **gestor rpm**.

APT nos ofrece herramientas en línea de comandos, que nos permiten buscar y administrar los paquetes del sistema, también obtener información de las aplicaciones, herramientas y librerías instaladas, así como el acceso en bajo nivel a funciones especiales de **libapt-pkg** y **libapt-inst**.

APT como gestor, nos ofrece los siguientes comandos que podemos usar a nuestro libre albedrío.

- **apt-get** – Nos permite recuperar paquetes, obtener información sobre ellos, instalar, actualizar y eliminarlos incluyendo sus dependencias
- **apt-cache** – Con este comando, puedes consultar toda información disponible de los paquetes instalados y, de los que tienes disponibles.

- **apt-cdrom** – Permite utilizar dispositivos extraíbles, como fuente de origen de los paquetes.
- **apt-config** – Interfaz en terminal para modificar la configuración.
- **apt-key** – Excelente utilidad para administrar las claves de autenticación.
- **apt-extracttemplates** – Usado por **debconf** para realizar preguntas sobre la configuración, del paquete a instalar.
- **apt-ftparchive** – Tiene la capacidad de crear paquetes y otros archivos de índice, que resultan necesarios para publicar un archivo de **paquetes deb**
- **apt-sortpkgs** – Ordena los índices.

Todos estos comandos son de APT, algo que no debes confundir con el **comando apt**, no tiene nada que ver. Puedes ver la explicación de todos los comandos anteriores, con el sencillo paso de ejecutarlos en tu terminal (no es necesario **sudo**). Ejecuta el siguiente comando.

```
apt-get
sergio@sololinux:~$ apt-get
apt 1.6.14 (amd64)
Uso: apt-get [opciones] orden
      apt-get [opciones] install|remove paq1 [paq2 ...]
      apt-get [opciones] source paq1 [paq2 ...]

apt-get es una interfaz de línea de órdenes para la obtención de
paquetes y de información sobre ellos de orígenes autenticados y
para la instalación, actualización y eliminación de paquetes junto
a sus dependencias.

Órdenes más utilizadas:
update - Descarga nuevas listas de paquetes
upgrade - Realiza una actualización
install - Instala nuevos paquetes (paquete es libc6 y no libc6.deb)
remove - Elimina paquetes
purge - Elimina y purga paquetes
autoremove - Elimina automáticamente todos los paquetes sin utilizar
dist-upgrade - Actualiza la distribución, vea apt-get(8)
dselect-upgrade - Sigue las selecciones de dselect
build-dep - Configura las dependencias de construcción para paquetes fuente
clean - Elimina los archivos descargados
autoclean - Elimina los archivos descargados antiguos
check - Verifica que no haya dependencias incumplidas
source - Descarga archivos fuente
download - Descarga el paquete binario al directorio actual
changelog - Descarga y muestra el informe de cambios para el paquete proporcionado

Vea apt-get(8) para más información sobre las órdenes disponibles.
Las opciones y sintaxis de configuración se detallan en apt.conf(5).
Puede encontrar información sobre cómo configurar las fuentes en sources.list(5).
Las elecciones de versión y de paquetes se pueden expresar por medio de apt_preferences(5).
Los detalles de la seguridad están disponibles en apt-secure(8).
Este APT tiene poderes de Super Vaca.
sergio@sololinux:~$
```

Otro interesante ejemplo...

```
apt-cache
sergio@sololinux:~$ apt-cache
apt 1.6.14 (amd64)
Uso: apt-cache [opciones] orden
      apt-cache [opciones] show paq1 [paq2 ...]

apt-cache solicita y muestra la información disponible sobre los paquetes
instalados y los instalables. Funciona exclusivamente con los datos obtenidos
en el caché local mediante la orden «update», como «apt-get». Por tanto,
la información mostrada puede estar desactualizada si la última actualización
se hizo hace mucho, pero a cambio apt-cache funciona independientemente de la
disponibilidad de las fuentes configuradas (por ejemplo, sin conexión).

Órdenes más utilizadas:
showsrc - Muestra la información de fuentes
search - Busca en la lista de paquetes según un patrón de expresión regular
depends - Muestra la información de dependencias en bruto para el paquete
rdepends - Muestra la información de dependencias inversas del paquete
show - Muestra un registro legible para el paquete
pkgnames - Lista los nombres de todos los paquetes en el sistema
policy - Muestra parámetros de las normas

Vea apt-cache(8) para más información sobre las órdenes disponibles.
Las opciones y sintaxis de configuración se detallan en apt.conf(5).
Puede encontrar información sobre cómo configurar las fuentes en sources.list(5).
Las elecciones de versión y de paquetes se pueden expresar por medio de apt_preferences(5).
Los detalles de la seguridad están disponibles en apt-secure(8).
sergio@sololinux:~$
```

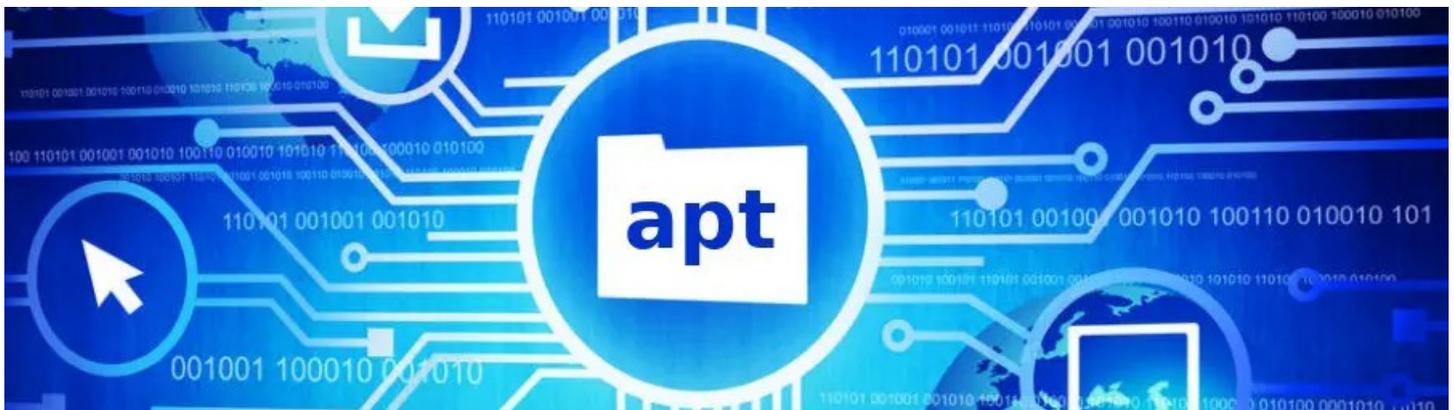
En las imágenes anteriores, al revisarlas detenidamente (las puedes ejecutar en tu sistema basado en Debian) nos damos cuenta de que no nos informa de todos los subcomandos, dice «**Órdenes más utilizadas**». Bien esto tiene una explicación bastante sencilla, pues sobre finales del 2013 y principios de 2014 se desarrolló un derivado directo del **comando apt-get**. Ahora sí que hablamos del **comando apt**.

Existen especulaciones sobre quien desarrollo en sus principios el **comando apt**, muchos dicen que fue Ubuntu. Personalmente... creo que el propio equipo de desarrolladores de **APT**, que tuvo la brillante idea de hacer una especie de comando todo en uno.

Estos programadores tomaron las «**Órdenes más utilizadas**», de «**apt-get**» y «**apt.cache**», más algunos otros incluyendo novedades para posteriormente mejorarlos y fusionar en uno solo. Es evidente que hablamos del **comando apt**.

El resultado final fue adoptado rápidamente por todas las distribuciones que utilizaban APT, para que me entiendas... un auténtico éxito. Incluso a día de hoy, **APT** ya lo considera un comando más de su amplia familia y la verdad, es que no es para menos.

- **apt** – Interfaz en línea de comandos (alto nivel), para mejorar la interacción y experiencia de usuario.



El problema de **apt**, es que no es un comando completo. Perdona, tal vez me expresé mal, quería decir que no es tan completo como **apt-get** o **apt-cache**, tan solo es una fusión limitada de los dos y añadidos, pero realmente funcional para los usuarios menos exigentes.

No sería la primera vez, que me encuentro con sistemas que **apt** falla con comandos básicos como **upgrade**, y sin embargo funciona correctamente el **apt-get upgrade**. Por todo ello, podríamos afirmar que bloggers que lanzan afirmaciones como esta...

El administrador de paquetes apt reemplaza el antiguo apt-get y apt-cache. Directamente, se implementan todos los comandos más utilizados para instalar, eliminar paquetes, actualizar el sistema y buscar paquetes.

Son lo que acabamos de decir, simples bloggers tal vez aficionados a Linux, o tal vez ni eso. El **comando apt** no reemplaza nada, es una división de varios comandos que actualmente copa el mercado por un motivo muy claro, es realmente bueno y está muy logrado. Aun así... no tiene el potencial de los comandos primarios creados para **manejar APT**.

Conclusión: **APT** no es lo mismo que **apt**.

Como instalar VirusTotal Uploader en Linux



VirusTotal es un servicio online gratuito, que escanea y analiza archivos y URL en busca de virus, gusanos, troyanos y otros tipos de contenido malicioso. Para llevar a cabo su labor de manera efectiva, hace uso de más de 80 motores de **antivirus**, además de otros escáneres y sitios web con **listas black** de diversas índoles.

VirusTotal Uploader es una herramienta de escritorio nativa para **Linux**, que brinda un servicio privado de conexión online con el servicio **VirusTotal**, desde tu propio sistema. La aplicación permite cargar archivos para escanear y verificar que son totalmente válidos y, que están libres de aplicaciones dañinas.



Como instalar VirusTotal Uploader en Linux

La herramienta de escritorio, está disponible como **paquete flatpak**, por tanto, su instalación es bastante simple desde la terminal de tu **distribución Linux** preferida. Lo primero que hacemos es verificar que disponemos del administrador de **Flathub**.

```
flatpak --version
```

En nuestro caso, el resultado es afirmativo.

```
sergio@sololinux:~$
sergio@sololinux:~$ flatpak --version
Flatpak 1.11.2
sergio@sololinux:~$ www.sololinux.es
```

Si no lo tienes, su instalación es bastante simple.

Ubuntu, Linux Mint y derivados:

```
sudo add-apt-repository ppa:alexlarsson/flatpak
sudo apt update
sudo apt install flatpak
```

Debian y derivados:

```
sudo apt-get install flatpak
```

Fedora, Alma Linux, CentOS y derivados:

```
sudo dnf install flatpak
# 0
sudo yum install flatpak
```

Arch Linux, Manjaro y derivados:

```
sudo pacman -S flatpak
# 0
sudo pamac install flatpak
```

```
sergio@sololinux:~$ sudo apt install flatpak
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
 flatpak
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 1.330 kB de archivos.
Se utilizarán 6.857 kB de espacio de disco adicional después de esta operación.
Desí http://ppa.launchpad.net/alexlarsson/flatpak/ubuntu/bionic/main amd64 flatpak amd64 1.11.2-1flatpak1-18.04 [1.330 kB]
Descargados 1.330 kB en 0s (2.827 kB/s)
Seleccionando el paquete flatpak previamente no seleccionado.
(Leyendo la base de datos ... 378883 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../flatpak 1.11.2-1flatpak1-18.04 amd64.deb ...
Desempaquetando flatpak (1.11.2-1flatpak1-18.04) ...
Configurando flatpak (1.11.2-1flatpak1-18.04) ...
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...
Procesando disparadores para dbus (1.12.2-1ubuntu1.2) ...
Procesando 1 archivo doc-base añadido...
Registrando documentos con scrollkeeper...
sergio@sololinux:~$
```

Ahora solo falta instalar la herramienta **Uploader de VirusTotal**. Pero antes agregamos la aplicación remota, para ello ejecuta el siguiente comando.

```
sudo flatpak remote-add --if-not-exists flathub
https://flathub.org/repo/flathub.flatpakrepo
```

Ahora si podemos instalar.

```
sudo flatpak install flathub com.virustotal.VirusTotalUploader
```

Responde a todo «Y», instala

```
sergio@sololinux:~$ sudo flatpak install flathub
com.virustotal.VirusTotalUploader
Buscando coincidencias...
Required runtime for com.virustotal.VirusTotalUploader/x86_64/stable
(runtime/org.kde.Platform/x86_64/5.15) found in remote flathub
Do you want to install it? [Y/n]: y
com.virustotal.VirusTotalUploader permissions:
ipc                network            fallback-x11       wayland
x11                dri                file access [1]
dbus access [2]
[1] host:ro, xdg-config/kdeglobals:ro
[2] com.canonical.AppMenu.Registrar
ID                                     Rama
Op                                     Remoto                               Descarga
1. [✓] org.freedesktop.Platform.GL.default 20.08
u                                     flathub                             105,8 MB / 106,4 MB
2. [✓] org.freedesktop.Platform.VAAPI.Intel 20.08
u                                     flathub                             11,5 MB / 11,6 MB
3. [✓] org.freedesktop.Platform.openh264    2.0
u                                     flathub                             1,5 MB / 1,5 MB
4. [✓] org.gtk.Gtk3theme.Mint-Y             3.22
u                                     flathub                             75,7 kB / 118,5 kB
5. [✓] org.kde.Platform.Locale             5.15
i                                     flathub                             4,4 MB / 341,2 MB
6. [ ] org.kde.Platform                    5.15
i                                     flathub                             107,2 MB / 370,3 MB
7. [ ] com.virustotal.VirusTotalUploader    stable
i                                     flathub                             < 178,3 kB
Installing 6/7... ██████████ 39% 4,9 MB/s 00:34
```

Una vez concluya la instalación, la ejecutamos desde la **terminal** de nuestro **sistema linux**.

```
flatpak run com.virustotal.VirusTotalUploader
```

Se abre la herramienta.



Selecciona los archivos a escanear en «File —> Selec Files». El análisis comienza automáticamente, espera a que termine el proceso y te ofrezca el resultado.

VirusTotal Uploader 1.3 (Drag and drop files to window)

File Help

Scan Log

Drag and drop files here to scan.

	File	Status	Detections	Scan Date	
1	Databases.db	Done	0 / 56	26/8/21 12:38	https://www.virustotal.com/gui/file/1ee56f963da4f0ce325a89f9c532b76a7a16af511
2	jmeter.log	Wait 29		26/8/21 15:41	https://www.virustotal.com/gui/file/2c5cf476ef874b30bd81e5637d67a7f9808ae734
3					

Quota Remaining:4 Request(s)/Minute

www.sololinux.es

Si quieres deshacerte de la herramienta, la puedes desinstalar con este comando que te propongo.

```
sudo flatpak uninstall com.virustotal.VirusTotalUploader
```

Ejemplo.

```
root@sololinux:~# sudo flatpak uninstall com.virustotal.VirusTotalUploader
ID                               Rama      Op
1. [-] com.virustotal.VirusTotalUploader  stable  r
Uninstall complete.
root@sololinux:~#
```

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

Chat de SoloLinux en Telegram

PayPal
 Donar a SoloLinux

PayPal
 Donar a Revistalinux

designed by freepik

30 años de Kernel Linux entre nosotros

Corría el 25 de agosto de 1991, después de cinco meses de desarrollo y, cuando el resto del mundo trabajaba sobre **Unix**, **BSD**, etc.; Un estudiante de 21 años llamado **Linus Torvalds**, anunció al mundo mediante teleconferencia en **comp.os.minix** que estaba trabajando en un prototipo funcional de un nuevo sistema operativo, al que hoy conocemos como Linux. La portabilidad de **bash 1.08** y **gcc 1.40** ya la tenía completada. El 17 de septiembre, ve la luz la primera versión pública del kernel Linux.

El **kernel 0.0.1** tenía un peso comprimido de 62 KB, con un contenido de unas 10 mil líneas de código fuente (los actuales rondan los 30 millones de líneas). **Torvals** se inspiró en el sistema **MINIX**, pero al no estar de acuerdo con su licencia limitada se lanzó a la aventura de crear su propio sistema. Hoy celebramos los 30 años de Kernel Linux.

Linux se extendió de forma muy rápida, tanto, que no tardó mucho en pasar por encima de MINIX. Esto no pasó desapercibido para **Andrew Tanenbaum** (el creador de **MINIX**), que lo acusó públicamente de haber copiado su código. Un estudiante, alumno de **Andrew Tanenbaum**, realizó una comparación detallada del código Minix con las primeras versiones públicas de Linux. El resultado fue aplastante, se demostró que solo existían cuatro coincidencias de código insignificantes, que además eran obligadas por los requisitos de **POSIX** y **ANSI C**.



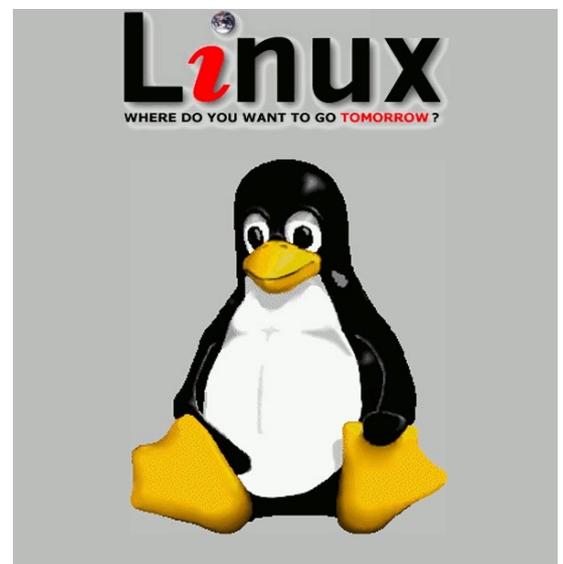
30 años de Kernel Linux entre nosotros

En un principio, **Linus Torvalds** llamó al núcleo **Freax**, a partir de las palabras «**freak**» y **X (Unix)**. Pero el kernel recibió el nombre «Linux» de la mano de **Ari Lemmke**, quien por solicitud de **Torvals**, instaló el nuevo kernel en el **servidor FTP** de la universidad, renombrando el directorio como Linux. A partir de ese momento, todo el mundo comenzó a llamarle **Linux**, y como es lógico... nunca faltan los avispados, pues el empresario **William Della Croce** logró registrar la marca Linux y llegó al punto de querer cobrar royalties por el uso del nombre.

Un poco más tarde, debido a las presiones que sufría decidió transferir todos los derechos de la marca a **Linus**, su autor original. La inconfundible mascota oficial del kernel, es el **pingüino Tux**, que fue seleccionada a través de un **concurso** realizado en la comunidad en 1996. Su nombre, **Tux**, quiere decir **Torvalds UniX**.

Aquí vemos la cronología de lanzamientos, de las versiones principales del Linux Kernel (incluyendo el número de líneas de su código fuente).

- 0.0.1 – Septiembre de 1991 – 10 mil líneas de código.
- 1.0.0 – Marzo de 1994 – 176 mil líneas de código.
- 1.2.0 – Marzo de 1995 – 311 mil líneas de código.
- 2.0.0 – Junio de 1996 – 778 mil líneas de código.
- 2.2.0 – Enero de 1999 – 1,8 millones de líneas de código.
- 2.4.0 – Enero de 2001 – 3,4 millones de líneas de código.
- 2.6.0 – Diciembre de 2003 – 5,9 millones de líneas de código.
- 2.6.28 – Diciembre de 2008 – 10,2 millones de líneas de código.
- 2.6.35 – Agosto de 2010 – 13,4 millones de líneas de código.
- 3.0 – Agosto de 2011 – 14,6 millones de líneas de código.
- 3.5 – Julio de 2012 – 15,5 millones de líneas de código.
- 3.10 – Julio de 2013 – 15,8 millones de líneas de código.
- 3.16 – Agosto de 2014 – 17,5 millones de líneas de código.
- 4.1 – Junio de 2015 – 19,5 millones de líneas de código.
- 4.7 – Julio de 2016 – 21,7 millones de líneas de código.
- 4.12 – Julio de 2017 – 24,1 millones de líneas de código.
- 4.18 – Agosto de 2018 – 25,3 millones de líneas de código.
- 5.2 – Julio de 2019 – 26,55 millones de líneas de código.
- 5.8 – Agosto de 2020 – 28,4 millones de líneas de código.
- 5.13 – Junio de 2021 – 29,2 millones de líneas de código.



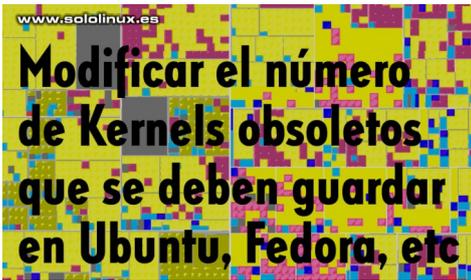
Ahora, también cronológicamente conocemos su evolución en el desarrollo, y otros acontecimientos importantes del Kernel.

- **Linux 0.0.1** – Septiembre de 1991, primera versión pública que solo admitía CPU i386 y se inicia desde disquete.
- **Linux 0.12** – Enero de 1992, el código se distribuye bajo la licencia GPLv2.
- **Linux 0.95** – Marzo de 1992, se ofrece la capacidad de ejecutar el sistema X Window, soporte para memoria virtual e intercambio de particiones.
- **Linux 0.96-0.99, 1992-1993** – Se inició el trabajo en los paquetes sobre redes. Se introdujo el sistema de archivos Ext2, se agregó soporte para el formato de archivo ELF, también los controladores para tarjetas de sonido y controladores SCSI, se implementaron los módulos de carga del kernel y el sistema de archivos /proc.
- **1992** – Aparecieron las primeras distribuciones SLS e **Yggdrasil**. En el verano de 1993, se fundaron los proyectos **Slackware** y **Debian**. Había nacido el mundo Linux.
- **Linux 1.0** – Marzo de 1994, primera versión oficial considerada como estable.
- **Linux 1.2** – Marzo de 1995, aumento significativo en el número de controladores desarrollados para Linux, soporte para otras plataformas como Alpha, MIPS y SPARC, expansión de capacidades de los paquetes desarrollados para la red, nace un filtro de paquetes y el soporte NFS.
- **Linux 2.0** – Junio de 1996, agregado el soporte para sistemas multiprocesador.
- **Marzo de 1997** – Se funda **LKML**, la lista de correo oficial para desarrolladores del kernel Linux.
- **1998** – Lanzamiento del primer clúster Top500 basado en Linux, que consta de 68 nodos con CPU Alpha.
- **Linux 2.2** – Enero de 1999, se aporta una mayor eficiencia al sistema de administración de la memoria, ahora soporta IPv6, se implanta un nuevo firewall y un nuevo subsistema de sonido.
- **Linux 2.4** – Febrero de 2001, soporte para sistemas de 8 procesadores y 64 GB de RAM, sistema de archivos Ext3, USB y soporte ACPI.
- **Linux 2.6** – Diciembre de 2003, soporte SELinux, herramientas de ajuste automático del kernel, sysfs, el sistema de administración de memoria ha sido rediseñado.
- **En 2005** – Se introduce el **hipervisor Xen**, es el comienzo de la era de la virtualización.
- **Septiembre de 2008** – Primera versión de la plataforma **Android** basada en el kernel Linux.
- **Julio de 2011** – Después de 10 años de desarrollo de la rama 2.6.x, se salta a la 3.x. Los objetos en el repositorio de Git alcanzan los 2 millones.
- **2015** – Lanzamiento del kernel Linux 4.0. El número de objetos git en el repositorio, sube hasta los 4 millones.
- **Abril de 2018** – Se supera la barrera de los 6 millones de objetos git-core en el repositorio.
- **Enero de 2019** – Nace una nueva rama del Kernel, la 5.0. El repositorio git crece de manera imparable.
- **Agosto de 2020** – El nuevo kernel 5.8 supone una revolución, se convierte en la versión con más cambios y mejoras desde que nació el proyecto.
- **Kernel 5.13** – Se establece un récord en el número de desarrolladores activos del Kernel (2.150).
- **2021** – Se añade un código para desarrollar controladores bajo **lenguaje Rust**. En la actualidad, se continúa trabajando para incluir más componentes de apoyo a Rust en el núcleo central.



A lo largo de estos 30 años, Universidades, programadores certificados, otros anónimos y más han sido fundamentales en su desarrollo. En realidad, muchos han sido los que han colaborado, para hacer de Linux un producto único. Sorprendentemente, también se debe agradecer las aportaciones de grandes empresas, como pueden ser: **AMD, Intel, Huawei, Red Hat, Google, Linaro, NVIDIA, Facebook, SUSE, Oracle, IBM, Canonical, ARM** y muchas más. Gracias a todos por hacer que Linux, sea lo que es hoy en día. Viva Linux.

Modificar los kernel que se guardan en Ubuntu

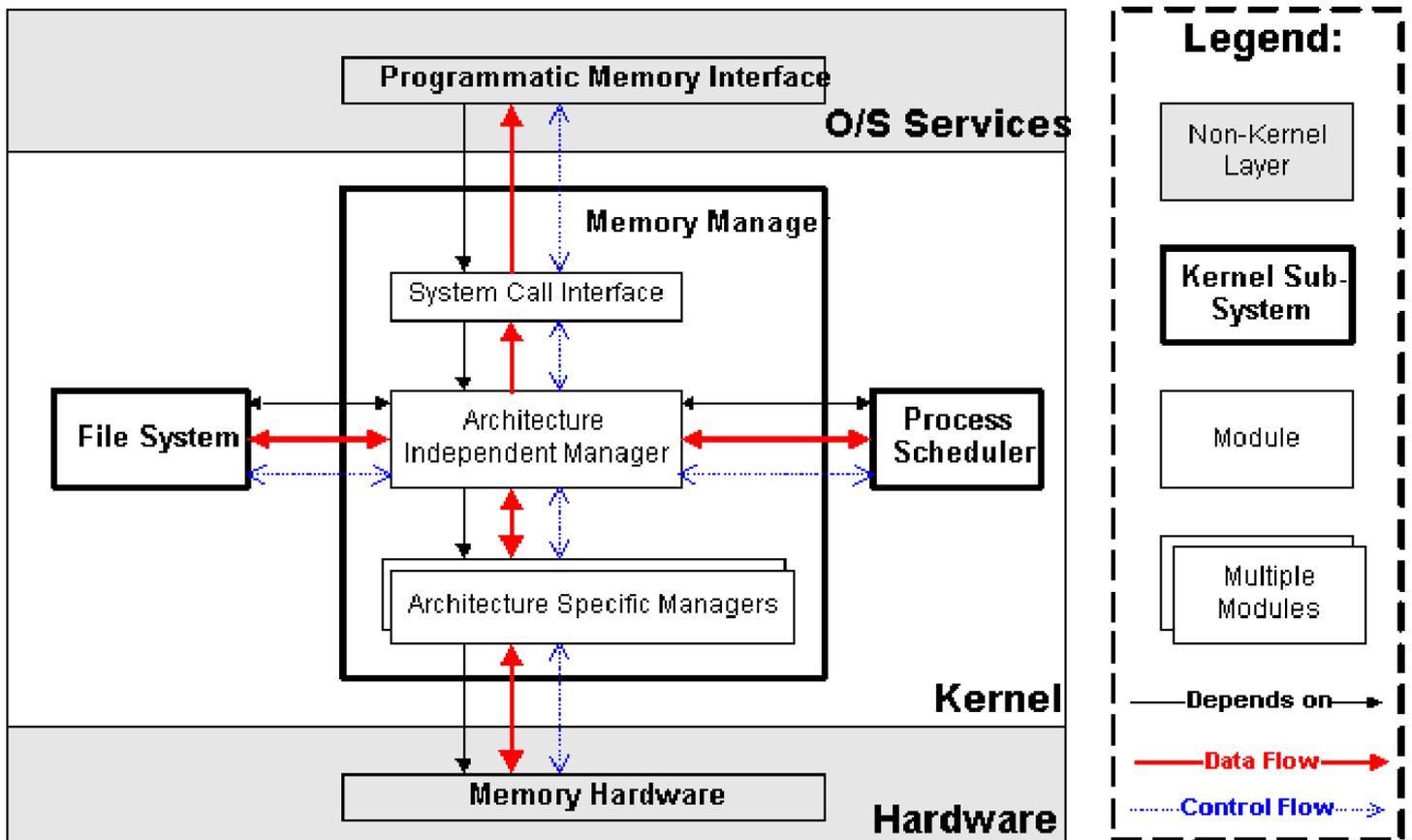


En este artículo quiero responder a varios usuarios, que me han hecho la misma pregunta por privado. Vemos una de ellas a modo de ejemplo. Nuestro fiel lector **Octavio de Cuba**, nos escribe...

Hola sololinux, una duda que tengo. ¿Los archivos de arranque en el grub, los initrd.img-5.4.0-26-generic y demás versiones, se podrán borrar? Y como se hace?, en estos momentos la PC debe estar arrancando por la versión initrd.img-5.4.0-83-generic que es la última actualizada, si sabe usted y si es recomendable hacerlo claro, saludos

Buenas, amigo Octavio, es evidente que si se pueden borrar. El principal problema reside en que al instalar o **actualizar un kernel**, a veces se generan más archivos o se modifican configuraciones, que si no tienes los suficientes conocimientos pueden resultar difíciles de identificar.

En realidad no resulta nada práctico **borrar los kernel** de forma manual, cualquier error puede romper tu sistema de manera casi irremediable. Para **Ubuntu, Linux Mint** y derivados, te quiero proponer un método que consiste en limitar el número de núcleos antiguos que debe guardar tu sistema de forma automática.



Modificar los kernel que se guardan en Ubuntu

Si hablamos de Fedora, CentOS, Alma Linux y derivados. Modificar el número de Kernel que se guardan es tarea sencilla, tan solo debes editar el archivo correspondiente a tu administrador de paquetes (**dnf o yum**).

```
sudo nano /etc/dnf.conf
# 0
sudo nano /etc/yum.conf
```

Busca la siguiente línea y modifica el valor numérico, por los kernel que quieres que guarde el sistema.

```
installonly_limit=5
```

```
[main]
cachedir=/var/cache/yum/$basearch/$releasever
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
exactarch=1
obsoletes=1
ppgcheck=1
plugins=1
installonly_limit=5
bugtracker_url=http://bugs.centos.org/set_project.php?project_id=23&ref=http://bugs.centos.org/bug_report_page.php?category=yum
distroverpkg=centos-release

# This is the default, if you make this bigger yum won't see if the metadata
# is newer on the remote and so you'll "gain" the bandwidth of not having to
# download the new metadata and "pay" for it by yum not having correct
# information.
# It is esp. important, to have correct metadata, for distributions like
# Fedora which don't keep old packages around. If you don't like this checking
# interrupting your command line usage, it's much better to have something
# manually check the metadata once an hour (yum-updatesd will do this).
# metadata_expire=90m

PUT YOUR REPOS HERE OR IN separate files named file.repo
in /etc/yum.repos.d
```

En **Ubuntu, Linux Mint** y derivados el proceso es muy diferente. Existe un archivo de configuración llamado **«apt-auto-removal»**, en el cual puedes modificar los núcleos que quieres que se guarden en el sistema. Lamentablemente, este sistema resulta algo complejo y puede inducir a errores fatales. De todas formas vemos un ejemplo...

```
sudo nano /etc/kernel/postinst.d/apt-auto-removal
```

Busca la siguiente línea:

```
previous_version="$(echo "$debverlist" | sed -n 2p)"
```

Y la modificas por ejemplo así..., guarda el archivo y cierra el editor.

```
previous_version="$(echo "$debverlist" | sed -n 2,4p)"
```

De esta forma se guardarán 3 núcleos viejos, más el que usas actualmente. El archivo **«apt-auto-removal»**, es un **script bash**, por tanto los cambios no surtirán efecto hasta que lo ejecutas manualmente, o se instale un nuevo kernel.

```
sergio@solinux:~$ dpkg -l | grep linux-image
ii  Linux-image-5.0.0-32-generic          5.0.0-32.34-18.04.2   amd64   Signed kernel image gener
ic
ii  Linux-image-5.4.0-80-generic         5.4.0-80.99-18.04.1   amd64   Signed kernel image gener
ic
ii  Linux-image-5.4.0-91-generic         5.4.0-91.91-18.04.1   amd64   Signed kernel image gener
ic
ii  Linux-image-generic-hwe-18.04      5.4.0-91.91-18.04.73   amd64   Generic Linux kernel imag
e
sergio@solinux:~$
```

Yo te propongo otra opción que te resultara más atractiva, y no es otra que modificar el **script bash** con una variable que adjuntara el valor deseado. Abrimos el archivo con nano.

```
sudo nano /etc/kernel/postinst.d/apt-auto-removal
```

Al principio del script, agrega lo siguiente. En el primer valor, debes indicar los kernel antiguos que quieres guardar.

```
NKER=2
NKER=$((NKER+1))
```

En nuestro ejemplo, dos kernel + uno.

```
#!/bin/sh
set -e
# Mark as not-for-autoremoval those kernel packages that are:
# - the currently booted version
# - the kernel version we've been called for
# - the latest kernel version (as determined by debian version number)
# - the second-latest kernel version
#
# In the common case this results in two kernels saved (booted into the
# second-latest kernel, we install the latest kernel in an upgrade), but
# can save up to four. Kernel refers here to a distinct release, which can
# potentially be installed in multiple flavours counting as one kernel.
# Especifica la cantidad de núcleos que quieres conservar.
NKER=2
NKER=$((NKER+1))
eval $(apt-config shell APT_CONF_D Dir::Etc::parts/d)
test -n "${APT_CONF_D}" || APT_CONF_D="/etc/apt/apt.conf.d"
config_file="${APT_CONF_D}/01autoremove-kernels"
eval $(apt-config shell DPKG Dir::bin::dpkg/f)
test -n "$DPKG" || DPKG="/usr/bin/dpkg"
```

Ahora buscas la línea...

```
previous_version="$(echo "$debverlist" | sed -n 2p)"
```

Y la sustituyes por esta otra.

```
previous_version="$(echo "$debverlist" | sed -n 2,${NKER}p)"
```

Guardas el archivo y cierra el **editor nano**. Recuerda que los nuevos valores no surtirán efecto, hasta que no ejecutes el script manualmente o no se instale un nuevo kernel.

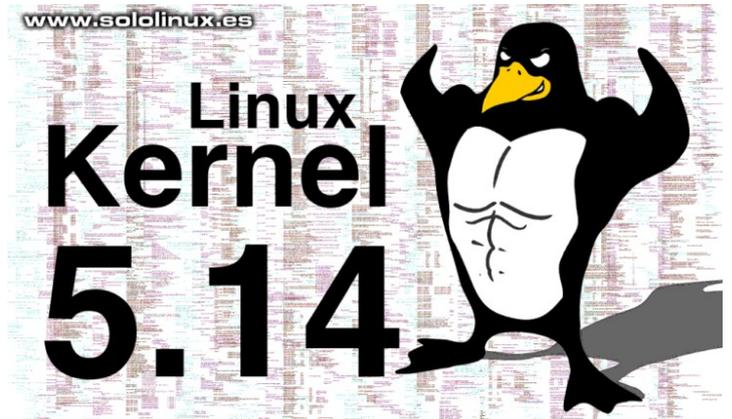
Hace tiempo que este script no sufre ningún cambio, es el mismo en las últimas versiones LTS de **Ubuntu, Mint** y derivados. Por tanto... si quieres puedes sustituir el script original de tu sistema, por uno exactamente igual, pero ya modificado con los añadidos que hemos comentado anteriormente. Copia y pega lo siguiente.

```
#!/bin/sh
set -e
# Mark as not-for-autoremoval those kernel packages that are:
# - the currently booted version
# - the kernel version we've been called for
# - the latest kernel version (as determined by debian version number)
# - the second-latest kernel version
#
# In the common case this results in two kernels saved (booted into the
# second-latest kernel, we install the latest kernel in an upgrade), but
# can save up to four. Kernel refers here to a distinct release, which can
# potentially be installed in multiple flavours counting as one kernel.
# Especifica la cantidad de núcleos que quieres conservar.
NKER=2
NKER=$((NKER+1))
eval $(apt-config shell APT_CONF_D Dir::Etc::parts/d)
test -n "${APT_CONF_D}" || APT_CONF_D="/etc/apt/apt.conf.d"
config_file="${APT_CONF_D}/01autoremove-kernels"
eval $(apt-config shell DPKG Dir::bin::dpkg/f)
test -n "$DPKG" || DPKG="/usr/bin/dpkg"
list="$(dpkg -l | awk '/^[ih][^c][ ]+(linux|kfreebsd|gnumach)-image-[0-9]+\./
&& $2 !~ /-dbg(?:.*)?$/ && $2 !~ /-dbg(?:.*)?$/ { print $2,$3; }' \
| sed -e 's#^(linux|kfreebsd|gnumach)-image-##' -e 's#:[^:]+\# #')
debverlist="$(echo "$list" | cut -d' ' -f 2 | sort --unique --reverse --version-
sort)"
if [ -n "$list" ]; then
    installed_version="$(echo "$list" | awk "\$1 == \"\$1\" { print \$2;exit; }")"
fi
uname_r="$(uname -r | tr '[:A-Z]' '[:a-z]')
if [ -n "$uname_r" ]; then
    running_version="$(echo "$list" | awk "\$1 == \"\$uname_r\" { print \$2;exit; }")"
fi
# ignore the currently running version if attempting a reproducible build
if [ -n "${SOURCE_DATE_EPOCH}" ]; then
    uname_r=""
    running_version=""
fi
latest_version="$(echo "$debverlist" | sed -n 1p)"
previous_version="$(echo "$debverlist" | sed -n 2,${NKER}p)"
debkernels="$(echo "$latest_version
$installed_version
$running_version
$previous_version" | sort -u | sed -e '/^$/ d')
kernels="$( (echo "$1
$uname_r"; for deb in $debkernels; do echo "$list" | awk "\$2 == \"\$deb\" { print \$
1; }; done; ) \
| sed -e 's#([^\.\+])#\1#i#g' -e '/^$/ d' | sort -u)"
generateconfig() {
    cat <<EOF
// DO NOT EDIT! File autogenerated by $0
APT::NeverAutoRemove
{
EOF
    for package in $(apt-config dump --no-empty --format '%v\n'
'APT::VersionedKernelPackages'); do
        for kernel in $kernels; do
            echo " \${package}=${kernel}";
        done
    done
    echo '};'
    if [ "${APT_AUTO_REMOVAL_KERNELS_DEBUG:-true}" = 'true' ]; then
        cat <<EOF
/* Debug information:
# dpkg list:
$(dpkg -l | grep '\(linux|kfreebsd|gnumach)-image-')
# list of installed kernel packages:
$list
# list of different kernel versions:
$debverlist
# Installing kernel: $installed_version ($1)
# Running kernel: ${running_version:-ignored} (${uname_r:-ignored})
# Last kernel: $latest_version
# Previous kernel: $previous_version
# Kernel versions list to keep:
$debkernels
# Kernel packages (version part) to protect:
$kernels
*/
EOF
    fi
}
generateconfig "$@" > "${config_file}.dpkg-new"
mv -f "${config_file}.dpkg-new" "${config_file}"
chmod 444 "${config_file}"
```

El Linux Kernel 5.14 ha sido liberado

Después de dos meses de desarrollo, y cuando hace apenas 4 días que celebramos el **30 aniversario del lanzamiento del Kernel Linux**; **Linus Torvalds** nos sorprende con el lanzamiento del nuevo **kernel 5.14**. Entre sus cambios más notables, destacamos las nuevas llamadas al sistema **quotactl_fd ()** y **memfd_secret ()**, la eliminación de algunos controladores IDE y RAW, se añade un nuevo controlador de prioridad I/O para **cgroup**, así como un nuevo modo de programación de tareas **SCHED_CORE**.

Esta versión ha recibido la increíble cifra de 15883 correcciones, con un tamaño de 69 MB. Las modificaciones afectan a 12,580 archivos, se agregaron 861501 nuevas líneas de código y se han eliminado 321,654 líneas. El 47% de los cambios introducidos en esta versión, están relacionados con los controladores de dispositivos. Esto nos indica que viene con un soporte para nuevos dispositivos, realmente impresionante.



El Linux Kernel 5.14 ha sido liberado

El **kernel 5.14** tiene buenas noticias para los usuarios de los **ARM de 64 bits (AArch64)**, ya que el módulo de virtualización **KVM** (máquina virtual basada en kernel) ahora admite la extensión de etiquetado de memoria ARM64 en invitados y, también posibilita configurar la autenticación **ARM64** de forma independiente en el kernel y, para el usuario.

Otro cambio importante en este Kernel, es la eliminación de **controladores IDE** antiguos, pero no te preocupes... seguirán funcionando, pero a partir de ahora los manejará el subsistema **libATA**. El controlador de gráficos **AMDGPU** para las **GPU AMD**, también ha mejorado de forma considerable.

El nuevo mecanismo de **socket SO_REUSEPORT** permite más control, sobre como manejar los fallos de las herramientas y aplicaciones. Se añade soporte para el sistema de archivos **F2FS**, ahora se permite comprimir archivos mapeados en **mmap ()**. En el sistema de archivos **EXT4**, se permite forzar las transacciones pendientes para evitar las fugas de información.

El nuevo **Linux 5.14** detecta la fusión inicial de la infraestructura, para los cargadores de software BPF, el soporte **KFENCE** y el soporte para páginas transparentes en la arquitectura **RISC-V**. Otro añadido interesante, consiste en la fusión del conjunto de parches del pulsador de interrupción del grupo de control, así se logra el poder eliminar rápidamente a todos los presentes de un grupo de control. En fin... si tienes un equipo muy moderno, y dado los numerosos nuevos controladores, también por otros actualizados, vale la pena que actualices al nuevo Kernel si quieres tener un soporte de hardware de primer nivel.

Puedes descargar esta nueva versión, desde su página oficial:

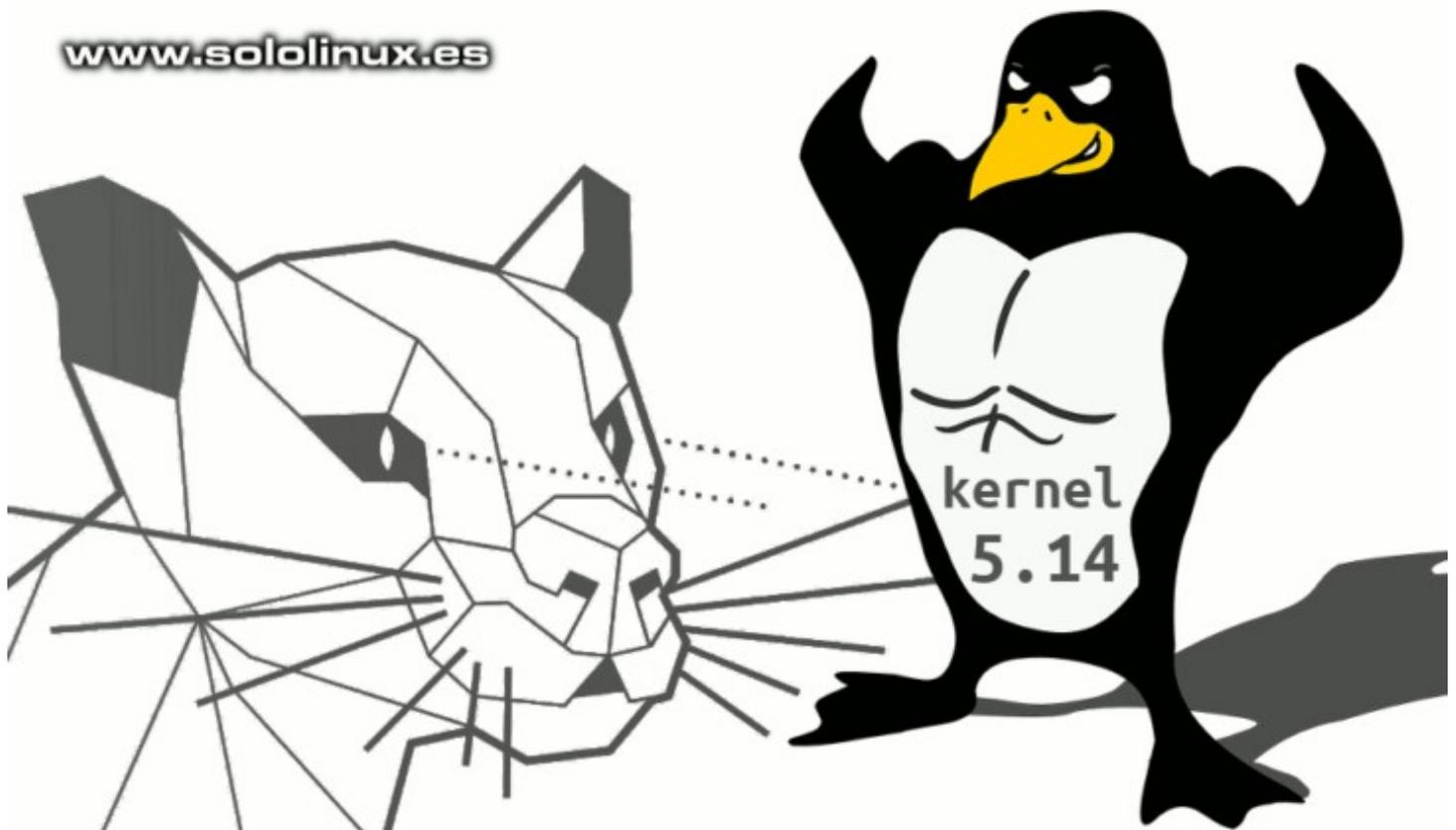
- [Descargar Linux Kernel 5.14](#)

Protocol	Location	Latest Release	
HTTP	https://www.kernel.org/pub/	5.14	
GIT	https://git.kernel.org/		
RSYNC	rsync://rsync.kernel.org/pub/		
www.sololinux.es			
mainline:	5.14	2021-08-29	[tarball] [pgp] [patch] [view diff] [browse]
stable:	5.13.13	2021-08-26	[tarball] [pgp] [patch] [inc. patch] [view diff] [browse] [changelog]
longterm:	5.10.61	2021-08-26	[tarball] [pgp] [patch] [inc. patch] [view diff] [browse] [changelog]
longterm:	5.4.143	2021-08-26	[tarball] [pgp] [patch] [inc. patch] [view diff] [browse] [changelog]
longterm:	4.19.205	2021-08-26	[tarball] [pgp] [patch] [inc. patch] [view diff] [browse] [changelog]
longterm:	4.14.245	2021-08-26	[tarball] [pgp] [patch] [inc. patch] [view diff] [browse] [changelog]
longterm:	4.9.281	2021-08-26	[tarball] [pgp] [patch] [inc. patch] [view diff] [browse] [changelog]
longterm:	4.4.282	2021-08-26	[tarball] [pgp] [patch] [inc. patch] [view diff] [browse] [changelog]



LINUX KERNEL 5.14

Instalar el Kernel 5.14 en Ubuntu y derivados



En el **anterior artículo**, ya tratamos las bondades y beneficios del nuevo **Linux Kernel 5.14**. Como es lógico en este tipo de versiones, nos presenta una avalancha de mejoras, novedades y un soporte de hardware muy mejorado.

En **SoloLinux** siempre intentamos ir por delante, así que como es habitual, hemos realizado una instalación limpia de **Ubuntu 18.04** para intentar instalar este nuevo Kernel. Éramos conscientes del riesgo, pues ya sabíamos a ciencia cierta que en **Ubuntu 20.04** sí que era posible; Pero claro... Ubuntu 18.04 viene con un Linux Kernel 5.4 y el salto es enorme.

La instalación ha sido un éxito rotundo, su forma de operar ha mejorado partes sensibles del sistema (sobre todo en controladores). Por tanto... podemos afirmar que este artículo es efectivo en **Ubuntu** (a partir de la versión 18.04), **Linux Mint** y todos sus derivados. Comenzamos.

Instalar el Kernel 5.14 en Ubuntu y derivados

Descargar los archivos deb del kernel es tarea sencilla, **Canonical** nos lo pone muy fácil, tan solo debes ejecutar los comandos que te propongo en la terminal de tu sistema. Usamos el **comando wget** para descargar los archivos necesarios, en la carpeta **/tmp**, pero antes verificamos el Kernel instalado en el sistema.

```
uname -a
```

```
sergio@demo-sololinux:~$ uname -a
Linux demo-sololinux 5.4.0-81-generic #91~18.04.1-Ubuntu SMP Fri Jul 23 13:36:29 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
sergio@demo-sololinux:~$
```

Bueno, ya sabemos que nuestro Kernel es el 5.4. Abrimos la carpeta **/tmp** y descargamos los archivos.

```
cd /tmp
wget -c https://kernel.ubuntu.com/~kernel-ppa/mainline/v5.14/amd64/linux-headers-5.14.0-051400-generic_5.14.0-051400.202108292331_amd64.deb
wget -c https://kernel.ubuntu.com/~kernel-ppa/mainline/v5.14/amd64/linux-headers-5.14.0-051400_5.14.0-051400.202108292331_all.deb
wget -c https://kernel.ubuntu.com/~kernel-ppa/mainline/v5.14/amd64/linux-image-unsigned-5.14.0-051400-generic_5.14.0-051400.202108292331_amd64.deb
wget -c https://kernel.ubuntu.com/~kernel-ppa/mainline/v5.14/amd64/linux-modules-5.14.0-051400-generic_5.14.0-051400.202108292331_amd64.deb
```

El proceso es muy rápido.

```

Actividades Terminal mar 10:01 sergio@demo-sololinux: /tmp
sergio@demo-sololinux:~$ cd /tmp
sergio@demo-sololinux:~/tmp$ wget -c https://kernel.ubuntu.com/~kernel-ppa/mainline/v5.14/amd64/linux-headers-5.14.0-051400-generic_5.14.0-051400.202108292331_amd64.deb
--2021-08-31 09:59:55-- https://kernel.ubuntu.com/~kernel-ppa/mainline/v5.14/amd64/linux-headers-5.14.0-051400-generic_5.14.0-051400.202108292331_amd64.deb
Resolviendo kernel.ubuntu.com (kernel.ubuntu.com)... 91.189.94.216
Conectando con kernel.ubuntu.com (kernel.ubuntu.com)[91.189.94.216]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 2625706 (2,5M) [application/x-debian-package]
Guardando como: "linux-headers-5.14.0-051400-generic_5.14.0-051400.202108292331_amd64.deb"

linux-headers-5.14.0-051400-generic_5.14.0-051400.202108292331_amd64.deb
100%[=====] 2,50M 5,55MB/s en 0,5s
2021-08-31 09:59:55 (5,55 MB/s) - "linux-headers-5.14.0-051400-generic_5.14.0-051400.202108292331_amd64.deb" guardado [2625706/2625706]

sergio@demo-sololinux:~/tmp$ wget -c https://kernel.ubuntu.com/~kernel-ppa/mainline/v5.14/amd64/linux-headers-5.14.0-051400_5.14.0-051400.202108292331_all.deb
--2021-08-31 10:00:16-- https://kernel.ubuntu.com/~kernel-ppa/mainline/v5.14/amd64/linux-headers-5.14.0-051400_5.14.0-051400.202108292331_all.deb
Resolviendo kernel.ubuntu.com (kernel.ubuntu.com)... 91.189.94.216
Conectando con kernel.ubuntu.com (kernel.ubuntu.com)[91.189.94.216]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 12086228 (12M) [application/x-debian-package]
Guardando como: "linux-headers-5.14.0-051400_5.14.0-051400.202108292331_all.deb"

linux-headers-5.14.0-051400_5.14.0-051400.202108292331_all.deb
100%[=====] 11,53M 8,34MB/s en 1,4s
2021-08-31 10:00:18 (8,34 MB/s) - "linux-headers-5.14.0-051400_5.14.0-051400.202108292331_all.deb" guardado [12086228/12086228]

sergio@demo-sololinux:~/tmp$ wget -c https://kernel.ubuntu.com/~kernel-ppa/mainline/v5.14/amd64/linux-headers-5.14.0-051400-generic_5.14.0-051400.202108292331_amd64.deb
--2021-08-31 10:00:37-- https://kernel.ubuntu.com/~kernel-ppa/mainline/v5.14/amd64/linux-headers-5.14.0-051400-generic_5.14.0-051400.202108292331_amd64.deb
Resolviendo kernel.ubuntu.com (kernel.ubuntu.com)... 91.189.94.216
Conectando con kernel.ubuntu.com (kernel.ubuntu.com)[91.189.94.216]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK

```

Sin salir de /tmp, ejecuta el siguiente comando para instalar el nuevo Kernel.

```
sudo dpkg -i *.deb
```

El proceso de instalación, apenas tarda unos pocos minutos.

```

sergio@demo-sololinux:~/tmp$ sudo dpkg -i *.deb
[sudo] contraseña para sergio:
Seleccionando el paquete linux-headers-5.14.0-051400 previamente no seleccionado.
(Leyendo la base de datos ... 153201 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar linux-headers-5.14.0-051400_5.14.0-051400.202108292331_all.deb ...
Desempaquetando linux-headers-5.14.0-051400 (5.14.0-051400.202108292331) ...
Seleccionando el paquete linux-headers-5.14.0-051400-generic previamente no seleccionado.
Preparando para desempaquetar linux-headers-5.14.0-051400-generic_5.14.0-051400.202108292331_amd64.deb ...
Desempaquetando linux-headers-5.14.0-051400-generic (5.14.0-051400.202108292331) ...
Seleccionando el paquete linux-image-unsigned-5.14.0-051400-generic previamente no seleccionado.
Preparando para desempaquetar linux-image-unsigned-5.14.0-051400-generic_5.14.0-051400.202108292331_amd64.deb ...
Desempaquetando linux-image-unsigned-5.14.0-051400-generic (5.14.0-051400.202108292331) ...
Seleccionando el paquete linux-modules-5.14.0-051400-generic previamente no seleccionado.
Preparando para desempaquetar linux-modules-5.14.0-051400-generic_5.14.0-051400.202108292331_amd64.deb ...
Desempaquetando linux-modules-5.14.0-051400-generic (5.14.0-051400.202108292331) ...
Configurando linux-headers-5.14.0-051400 (5.14.0-051400.202108292331) ...
.....etc.....etc.....

```

Una vez concluya el proceso, reiniciemos nuestro sistema Linux.

```
sudo reboot
```

Cuando la máquina haya terminado de iniciar, verificamos la nueva versión de Kernel que tenemos instalada.

```
uname -a
```

La operación ha sido un éxito, observa nuestro ejemplo de salida.

```

sergio@demo-sololinux:~$ uname -a
Linux demo-sololinux 5.14.0-051400-generic #202108292331 SMP Sun Aug 29 23:33:39 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
sergio@demo-sololinux:~$

```

Otras opciones para verificar la versión instalada.

```

sergio@demo-sololinux:~$ uname -r
5.14.0-051400-generic
sergio@demo-sololinux:~$ uname -a
Linux demo-sololinux 5.14.0-051400-generic #202108292331 SMP Sun Aug 29 23:33:39
UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
sergio@demo-sololinux:~$

```

Nota: Existe un error conocido con **Libc6** (Ubuntu 18.04), pero Canonical ya está trabajando en actualizar esa librería de tiempo de ejecución. De todas formas, no afecta al rendimiento del sistema.

