

SOLOLINUX

MAGAZINE

Visita nuestro BLOG: www.sololinux.es

USA
ESTUDIA
DISTRIBUYE Y MEJORA



Y MUCHOS MAS MANUALES
EN EL INTERIOR

Instalar Lamp en Ubuntu,
CentOS y derivados

Instalar Node.js 8, 10 o 11
en Ubuntu y derivados

Borrar archivos y carpetas en
Linux desde la consola

Instalar nuevos tipos de
letra en Linux



Uso de la herramienta
cURL

REVISTA DE FEBRERO, ENTRA
EN WWW.SOOLINUX.ES Y
ENCUENTRA MUCHO MAS



SoloLinux

"Los que pueden, lo hacen; los que no, sólo saben quejarse."

Linus Torvalds



SoloLinux Magazine
No. 1
Febrero 2019
81 Paginas

Blog:
www.sololinux.es

Redacción y publicación:
Sergio G. B.
Sergio@sololinux.es

Diseño y maquetación:
Adrián A. A.
adrian@sololinux.es

Publicidad:
Quieres aparecer en la revista, escríbenos a adrian@sololinux.es

Colabora:
Quieres colaborar con algún articulo, mándalo a
adrian@sololinux.es
Y si gusta será publicado.

Revista de distribución gratuita, comparte conocimientos.



 **MANUALES**

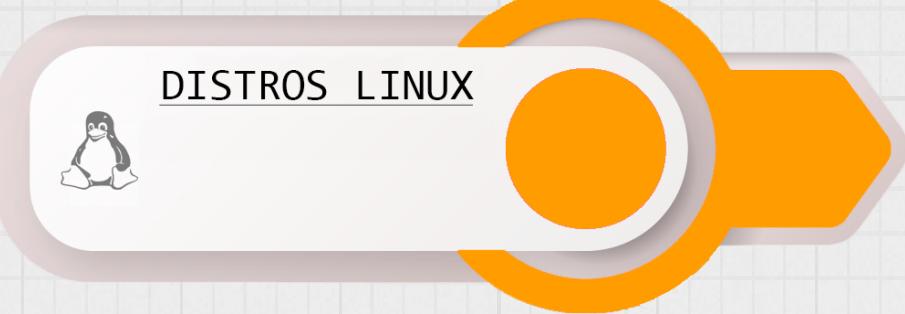
- [**08. Uso de la herramienta cURL**](#)
- [**11. Borrar archivos y carpetas en Linux desde la consola**](#)
- [**14. Instalar nuevos tipos de letra en Linux**](#)
- [**18. Configurar las DNS en VestaCP correctamente**](#)
- [**25. Añadir registros DKIM y SPF a un dominio en VestaCP**](#)
- [**31. Cyberpanel: El mejor panel de control web gratis**](#)
- [**36. Instalar PHP 7.3 en Ubuntu, Debian y derivados**](#)
- [**44. Instalar Opera en Fedora, OpenSuse, CentOS y derivados**](#)
- [**57. Como instalar Xampp en Linux**](#)
- [**63. Averiguar el códec de un archivo multimedia con MediaInfo**](#)

 **SCRIPTS**

- [**03. Script Bash: Instalar Lamp en Ubuntu, CentOS y derivados**](#)
- [**06. Script Bash: Instalar Node.js 8, 10 o 11 en Ubuntu y derivados**](#)
- [**66. Los script bash que necesita un SysAdmin**](#)

 **SOFTWARE Y HARDWARE**

- [**29. Instalar Visual Studio Code en Ubuntu y derivados**](#)
- [**69. Ejecutar Webex en Ubuntu 18.04**](#)

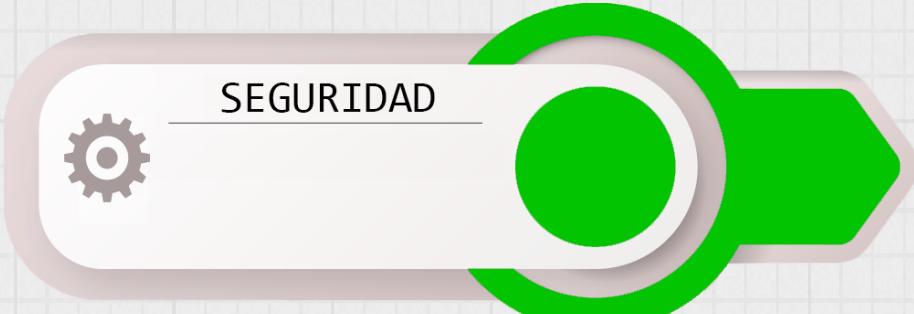


DISTRROS LINUX



[**38. Instalar Arch linux paso a paso**](#)

[**46. Instalación y uso de COCKPIT en tu Servidor**](#)



SEGURIDAD



[**52. Las reglas de iptables que debes conocer**](#)

[**73. Proteger WordPress con Fail2ban**](#)

[**75. Permisos de archivos en Linux**](#)

[**77. Uso del comando chmod**](#)



REDES



Limitar el ancho de banda con Wondershaper



Designed by freepik
Designed by pikisuperstar



Script bash:

Instalar Lamp en Ubuntu, CentOS y derivados

Script bash: Instalar [Lamp](#) en [Ubuntu](#), [Debian](#), [CentOS](#) y derivados.

LAMP es uno de los entornos de servidor web más populares, pero tiene un inconveniente para los usuarios más noveles, recordemos que un **server web** completo requiere de muchos [comandos](#) hasta lograr un entorno de trabajo que sea estable y veloz.

La solución es [LAMP One-Click](#).



LAMP One-Click es un potente [script bash](#) que realiza la instalación de **Apache + PHP + MySQL, MariaDB o Percona Server** de forma prácticamente automática, tan solo requiere que selecciones las versiones a instalar.

Es compatible con las siguientes [distribuciones Linux](#) (y sus derivados):

- **CentOS-6.x**
- **CentOS-7.x**
- **Fedora-29**
- **Debian-8.x**
- **Debian-9.x**
- **Ubuntu-14.x**
- **Ubuntu-16.x**
- **Ubuntu-18.x**

A día de hoy (1-Febrero-2019), las versiones de las aplicaciones que se instalan, son estas (se actualizan constantemente):

- **Apache 2.4**, incluyendo los módulos de **HTTP2: [nghttp2](#), [mod http2](#)**.
- Ofrece la opción de instalar los Módulos adicionales de **Apache: [mod wsgi](#), [mod security](#), [mod jk](#)**.
- **Bases de datos:** MySQL-5.5, MySQL-5.6, MySQL-5.7, MySQL-8.0, MariaDB-5.5, MariaDB-10.0, MariaDB-10.1, MariaDB-10.2, MariaDB-10.3, Percona-Server-5.5, Percona-Server-5.6, Percona-Servidor 5.7, Percona Servidor 8.0.
- **Versiones de php:** PHP-5.6, PHP-7.0, PHP-7.1, PHP-7.2 y PHP-7.3.
- **Módulos adicionales de PHP que permite instalar:** [Zend OPcache](#), [ionCube Loader](#), [XCache](#), [imagick](#), [gmagick](#), [libsodium](#), [memcached](#), [redis](#), [mongodb](#), [swoole](#), [xdebug](#).
- Otras aplicaciones que deberías instalar:
[OpenSSL](#), [ImageMagick](#), [GraphicsMagick](#), [Memcached](#), [phpMyAdmin](#), [Redis](#), [KodExplorer](#).



Instalar LAMP en Ubuntu, CentOS y derivados

Instalar LAMP en Ubuntu, Debian, CentOS y derivados es muy sencillo con este [script](#). Tan solo debes seguir los pasos de instalación que te indico, así lo mismo con los de configuración (en imágenes).

Abrimos la consola / terminal (recuerda que si estas en [Ubuntu y no eres root](#), debes agregar “[sudo](#)” antes de cada linea de comandos) y comenzamos.

En **CentOS, Fedora** y derivados:

```
1 yum -y install wget screen git
2 git clone https://github.com/teddysun/lamp.git
3 cd lamp
4 chmod 755 *.sh
5 screen -S lamp
```

En **Debian, Ubuntu** y derivados:

```
1 apt-get -y install wget screen git
2 git clone https://github.com/teddysun/lamp.git
3 cd lamp
4 chmod 755 *.sh
5 screen -S lampv
```

Para comenzar la instalación, independientemente de la [distribución linux](#) que estés usando, ejecuta lo siguiente:

```
1 ./lamp.sh
```

Arranca la instalación, pero debemos seleccionar las aplicaciones que necesitamos, y sus versiones.

Primero nos aparece los datos del server o vps, y

nos pregunta si queremos instalar **Apache**.

Normalmente seleccionas el “**I**“.

The screenshot shows the initial steps of the ./lamp.sh script. It displays system information such as CPU model (Intel(R) Xeon(R) CPU E3-1230 v6 @ 3.50GHz), number of cores (1), CPU frequency (3762.329 MHz), total amount of ram (512 MB), total amount of swap (512 MB), system uptime (1 days, 3:50:45), load average (0.08, 0.21, 0.38), OS (Debian GNU/Linux 9), Arch (x86_64 (64 Bit)), Kernel (4.9.0), Hostname (host.linuxmail.es), and IPv4 address (185.185.42.63). Below this, it asks if you want to install Apache, with options 1) httpd-2.4.38 and 2) do_not_install. The default option is selected (httpd-2.4.38).

Segundo paso.

Nos dice si queremos instalar algún modulo extra. Si no quieres ninguno escribe el numero “**4**“.

The screenshot shows the apache_modules_install step. It lists several modules: mod_wsgi-4.6.5, mod_security-2.9.3, mod_jk-1.2.46, and do_not_install. A message at the bottom says "Please input one or more number between 1 and 4 (default do_not_install) (for example 1 2 3): 1 2 3".

Seleccionamos la versión de [MySQL](#), [MariaDB](#) o [Percona server](#) que nos interese y pulsamos la tecla “[Enter](#)“.

Seguidamente nos pregunta la ruta, recomiendo encarecidamente que dejes la predeterminada.

The screenshot shows the mysql_setting step. It lists various MySQL/MariaDB versions: mysql-5.5.62, mysql-5.6.43, mysql-5.7.25, mysql-8.0.14, mariadb-5.5.62, mariadb-10.0.37, mariadb-10.1.37, mariadb-10.2.21, mariadb-10.3.12, Percona-Server-5.5.62-38.14, Percona-Server-5.6.43-84.3, Percona-Server-5.7.24-27, Percona-Server-8.0.13-4, and do_not_install. A message at the bottom says "which mysql you'd select (default mysql-5.7.25): 8". The selection is set to mariadb-10.2.21.

También nos consulta si queremos modificar la contraseña, por defecto es la del usuario “**root**“, y si necesitamos [phpMyAdmin](#), tu decides.

The screenshot shows the configuration of MariaDB. It asks for the data location (mariadb data location: /usr/local/mariadb/data), the server root password (mariadb server root password (default:lamp.sh, leave blank for default):), and the server root password again (mariadb server root password: lamp.sh).

Seleccionamos la **versión PHP** que queremos usar, y añadimos los [módulos](#) que nos resulten necesarios.

Al aceptar, nos preguntara si queremos instalar alguna de las aplicaciones opcionales de las que dispone: [OpenSSL](#), [ImageMagick](#), [GraphicsMagick](#), [Memcached](#), [phpMyAdmin](#), [Redis](#) y [KodExplorer](#). Instala la que necesites.

```

----- php setting -----
1) php-5.6.40
2) php-7.0.33
3) php-7.1.26
4) php-7.2.14
5) php-7.3.1
6) do_not_install

which php you'd select (default php-7.2.14):
your selection: php-7.2.14

php-7.2.14 available modules:

----- php_modules install -----

1) ioncube_loaders
2) php-imagick-3.4.3
3) php-gmagick-2.0.5RC1
4) php-memcached-3.1.3
5) php-redis-4.2.0
6) php-mongodb-1.5.3
7) libsodium-php-2.0.20
8) php-swoole-4.2.12
9) xdebug-2.7.0beta1
10) do_not_install

Please input one or more number between 1 and 10 (default do_not_install) (for example: 1 2 3): 1 2 3 4

```

Ya lo tenemos.

Es el final, por tanto nos imprime en pantalla nuestra selección.

Pulsa la tecla “Enter” para comenzar la instalación.

```

----- Install Overview -----
Apache: httpd-2.4.38
Apache Location: /usr/local/apache

Database: mariadb-10.2.21
MariaDB Location: /usr/local/mariadb
MariaDB Data Location: /usr/local/mariadb/data
MariaDB Root Password: lamp.sh

PHP: php-7.2.14
PHP Location: /usr/local/php
PHP Additional Modules:
ioncube_loaders
imagick-3.4.3
gmagick-2.0.5RC1
memcached-3.1.3

phpMyAdmin: phpMyAdmin-4.8.5-all-languages
phpMyAdmin Location: /data/www/default/phpmyadmin

KodExplorer: kodfile-4.35
KodExplorer Location: /data/www/default/kod

----- 

Press any key to start...or Press Ctrl+C to cancel
[Info] Starting to install development tools...
[Info] Starting to install package gcc

```

La instalación comienza, se paciente dependiendo de tu sistema puede tardar más o menos.

Espero que te resulte de utilidad este **Script Bash**.

Script Bash: Instalar Node.js 8, 10 ó 11 en Ubuntu y derivados



Script Bash: Instalar [Node.js](#) 8, 10 ó 11 en [Ubuntu](#), [Debian](#) y derivados.

Node.js es un entorno de desarrollo y trabajo en tiempo de ejecución de [código abierto](#), que trabaja en una capa del servidor. Esta basado en [ECMAScript](#).

Destacamos que trabaja en asíncrono, y hace uso de una [I/O](#) con una [arquitectura orientada a eventos](#), a la que añadimos el beneficio de que está basado en el motor [V8](#) de [Google](#).

Fue desarrollado por [Ryan Dahl](#), para la creación de aplicaciones en red escalables, por ejemplo [servidores web](#).

Hoy vemos como **instalar Node.js** en sus versión 8,10 ó 11 mediante un [script bash](#) automáticamente.

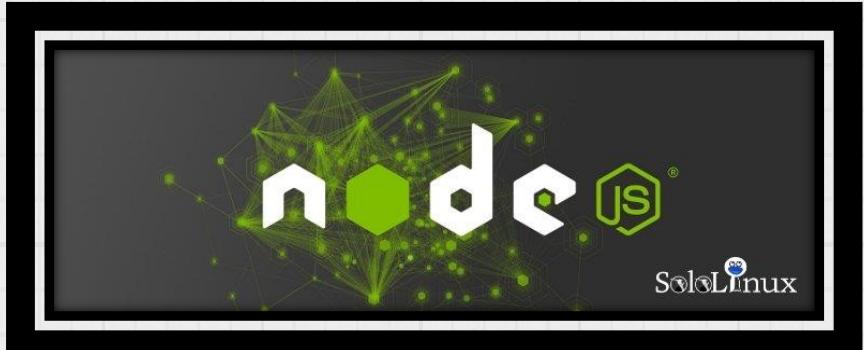
Su uso es... como siempre
súper-simple.



Solo sigue los pasos que te indico.

Instalar Node.js 8,9 ó 10 en Ubuntu

Creamos el ejecutable.



```
1 nano install-nodejs.sh
```

Copia lo siguiente.

```

1 #!/bin/bash
2 # /install-nodejs.sh
3 # Developed by sololinux.es
4 # Sergio G.B.
5 #
6
7 nodejs.sh() {
8     # Show options
9     echo -e "
10     ${txtblu}
11     What version you want to install?
12     ${txtrst}
13     1) Node.js 8
14     2) Node.js 10
15     3) Node.js 11
16     ";
17
18     # Get value
19     read version
20
21     # Case
22     case $version in
23         # Performs the function with the name of the variable passed
24         1) node8;;
25         2) node10;;
26         3) nodell;;
27         *) nodejs.sh;;
28     esac
29 }
30
31 node8() {
32     curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -
33     sudo apt-get install -y nodejs;
34     clear;
35 }
36
37 node10() {
38     curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -
39     sudo apt-get install -y nodejs;
40     clear;
41 }
42
43 nodell() {
44     curl -sL https://deb.nodesource.com/setup_11.x | sudo -E bash -
45     sudo apt-get install -y nodejs;
46     clear;
47 }
```

Guarda el archivo y cierra el editor.

Ahora le damos permisos.

1 chmod 755 install-nodejs.sh

Y ejecutamos cualquiera de los dos **comandos** siguientes:

1 ./install-nodejs.sh

1 bash install-nodejs.sh

Espero te resulte útil este **script bash**.

MANUAL: Uso de la herramienta cURL



SoloLinux

[Visita su github.](#)

Uso de la [herramienta cURL](#).

Hoy hablaremos de la herramienta [cURL](#), es necesario que la conozcas, pues resulta de mucha utilidad si lo que quieras es descargar o transferir datos a un servidor.

La [herramienta cURL](#) tiene una particularidad que la distingue de otras, no requiere de la interacción de un usuario para su correcto funcionamiento.

[Daniel Stenberg](#) fue el desarrollador del proyecto [cURL](#) (lo creó como un bot que actualizaba las tasas de cambio de monedas), actualmente su comunidad es grande, pero Daniel sigue siendo el principal baluarte sobre el que se basa el continuo desarrollo de la herramienta.

Uno de sus puntos fuertes es la multitud de protocolos que admite, son los siguientes: DICT, FILE, FTP, FTPS, GOPHER, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, POP3, POP3S, RTMP, RTSP, SCP, SFTP, SMB, SMBS, SMTP, SMTPS, TELNET y TFTPL. Además admite los certificados SSL, HTTP POST, HTTP PUT, y permite la carga por FTP, por HTTP basada en formularios, proxies, HTTP/2, cookies, etc...

Estamos ante una de las herramientas más usadas por los desarrolladores, lo que no le resta robustez y seguridad.

En este artículo veremos las formas de uso más comunes (con ejemplos), seguro que alguna te resultará útil en el desarrollo de tu aplicación.

Uso de la herramienta cURL

Dependiendo de la versión de tu [distribución linux](#), tal vez no esté instalada por defecto.

Debian, Ubuntu y derivados:

```
1 apt-get install curl
```

CentOS derivados:

```
1 yum install curl
```

Puedes comprobar la **versión cURL** instalada, con:

```
1 curl --version
```

Ejemplo...

```
[root@host ~]# curl -version
curl 7.29.0 (x86_64-redhat-linux-gnu) libcurl/7.29.0 NSS/3.36 zlib/1.2.7 libidn/1.28 libssh2/1.4.3
Protocols: dict file ftp ftps gopher http https imap imaps ldap ldaps pop3 pop3s rtsp scp sftp smtp smt
telnet tftp
Features:AsynchDNS GSS-Negotiate IDN IPv6 Largefile NTLM NTLM_WB SSL libz unix-sockets
```

Descargar un archivo:

La descarga de un archivo nos ofrece dos opciones...

- **O** – Guarda el archivo en la carpeta actual y con su nombre original.
- **o** – Permite indicar la carpeta donde se guardara el archivo, y modificar el nombre de la descarga.

O

```
1 curl -O
```

Ejemplo...

```
curl -O http://domain.com/archivo.tar.gz
```

o

```
1 curl -o
```

Ejemplo...

```
curl -o nuevo-nombre-de-archivo.tar.gz http://domain.com/archivo.tar.gz
```

Descargar varios archivos a la vez:

Simplemente añadiendo url's a continuación de la opción, puedes descargar varios archivos a la vez.

```
1 curl -O "URL" -O "URL" -O "URL"
```

ejemplo...

```
curl -O http://domain.com/readme.html -O http://domain.com/creditos.txt
```

Descargar varios archivos a la vez desde una lista de url's:

Si hacemos uso de la herramienta **cURL** junto con el comando “**xargs**”, nos permitirá descargar las direcciones **url** de un listado, por ejemplo “**archivos-a-descargar.txt**”.

```
1 xargs -n 1 curl
```

ejemplo...

```
xargs -n 1 curl -O < archivos-a-descargar.txt
```

LIMITAR LA VELOCIDAD de descarga:

En el ejemplo limitaremos la tasa a 500Kb/s.

```
1 curl --limit-rate
```

Ejemplo...

```
curl -limit-rate 500K http://domain.com/archivo.tar.gz -O
```

CONSULTAR ENCABEZADOS HTTP:

Hacer una consulta sobre los encabezados HTTP, nos aportara información detallada sobre como de procesa nuestra solicitud. Muy útil a nivel de seguridad.

```
1 curl -I
```

Ejemplo...

```
curl -I dirección-web
```

```
[root@host ~]# curl -I www.adminserver.es
HTTP/1.1 301 Moved Permanently
Server: nginx
Date: Mon, 04 Feb 2019 11:16:32 GMT
Content-Type: text/html
Content-Length: 178
Connection: keep-alive
Location: https://adminserver.es/
```

**DESCARGAR ARCHIVOS POR FTP**:

Donde “**username:password**” debes ingresar tu usuario y la **password**, si el FTP admite usuarios anónimos... simplemente lo omites.

```
1 curl -u username:password -O ftp://server-ftp/archivo.tar.gz
```

CARGAR ARCHIVOS POR FTP:

Donde “**username:password**” debes ingresar tu usuario y la **password**, si el FTP admite usuarios anónimos... simplemente lo omites.

```
1 curl -u username:password -T archivo.tar.gz ftp://server-ftp
```

MODIFICAR EL AGENTE DE USUARIO:

El agente de usuario indica con que **navegador web** se conecta un cliente.

```
1 curl -I http://localhost-o-URL --user-agent "Hola soy un agente de usuario"
```

VISUALIZAR LAS COOKIES QUE SE DESCARGAN AL VISITAR UN SITIO WEB:

Generamos el archivo “**cookies.txt**” y lo visualizaremos con el comando “**cat**”.

```
1 curl -I http://localhost-o-URL --user-agent "Hola soy un agente de usuario"
```

Ahora...

```
1 cat cookies.txt
```

Estos **comandos** que acabamos de ver, son los más usados por la **herramienta cURL**, puedes ampliar información visitando su [documentación oficial](#).

MANUAL: Borrar archivos y carpetas en Linux desde la consola

Borrar archivos y carpetas en [Linux](#) desde la [consola / terminal](#).

Una de las formulas más populares de borrar archivos y carpetas de tu [distribución Linux](#), es desde la [consola](#) (también conocida como [terminal](#)).

También es la más segura.



Existen diversas opciones para borrar archivos y carpetas, en este articulo de hoy nos centraremos en las dos más conocidas, funcionales y seguras (según mi opinión).

Veremos:

- Comando “[rm](#)”.
- Herramienta “[Trash-cli](#)”.

Comenzamos...

[~]\$ [rm](#)

SoloLinux

Borrar archivos y carpetas en Linux

Uso del comando “rm”.

El comando “[rm](#)”, es sencillamente el mejor si no quieres dejar rastro de nada.

Sabemos que no hay nada imposible, aun así te digo que es complicado recuperar un archivo que fue removido con este comando. Lo vemos:

Borrar un único archivo o una carpeta que este vacía.	<code>rm archivo-o-carpeta</code>
Borrar todos los archivos y subdirectorios contenidos en una carpeta (comúnmente se conocen como carpetas, pero realmente su denominación es “directorios”).	<code>rm -R carpeta</code>
Ya comentamos anteriormente que el comando “ rm ” es brutal, si quieres que te consulte si estas seguro de borrar (una interrupción), sería de esta forma:	<code>rm -i archivo-o-carpeta</code>

La opción anterior te preguntara en el borrado de cada archivo, si se hace tedioso... puedes indicar que te consulte por lotes de tres archivos.	<code>rm -l archivo-o-carpeta</code>
Borrar solo los archivos (todos) de una carpeta, que tengan una extensión en particular. Por ejemplo todos los “txt”.	<code>rm -R /carpeta/*.txt</code>

Trash-cli

Command Line Trashcan For Linux



Instalación y uso de [Trash-cli](#)

La herramienta **Trash-cli** es muy interesante, pues aporta una papelera ([de reciclaje](#)) desde la linea de [comandos](#). Para que me entiendas... no elimina directamente el archivo, lo envía primero a una papelera.

Normalmente no viene instalada por defecto en casi ninguna [distribución Linux](#), así que la instalamos.

Debian, Ubuntu, Linux Mint y derivados	<code>sudo apt-get install trash-cli</code>
CentOS, Fedora y derivados	<code>sudo yum install trash-cli</code>
Arch Linux y derivados	<code>sudo pacman -S trash-cli</code>
Suse, OpenSuse y derivados	<code>sudo zypper -i trash-cli</code>

Una vez instalado, su uso es tan simple como ejecutar el comando...

trash

Por ejemplo...

`trash archivo-o-carpeta`

OJO!!!!!!!, al usar “**trash**” para eliminar una carpeta, si que se envía a la papelera todo el contenido de la misma.

Listar el contenido de la papelera.

Pues muy fácil.....

trash-list**Restaurar archivos de la papelera.****trash-restore**

Si quieres seleccionar los archivos a restaurar, ejecuta lo siguiente:

restore-trash**Vaciar todo el contenido de la papelera.****trash-empty****Vaciar el contenido de la papelera que llevan más de los días definidos.**

Por ejemplo... eliminamos los archivos contenidos en la papelera que fueron enviados hace más de 5 días.

trash-empty 5

Espero te resulte practico este articulo:

Compártelo.

Instalar nuevos tipos de letra en Linux

Instalar nuevos [tipos de letra](#) en [Linux](#) (más conocidas como fuentes).

Hasta la segunda mitad de la década de los 90, los usuarios de las [distribuciones desktop](#) con [GUI](#) que querían instalar nuevas fuentes, sufrían un autentico quebradero de cabeza.

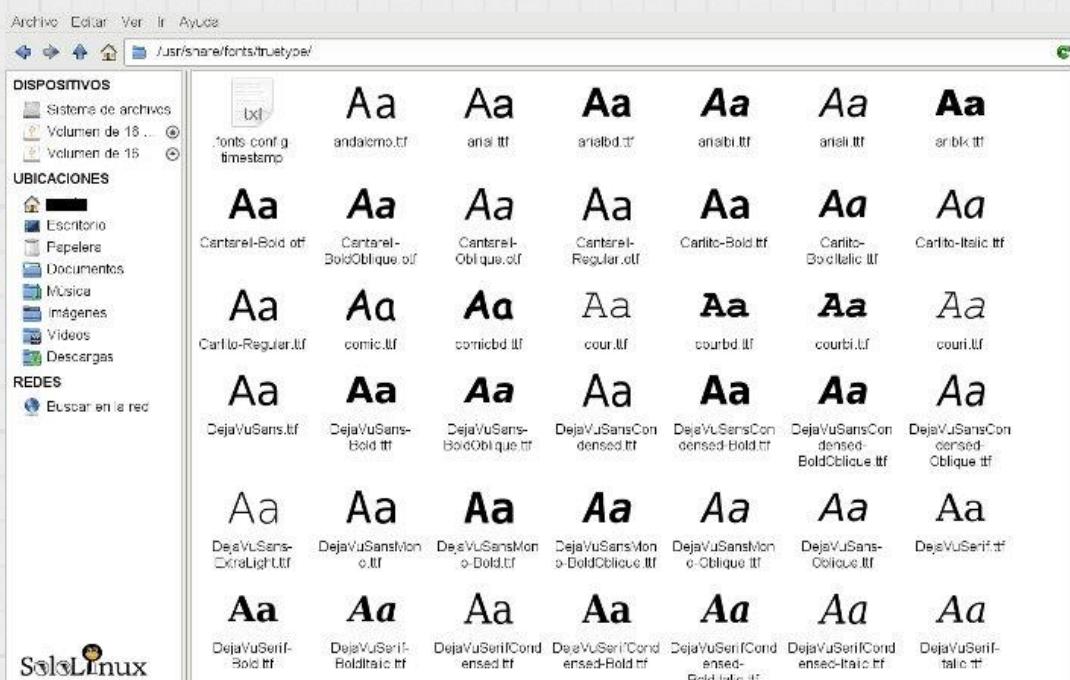
Por fortuna todo cambio en 1996, en ese año se puso en marcha el proyecto “**Core fonts for the Web**”, que liderado por [Microsoft](#) pretendía estandarizar un conjunto de [tipos de letra](#) para su uso en la web.

El proyecto se dio por finalizado en 2002, ademas con gran éxito y aceptación por parte de la mayoría de los desarrolladores y contribuyentes de [sistemas operativos](#) y aplicaciones.

Es evidente que gran parte de la culpa del éxito fue debido a que se permitía su redistribución siempre que se mantuviese el nombre y formato del fichero original y no fueran usadas comercialmente.

Lo importante para Linux, es que a raíz del proyecto “**Core fonts for the Web**” se replanteo la implantación de las mismas en la practica totalidad de [distros Linux](#), todo ello condujo a rediseñar un nuevo sistema desde donde se tomarían los diferentes tipos de letra, presentes y futuras.

Vemos como y donde instalar nuevos tipos de letra (si prefieres las llamas fuentes), en nuestro linux. Estas instrucciones son validas para la gran mayoría de [distribuciones linux](#) que siguen el standard.



Hoy en día es muy sencillo, de hecho puede resultar tan simple como copiar y pegar.

Instalar nuevos tipos de letra en Linux (fuentes)

Instalar nuevas fuentes es algo sencillo, tan solo debes tener en cuenta un concepto, si las quieres para todos los usuarios (incluido el [root](#)), o por el contrario para un usuario específico (es la opción correcta si eres el único usuario).

Lo primero que debemos saber es **donde están los archivos de las fuentes**, la localización es diferente según la formula que elijas.

I. Si es para todos los usuarios.

```
/usr/share/fonts/
```

2. Si es para un usuario exclusivo.

```
/home/tu-usuario/.fonts/
```

I.I – En la primera opción, la de todos los usuarios, y dependiendo de tu distribución normalmente te encontrarás un mínimo de tres carpetas, las cuales contienen tipos de letra diferentes, las vemos.

- /usr/share/fonts/opentype – [Fuentes opentype](#) con extensión “otf”.
- /usr/share/fonts/truetype – [Fuentes TrueType](#) con extensión “ttf”.
- /usr/share/fonts/truetype1 – [Fuentes TrueType](#) con extensión “ttf”.

Ahora... pongamos el ejemplo que descargamos la fuente “arial.ttf”, si estas en la sesión de root (no recomendable) tan solo tienes que copiar el “ttf” y pegarlo en “[/usr/share/fonts/truetype](#)”.

En el caso que estés en la sesión de tu usuario, ejecutas lo siguiente:

```
sudo mv ~/Downloads/arial.ttf /usr/share/fonts/truetype/
```

Si has descargado varias fuentes lo haces así:

```
sudo mv ~/Downloads/*.ttf /usr/share/fonts/truetype/
```

Como punto final recargas las fuentes.

```
sudo fc-cache -fv
```

Ya las tienes disponibles para todos los usuarios del sistema.



2.1 – La segunda opción, la de un usuario en exclusividad es aun más sencilla.

Tan solo debes pegar las fuentes descargadas en la carpeta “**/home/tu-usuario/.fonts/**“. Si quieres hacer la misma operación desde la consola, ejecuta lo que te indico a continuación.

```
mv ~/Downloads/arial.ttf ~/.fonts
```

Si son varias fuentes...

```
mv ~/Downloads/*.ttf ~/.fonts
```

Recargas las fuentes.

```
sudo fc-cache -fv
```

Ya las tienes disponibles para el usuario seleccionado.

Instalar los tipos de letra de Microsoft

Ejecuta dependiendo de tu distribución.

Debian, Ubuntu, Linux Mint y derivados:

```
sudo apt-get install ttf-mscorefonts-installer
```

Redhat, Fedora, Centos, SUSE, y todas las que usen paquetes “[rpm](#)“:

Nos aseguramos de que tenemos instaladas todas las dependencias instaladas (en caso de que seas usuario de **OpenSuse** o **Fedora**, sustituyes **yum** por **zypper** o **dnf**).

```
yum install curl cabextract xorg-x11-font-utils fontconfig
```

Instalamos.

```
rpm -i https://downloads.sourceforge.net/project/mscorefonts2/rpms/msttcore-fonts-installer-2.6-1.noarch.rpm
```

Concluimos recargando las fuentes.

```
sudo fc-cache -fv
```

Ya las tienes disponibles.

Instalar las fuentes de Google

Hace tiempo que **Google** crea y mantiene [una base de datos de tipos de letra tremenda](#), altamente recomendable si eres diseñador.

Vemos dos opciones de instalación, en la primera descargamos e instalamos todas sus fuentes, en la segunda instalaremos una aplicación externa que manejará dichas fuentes.

Primera opción

- Personalmente me quedo con la primera opción aunque se toma más tiempo para instalarlas todas, ademas recuerdo que hace ya un par de años realizamos [un articulo al respecto que aun es funcional](#).

Primero nos aseguramos que tenemos instalada la librería “[fontconfig](#)“.

Seguidamente creamos la carpeta de las fuentes (si es que no la tenemos).

Las descargamos y descomprimimos.

➤ sudo apt-get install fontconfig

➤ mkdir ~/.fonts

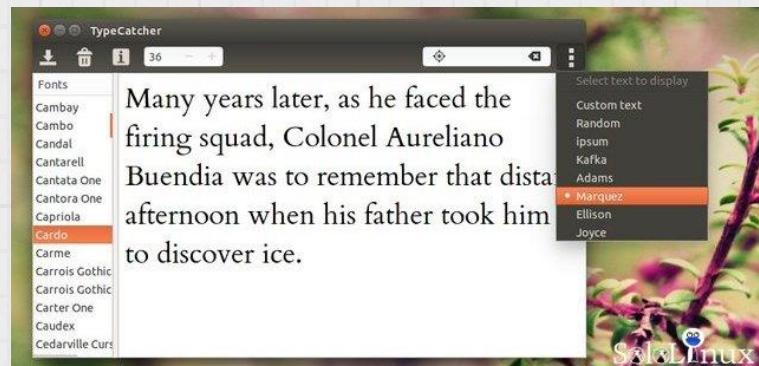
➤ cd ~/.fonts && wget

[&& unzip master.zip](https://github.com/google/fonts/archive/master.zip)

➤ sudo fc-cache -fv

Recargamos los tipos de letra..

Ya las tienes disponibles.



Segunda opción

- En la segunda opción instalaremos la aplicación “[TypeCatcher](#)“, esta herramienta te permite buscar, navegar y descargar las fuentes web de Google, ademas permite previsualizar las fuentes con tamaño y texto. Tiene una pega... solo es valida para Debian, Ubuntu y derivados.

Para instalarlo debes ejecutar lo siguiente:

➤ sudo apt-get install typecatcher

Si quieres disponer siempre de la ultima versión, agrega el PPA.

➤ sudo add-apt-repository
ppa:andrewsomething/typecatcher

sudo apt-get update && sudo apt-get install

Como punto final del artículo, unos enlaces a dos sitios web con miles de fuentes gratuitas.

- [Dafont](#)
- [1001freefonts](#)

Si te gusto el artículo, compártelo.

Configurar las DNS en VestaCP correctamente



Configurar las DNS en..... VestaCP



Configurar las **DNS** en **VestaCP** correctamente con imágenes. A raíz de una duda de un lector de “[sololinux.es](#)”, pude comprobar que apenas existen tutoriales sobre **como configurar tu propio servidor de DNS en VestaCP**. Prácticamente la totalidad de información existente es sobre como configurar las **dns** en tu registrador, de manera que apunte a [VestaCP](#).

Y yo me pregunto para que?...

Como siempre, prefiero tener bajo mi control absolutamente todo, el **servidor de DNS** también. Así que hoy vemos como configurarlo.

Antes de comenzar, un pequeño inciso. Si usas **VestaCP**, tal vez te interese integrar un [administrador de archivos](#) totalmente gratuito, si es afirmativo, [revisa este artículo](#).

Bien comenzamos...

Configurar las DNS en VestaCP

Este tutorial lo basaremos en pequeñas indicaciones, en las cuales incluiremos imágenes de demostración.

Evidentemente, lo primero que haremos es dirigirnos a la pestaña “**WEB**” y agregar el que nos interese.

USUARIO	WEB	DNS	CORREO	BD	TAREAS	RESPALDOS
usuarios: 1 suspendido: 0	dominios: 0 alias: 0 suspendido: 0	dominios: 0 registros: 0 suspendido: 0	dominios: 0 cuentas: 0 suspendido: 0	bases de datos: 1 suspendido: 0	tareas: 8 suspendido: 0	respaldos: 1

Introduces tu dominio y marcas la casilla de “**soportar DNS**”. Haces clic en “**opciones avanzadas**”.

USUARIO	WEB	DNS	CORREO	BD	TAREAS	RESPALDOS
usuarios: 1 suspendido: 0	dominios: 0 alias: 0 suspendido: 0	dominios: 0 registros: 0 suspendido: 0	dominios: 0 cuentas: 0 suspendido: 0	bases de datos: 1 suspendido: 0	tareas: 8 suspendido: 0	respaldos: 1

AÑADIENDO DOMINIO WEB

Domino: linuxmail.es

Dirección IP: 185.185.42.63

Soportar DNS

Soportar Correo

OPCIONES AVANZADAS

En opciones avanzadas te recomiendo que actives el soporte **SSL** y **Let's encrypt**. El directorio local déjalo como está.

WEB

AÑADIENDO DOMINIO WEB

OPCIONES AVANZADAS

Alias
www.linuxmail.es

Soportar Proxy

Soportar SSL

Soportar Lets Encrypt
Tu certificado será emitido automáticamente en 5 minutos.

Directorio local del SSL
public_html

SoloLinux

Deberás esperar cinco minutos para que se generen las claves del certificado, si tienes prisa puedes modificar el tiempo de demora en **tareas cron**.

WEB

EDITANDO DOMINIO → Los cambios han sido guardados exitosamente.

Directorio local del SSL
public_html

Certificado SSL
----BEGIN CERTIFICATE----
MIIGYzCCBUugAwIBAgISA3K1w9FwzoC5ELWj0NuA1uCOM
AQGCSqGSlb3DQEBCwUA
MEoxCzAJBgNVBAYTAiVTMRYwFAYDVQQKEw1MZXQncyB

Clave SSL
----BEGIN RSA PRIVATE KEY----
MIJKAIBAAKCgEAmwsg2JZp7XjnvoO2KbHWl5rMWpkKLH
KSVwo1hZTfsdfOnz
BFUPOIDeXXgAX/APTbKM7lrKbeV0rVc8AnDIU4+yxi8b6SQD

Autoridad de Certificación del SSL / Intermediario (opcional)
----BEGIN CERTIFICATE----
MIIEkjCCA3qgAwIBAgIQCgFBQgAAAVOFc2oLheynCDANBgk
qhkiG9wBAQsFADA/
MSQwlgYDVQQKExtEaWdpdGFsfNpZ25hdHVyZSBUcnVzd

EMITIDO PARA: linuxmail.es
ALIAS: linuxmail.es, www.linuxmail.es
VÁLIDO DESDE: Feb 6 07:34:26 2019 GMT
VÁLIDO HASTA: May 7 07:34:26 2019 GMT
FIRMA: sha256WithRSAEncryption
CLAVE PÚBLICA: 4096 bit
EMITIDO POR: C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3

SoloLinux

Puedes añadir estadísticas del sitio o una cuenta [FTP](#).



Se creo correctamente.

Ahora vamos a la pestaña “DNS”.

Pulsamos en “EDITAR”.

The screenshot shows the SoloLinux VESTA control panel. At the top, there's a navigation bar with links like VESTA, Planes, IP, Gráficos, Estadísticas, Registros, Actualizaciones, Certificados, Servidor, and user admin. Below the navigation bar, there's a summary table with counts for various services. The main area is titled "DNS" and shows a list of 14 DNS records for the domain "linuxmail.es". One record is selected, highlighted with a red background. The selected record details are as follows:

★	185.185.42.63	TTL: 14400	Plantilla: default
	ns1.domain.tld	Serial: 2019020603	Expira: 2020-02-06

Below the list, there are buttons for "LISTAR LOS 14 REGISTROS", "AÑADIR REGISTRO", "EDITAR" (which is highlighted), "SUSPENDER", and "ELIMINAR".

En la sección “DNS”, es imprescindible que en plantilla indiques que quieras usar “**child-ns**”, de manera predeterminada esta en “**default**”.

Guardas.

The screenshot shows the "EDITANDO DOMINIO DNS" form for the domain "linuxmail.es". The form fields are as follows:

- Dominio:** linuxmail.es
- Dirección IP:** 185.185.42.63
- Plantilla NAME:** A dropdown menu shows the following options: default, child-ns (which is highlighted with a blue arrow pointing to it), default, and gmail.
- SOA:** ns1.domain.tld
- TTL:** 14400

At the bottom of the form, there are "Guardar" and "Volver" buttons. To the right of the form, there's a "SoloLinux" logo.

Ahora vamos a la sección “**Planes**”, y en el plan **default** (**child-ns** usa este) le das a editar.

USUARIO	WEB	DNS	CORREO	BD	TAREAS	RESPALDOS
usuarios: 1 suspendido: 0	dominios: 1 alias: 1 suspendido: 0	dominios: 1 registros: 14 suspendido: 0	dominios: 1 cuentas: 0 suspendido: 0	bases de datos: 1 suspendido: 0	tareas: 10 suspendido: 0	respaldos: 1

28 dic 2017 default

Plantilla Web:	default	Dominios DNS:	100	Respaldos:	3
Plantilla Proxy:	default	Registros DNS:	100	Ancho de Banda:	97.66 gb
Plantilla DNS:	default	Dominios de Correo:	100	Disco:	Ilimitado
Acceso SSH:	nologin	Cuentas de Correo:	100	Servidor de Nombre:	ns1.domain.tld ns2.domain.tld
Dominios Web:	100	Base de Datos:	100		
Alias Web:	100	Tareas Programadas:	100		

28 dic 2017 gainsboro

Plantilla Web:	default	Dominios DNS:	10	Respaldos:	1
Plantilla Proxy:	default	Registros DNS:	10	Ancho de Banda:	9.77 gb
Plantilla DNS:	default	Dominios de Correo:	10	Disco:	9.77 gb
Acceso SSH:	nologin	Cuentas de Correo:	10	Servidor de Nombre:	ns1.domain.tld ns2.domain.tld
Dominios Web:	10	Base de Datos:	10		
Alias Web:	10	Tareas Programadas:	10		

SoloLinux

Por defecto verás los ns:

- **ns1.localhost.tld**
- **ns1.localhost.tld**

Los modificas por los de tu dominio y guardas.

Cuota (en megabytes)	<input type="text" value="ilimitado"/>	<input type="text" value="∞"/>
Ancho de Banda (en megabytes)	<input type="text" value="100000"/>	<input type="text" value="∞"/>
Servidor de Nombre	<input type="text" value="ns1.linuxmail.es"/> <input type="text" value="ns2.linuxmail.es"/>	
AÑADIR OTRO SERVIDOR DE NOMBRE		
Guardar		Volver

En la sección “DNS” también puedes poner tu “ns1”.

EDITANDO DOMINIO DNS

DOMINIO
06 feb 2019
10:30:21
ACTIVO
Dirección IP
185.185.42.63

PLANTILLA NAMED
child-ns

FECHA DE EXPIRACIÓN (AAAA-MM-DD)
2020-02-06

SOA
ns1linuxmail.es

TTL
14400

Guardar **Volver**



Al guardar, podrás ver que el cambio es efectivo.

ESTADÍSTICAS
USUARIO: 1
- suspendido: 0
WEB: dominios: 1
- alias: 1
- suspendido: 0
DNS: dominios: 1
- registros: 14
- suspendido: 0
CORREO: dominios: 1
- cuentas: 0
- suspendido: 0
BD: bases de datos: 1
- suspendido: 0
TAREAS: tareas: 10
- suspendido: 0
RESPALDOS: respaldos: 1

LISTAR LOS 14 REGISTROS

dominio	IP	TTL	Plantailla	Expira
linuxmail.es	185.185.42.63	14400	child-ns	2020-02-06
ns1linuxmail.es		14400		



Puedes listar los registros, como veras están todos bajo tu dominio, y manejados desde **VestaCP**.

The screenshot shows the VestaCP control panel with the 'DNS' tab selected. The interface includes a header with links for VESTA, Planes, IP, Gráficos, Estadísticas, Registros, Actualizaciones, Correos, Servidor, and user status (admin). Below the header is a navigation bar with tabs: USUARIO, WEB, DNS (highlighted in orange), CORREO, BD, TAREAS, and RESPALDOS. A search bar at the top right contains filters for 'todos' and 'confirmar selección'.

The main content area displays a list of DNS records for the domain 'linuxmail.es'. Each record is shown in a row with columns for Name, Type, TTL, and Content. The records include:

- _dmarc TXT "v=DMARC1; p=none"
- @ TXT "v=spf1 a mx ip4:185.185.42.63 ~all"
- @ MX 10 mail.linuxmail.es.
- imap A 185.185.42.63
- pop A 185.185.42.63
- smtp A 185.185.42.63
- mail A 185.185.42.63
- ftp A 185.185.42.63
- ftp A 185.185.42.63
- www A 185.185.42.63
- ns2 A 185.185.42.63
- ns1 A 185.185.42.63
- @ A 185.185.42.63
- @ NS ns2.linuxmail.es.
- @ NS ns1.linuxmail.es.

Si te ayudo el articulo sobre las **DNS** en **VestaCP**, compártelo.

Añadir registros DKIM y SPF a un dominio en VestaCP

Añadir registros **DKIM** y **SPF** a un dominio en **VestaCP**. En un anterior artículo ya explicamos como crear **tus propias DNS en VestaCP**, pero si listas los registros de las “DNS”, podrás observar que aunque los solicitaras, no se crearon los **registros DKIM y SPF**.

Hoy veremos como solucionar este problema



Instalar
en
VestaCP



Los registros **DKIM** y **SPF**, son métodos de verificación que se usan en los **clientes de correo** para confirmar que el correo entrante tiene los permisos necesarios para poder ser enviado desde el dominio remitente.

Por lo dicho... estos registros son importantes, dado que los servidores los detectan como correo valido, y de esa forma se disminuye la detección como posible spam y baja la tasa de rechazo del mail. Aleatoriamente en **VestaCP** se produce un error muchas veces imperceptible, y es que no crea tales registros aunque indiques la casilla del uso de **DKIM**.

Vemos como solucionarlo.

Añadir registros DKIM y SPF en VestaCP

Al acceder a la pestaña “Correo” podemos observar que en el dominio seleccionado tenemos activado el soporte **DKIM** (imagen 1), pero si listamos las **dns** del dominio no aparecen por ningún sitio esos registros (imagen 2).

Para solucionar este problema, vamos a editar la **configuración mail** del dominio pulsando en la pestaña “editar” (imagen 1).

The screenshot shows the VestaCP control panel. At the top, there's a navigation bar with links like VESTA, Planes, IP, Gráficos, Estadísticas, Registros, Actualizaciones, Cortafuegos, Servidor, and user admin/Salir. Below the navigation bar, there are several status counters: USUARIO (usuarios: 1, suspendido: 0), WEB (dominios: 1, alias: 1, suspendido: 0), DNS (dominios: 1, registros: 16, suspendido: 0), CORREO (dominios: 1, cuentas: 1, suspendido: 0), BD (bases de datos: 1, suspendido: 0), TAREAS (tareas: 10, pendido: 0), and RESPALDOS (respaldos: 2). In the center, there's a large orange button labeled 'CORREO'. Below the status counters, there's a toolbar with buttons for ABRIR WEBMAIL, todos, confirmar selección, ordenar por: FECHA, LISTAR 1 CUENTAS, AÑADIR CUENTA, EDITAR (highlighted in green), SUSPENDER, and ELIMINAR. The date 06 feb 2019 is shown. The domain linuxmail.es is selected. Below the toolbar, there are configuration options for Disco (0 mb), Soportar AntiVirus: si, Soportar DKIM: si, Soportar AntiSpam: si, and Correo Catch-All: /dev/null. At the bottom, there's another navigation bar with links like VESTA, Planes, IP, Gráficos, Estadísticas, Registros, Actualizaciones, Cortafuegos, Servidor, and user admin/Salir. The 'DNS' tab is highlighted. Below it, the 'LISTANDO LINUXMAILS' section shows two entries: '_dmarc' with type TXT and value '_v=DMARC1; p=none', and '@' with type TXT and value '_v=spf1 a mx ip4:185.185.42.63 -all'.

Al acceder a la edición del dominio de correo, podemos comprobar que efectivamente el “**Soporte DKIM**” esta activado, pero no funciona. Vamos a solucionarlo.

Dominio
06 feb 2019
10:30:22

ACTIVO

- Soportar AntiSpam
- Soportar AntiVirus
- Soportar DKIM

Correo Catch-All

Guardar **Volver**



Desactivamos el “**Soporte DKIM**” y guardamos.

Dominio
06 feb 2019
10:30:22

ACTIVO

- Soportar AntiSpam
- Soportar AntiVirus
- Soportar DKIM

Correo Catch-All

Guardar **Volver**



Sin movernos de la pestaña, lo activamos de nuevo y guardamos.

VESTA Planes IP Gráficos Estadísticas Registros Actualizaciones Contafuegos Servidor admin Salir

USUARIO	WEB	DNS	CORREO	BD	TAREAS	RESPALDOS
usuarios: 1 suspendido: 0	dominios: 1 alias: 1 suspendido: 0	dominios: 1 registros: 16 suspendido: 0	dominios: 1 cuentas: 1 suspendido: 0	bases de datos: 1 suspendido: 0	tareas: 10 suspendido: 0	respaldos: 1

EDITANDO DOMINIO DE CORREO → Los cambios han sido guardados exitosamente.

DOMINIO

06 feb 2019 10:30:22

ACTIVO

- Soportar AntiSpam
- Soportar AntiVirus
- Soportar DKIM

Correo Catch-All

Guarda Volver

Sololinux

Ahora veremos que es lo que a sucedido.....

Nos dirigimos a la sección “DNS” y en el dominio ya nos sorprende la indicación de que existen 16 registros, si recuerdas el artículo anterior teníamos 14.

Listamos los registros a ver, a ver....

VESTA Planes IP Gráficos Estadísticas Registros Actualizaciones Contafuegos Servidor admin Salir

USUARIO	WEB	DNS	CORREO	BD	TAREAS	RESPALDOS
usuarios: 1 suspendido: 0	dominio: 1 alias: 0 suspendido: 0	dominios: 1 registros: 16 suspendido: 0	dominios: 1 cuentas: 1 suspendido: 0	bases de datos: 1 suspendido: 0	tareas: 10 suspendido: 0	respaldos: 2

06 feb 2019

LISTAR LOS 16 REGISTROS todos confirmar selección ordenar por FECHA

linuxmail.es

★	185.185.42.63 ns1.linuxmail.es	TTL: 14400 Serial: 2019020607	Plantilla: child-ns Expira: 2020-02-06
---	-----------------------------------	----------------------------------	---

Sololinux

Sorpresa, ya tenemos los registros correctos en la sección **DNS**.

The screenshot shows the VestaCP control panel with the 'DNS' section selected. At the top, there are summary statistics for various services: USUARIO (1 user), WEB (1 domain, 1 alias, 0 suspended), DNS (1 domain, 16 records, 0 suspended), CORREO (1 domain, 1 account, 0 suspended), BD (1 database, 0 suspended), TAREAS (10 tasks, 0 suspended), and RESPALDOS (1 backup). Below this is a search bar with filters for 'todos' and 'confirmar selección'.

LISTANDO LINUXMAILS

<input type="checkbox"/>	 mail._domain...	TXT	"v=DKIM1; k=rsa; p=MIGIMA0GCSqGSib3DQFBAQUAA4GNADCBiQKBgQDDKd5sKV83wSKhat1CncQN88gqrbl2I
<input type="checkbox"/>	 _domainkey	TXT	"t=y; o=~;"
<input type="checkbox"/>	 _dmarc	TXT	"v=DMARC1; p=none"
<input type="checkbox"/>	 @	TXT	"v=spf1 a mx ip4:185.185.42.63 ~all"
<input type="checkbox"/>	 @	MX 10	mail.linuxmail.es.

SoloLinux

Espero haberte ayudado con esta sencilla explicación, si es así comparte el artículo.

Instalar Visual Studio Code en Ubuntu y derivados

Instalar [Visual Studio Code](#) en [Ubuntu](#), [Linux Mint](#), [Debian](#) y derivados.

[Visual Studio Code](#) es un editor [opensource](#) multiplataforma (su descarga oficial esta bajo software privativo, aun así también es gratis) desarrollado y lanzado por [Microsoft](#) en el año 2015.

Como puntos fuertes destacamos que integra soporte de depuración, soporte de [control de versiones Git](#), resultado de sintaxis, finalización de código, y refactorización de código y fragmentos.

Se dispone de múltiples [extensiones para ampliar las funcionalidades de Visual Studio Code](#).

En este artículo, vemos cómo instalar este destacado editor profesional en [Ubuntu 18.04, Linux Mint 19, Debian 9](#)y todos sus derivados.

Para no complicarnos la vida vamos a **instalar Visual Code Studio** desde el repositorio oficial [VS Code](#) que nos ofrece [Microsoft](#).



Instalar en Ubuntu
Debian y derivados



Instalar Visual Studio Code en Ubuntu y derivados

Comenzamos actualizando el índice de "apt". → `sudo apt update`

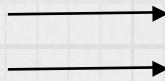
Continuamos con paquetes necesarios. → `sudo apt install -y curl apt-transport-https`

Descargamos e instalamos la clave "[GPG](#)" de [Microsoft](#). → `curl -sSL https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add microsoft.asc`

→ `sudo apt-key add microsoft.asc`

Agregamos el nuevo [repositorio](#). → `echo "deb [arch=amd64] https://packages.microsoft.com/repos/vscode stable main" | sudo tee /etc/apt/sources.list.d/vscode.list`

La instalación es simple...



`sudo apt update`
`sudo apt install -y code`

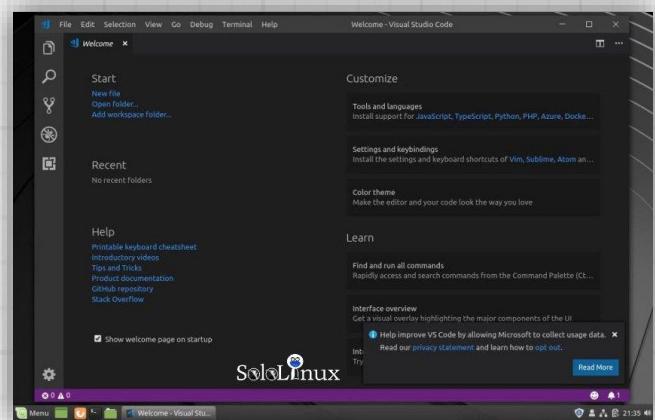
Ya lo tenemos instalado, ahora lo podemos iniciar desde el menú clásico de tu **Linux**, o directamente desde la consola (más rápido). Iniciar desde la consola:



`code`



Ejemplo de pantalla principal del editor.



Si tienes varios editores, y quieres que después de instalar **Visual Studio Code** sea el predeterminado, ejecuta el siguiente comando.

`sudo update-alternatives --set editor /usr/bin/code`

Como norma general este editor se actualiza cada mes, no te olvides de ejecutar de vez en cuando...

`sudo apt update`

`sudo apt upgrade code`

Ya puedes revisar las [extensões](#) que puedes necesitar, así como [el primer manual](#) que te ayudara a configurar **VS-Code** a tu gusto.

Instalar [**Visual Studio Code**](#) en [**Ubuntu**](#), [**Linux Mint**](#), [**Debian**](#) y derivados.

Cyberpanel: El mejor panel de control web gratis

Cyberpanel: El mejor **panel de control web gratis** junto con **VestaCP**.

En “sololinux” no es la primera vez que hablamos de este excelente **panel de control web**. Ya indicamos en anteriores artículos que esta basado en **OpenLiteSpeed**, con todo lo que ello conlleva. Te recomiendo especialmente que revises estos artículos:

- **Comparativa OpenLiteSpeed vs Apache vs Nginx**.
- **Manual OpenLiteSpeed**.

Tal vez os preguntéis el porque trato a este panel de control como “**posiblemente el mejor panel de control web gratis**”, pues... muy fácil, ahora mismo después del lanzamiento de la ultima versión de **OpenLiteSpeed**, no recuerdo ningún panel de control web gratis que ofrezca lo que brinda este. Y es que aparte de todo lo que ya ofrecían las versiones anteriores, en esta superan con creces a muchos paneles de control, incluso de pago.



Destacamos que es compatible con CentOS y Ubuntu 18.x.

Ultimas características añadidas:

- Creación y administración de **contenedores Docker**.
- Instalación y configuración automatizada de **ConfigServer Security & Firewall (CSF)**.
- Limites en los emails por dominio.
- Funciones de **Containerization**, ojo con esto amigos lectores, ojo con esto. Su misión es permitir crear limites a cada dominio, para que me entiendas, como si instalaras un “**CloudLinux**” pero sin su peso y su coste.

La **Containerization** es capaz de limitar:

1. **Uso de CPU.**
2. **Uso de memoria.**
3. **Uso de la red.**
4. **Uso del disco (IO y IOPS)**

Es fabuloso, así que vamos a ver como instalar la nueva versión con imágenes, y también le echaremos un vistazo a las nuevas funciones.

Cyberpanel: El mejor panel de control web gratis

Desde la consola comenzamos la instalación (no instales el [repositorio epel](#)) de **OpenLiteSpeed gratis**.

`yum update`

Sigue los pasos que te indico, pero recuerda sus requisitos mínimos recomendados que son muy básicos.

- **Centos 7.x / Ubuntu 18.x.**
- **Python 2.7.**
- **1024MB de Ram.**
- **10GB Disk Space.**

Comenzamos...

```
sh <(curl https://cyberpanel.net/install.sh || wget -O - https://cyberpanel.net/install.sh)
```

Después de una comprobación rutinaria del sistema, selecciona “1”.

```
Detecting CentOS 7.X...

Pre-flight check completed...

Removed symlink /etc/systemd/system/multi-user.target.wants/httpd.service.

httpd process detected, disabling...

Process check completed...

CyberPanel Installer v2.0

1. Install CyberPanel.
2. Install Addons.
3. Exit.

Please enter the number[1-3]: |
```



Selecciona de nuevo la opción “1”.

```
CyberPanel Installer v2.0

RAM check : 39/512MB (7.62%)

Disk check : 804/9GB (9%) (Minimal 10GB free space)

1. Install CyberPanel with OpenLiteSpeed.
2. Install Cyberpanel with LiteSpeed Enterprise.
3. Exit.

Please enter the number[1-3]: |
```



Te pregunta si quieres modificar la **password** por defecto, cuando modifiques tu usuario admin la puedes cambiar, usa la opción “d”.

```
CyberPanel Installer v2.0
RAM check : 39/512MB (7.62%)
Disk check : 804/9GB (9%) (Minimal 10GB free space)
1. Install CyberPanel with OpenLiteSpeed.
2. Install Cyberpanel with LiteSpeed Enterprise.
3. Exit.

Please enter the number[1-3]: 1

Please choose to use default admin password 1234567, randomly generate one (recommended) or specify the admin password?
Choose [d]efault, [r]andom or [s]et password: [d/r/s] | Sololinux
```

Installing from official server or mirror server?
 Mirror server network is optimized for **Asia Pacific region...**
 If you experience very slow download speed during installation, please try use mirror server on clean system...
 Use mirror server [Y/N]: | Sololinux

Pulsa “Y”.

Te recomiendo “Y”

Replace JS/CSS files to JS Delivr?
 This may improve panel loading speed in **Asia Pacific region...**
 Please select [y/N]: | Sololinux

Te pregunta si quieres instalar la extensión **Memcached** (recomendado), ojo... mas tarde ofrece **Redis**..

Install Memcached extension for PHP?
 Please select [y/N]: | Sololinux

Si instalas **Memcached**, debes responder “Y”.

Install LiteSpeed Memcached?
 Please select [y/N] Sololinux

Install Redis extension for PHP?
 Please select [y/N]: | Sololinux

Te pregunta si quieres instalar la extensión **Redis**, depende de tu decisión anterior.

Siquieres instalar la aplicación **Redis**.

Install Redis? Sololinux
 Please select [y/N]: |

Comienza la instalación de **OpenLiteSpeed “el panel de control web gratis”**.

```
--> Paquete epel-release.noarch 0:7-11 debe ser eliminado
--> Resolución de dependencias finalizada
--> Buscando dependencias sobrantes innecesarias
Found and removing 0 unneeded dependencies

Dependencias resueltas

=====
Package          Arquitectura    Versión      Repositorio   Tamaño
=====
Eliminando:
epel-release     noarch         7-11        @extras       24 k

Resumen de la transacción
=====
Eliminar 1 Paquete

Tamaño instalado: 24 k
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
```

Una vez concluida la instalación, accedes al panel desde el puerto “8089”

<http://tu-ip:8089>

Si no modificaste la pass, es “**1234567**“. Accedes al panel.

CYBERPANEL V 1.8
Web Hosting Control Panel

- LiteSpeed
- Multiple PHPs
- Request throttling.

SoloLinux

El panel es sencillo e intuitivo, pero a la vez super completo. Recuerda [revisar el manual que te indique anteriormente](#) para comenzar a crear tu sitio web.

En la siguiente imagen señalamos donde puedes realizar la **Containerization** (limites del dominio).

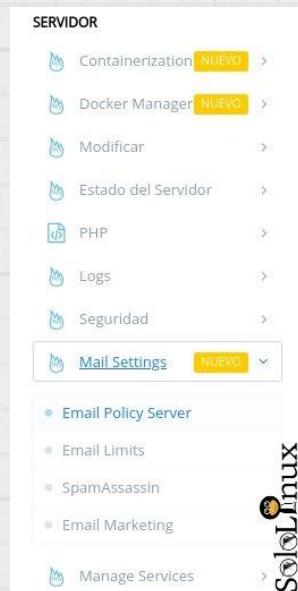
SoloLinux

Aquí donde instalar “**Docker**“.

SoloLinux



Los límites de [email](#), por ejemplo de envío.



Instalar y activar “[CSF Firewall](#)”.
Te recomiendo revisar [este artículo](#) al respecto.

CSF Firewall.

El uso de este panel de control web es muy sencillo, además está en Castellano- Español. Pruébalo, vale la pena, además es muy rápido.

Si crees que este artículo es interesante, compártelo por favor.

Instalar PHP 7.3 en Ubuntu, Debian y derivados



Instalar [PHP 7.3](#) en [Ubuntu](#), [Debian](#) y derivados.

De todos es conocido que pese a sus detractores (que son muchos), [PHP](#) es uno de los lenguajes más populares, es bastante seguro, rápido, simple, eficiente, flexible, y fácil de usar. La nueva [PHP 7.3](#) fue lanzada el seis de diciembre del 2018. Aun así... y pese a que su evolución es constante, aun “**no recomiendo que migres a él**” si tienes un sitio web en producción.

Sobre todo si usas un [CMS](#) como puede ser [WordPress](#).

La integración y compatibilidad con muchos [plugins](#), aun no es efectiva. Si tu sitio web es nuevo, lo puedes probar, si está en producción, insisto en que **no**.

En este artículo vemos como instalar o actualizar ([upgrade](#)) a [PHP 7.3](#) en [Ubuntu](#), [Debian](#) y todos sus derivados.



Instalar PHP 7.3 en Ubuntu, Debian y derivados

Necesitaremos agregar el “[PPA:ondrej/php](#)”, que considero que es el mejor, pues instala las últimas versiones php en todas las versiones activas de Ubuntu, y en las versiones [Stretch](#) y [Jessie](#) (8 y 9) de Debian.

En Debian:

```
sudo apt install apt-transport-https lsb-release
sudo wget -O /etc/apt/trusted.gpg.d/php.gpg
https://packages.sury.org/php/apt.gpg
sudo sh -c 'echo "deb
https://packages.sury.org/php/ $(lsb_release -sc)
main" > /etc/apt/sources.list.d/php.list'
sudo apt update
```

En Ubuntu:

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
Antes de continuar... si lo que haces es una
actualización (upgrade), deberías de ejecutar
el siguiente comando.
dpkg -l | grep php | tee paquetes-php.txt
```

Al ejecutar el anterior comando se generará un archivo llamado “paquetes-php.txt”, en el tendrás un listado de todos los paquetes, módulos, etc..., que tienes instalados actualmente. Por si acaso te olvidas después de alguno.

Comenzamos la instalación de PHP 7.3

La instalación es común para todas las [distribuciones](#) basadas en Debian y Ubuntu.

```
sudo apt install php7.3 php7.3-common php7.3-cli
```

Puedes comprobar la versión instalada.

```
php -v
```

Bien... ya lo tenemos instalado. Ahora vamos con los módulos.



Anteriormente hablamos de crear un archivo de texto con los módulos que tenías instalados, si es tu caso tan solo tienes que sustituir, por ejemplo 7.1, por 7.3.

Un ejemplo de módulos que deberías instalar:

```
apt install php-pear php7.3-curl php7.3-dev php7.3-gd php7.3-mbstring php7.3-zip php7.3-mysql  
php7.3-xml php7.3-fpm libapache2-mod-php7.3 php7.3-imagick php7.3-recode php7.3-tidy php7.3-  
xmlrpc php7.3-intl
```

También puedes buscar e instalar módulos específicos, con:

```
apt-cache search php7.3
```

Oh bien listar todos los módulos disponibles para su instalación.

```
apt-cache search --names-only ^php
```

Si quieres que PHP 7.3 sea la versión predeterminada, escribe:

```
update-alternatives --set php /usr/bin/php7.3
```

Desactivamos y borramos la versión anterior (si es que estaba presente), en el ejemplo **PHP 7.1**.

```
a2dismod php7.1  
apt purge php7.1 php7.1-common
```

Finalmente activamos la **7.3**.

```
a2enmod php7.3
```

Como ultimo punto y no por ello menos importante, te recuerdo que las [PPA de Ondrej](#) no cuentan con actualizaciones automáticas, cada cierto tiempo deberás ejecutarla de manera manual.

```
apt update
```

```
apt upgrade
```

Es pero que te ayude este artículo, si es así compártelo.

Instalar Arch linux paso a paso

Instalar [Arch linux](#) **paso a paso**, manual completo.

Arch Linux es una [distribución Linux](#) muy popular entre los usuarios de nivel medio y avanzado. Destacamos que su desarrollo corresponde exclusivamente a [la comunidad](#).

La [wiki de archilinux](#) es buenísima, obtendrás respuestas para todo.

Como norma general **Arch Linux** sigue el [principio KISS](#), lo que quiere decir que solo se ofrece una versión base mínima, de forma que lo instalas y configures a tu gusto o necesidad.

Una vez instalado puedes integrar un escritorio y tus programas preferidos, ademas cuenta con el beneficio de que los nuevos paquetes o actualizaciones son de lanzamiento continuo (se lanzan novedades 24h al día). El administrador de paquetes que usa **Arch Linux** es [Pacman](#), que es muy, muy... bueno.

Parece difícil, pero te aseguro que no lo es tanto.

En este artículo veremos como **instalar Arch linux** paso a paso y con imágenes, pero antes no puedo olvidarme de comentar sus requisitos (no necesitas una gran maquina).

Requisitos mínimos:

- Un sistema que soporte **x86_64** (64 bit), desde 2017 ya no se ofrece el vetusto 32 bits.
- 512 MB de RAM.
- Un mínimo de 1 GB de [HDD](#).
- Conexión a internet.
- Unidad flash USB o CD para grabar la imagen ISO.



Comenzamos la instalación.

Instalar Arch linux paso a paso

Descargas la ultima versión de Archi linux, y grabas la [iso en un USB](#) o **CD-ROM**. La zona oficial de descargas es:

- [Zona oficial de descargas](#).

Arrancas la maquina con el USB o CD, veras la primera pantalla de instalación.

Elije **Boot Arch Linux (x86_64)** y pulsa la [tecla Enter](#).



Automáticamente comprobara si tu sistema cumple los requisitos mínimos. Al terminar aparecerá la consola raíz.

```
archiso login: root (automatic login)
root@archiso ~ #
```



Ahora debemos de configurar la red (es necesario Internet), haz lo posible para que tu red (router) disponga del [servidor DHCP](#) activado, de esa forma se configura todo (IP-DNS) automáticamente. Comprobamos la conexión comprobando que hacemos [ping](#), por ejemplo a "[Google.com](#)".

```
archiso login: root (automatic login)
root@archiso ~ # ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.8 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fd50:1d9:9fe3:1400:aad0:1cb9:aefc:fe6b prefixlen 64 scopeid 0x0<global>
        inet6 fe80::7fa2:9483:7172:d905 prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:39:1e:03 txqueuelen 1000 (Ethernet)
            RX packets 25 bytes 2102 (2.0 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 18 bytes 2064 (2.0 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 0 bytes 0 (0.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 0 bytes 0 (0.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@archiso ~ # ping google.com -c 2
PING google.com (172.217.163.142) 56(84) bytes of data.
64 bytes from maa05s04-in-f14.1e100.net (172.217.163.142): icmp_seq=1 ttl=56 time=10.9 ms
64 bytes from maa05s04-in-f14.1e100.net (172.217.163.142): icmp_seq=2 ttl=56 time=10.5 ms

--- google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 10.576/10.764/10.953/0.215 ms
root@archiso ~ #
```



Llegamos a uno de los momentos más temidos por los usuarios noveles...

Vamos a crear las particiones del disco.

Comprobamos los discos instalados.

```
root@archiso ~ # fdisk -l
Disk /dev/sda: 40 GiB, 42949672960 bytes, 83886000 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

fdisk -l

Disk /dev/loop0: 446.8 MiB, 468516864 bytes, 915072 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
root@archiso ~ #
```



En el ejemplo usaremos al completo un disco de 40 GB (/dev/sda), y crearemos tres particiones:

- La partición boot: Será de 500Mg
- La SWAP: 4GB
- Resto del disco: 35GB

Para particionar usaremos la consola de **fdisk**, tal vez te resuelva alguna duda [este articulo](#).

Comenzamos con la partición boot, y continuaremos con las siguientes. Observa detenidamente las imágenes, pues en ellas están las explicaciones.

fdisk /dev/sda ##coloca tu ruta valida

Con la letra “**n**” indicamos que queremos una nueva partición, y con la “**p**” que será primaria.

```
root@archiso ~ # fdisk /dev/sda
Welcome to fdisk (util-linux 2.32).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x1520463b.

Command (m for help): p ← LISTAR PARTICIONES
Disk /dev/sda: 40 GiB, 42949672960 bytes, 83886080 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x1520463b

Command (m for help): n ← CREAR NUEVA PARTICION
Partition type
  p  primary (0 primary, 0 extended, 4 free)
  e  extended (container for logical partitions)
Select (default p): p ← P DE PRIMARIA
Partition number (1-4, default 1): 1 ← SELECCIONA EL NUMERO, NORMALMENTE 1
First sector (2048-83886079, default 2048): ← DEJA LOS SECTORES POR DEFECTO
Last sector, +sectors or +size{K,M,G,T,P} (2048-83886079, default 83886079): +500M ← 500 /boot

Created a new partition 1 of type 'Linux' and of size 500 MiB.

Command (m for help):
```

Sololinux

Recuerda que seguimos estando en la **consola de fdisk**.

Creamos la **partición SWAP**.

```
Command (m for help): n ← NUEVA PARTICION
Partition type
  p  primary (1 primary, 0 extended, 3 free)
  e  extended (container for logical partitions) ← PUESTA A PRIMARIA
Select (default p): p ← SELECCIONA 2
Partition number (2-4, default 2): 2 ← SELECCIONA 2
First sector (1026048-83886079, default 1026048): ← DEJALO POR DEFECTO
Last sector, +sectors or +size{K,M,G,T,P} (1026048-83886079, default 83886079): +4G ← TAMAÑO DE LA SWAP

Created a new partition 2 of type 'Linux' and of size 4 GiB.

Command (m for help): t ← CAMBIAMOS EL TIPO DE PARTICION
Partition number (1,2, default 2): 2 ← AL IGUAL QUE ANTES SELECCIONAMOS 2
Hex code (type L to list all codes): 82 ← 82 ES POR DEFECTO "SWAP"

Changed type of partition 'Linux' to Linux swap

Command (m for help):
```

Sololinux

Creamos la **partición raíz**.

```
Command (m for help): n ← NUEVA PARTICION
Partition type
  p  primary (2 primary, 0 extended, 2 free)
  e  extended (container for logical partitions) ← PUESTA A PRIMARIA
Select (default p): p ← PARTICION PRIMARIA
Partition number (3,4, default 3): 3 ← AHORA CREAMOS LA 3
First sector (9414656-83886079, default 9414656): ← DEJALO POR DEFECTO
Last sector, +sectors or +size{K,M,G,T,P} (9414656-83886079, default 83886079): ← PULSA ENTER PARA LA PARTICION "ROOT"

Created a new partition 3 of type 'Linux' and of size 35.5 GiB.

Command (m for help):
```

Sololinux

Nos aseguramos que las particiones se crearon correctamente con “p”, y guardamos los cambios con “w”.

```
Command (m for help): p LISTAMOS LAS PARTICIONES CON "P"
Disk /dev/sda: 40 GiB, 42949672960 bytes, 83886080 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x1520463b

Device     Boot   Start   End   Sectors  Size Type
/dev/sda1        2048 1026047 1024000 500M 83 Linux
/dev/sda2    1026048 9414655 8388608 4G 82 Linux swap / Solaris
/dev/sda3    9414656 83886079 74471424 35.5G 83 Linux

Command (m for help): w GUARDAMOS LOS CAMBIOS CON "w"
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

root@archiso ~ #
```

Al ejecutar: **fdisk -l**

Comprobamos que se han creado y guardado correctamente.

/dev/sda1 – /boot
/dev/sda2 – SWAP
/dev/sda3 – / (root)

```
root@archiso ~ # fdisk -l
Disk /dev/sda: 40 GiB, 42949672960 bytes, 83886080 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x1520463b

Device     Boot   Start   End   Sectors  Size Type
/dev/sda1        2048 1026047 1024000 500M 83 Linux
/dev/sda2    1026048 9414655 8388608 4G 82 Linux swap / Solaris
/dev/sda3    9414656 83886079 74471424 35.5G 83 Linux

Disk /dev/loop0: 446.8 MiB, 468516864 bytes, 915072 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
root@archiso ~ #
```

En este punto formatearemos las particiones con sus respectivos [sistemas de archivos](#).

Tipos de sistemas de archivos que usaremos:

- Boot: como EXT2 o EXT3.
- Swap: como Swap.
- Root: actualmente existen otras alternativas pero EXT4 funciona muy bien.

Ejecutamos:

Boot: mkfs.ext2 /dev/sda1
Swap: mkswp /dev/sda2
Raíz / Root: mkfs.ext4 /dev/sda3

```
root@archiso ~ # mkfs.ext2 /dev/sda1
mke2fs 1.44.1 (24-Mar-2018)
Creating filesystem with 512000 1k blocks and 128016 inodes
Filesystem UUID: 132c04c1-c038-4f2f-9ef2-7cdb32ee94db
Superblock backups stored on blocks:
8193, 24577, 40961, 57345, 73729, 204801, 221185, 401409

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done

root@archiso ~ # mkfs.ext4 /dev/sda3
mke2fs 1.44.1 (24-Mar-2018)
Creating filesystem with 9300928 4k blocks and 2330160 inodes
Filesystem UUID: d70c72f6-97a7-4be6-bac4-d256b1c3fb15
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
4096000, 7962624

Allocating group tables: done
Writing inode tables: done
Creating journal (65536 blocks): done
Writing superblocks and filesystem accounting information: done

root@archiso ~ # mkswp /dev/sda2
Setting up swapspace version 1, size = 4 GiB (4294963200 bytes)
no label, UUID=6f8b75a9-b733-4b2a-82d7-6fba3a5c39fa
root@archiso ~ #
```

Montamos las particiones.

```
mount /dev/sda3 /mnt
swapon /dev/sda2
```

Por fin llegamos a la instalación propiamente dicho, vamos a ello.
Ejecuta.

```
pacstrap /mnt/ base base-devel
```

Comienza la instalación.

```
root@archiso ~ # pacstrap /mnt base base-devel
==> Creating install root at /mnt
==> Installing packages to /mnt
:: Synchronizing package databases...
core                               130.4 KiB  105K/s 00:01 [#####
extra                             1598.8 KiB  651K/s 00:02 [#####]
```



Ten en cuenta que la instalación necesita descargar paquetes desde Internet, la demora dependerá de tu conexión.

```
libtool-2.4.6+40+g6ca5e224-5  licenses-20171006-1  linux-4.15.15-1
logrotate-3.14.0-1  lvm2-2.02.177-5  m4-1.4.18-1  make-4.2.1-2  man-db-2.8.3-1
man-pages-4.15-1  mdadm-4.0-1  nano-2.9.5-1  netctl-1.16-1  pacman-5.0.2-2
patch-2.7.6-1  pciutils-3.5.6-1  pcmciautils-018-7  perl-5.26.1-2
pkg-config-0.29.2-1  procps-ng-3.3.13-1  psmisc-23.1-1  reiserfsprogs-3.6.27-1
s-nail-14.9.10-1  sed-4.5-1  shadow-4.5-4  sudo-1.8.22-1  sysfsutils-2.1.0-9
systemd-238.76-1  systemd-syscompat-238.76-1  tar-1.30-1  texinfo-6.5-1
usbutils-009-1  util-linux-2.32-3  vi-1:070224-2  which-2.21-2  xfsprogs-4.15.1-1

Total Download Size: 286.02 MiB
Total Installed Size: 1033.92 MiB

:: Proceed with installation? [Y/n]
:: Retrieving packages...
linux-api-headers-4.14.8-1-any    915.1 KiB  712K/s 00:01 [#####
tzdata-2018d-1-any                237.1 KiB  57.8K/s 00:04 [#####
iana-etc-20180221-1-any          364.0 KiB  444K/s 00:01 [#####
filesystem-2018.1-2-x86_64       7.8 KiB   0.00B/s 00:00 [#####
glibc-2.26-11-x86_64              3.1 MiB   496K/s 00:11 [#####]
```



Una vez concluida la instalación, generamos el “[fstab](#)“.

```
genfstab /mnt >> /mnt/etc/fstab
```

Verificamos que es valido.

```
cat /mnt/etc/fstab
root@archiso ~ # genfstab /mnt >> /mnt/etc/fstab
root@archiso ~ #
root@archiso ~ # cat /mnt/etc/fstab
# Static information about the filesystems.
# See fstab(5) for details.

# <file system> <dir> <type> <options> <dump> <pass>
# UUID=d70c72f6-97a7-4be6-bac4-d256b1c3fb15
/dev/sda3      /           ext4      rw,relatime,data=ordered     0 1
# UUID=6f8b75a9-b733-4b2a-82d7-6fba3a5c39fa
/dev/sda2      none        swap      defaults,pri=-2 0 0
root@archiso ~ # _
```



Chroot modifica el directorio raíz del proceso que se ejecuta actualmente y sus hijos.
Ejecuta...

arch-chroot /mnt

Seleccionamos el lenguaje del sistema.

nano /etc/locale.gen

Descomenta el lenguaje de tu elección, guarda y cierra el editor. Tenemos que generar el archivo “locale”.

locale-gen

Establecemos la configuración del sistema con el idioma seleccionado anteriormente.

echo "LANG=es_ES.UTF-8" > /etc/locale.conf

Creamos la zona horaria:

In -sf /usr/share/zoneinfo/ZONA/PAIS /etc/localtime

Si tienes dudas puedes listar las disponibles.

ls /usr/share/zoneinfo

```
[root@archiso ~]# ls /usr/share/zoneinfo/
Africa      Canada    EST      GMT0      Iran       Mexico     Portugal   Singapore zone1970.tab
America     CET        EST5EDT  GMT-0     iso3166.tab MST       posix      Turkey    zone.tab
Antarctica  Chile      Etc      GMT+0     Israel     MST7MDT   posixrules UCT       Zulu
Arctic      CST6CDT   Europe   Greenwich Jamaica   Navajo    PRC       Universal
Asia        Cuba       Factory  Hongkong Japan      NZ        PST8PDT   US
Atlantic    EET        GB       HST       Kwa.jalein NZ-CHAT   right    UTC
Australia   Egypt      GB-Eire Iceland   Libya     Pacific   ROC       WET
Brazil      Eire       GMT      Indian    MET       Poland   ROK       W-SU
[root@archiso ~]#
```



También puedes **configurar el hardware con UTC**.

Establecemos el hostname.

Establecemos la password del root.

Instalamos el GRUP (cargador de arranque).

```
[root@archiso ~]# grub-install /dev/sda
Installing for i386-pc platform.
Installation finished. No error reported.
[root@archiso ~]# grub-mkconfig -o /boot/grub/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-linux
Found initrd image: /boot/initramfs-linux.img
Found fallback initrd image(s) in /boot: initramfs-linux-fallback.img
done
[root@archiso ~]#
```



```
[root@archiso ~]# exit
exit
arch-chroot /mnt 3.79s user 2.07s system 1% cpu 9:13.67 total
root@archiso ~ # reboot
```



hwclock --systohc --utc

echo "sistema.archlinux" > /etc/hostname

Passwd

pacman -S grub

grub-install /dev/sda

grub-mkconfig -o /boot/grub/grub.cfg

Ya solo nos falta salir del sistema y reiniciar la maquina.

exit

reboot

Una vez reinicie el sistema recuerda que debes iniciar la sesión como root, y con la password que indicamos anteriormente.

Ya lo tienes instalado, te recomiendo que revises estos artículos:

- [Instalar un escritorio en Arch linux.](#)
- [Que hacer después de instalar Arch linux.](#)

Damos por concluido el articulo de hoy, si lo consideras útil... “**compártelo**“.

Instalar Opera en Fedora, OpenSuse, CentOS y derivados

Instalar [Opera](#) en [Fedora](#), [OpenSuse](#), [CentOS](#) y derivados.

No vamos a descubrir ahora las bondades del navegador web “Opera”, son de todos conocidas.

Últimamente sorprende en cada versión que lanza, se superan a si mismos. Es rápido (similar a [Vivaldi](#)) si lo comparamos en navegadores de su categoría, ademas ofrece una alta seguridad entre otras excelentes características.

Opera ofrece la descarga del navegador desde su propio sitio web, lamentablemente solo la [versión en “deb”](#) para instalar en Debian, Ubuntu y derivados.

Instalar Opera en Fedora

Importamos la key.

Creamos el repositorio.

Agregamos lo siguiente:

```
rpm --import https://rpm.opera.com/rpmrepo.key
nano /etc/yum.repos.d/opera.repo
[opera]
name=Opera packages
type=rpm-md
baseurl=https://rpm.opera.com/rpm
gpgcheck=1
gpgkey=https://rpm.opera.com/rpmrepo.key
enabled=1
RPMPREPO
```

Guarda el archivo y cierra el editor.

Lo instalamos...

```
dnf -y install opera-stable
```

Instalar Opera en OpenSuse

Importamos la key.

Creamos el repositorio.

Agregamos lo siguiente:

```
rpm --import https://rpm.opera.com/rpmrepo.key
nano /etc/zypp/repos.d/opera.repo
[opera]
name=Opera packages
type=rpm-md
baseurl=https://rpm.opera.com/rpm
gpgcheck=1
gpgkey=https://rpm.opera.com/rpmrepo.key
enabled=1
autorefresh=1
keeppackages=0
RPMPREPO
```

Guarda el archivo y cierra el editor.

Lo instalamos...

```
<span class="code_sectcolor7">zyppler install opera-stable</span>
```

Instalar



Las versiones “rpm” solo están en sus repositorios.

Instalar Opera en CentOS y derivados

Importamos la key.

Creamos el repositorio.

Agregamos lo siguiente:

```
rpm --import https://rpm.opera.com/rpmrepo.key
nano /etc/yum.repos.d/opera.repo
[opera]
name=Opera packages
type=rpm-md
baseurl=https://rpm.opera.com/rpm
gpgcheck=1
gpgkey=https://rpm.opera.com/rpmrepo.key
enabled=1
RPMREPO
```

Guarda el archivo y cierra el editor.

Lo instalamos...

```
yum -y install opera-stable
```

Una vez instalado el navegador web puedes acceder desde la consola o desde tu escritorio preferido.

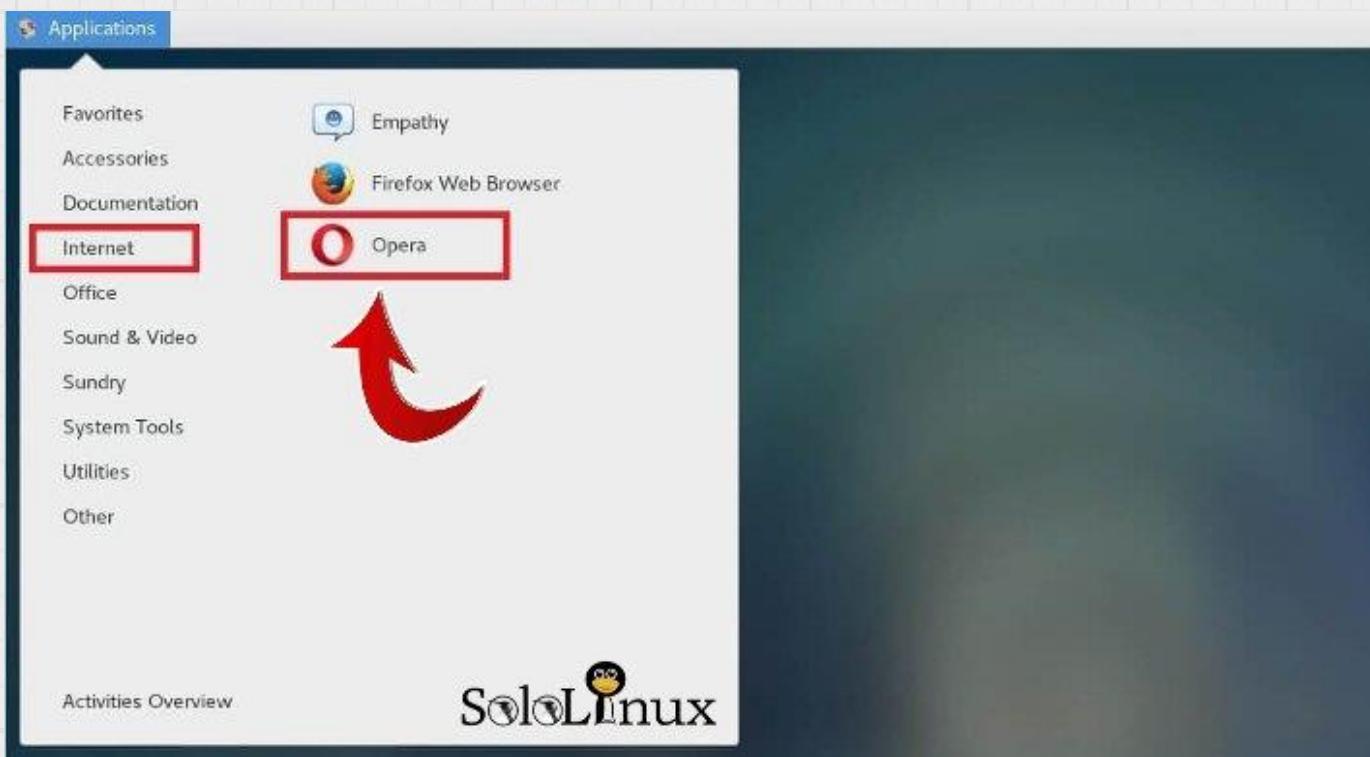
En consola.

```
opera
```

Si prefieres usar un escritorio añade el icono de inicio en tu panel.

Ejecuta lo siguiente independientemente del Linux que uses:

```
sed -i 's/Icon=/Icon=/g' /usr/share/applications/opera.desktop
```



Espero te sea útil este mini-tuto.

Instalación y uso de COCKPIT en tu Servidor

Instalación y uso de **COCKPIT** en tu **Servidor** o **VPS**.

¿Quieres visualizar tu servidor en un navegador web?

¿Crees que es posible realizar tareas de sistema desde una interfaz gráfica y con el ratón (mouse)?.

La respuesta es “SI”, es posible con Cockpit

Cockpit es una herramienta de administración de servidores y vps basada en la web, ademas es opensource.

Sigue en desarrollo y cada vez se acoplan nuevas funciones, pero ya es capaz de administrar el sistema, realizar tareas con su consola integrada, controlar y manejar los dispositivos de almacenamiento, la configuración de la red, visualizar registros, administrar contenedores, gráficas en tiempo real de uso del servidor, etc....

Y todo con el ratón (mouse).

Interesante proyecto, verdad?, estas son sus principales características.

1. Administrar máquinas desde una sola sesión de **Cockpit**.
2. Crea y gestiona contenedores Docker.
3. Crea y gestiona maquinas virtuales.
4. Modificar la configuración de red.
5. Administrar las cuentas de usuario.
6. Shell basado en web con consola / terminal.
7. Gráficas en tiempo real de uso y rendimiento del sistema.
8. Recopila la configuración del sistema y diagnósticos.

Como puedes ver es un aplicación realmente interesantes, así que vamos a ver como instalar y hacer **uso de COCKPIT** en tu Servidor (en este ejemplo usaremos CentOS 7).

Instalar COCKPIT

Su instalación es simple, y su único requisito es que tengas el repositorio **epel** instalado, si no recuerdas como hacerlo revisa este anterior articulo.

Comenzamos...

`yum install cockpit`

`systemctl enable --now cockpit.socket`

`sudo firewall-cmd --permanent --zone=public --add-service=cockpit`

`sudo firewall-cmd --reload`

Ya lo tenemos instalado, sencillo eh.



Uso de COCKPIT

Una vez instalado, abrimos con nuestro [navegador web](#) favorito y accedemos con la ip y el **puerto 9090**.

<https://ip-del-servidor:9090>

Veras el mensaje “**la conexión no es privada**“.

Accede de todas formas.

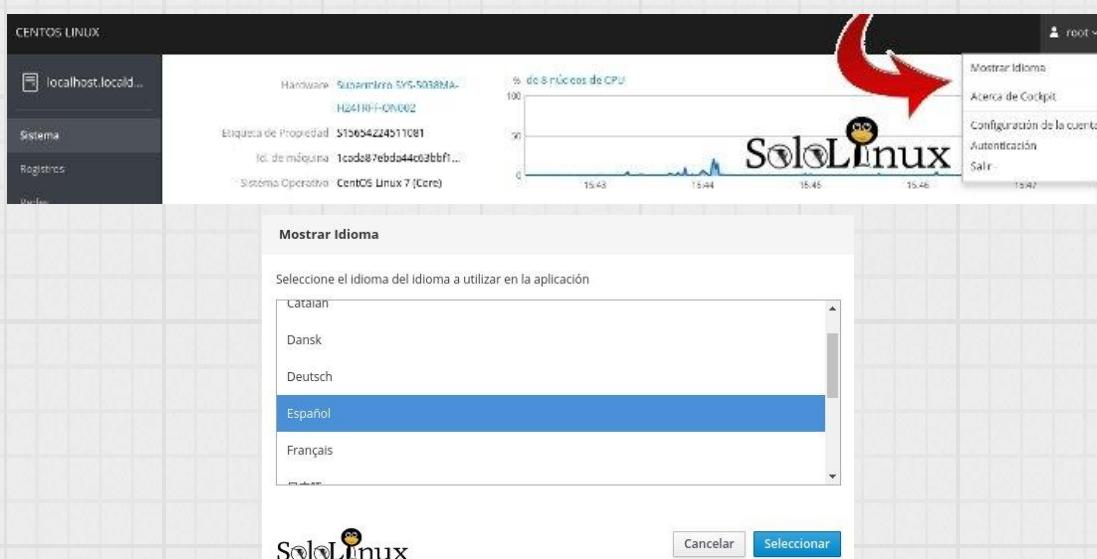


Sololinux

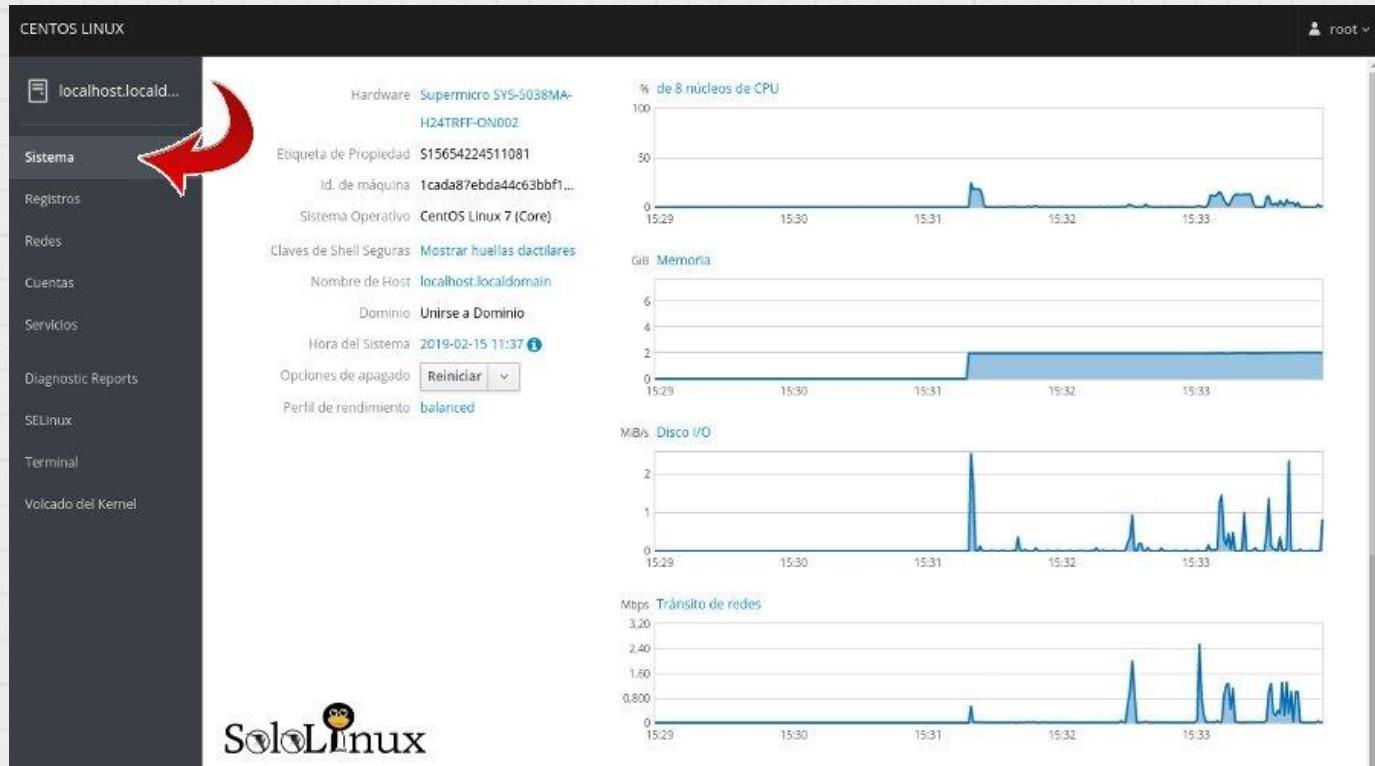
Introduces los datos de [acceso de usuario con privilegios o el root](#) (como al entrar en consola).



Una vez dentro, ves a la esquina superior derecha y modifica el lenguaje por defecto.



Bien, ya lo tenemos en **Español**, aceptas y vamos a la columna de la izquierda, en sistema podrás ver la info del sistema y gráficas de uso en tiempo real.



En la siguiente pestaña vemos los registros del sistema.

CENTOS LINUX

Registros

February 15, 2019

Fecha	Mensaje	Fuente	Número de entradas
11:16	Message aborted.	/usr/lib64/plesk...	38
11:12	error: Received disconnect from 58.87.180.72 port 53699:3: com.jcraft.jsch.JSchException: Auth fail [p...]	sshd	
11:12	error: Received disconnect from 58.87.180.72 port 53664:3: com.jcraft.jsch.JSchException: Auth fail [p...]	sshd	
10:26	Message aborted.	/usr/lib64/plesk...	18
10:20	PAM service(sshd) ignoring max retries; 6 > 3	sshd	
10:20	error: maximum authentication attempts exceeded for root from 95.26.94.89 port 27517 ssh2 [preauth]	sshd	
05:36	Message aborted.	/usr/lib64/plesk...	74
05:30	can not send HUP signal to Dr.Web daemon (pid=7349, error=No such process)	update.pl	
02:45	Message aborted.	/usr/lib64/plesk...	48
02:35	error: Received disconnect from 132.232.97.57 port 52126:3: com.jcraft.jsch.JSchException: Auth fail [...]	sshd	
02:35	error: Received disconnect from 132.232.97.57 port 51950:3: com.jcraft.jsch.JSchException: Auth fail [...]	sshd	
00:05	Message aborted.	/usr/lib64/plesk...	42

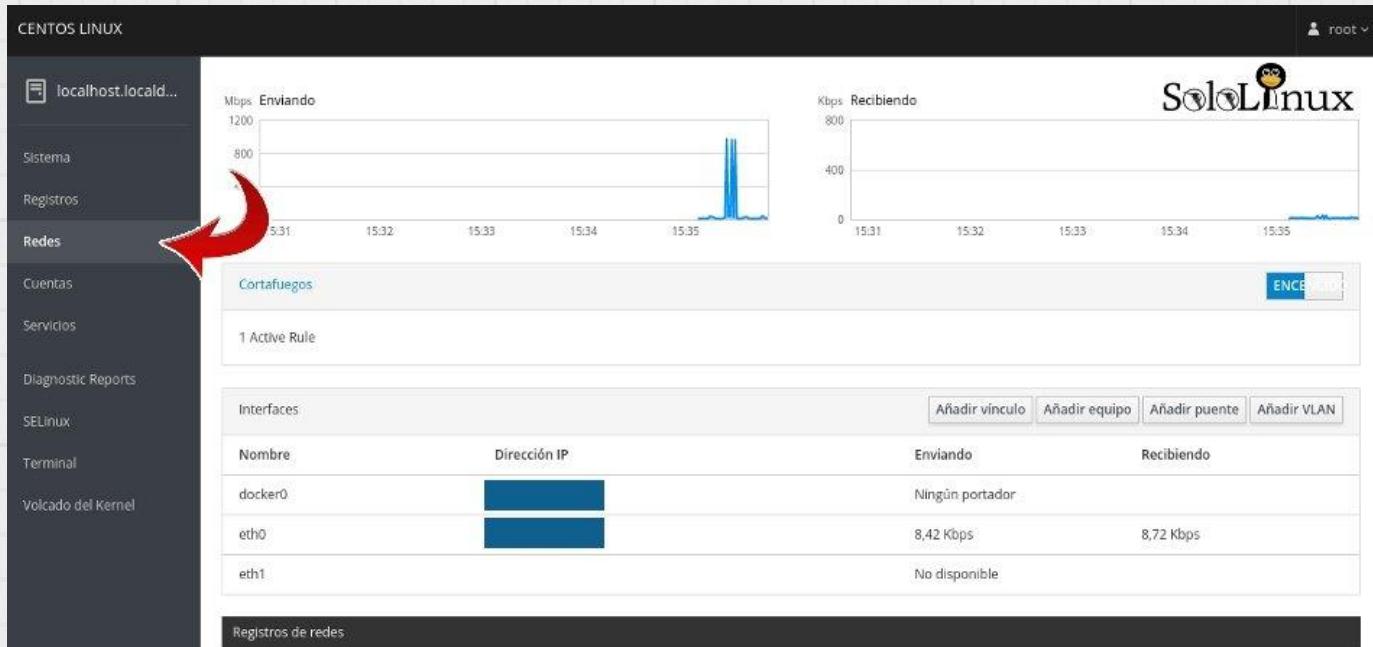
February 14, 2019

Fecha	Mensaje	Fuente	Número de entradas
19:44	Message aborted.	/usr/lib64/plesk...	78
19:39	PAM service(sshd) ignoring max retries; 6 > 3	sshd	
19:39	error: maximum authentication attempts exceeded for root from 5.36.23.60 port 53709 ssh2 [preauth]	sshd	
14:24	Message aborted.	/usr/lib64/plesk...	86
14:17	error: Received disconnect from 103.207.36.187 port 56337:3: com.jcraft.jsch.JSchException: Auth fail [...]	sshd	
12:14	Message aborted.	/usr/lib64/plesk...	36
12:09	Unable to connect to Plesk Database: Can't connect to local MySQL server through socket '/var/lib/mysql...	sshd	

SoloLinux



En redes aparecerán las gráficas de uso totales, entrantes o salientes, por dispositivo, y también sus registros.



En la pestaña cuentas.... sobran las palabras.

Tan solo se agradece que permite crear nuevos usuarios o modificar los existentes, incluyendo sus roles o permisos.

The screenshot shows the SoloLinux interface with the 'Cuentas' tab selected in the sidebar. On the right, there is a 'Crear nueva Cuenta' button and a list of users: 'root' and another user represented by a blue icon. A red arrow points to the 'Cuentas' tab in the sidebar.

En la siguiente pantalla vemos todos los servicios instalados en el servidor (por ejemplo [apache](#)), activos e inactivos.

The screenshot shows the SoloLinux interface with the 'Servicios' tab selected in the sidebar. On the right, there is a table listing various system services. A red arrow points to the 'Servicios' tab in the sidebar.

Estado	Descripción	Id
activo (exited)	Install ABRT coredump hook	abrt-ccpp.service
activo (running)	ABRT kernel log watcher	abrt-oops.service
inactive (dead)	Test vmcores for ABRT	abrt-vmcore.service
inactive (dead)	ABRT Xorg log watcher	abrt-xorg.service
activo (running)	ABRT Automated Bug Reporting Tool	abrtd.service
activo (running)	Job spooling tools	atd.service

Si planeas hacer **uso de COCKPIT** de manera continua, tal vez te interese activar el guardado de registro de reportes.

CENTOS LINUX

- localhost.localdomain...
- Sistema
- Registros
- Redes
- Cuentas
- Servicios
- Diagnostic Reports**
- SELinux
- Terminal
- Volvido del Kernel

Esta herramienta recogerá la configuración del sistema e información de diagnóstico desde este sistema para su uso en la diagnosis de problemas con el sistema.

La información recolectada será almacenada localmente en el sistema.

Crear Reporte

SoloLinux

A no ser que tengas activado el **SELinux**, en su pestaña no aparece nada. Me lo salto, nunca lo uso. En la siguiente tenemos la consola terminal, debo decir que funciona bastante bien.

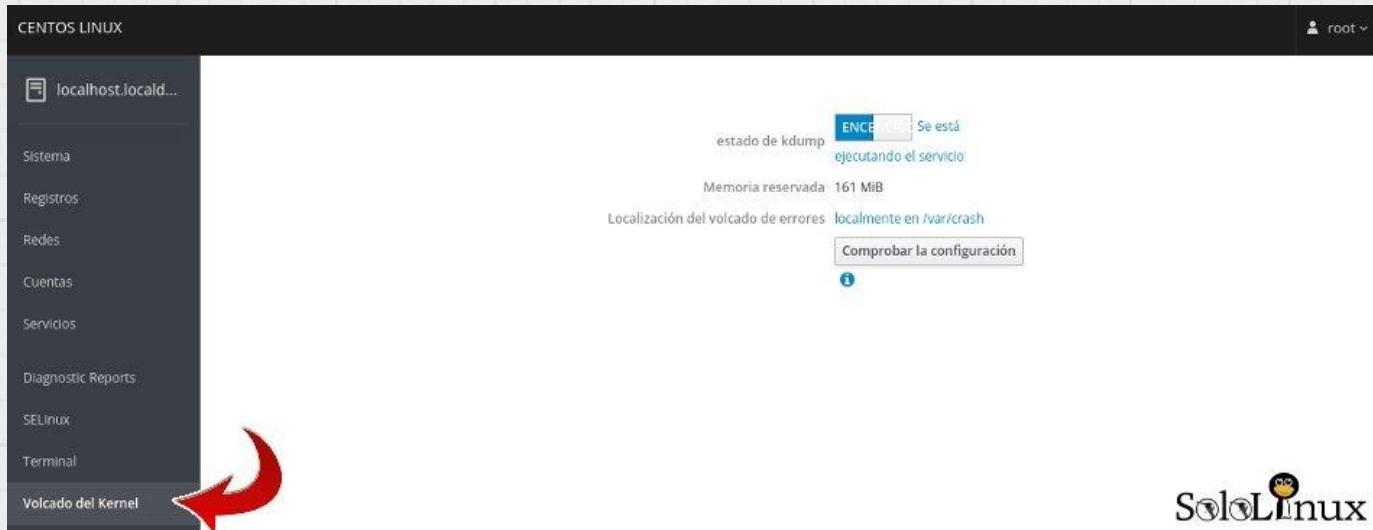
CENTOS LINUX

- localhost.localdomain...
- Sistema
- Registros
- Redes
- Cuentas
- Servicios
- Diagnostic Reports**
- SELinux
- Terminal**
- Volvido del Kernel

```
[root@62-210-75-231 ~]# bash libera.sh
<----->
<----->
Comprobando estado de memoria
total        used        free       shared      buff/cache   available
Mem:    7995404     744708     5902720      92548      1347976     6838312
Swap:  1048572          0     1048572
OK - Comprobacion completada
<----->
"Limpieza de memoria cache y swap"
<----->
Deshabilitando HTTP
OK - Http deshabilitado
<----->
Deshabilitando Swap
OK - Swap deshabilitado
<----->
Libero de pagecaches, dentries e inodes
vm.dirty_caches = 3
OK - Swap liberado
<----->
Libro cuando la Swap
OK - Swap habilitado
<----->
Habilitar HTTP y reiniiciar MariaDB
Restarting nginx (via systemctl):
Redirecting to /bin/systemctl start httpd.service
[ OK ]
OK - Http habilitado
<----->
Habilitando Memcached
Nota: Reenviando petición a 'systemctl enable memcached.service'.
Redirecting to /bin/systemctl start memcached.service
OK - Memcached habilitado
<----->
Redirecting to /bin/systemctl restart ddos.service
Failed to restart ddos.service: Unit not found.
Comprobacion rutinaria
Script liberador.sh
Created by Sergio.G.B.
https://www.sololinux.es
<----->
total        used        free       shared      buff/cache   available
Mem:    7995404     628088     7078572      92556      288744     7019592
Swap:  1048572          0     1048572
.... TODO CORRECTO.....
<----->
<----->
[root@62-210-75-231 ~]#
```

SoloLinux

El registro del **volcado del kernel** recomiendo que lo tengas activado. Puede resultar muy útil en caso de problema en el servidor.



Bueno, ya ves que es una herramienta realmente interesante, ademas su consumo y peso es casi nulo. Te gustara!!!!!!

Si por algún motivo la quieres desinstalar, ejecuta:

yum remove cockpit*

```
[root@ ~]# yum remove cockpit*
Complementos cargados:fastestmirror, langpacks
Resolviendo dependencias
--> Ejecutando prueba de transacción
---> Paquete cockpit.x86_64 0:176-4.el7.centos debe ser eliminado
---> Paquete cockpit-bridge.x86_64 0:176-4.el7.centos debe ser eliminado
---> Paquete cockpit-system.noarch 0:176-4.el7.centos debe ser eliminado
---> Paquete cockpit-ws.x86_64 0:176-4.el7.centos debe ser eliminado
--> Resolución de dependencias finalizada

Dependencias resueltas

=====
          Package           Arquitectura Versión           Repositorio   Tamaño
=====
Eliminando:
cockpit        x86_64      176-4.el7.centos     @extras       47 k
cockpit-bridge x86_64      176-4.el7.centos     @extras      802 k
cockpit-system noarch     176-4.el7.centos     @extras      1.2 M
cockpit-ws     x86_64      176-4.el7.centos     @extras      1.5 M

Resumen de la transacción
=====
Eliminar: 4 Paquetes

Tamaño instalado: 3.6 M
Está de acuerdo [s/N]:|
```



Si te gusto el articulo, compártelo.

Las reglas de iptables que debes conocer

Las reglas de Firewall Iptables más comunes

SoloLinux

Las **reglas** de **iptables** que debes conocer.

En este artículo veremos las reglas de iptables más utilizadas, por tanto son las que debes conocer.

Si eres, o quieres ser un **administrador de sistemas (sysadmin)** harás uso de ellas continuamente.

Las vemos.



Las reglas de iptables que debes conocer

En todos los ejemplos propuestos usamos la [interfaz de red](#) “eth0“.

Puedes comprobar tu interfaz con el comando:

`ifconfig`

Las vemos:

Borrar todas las reglas:

`iptables -F` o `iptables --flush`

Modificar la política de las reglas predeterminada:

Por defecto las reglas de [iptables](#) aceptan la conexión, podemos modificar este comportamiento ejecutando lo siguiente (depende de tus necesidades).

`iptables -P INPUT DROP`

`iptables -P FORWARD DROP`

`iptables -P OUTPUT DROP`

Bloquear una dirección ip específica:

Donde “**BLOCK_THIS_IP**” debes insertar la IP a bloquear.

```
iptables -A INPUT -i eth0 -s "$BLOCK_THIS_IP" -j DROP
iptables -A INPUT -i eth0 -p tcp -s "$BLOCK_THIS_IP" -j DROP
```

Permitir todo el tráfico entrante por [SSH](#):

```
iptables -A INPUT -i eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
```

Permitir trafico entrante por SSH desde una red especifica, o una IP definida:

Modifica “192.168.100.X” por tu rango, o por una IP especifica.

```
iptables -A INPUT -i eth0 -p tcp -s 192.168.100.0/24 --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
```

Permitir todo el trafico entrante HTTP y HTTPS:

HTTP por el puerto 80.

```
iptables -A INPUT -i eth0 -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
```

HTTPS por el puerto 443

```
iptables -A INPUT -i eth0 -p tcp --dport 443 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT
```

Combinar reglas con “multiport“:

Permite combinar reglas incluso a varios puertos.

En el ejemplo aceptamos todo el tráfico entrante por SSH, HTTP y HTTPS.

```
iptables -A INPUT -i eth0 -p tcp -m multiport --dports 22,80,443 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp -m multiport --sports 22,80,443 -m state --state ESTABLISHED -j ACCEPT
```

Permitir la salida por SSH:

```
iptables -A OUTPUT -o eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
```

Permitir salida de trafico por SSH desde una red especifica, o una IP definida:

Modifica “192.168.100.X” por tu rango, o por una IP especifica.

```
iptables -A OUTPUT -o eth0 -p tcp -d 192.168.100.0/24 --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
```

Permitir la salida de trafico por HTTP y HTTPS:

HTTP por el puerto 80.

```
iptables -A OUTPUT -o eth0 -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
```

HTTP por el puerto 443.

```
iptables -A OUTPUT -o eth0 -p tcp --dport 443 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT
```

Equilibrar la carga del tráfico web entrante:

Para esta interesante función usaremos la extensión “[iptables nth](#)“.

En el ejemplo balancearemos el **tráfico HTTPS** sobre tres direcciones IP (tres [servidores](#)). Por cada 3º paquete, carga el siguiente servidor (con el contador 0). Los servidores tienen las siguientes ip:

192.168.1.117:443
192.168.1.118:443
192.168.1.119:443

```
iptables -A PREROUTING -i eth0 -p tcp --dport 443 -m state --state NEW -m nth --counter 0 --every 3 --packet 0 -j DNAT --to-destination 192.168.1.117:443
iptables -A PREROUTING -i eth0 -p tcp --dport 443 -m state --state NEW -m nth --counter 0 --every 3 --packet 1 -j DNAT --to-destination 192.168.1.118:443
iptables -A PREROUTING -i eth0 -p tcp --dport 443 -m state --state NEW -m nth --counter 0 --every 3 --packet 2 -j DNAT --to-destination 192.168.1.119:443
```

Permitir que nos hagan [ping](#) desde el exterior:

```
iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT
```

Hacer ping desde el servidor (al exterior):

```
iptables -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
```

Permitir el acceso “[Loopback](#)“:

```
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
```

Permitir que la red interna acceda a la externa:

Damos por supuesto que el servidor cuenta con dos [tarjetas de red](#), en este caso “**eth0**” y “**eth1**“. En el ejemplo, **eth1** conecta con la externa (internet), y **eth0** a la red interna (por ejemplo: 192.168.100.x).

```
iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT
```

Permitir conexiones [DNS](#) salientes:

```
iptables -A OUTPUT -p udp -o eth0 --dport 53 -j ACCEPT
iptables -A INPUT -p udp -i eth0 --sport 53 -j ACCEPT
```

Permitir conexiones [NIS](#) (Network Information System):

Para conocer los puertos [ypbind](#)...

```
rpcinfo -p | grep ypbind
```

Ahora permitimos la conexión entrante al puerto 111 y los puertos que usa ypbind (en este caso 863 y 865).

```
iptables -A INPUT -p tcp --dport 111 -j ACCEPT
iptables -A INPUT -p udp --dport 111 -j ACCEPT
iptables -A INPUT -p tcp --dport 863 -j ACCEPT
iptables -A INPUT -p udp --dport 863 -j ACCEPT
iptables -A INPUT -p tcp --dport 865 -j ACCEPT
iptables -A INPUT -p udp --dport 865 -j ACCEPT
```

Permitir el tráfico de “[Rsync](#)” desde una red o ip específica:

Donde “192.168.100/24”, coloca tu red o ip.

```
iptables -A INPUT -i eth0 -p tcp -s 192.168.100.0/24 --dport 873 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 873 -m state --state ESTABLISHED -j ACCEPT
```

Permitir la conexión [MySQL](#) desde una red específica (exclusivamente):

```
iptables -A INPUT -i eth0 -p tcp -s 192.168.100.0/24 --dport 3306 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 3306 -m state --state ESTABLISHED -j ACCEPT
```

Permitir el tráfico de [Sendmail](#) o [Postfix](#) (puerto 25):

```
iptables -A INPUT -i eth0 -p tcp --dport 25 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 25 -m state --state ESTABLISHED -j ACCEPT
```

Permitir el tráfico de [IMAP](#) / [IMAPS](#):**IMAP**

```
iptables -A INPUT -i eth0 -p tcp --dport 143 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 143 -m state --state ESTABLISHED -j ACCEPT
```

IMAPS

```
iptables -A INPUT -i eth0 -p tcp --dport 993 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 993 -m state --state ESTABLISHED -j ACCEPT
```

Permitir el tráfico de [POP3](#) / [POP3S](#):**POP3**

```
iptables -A INPUT -i eth0 -p tcp --dport 110 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 110 -m state --state ESTABLISHED -j ACCEPT
```

POP3S

```
iptables -A INPUT -i eth0 -p tcp --dport 995 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 995 -m state --state ESTABLISHED -j ACCEPT
```

Bloquear [ataques Dos](#):

Explicación y ejemplo:

-m limit: Usa la extensión límite de iptables.

-limit 25/minute: Limita a un máximo de 25 conexiones por minuto. Modifica este valor según tus necesidades.

-limit-burst 100: Este valor indica que el límite / minuto se aplicará solo después de que el número total de conexiones haya alcanzado el nivel límite de ráfaga.

```
iptables -A INPUT -p tcp --dport 80 -m limit --limit 25/minute --limit-burst 100 -j ACCEPT
```

En [un anterior articulo](#) tienes un desarrollo más extenso.

[Port Forwarding](#) (enrutar el tráfico de un puerto a otro):

En el ejemplo enrutamos todo el tráfico que accede al puerto 450, al puerto 22 (ssh).

```
iptables -t nat -A PREROUTING -p tcp -d 192.168.100.117 --dport 450 -j DNAT --to 192.168.100.117:22
```

Recuerda que entonces también debes abrir el puerto 450.

```
iptables -A INPUT -i eth0 -p tcp --dport 450 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 450 -m state --state ESTABLISHED -j ACCEPT
```

Registrar los dropped packets:

Primero creamos la cadena LOGGING.

```
iptables -N LOGGING
```

Nos aseguramos que todas las conexiones entrantes restantes salten a la cadena LOGGING.

```
iptables -A INPUT -j LOGGING
```

Registraremos los paquetes con un prefijo personalizado, esta practica es para identificarlos correctamente.

```
iptables -A LOGGING -m limit --limit 2/min -j LOG --log-prefix "IPTables Packet Dropped: " --log-level 7
```

Y por ultimo...

```
iptables -A LOGGING -j DROP
```

Aquí concluye este manual, si te gusto compártelo.

Como instalar Xampp en Linux

Como instalar Xampp en Linux, independientemente de la [distribución](#) que uses.

En este artículo vemos como instalar “**XAMPP**” en cualquier [distribución linux](#), ademas insertamos imágenes para que puedas visualizar la instalación en consola, o de forma gráfica.

Básicamente XAMPP instala un servidor local.



Con este servidor local, puedes probar aplicaciones o sitios web antes de subirlos a tu Hosting o servidor web remoto.

Los [comandos](#) y procedimientos que veremos en este artículo, se han ejecutado con éxito en sistemas tan dispares como pueden ser [Ubuntu 18.04 LTS](#) y [OpenSuse 42.3](#) (deb y rpm). Como “**Xampp**” es una aplicación muy conocida, no nos andaremos con más rodeos y nos dirigimos directamente al tutorial “[Como instalar Xampp paso a paso en cualquier Linux](#)”.

Como instalar Xampp en Linux

Lo primero que haremos es descargar la versión que más nos interese, en este caso (ejemplo) la última estable, “[xampp-linux-x64-7.3.1-0-installer.run](#)“.

- [Descargar Xampp](#)

Ahora vamos a la carpeta de descargas.

cd /home/tu-usuario/Downloads

Modificamos permisos.

```
## Coloca tu versión
chmod 755 xampp-linux-x64-7.3.1-0-installer.run
```

Verificamos que puede ser ejecutado.

ls -l xampp-linux-x64-7.3.1-0-installer.run

La salida [-rwxr](#) nos indica que el archivo puede ser ejecutado por el usuario cuyo nombre también se imprime en la respuesta.

```
sergio@linux-ozl1:~/Descargas> chmod 755 xampp-linux-x64-7.3.1-0-installer.run
sergio@linux-ozl1:~/Descargas> ls -l xampp-linux-x64-7.3.1-0-installer.run
-rwxr-xr-x 1 sergio users 142524589 feb 20 12:38 xampp-linux-x64-7.3.1-0-installer.run
SoloLinux
```

Ya tenemos el paquete preparado, así que vamos a ver **como instalar Xampp**.

Ejecutamos:
en nuestro caso... **sudo ./paquete-a-instalar**
sudo ./xampp-linux-x64-7.3.1-0-installer.run

Debo decirte que no todas las distribuciones (o dependiendo de tu configuración) no permiten la instalación en modo gráfico. En este articulo vemos las dos.

En modo consola es tan simple como esto...

```
sergio@linux-ozli:~/Descargas> sudo ./xampp-linux-x64-7.3.1-0-installer.run
[sudo] password for root:
-----
Welcome to the XAMPP Setup Wizard.

-----
Select the components you want to install; clear the components you do not want
to install. Click Next when you are ready to continue.

XAMPP Core Files : Y (Cannot be edited)

XAMPP Developer Files [Y/n] :y

Is the selection above correct? [Y/n]: y

-----
Installation Directory

XAMPP will be installed to /opt/lampp
Press [Enter] to continue:

-----
Setup is now ready to begin installing XAMPP on your computer.

Do you want to continue? [Y/n]: y

-----
Please wait while Setup installs XAMPP on your computer.

Installing
0% ----- 50% ----- 100%
#####
```



Si quieres usar la consola:

Iniciar Xampp	sudo /opt/lampp/lampp start
Detener Xampp	sudo /opt/lampp/lampp stop
Reiniciar Xampp	sudo /opt/lampp/lampp restart

En modo gráfico inicia el auto instalador.





Recuerda la ruta indicada, **Xampp** y sus servicios se instalaran en ella.

Por defecto esta marcada la instalación de **Bitnami** que es (entre otras cosas) un auto instalador de aplicaciones. A titulo personal, yo siempre lo desactivo, considero que es un peso innecesario para realizar pruebas en local. Es tu decisión instalarlo o no.



Comienza la instalación.



Pulsa en “Manage Servers”.





Tal como te indica la siguiente imagen, desde aquí puedes controlar o manejar todos los servicios.



Una vez activados puedes volver a la pantalla principal pulsando en “Welcome”



Si activaste el servicio puedes comprobar que el sitio web ya está operativo pulsando en **Application**.

Como puedes ver ya funciona correctamente.
Claro... ahora te preguntaras...

“vale muy bonito, ya funciona, pero donde pongo los archivos de la web en Xampp?”

La respuesta es rápida, aquí...

/opt/lampp/htdocs

Si recuerdas durante la instalación ya te indique que debías recordar la ruta de **Xampp**, **/opt/lampp/htdocs** no la olvides.

Si por alguna razón quieras desinstalar Xampp, ejecuta lo siguiente:

```
cd /opt/lampp
```

Podemos visualizar los archivos contenidos.

```
dir
```

```
linux-ozli:/home/sergio # cd /opt/lampp
linux-ozli:/opt/lampp # dir
total 12476
drwxr-xr-x  5 root  root      4096 feb 20 16:08 apache2
drwxrwxr-x  2 root  root    12288 feb 20 16:09 bin
drwxr-xr-x  2 root  root      4096 feb 20 16:10 build
drwxr-xr-x  2 root  root      4096 feb 20 16:09 cgi-bin
-rw-r--r--  1 root  root   86263 nov  1 00:54 COPYING.thirdparty
-rwxr-xr-x  1 root  root   27400 feb 20 16:08 ctlscript.sh
drwxr-xr-x  2 root  root      4096 feb 20 16:10 docs
drwxrwxr-x  3 root  root      4096 feb 20 16:09 error
drwxr-xr-x  8 root  root      4096 feb 20 16:10 etc
```



Para desinstalar Xampp o borrar Xampp ejecutamos “uninstall”.

```
sudo ./uninstall
```

```
linux-ozli:/opt/lampp # sudo ./uninstall
Do you want to uninstall XAMPP and all of its modules? [Y/n]: y

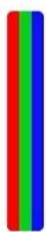
-----
Uninstall Status

Uninstalling XAMPP
0% ----- 50% ----- 100%
#####
Info: Uninstallation completed
Press [Enter] to continue:
```



Si te fue útil el articulo, compártelo.

Averiguar el códec de un archivo multimedia con:



Media Info

Sololinux

Sololinux

Averiguar el **códec** de un **archivo multimedia** con **MediaInfo**.

MediaInfo es una aplicación “**Open Source**” que te expone todos los datos técnicos e información de cualquier archivo multimedia que tu le indiques.

Su uso es bastante simple, tan solo tienes que indicar el, o los archivos de los cuales quieras averiguar el códec que usan para que la propia aplicación de forma automatizada te imprima una ficha con sus datos técnicos.

Los datos que puedes obtener son los siguientes:

- **Vídeo:** códec, relación de aspecto, fotogramas por segundo, tasa de bits o bitrate.
- **Audio:** códec, tasa de muestreo, canales, idioma, tasa de bits o bitrate.
- **General:** Título, autor, director, álbum, número de pista, fecha, duración.
- **Texto:** subtítulo, idioma del subtítulo.
- **Capítulos:** número de capítulos, lista de capítulos.

Soporta múltiples formatos, por ejemplo:

- **Vídeo:** MKV, OGM, AVI, DivX, WMV, QuickTime, Real, MPEG-1, MPEG-2, MPEG-4, DVD (VOB), etc...
- **Audio:** OGG, MP3, WAV, RA, AC3, DTS, AAC, M4A, AU, AIFF, etc...
- **Subtítulos:** SRT, SSA, ASS, SAMI, etc...

Su instalación y uso es sencillo, lo vemos.

Saber el códec de un archivo con MediaInfo

MediaInfo viene de forma predeterminada en la gran mayoría de **repositorios** de todas las **distribuciones linux**, en el ejemplo lo instalamos en **Ubuntu** y todos sus derivados.

Instalamos la versión con “**GUI**” incluido, para que me entiendas... con interfaz gráfica.

```
apt-get install mediainfo-gui
```

Para otras distribuciones puedes visitar
[su pagina oficial](#).

Una vez instalado, puedes abrir la aplicación
desde tu menú de aplicaciones, veras lo siguiente...



Una vez en la aplicación puedes seleccionar el archivo/s de diversas formas.

Por orden en la imagen:

1. **File:** Buscar y seleccionar cualquier archivo.
2. **Icono Multimedia:** Buscar y seleccionar archivos multimedia.
3. **Icono Carpeta:** Buscar y seleccionar carpetas.

El uso de la opción “File” es la más cómoda y rápida.



La salida es inmediata, vemos un par de ejemplos:

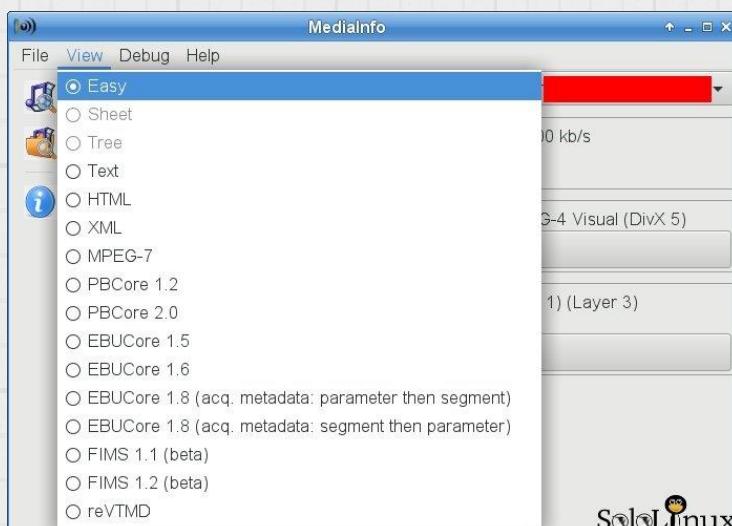
Left Window (MP4 File):

- File: /run/media/sergio/7E95-86AB/[REDACTED].mp4
- General: MPEG-4 (Base Media / Version 2): 42.6 MiB, 3 min 40 s
- Video: 1 Video stream: AVC, 1280x720 (16:9), at 30.000 FPS, AVC (High@L3.1) (CABAC / 3 Ref Frames)
- Audio: 1 Audio stream: AAC, 128 kb/s, 44.1 kHz, 2 channels, AAC (LC)

Right Window (AVI File):

- File: /run/media/sergio/7E95-86AB/[REDACTED]
- General: AVI: 687 MiB, 1 h 46 min
- Video: 1 Video stream: MPEG-4 Visual (DivX 5), 762 kb/s (6 951 kb/s), 536*388 (1.381), at 25.000 FPS, MPEG-4 Visual (DivX 5)
- Audio: 1 Audio stream: MPEG Audio (MP3), 128 kb/s, 44.1 kHz, 2 channels, MPEG Audio (MP3) (Version 1) (Layer 3)

En la pestaña “View“, podemos seleccionar el formato de la impresión de datos.



Una vez que ya sabes el tipo de archivo, formato o la información que te interesa, puedes cerrar la aplicación.

Fácil verdad?, si te parece interesante el artículo, compártelo...



Los [script bash](#) que necesita un **SysAdmin** para manejar un [servidor](#) de forma básica.

En “[SoloLinux](#)” intentamos publicar los máximos **script bash** posibles, unos a modo didáctico, otros como simple curiosidad, pero sobre todo hacemos hincapié en los que creemos que necesita un **SysAdmin** (administrador de sistemas).

En este artículo insertamos los que son prácticamente indispensables, y son cinco...

- **Uso del disco.**
- **Backup incremental del “./home”.**
- **Uso de la cpu.**
- **Agregar usuarios al sistema.**
- **Backup de la base de datos.**

Existen en “[SoloLinux](#)” scripts con funciones más avanzadas. Sin embargo, estos te ofrecen una visión general muy rápida, incluyendo en alguno el envío de un [correo electrónico](#) en caso de superar el umbral máximo definido.

Script bash que necesita un SysAdmin

Uso del disco:

Este **script bash** te enviará un [email](#) cuando el disco se esté quedando sin espacio (ejecutar con una [tarea cron](#)).

```
#!/bin/sh
df -H | grep -vE '^Filesystem|tmpfs|cdrom' | awk '{ print $5 " " $1 }' | while read output;
do
    echo $output
    used=$(echo $output | awk '{ print $1}' | cut -d'%' -f1 )
    partition=$(echo $output | awk '{ print $2 }')
    if [ $used -ge 80 ]; then
        echo "The partition \"$partition\" on $(hostname) has used $used% at $(date)" | mail
        -s "Disk space alert: $used% used" your@email.com
    fi
done
```

Backup incremental del “./home”:

Este script realizara una copia de seguridad incremental en un disco externo (que este montado). De forma predeterminada el **backup** sera de la carpeta “**/home**”, puedes agregar o modificar las carpetas que te interesen fácilmente.

Revisa y modifica las rutas establecidas

```
#!/bin/bash
# this needs to be a lot more general, but the basic idea is it makes
# rotating backup-snapshots of /home whenever called
unset PATH
# ----- system commands used by this script -----
ID=/usr/bin/id;
ECHO=/bin/echo;
MOUNT=/bin/mount;
RM=/bin/rm;
MV=/bin/mv;
CP=/bin/cp;
TOUCH=/bin/touch;
RSYNC=/usr/bin/rsync;
# ----- file locations -----
MOUNT_DEVICE=/dev/hdb1;
SNAPSHOT_RW=/root/snapshot;
EXCLUDES=/usr/local/etc/backup_exclude;
# ----- the script itself -----
# make sure we're running as root
if (( `\$ID -u` != 0 )); then { $ECHO "Sorry, must be root. Exiting..."; exit; } fi
# attempt to remount the RW mount point as RW; else abort
$MOUNT -o remount,rw $MOUNT_DEVICE $SNAPSHOT_RW ;
if (( $? )); then
{
    $ECHO "snapshot: could not remount $SNAPSHOT_RW readwrite";
    exit;
}
fi;
# rotating snapshots of /home (fixme: this should be more general)
# step 1: delete the oldest snapshot, if it exists:
if [ -d $SNAPSHOT_RW/home/hourly.3 ] ; then      \
$RM -rf $SNAPSHOT_RW/home/hourly.3 ;           \
fi;
# step 2: shift the middle snapshots(s) back by one, if they exist
if [ -d $SNAPSHOT_RW/home/hourly.2 ] ; then      \
$MV $SNAPSHOT_RW/home/hourly.2 $SNAPSHOT_RW/home/hourly.3 ; \
fi;
if [ -d $SNAPSHOT_RW/home/hourly.1 ] ; then      \
$MV $SNAPSHOT_RW/home/hourly.1 $SNAPSHOT_RW/home/hourly.2 ; \
fi;
# step 3: make a hard-link-only (except for dirs) copy of the latest snapshot,
# if that exists
if [ -d $SNAPSHOT_RW/home/hourly.0 ] ; then      \
$CP -al $SNAPSHOT_RW/home/hourly.0 $SNAPSHOT_RW/home/hourly.1 ; \
fi;
# step 4: rsync from the system into the latest snapshot (notice that
# rsync behaves like cp --remove-destination by default, so the destination
# is unlinked first. If it were not so, this would copy over the other
# snapshot(s) too!
$RSYNC                                \
--delete --delete-excluded             \
--exclude-from="$EXCLUDES"            \
/home/ $SNAPSHOT_RW/home/hourly.0 ;
# step 5: update the mtime of hourly.0 to reflect the snapshot time
$TOUCH $SNAPSHOT_RW/home/hourly.0 ;
# and thats it for home.
# now remount the RW snapshot mountpoint as readonly
$MOUNT -o remount,ro $MOUNT_DEVICE $SNAPSHOT_RW ;
if (( $? )); then
{
    $ECHO "snapshot: could not remount $SNAPSHOT_RW readonly";
    exit;
}
```

#!/bin/bash

#!/bin/bash

Uso de la cpu:

Este **script bash** te enviara un [email](#) cuando la [cpu](#) este alcanzando sus limites (ejecutar con una [tarea cron](#)).

```
#!/bin/bash
while [ true ] ;do
used=`free -m |awk 'NR==3 {print $4}'` 
if [ $used -lt 1000 ] && [ $used -gt 800 ]; then
echo "Free memory is below 1000MB. Possible memory leak!!!" | /bin/mail -s "HIGH MEMORY
ALERT!!!" mimail@mydomain.com
fi
sleep 5
done
```

Agregar usuarios al sistema:

Este [bash](#) permite al [root](#) o usuario con permisos, agregar nuevos usuarios al sistema de manera rapida y sencilla, solo debe escribir el nombre de usuario y la [contraseña](#).

```
#!/bin/bash
# Script to add a user to Linux system
if [ $(id -u) -eq 0 ]; then
    read -p "Enter username : " username
    read -s -p "Enter password :" password
    egrep "^\$username" /etc/passwd >/dev/null
    if [ $? -eq 0 ]; then
        echo "$username exists!"
        exit 1
    else
        pass=$(perl -e 'print crypt($ARGV[0], "password")' $password)
        useradd -m -p $pass $username
        [ $? -eq 0 ] && echo "User has been added to system!" || echo "Failed to add a user!"
    fi
else
    echo "Only root may add a user to the system"
    exit 2
fi
```

```
Enter username : sololinux
Enter password : |
```

Backup de la base de datos:

Este **script**, es de lo más básico. Se limita a hacer un **backup de la base de datos** en la ruta que le indiques.

Modifica los datos y rutas establecidas por defecto

```
#!/bin/sh
now="$(date +'%d_%m_%Y_%H_%M_%S')"
filename="db_backup_$now".gz
backupfolder="/var/www/vhosts/example.com/httpdocs/backups"
fullpathbackupfile="$backupfolder/$filename"
logfile="$backupfolder/backup_log_$(date +'%Y_%m').txt
echo "mysqldump started at $(date +'%d-%m-%Y %H:%M:%S')" >> "$logfile"
mysqldump --user=mydbuser--password=mypass --default-character-set=utf8 mydatabase | gzip >
"$fullpathbackupfile"
echo "mysqldump finished at $(date +'%d-%m-%Y %H:%M:%S')" >> "$logfile"
chown myuser "$fullpathbackupfile"
chown myuser "$logfile"
echo "file permission changed" >> "$logfile"
find "$backupfolder" -name db_backup_* -mtime +8 -exec rm {} \;
echo "old files deleted" >> "$logfile"
echo "operation finished at $(date +'%d-%m-%Y %H:%M:%S')" >> "$logfile"
echo "*****" >> "$logfile"
exit 0
```

Espero te ayude alguno de estos **scripts** que “**necesita un SysAdmin**“.

Ejecutar Webex en Ubuntu 18.04

Ejecutar Webex en [Ubuntu 18.04](#), [Ubuntu 16.04](#), [Linux Mint 19/19.1](#) y todos sus derivados.

Cada vez es más común trabajar de manera compartida a través de la Web, es cómodo, rápido, directo, y sobre todo no importan las distancias.

[Messenger](#), [skype](#), etc..., así como el uso de videoconferencias era lo más común hasta poco tiempo. A nivel a académico, de investigación y empresarial, las formulas de conexión mencionadas están en desuso, tenemos “[Webex](#)”, bueno mejor dicho “[Cisco Webex](#)” ([cisco compro Webex](#)).

Webex es una solución que ofrece una serie de excelentes herramientas que brindan una experiencia de colaboración segura, confiable y por demanda en la web.

Cisco apostó muy fuerte al adquirir Webex en el 2007 por 3.200 millones de dólares.

Permite tener reuniones en cualquier lugar y en tiempo real, presentaciones, documentos, compartir cualquier objeto o el escritorio completo (incluyendo las aplicaciones), ventanas, directorios de los archivos que tengas abiertos. También es posible conceder permisos a los asistentes para que puedan anotar, guardar, imprimir y mostrar vistas de los documentos o presentaciones que se hayan compartido.

Es muy usada para realizar reuniones virtuales, ventas remotas, ofrecer soporte técnico y ayuda. Cualquier tipo de evento virtual tiene cabida en “[Webex](#)”.

Pero tiene un problema, no es 100% compatible con linux.

El principal hándicap es que solo es compatible con distribuciones de [32bits](#) (actualmente en desuso), y aun así con ciertas limitaciones que se describen en un articulo de la propia [web de Cisco](#).

Aun con esa limitación sigue siendo una aplicación muy útil, pues si podemos compartir la pantalla y el audio.

Vemos que tenemos que hacer para poder ejecutar **Webex en Ubuntu**.

Ejecutar Webex en Ubuntu 18.04 y más

Debemos de agregar soporte para aplicaciones con **arquitectura de 32 bits**.

```
sudo dpkg --add-architecture i386
```

Instalamos las bibliotecas necesarias que necesitan los [ELF-32](#).

```
sudo apt-get update
sudo apt-get install libc6:i386 libncurses5:i386 libstdc++6:i386
```

Firefox 52 ESR de 32 bits

[Firefox](#) ya no ofrece soporte [NPAPI](#) ([comunicado oficial](#)), tan solo para [Adobe Flash](#), por tanto nos obliga a instalar la versión **52 ESR** (soporte de largo plazo) que si que tiene soporte NPAPI. Descargamos la versión **Firefox ESR para Linux** 32 bits en tu lenguaje desde la pagina oficial. Creamos la carpeta **Webex** y descomprimiremos **Firefox ESR** en ella.

Descarga tu Firefox ESR 32 bits — [AQUÍ](#) —

```
sudo mkdir -p /opt/webex/
sudo tar -xjvf firefox*esr*.tar.bz2 -C /opt/webex/
```

Instalamos las bibliotecas que necesita **Firefox ESR**.

```
sudo apt-get install \
libgtk-3-0:i386 \
libasound2:i386 \
libdbus-glib-1-2:i386 \
libxt6:i386 \
libxtst6:i386 \
libcanberra-gtk-module:i386 \
libcanberra-gtk-3-module:i386 \
topmenu-gtk3:i386
```

Instalar JRE de 32 bits

Descargamos el [*.tar.gz](#) de **Oracle JRE** para Linux (versión i586), desde su [—pagina oficial—](#)

Java SE Runtime Environment 8u201

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux x86	68.1 MB	jre-8u201-linux-i586.rpm
Linux x86	83.8 MB	jre-8u201-linux-i586.tar.gz
Linux x64	64.91 MB	jre-8u201-linux-x64.rpm
Linux x64	80.73 MB	jre-8u201-linux-x64.tar.gz
Mac OS X x64	76.18 MB	jre-8u201-macosx-x64.dmg
Mac OS X x64	67.77 MB	jre-8u201-macosx-x64.tar.gz
Solaris SPARC 64-bit	46.27 MB	jre-8u201-solaris-sparcv9.tar.gz
Solaris x64	50.14 MB	jre-8u201-solaris-x64.tar.gz
Windows x86 Online	1.87 MB	jre-8u201-windows-i586-ifw.exe
Windows x86 Offline	63.53 MB	jre-8u201-windows-i586.exe
Windows x86	66.51 MB	jre-8u201-windows-i586.tar.gz
Windows x64	71.44 MB	jre-8u201-windows-x64.exe
Windows x64	71.29 MB	jre-8u201-windows-x64.tar.gz

SoloLinux

Desde la carpeta donde descargaste **Oracle JRE**, ejecuta lo siguiente:

```
sudo tar -xzvf jre*linux-i586.tar.gz -C /opt/webex/
sudo mv /opt/webex/jre* /opt/webex/jre
```

Habilitamos el soporte [Java](#) en **Firefox 52**.

```
sudo mkdir /opt/webex/firefox/plugins/
ln -s \
/opt/webex/jre/lib/i386/libawt.so \
/opt/webex/jre/lib/i386/libjawt.so \
/opt/webex/jre/lib/i386/libnpjp2.so \
/opt/webex/firefox/plugins
```

Antes de ejecutar Webex

Instalamos las librerías necesarias para su ejecución.

```
sudo apt-get install \
libpangoxtf-1.0-0:i386 \
libxft2:i386 \
libpango-1.0-0:i386 \
libxmu6:i386 \
libxv1:i386 \
libasound2-plugins:i386
```

Creamos un [script bash](#) de inicio (copia y pega en tu consola).

```
cat << 'EOF' | sudo tee /opt/webex/firefox.sh
#!/bin/bash

export ENV_HOME=/opt/webex
export FIREFOX_HOME=$ENV_HOME/firefox
export MOZ_PLUGIN_PATH=$ENV_HOME/firefox/plugins
export JAVA_HOME=$ENV_HOME/jre
export PATH=$JAVA_HOME/bin:$PATH

#export JPI_PLUGIN2_DEBUG=I

$FIREFOX_HOME/firefox --no-remote -P
EOF
```

Le concedemos permisos.

```
sudo chmod a+x /opt/webex/firefox.sh
```

Crea un enlace simbólico.

```
sudo ln -s /opt/webex/firefox.sh /usr/local/bin/firefox-i386
```

Este [script](#) iniciará una nueva sesión de Firefox con el [ProfileManager](#). Debes seleccionar un perfil que ya exista en **Firefox**, o crear uno nuevo.



Ejecutar Webex

Para iniciar sesión en Webex, ejecutas como cualquier otro *.sh, el script que creamos hace un momento.

Si webex.com te envía un archivo jnl, Firefox te preguntara qué hacer con él, simplemente descárgalo. Otro detalle... en Preferencias de Firefox> Aplicaciones, configura una acción para los [archivos JNLP](#): Deben abrirse con javaws(Java Web Start) que lo encontraras en la siguiente ruta:

`/opt/webex/jre/bin/javaws`

Si notas algún comportamiento extraño, tal vez te falte alguna biblioteca, ejecuta lo siguiente:

```
ldd ~/.webex/T30_MC/*.so | grep "not found"
```

Para saber que paquetes contienen las bibliotecas accede a "<https://packages.ubuntu.com/>".

Comparte el articulo en tus redes sociales y ayúdanos a crecer.

Proteger Wordpress **FAIL2BAN**



Proteger [WordPress](#) con [Fail2ban](#).

Este método que explicaremos hoy, no solo es valido para proteger WordPress con [Fail2ban](#), también es aplicable a la gran mayoría de [CMS](#) o aplicaciones que crees tu mismo.

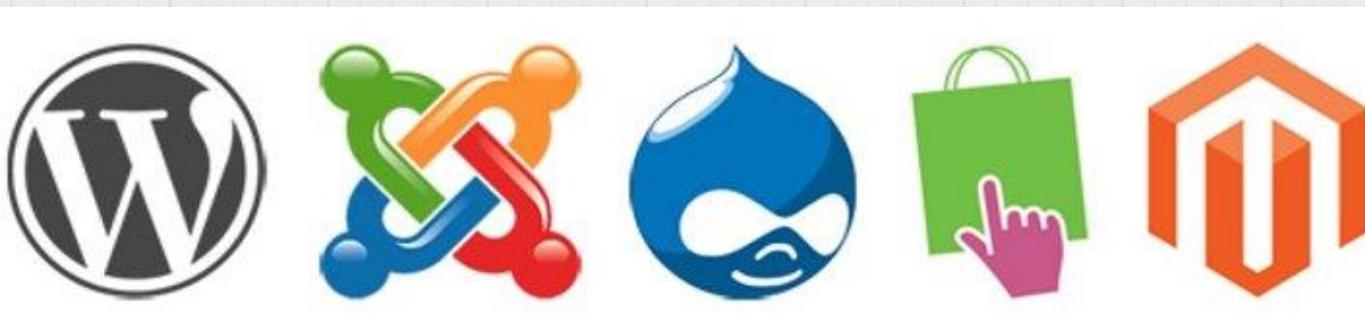
Por ejemplo:

- [WordPress](#)
- [Joomla](#)
- [Drupal](#)
- [Prestashop](#)
- [Magento](#)
- [Aplicaciones propias](#)
- etc...

Este sistema esta indicado para protección contra Web Exploits.

Lo que haremos es crear un nuevo archivo de configuración en “[filter.d](#)”, y después el “[jail](#)” que corresponda.

Vemos como proteger [WordPress](#), [Joomla](#), [Drupal](#), [Prestashop](#), etc...



Proteger WordPress con Fail2ban (y otros CMS)

Creamos el archivo “**webexploits.conf**” en la carpeta “**filter.d**”.

```
nano /etc/fail2ban/filter.d/webexploits.conf
```

Copia y pega lo siguiente:

[ENTRA AQUÍ Y COPIA LO QUE HAY EN
EL ARCHIVO TXT](#)

El archivo “webexploits.conf” que te propongo es bastante completo (los CMS más utilizados), aun así puedes borrar o agregar rutas que te interesen ser protegidas.

Guarda el archivo y cierra el editor.



Si utilizamos **Apache**, copia y pega lo siguiente:

```
[webexploits]
enabled = true
port    = http,https
filter   = webexploits
logpath = %(apache_access_log)s
maxretry = 3
```

Guarda el archivo y cierra el [editor](#).

Ya lo tenemos... solo falta **reiniciar Fail2ban**.

```
sudo service fail2ban stop && sudo service fail2ban start
```

Espero te resulte útil este artículo, compártelo

Ahora creamos el “**jail**”, por tanto debemos editar el archivo “**jail.local**” (asegúrate de insertar el **jail** correcto dependiendo si usas [apache](#) o [nginx](#)).

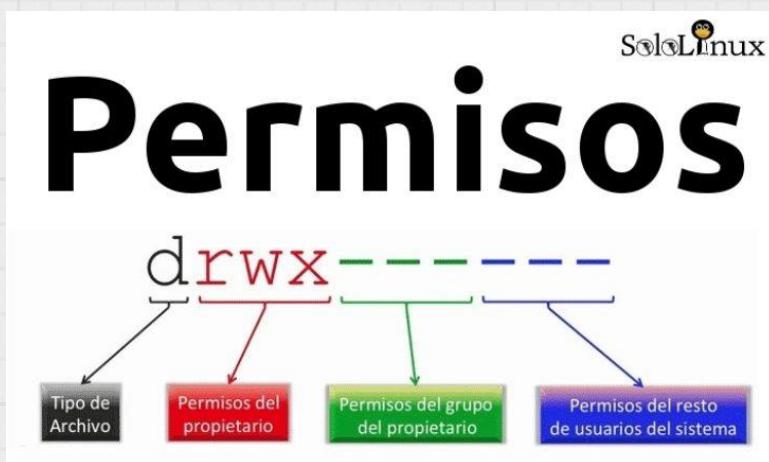
Editamos el archivo:

```
nano /etc/fail2ban/jail.local
```

Si es **Nginx**, copia y pega lo siguiente:

```
[webexploits]
enabled = true
port    = http,https
filter   = webexploits
logpath = %(nginx_access_log)s
maxretry = 3
```

Permisos de archivos en Linux



Permisos de archivos en Linux.

Este es el tema más importante respecto a la seguridad de archivos y directorios en los sistemas [Unix / Linux](#).

Conoceremos los permisos de archivos, así como su estructura.

Antes de continuar, debes saber que este artículo es la presentación de otros sucesivos en los cuales se profundizara mucho más en los formatos de permisos.

La serie constará de cuatro partes:

- [Permisos de archivos en Linux](#) (presentación).
- [Uso del comando chmod](#).
- [Uso de los comandos chown y chgrp](#).
- [Uso del comando chattr](#).

Comenzamos.

Permisos de archivos en Linux

Como buen comienzo deberías identificar los permisos de tus propios archivos, ejecuta lo siguiente:

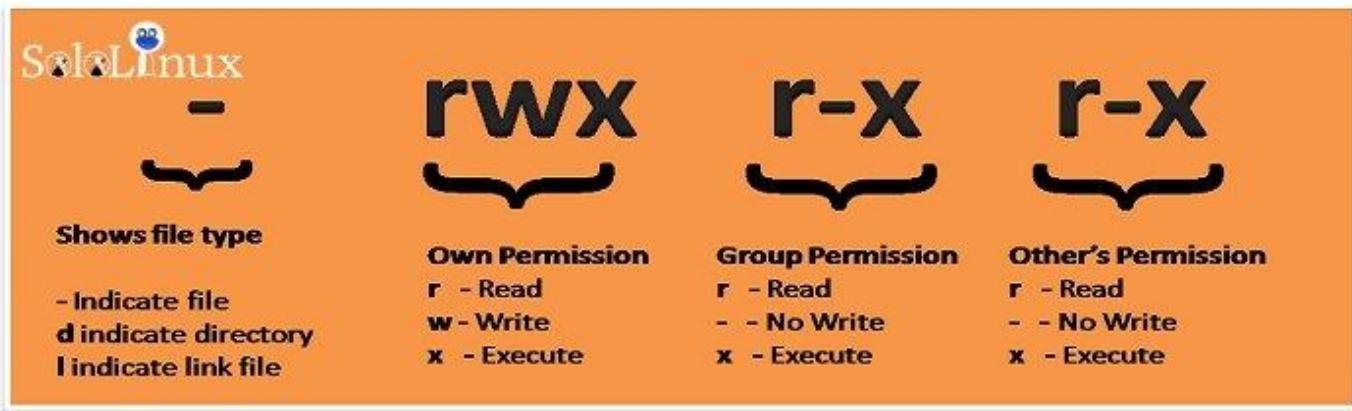
ls -l

Ejemplo de salida donde se imprimen los permisos de archivos y carpetas.

```
sergio@[REDACTED]:~> ls -l
total 32020
drwxr-xr-x  2 sergio users        4096 may 11  2018 bin
-rw-r--r--  1 root   root         89 jun 18  2018 check-bash.sh
-rw-----  1 sergio users       582 ene  7 21:31 [REDACTED]
drwxr-xr-x 24 sergio users      16384 feb 28 08:44 Descargas
drwxr-xr-x  2 root   root        4096 jun 10  2018 Desktop
drwxr-xr-x  8 sergio users       4096 jul  3  2018 Documentos
-rw-r--r--  1 root   root        829 sep  1 19:17 elfm.sh
drwxr-xr-x  6 sergio users       4096 feb 27 12:10 Escritorio
-rw-r--r--  1 sergio users          0 oct  2 15:05 hardwareinfo.html
drwxr-xr-x  3 sergio users       4096 dic 16 23:09 Imágenes
-rw-r--r--  1 root   root        421 oct  2 15:10 info-hardware.html
drwxr----  7 sergio users       4096 sep  9 22:12 Mail
drwxr-xr-x  1 root   root       1479 jul 19  2015 msfonts.sh
drwxr-xr-x  4 sergio users       4096 feb 27 16:19 Música
-rw-----  1 sergio users      550 oct  4 17:56 nano.save
```

Si observas la imagen superior, los permisos se generan con diez bits, y cada uno de ellos tiene un significado que indica al archivo o carpeta como debe comportarse.

En la siguiente imagen puedes ver como se definen los permisos de archivo.



Cada permiso se divide en cuatro bloques, el primero no siempre es necesario dado que tan solo indica lo que es, el segundo, tercero y cuarto, indican el propietario, grupo y otros que se asocian a el archivo.

- El primer bit nos indica el tipo de archivo.
- El bit 2, 3 y 4 nos dice el permiso del propietario del archivo.
- Los bits 5, 6 y 7 nos muestra el permiso para los miembros del grupo.
- El bit 8, 9 y 10 nos indica el permiso concedido a otros usuarios.

Roles de usuario y permisos

Para poder entender correctamente el permiso de un archivo o carpeta, debes conocer los tipos de roles y permisos.

Los [sistemas Linux](#) cuentan con tres [roles](#) disponibles: Usuario, Grupo y Otros.

Cada rol tiene tres tipos de permisos: Lectura, escritura y ejecución.

Roles:

Usuario (Propietario).

Grupo (Todos los miembros del grupo).

Otros (Resto de usuarios).

Permisos:

Lectura (**r**): Permite leer el contenido del archivo o los archivos de una carpeta.

Escritura (**w**): Permite escribir contenido en un archivo, crear uno nuevo, listar archivos, renombrar, y borrar un archivo de una carpeta.

Ejecución (**x**): Puede ejecutar cualquier archivo o [script](#), también permite el acceso a carpetas y archivos.

Formatos que modifican los permisos

Como norma general se utilizan dos tipos de notaciones, la simbólica, y la octal (las dos son validas).

Notación Simbólica: Hace uso del alfabeto (letras).

Notación Octal: Establece permisos mediante números.

Notación simbólica

Roles:

u – Usuario

g – Grupo

o – Otros

Permisos:

r – Permiso de lectura

w – Permiso de escritura

x – Permiso de ejecución

Notación octal

En la notación **octal**, cada permiso tiene el número predefinido.

Lectura (**r**) – 4

Escritura (**w**) – 2

Ejecución (**x**) – 1

Ejemplo de combinaciones:

7 – 4+2+1 (rwx) (Read + Write + Execute)

6 – 4+2 (rw-) (Read + Write)

5 – 4+1 (r-x) (Read + Execute)

4 – 4 (r-) (Read)

3 – 2+1 (-wx) (Write + Execute)

2 – 2 (-w-) (Write)

1 – 1 (-x) (Execute)

0 – 0 (—) (None)

Continuaremos con el articulo “Uso del comando CHMOD”

Uso del comando chmod

Uso del comando "CHMOD"



Uso del [comando chmod](#) (change mode).

En este artículo, aprenderás cómo variar los permisos de cualquier archivo o directorio con el **comando chmod**. El comando [chmod](#) de [Linux](#) se usa para modificar los permisos de archivos y directorios.

Ya hemos descrito a grandes rasgos los [permisos de archivos en Linux](#) en un [anterior articulo](#), que ademas recomiendo encarecidamente su lectura para poder asumir “el cómo” y “el porqué”, de los permisos en archivos y carpetas de Linux.

El uso de “**chmod**” es muy fácil, vemos como trabajar con este [comando](#).

```
[root@sololinux ~]# ls -lrt
total 499936
drwxr-x--- 3 root root      4096 may 15 2016 [REDACTED]
drwxr-xr-x  2 root root      4096 ene 16 2017 cleanup
drwxr-xr-x  3 root root      4096 jul 14 2017 [REDACTED]
-rw-r--r--  1 root root  1605546 jul 14 2017 [REDACTED]
-rw-r--r--  1 root root    77074 sep 20 2017 [REDACTED]
drwxr-xr-x  2 root root      4096 ene  9 2018 [REDACTED]
drwxr-xr-x  5 root root      4096 ene 11 2018 [REDACTED]
-rw-r--r--  1 root root   14240 ene 16 2018 remi-release-7.rpm
-rw xr-xr-x  1 root root 1861877 mar 17 2018 composer.phar
-rw-r--r--  1 root root   19704 mar 18 2018 master.zip
-rw-r--r--  1 root root 508307276 may  8 2018 [REDACTED]
-rw-r--r--  1 root root    5953 may 15 2018 block-country.sh
-rw-r--r--  1 root root     1984 jul  1 2018 libera.sh
```

Uso del comando chmod

La sintaxis de “**chmod**” es la siguiente:

Un ejemplo...

También puedes conceder permisos de manera [recursiva](#), lo conseguimos aplicando la indicación “-R”.

chmod [PERMISOS] [ARCHIVO]

chmod 755 sololinux.txt

chmod -R 755 /var/www/html

Uso de chmod con notación simbólica

Owner	Group	Other
r w x	r - x	r - x
4+2+1	4+0+1	4+0+1
7	5	5

Ahora recordamos los roles y permisos que vimos en el [artículo anterior](#).

Roles:

- u** – Usuario
- g** – Grupo
- o** – Otros

Permisos:

- r** – Permiso de lectura
- w** – Permiso de escritura
- x** – Permiso de ejecución

Al conocer estos permisos ya podemos comenzar a usar **chmod con símbolos** de manera correcta, las variables más comunes son:

- Lectura + Escritura: **rw**
- Lectura + Ejecución: **rx**
- Lectura + Escritura + Ejecución: **rwx**

Algunos ejemplos validos:

Usuario => Leer + Ejecutar	<code>chmod u+rx sololinux.txt</code>
Usuario + Grupo => Leer + Ejecutar	<code>chmod ug+rx sololinux.txt</code>
Usuario => Todo, Grupo => Leer + Ejecutar, Otro => Leer	<code>chmod u+rwx,g+rx,o+r sololinux.txt</code>
Usuario => Todo, Grupo + Otros => Leer + Ejecutar	<code>chmod u+rwx,go+rx sololinux.txt</code>
Todos los permisos para todos (no recomendado)	<code>chmod ugo+rwx sololinux.txt</code>

Uso de chmod con notación octal

Octal	Decimal	Permission	Representation
000	0 (0+0+0)	No Permission	---
001	1 (0+0+1)	Execute	--x
010	2 (0+2+0)	Write	-w-
011	3 (0+2+1)	Write + Execute	-wx
100	4 (4+0+0)	Read	r--
101	5 (4+0+1)	Read + Execute	r-x
110	6 (4+2+0)	Read + Write	rw-
111	7 (4+2+1)	Read + Write + Execute	rwx

Usando la **notación octal** podemos establecer permisos haciendo uso de números, entre el (0) y el (7). Debemos hacer la suma de lo siguiente...

- Lectura (r) – 4
- Escritura (w) – 2
- Ejecución (x) – 1

Por ejemplo:

El permiso 6, significa 4 + 2 (lectura + escritura).

El permiso 5 significa 4 + 1 (lectura + ejecución).

Importante: Los permisos deben establecerse en esta secuencia: **Usuarios, Grupos, Otros.**

Por ejemplo, si ponemos el permiso 754, su significado es: **usuario: 7, grupo: 5 y otro: 4.**

Algunos ejemplos válidos:

Usuario => leer + escribir + ejecutar, Grupo => leer + ejecutar, Otro => leer

chmod 754 sololinux.txt

Si desgranamos el “**permiso 754**”, nos daremos cuenta de los permisos que se conceden en realidad y a quien.

- 7 – es para el usuario, se combina lectura-4 + escritura-2 + ejecución-1.
- 5 – es para que el grupo, se combina lectura-4 + ejecución-1.
- 4 – es para otros, solo lectura-1.

Otro ejemplo para concluir con chmod...

Usuario => leer + escribir, Grupo => leer + escribir, Otro => leer

chmod 664 sololinux.txt

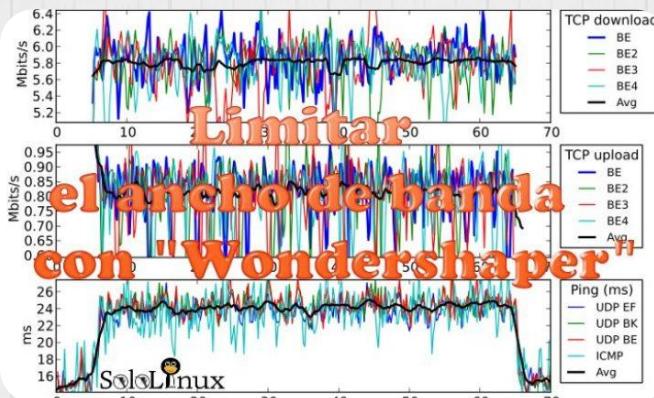
Continuamos en un próximo artículo (Uso de los comandos chown y chgrp).

La serie tiene cuatro partes:

- [Permisos de archivos en Linux](#) (presentación).
- [Uso del comando chmod.](#)
- [Uso de los comandos chown y chgrp.](#)
- [Uso del comando chattr.](#)

Si te resultó útil el artículo compártelo.





Limitar el ancho de banda con [Wondershaper](#).
En el articulo de hoy vemos como limitar el ancho de banda con **Wondershaper**, en [sistemas operativos Linux](#).

Al limitar el uso del ancho de banda de la red, ahorraremos un consumo innecesario de ciertas aplicaciones, como pueden ser los administradores de paquetes, navegadores web, clientes de [torrent](#), administradores de descargas, etc...

Si estas en una red interna es importante limitar el ancho de banda, evitaremos un posible abuso por parte de los usuarios de la red.

El uso de **Wondershaper** es una de las formas más fáciles y rápidas a la hora de limitar tanto la velocidad de descarga, como la velocidad de subida de datos a [internet](#).
Vemos como instalar **Wondershaper** en cualquier [distribución linux](#).



Limitar el ancho de banda con [Wondershaper](#)

Instalamos **Wondershaper** en nuestra distribución preferida.

En Debian, Ubuntu, Linux Mint y derivados:

```
sudo apt-get install wondershaper
```

En Rhel, CentOS y derivados (requiere el [repositorio EpeL](#)):

```
sudo yum install epel-release
sudo yum install wondershaper
```

Fedora y derivados:

```
sudo dnf install wondershaper
```

OpenSuse, Suse y derivados:

```
zypper install wondershaper
```

Arch Linux y derivados:

```
yay -S wondershaper-git
```

En cualquier Linux desde “Github“:

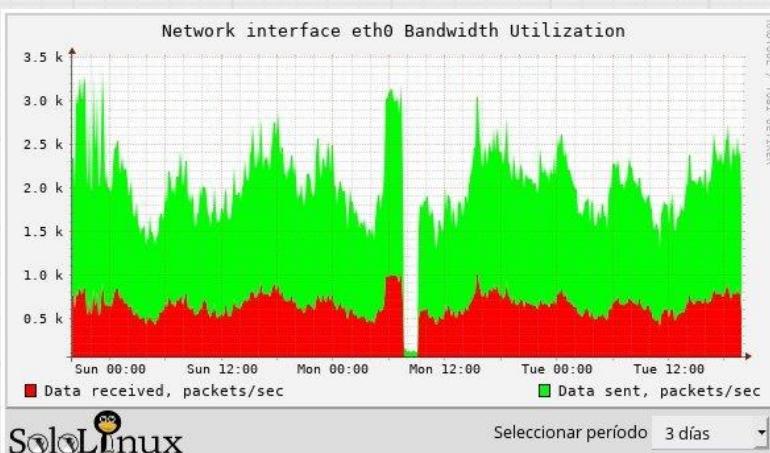
Clonamos el repositorio.

`git clone https://github.com/magnific0/wondershaper.git`

Abrimos el directorio e instalamos Wondershaper.

`cd wondershaper
sudo make install`

Ya lo tenemos instalado, le indicamos que inicie con el sistema (valido para cualquier tipo de instalación, independientemente de la que hayas seleccionado).

`sudo systemctl enable wondershaper.service
sudo systemctl start wondershaper.service`**Como usar Wondershare**Lo primero que debes saber es el nombre de la interfaz de red, lo podemos averiguar con cualquiera de estos dos comandos (modo de ejemplo usaremos “**eth0**”):`ip addr
ifconfig`

Una vez tenemos claro nuestro adaptador vemos como limitar la red, y comenzamos con las indicaciones de la herramienta, que son:

- **-a** : Nombre de la tarjeta de red
- **-d** :Velocidad de descarga
- **-u** :Velocidad de subida

El comando de uso sería:

`sudo wondershaper -a <interfazred> -d <veldedescarga> -u <veldesubida>`Vemos un ejemplo de uso con una interfaz de red “**eth0**”, una descarga de “**1024 Kbps**” y una subida de “**512 kbps**”.`sudo wondershaper -a eth0 -d 1024 -u 512`

Para anular los límites ejecuta lo siguiente:

```
sudo wondershaper -c -a eth0
# Las dos opciones son validas.
sudo wondershaper -c eth0
```

Si haces uso de más de una tarjeta de red en el sistema, debes configurar manualmente las tasas de carga y descarga en cada tarjeta de red como explicamos en este artículo que ahora concluye.

Si crees que este articulo es útil, compártelo.



Blog:
www.sololinux.es



Revista de distribución gratuita,
comparte conocimientos.

