

APACHE + PHP:

1. Instalamos apache:

```
sudo apt install apache2
```

2. Ajustamos el cortafuegos:

```
sudo ufw app list
```

3. Deberá salirnos:

```
Available applications:  
Apache  
Apache Full  
Apache Secure  
OpenSSH
```

4. Permitimos el tráfico HTTP y HTTPS:

```
sudo ufw allow in "Apache Full"
```

5. Instalamos Mysql:

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

6. Ejecutamos archivo de comandos de seguridad:

```
sudo mysql_secure_installation
```

```
Archivo Editar Ver Buscar Terminal Ayuda  
usuario@usuario-ALDA: ~  
By default, a MySQL installation has an anonymous user,  
allowing anyone to log into MySQL without having to have  
a user account created for them. This is intended only for  
testing, and to make the installation go a bit smoother.  
You should remove them before moving into a production  
environment.  
Remove anonymous users? (Press y|Y for Yes, any other key for No) : y  
Success.  
  
Normally, root should only be allowed to connect from  
'localhost'. This ensures that someone cannot guess at  
the root password from the network.  
Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y  
Success.  
  
By default, MySQL comes with a database named 'test' that  
anyone can access. This is also intended only for testing,  
and should be removed before moving into a production  
environment.  
  
Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y  
- Dropping test database...  
Success.  
  
- Removing privileges on test database...  
Success.  
  
Reloading the privilege tables will ensure that all changes  
made so far will take effect immediately.  
Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y  
Success.  
  
All done!  
usuario@usuario-ALDA:~$
```

7. Instalamos php:

```
sudo apt install php libapache2-mod-php php-mysql
```

```
usuario@usuario-ALDA:~$ sudo apt install php libapache2-mod-php php-mysql
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libapache2-mod-php7.2 php-common php7.2 php7.2-cli php7.2-common php7.2-json php7.2-mysql php7.2-opcache php7.2-readline
Paquetes sugeridos:
  php-pear
Se instalarán los siguientes paquetes NUEVOS:
  libapache2-mod-php libapache2-mod-php7.2 php php-common php-mysql php7.2 php7.2-cli php7.2-common php7.2-json php7.2-mysql php7.2-opcache
  php7.2-readline
0 actualizados, 12 nuevos se instalarán, 0 para eliminar y 4 no actualizados.
Se necesita descargar 3.985 kB de archivos.
Se utilizarán 17,6 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
Des:1 http://es.archive.ubuntu.com/ubuntu bionic/main amd64 php-common all 1:60ubuntu1 [12,1 kB]
Des:2 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 php7.2-common amd64 7.2.15-0ubuntu0.18.04.1 [882 kB]
Des:3 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 php7.2-json amd64 7.2.15-0ubuntu0.18.04.1 [18,8 kB]
Des:4 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 php7.2-opcache amd64 7.2.15-0ubuntu0.18.04.1 [165 kB]
Des:5 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 php7.2-readline amd64 7.2.15-0ubuntu0.18.04.1 [12,1 kB]
Des:6 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 php7.2-cli amd64 7.2.15-0ubuntu0.18.04.1 [1.409 kB]
Des:7 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libapache2-mod-php7.2 amd64 7.2.15-0ubuntu0.18.04.1 [1.351 kB]
Des:8 http://es.archive.ubuntu.com/ubuntu bionic/main amd64 libapache2-mod-php all 1:7.2+60ubuntu1 [3.212 kB]
```

8. Hacemos que apache busque index.php en primer lugar:

```
sudo nano /etc/apache2/mods-enabled/dir.conf
```

9. Ponemos index.php en primero de la lista y reseteamos el sistema:

```
sudo systemctl restart apache2
```

10. Comprobamos el estado:

```
sudo systemctl status apache2
```

```
usuario@usuario-ALDA:~
usuario@usuario-ALDA:~$ sudo systemctl status apache2
[sudo] password for usuario:
● apache2.service - LSB: Apache2 web server
  Loaded: loaded (/etc/init.d/apache2; bad; vendor preset: enabled)
  Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
    Active: active (running) since lun 2019-03-18 08:20:29 CET; 1 day 2h ago
      Docs: man:systemd-sysv-generator(8)
   Process: 16361 ExecReload=/etc/init.d/apache2 reload (code=exited, status=0/SU
     Tasks: 6
    Memory: 21.2M
       CPU: 2.668s
      CGroup: /system.slice/apache2.service
              ├─3025 /usr/sbin/apache2 -k start
              ├─16379 /usr/sbin/apache2 -k start
              ├─16380 /usr/sbin/apache2 -k start
              ├─16381 /usr/sbin/apache2 -k start
              ├─16382 /usr/sbin/apache2 -k start
              └─16384 /usr/sbin/apache2 -k start

mar 18 08:20:29 usuario-ALDA apache2[2527]: *
mar 18 08:20:29 usuario-ALDA systemd[1]: Started LSB: Apache2 web server.
mar 18 08:24:26 usuario-ALDA systemd[1]: Reloading LSB: Apache2 web server.
mar 18 08:24:26 usuario-ALDA apache2[6016]: * Reloading Apache httpd web server
mar 18 08:24:26 usuario-ALDA apache2[6016]: *
lines 1-23
```

11. Comprobamos que nuestro equipo usa de manera correcta php, creamos el archivo info.php:

```
sudo nano /var/www/html/info.php
```

12. Colocamos dentro lo siguiente:

```
<?php  
phpinfo();  
?>
```

13. Ahora abrimos el navegador y poniendo "localhost" deberia salir lo siguiente:

The screenshot shows a web browser window with the URL `192.168.115.0/info.php`. The page title is "PHP Version 7.2.15-0ubuntu0.18.04.1". The content is a table of PHP configuration settings. Key entries include:

System	
Build Date	Feb 8 2019 14:54:22
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-mysqlind.ini, /etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/20-sqlite3.ini, /etc/php/7.2/apache2/conf.d/20-sybase.ini, /etc/php/7.2/apache2/conf.d/20-sybase_ct.ini, /etc/php/7.2/apache2/conf.d/20-xml.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysmsg.ini, /etc/php/7.2/apache2/conf.d/20-syssem.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-sysvsem.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API20170718-NTS
PHP Extension Build	API20170718-NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tsv1.0, tsv1.1, tsv1.2
Registered Stream Filters	zlib*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert*, consumed, dechunk, convert.iconv*

A "zend engine" logo is visible at the bottom right of the page.

ISC-DHCP-SERVER

Es un servicio de red que permite que las computadoras tengan asignadas configuraciones automáticamente desde un servidor en lugar de configurar manualmente cada host de red. Las computadoras configuradas para ser clientes DHCP no tienen control sobre las configuraciones que reciben del servidor DHCP. Para instalarlo en ubuntu, ponemos en la terminal:

```
$ sudo apt-get install isc-dhcp-server
```

A partir de aqui, ejecutar todos los comandos como **ROOT**.

La instalación del paquete dhcp crea un archivo: **/etc/dhcp/dhcpd.conf** que es simplemente un archivo de configuración vacío. DHCP también usa el archivo:

/var/lib/dhcpd/dhcpd.leases para almacenar la base de datos de los usuarios.

* Archivos a editar:

- 1.- nano /etc/default/dhcpd.conf
- 2.- nano /etc/dhcp/dhcpd.conf
- 3.- nano /etc/default/hostapd
- 4.- nano /etc/default/isc-dhcp-server
- 5.- nano /etc/dhcp/dhcpd.confifconfig -a

```
root@usuario-H81M-D2V: /etc/dhcp
feb 18 10:57:29 usuario-H81M-D2V sh[12189]: process and the information we find helpful for debugging..
feb 18 10:57:29 usuario-H81M-D2V sh[12189]: exiting.
feb 18 10:57:29 usuario-H81M-D2V systemd[1]: tsc-dhcp-server.service: Main process exited, code=exited, status=0
feb 18 10:57:29 usuario-H81M-D2V systemd[1]: tsc-dhcp-server.service: Unit entered failed state.
feb 18 10:57:29 usuario-H81M-D2V systemd[1]: tsc-dhcp-server.service: Failed with result 'exit-code'.
root@usuario-H81M-D2V:/etc/dhcp# /etc/default/dhcpd+
bash: /etc/default/dhcpd+: No existe el archivo o el directorio
root@usuario-H81M-D2V:/etc/dhcp# nano /etc/default/dhcpd
root@usuario-H81M-D2V:/etc/dhcp# nano /etc/default/dhcpd.conf
root@usuario-H81M-D2V:/etc/dhcp# nano /etc/dhcp/dhcpd.conf
root@usuario-H81M-D2V:/etc/dhcp# nano /etc/dhcpd/dhcpd.conf
root@usuario-H81M-D2V:/etc/dhcp# nano /etc/dhcp/dhcpd.conf
root@usuario-H81M-D2V:/etc/dhcp# nano /etc/default/hio
root@usuario-H81M-D2V:/etc/dhcp# nano /etc/default/hostapd
```

```
root@usuario-H81M-D2V:/etc/dhcp# nano /etc/default/hostapd
root@usuario-H81M-D2V:/etc/dhcp# cd /etc/default
root@usuario-H81M-D2V:/etc/default# ls
acpid          aufs          console-setup  google-chrome   kerneloops    nss      speech-dispa
acpi-support    avahi-daemon  crda         grub           keyboard     proftpd   ssh
alsa           bridge-utils  cron         halt          libvirt-bin  qemu-kvm  ufw
amd64-microcode brltty       dbus         im-config    libvirt-guests rcs      useradd
anacron        bsdmainutils  devpts       intel-microcode locale      rsync     virtlockd
apache-htcacheclean  cacerds   docker       irqbalance   motion      rsyslog   virtlogd
apport         chrome-remote-desktop ebttables  isc-dhcp-server networking saned    virtualbox
root@usuario-H81M-D2V:/etc/default# nano hostapd
root@usuario-H81M-D2V:/etc/default# ls
acpid          aufs          console-setup  google-chrome   kerneloops    nss      speech-dispatch
acpi-support    avahi-daemon  crda         grub           keyboard     proftpd  ssh
alsa           bridge-utils  cron         halt          libvirt-bin  qemu-kvm ufw
amd64-microcode brltty       dbus         im-config    libvirt-guests rcs      useradd
anacron        bsdmainutils  devpts       intel-microcode locale      rsync     virtlockd
apache-htcacheclean  cacerds   docker       irqbalance   motion      rsyslog   virtlogd
apport         chrome-remote-desktop ebttables  isc-dhcp-server networking saned    virtualbox
root@usuario-H81M-D2V:/etc/default# nano isc-dhcp-server
root@usuario-H81M-D2V:/etc/default# cd /etc/dhcp/
root@usuario-H81M-D2V:/etc/dhcp# nano dhcpd.conf
root@usuario-H81M-D2V:/etc/dhcp# echo "">dhcpd.conf
root@usuario-H81M-D2V:/etc/dhcp# nano dhcpd.conf
root@usuario-H81M-D2V:/etc/dhcp# ifconfig -a
br0      Link encap:Ethernet direcciónHW 74:d4:35:2e:ad:00
        Direc. inet:192.168.14.122  Difus.:192.168.15.255 Másc:255.255.248.0
        Dirección inet6: fe80::76d4:35ff:fe2e:ad00/64 Alcance:Enlace
        ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
        Paquetes RX:147580 errores:0 perdidos:116 overruns:0 frame:0
```

* Poner en la terminal:

- 1.- ifconfig enp2s0 192.168.37.4
- 2.- /etc/init.d/isc-dhcp-server start
- 3.- /etc/init.d/isc-dhcp-server status

Si en el estado pone **active (running)** está funcionando.

```
root@usuario-H81M-D2V:/etc/dhcp# ifconfig enp2s0 192.168.37.4
[ ok ] Starting isc-dhcp-server (via systemctl): isc-dhcp-server.service,
root@usuario-H81M-D2V:/etc/dhcp# ps -A | grep dh
```

```
root@usuario-H81M-D2V:/etc/dhcp# /etc/init.d/isc-dhcp-server status
● isc-dhcp-server.service - ISC DHCP IPv4 server
  Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor preset: enabled)
  Active: active (running) since lun 2019-02-18 11:08:11 CET; 21s ago
    Docs: man:dhcpd(8)
   Main PID: 12346 (dhcpd)
     Tasks: 1
    Memory: 9.5M
       CPU: 41ms
      CGroup: /system.slice/isc-dhcp-server.service
              └─12346 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/dhcpd.pid -cf /etc/dhcp/dhcpd.conf enp2s0

feb 18 11:08:30 usuario-H81M-D2V dhcpcd[12346]: DHCPREQUEST for 192.168.37.17 (192.168.37.4) from fe:99:bf:5b:3e:4d (iPhonedeGerman) via enp2s0
feb 18 11:08:30 usuario-H81M-D2V dhcpcd[12346]: DHCPPACK on 192.168.37.17 to fe:99:bf:5b:3e:4d (iPhonedeGerman) via enp2s0
feb 18 11:08:31 usuario-H81M-D2V dhcpcd[12346]: DHCPDISCOVER from 38:ca:da:43:c1:87 via enp2s0
feb 18 11:08:31 usuario-H81M-D2V dhcpcd[12346]: DHCPREQUEST for 192.168.14.57 from 38:ca:da:43:c1:87 via enp2s0: ignored (not authoritative).
feb 18 11:08:31 usuario-H81M-D2V dhcpcd[12346]: DHCPDISCOVER from 38:ca:da:43:c1:87 via enp2s0: ignored (not authoritative).
feb 18 11:08:32 usuario-H81M-D2V dhcpcd[12346]: DHCPDISCOVER from 54:a0:50:7c:18:58 via enp2s0
feb 18 11:08:32 usuario-H81M-D2V dhcpcd[12346]: DHCPOFFER on 192.168.37.18 to 38:ca:da:43:c1:87 (iPhonedeNieves) via enp2s0
feb 18 11:08:32 usuario-H81M-D2V dhcpcd[12346]: DHCPOFFER on 192.168.37.19 to 54:a0:50:7c:18:58 (pc225-11) via enp2s0
feb 18 11:08:32 usuario-H81M-D2V dhcpcd[12346]: DHCPREQUEST for 192.168.37.19 (192.168.37.4) from 54:a0:50:7c:18:58 (pc225-11) via enp2s0
Hint: Some lines were ellipsized, use -l to show in full.
root@usuario-H81M-D2V:/etc/dhcp#
```

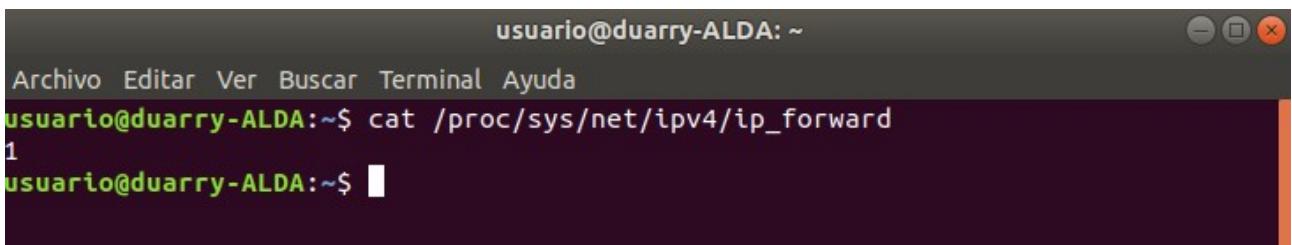
Enrutamiento:

1. Usando procfs:

Este es el método más usado, pero es temporal, y debemos realizarlo después de cada reinicio el servidor.

Si agregamos **sudo echo “1” > /proc/sys/net/ipv4/ip_forward** al archivo **nano /etc/rc.local** , para que se ejecute cada vez que encendamos. Así no tenemos que escribirlo cada vez.

Podemos verificar si el IP forward, esta habilitado o no, revisando el estado del archivo **cat /proc/sys/net/ipv4/ip_forward**. 1 = activado, 0 = desactivado.



```
usuario@duarry-ALDA: ~
Archivo Editar Ver Buscar Terminal Ayuda
usuario@duarry-ALDA:~$ cat /proc/sys/net/ipv4/ip_forward
1
usuario@duarry-ALDA:~$
```

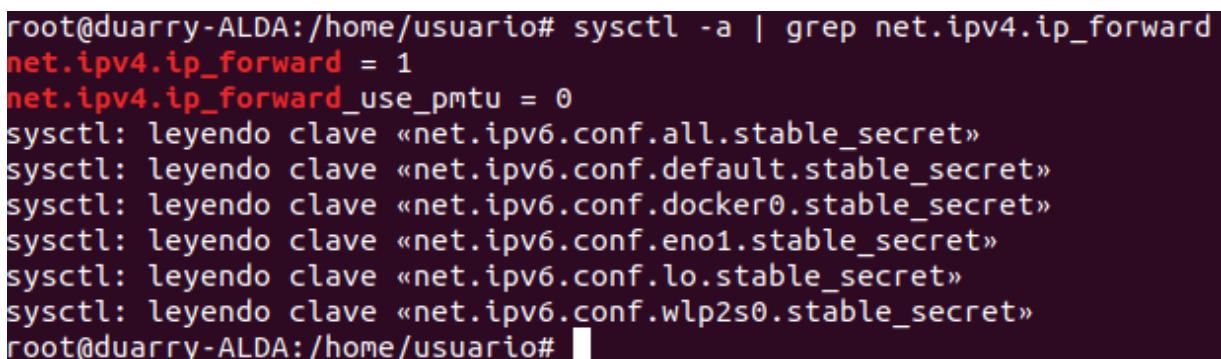
2. Usando sysctl

Sysctl, permite cambiar valores en el kernel en tiempo real, usando esta herramienta, se puede cambiar el comportamiento del kernel, y con ello habilitar el IP forward o no.

* Ejecutamos los comandos como root.

1. Primero vemos si esta activo o no.

```
$ sysctl -a | grep net.ipv4.ip_forward
```



```
root@duarry-ALDA:/home/usuario# sysctl -a | grep net.ipv4.ip_forward
net.ipv4.ip_forward = 1
net.ipv4.ip_forward_use_pmtu = 0
sysctl: leyendo clave «net.ipv6.conf.all.stable_secret»
sysctl: leyendo clave «net.ipv6.conf.default.stable_secret»
sysctl: leyendo clave «net.ipv6.conf.docker0.stable_secret»
sysctl: leyendo clave «net.ipv6.conf.eno1.stable_secret»
sysctl: leyendo clave «net.ipv6.conf.lo.stable_secret»
sysctl: leyendo clave «net.ipv6.conf.wlp2s0.stable_secret»
root@duarry-ALDA:/home/usuario#
```

2. Podemos colocar su valor en 1, para encender el ip forwarding con:

```
$ sysctl -w net.ipv4.ip_forward=1
```

* Si queremos que sea permanente, debemos editar el archivo **nano /etc/sysctl.conf** y modificar esta línea: **net.ipv4.ip_forward = 1**.

3. Para cargar los cambios recién realizados, sin apagar el servidor, ejecuta este comando

```
$ sysctl -p
```

4. Podríamos realizar un script que activara el enrutamiento y el NAT, y otro para desactivarlo:

```
Activar-enrutamiento.sh
echo "1" > /proc/sys/net/ipv4/ip_forward
iptables -A FORWARD -j ACCEPT
iptables -t nat -A POSTROUTING -s 10.0.0.0/8 -o eth0 -j MASQUERADE
```

5. Rutas fijas:

- a) Crear una ruta para una IP concreta → **sudo route add 80.58.12.27 eth1**
- b) Eliminar una ruta → **\$ sudo route del -net 193.144.238.0/24**
- c) Ver rutas → **\$ sudo route**

```
usuario@duarry-ALDA:~$ sudo route
Tabla de rutas IP del núcleo
Destino      Pasarela        Genmask        Indic Métric Ref    Uso Interfaz
default      f0.41700041.41. 0.0.0.0      UG      600      0          0 wlp2s0
link-local   0.0.0.0         255.255.0.0    U       1000     0          0 wlp2s0
172.17.0.0   0.0.0.0         255.255.0.0    U       0        0          0 docker0
192.168.8.0  0.0.0.0         255.255.248.0   U       600      0          0 wlp2s0
usuario@duarry-ALDA:~$
```

Escrítorio remoto:

La función Escritorio Remoto o también conocida como VNC (Virtual Network Computing) es una función que nos permite manejar cualquier equipo de manera remota, de tal manera que fuera el propio usuario o administrador el que lo manejasé.

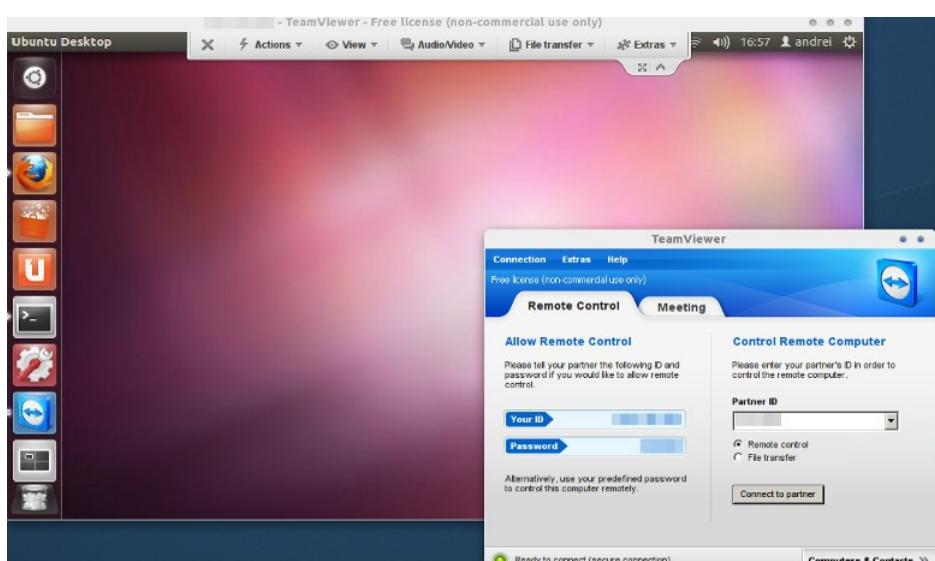
Una vez que estemos conectados, en nuestro equipo se abrirá una ventana en la que veremos el escritorio del sistema remoto y por cuya ventana podremos hacer funcionar nuestro ratón o nuestro teclado para introducir un código, contraseña o simplemente realizar una configuración exacta. Un funcionamiento similar al que ocurre con las máquinas virtuales de VirtualBox.

Las 5 herramientas más populares y potentes para tener esta función en nuestro sistema operativo:

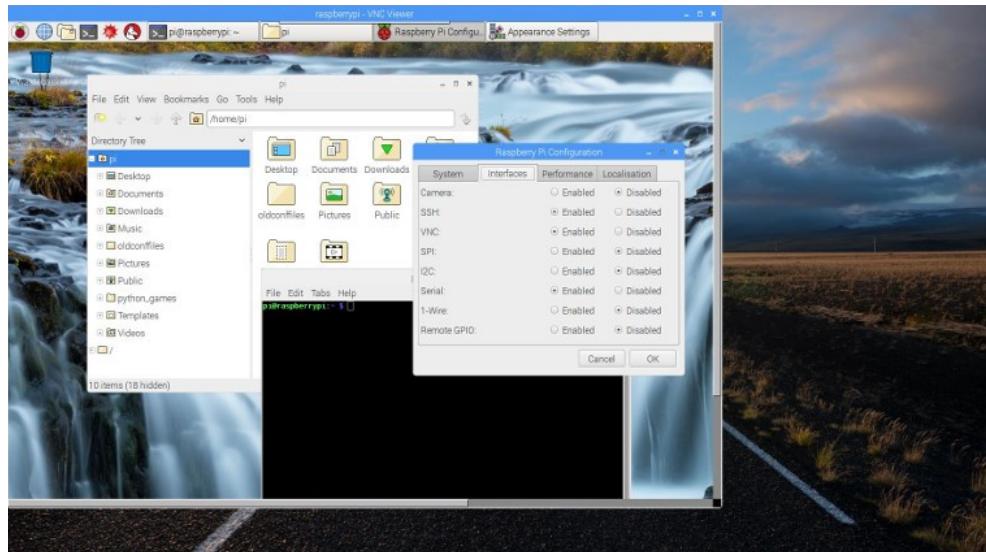
1. **Vinagre** → Para usar Vinagre sólo hemos de indicar la dirección IP del equipo a controlar y la dirección IP nuestra. En el caso del otro equipo hemos de añadir la dirección IP de nuestro ordenador y la suya. Para los usuarios que utilizan las librerías QT, la opción similar a Vinagre se llama KRDC.



2. **TeamViewer** → Instalamos la aplicación anfitrión y cliente en ambos ordenadores. El programa nos dará un código que se aplica a ese ordenador, luego, para conectarnos y controlar el equipo de manera remota sólo hemos de introducir la clave del otro ordenador y listo. No hace falta ni saber de redes ni de direcciones IP, además las configuraciones de otros programas no influyen en la utilización de TeamViewer.



3. **RealVNC** → es un programa con similar código a Vinagre. Ambos proyectos parten del mismo código, pero RealVNC tiene versiones para muchas más plataformas, además es un programa de Escritorio Remoto más completo que Vinagre pero también más difícil.. Hay que utilizar repositorios o paquetes externos.



4. **SSH (Realizado por mi)** → es un programa que utiliza el protocolo SSH y que permite conectarnos de manera remota a un sistema operativo. Aunque en este caso la conexión se realiza a través de la terminal. Algo que hace que no podamos usar el ratón y que además necesitemos menos recursos que en otras aplicaciones, pero también es un protocolo problemático que puede ocasionar problemas de seguridad. Para conectarnos al servidor SSH debemos escribir en nuestro terminal el siguiente comando: ssh usuario@[IP]:[puerto] Pulsamos enter y se establecerá la conexión. Al ser un servidor seguro debemos crear la clave de seguridad y almacenarla en nuestro ordenador para futuras conexiones al mismo servidor. El mismo cliente nos preguntará si queremos crearla y continuar en caso de que no exista dicha clave. A continuación debemos introducir la contraseña del usuario con el que vamos a iniciar sesión. Una vez introducida la contraseña el cliente SSH se conectará al servidor y nos cambiará tanto el usuario como el dominio de nuestro terminal. Para cerrar la conexión debemos teclear la palabra “exit” que finalizará la misma y evitar así que otros usuarios puedan tomar el control de nuestra conexión SSH remota.

```

angela@usuario-ALDA: ~
usuario@usuario-ALDA:~$ ssh angela@192.168.10.215
angela@192.168.10.215's password:
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.15.1 x86_64)

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

Pueden actualizarse 82 paquetes.
0 actualizaciones son de seguridad.

New release '18.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** Es necesario reiniciar el sistema ***

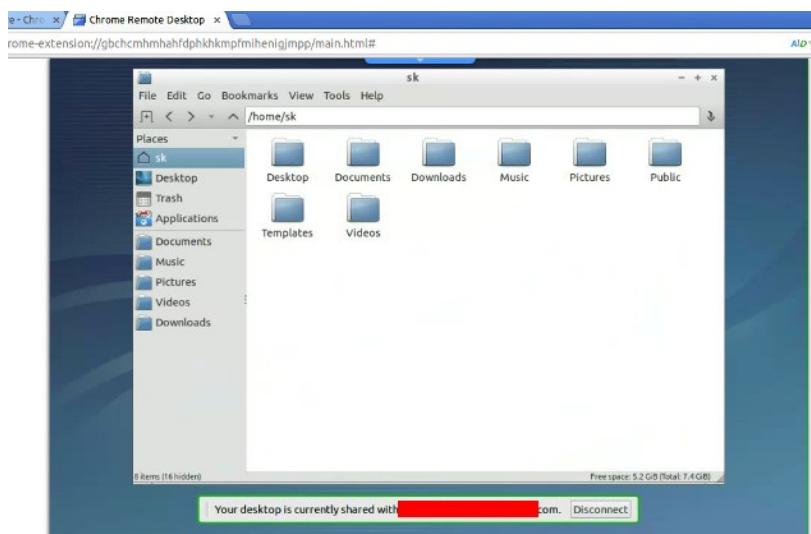
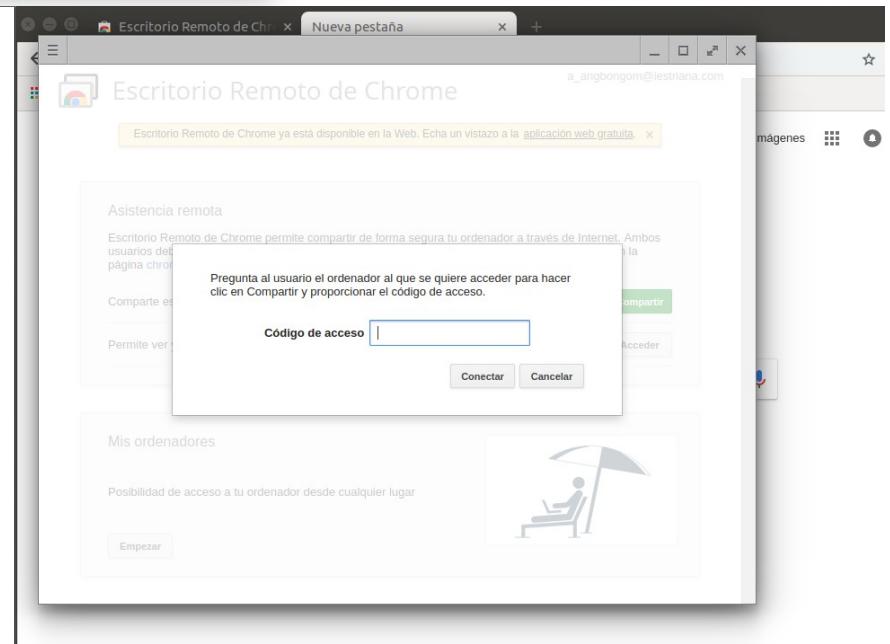
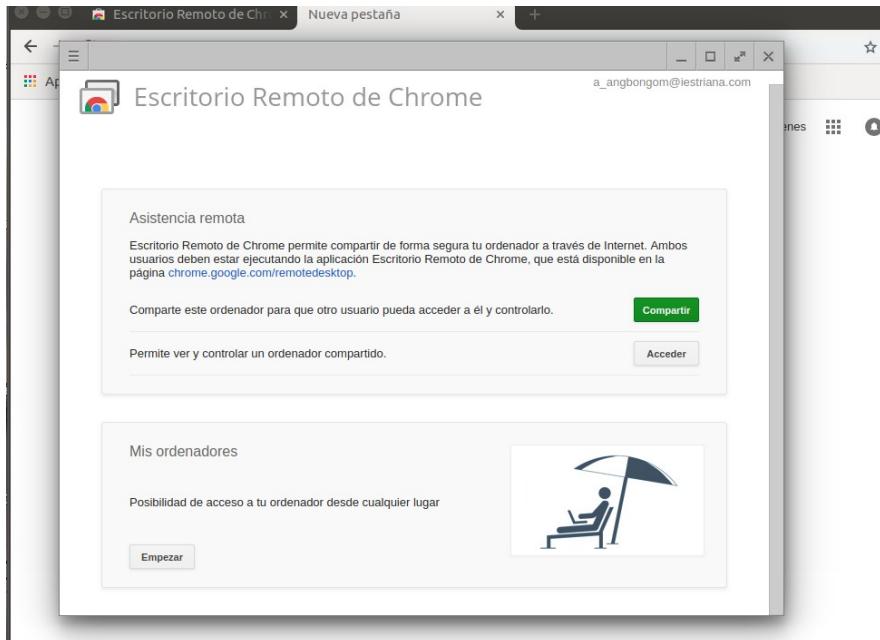
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

angela@usuario-ALDA:~$ 

```

5. **Chrome Remote Beta** → Esta solución de Google nos permite poder utilizar el Escritorio Remoto con un plugin del navegador. Chrome es un navegador muy usado entre los ordenadores, por ello casi podemos utilizar esta función en cualquier sistema operativo. Para su utilización e instalación sólo hemos de instalar el plugin en los navegadores Chrome de ambos sistemas operativos.



Servidor ftp:

1. \$ sudo apt-get install proftpd
2. Lo primero que debemos hacer es establecer una carpeta como directorio principal para todos los usuarios del FTP. Para ello, abriremos con nuestro editor de texto preferido el archivo de configuración anterior:
 - ✓ \$ sudo gedit /etc/proftpd/proftpd.conf
3. Una vez allí, buscaremos la línea comentada “DefaultRoot”, la descomentamos borrando la almohadilla # y escribimos /home/proftpd. Esto nos va a permitir que cuando cada usuario acceda a su cuenta del FTP, estos accederán directamente a su carpeta “home”.
 - ✓ DefaultRoot /home/proftpd

The screenshot shows a text editor window titled "proftpd.conf /etc/proftpd". The file contains configuration options for the ProFTPD server. Key settings include:

- DisplayLogin welcome.msg
- DisplayChdir .message true
- ListOptions "-l"
- DenyFilter *.*
- # Use this to jail all users in their homes
- DefaultRoot /home/proftpd
- # Users require a valid shell listed in /etc/shells to login.
- # Use this directive to release that constrain.
- # RequireValidShell off
- # Port 21 is the standard FTP port.
- Port 21
- # In some cases you have to specify passive ports range to by-pass
- # firewall limitations. Ephemeral ports can be used for that, but
- # feel free to use a more narrow range.
- # PassivePorts 49152 65534
- # If your host was NATted, this option is useful in order to
- # allow passive transfers to work. You have to use your public
- # address and opening the passive ports used on your firewall as well.
- # MasqueradeAddress 1.2.3.4
- # This is useful for masquerading address with dynamic IPs:
- # refresh any configured MasqueradeAddress directives every 8 hours
- <IfModule mod_dynamashq.c>
- # DynMasqRefresh 28800
- </IfModule>
- # To prevent DoS attacks, set the maximum number of child processes
- # to 30. If you need to allow more than 30 concurrent connections
- # at once, simply increase this value. Note that this ONLY works
- # in standalone mode, in inetd mode you should use an inetd server
- # that allows you to limit maximum number of processes per service

At the bottom of the editor, there are status indicators: "Texto plano ▾", "Anchura de la pestaña: 8 ▾", "Ln 1, Col 1 ▾", and "INS".

4. Otros parámetros interesantes para configurar son:
 - ✓ ServerName: nos permite establecer un nombre al servidor.
 - ✓ AccessGrantMsg: Mensaje de bienvenida. (Hay que añadirlo manualmente al final del archivo).
 - ✓ AccessDenyMsg: Mensaje de error al iniciar. (Hay que añadirlo manualmente al final del archivo).

5. A continuación reiniciaremos el servidor para que se apliquen los cambios realizados tecleando:
✓ \$ sudo /etc/init.d/proftpd restart

```
to Metadata::getall-position no esta soportado
usuario@usuario-ALDA:~$ sudo /etc/init.d/proftpd restart
[ ok ] Restarting proftpd (via systemctl): proftpd.service.
usuario@usuario-ALDA:~$ █
```

6. Para permitir la conexión a un usuario escribimos al final del archivo de configuración:
<Limit LOGIN>
AllowUser usuario
DenyAll
</Limit>

```
# Include other custom configuration files
Include /etc/proftpd/conf.d/
AccessGrantMsg ";BIENVENIDO!"
AccessDenyMsg ";ERROR!"

<Limit LOGIN>
AllowUser usuario
DenyAll
</Limit>
```

-
7. Otros comandos de utilidad:
✓ Podemos consultar los usuarios que están conectados a nuestro servidor FTP en tiempo real con los siguientes comandos:
✗ \$ ftpwho
✗ \$ ftptop

MODULOS HELLOWORLD

1.- Creamos un archivo con nano llamado hello.c

```
/***
 * @file hello.c
 * @author Akshat Sinha
 * @date 10 Sept 2016
 * @version 0.1
 * @brief An introductory "Hello World!" loadable kernel
 * module (LKM) that can display a message in the /var/log/kern.log
 * file when the module is loaded and removed. The module can accept
 * an argument when it is loaded -- the name, which appears in the
 * kernel log files.
*/
#include <linux/module.h> /* Needed by all modules */
#include <linux/kernel.h> /* Needed for KERN_INFO */
#include <linux/init.h> /* Needed for the macros */

///< The license type -- this affects runtime behavior
MODULE_LICENSE("GPL");

///< The author -- visible when you use modinfo
MODULE_AUTHOR("Angela Bonilla");

///< The description -- see modinfo
MODULE_DESCRIPTION("Modulo Hola de Angela");

///< The version of the module
MODULE_VERSION("0.1");

static int __init hello_start(void)
{
    printk(KERN_INFO "Cargando modulo Angela...\n");
    printk(KERN_INFO "Hola soy Angela\n");
    return 0;
}

static void __exit hello_end(void)
{
    printk(KERN_INFO "Bye\n");
}

module_init(hello_start);
module_exit(hello_end);
```

2.- Creamos otro archivo llamado Makefile.

```
obj-m=hola.o
all:
    make -C /lib/modules/$(shell uname -r)/build/ M=$(PWD) modules
clean:
```

```
make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

3.- En **obj-m** ponemos como queremos que se llame nuestro módulo, acabado en **.o**.

4.- A continuación, realizamos un **make**.

5.- Aparecen estas carpetas si hacemos **ls**.

```
hello.c  hola.c  hola.ko  hola.mod.c  hola.mod.o  hola.o  Makefile  modules.order  Module.symvers
```

6.- Cargamos el módulo en memoria con el comando **insmod hola.ko**.

7.- Podemos ver los módulos cargados con el comando **lsmod**.

```
root@usuario-H81M-D2V:/home/usuario/Escritorio/carpeta_gonzalo/helloworld# insmod hola.ko
root@usuario-H81M-D2V:/home/usuario/Escritorio/carpeta_gonzalo/helloworld# lsmod
Module            Size  Used by
hola             16384  0
ebtable_filter   16384  0
ebtables          32768  1 ebtable_filter
ip6table_filter  16384  0
ip6_tables        28672  1 ip6table_filter
ipt_MASQUERADE  16384  1
nf_nat_masquerade_ipv4 16384  1 ipt_MASQUERADE
nf_conntrack_netlink 40960  0
nfnetlink         16384  2 nf_conntrack_netlink
xfrm_user         32768  1
xfrm_algo         16384  1 xfrm_user
iptable_nat       16384  1
nf_conntrack_ipv4 16384  3
nf_defrag_ipv4   16384  1 nf_conntrack_ipv4
nf_nat_ipv4      16384  1 iptable_nat
xt_addrtype      16384  2
iptable_filter   16384  1
ip_tables         28672  2 iptable_filter,iptable_nat
xt_conntrack     16384  1
x_tables          40960  8 ebttables,ip6table_filter,xt_conntrack,iptable_filter,ipt_MASQUERADE,xt_addrtype,
ip6_tables,ip_tables
nf_nat            32768  2 nf_nat_masquerade_ipv4,nf_nat_ipv4
nf_conntrack     131072  7 xt_conntrack,nf_nat_masquerade_ipv4,nf_conntrack_ipv4,nf_nat,ipt_MASQUERADE,nf_n
at_ip4v,nf_conntrack_netlink
libcrc32c         16384  2 nf_conntrack,nf_nat
br_netfilter      24576  0
aufs              237568  0
pci_stub          16384  1
vboxpci           24576  0
```

8.- Además de ver el módulo en nuestra carpeta, aparece en la carpeta **/var/log**. Entramos con el comando **cd /var/log**.

9.- Podemos ver el mensaje que sale a la entrada del módulo, poniendo el comando **cat kern.log**.

```
root@usuario-H81M-D2V:/var/log#
root@usuario-H81M-D2V:/var/log# ls
alternatives.log      apport.log.6.gz    btmp          fsck          proftpd        wtmp
alternatives.log.1    apport.log.7.gz    btmp.1        gpu-manager.log samba          wtmp.1
alternatives.log.2.gz  apt              cups          hp            speech-dispatcher Xorg.0.log
alternatives.log.3.gz  aptitude         dist-upgrade installer    syslog          Xorg.0.log.old
alternatives.log.4.gz  aptitude.1.gz   dmesg         kern.log      syslog.1      Xorg.1.log
alternatives.log.5.gz  aptitude.2.gz   dpkg.log     kern.log.1    syslog.2.gz  Xorg.1.log.old
apache2               auth.log        dpkg.log.1   kern.log.2.gz  syslog.3.gz  Xorg.2.log
apport.log            auth.log.1     dpkg.log.2.gz kern.log.3.gz  syslog.4.gz  Xorg.2.log.old
apport.log.1          auth.log.2.gz   dpkg.log.3.gz kern.log.4.gz  syslog.5.gz
apport.log.2.gz        auth.log.3.gz   dpkg.log.4.gz lastlog       syslog.6.gz
apport.log.3.gz        auth.log.4.gz   dpkg.log.5.gz libvirt        syslog.7.gz
apport.log.4.gz        boot.log       faillog      lightdm      unattended-upgrades
apport.log.5.gz        bootstrap.log fontconfig.log mysql        upstart
```

Instalación de MySQL

Para la instalación de MySQL usaremos la terminal, siendo usuario normal, ya que usaremos los comandos con el principio sudo:

A) Tenemos que instalar los paquetes mysql-server y mysql-client:

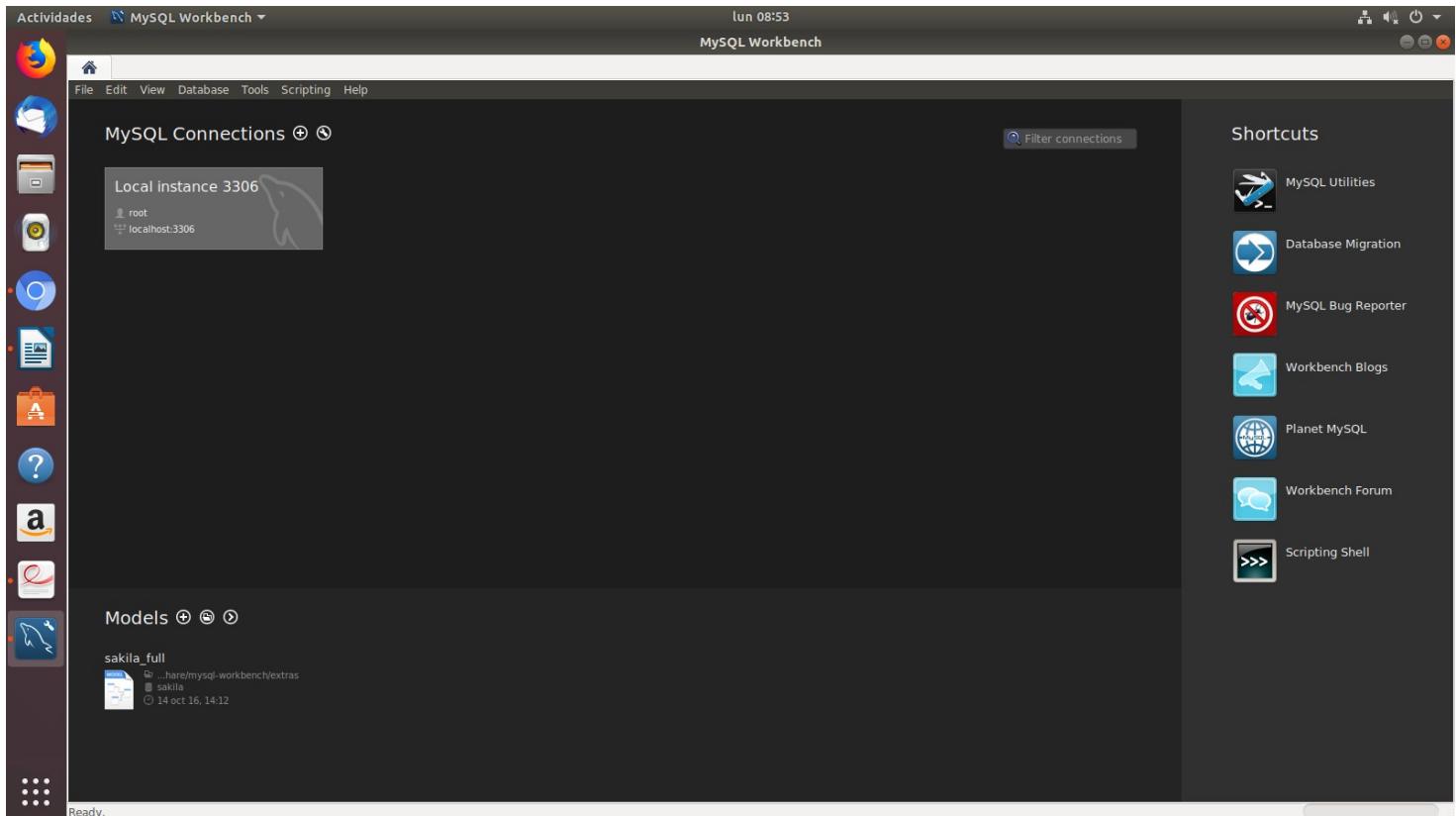
- ✓ \$ sudo apt-get install mysql-server
- ✓ \$ sudo apt-get install mysql-client

B) Después de instalar los paquetes anteriores, le cambiamos la contraseña a mysql de la siguiente manera:

- ✓ \$ sudo /usr/bin/mysqladmin -u root password manager → Este comando lo que indica es el usuario y password que administrará la base.

C) Para finalizar nos conectamos como root de la siguiente forma:

- ✓ \$ mysql -u root -p → Al poner este comando, nos pedirá la contraseña y se nos abrirá.



¿Qué es iptables?

IPTables es un sistema de firewall vinculado al kernel de linux. Un firewall de iptables no es como un servidor que lo iniciamos o detenemos o que se pueda caer por un error de programación; iptables esta integrado con el kernel, es parte del sistema operativo.

¿Cómo se pone en marcha?

Realmente lo que se hace es aplicar reglas. Para ellos se ejecuta el comando iptables, con el que añadimos, borramos, o creamos reglas. Por ello un firewall de iptables no es sino un simple script de shell en el que se van ejecutando las reglas de firewall.

Para los paquetes que van a la propia maquina se aplican las reglas INPUT y OUTPUT, y para filtrar paquetes que van a otras redes o maquinas se aplican simplemente reglas FORWARD.

Pero antes de aplicar esas reglas es posible aplicar reglas de NAT: estas se usan para hacer redirecciones de puertos o cambios en las IPs de origen y destino.

Incluso antes de las reglas de NAT se pueden meter reglas de tipo MANGLE, para modificar los paquetes; son reglas poco conocidas y es probable que no se usen.

Por tanto tenemos tres tipos de reglas en iptables:

- MANGLE
- NAT: reglas PREROUTING, POSTROUTING
- FILTER: reglas INPUT, OUTPUT, FORWARD

Denegar Ping con Iptables:

1.- sudo -i

2.- iptables -A OUTPUT --proto icmp -j DROP → Regla que impide que mi maquina pueda hacer ping fuera.

4.- Realizar ping a la 192.168.8.1 para ver si hace ping o no.

The screenshot shows a terminal window with two panes. The left pane shows a user session where they attempt to ping 192.168.8.1 but receive 'Operation not permitted' errors. The right pane shows a root session where the user runs 'sudo -i' and then applies the iptables rule to drop ICMP traffic. Finally, they ping 192.168.8.1 from the root session and get a successful response.

```
usuario@usuario-H81M-D2V: ~
collisions:0 long.colatTX:1000
Bytes RX:22351 (22.3 KB) TX bytes:22351 (22.3 KB)

usuario@usuario-H81M-D2V:~$ ping 192.168.8.
ping: unknown host 192.168.8.
usuario@usuario-H81M-D2V:~$ ping 192.168.8.1
PING 192.168.8.1 (192.168.8.1) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
^C
--- 192.168.8.1 ping statistics ---
12 packets transmitted, 0 received, 100% packet loss, time 11252ms

usuario@usuario-H81M-D2V:~$ 
root@usuario-H81M-D2V:~# ping 192.168.8.1
PING 192.168.8.1 (192.168.8.1) 56(84) bytes of data.
64 bytes from 192.168.8.1: icmp_seq=1 ttl=64 time=0.446 ms
64 bytes from 192.168.8.1: icmp_seq=2 ttl=64 time=0.481 ms
64 bytes from 192.168.8.1: icmp_seq=3 ttl=64 time=0.484 ms
64 bytes from 192.168.8.1: icmp_seq=4 ttl=64 time=0.340 ms
64 bytes from 192.168.8.1: icmp_seq=5 ttl=64 time=0.531 ms
64 bytes from 192.168.8.1: icmp_seq=6 ttl=64 time=0.390 ms
64 bytes from 192.168.8.1: icmp_seq=7 ttl=64 time=0.332 ms
^C
--- 192.168.8.1 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6136ms
rtt min/avg/max/mdev = 0.332/0.429/0.531/0.071 ms
root@usuario-H81M-D2V:~# sudo -i
root@usuario-H81M-D2V:~# iptables -A INPUT --proto icmp -j DROP
root@usuario-H81M-D2V:~# iptables -A OUTPUT --proto icmp -j DROP
root@usuario-H81M-D2V:~# 
```

LOGS:

El sistema de logs de Linux (log = registro), es un mecanismo estándar que se encarga de recoger los mensajes generados por los programas, aplicaciones y demonios y enviarlos a un destino predefinido. En cada mensaje consta la fuente (el programa que generó el mensaje), la prioridad (nivel de importancia del mensaje), la fecha y la hora.

1. Cómo funciona el sistema de logs:

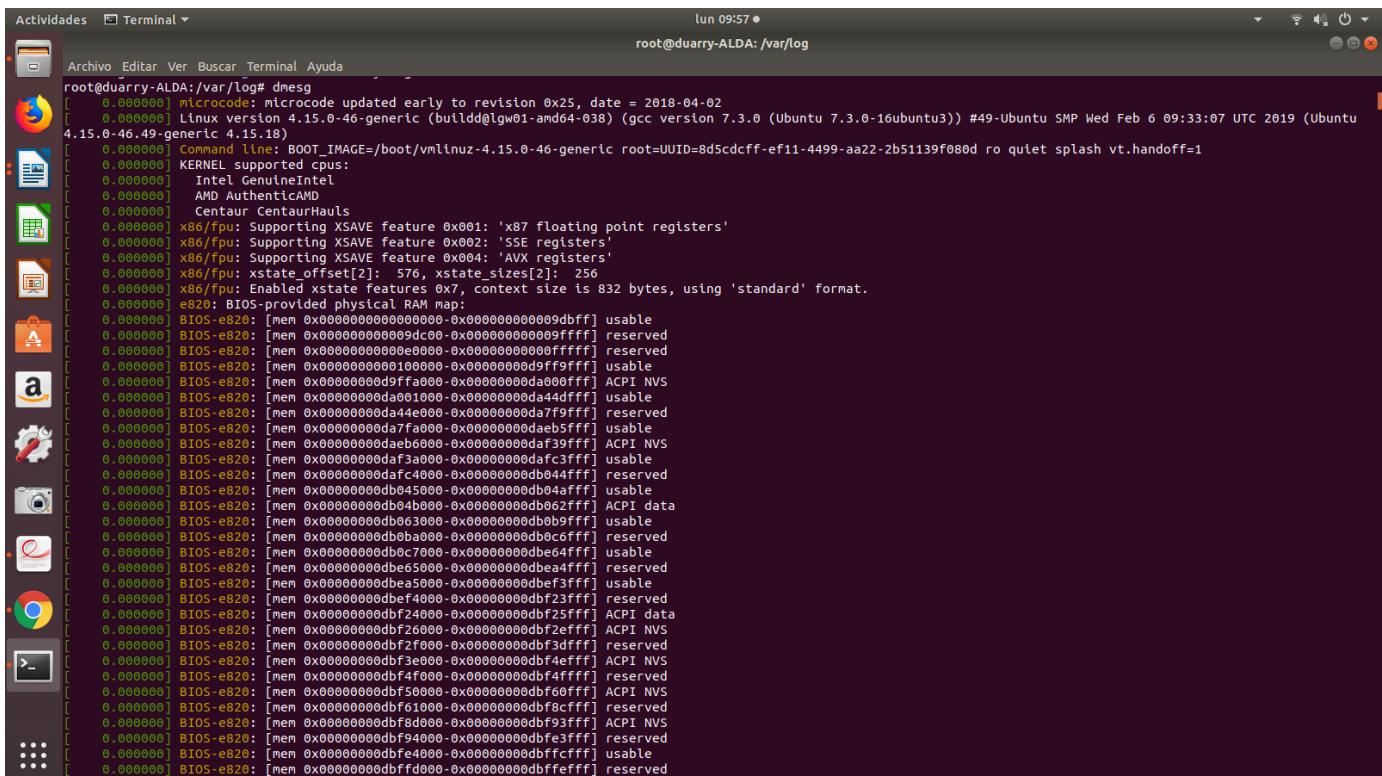
El sistema de logs arranca con el script `/etc/init.d/sysklogd`, y tiene dos demonios:

- ✓ **syslogd**: gestiona los logs del sistema. Distribuye los mensajes a archivos, tuberías, destinos remotos, terminales o usuarios, usando las indicaciones especificadas en su archivo de configuración `/etc/syslog.conf`, donde se indica qué se loguea y a dónde se envían estos logs.
- ✓ **klogd**: se encarga de los logs del kernel. Lo normal es que klogd envíe sus mensajes a syslogd pero no siempre es así, sobre todo en los eventos de alta prioridad, que salen directamente por pantalla.

Los logs se guardan en archivos ubicados en el directorio `/var/log`, aunque muchos programas manejan sus propios logs y los guardan en `/var/log/<programa>`. Además, es posible especificar múltiples destinos para un mismo mensaje.

* **Ejemplo:** `/var/log/dmesg`: en este archivo se almacena la información que genera el kernel durante el arranque del sistema. Podemos ver su contenido con el comando dmesg:

\$ dmesg



```
lun 09:57 ●
root@duarry-ALDA: /var/log
[ 0.000000] microcode: microcode updated early to revision 0x25, date = 2018-04-02
[ 0.000000] Linux version 4.15.0-46-generic (build@lgw01-amd64-038) (gcc version 7.3.0 (Ubuntu 7.3.0-16ubuntu3)) #49-Ubuntu SMP Wed Feb 6 09:33:07 UTC 2019 (Ubuntu 4.15.0-46.49-generic 4.15.18)
[ 0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-4.15.0-46-generic root=UUID=8d5cdcff-ef11-4499-aa22-2b51139f080d ro quiet splash vt.handoff=1
[ 0.000000] KERNEL supported cpus:
[ 0.000000]   Intel GenuineIntel
[ 0.000000]   AMD AuthenticAMD
[ 0.000000]   Centaur CentaurHauls
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[ 0.000000] x86/fpu: xstate_offset[2]: 576, xstate_stzsz[2]: 256
[ 0.000000] x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard' format.
[ 0.000000] e820: BIOS-provided physical RAM map:
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x0000000000009dbfff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000009dc00-0x0000000000009ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x0000000000ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000100000-0x00000000009ffff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000dfffa000-0x000000000da00ff] ACPI NVS
[ 0.000000] BIOS-e820: [mem 0x0000000000da001000-0x000000000da40ffff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000da44e000-0x000000000da7f9ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000da7fa000-0x000000000daeb5ffff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000daeb6000-0x000000000daef3ffff] ACPI NVS
[ 0.000000] BIOS-e820: [mem 0x0000000000daf3a000-0x000000000dafc3ffff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000dafc4000-0x000000000db044ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000db045000-0x000000000db04affff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000db04b000-0x000000000db062ffff] ACPI data
[ 0.000000] BIOS-e820: [mem 0x0000000000db063000-0x000000000db0b9ffff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000db0ba000-0x000000000db0c6ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000db07000-0x000000000db0e4ffff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000db0e5000-0x000000000dbeadffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000dbea5000-0x000000000dbef3ffff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000dbe4000-0x000000000dbf23ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000dbf24000-0x000000000dbf25ffff] ACPI data
[ 0.000000] BIOS-e820: [mem 0x0000000000dbf26000-0x000000000dbf2effff] ACPI NVS
[ 0.000000] BIOS-e820: [mem 0x0000000000dbf2f000-0x000000000dbf3df] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000dbf3e000-0x000000000dbf4effff] ACPI NVS
[ 0.000000] BIOS-e820: [mem 0x0000000000dbf4f000-0x000000000dbf4ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000dbf50000-0x000000000dbf60ffff] ACPI NVS
[ 0.000000] BIOS-e820: [mem 0x0000000000dbf61000-0x000000000dbf8cffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000dbf6d000-0x000000000dbf93ffff] ACPI NVS
[ 0.000000] BIOS-e820: [mem 0x0000000000dbf94000-0x000000000dbfe3ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000dbfe4000-0x000000000dbffcff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000dbffd000-0x000000000dbffeffff] reserved
```

Los archivos de log crecen y con el tiempo se pueden volver muy extensos, pero en `/etc/cron.daily` está el script `/etc/cron.daily/logrotate`, (cuyo archivo de configuración es `/etc/logrotate.conf`), que se encarga de comprimirlos, añadiéndoles la extensión .1.gz, .2.gz, etc., volviendo a crear uno vacío.

```
root@duarry-ALDA: /etc
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.3 cron.daily/logrotate
#!/bin/sh

# Clean non existent log file entries from status file
cd /var/lib/logrotate
test -e status || touch status
head -1 status > status.clean
sed 's/'//g' status | while read logfile date
do
    [ -e "$logfile" ] && echo "\"$logfile\" $date"
done >> status.clean
mv status.clean status

test -x /usr/sbin/logrotate || exit 0
/usr/sbin/logrotate /etc/logrotate.conf

[ 14 líneas leídas ]
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Tex^J Justificar^C Posición
^X Salir      ^R Leer fich.^V Reemplazar^U Pegar txt ^T Corrector ^_ Ir a línea
```

```
root@duarry-ALDA: /etc
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.3 /etc/logrotate.conf
# see "man logrotate" for details
# rotate log files weekly
weekly

# use the syslog group by default, since this is the owning group
# of /var/log/syslog.
su root syslog

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp, or btmp -- we'll rotate them here
/var/log/wtmp {
    missingok
    monthly
    create 0664 root utmp
    rotate 1
}

/var/log/btmp {
    missingok
    monthly
    create 0660 root utmp
    rotate 1
}

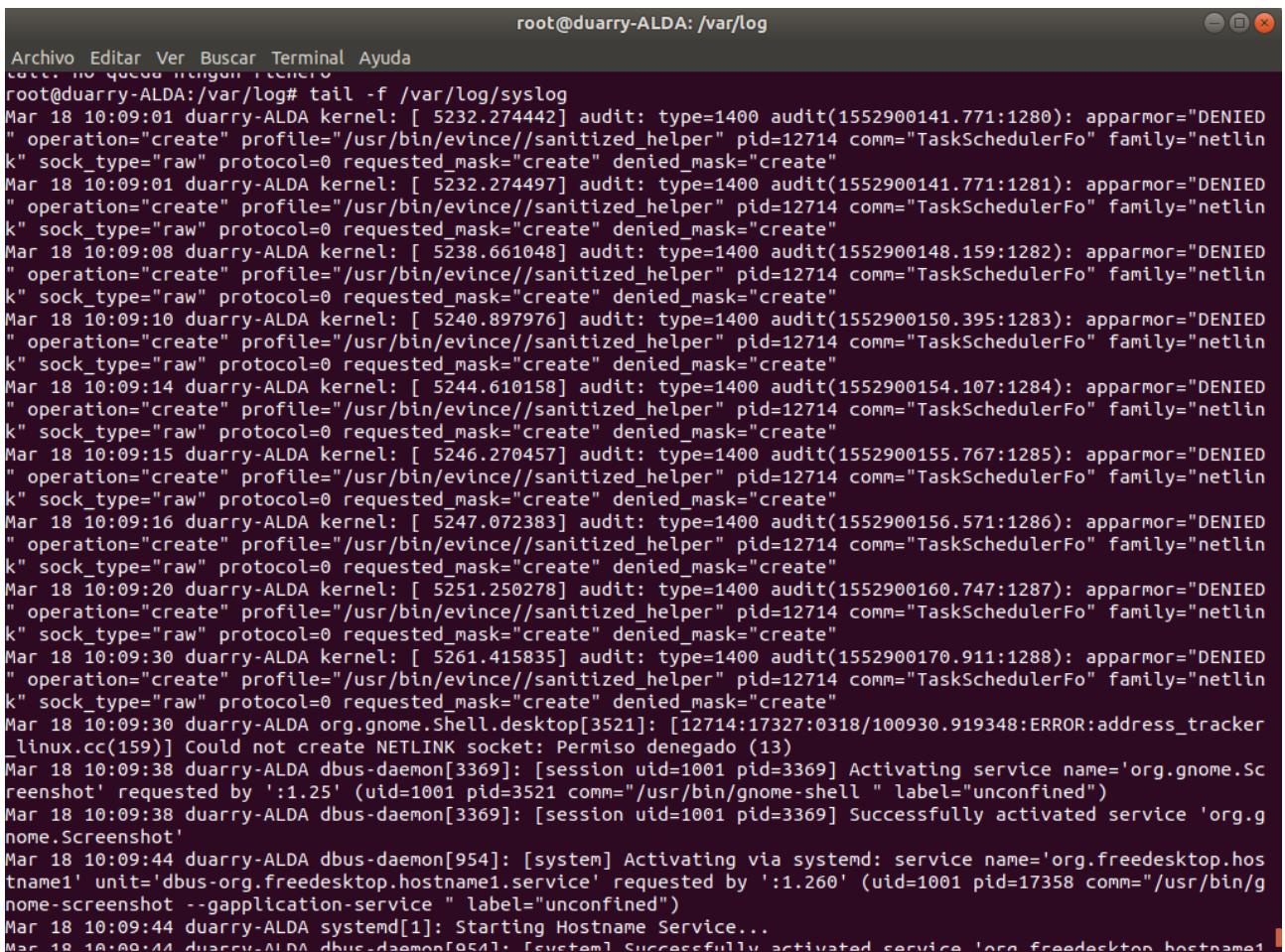
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Texto ^J Justificar ^C Posición M-U Deshacer
^X Salir      ^R Leer fich. ^V Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea M-E Rehacer
```

2. Monitorizar los logs en la consola:

Para monitorizar los logs en la consola, utilizaremos el comando **tail**. Este comando muestra las últimas líneas de uno o más archivos de texto, pero con la opción -f, en lugar de mostrar las últimas diez líneas y terminar, tail seguirá activo y conforme se añadan nuevas líneas al fichero las imprimirá, lo que es muy útil para monitorizar archivos.

Para monitorizar los logs en la consola, por ejemplo el archivo **/var/log/syslog**, haremos lo siguiente:

```
$ tail -f /var/log/syslog
```



A terminal window titled "root@duarry-ALDA: /var/log". The window shows the command "tail -f /var/log/syslog" being run and its output. The output consists of numerous audit log entries from the kernel, detailing various system calls and their parameters. The log entries are timestamped and show details like audit type, profile, pid, comm, family, and specific system call details like "audit(1552900141.771:1280): apparmor=DENIED". The window has a standard Linux terminal interface with a title bar, menu bar, and scroll bars.

```
root@duarry-ALDA: /var/log
Archivo Editar Ver Buscar Terminal Ayuda
root@duarry-ALDA:/var/log# tail -f /var/log/syslog
Mar 18 10:09:01 duarry-ALDA kernel: [ 5232.274442] audit: type=1400 audit(1552900141.771:1280): apparmor="DENIED" operation="create" profile="/usr/bin/evince//sanitized_helper" pid=12714 comm="TaskSchedulerFo" family="netlink" sock_type="raw" protocol=0 requested_mask="create" denied_mask="create"
Mar 18 10:09:01 duarry-ALDA kernel: [ 5232.274497] audit: type=1400 audit(1552900141.771:1281): apparmor="DENIED" operation="create" profile="/usr/bin/evince//sanitized_helper" pid=12714 comm="TaskSchedulerFo" family="netlink" sock_type="raw" protocol=0 requested_mask="create" denied_mask="create"
Mar 18 10:09:08 duarry-ALDA kernel: [ 5238.661048] audit: type=1400 audit(1552900148.159:1282): apparmor="DENIED" operation="create" profile="/usr/bin/evince//sanitized_helper" pid=12714 comm="TaskSchedulerFo" family="netlink" sock_type="raw" protocol=0 requested_mask="create" denied_mask="create"
Mar 18 10:09:10 duarry-ALDA kernel: [ 5240.897976] audit: type=1400 audit(1552900150.395:1283): apparmor="DENIED" operation="create" profile="/usr/bin/evince//sanitized_helper" pid=12714 comm="TaskSchedulerFo" family="netlink" sock_type="raw" protocol=0 requested_mask="create" denied_mask="create"
Mar 18 10:09:14 duarry-ALDA kernel: [ 5244.610158] audit: type=1400 audit(1552900154.107:1284): apparmor="DENIED" operation="create" profile="/usr/bin/evince//sanitized_helper" pid=12714 comm="TaskSchedulerFo" family="netlink" sock_type="raw" protocol=0 requested_mask="create" denied_mask="create"
Mar 18 10:09:15 duarry-ALDA kernel: [ 5246.270457] audit: type=1400 audit(1552900155.767:1285): apparmor="DENIED" operation="create" profile="/usr/bin/evince//sanitized_helper" pid=12714 comm="TaskSchedulerFo" family="netlink" sock_type="raw" protocol=0 requested_mask="create" denied_mask="create"
Mar 18 10:09:16 duarry-ALDA kernel: [ 5247.072383] audit: type=1400 audit(1552900156.571:1286): apparmor="DENIED" operation="create" profile="/usr/bin/evince//sanitized_helper" pid=12714 comm="TaskSchedulerFo" family="netlink" sock_type="raw" protocol=0 requested_mask="create" denied_mask="create"
Mar 18 10:09:20 duarry-ALDA kernel: [ 5251.250278] audit: type=1400 audit(1552900160.747:1287): apparmor="DENIED" operation="create" profile="/usr/bin/evince//sanitized_helper" pid=12714 comm="TaskSchedulerFo" family="netlink" sock_type="raw" protocol=0 requested_mask="create" denied_mask="create"
Mar 18 10:09:30 duarry-ALDA org.gnome.Shell.desktop[3521]: [12714:17327:0318/100930.919348:ERROR:address_tracker_linux.cc(159)] Could not create NETLINK socket: Permiso denegado (13)
Mar 18 10:09:38 duarry-ALDA dbus-daemon[3369]: [session uid=1001 pid=3369] Activating service name='org.gnome.Screenshot' requested by ':1.25' (uid=1001 pid=3521 comm="/usr/bin/gnome-shell" label="unconfined")
Mar 18 10:09:38 duarry-ALDA dbus-daemon[3369]: [session uid=1001 pid=3369] Successfully activated service 'org.gnome.Screenshot'
Mar 18 10:09:44 duarry-ALDA dbus-daemon[954]: [system] Activating via systemd: service name='org.freedesktop.hostname1' unit='dbus-org.freedesktop.hostname1.service' requested by ':1.260' (uid=1001 pid=17358 comm="/usr/bin/gnome-screenshot --gapplication-service" label="unconfined")
Mar 18 10:09:44 duarry-ALDA systemd[1]: Starting Hostname Service...
Mar 18 10:09:44 duarry-ALDA dbus-daemon[954]: [system] Successfully activated service 'org.freedesktop.hostname1'
```

3. Cómo enviar todos los logs a un archivo:

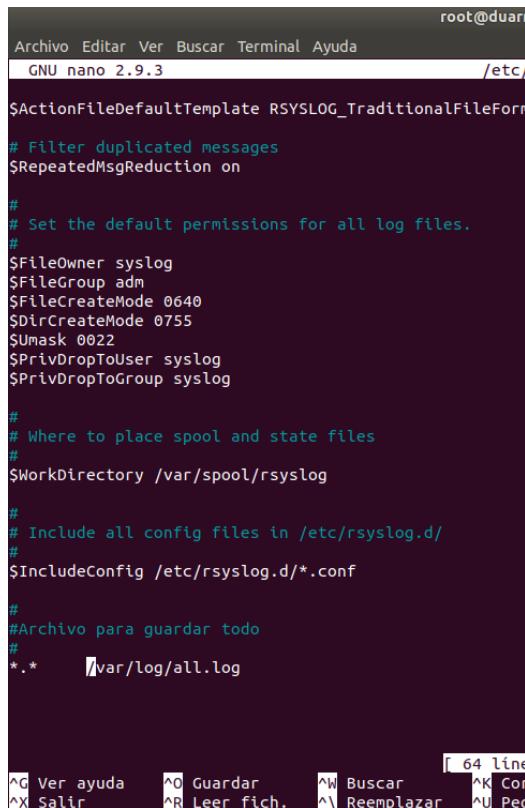
Muchas veces los logs no se miran, por lo que es buena idea configurar el sistema para enviar todos los logs a un archivo y visualizarlo en un terminal, para así poder ver en tiempo real todo lo que pasa en nuestra máquina. Para ello:

- a. Creamos el archivo vacío **/var/log/all.log**:

```
$ touch /var/log/all.log
```

b. Añadimos al archivo `/etc/rsyslog.conf` la línea:

```
*.*    /var/log/all.log
```



```
root@duarry-ALDA:/etc# nano /etc/rsyslog.conf
root@duarry-ALDA:/etc# cat /etc/rsyslog.conf
# ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat
#
# Filter duplicated messages
$RepeatedMsgReduction on
#
# Set the default permissions for all log files.
#
$FileOwner syslog
$FileGroup adm
$FileCreateMode 0640
$DirCreateMode 0755
$Umask 0022
$PrivDropToUser syslog
$PrivDropToGroup syslog
#
# Where to place spool and state files
#
$WorkDirectory /var/spool/rsyslog
#
# Include all config files in /etc/rsyslog.d/
#
$IncludeConfig /etc/rsyslog.d/*.conf
#
# Archivo para guardar todo
#
*.*    /var/log/all.log

root@duarry-ALDA:/etc#
```

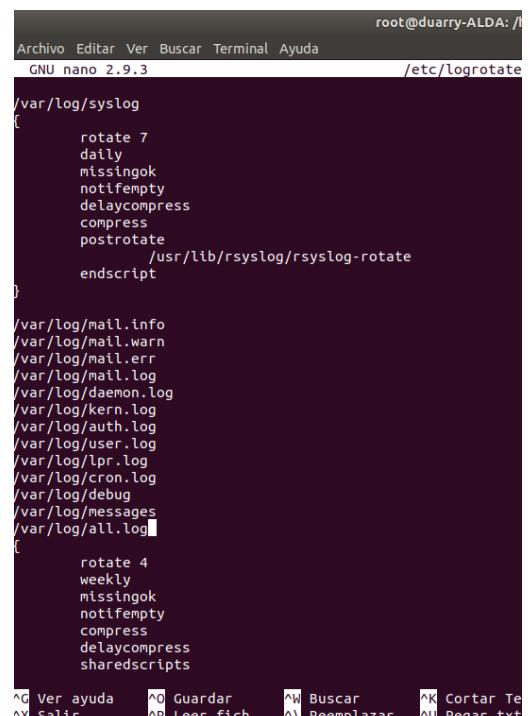
c. Volvemos a cargar la configuración de rsysklogd:

```
$ /etc/init.d/rsyslog force-reload
```

```
root@duarry-ALDA:/etc/init.d# /etc/init.d/rsyslog force-reload
[ ok ] Reloading rsyslog configuration (via systemctl): rsyslog.service.
```

d. Añadimos al archivo `/etc/logrotate.d/rsyslog` la línea:

```
/var/log/all.log
```



```
root@duarry-ALDA:/etc# nano /etc/logrotate.d/rsyslog
root@duarry-ALDA:/etc# cat /etc/logrotate.d/rsyslog
/var/log/syslog
{
    rotate 7
    daily
    missingok
    notifempty
    delaycompress
    compress
    postrotate
        /usr/lib/rsyslog/rsyslog-rotate
    endscript
}

/var/log/mail.info
/var/log/mail.warn
/var/log/mail.err
/var/log/mail.log
/var/log/daemon.log
/var/log/kern.log
/var/log/auth.log
/var/log/user.log
/var/log/lpr.log
/var/log/cron.log
/var/log/debug
/var/log/messages
/var/log/all.log
{
    rotate 4
    weekly
    missingok
    notifempty
    compress
    delaycompress
    sharedscripts
}

root@duarry-ALDA:/etc#
```

e. Por último, abrimos una consola y ejecutamos el comando:

```
$ tail -f /var/log/all.log
```

4. Scripts que generan logs:

El comando logger permite enviar eventos al demonio syslogd, por lo que se utiliza en scripts para registrar mensajes vía sysklogd. Por ejemplo, si hacemos:

```
$ logger -t mi_programa -f /var/log/messages "Mensaje ejemplo"
```

veremos que se ha añadido a </var/log/messages> la línea:

May 14 23:10:13 pc450 mi_programa: Mensaje ejemplo

Manejo de servicios Linux:

Linux ofrece multitud de servicios o servidores, estos pueden iniciar o arrancar junto con la carga del sistema o pueden después ser puestos a funcionar cuando se requieran (es lo mejor). Parte esencial de la administración de sistemas Linux es continuamente trabajar con los servicios que este proporciona, cosa que es bastante sencilla. En este tutorial aprenderás todo lo necesario sobre como iniciar/detener/reiniciar etc. los servicios de tu equipo GNU/Linux.

1. Iniciando servicios desde el arranque del sistema

En muchos casos es conveniente que un servidor o servicio inicien junto con el arranque del equipo en si, por ejemplo, el servidor web Apache o alguna base de datos, esto es para que estén disponibles todo el tiempo y no se requiera de intervención del administrador para iniciarlos.

En Linux, a diferencia de otros sistemas operativos, es posible configurarlo en base a niveles de ejecución (run levels), cada nivel de ejecución (en la mayoría de las distribuciones son 7), inicia o detiene (Start o Kill) ciertos servicios. Estos niveles son los siguientes:

- **0** Detener o apagar el sistema
- **1** Modo monousuario, generalmente utilizado para mantenimiento del sistema
- **2** Modo multiusuario, pero sin soporte de red
- **3** Modo multiusuario completo, con servicios de red
- **4** No se usa, puede usarse para un inicio personalizado
- **5** Modo multiusuario completo con inicio gráfico (X Window)
- **6** Modo de reinicio (reset)

Por ejemplo, el nivel 0, que apaga el equipo, mata o detiene a todos los procesos del sistema, todos los servicios, lo mismo hace el 6 con la diferencia que después inicia un script que permite reiniciar el sistema.

El nivel por omisión o por default del sistema está definido en [*/etc/inittab*](#), en la línea init default:

```
$> grep initdefault /etc/inittab id:5:initdefault:  
# runlevel 0 is System halt  
# runlevel 6 is System reboot
```

La salida es de una distro OpenSuse 10.

Existe una línea que comienza con id, seguido por 5 que indicaría el nivel por default del sistema, basta con cambiar este valor con cualquier editor, reiniciar el equipo y se entraría a otro nivel, por ejemplo, el 3 que tiene una funcionalidad completa pero sin sistema de ventanas X Windows.

En el listado previo aparecen dos líneas de comentarios, las que empiezan con #, que hacen la advertencia de no usar nunca los niveles 0 y 6 de init default, ya que jamás arrancaría el equipo. Se tendría que usar un método de rescate.

El nivel de ejecución actual del sistema puede ser consultado con **who -r** y con el comando **runlevel**, este último muestra dos números el primero es el nivel previo en que se estaba y el segundo el nivel actual, si no se ha cambiado de nivel aparece una N.

```
usuario@duarry-ALDA:~$ who -r
    `run-level' 5 2019-03-19 12:04
usuario@duarry-ALDA:~$ runlevel
N 5
usuario@duarry-ALDA:~$ █
```

2. Comando init

Este comando te permite cambiar el nivel de ejecución actual por otro, es decir puedes estar en el nivel 3 y en cualquier momento pasar al 5 o viceversa, o entrar a un modo monousuario para tareas de mantenimiento. Esto sin necesidad de cambiar la línea init default en /etc/inittab, simplemente usando el comando init:

\$ init 3

Puedes **reiniciar** el equipo con **init 6** o incluso **apagarlo** con **init 0**, de hecho, shutdown y otros comandos como halt, lo que hacen es invocar a init para apagar el equipo más otras tareas como sincronizar buffers, etc.

Si estás en una máquina que tenga particiones con sistemas de archivos ext2 o de windows como FAT, si invocas init 0 o init 6 tal cual, corres el riesgo de perder datos que no estén guardados en disco todavía, para esto combina init 0 con el comando sync que escribe lo que esté en buffers de memoria a disco:

\$ sync; sync; init 0

Con un doble sync nos aseguramos que todo se escriba en disco antes de comenzar a apagar.

3. Iniciando servicios manualmente, directorio init.d

Dentro de esta carpeta ubicada en **/etc** o en **/etc/rc.d** dependiendo de la distribución, se encuentran una serie de scripts que permiten iniciar/detener la gran mayoría de los servicios/servidores que estén instalados en el equipo.

Estos scripts están programados de tal manera que la mayoría reconoce los siguientes argumentos:

- start
- stop
- restart
- status

Los argumentos son auto descriptivos, y tienen permisos de ejecución, entonces siendo root es posible iniciar un servicio de la siguiente manera, por ejemplo, samba:

```
#> /etc/init.d/docker start
```

```
usuario@duarry-ALDA:/etc/init.d$ /etc/init.d/docker start
[ ok ] Starting docker (via systemctl): docker.service.
usuario@duarry-ALDA:/etc/init.d$ █
```

Solo que hay que cambiar `start` por `stop | restart | status` para detenerlo, reiniciarlo (releer archivos de configuración) o checar su estatus.

4. El comando Service:

Este comando permite también iniciar y/o detener servicios, de hecho funciona exactamente igual a como si escribieramos la ruta completa hacia el directorio `init.d`, con `service` se indica de la siguiente manera:

```
$ service mysql status
```

```
usuario@duarry-ALDA:/etc/init.d$ service mysql status
● mysql.service - MySQL Community Server
  Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: en
  Active: active (running) since Tue 2019-03-19 12:04:12 CET; 1h 15min ago
    Main PID: 1578 (mysqld)
      Tasks: 28 (limit: 4915)
     CGroup: /system.slice/mysql.service
             └─1578 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid

Warning: Journal has been rotated since unit was started. Log output is incomplete
lines 1-9/9 (END)
```

Si se desea iniciararlo/pararlo:

```
$ service mysql start/stop
```

5. Los directorios rc:

Deabajo de `/etc` se encuentra el directorio `rc.d` que a la vez contiene un directorio para cada nivel de ejecución, así tenemos `rc0.d`, `rc1.d`, `rc2.d`, `rc3.d`, etc. Hay algunos SO en los que estos directorios están ubicados directamente en `/etc`.

Cada uno de estos directorios contiene scripts (o más bien enlaces a scripts) que apuntan al directorio `init.d`, entonces el comando init dependiendo del nivel indicado leerá cada uno de los enlaces o accesos directos del directorio respectivo.

Un ejemplo:

lrwxrwxrwx	1	root	root	7 Oct 20 10:05	K22dbus -> ..init.d/dbus
lrwxrwxrwx	1	root	root	9 Oct 20 10:05	K22resmgr -> ..init.d/resmgr
lrwxrwxrwx	1	root	root	8 Oct 20 10:05	K24fbset -> ..init.d/fbset
lrwxrwxrwx	1	root	root	9 Oct 20 10:05	K24random -> ..init.d/random
lrwxrwxrwx	1	root	root	8 Oct 20 18:23	S01fbset -> ..init.d/fbset
lrwxrwxrwx	1	root	root	9 Oct 20 18:22	S01random -> ..init.d/random

Todos son enlaces al directorio `init.d` que es donde realmente están ubicados los scripts de arranque de los servicios o servidores. También, todos los enlaces comienzan con K (kill) o con S (start), los que comienzan con K son scripts que recibirán el argumento `stop` y los que comienzan con S el de `start`.

Después de la K o S sigue un número consecutivo, seguido generalmente del nombre del servicio que afectan, el número secuencial es simplemente el orden en que se leerán los scripts, primero los K comenzando con el 01 y hacia adelante y después los S. Entonces si por ejemplo no queremos que se inicie samba en el nivel 3 bastaría con borrar su enlace en este directorio:

```
$ pwd /etc/rc.d/rc3.d  
$ rm S54smb
```

Si por lo contrario lo que deseamos es iniciar (o apagar) un servicio, basta con crear su enlace en el directorio respectivo:

```
$ pwd /etc/rc.d/rc5.d  
$ ln -s /etc/rc.d/init.d/mysql S90mysql  
$ ls -l S90mysql
```

Con esto la siguiente vez que iniciemos el equipo o cambiemos a nivel 5 con init, también se iniciará el servidor de la base de datos MySQL. El número 90 es escogido al azar entre 01 y 99 es simplemente el orden en que serán iniciados o detenidos los servicios.

6. Comando chkconfig

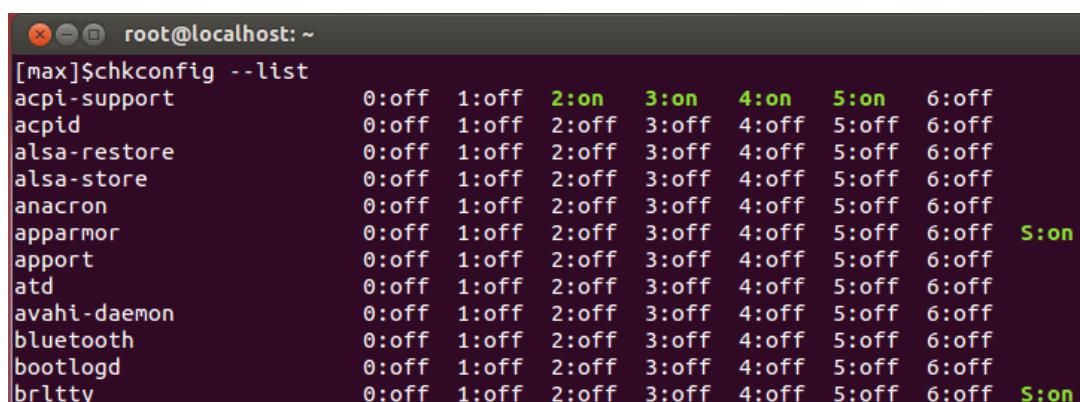
Es una herramienta para configurar los servicios más ágilmente, y que permite configurar que servicios se arrancan/detienen en cada nivel de ejecución.

Chkconfig no inicia ni detiene servicios al momento, tan solo crea o elimina precisamente los enlaces. Si lo que se quiere es iniciar o detener el servicio manualmente hay que usar service o directamente el script con su argumento conveniente.

- Ejemplos:

a) Con la opción --list nos da una lista completa de todos los servicios instalados y para cada nivel si arrancará (on) al entrar a ese nivel o se detendrá (off) o simplemente no se iniciará. Nótese que al final de la lista vienen los servicios que dependen del superservidor xinetd.

```
$ chkconfig --list
```



```
root@localhost: ~  
[max]$ chkconfig --list  
acpi-support          0:off  1:off  2:on   3:on   4:on   5:on   6:off  
acpid                0:off  1:off  2:off  3:off  4:off  5:off  6:off  
alsa-restore          0:off  1:off  2:off  3:off  4:off  5:off  6:off  
alsa-store            0:off  1:off  2:off  3:off  4:off  5:off  6:off  
anacron              0:off  1:off  2:off  3:off  4:off  5:off  6:off  
apparmor             0:off  1:off  2:off  3:off  4:off  5:off  6:off  
apport               0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on  
atd                  0:off  1:off  2:off  3:off  4:off  5:off  6:off  
avahi-daemon         0:off  1:off  2:off  3:off  4:off  5:off  6:off  
bluetooth            0:off  1:off  2:off  3:off  4:off  5:off  6:off  
bootlogd             0:off  1:off  2:off  3:off  4:off  5:off  6:off  
brltty               0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
```

- b) Si se indica como argumento el nombre de algún servicio nos regresará su estatus:
\$ chkconfig smb

- c) Sin ningún argumento nos dará una lista con el status de todos los servicios:
\$ chkconfig

*httpd off
mysql on
smb on*

- d) Si se desea que el servidor web apache (httpd) inicie cuando se entra en el nivel 5, entonces usamos la opción --level:

\$ chkconfig --level 5 httpd on

- e) Si la base de datos MySQL no se desea que inicie en los niveles 3 y 5, solo hasta que el administrador decida arrancarla, se ejecuta:

\$ chkconfig --level 35 mysql off

MD5SUM

El programa md5sum está diseñado para verificar la integridad de datos usando el hash criptográfico MD5 de 128-bit. MD5 hashes usados adecuadamente pueden confirmar la integridad de los archivos y la autenticidad.

La mayoría de las distribuciones linux vienen con la utilidad md5sum, por eso la instalación es innecesaria.

- Empezamos:

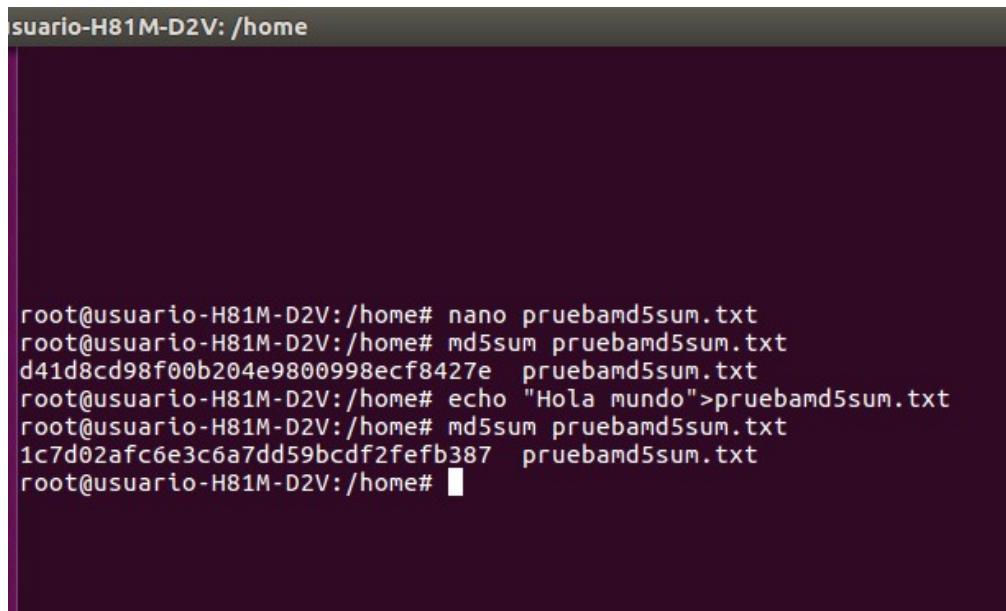
1.- Crear un archivo .txt y comprobar si cambia la codificación al realizar cambios.

2.- Crear archivo.txt y ejecutar:

```
$ md5sum archivo.txt
```

```
$ echo "Hola mundo" > archivo.txt
```

```
$ md5sum archivo.txt
```



A screenshot of a terminal window titled "usuari-H81M-D2V: /home". The terminal shows the following command sequence:

```
usuari-H81M-D2V: /home
root@usuario-H81M-D2V:/home# nano pruebamd5sum.txt
root@usuario-H81M-D2V:/home# md5sum pruebamd5sum.txt
d41d8cd98f00b204e9800998ecf8427e  pruebamd5sum.txt
root@usuario-H81M-D2V:/home# echo "Hola mundo">>pruebamd5sum.txt
root@usuario-H81M-D2V:/home# md5sum pruebamd5sum.txt
1c7d02afc6e3c6a7dd59bcdff2fefb387  pruebamd5sum.txt
root@usuario-H81M-D2V:/home#
```

MOTION:

1.- Conectar la camara web por usb.

2.- Comprobar que está conectada (Philips) → **lsusb**

```
root@usuario-ALDA:/home/usuario/Escritorio/motion# lsusb
Bus 002 Device 002: ID 8087:8000 Intel Corp.
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 006 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 005 Device 004: ID 045e:07b9 Microsoft Corp.
Bus 005 Device 003: ID 04f3:0235 Elan Microelectronics Corp.
Bus 005 Device 002: ID 2109:3431 VIA Labs, Inc. Hub
Bus 005 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 8087:8008 Intel Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 004: ID 0471:0331 Philips (or NXP) SPC 1300NC PC Camera
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
root@usuario-ALDA:/home/usuario/Escritorio/motion#
```

3.- Instalar motion en linux → **sudo apt-get install motion**

4.- En caso de descargar alguna librería instalamos gdebi, aunque en este caso no tenemos que descargar ninguna librería de motion. → **sudo apt-get install gdebi**

5.- Entramos en el archivo → **nano /etc/motion/motion.conf**

- ✓ Buscamos: **#Target base directory for pictures and films** que tiene el directorio **target_dir /var/lib/motion**
- ✓ Cambiamos por la carpeta donde queramos guardar las fotos recogidas. En mi caso: **/home/usuario/Escritorio/motion**

```
root@usuario-ALDA:/home/usuario/Escritorio/motion
GNU nano 2.5.3          Archivo: /etc/motion/motion.conf

# %C = value defined by text_event
# Quotation marks round string are allowed.
#####
# Target base directory for pictures and films
# Recommended to use absolute path. (Default: current working directory)
target_dir /home/usuario/Escritorio/motion

# File path for snapshots (jpeg or ppm) relative to target_dir
# Default: %v-%Y%m%d%H%M%S-snapshot
# Default value is equivalent to legacy oldlayout option
# For Motion 3.0 compatible mode choose: %Y/%m/%d/%H/%M/%S-snapshot
# File extension .jpg or .ppm is automatically added so do not include this.
# Note: A symbolic link called lastsnap.jpg created in the target_dir will always
# point to the latest snapshot, unless snapshot_filename is exactly 'lastsnap'
snapshot_filename %v-%Y%m%d%H%M%S-snapshot

# File path for motion triggered images (jpeg or ppm) relative to target_dir
# Default: %v-%Y%m%d%H%M%S-%q

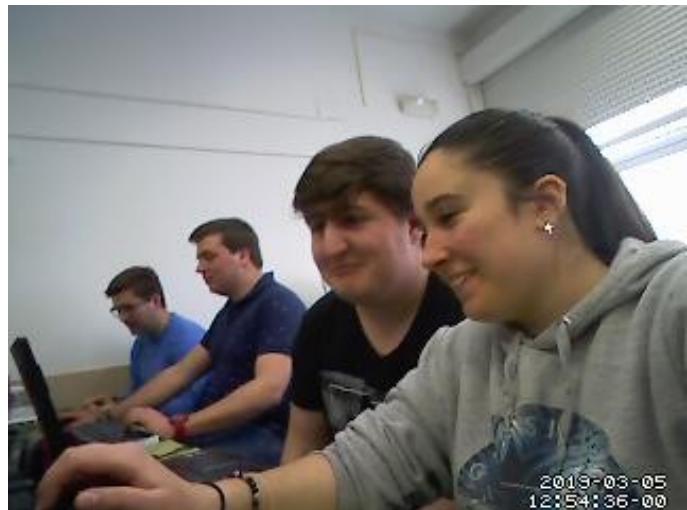
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Tex^J Justificar^C Posición
^X Salir    ^R Leer fich.^U Reemplazar^U Pegar txt ^T Ortografía^I Ir a línea
```

6.- Para iniciar el sistema de video ejecutamos el comando **sudo motion**. Si queremos que se active de forma automática, entramos en sudo **nano /etc/default/motion** y cambiamos:

- ✓ **start_motion_daemon=no** por **start_motion_daemon=yes**

7.- Para parar el motion escribimos en la terminal → **sudo killall motion**

8.- Fotos hechas por la cámara con motion:

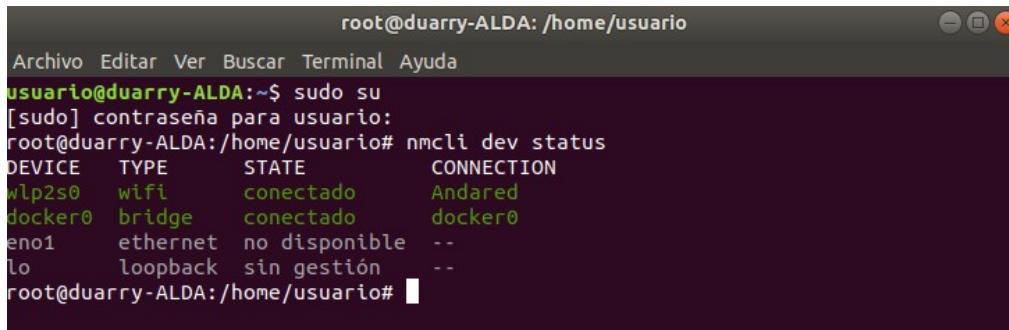


NETWORK-MANAGER:

Seguiremos estos pasos siendo root y escribiendo en la terminal del equipo. Estos pasos garantizan que el servicio NetworkManager esté deshabilitado o habilitado y podremos comprobar las conexiones que controla.

1. Verificamos que está habilitado y comprobamos que conexiones controla.

```
$ chkconfig –list NetworkManager  
$ nmcli dev status
```



A terminal window titled 'root@duarry-ALDA: /home/usuario'. The window shows a command-line session where the user runs 'sudo su' to become root. Then, they run 'nmcli dev status' which outputs a table of network devices and their states. The table includes columns for DEVICE, TYPE, STATE, and CONNECTION. The output shows three active connections: 'wlp2s0' (wifi, connected to 'Andared'), 'docker0' (bridge, connected to 'docker0'), and 'lo' (loopback, no gestion). There are also two inactive entries: 'eno1' (ethernet, no disponible) and 'eth0' (loopback, sin gestion).

DEVICE	TYPE	STATE	CONNECTION
wlp2s0	wifi	conectado	Andared
docker0	bridge	conectado	docker0
eno1	ethernet	no disponible	--
lo	loopback	sin gestión	--

* El comando chkconfig indica si el servicio de Network Manager está habilitado o no.

- x El sistema muestra un error si el servicio de Network Manager no está instalado actualmente:

Error al leer la información en el servicio NetworkManager: no existe tal archivo o directorio

- x Si se muestra este error, no se requiere ninguna acción adicional para deshabilitar el servicio de Network Manager.
- x El sistema muestra una lista de niveles de ejecución numéricos junto con un valor de activación o desactivación que indica si el servicio de Network Manager está habilitado cuando el sistema está operando en el nivel de ejecución dado.

NetworkManager 0: apagado 1: apagado 2: apagado 3: apagado 4: apagado 5: apagado 6: apagado

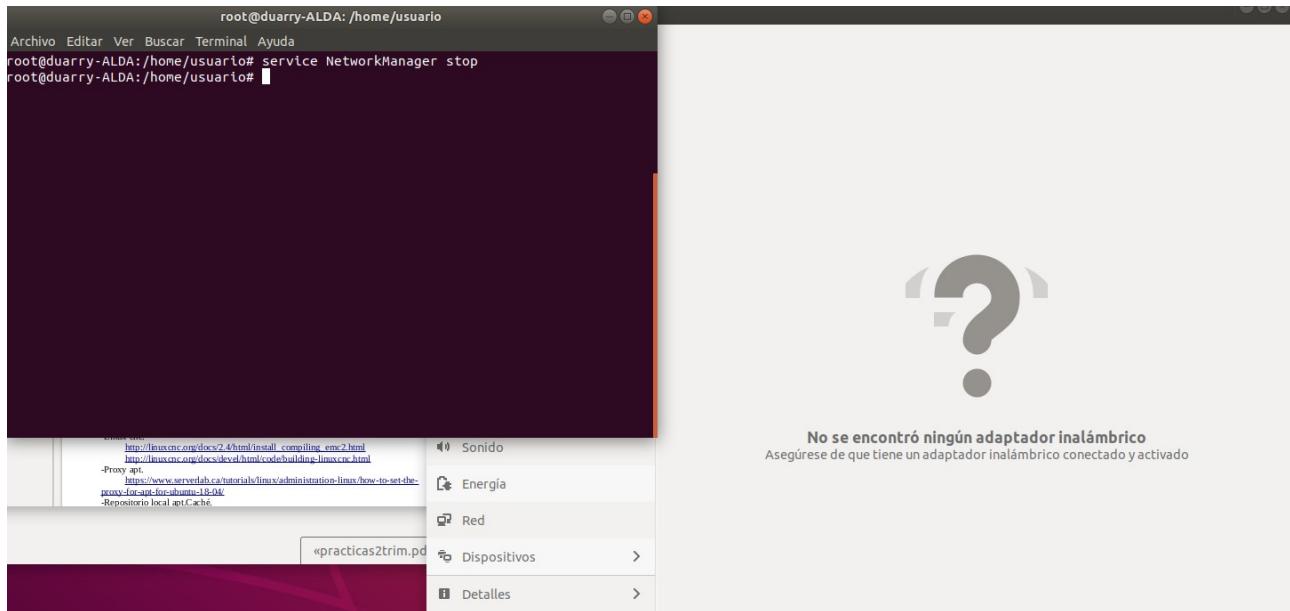
- x Si el valor que se muestra para todos los niveles de ejecución está desactivado, el servicio de Network Manager se desactiva y no se requiere ninguna acción adicional. Si el valor que se muestra para cualquiera de los niveles de ejecución está activado, el servicio Network Manager está habilitado y se requieren acciones adicionales.

2. Nos aseguramos que el servicio esté parado usando el siguiente comando:

```
$ service NetworkManager stop
```

3. Nos aseguramos que esta deshabilitado con el siguiente comando:

```
$ chkconfig NetworkManager off
```



* Si queremos cambiar los ajustes, abrimos los archivos de configuración en un editor de texto. Los archivos de configuración de la interfaz se encuentran en el directorio **/etc/sysconfig/network-scripts/** y tienen nombres de la forma **ifcfg-X**, donde X se reemplaza por el nombre de la interfaz (eth0, p1p5 o em1) .

En cada archivo, nos aseguramos de que:

```
NM_CONTROLLED = no  
ONBOOT = sí
```

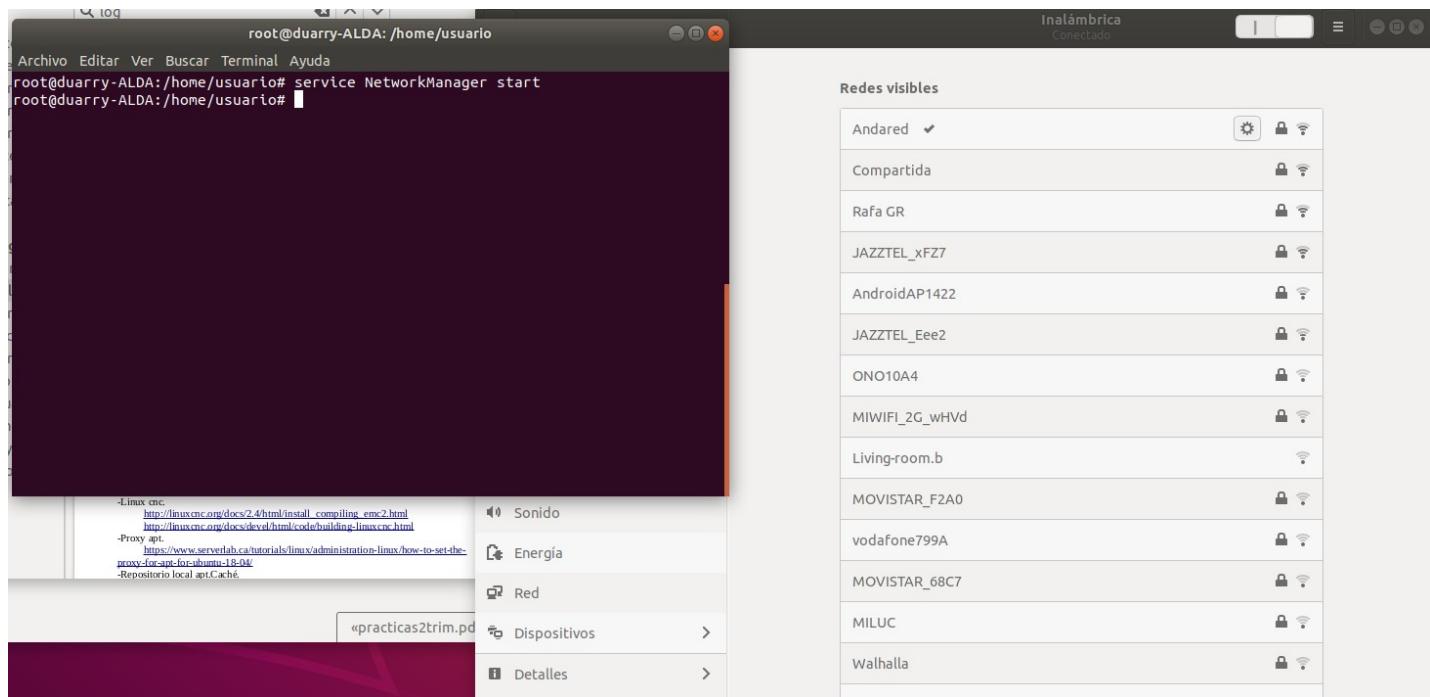
Esta acción garantiza que el servicio de red estándar tomará el control de las interfaces y las activará automáticamente en el arranque.

4. Arrancamos el servicio usando el siguiente comando:

```
$ service network start
```

5. Nos aseguramos de que está habilitado con el comando:

```
$ chkconfig network on
```



NFS:

Servidor (IP)

Instala → nfs-kernel-server

/etc/init.d/nfs-kernel-server start

nano /etc/exports → carpeta que quieras compartir (ruta) y permisos
(rwo,sync,no_subtree_check)

cat /var/log/syslog

Cliente

mount -t nfs (IP):/home/pepe /home/usuario/carpeta2

cd /home/usuario/carpeta2

OPENWRT

Lo haremos todo como un usuario normal, no usaremos el sudo ni root.
No construir en un directorio que tiene espacios en su ruta completa.

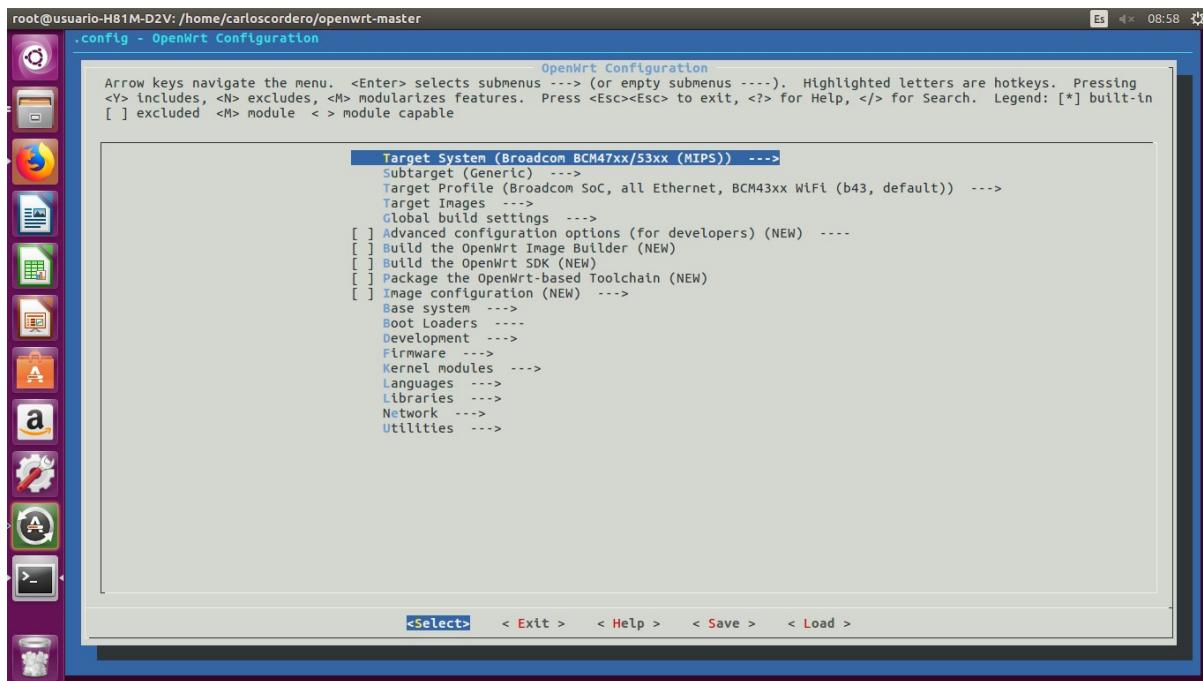
1.- Proceso de instalación:

1. Actualizar feeds: **\$./scripts/feeds update -a**
2. Haga que los paquetes descargados estén disponibles en make menuconfig:
\$./scripts/feeds install -a

2.- Configuración de la imagen

\$ make menuconfig

Cuando abramos menuconfig, debemos configurar los ajustes en este orden (también se muestra en este orden en la interfaz de menuconfig):



1. Target system → categoría general de dispositivos similares.
2. Subtarget → subcategoría del Target system, agrupando dispositivos similares.
3. Target profile → cada dispositivo específico.
4. Package selection
5. Build system settings
6. Kernel modules

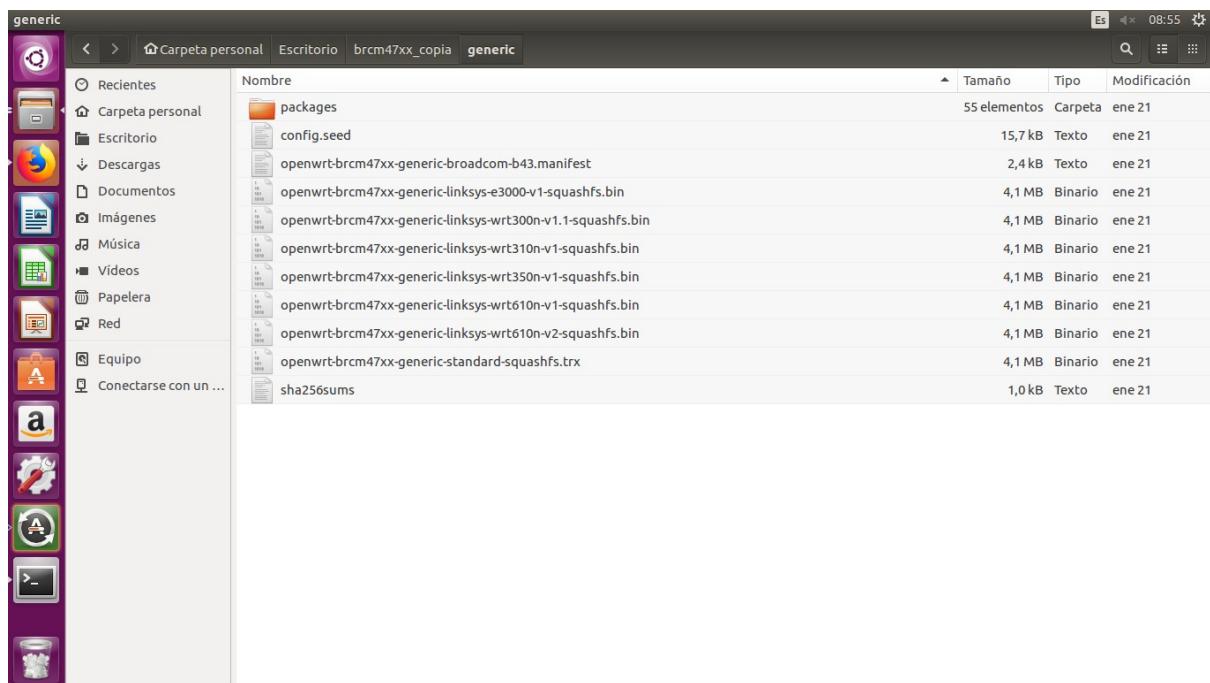
3.- Construcción de la imagen:

\$ make download

\$ make

* Ejemplos:

- <https://forum.openwrt.org/viewtopic.php?pid=129319#p129319>
- <https://forum.openwrt.org/viewtopic.php?id=28267>



Parchear núcleo linux:

En este escenario es necesario compilar la versión en cuestión aplicando todos los parches que se han liberado desde la misma. Para ello se debe utilizar la herramienta patch.

La herramienta patch se utiliza para realizar modificaciones en archivos en modo batch, a partir de archivos diff fuente. Por defecto, patch lee desde entrada estándar.

En mi caso necesitaba compilar la versión 3.2 de Bash. Entonces el primer paso consiste en descargar su código fuente desde el sitio oficial (www.gnu.org/software/bash/):

1. Cambiar a un directorio conveniente para guardar código fuente:

```
$ cd /usr/local/src/
```

2. Descargar el paquete bash-*.tar.gz correspondiente a la versión específica, por ejemplo bash-3.2.48.tar.gz si se desea compilar la versión 3.2

```
$ wget https://mirrors.kernel.org/gnu/bash/bash-3.2.48.tar.gz
```

3. Luego, para cada versión existe una carpeta *-patches/ que contiene todos los parches específicos de cada rama. Para la versión 3.2, navegar hasta la carpeta bash-3.2-patches/. Si se descargó el paquete bash-3.2.48.tar.gz se deben descargar los parches a partir del 49 en adelante.

```
$ wget https://mirrors.kernel.org/gnu/bash/bash-3.2-patches/bash32-049
```

```
$ wget https://mirrors.kernel.org/gnu/bash/bash-3.2-patches/bash32-050
```

```
$ wget https://mirrors.kernel.org/gnu/bash/bash-3.2-patches/bash32-051
```

```
$ wget https://mirrors.kernel.org/gnu/bash/bash-3.2-patches/bash32-052
```

```
$ wget https://mirrors.kernel.org/gnu/bash/bash-3.2-patches/bash32-053
```

```
$ wget https://mirrors.kernel.org/gnu/bash/bash-3.2-patches/bash32-054
```

```
$ wget https://mirrors.kernel.org/gnu/bash/bash-3.2-patches/bash32-055
```

```
$ wget https://mirrors.kernel.org/gnu/bash/bash-3.2-patches/bash32-056
```

```
$ wget https://mirrors.kernel.org/gnu/bash/bash-3.2-patches/bash32-057
```

4. Extraer el contenido del paquete:

```
$ tar -xvf bash-3.2.48.tar.gz
```

5. Revisar cada parche para determinar qué cambios realizan y cómo aplicarlos:

```
$ less bash32-049  
$ less bash32-050  
$ less bash32-051  
$ less bash32-052  
$ less bash32-053  
$ less bash32-054
```

Los parches indican que deben ser aplicado con la opción -p0.

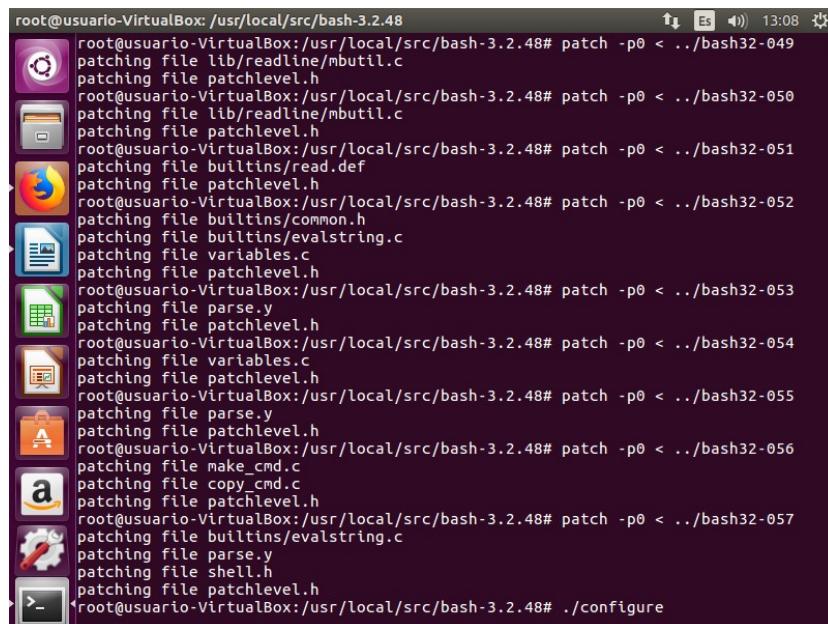
6. Cambiar al directorio extraído:

```
$ cd bash-3.2.48/
```

7. apt-get install patch

8. Aplicamos los parches:

```
$ patch -p0 < ./bash32-049  
$ patch -p0 < ./bash32-050  
$ patch -p0 < ./bash32-051  
$ patch -p0 < ./bash32-052  
$ patch -p0 < ./bash32-053  
$ patch -p0 < ./bash32-054  
$ patch -p0 < ./bash32-055  
$ patch -p0 < ./bash32-056  
$ patch -p0 < ./bash32-057
```



```
root@usuario-VirtualBox:/usr/local/src/bash-3.2.48# patch -p0 < ./bash32-049  
patching file lib/readline/mbutil.c  
patching file patchlevel.h  
root@usuario-VirtualBox:/usr/local/src/bash-3.2.48# patch -p0 < ./bash32-050  
patching file lib/readline/mbutil.c  
patching file patchlevel.h  
root@usuario-VirtualBox:/usr/local/src/bash-3.2.48# patch -p0 < ./bash32-051  
patching file builtins/read.def  
patching file patchlevel.h  
root@usuario-VirtualBox:/usr/local/src/bash-3.2.48# patch -p0 < ./bash32-052  
patching file builtins/common.h  
patching file builtins/evalstring.c  
patching file variables.c  
patching file patchlevel.h  
root@usuario-VirtualBox:/usr/local/src/bash-3.2.48# patch -p0 < ./bash32-053  
patching file parse.y  
patching file patchlevel.h  
root@usuario-VirtualBox:/usr/local/src/bash-3.2.48# patch -p0 < ./bash32-054  
patching file variables.c  
patching file patchlevel.h  
root@usuario-VirtualBox:/usr/local/src/bash-3.2.48# patch -p0 < ./bash32-055  
patching file parse.y  
patching file patchlevel.h  
root@usuario-VirtualBox:/usr/local/src/bash-3.2.48# patch -p0 < ./bash32-056  
patching file make_cmd.c  
patching file copy_cmd.c  
patching file patchlevel.h  
root@usuario-VirtualBox:/usr/local/src/bash-3.2.48# patch -p0 < ./bash32-057  
patching file builtins/evalstring.c  
patching file parse.y  
patching file shell.h  
patching file patchlevel.h  
root@usuario-VirtualBox:/usr/local/src/bash-3.2.48# ./configure
```

9. Ahora es posible compilar el paquete:

```
# ./configure  
  
# make  
  
# make install
```

PHP:

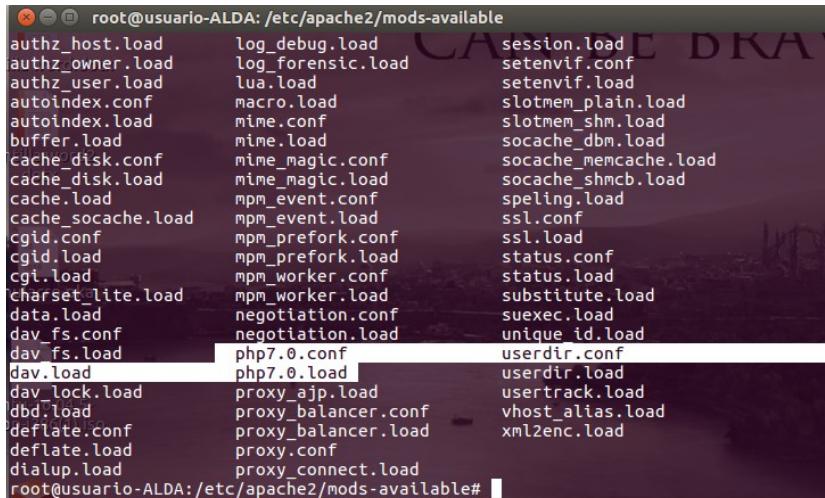
1. Instalamos → apt-get install libapache2-mod-php7.0

2. cd /etc/apache2

3. ls

4. cd mods-available/

5. ls



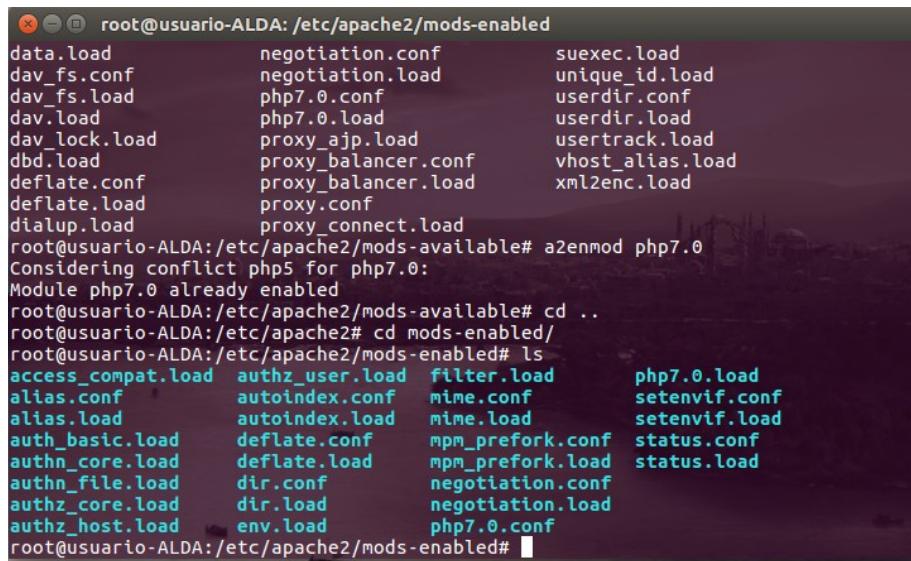
```
root@usuario-ALDA:/etc/apache2/mods-available
authz_host.load      log_debug.load      session.load
authz_owner.load     log_forensic.load   setenvif.conf
authz_user.load      lua.load           setenvif.load
autoindex.conf       macro.load         slotmem_plain.load
autoindex.load        mime.conf          slotmem_shm.load
buffer.load          mime.load          socache_dbm.load
cache_disk.conf      mime_magic.conf   socache_memcache.load
cache_disk.load      mime_magic.load   socache_shmcbl.load
cache.load           mpm_event.conf    spelling.load
cache_socache.load   mpm_event.load    ssl.conf
cgid.conf            mpm_prefork.conf  ssl.load
cgid.load            mpm_prefork.load  status.conf
cgi.load             mpm_worker.conf   status.load
charset_lite.load   mpm_worker.load   substitute.load
data.load            negotiation.conf  suexec.load
dav_fs.conf          negotiation.load  unique_id.load
dav_fs.load          php7.0.conf       userdir.conf
dav.load             php7.0.load       userdir.load
dav_lock.load        proxy_ajp.load    usertrack.load
dbd.load             proxy_balancer.conf vhost_alias.load
deflate.conf         proxy_balancer.load xml2enc.load
deflate.load         proxy.conf        dialup.load
proxy_connect.load  proxy_connect.load
root@usuario-ALDA:/etc/apache2/mods-available#
```

6. a2enmod php7.0

7. cd ..

8. cd mods-enabled/

9. ls → tiene que estar el módulo que hemos cargado (php7.0).



```
root@usuario-ALDA:/etc/apache2/mods-enabled
data.load          negotiation.conf  suexec.load
dav_fs.conf        negotiation.load  unique_id.load
dav_fs.load        php7.0.conf       userdir.conf
dav.load           php7.0.load       userdir.load
dav_lock.load      proxy_ajp.load   usertrack.load
dbd.load           proxy_balancer.conf vhost_alias.load
deflate.conf       proxy_balancer.load xml2enc.load
deflate.load       proxy.conf        dialup.load
proxy_connect.load proxy_connect.load
root@usuario-ALDA:/etc/apache2/mods-available# a2enmod php7.0
Considering conflict php5 for php7.0:
Module php7.0 already enabled
root@usuario-ALDA:/etc/apache2/mods-available# cd ..
root@usuario-ALDA:/etc/apache2# cd mods-enabled/
root@usuario-ALDA:/etc/apache2/mods-enabled# ls
access_compat.load authz_user.load filter.load  php7.0.load
alias.conf         autoindex.conf  mime.conf   setenvif.conf
alias.load         autoindex.load  mime.load   setenvif.load
auth_basic.load    deflate.conf   mpm_prefork.conf  status.conf
authn_core.load    deflate.load   mpm_prefork.load  status.load
authn_file.load   dir.conf      negotiation.conf
authz_core.load    dir.load      negotiation.load
authz_host.load   env.load      php7.0.conf
root@usuario-ALDA:/etc/apache2/mods-enabled#
```

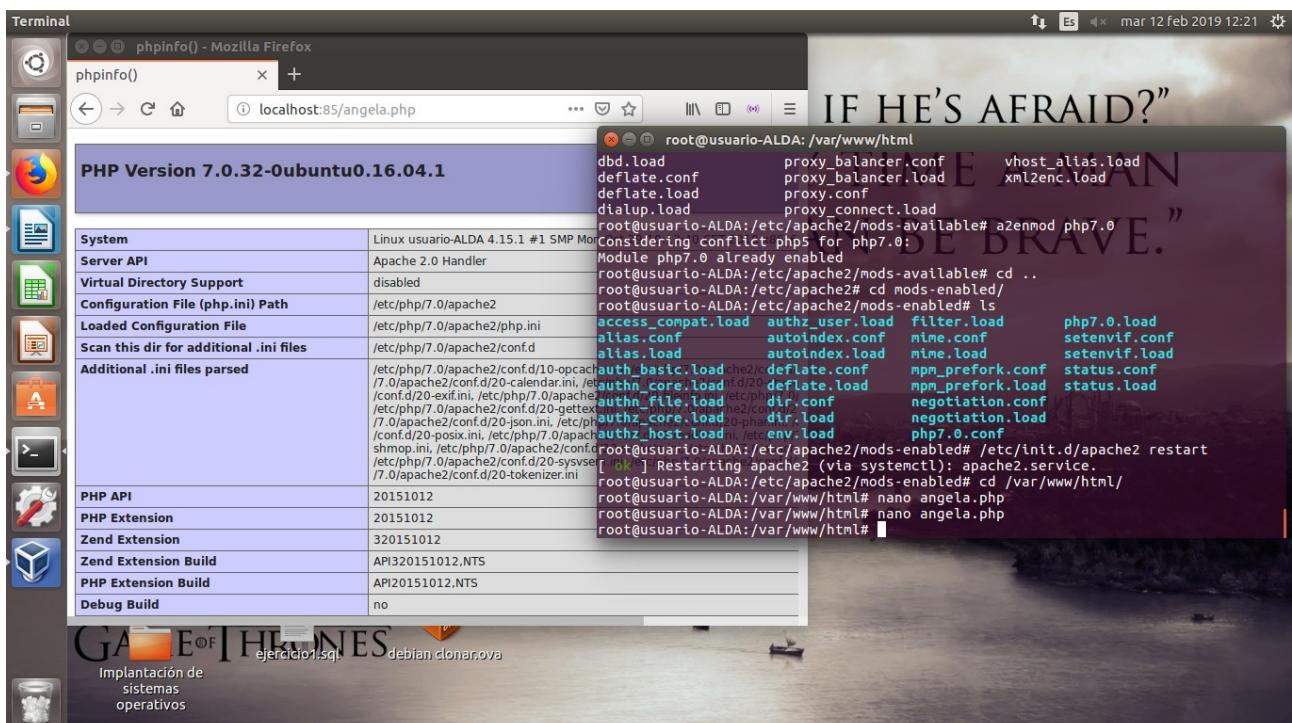
10. /etc/init.d/apache2 restart → reiniciamos el apache.

11. cd /var/www/html/

12. nano angela.php → escribimos dentro:

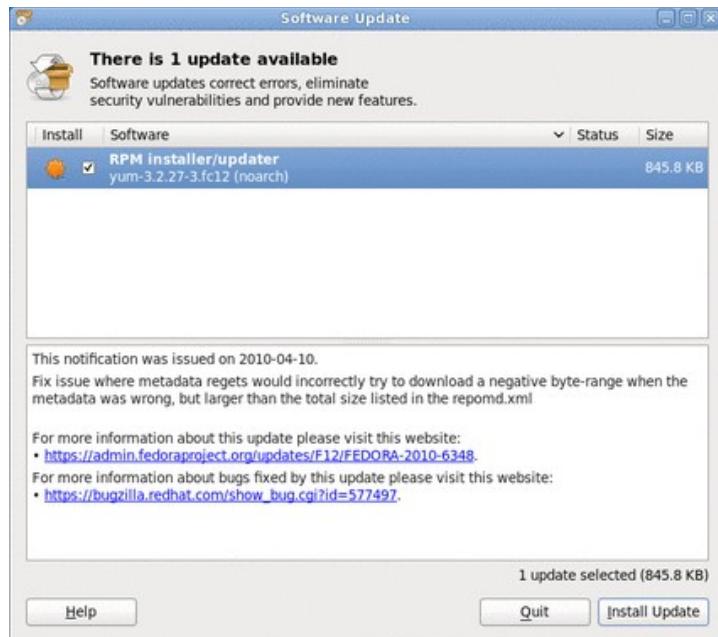
```
<?php  
    Phpinfo();  
?>
```

13. Nos metemos en el navegador y escribimos → localhost/angela.php (o el nombre que le hayas puesto.php)

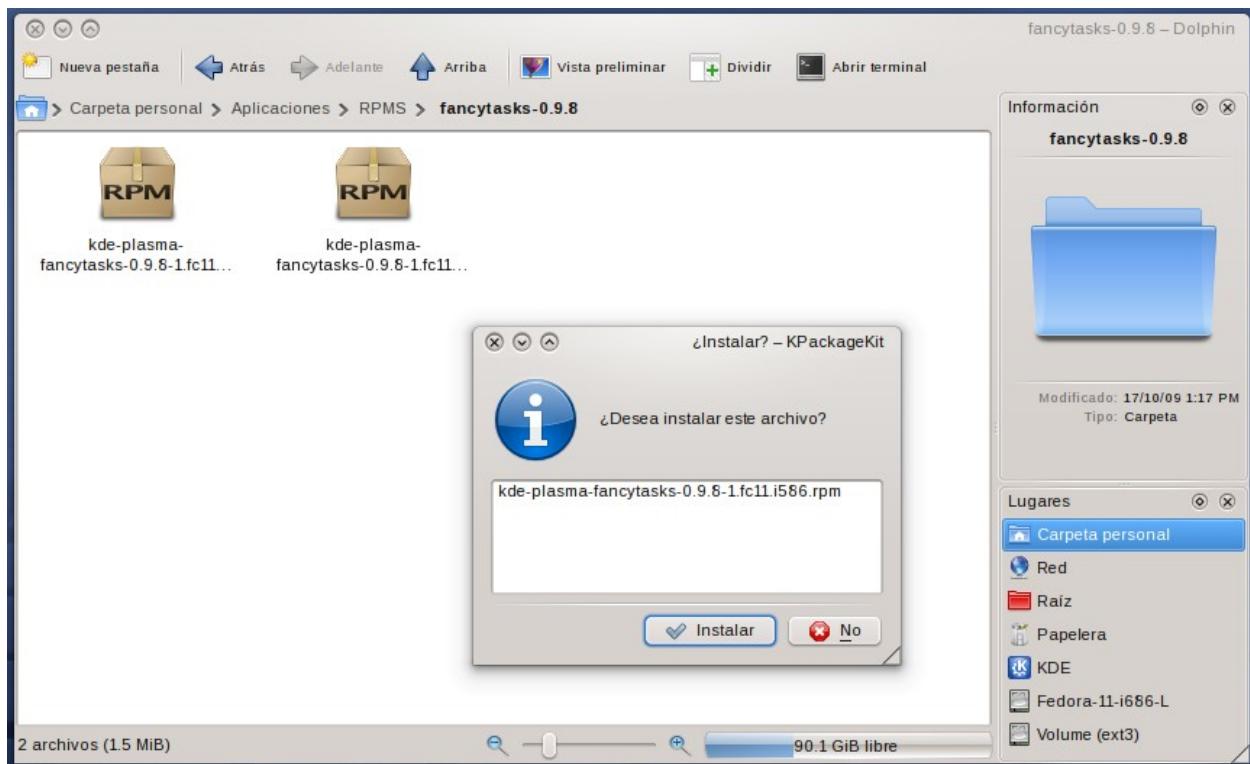


ARCHIVOS RPM:

RPM Package Manager es una herramienta de administración de paquetes pensada básicamente para GNU/Linux. Es capaz de instalar, actualizar, desinstalar, verificar y solicitar programas. RPM es el formato de paquete de partida del Linux Standard Base.



Originalmente desarrollado por Red Hat para Red Hat Linux, en la actualidad muchas distribuciones GNU/Linux lo usan, dentro de las cuales las más destacadas son Fedora, Mandriva, Mageia, PCLinuxOS, openSUSE, SuSE Linux. También se ha portado a otros sistemas operativos.



Algunos Comandos:

rpm -qa --> muestra paquetes instalados.

rpm -qi foo --> muestra la información de un paquete RPM.

rpm -ql foo --> lista ficheros de un paquete RPM instalado.

rpm -qc foo --> lista solo los ficheros de configuración.

rpm --checksig foo --> verifica firma de un paquete RPM.

rpm -ivh "localFile.rpm" --> instala un paquete.

rpm -e "localFile.rpm" --> desinstala un paquete.

```
[gacanepa@server ~]$ rpm -qa | grep htop
htop-2.0.2-1.el7.x86_64
[gacanepa@server ~]$ rpm -ql htop
/usr/bin/htop
/usr/share/doc/htop-2.0.2
/usr/share/doc/htop-2.0.2/AUTHORS
/usr/share/doc/htop-2.0.2/COPYING
/usr/share/doc/htop-2.0.2/ChangeLog
/usr/share/doc/htop-2.0.2/README
/usr/share/man/man1/htop.1.gz
/usr/share/pixmaps/htop.png
[gacanepa@server ~]$ rpm -qf /usr/share/pixmaps/htop.png
htop-2.0.2-1.el7.x86_64
[gacanepa@server ~]$ rpm -q --requires htop
libc.so.6()(64bit)
libc.so.6(GLIBC_2.14)(64bit)
libc.so.6(GLIBC_2.15)(64bit)
libc.so.6(GLIBC_2.2.5)(64bit)
libc.so.6(GLIBC_2.3)(64bit)
libc.so.6(GLIBC_2.3.4)(64bit)
libc.so.6(GLIBC_2.4)(64bit)
libgcc_s.so.1()(64bit)
libgcc_s.so.1(GCC_3.0)(64bit)
libgcc_s.so.1(GCC_3.3.1)(64bit)
libm.so.6()(64bit)
libm.so.6(GLIBC_2.2.5)(64bit)
libncursesw.so.5()(64bit)
libtinfo.so.5()(64bit)
rpmlib(CompressedFileNames) <= 3.0.4-1
rpmlib(FileDigests) <= 4.6.0-1
rpmlib(PayloadFilesHavePrefix) <= 4.0-1
rtld(GNU_HASH)
rpmlib(PayloadIsXz) <= 5.2-1
[gacanepa@server ~]$
```

Instalar servidor dns → bind9

1. apt-get install bind9

2. cd /etc/bind

primer fichero de config → named.conf

La clave de un servidor dns no es mas que crear dos ficheros de configuración o en terminos dns dos pequeñas bases de datos, donde estan en una columna nombres y en otra IP (zona directa), otro fichero donde habrá IP y nombres (zona inversa).

3. ls /etc/init.d

4. cat named.conf

5. cat named.conf options |less

6. cat named.conf.local |less → editamos el fichero con el nano (nano named.conf.local)

Zona directa:

```
zone "angela.edu" {  
    type master;  
    file "/etc/bind/db.zonadirecta"; # zone file path  
};
```

Zona inversa: (se pone la ip de la red en la que estamos-maquina virtual 2.0.10)
zone "11.168.192.in-addr.arpa" {

```
    type master;  
    file "/etc/bind/db.zonainversa"; # 10.128.0.0/16 subnet  
};
```

7. Para ver los errores → cat /var/log/syslog

8. nano db.zonadirecta

9. Escribimos dentro: (ip maquina virtual = 10.0.2.15)

```
$ORIGIN 1asir.edu.  
$TTL 86400  
@ IN SOA dns.1asir.edu. root.1asir.edu. (  
    2001062501 ; serial  
    21600       ; refresh after 6 hours  
    3600        ; retry after 1 hour  
    604800      ; expire after 1 week  
    86400 )     ; minimum TTL of 1 day  
  
    IN NS dns.1asir.edu.  
  
    IN MX 10 mail.example.com.  
  
    IN A 192.168.11.34  
gonzalo IN A 192.168.11.14
```

```
www           IN      A       192.168.11.34
dns           IN      A       192.168.11.34
dns2          IN      A       192.168.11.35
```

10. nano db.zonainversa

11. Escribimos dentro:

```
$ORIGIN 11.168.192.in-addr.arpa.
$TTL 604800
@ IN SOA 1asir.edu. root.1asir.edu. (
    1 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800 ) ; Default TTL

@ IN NS dns.1asir.edu.
14 IN PTR gonzalo.1asir.edu.
34 IN PTR www.1asir.edu.
34 IN PTR dns.1asir.edu.
35 IN PTR dns2.1asir.edu.
```

12. Para poder comprobar que cuando cargamos el servicio podemos hacer consulta en el servidor dns, editamos el fichero conf. → nano /etc/resolv.conf

Comentamos todo lo que hay dentro con # y escribimos nameserver 127.0.0.1

13. Cargamos los servicios → /etc/init.d/bind9 start (da OK)

14. apt-get install dnsutils (por si no tenemos el paquete, hace falta)

15. nslookup www.angela.com

16. Ver problemas → tail -f /var/log/syslog

DIRECTA:

```
root@usuario-VirtualBox:/etc/bind# nslookup gonzalo.angela.edu
Server:      127.0.0.1
Address:     127.0.0.1#53

Name:   gonzalo.angela.edu
Address: 192.168.11.14

root@usuario-VirtualBox:/etc/bind# nslookup www.angela.edu
Server:      127.0.0.1
Address:     127.0.0.1#53

Name:   www.angela.edu
Address: 192.168.11.34

root@usuario-VirtualBox:/etc/bind#
```

INVERSA:

```
root@usuario-VirtualBox:/etc/bind# nslookup 192.168.11.34
Server:      127.0.0.1
Address:     127.0.0.1#53

34.11.168.192.in-addr.arpa      name = dns.angela.edu.
34.11.168.192.in-addr.arpa      name = www.angela.edu.

root@usuario-VirtualBox:/etc/bind# nslookup 192.168.11.35
Server:      127.0.0.1
Address:     127.0.0.1#53

35.11.168.192.in-addr.arpa      name = dns2.angela.edu.

root@usuario-VirtualBox:/etc/bind#
```

SSH sin Password:

En este trabajo, tendremos que conseguir que el PC1 se conecte al PC2 sin tener que poner la contraseña. Para realizarlo haremos lo siguiente:

- 1) Lo primero que tenemos que hacer es activar el servicio de SSH en ambos PCs. Ponemos el siguiente comando:

✓ \$ sudo service ssh start

- 2) Para comprobar que está encendido:

✓ \$ service ssh status

```
usuario@usuario-ALDA:~$ sudo service ssh start
[sudo] password for usuario:
usuario@usuario-ALDA:~$ service ssh status
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enab
  Active: active (running) since lun 2019-02-25 08:19:53 CET; 11min ago
    Process: 4844 ExecReload=/bin/kill -HUP $MAINPID (code=exited, status=0/SUCCES
    Process: 4843 ExecReload=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
    Process: 1781 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 2067 (sshd)
     Tasks: 1
    Memory: 3.2M
      CPU: 37ms
     CGroup: /system.slice/ssh.service
             └─2067 /usr/sbin/sshd -D

feb 25 08:21:37 usuario-ALDA sshd[2067]: Server listening on 0.0.0.0 port 22.
feb 25 08:21:37 usuario-ALDA sshd[2067]: Server listening on :: port 22.
feb 25 08:21:37 usuario-ALDA systemd[1]: Reloading OpenBSD Secure Shell server.
feb 25 08:21:37 usuario-ALDA sshd[2067]: Received SIGHUP; restarting.
feb 25 08:21:37 usuario-ALDA sshd[2067]: Server listening on 0.0.0.0 port 22.
feb 25 08:21:37 usuario-ALDA sshd[2067]: Server listening on :: port 22.
feb 25 08:21:37 usuario-ALDA systemd[1]: Reloaded OpenBSD Secure Shell server.
feb 25 08:30:44 usuario-ALDA systemd[1]: Started OpenBSD Secure Shell server.
```

- 3) Después creamos la contraseña aleatoria para que nos podamos conectar al PC2 sin tener que poner la contraseña con el siguiente comando:

\$ ssh-keygen -b 4096 -t rsa

- ✓ Al poner el comando, le damos a enter tres veces y nos saldrá lo siguiente:

```
usuario@usuario-ALDA:~$ feb 25 08:31:09 usuario-ALDA systemd[1]: Started OpenBSD Secure Shell server.

usuario@usuario-ALDA:~$ ssh-keygen -b 4096 -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/usuario/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/usuario/.ssh/id_rsa.
Your public key has been saved in /home/usuario/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:Q5LY7KS+0Px847IWzr743E2hLCnI/DHv7fjuOTBwqmI usuario@usuario-ALDA
The key's randomart image is:
+---[RSA 4096]----+
| + . |
| . * . |
| +.o. |
| o ++S |
| B.=.ooo |
| ooXoB .o |
| E==O B o.. |
| ...B.+.=B. |
+---[SHA256]----+
usuario@usuario-ALDA:~$
```

- 4) Tras generar la contraseña publica, tenemos que enviársela al PC2 para que nos deje entrar en su ordenador sin tener que poner su contraseña.

```
$ sudo ssh-copy-id usuario@192.168.42.1
```

- ✓ En el nombre de usuario le tenemos que poner el nombre del usuario en el que queremos entrar, y tiene que existir en el PC2, y la IP tenemos que poner la IP del PC2.

```
usuario@usuario-ALDA:~$ sudo ssh-copy-id usuario@192.168.42.1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/usuario/.ssh/id_rsa.pub"
The authenticity of host '192.168.42.1 (192.168.42.1)' can't be established.
ECDSA key fingerprint is SHA256:JpeJYryqi7Ig94dn+eOhkLs0eJ1EuVIRh5oFb6WaiR0.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
usuario@192.168.42.1's password:

Number of key(s) added: 1

Now try logging into the machine, with:    "ssh 'usuario@192.168.42.1'"
and check to make sure that only the key(s) you wanted were added.

usuario@usuario-ALDA:~$
```

Nos pedirá la contraseña del usuario y después de esto, solo tendríamos que conectar con el comando:

```
$ ssh usuario@192.168.42.1
```

y no nos volverá a pedir contraseña para volver a conectarnos.

```
usuario@usuario-ALDA:~$ ssh usuario@192.168.42.1
The authenticity of host '192.168.42.1 (192.168.42.1)' can't be established.
ECDSA key fingerprint is SHA256:JpeJYryqi7Ig94dn+eOhkLs0eJ1EuVIRh5oFb6WaiR0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.42.1' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.15.1 x86_64)

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

113 packages can be updated.
23 updates are security updates.

New release '18.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

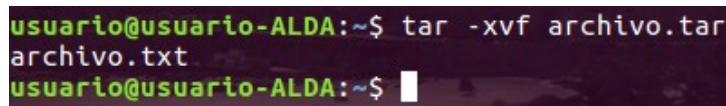
Last login: Mon Feb 25 08:36:22 2019 from 192.168.11.65
usuario@usuario-ALDA:~$ s
```

Métodos para crear y descomprimir un fichero en la terminal.

1. Primero de todo abriremos la terminal de nuestro sistema operativo.
2. Crearemos un fichero de prueba y descomprimimos el fichero.



```
usuario@usuario-ALDA:~$ touch archivo.txt
usuario@usuario-ALDA:~$ tar -cvf archivo.tar archivo.txt
archivo.txt
usuario@usuario-ALDA:~$
```



```
usuario@usuario-ALDA:~$ tar -xvf archivo.tar
archivo.txt
usuario@usuario-ALDA:~$
```

3. En este paso ya habremos acabado.

Tipos de ficheros:

✓ Ficheros gz

Para comprimir ficheros en formato .gz, utilizamos el siguiente comando:
gzip -9 fichero

Para descomprimir ficheros .gz, utilizamos el siguiente comando:
gzip -d fichero.gz

✓ Ficheros bz2

Para comprimir ficheros en formato bz2, utilizamos el siguiente comando:
bzip fichero

Para descomprimir ficheros .bz2, utilizamos el comando siguiente:
bzip2 -d fichero.bz2

✓ Ficheros tar.gz

Para comprimir ficheros en formato tar.gz, utilizamos el siguiente comando:
tar -czfv archivo.tar.gz ficheros

Para descomprimir ficheros con extensión tar.gz, utilizamos el siguiente comando:
tar -xzvf archivo.tar.gz

✓ Ficheros rar

Para comprimir en formato rar, utilizamos el siguiente comando:
rar -a archivo.rar ficheros

Para descomprimir ficheros en formato rar, utilizamos el siguiente comando:
unrar -x archivo.rar

X WINDOWS:

Para realizar las X Windows debemos descargar un sistema operativo server o bien instalar el sistema operativo sin interfaz gráfica.

1. Deberemos ejecutar el siguiente comando:

```
$ sudo apt install xorg
```

* O bien el este otro:

```
$ sudo apt install x-window-system
```

2. Ya tenemos nuestro programa de X Window instalado, para empezar a usarlo escribimos el siguiente comando:

```
$ startx
```



3. Se abre otra “interfaz” en la que por ejemplo ponemos el comando de la calculadora (xcalc) y sale su interfaz gráfica.



4. Para configurarlo a nuestro gusto debemos configurar los archivos de las X, con el siguiente comando:

```
$ nano /etc/X11/xorg.conf
```

5. Si no hay nada en el archivo de configuración realizamos los siguientes pasos:

```
$ /etc/init.d/gdm stop || /etc/init.d/gdm3 stop ||  
/etc/init.d/kdm stop || /etc/init.d/xdm stop || /etc/init.d/lightdm stop
```

```
$ cd /etc/X11/
```

```
$ Xorg -configure
```