A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

17-11-2016

# Memoria Práctica 2

Ingeniería de Servidores

Several thin, curved lines in dark blue and light grey that originate from the bottom left and sweep upwards and to the right.

Andrés Molina López  
UNIVERSIDAD DE GRANADA

## **Índice:**

1ª Cuestión .....	5
2ª Cuestión .....	6
3ª Cuestión .....	8
4ª Cuestión .....	14
5ª Cuestión .....	15
6ª Cuestión .....	16
7ª Cuestión .....	18
8ª Cuestión .....	19
Cuestión Opcional 2 .....	20
9ª Cuestión .....	22
10ª Cuestión .....	26
11ª Cuestión .....	30
12ª Cuestión .....	32
13ª Cuestión .....	36
14ª Cuestión .....	39
15ª Cuestión .....	41
16ª Cuestión .....	43
17ª Cuestión .....	44
Bibliografía .....	47

## **Índice de figuras:**

1ª Ilustración .....	5
2ª Ilustración .....	6
3ª Ilustración .....	7
4ª Ilustración .....	8
5ª Ilustración .....	8

6ª Ilustración .....	9
7ª Ilustración .....	9
8ª Ilustración .....	10
9ª Ilustración .....	11
10ª Ilustración .....	11
11ª Ilustración .....	12
12ª Ilustración .....	13
13ª Ilustración .....	13
14ª Ilustración .....	13
15ª Ilustración .....	14
16ª Ilustración .....	14
17ª Ilustración .....	15
18ª Ilustración .....	16
19ª Ilustración .....	16
20ª Ilustración .....	17
21ª Ilustración .....	17
22ª Ilustración .....	18
23ª Ilustración .....	19
24ª Ilustración .....	20
25ª Ilustración .....	20
26ª Ilustración .....	20
27ª Ilustración .....	21
28ª Ilustración .....	21
29ª Ilustración .....	21
30ª Ilustración .....	22
31ª Ilustración .....	23
32ª Ilustración .....	23

33ª Ilustración .....	24
34ª Ilustración .....	25
35ª Ilustración .....	26
36ª Ilustración .....	27
37ª Ilustración .....	27
38ª Ilustración .....	28
39ª Ilustración .....	28
40ª Ilustración .....	29
41ª Ilustración .....	30
42ª Ilustración .....	30
43ª Ilustración .....	31
44ª Ilustración .....	31
45ª Ilustración .....	31
46ª Ilustración .....	32
47ª Ilustración .....	32
48ª Ilustración .....	33
49ª Ilustración .....	33
50ª Ilustración .....	34
51ª Ilustración .....	34
52ª Ilustración .....	35
53ª Ilustración .....	35
54ª Ilustración .....	36
55ª Ilustración .....	36
56ª Ilustración .....	37
57ª Ilustración .....	37
58ª Ilustración .....	38
59ª Ilustración .....	38

60ª Ilustración .....	39
61ª Ilustración .....	40
62ª Ilustración .....	40
63ª Ilustración .....	41
64ª Ilustración .....	41
65ª Ilustración .....	42
66ª Ilustración .....	42
67ª Ilustración .....	42
68ª Ilustración .....	43
69ª Ilustración .....	43
70ª Ilustración .....	44
71ª Ilustración .....	45
72ª Ilustración .....	45
73ª Ilustración .....	46

## 1ª Cuestión:

### a) Liste los argumentos de yum necesarios para instalar, buscar y eliminar paquetes.

El argumento para instalar paquetes es *install*, de manera que la orden quedaría como *yum install nombre\_paquete* y nos pediría una confirmación para instalarlo, pero si añadimos el argumento *-y* no nos solicita la confirmación.

Para buscar paquetes usamos el argumento *search*, la cual por defecto buscará el nombre dado entre los paquetes que hay instalados, las descripciones de los paquetes, y en caso de que no encuentre coincidencias, buscará el nombre en las descripciones y URLs si se le añade *all* después de *search*. Luego mostrará el contenido encontrado por orden de coincidencias, de más coincidencias a menos. El modo de ejecución sería *yum search (all) palabras\_buscar*.

Por ejemplo, en la *Ilustración 1* muestro una búsqueda de las palabras *for* y *httpd* que aparezcan en el nombre del paquete o descripción del mismo, como se puede apreciar, deben estar incluidas ambas palabras.

```
AndMolLop 19/11/16 #yum search for httpd
Complementos cargados:fastestmirror
Loading mirror speeds from cached hostfile
 * base: sunsite.rediris.es
 * extras: ftp.cixug.es
 * updates: ftp.cixug.es
===== N/S matched: for, httpd =====
httpd-devel.x86_64 : Development interfaces for the Apache HTTP server
httpd-manual.noarch : Documentation for the Apache HTTP server
httpd-tools.x86_64 : Tools for use with the Apache HTTP Server
libmicrohttpd.i686 : Lightweight library for embedding a webserver in
                    : applications
libmicrohttpd.x86_64 : Lightweight library for embedding a webserver in
                    : applications
libmicrohttpd-devel.i686 : Development files for libmicrohttpd
libmicrohttpd-devel.x86_64 : Development files for libmicrohttpd
libmicrohttpd-doc.noarch : Documentation for libmicrohttpd
mod_auth_mellon.x86_64 : A SAML 2.0 authentication module for the Apache Httpd
                    : Server
mod_dav_svn.x86_64 : Apache httpd module for Subversion server

Nombre completo y resumen que coinciden con y sólo , use "buscar todo" para t
odo.
```

*Ilustración 1. Búsqueda con yum de los paquetes que contenga las palabras httpd y for en el nombre o en la descripción.*

Por último, en caso de querer desinstalar un paquete, el argumento sería *remove* o *erase*, los cuales se utilizan de la siguiente manera, *yum remove nombre\_paquete* o *yum erase nombre\_paquete*. Esta orden eliminará el paquete especificado del sistema y todos los paquetes que dependan del paquete eliminado. Además, también se le puede añadir *-y* para que no nos solicite una confirmación para ejecutar la orden.

("yum(8) - Linux manual page", 2016)

**b) ¿Qué ha de hacer para que yum pueda tener acceso a Internet en el PC del aula?**

Tenemos que irnos a su archivo de configuración, situado en el directorio `/etc` y se llama `yum.conf`, y especificar la información del proxy que utiliza la universidad. Para ello vamos a la variable **proxy** y escribimos el servidor como una URL completa, además de añadirle el número de puerto TCP que se utiliza. De modo que tiene que quedarnos de la siguiente manera: `proxy=http://stargate.ugr.es:3128`.

Si el proxy necesitase de usuario y contraseña simplemente habría que modifica los parámetros **proxy\_username** (poniendo el usuario del servidor proxy) y **proxy\_password** (poniendo la contraseña del servidor proxy).

En la *Ilustración 2* muestro que debería añadirse al archivo `/etc/yum.conf` para que pudiese utilizarse yum desde los ordenadores de las aulas. Como no lo he podido probar en las aulas no sé si es necesario añadir usuario o contraseña, pero en tal caso simplemente sería completar los campos con los datos correspondientes.

```
# The proxy server - proxy server:port number
proxy=http://stargate.ugr.es:3128
# The account details for yum connections
proxy_username=
proxy_password=
AndMolLop 19/11/16 #_
```

*Ilustración 2. Configuración de yum para poder usarlo con el proxy de la UGR*

("10. Usando yum con un servidor Proxy", 2016)

**c) ¿Cómo añadimos un nuevo repositorio?**

Existen dos manera de añadir repositorios, una de ellas es añadir un `[archivo].repo` en el directorio `/etc/yum.repos.d` y yum lo leerá como tal al tener extensión `.repo`. Y la otra manera es instalar **yum-utils.noarch** para así poder usar la herramienta **yum-config-manager** la cual nos permite añadir repositorios con un sencillo comando, utilizándose de la siguiente forma: `sudo yum-config-manager --add-repo url_del_repositorio`.

("8.4.5. Adding, Enabling, and Disabling a Yum Repository", 2016)

**2º Cuestión:**

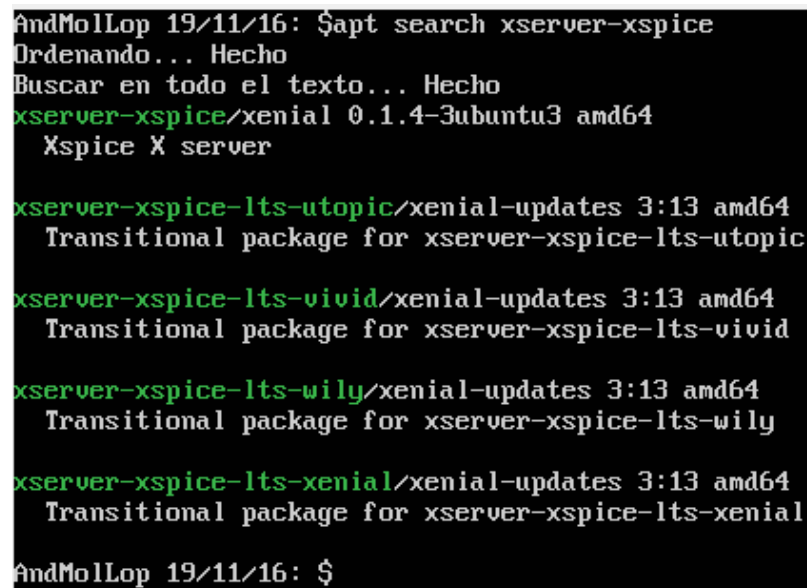
**a) Liste los argumentos de apt necesarios para instalar, buscar y eliminar paquetes.**

Para instalar un paquete con apt utilizamos el argumento `install`, quedando la orden como `sudo apt install nombre_paquete`, mientras que para eliminar un paquete utilizamos el argumento `remove`, con lo que la

orden queda como *sudo apt remove nombre\_paquete*. Estas dos órdenes puedes ser sobrecargadas añadiendo al final del nombre del paquete un + para que se instale el paquete, de modo que *sudo apt remove nombre\_paquete+* nos lo instalaría en lugar de eliminarlo, y también se puede añadir un - para hacer el efecto contrario, de manera que *sudo apt install nombre\_paquete-* nos lo desinstalaría en lugar de instalarlo.

Para buscar paquetes, el argumento a utilizar es search, el cual buscará coincidencias en los paquetes disponibles. La orden quedaría de la siguiente manera *apt search palabras*.

En la *Ilustración 3* muestro un ejemplo del uso de esta orden para buscar los paquetes que contengan la palabra xserver-xspice.



```
AndMolLop 19/11/16: $apt search xserver-xspice
Ordenando... Hecho
Buscar en todo el texto... Hecho
xserver-xspice/xenial 0.1.4-3ubuntu3 amd64
  Xspice X server

xserver-xspice-lts-utopic/xenial-updates 3:13 amd64
  Transitional package for xserver-xspice-lts-utopic

xserver-xspice-lts-vivid/xenial-updates 3:13 amd64
  Transitional package for xserver-xspice-lts-vivid

xserver-xspice-lts-wily/xenial-updates 3:13 amd64
  Transitional package for xserver-xspice-lts-wily

xserver-xspice-lts-xenial/xenial-updates 3:13 amd64
  Transitional package for xserver-xspice-lts-xenial

AndMolLop 19/11/16: $
```

*Ilustración 3. Búsqueda de los paquetes que tengan la palabra xserver-xspice*

("Ubuntu Manpage: apt - command-line interface", 2016)

## b) ¿Qué ha de hacer para que apt pueda tener acceso a Internet en el PC del aula?

Al igual que yum, tendremos que modificar el archivo de configuraciones para añadir los datos del proxy, solo que en esta ocasión para apt, ya que estamos en otra distribución Linux.

En este caso el archivo a modificar o crear en caso de que aún no exista será *apt.conf*, el cual se encuentra en */etc/apt*, de manera que accedemos al archivo dicho y añadimos los datos del proxy agregando la siguiente información:

**Acquire::http::Proxy** "[http://usuario:password@nombre\\_proxy:puerto](http://usuario:password@nombre_proxy:puerto)";



En esta ocasión, nuevamente no lo he podido probar en el ordenador de las aulas, así que no sé si requiere de autenticación, por lo que si no la requiere simplemente eliminamos esa parte.

En la *Ilustración 4* muestro como debería quedar el archivo *apt.conf* para poder hacer que apt tenga acceso a internet desde las aulas.

```
AndMolLop 19/11/16: $cat /etc/apt/apt.conf
# Datos del proxy
Acquire::http::Proxy "http://stargate.ugr.es:3128";
AndMolLop 19/11/16: $_
```

*Ilustración 4. Contenido del archivo apt.conf en /etc/apt para poder usar apt con el proxy de la UGR*

(Zamphirópolis, 2016)

### c) ¿Cómo añadimos un nuevo repositorio?

La forma más práctica para añadir un nuevo repositorio es usar el comando `add-apt-repository ppa: nombre_repositorio`. Aún para usarla debemos ser administradores del sistema, de modo que la orden quedaría de la siguiente manera: `sudo add-apt-repository ppa: nombre_repositorio`.

Si no, también se puede ir al archivo */etc/apt/sources.list* y editarlo de tal manera que añadamos al final del archivo los nuevos repositorios deseados. Para editarlo también hay que hacerlo como administrador.

("Añadir repositorios externos - Guía Ubuntu", 2016)

## 3º Cuestión:

### a) ¿Con qué comando puede abrir/cerrar un puerto usando ufw?

Para manejar ufw hay que tener derechos de administrador, de modo que una vez estemos en modo sudo, abrir o cerrar un puerto con ufw es bastante sencillo. Lo primero que hay que hacer es habilitar ufw, para ello vamos a ejecutar `ufw enable`, podemos comprobar si se habilitado ejecutando `ufw status`. Muestro este proceso y resultado del mismo en la *Ilustración 5*.

```
AndMolLop 19/11/16: $ufw status
Estado: inactivo
AndMolLop 19/11/16: $ufw enable
El cortafuegos está activo y habilitado en el arranque del sistema
AndMolLop 19/11/16: $ufw status
Estado: activo
AndMolLop 19/11/16: $
```

*Ilustración 5. Activación de ufw*

Una vez activado, ya solo tenemos que usar los argumentos *allow* y *deny* mas el número de puerto, para abrirlos o cerrarlos respectivamente.

Además, detrás del número de puerto podemos poner /tcp o /udp para especificar para que tipo de protocolo está abierto o cerrado. En caso de no especificar ninguno de los dos, ufw asume que la operación se realiza para ambos protocolos.

En la *Ilustración 6* muestro varios ejemplos de cómo abrir y cerrar puertos con diferentes configuraciones posibles, y además, muestro el status en el cual se pueden ver las reglas definidas anteriormente, que se van a aplicar a la iptables para abrir o cerrar los puertos correspondientes.

```

AndMolLop 19/11/16: $ufw allow 53
Regla añadida
Regla añadida (v6)
AndMolLop 19/11/16: $ufw allow 54/tcp
Regla añadida
Regla añadida (v6)
AndMolLop 19/11/16: $ufw deny 80
Regla añadida
Regla añadida (v6)
AndMolLop 19/11/16: $ufw deny 81/udp
Regla añadida
Regla añadida (v6)
AndMolLop 19/11/16: $ufw status
Estado: activo

Hasta          Acción          Desde
-----
53              ALLOW           Anywhere
54/tcp          ALLOW           Anywhere
80              DENY            Anywhere
81/udp          DENY            Anywhere
53 (v6)         ALLOW           Anywhere (v6)
54/tcp (v6)     ALLOW           Anywhere (v6)
80 (v6)         DENY            Anywhere (v6)
81/udp (v6)     DENY            Anywhere (v6)

AndMolLop 19/11/16: $

```

*Ilustración 6. Definición de reglas para abrir o cerrar puertos y visualización de las mismas*

Las reglas definidas se pueden eliminar añadiendo *delete* justamente detrás de ufw, en la regla original usada para abrir o cerrar el puerto correspondiente. En la *Ilustración 7*, se ve más claramente lo aquí mencionado, en la cual elimino las reglas que abren el puerto 54 para tcp y cierran el puerto 81 para udp, y luego vuelvo a mostrar el status.

```

AndMolLop 19/11/16: $ufw delete allow 54/tcp
Regla eliminada
Regla eliminada (v6)
AndMolLop 19/11/16: $ufw delete deny 81/udp
Regla eliminada
Regla eliminada (v6)
AndMolLop 19/11/16: $ufw status
Estado: activo

Hasta          Acción          Desde
-----
53              ALLOW           Anywhere
80              DENY            Anywhere
53 (v6)         ALLOW           Anywhere (v6)
80 (v6)         DENY            Anywhere (v6)

AndMolLop 19/11/16: $

```

*Ilustración 7. Eliminación de reglas relacionadas con los puertos 54 y 81*

("UFW - Community Help Wiki", 2016)

**b) ¿Con qué comando puede abrir/cerrar un puerto usando firewall-cmd en CentOS?**

Lo primero que vamos a hacer es comprobar si firewall-cmd está funcionando con `firewall-cmd --state`, y en caso de que no lo esté, lo activamos con `systemctl start firewalld.service`.

Una vez en funcionamiento, vamos a ver en qué zona estamos que seguramente será la public ya que es la que viene por defecto. Para ello vamos a ejecutar `firewall-cmd --get-active-zones`, además, ejecutando `firewall-cmd --list-all` también podemos saber que reglas tiene definidas ya la zona.

Todos estos pasos iniciales los muestro en la *Ilustración 8*, para que se vea mi configuración de reglas inicial de la zona public.

```
AndMolLop 20/11/16 #firewall-cmd --state
not running
AndMolLop 20/11/16 #systemctl start firewalld.service
AndMolLop 20/11/16 #firewall-cmd --state
running
AndMolLop 20/11/16 #firewall-cmd --get-active-zones
public
  interfaces: enp0s3
AndMolLop 20/11/16 #firewall-cmd --list-all
public (default, active)
  interfaces: enp0s3
  sources:
  services: dhcpv6-client ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:

AndMolLop 20/11/16 #_
```

*Ilustración 8. Inicio del servicio firewalld y visualización de la configuración inicial de la zona public*

Una vez que sabemos la configuración inicial, podemos proceder a abrir puertos, para ello vamos a usar el argumento `--zone=zona_deseada --add-port=numero/tcp o udp`. De esta manera, definiremos una regla para que cuando esté activa la zona deseada (si no se ha especificado zona al ejecutar la orden, el servicio asume que es a la zona por defecto), el número de puerto que hayamos puesto (también se puede poner un rango de puertos usando un guion entre ambos números, lo cual hará que la orden afecte a todos los puertos comprendidos en el rango de valores), esté abierto para el protocolo indicado. Si queremos que esta regla se quede definida de forma permanente y así no se borre al reiniciar, tendremos que añadir el argumento `--permanent` justo después de especificar la zona.

En la *Ilustración 9* muestro un ejemplo de apertura del puerto 82 para tcp y del 83 para udp, ambos para la zona public, ya que es la por defecto.

Además, voy a usar el argumento `--list-ports` para comprobar que los puertos se han añadido a la zona correctamente.

```
AndMolLop 20/11/16 #firewall-cmd --add-port=82/tcp
success
AndMolLop 20/11/16 #firewall-cmd --add-port=83/udp
success
AndMolLop 20/11/16 #firewall-cmd --list-ports
83/udp 82/tcp
AndMolLop 20/11/16 #firewall-cmd --list-all
public (default, active)
  interfaces: enp0s3
  sources:
  services: dhcpv6-client ssh
  ports: 83/udp 82/tcp
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:

AndMolLop 20/11/16 #_
```

*Ilustración 9. Apertura de puertos para la zona public y comprobación de la configuración de la zona*

Ahora, para cerrar los puertos, es decir, eliminar las reglas definidas anteriormente, utilizamos el argumento `--remove-port=numero/protocolo`. Nuevamente, si no se especifica la zona, esta orden se ejecutará para la zona por defecto.

En la *Ilustración 10* muestro la eliminación de las reglas definidas en la *Ilustración 9*, usando el argumento mencionado anteriormente.

```
AndMolLop 20/11/16 #firewall-cmd --remove-port=82/tcp
success
AndMolLop 20/11/16 #firewall-cmd --remove-port=83/udp
success
AndMolLop 20/11/16 #firewall-cmd --list-ports
AndMolLop 20/11/16 #firewall-cmd --list-all
public (default, active)
  interfaces: enp0s3
  sources:
  services: dhcpv6-client ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:

AndMolLop 20/11/16 #_
```

*Ilustración 10. Eliminación de las reglas que abrían los puertos 82 para tcp y 83 para udp en la zona public*

Por último, apuntar, que tal y como yo he ejecutado los comandos, si se reiniciase el servicio de firewalld con el comando `systemctl restart firewalld.service` o con `firewall-cmd --reload`, o `--complete-reload`, las

reglas no se quedarían guardadas, de manera que no importaría que no hubiese borrado las reglas, ya que al reiniciarse se perderían automáticamente. Por lo que, si queremos que esto no pase, es necesario añadir el argumento *--permanent*, lo cual es recomendable, ya que muchas veces para que la configuración se modifique en las iptables, es necesario reiniciar el servicio.

(Woerner, 2016)

("How To Set Up a Firewall Using FirewallD on CentOS 7 | DigitalOcean", 2016)

("3.8.13.5.7. Open Ports in the Firewall using the CLI", 2016)

### c) Utilice el comando nmap para ver que efectivamente, los puertos están accesibles

Para poder comprobar que los puertos están accesibles, hay que usar nmap desde otra máquina, y debido a que VirtualBox le había asignado la misma dirección IP a mi máquina virtual con el servidor de Ubuntu y a la del servidor de CentOS, he tenido que crear una red Nat y meter ambas máquinas en la red, para que las direcciones IP cambiasen.

De tal manera que la máquina con CentOS tiene la dirección IP 10.0.2.15, y la de Ubuntu tiene la dirección IP 10.0.2.5.

Una vez, sabiendo la IP de ambas, y haciendo ping de la una a la otra, para ver que efectivamente pueden comunicarse, lo primero que compruebo es que en CentOS borre las reglas que abrían los puertos, y con nmap compruebo desde sí misma cuales tengo abiertos, esto lo muestro en la *Ilustración 11*.

```
AndMolLop 23/11/16 #firewall-cmd --list-all
public (default, active)
  interfaces: enp0s3
  sources:
  services: dhcpv6-client ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:

AndMolLop 23/11/16 #nmap localhost

Starting Nmap 6.40 ( http://nmap.org ) at 2016-11-23 01:38 CET
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000050s latency).
Other addresses for localhost (not scanned): 127.0.0.1
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
631/tcp   open  ipp

Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds
AndMolLop 23/11/16 #
```

*Ilustración 11. Visualización de puertos abiertos en CentOS desde sí misma*

Con Ubuntu elimino todas las reglas que tengo y también compruebo con nmap que puertos tiene abiertos. Como podemos ver en la *Ilustración 12*, al borrar las reglas, no hay ningún puerto abierto, según nos dice nmap.

```

AndMolLop 23/11/16: $ufw status
Estado: activo
AndMolLop 23/11/16: $nmap localhost

Starting Nmap 7.01 ( https://nmap.org ) at 2016-11-23 01:42 CET
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000090s latency).
Other addresses for localhost (not scanned): ::1
All 1000 scanned ports on localhost (127.0.0.1) are closed

Nmap done: 1 IP address (1 host up) scanned in 1.64 seconds
AndMolLop 23/11/16: $

```

*Ilustración 12. Visualización de puertos abiertos en Ubuntu desde sí misma*

Ahora que sé que puertos hay abiertos en cada una, voy a declarar la regla necesaria para abrir el puerto 8080 para tcp en Ubuntu, tal y como muestro en la *Ilustración 13*. Y en CentOS voy a declarar la regla para abrir el puerto 9050 para tcp, como muestro en la *Ilustración 14*.

```

AndMolLop 23/11/16: $ufw allow 8080/tcp
Regla añadida
Regla añadida (v6)
AndMolLop 23/11/16: $ufw status
Estado: activo

Hasta          Acción          Desde
-----
8080/tcp       ALLOW          Anywhere
8080/tcp (v6)  ALLOW          Anywhere (v6)

AndMolLop 23/11/16: $

```

*Ilustración 13. Apertura de puerto en Ubuntu*

```

AndMolLop 23/11/16 #firewall-cmd --permanent --add-port=9050/tcp
success
AndMolLop 23/11/16 #systemctl restart firewalld.service
AndMolLop 23/11/16 #firewall-cmd --list-all
public (default, active)
  interfaces: enp0s3
  sources:
  services: dhcpv6-client ssh
  ports: 9050/tcp
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:

AndMolLop 23/11/16 #_

```

*Ilustración 14. Apertura de puerto en CentOS*

Ya abiertos los puertos, lo que resta es comprobarlo con nmap, pero para ello hay que comprobarlo de una máquina a otra, por lo que desde CentOS voy a ejecutar *nmap 10.0.2.5* para ver los puertos abiertos en Ubuntu, esto lo muestro en la *Ilustración 15*. Y desde Ubuntu voy a ejecutar *nmap 10.0.2.15* para ver los puertos abiertos en CentOS, esto lo vemos en la *Ilustración 16*.

```

AndMolLop 23/11/16 #nmap 10.0.2.5
Starting Nmap 6.40 ( http://nmap.org ) at 2016-11-23 02:10 CET
Nmap scan report for 10.0.2.5
Host is up (0.00031s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE
8080/tcp   closed http-proxy
MAC Address: 08:00:27:C3:52:47 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 4.95 seconds
AndMolLop 23/11/16 #_

```

*Ilustración 15. Visualización de puertos abiertos en Ubuntu desde CentOS*

```

AndMolLop 23/11/16: $nmap 10.0.2.15
Starting Nmap 7.01 ( https://nmap.org ) at 2016-11-23 02:11 CET
Nmap scan report for 10.0.2.15
Host is up (0.00032s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
9050/tcp   closed tor-socks
MAC Address: 08:00:27:22:21:3C (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 8.24 seconds
AndMolLop 23/11/16: $_

```

*Ilustración 16. Visualización de puertos abiertos en CentOS desde Ubuntu*

Como podemos ver, los puertos mencionados anteriormente, han aparecido en la captura de nmap correspondiente, y aunque ponga closed no significa que estén cerrados, si no que no están “escuchando” de ningún servicio. Si estuviesen cerrados, no saldrían en el nmap directamente.

“A closed port is accessible (it receives and responds to Nmap probe packets), but there is no application listening on it.” - ("Port Scanning Basics", 2016)

#### **4º Cuestión: ¿Qué diferencia hay entre telnet y ssh?**

La principal diferencia entre telnet y ssh es que en telnet las comunicaciones (usuarios, claves, mensaje) se pasan en texto plano, por lo que cualquiera que haga sniffing puede ver todo el contenido sin dificultad, mientras que en ssh las comunicaciones van cifradas, lo que hace que tengan una mayor seguridad.

Dejo una referencia que confirma lo dicho, pero la fuente de información ha sido el profesor Jorge Navarro de Fundamentos de Redes.

(Alvarez, 2016)

## 5ª Cuestión:

### **a) ¿Para qué sirve la opción -X?**

Sirve para habilitar *X11 forwarding*, lo cual nos permite ejecutar aplicaciones gráficas de manera remota.

("OpenBSD manual pages", 2016)

### **b) Ejecute remotamente, es decir, desde la máquina anfitriona (si tiene Linux) o desde la otra máquina virtual, el comando gedit en una sesión abierta con ssh ¿Qué ocurre?**

En mi caso la conexión la hago desde la otra máquina virtual con CentOS, y lo que ocurre es que si no has abierto la conexión usando el opción -X salta un error y no se arranca, mientras que si sí usaste -X al abrir la conexión pueden pasar dos cosas, dependiendo de si en CentOS tengo activada la GUI con *startx* o no.

En caso de que no la tenga activada, salta el error que vemos en la *Ilustración 17*. Y por el contrario si sí tengo activada la GUI, entonces sí que funciona gedit sin ningún problema, tal y como vemos en la *Ilustración 18*.

```
[AndMolLop 23/11/16 ~]#ssh -X andres@10.0.2.5
andres@10.0.2.5's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-31-generic x86_64)

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

Pueden actualizarse 10 paquetes.
7 actualizaciones son de seguridad.

Last login: Wed Nov 23 14:01:58 2016 from 10.0.2.15
AndMolLop 23/11/16:~$ gedit Hola.txt
Failed to connect to Mir: Failed to connect to server socket: No existe el archivo o el directorio
Unable to init server: Could not connect: Conexión rehusada

(gedit:6329): Gtk-WARNING **: cannot open display:
AndMolLop 23/11/16:~$ _
```

*Ilustración 17. Fallo al abrir gedit sin GUI con conexión ssh -X desde CentOS a Ubuntu*



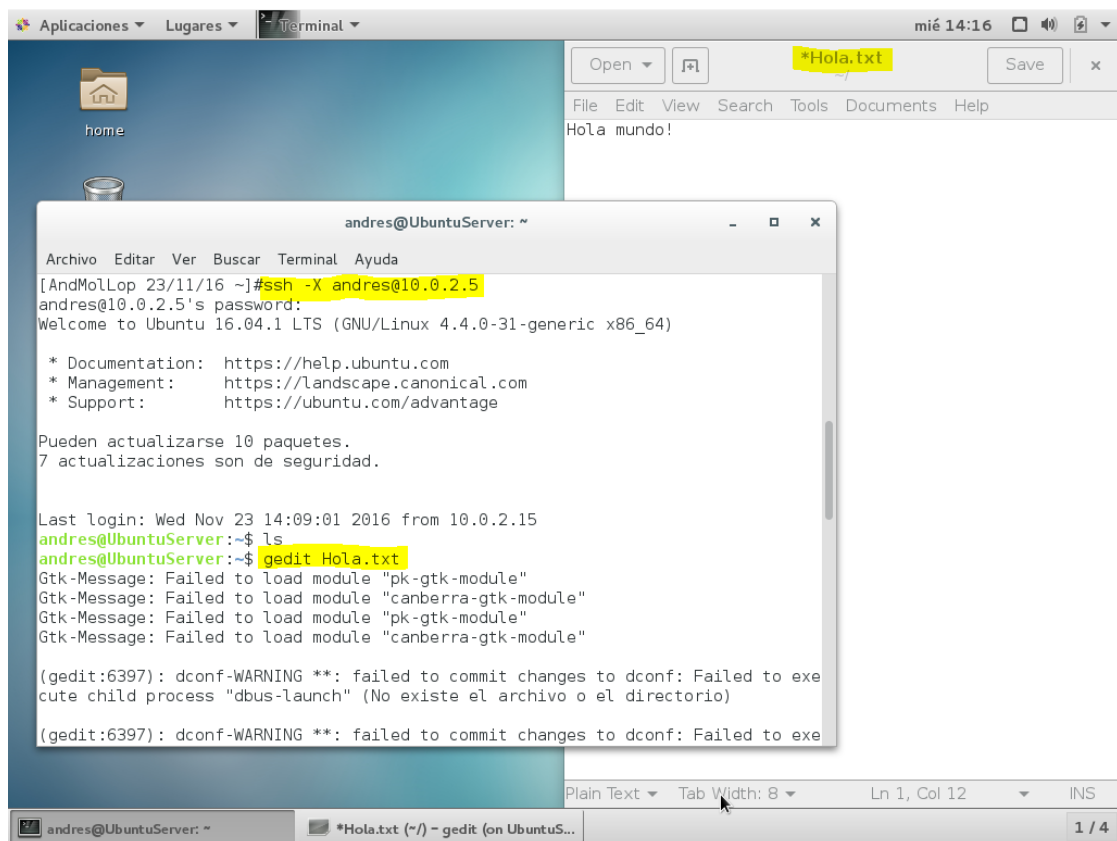


Ilustración 18. Apertura de gedit con GUI y conexión ssh -X desde CentOS a Ubuntu

Además, una vez guardado el archivo, desde el propio Ubuntu, podemos ver que este ha sido creado y guardado con éxito, haciendo `cat Hola.txt`, como muestro en *Ilustración 19*.

```
AndMolLop 23/11/16:/home/andres #ls
Hola.txt
AndMolLop 23/11/16:/home/andres #cat Hola.txt
Hola mundo!
AndMolLop 23/11/16:/home/andres #
```

Ilustración 19. Visualización del archivo Hola.txt creado y editado remotamente

## **6ª Cuestión: muestre la secuencia de comandos y las modificaciones a los archivos correspondientes para permitir acceder a la consola remota sin introducir la contraseña. Pruebe que funciona.**

Lo primero que he hecho ha sido crear en CentOS otro usuario, ya que solo tenía la cuenta de root, el cual se llama andresCentOS.

De aquí en adelante empiezo con la configuración para acceder al usuario recién creado en CentOS, desde mi máquina virtual con Ubuntu Server. Para ello, lo primero que vamos a hacer es ejecutar el comando `ssh-keygen -t rsa` para que nos genere un par

de claves (una pública y otra privada) con cifrado rsa. Esto lo muestro en la *Ilustración 20*.

```
AndMolLop 24/11/16:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/andres/.ssh/id_rsa): Pulsamos intro
Created directory '/home/andres/.ssh'.
Enter passphrase (empty for no passphrase): Pulsamos intro
Enter same passphrase again: Pulsamos intro
Your identification has been saved in /home/andres/.ssh/id_rsa.
Your public key has been saved in /home/andres/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:c53RzBEF10mNNWJQWUY3KxFBs9EMAZ1uPpWPKkZEQtM andres@UbuntuServer
The key's randomart image is:
+----[RSA 2048]-----+
|      .+=B:/X=|
|      oEXO=|
|      .++=o|
|      ..=+o|
|      S ..=...|
|      o +o. .|
|      . . .|
|      . . .|
|      . . .|
+----[SHA256]-----+
AndMolLop 24/11/16:~$ _
```

Se puede ver que estoy en la máquina virtual con Ubuntu Server

*Ilustración 20. Uso de ssh-keygen en Ubuntu Server desde la cuenta de usuario "andres"*

Luego con `ssh-copy-id -i` copiamos la clave pública (terminación .pub) al servidor remoto en el usuario deseado, en mi caso la voy a copiar en andresCentOS@10.0.2.15. Para ello hago lo que se muestra en la *Ilustración 21*.

```
AndMolLop 24/11/16:~$ ssh-copy-id -i ~/.ssh/id_rsa.pub andresCentOS@10.0.2.15
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/andres/.ssh/id_rsa.pub"
The authenticity of host '10.0.2.15 (10.0.2.15)' can't be established.
ECDSA key fingerprint is SHA256:mx1cRh5CCbQI/Pu101lu0.jp+LRD1Bi8SCUT9M4c774E.
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 prompted now it is to install all the new keys
andresCentOS@10.0.2.15's password: Introducimos la contraseña de andresCentOS

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'andresCentOS@10.0.2.15'"
and check to make sure that only the key(s) you wanted were added.

AndMolLop 24/11/16:~$ _
```

*Ilustración 21. Copia de la contraseña pública desde andres@UbuntuServer a andresCentOS@10.0.2.15*

Con la opción `-i` indicamos el directorio donde queremos que se copie, y si este no existe en el usuario en el servidor remoto, lo crea.

Para finalizar probamos a conectarnos a andresCentOS desde andres@UbuntuServer, tal y como dice al final de la *Ilustración 21*, para así verificar que sí que nos podemos conectar sin que nos pida contraseña. Y como vemos en la *Ilustración 22* el proceso se ha llevado acabo satisfactoriamente y puedo conectarme sin necesidad de poner ninguna contraseña.

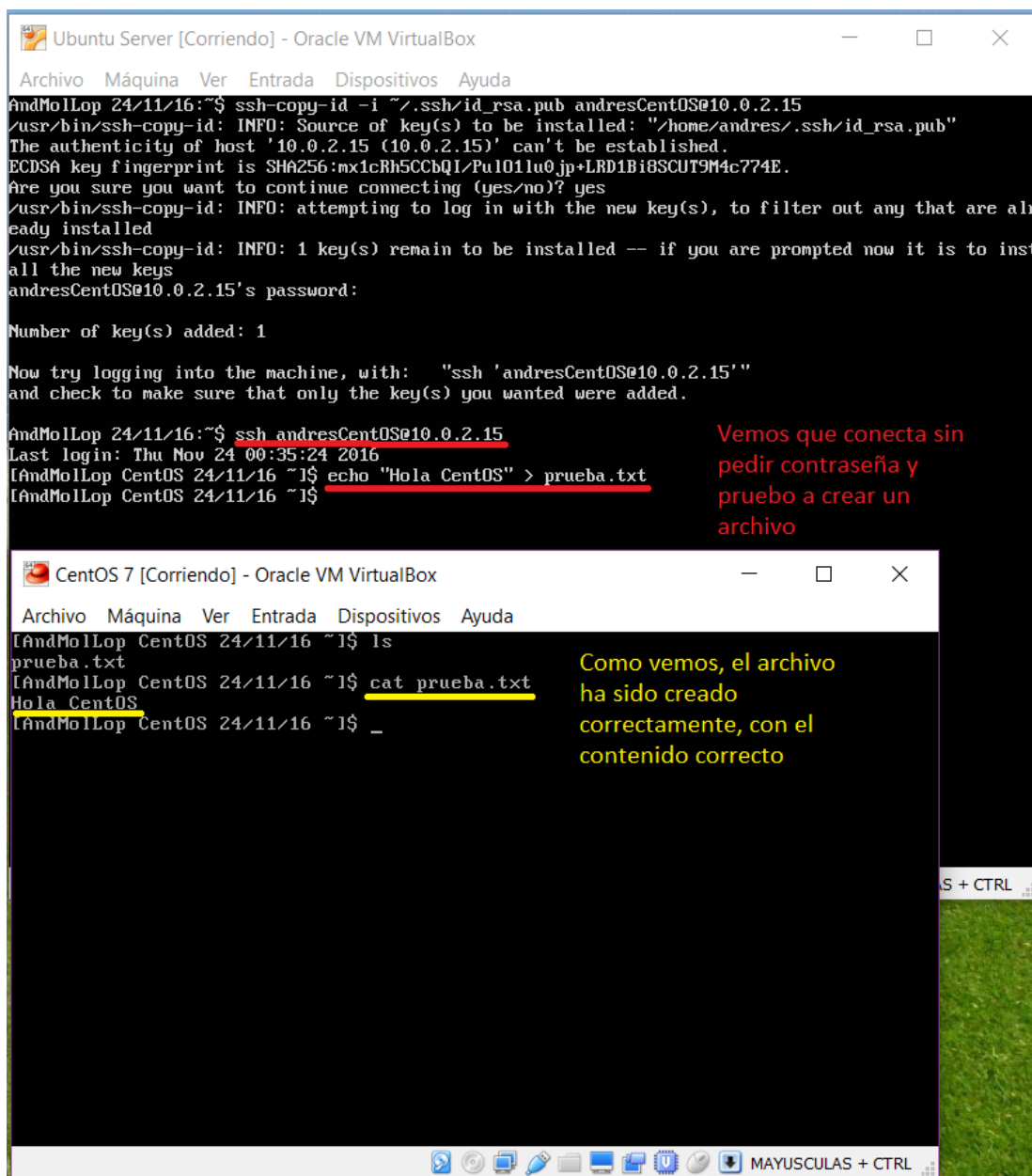


Ilustración 22. Conexión ssh sin solicitud de contraseña y prueba de correcto funcionamiento creando documento

(ssh-copy-id, 2016)

## **7ª Cuestión: ¿Qué archivo es el que contiene la configuración del servicio ssh? ¿Qué parámetro hay que modificar para evitar que el usuario root acceda? Cambie el puerto por defecto y compruebe que puede acceder.**

El archivo que contiene la configuración es `sshd_config` en el directorio `/etc/ssh/`.

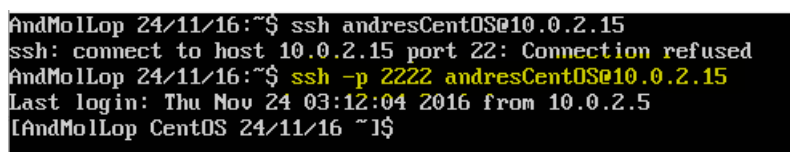
Para que no se pueda acceder al servidor con la cuenta de root lo que tenemos que hacer es buscar en el archivo anteriormente mencionado una línea que pone `#PermitRootLogin yes`, descomentarla y cambiar `yes` a `no`.

Por último, para cambiar el puerto, simplemente en el archivo de configuración *sshd\_config* buscamos una de las primeras líneas en la cual pone *port 22* (puede estar comentado o no, en caso de estarlo hay que descomentar la línea) y cambiamos el número por el que queramos (es recomendable poner uno que no esté reservado).

Tras esto, abrimos el puerto seleccionado en el firewall, y para finalizar reiniciamos el servicio de firewall y de ssh para que se realicen los cambios.

Yo para CentOS también he tenido que ejecutar *semanage port -a -t ssh\_port\_t -p tcp 2222*, donde 2222 es el número de puerto que he seleccionado para ssh. Para Ubuntu esto no ha sido necesario.

En la *Ilustración 23* muestro como se tendría que hacer la conexión ahora que el puerto ha sido cambiado.



```
AndMolLop 24/11/16:~$ ssh andresCentOS@10.0.2.15
ssh: connect to host 10.0.2.15 port 22: Connection refused
AndMolLop 24/11/16:~$ ssh -p 2222 andresCentOS@10.0.2.15
Last login: Thu Nov 24 03:12:04 2016 from 10.0.2.5
[AndMolLop CentOS 24/11/16 ~]$
```

*Ilustración 23. Conexión ssh por el puerto 2222*

Como vemos en la *Ilustración 23*, hay que añadir la opción *-p* seguida del número de puerto por el cual se debe hacer la conexión, para poder así acceder a la máquina de manera remota, si no, fallará.

("Configura un servidor SSH en Ubuntu para acceder a tu equipo de forma remota", 2016)

("Cambiar puerto de SSH en CentOS 7 - Blog de ADW.es", 2016)

### **8ª Cuestión: Indique si es necesario reiniciar el servicio ¿Cómo se reinicia un servicio en Ubuntu? ¿y en CentOS? Muestre la secuencia de comandos para hacerlo.**

Para que las configuraciones modificadas se hagan válidas, es necesario reiniciar el servicio de ssh, y tenemos varias maneras de hacerlo dependiendo del sistema operativo.

Para reiniciarlo en Ubuntu, podemos usar los siguientes comandos:

- `/etc/init.d/ssh restart`
- `service ssh restart`
- `systemctl restart ssh`

En la *Ilustración 24* muestro que cualquiera de estas tres formas reinicia el servicio.

```

AndMolLop 24/11/16:/home/andres#etc/init.d/ssh restart
[ ok ] Restarting ssh (via systemctl): ssh.service.
AndMolLop 24/11/16:/home/andres#service ssh restart
AndMolLop 24/11/16:/home/andres#systemctl restart ssh
AndMolLop 24/11/16:/home/andres#_

```

Ilustración 24. Reinicio del servicio ssh en Ubuntu

Para reiniciarlo en CentOS, podemos usar los siguientes comandos:

- `service sshd restart`
- `systemctl restart sshd(.service)` es irrelevante ponerlo

En la *Ilustración 25* muestro que ambas formas reinician el servicio.

```

[AndMolLop 24/11/16 ~]#service sshd restart
Redirecting to /bin/systemctl restart sshd.service
[AndMolLop 24/11/16 ~]#systemctl restart sshd
[AndMolLop 24/11/16 ~]#systemctl restart sshd.service
[AndMolLop 24/11/16 ~]#_

```

Ilustración 25. Reinicio del servicio ssh en CentOS

("HowTo: Restart SSH Service under Linux / UNIX", 2016)

## 2º Cuestión opcional: Instale el servicio (fail2ban) y pruebe su funcionamiento.

Lo primero, es instalarlo, yo lo voy a hacer en Ubuntu, por lo que voy a ejecutar `apt install fail2ban`. Una vez instalado, vamos a copiar su archivo de configuraciones, que es `jail.conf` y vamos llamar a la copia `jail.local`, vamos a dejar ambos situados en `/etc/fail2ban` que es donde se encuentra el origina. Ahora en `jail.local` vamos a cambiar las configuraciones que nos interesan, entre los cuales se encuentran `bantime` (tiempo que el host que ha intentado acceder va a estar baneado), `maxretry` (número de fallos al conectarse que va a tener el host antes de ser baneado) y `findtime` (tiempo en el cual se tienen que cometer el número de fallos en `maxretry` para ser baneado). En la *Ilustración 26* vemos que yo he reducido `bantime` a 120 segundos, que es el número por defecto, pero he reducido `findtime` a 60 segundos y `maxretry` a 3 intentos.

```

GNU nano 2.5.3      Archivo: /etc/fail2ban/jail.local

# External command that will take an tagged arguments to ignore, e.g. <ip>,
# and return true if the IP is to be ignored. False otherwise.
#
# ignorecommand = /path/to/command <ip>
ignorecommand =

# "bantime" is the number of seconds that a host is banned.
bantime = 120

# A host is banned if it has generated "maxretry" during the last "findtime"
# seconds.
findtime = 60

# "maxretry" is the number of failures before a host get banned.
maxretry = 3

```

Ilustración 26. Configuración de `bantime`, `findtime` y `maxretry` en `jail.local`

Lo siguiente que nos interesa es buscar la “cárcel” de sshd, y si hemos cambiado el puerto de conexiones o el sitio donde están los logins, se lo modificamos. En la *Ilustración 27* mostramos como queda mi “cárcel” de sshd, con el puerto cambiado a 2222, ya que es el que yo estoy usando.

```
#
# JAILS
#
#
#
# SSH servers
#
[sshd]
port      = 2222
logpath   = %(sshd_log)s
```

*Ilustración 27. Configuración ssh en jail.local*

Y, por último, iniciamos el servicio con `service fail2ban start` y probamos la conexión desde CentOS hacia Ubuntu, fallando a propósito, para ver si está funcionando correctamente y nos banea. Como vemos en la *Ilustración 28*, intento conectarme desde CentOS, hasta que llega un momento en el que directamente deniega la conexión directamente.

```
[AndMolLop 24/11/16 ~]#ssh -p 2222 andres@10.0.2.5
andres@10.0.2.5's password:
Permission denied, please try again.
andres@10.0.2.5's password:
Permission denied, please try again.
andres@10.0.2.5's password:
Permission denied (publickey,password).
[AndMolLop 24/11/16 ~]#ssh -p 2222 andres@10.0.2.5
andres@10.0.2.5's password:
Permission denied, please try again.
andres@10.0.2.5's password:

[AndMolLop 24/11/16 ~]#ssh -p 2222 andres@10.0.2.5
ssh: connect to host 10.0.2.5 port 2222: Connection refused
[AndMolLop 24/11/16 ~]#_
```

*Ilustración 28. Conexión ssh baneada*

En ese momento, voy a Ubuntu y miro el archivo de logs de fail2ban, el cual es `fail2ban.log`, y está definido en `/var/log`. Y como se puede ver en la *Ilustración 29*, fail2ban han encontrado 3 conexiones desde la IP 10.0.2.15, y al ser fallidas, la ha baneado, luego encuentra otra, pero ya no la deja entrar directamente, lo cual se puede apreciar en la *Ilustración 28*. Y como está definido en `jail.local` tras dos minutos de baneo, le elimina la restricción a la IP.

```
2016-11-24 15:29:37,440 fail2ban.jail      [31701: INFO   Jail 'sshd' started
2016-11-24 15:33:02,118 fail2ban.filter  [31701: INFO   [sshd] Found 10.0.2.15
2016-11-24 15:33:04,080 fail2ban.filter  [31701: INFO   [sshd] Found 10.0.2.15
2016-11-24 15:33:36,499 fail2ban.filter  [31701: INFO   [sshd] Found 10.0.2.15
2016-11-24 15:33:36,885 fail2ban.actions [31701: NOTICE [sshd] Ban 10.0.2.15
2016-11-24 15:33:38,462 fail2ban.filter  [31701: INFO   [sshd] Found 10.0.2.15
2016-11-24 15:35:37,262 fail2ban.actions [31701: NOTICE [sshd] Unban 10.0.2.15
AndMolLop 24/11/16:~/home/andres#_
```

*Ilustración 29. Archivo `/var/log/fail2ban.log` en el que se ve que IPs están baneadas y para qué servicio*

Y al eliminarse el baneo, podemos volver a conectarnos desde CentOS a Ubuntu mediante ssh, tal y como vemos en la *Ilustración 30*.

```
[AndMolLop 24/11/16 ~]#ssh -p 2222 andres@10.0.2.5
andres@10.0.2.5's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-31-generic x86_64)

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

Pueden actualizarse 10 paquetes.
7 actualizaciones son de seguridad.

Last login: Thu Nov 24 15:23:27 2016 from 10.0.2.15
AndMolLop 24/11/16:~$ time

real    0m0.000s
user    0m0.000s
sys     0m0.000s
AndMolLop 24/11/16:~$ date +%H:%M:%S
15:57:03
AndMolLop 24/11/16:~$ _
```

*Ilustración 30. Conexión establecida por ssh una vez eliminado el baneo*

("HOWTO fail2ban spanish - Fail2ban", 2016)

### **9ª Cuestión: Muestre los comandos que ha utilizado en Ubuntu Server y en CentOS. Compruebe que la instalación ha sido correcta.**

Primero voy a mostrar cómo se lleva a cabo la instalación en Ubuntu (con IP 10.0.2.8) y para ello voy a seguir los siguientes pasos:

- 1º) Instalamos apache2, para ello ejecutamos *apt-get install apache2*.
- 2º) Una vez instalado vamos a añadir en el archivo de configuración *apache2.conf*, situado en */etc/apache2*, una línea que ponga *ServerName nuestra\_IP*, para así eliminar los errores de sintaxis.
- 3º) Reiniciamos apache2 con *systemctl restart apache2* para que se hagan efectivos los cambios.
- 4º) Añadimos apache2 al cortafuegos, para ello hacemos *ufw allow in "Apache Full"* para que habrá los puertos destinados a HTTP y HTTPS.
- 5º) Con esto ya está apache2 instalado y funcionando, para comprobarlo he establecido una conexión con ssh desde CentOS a Ubuntu con la opción -X, para poder cargar el navegador, y una vez conectado he ejecutado *gnome-open http://10.0.2.8*, y efectivamente carga la página en el navegador y funciona. Esto lo muestro en la *Ilustración 31*.

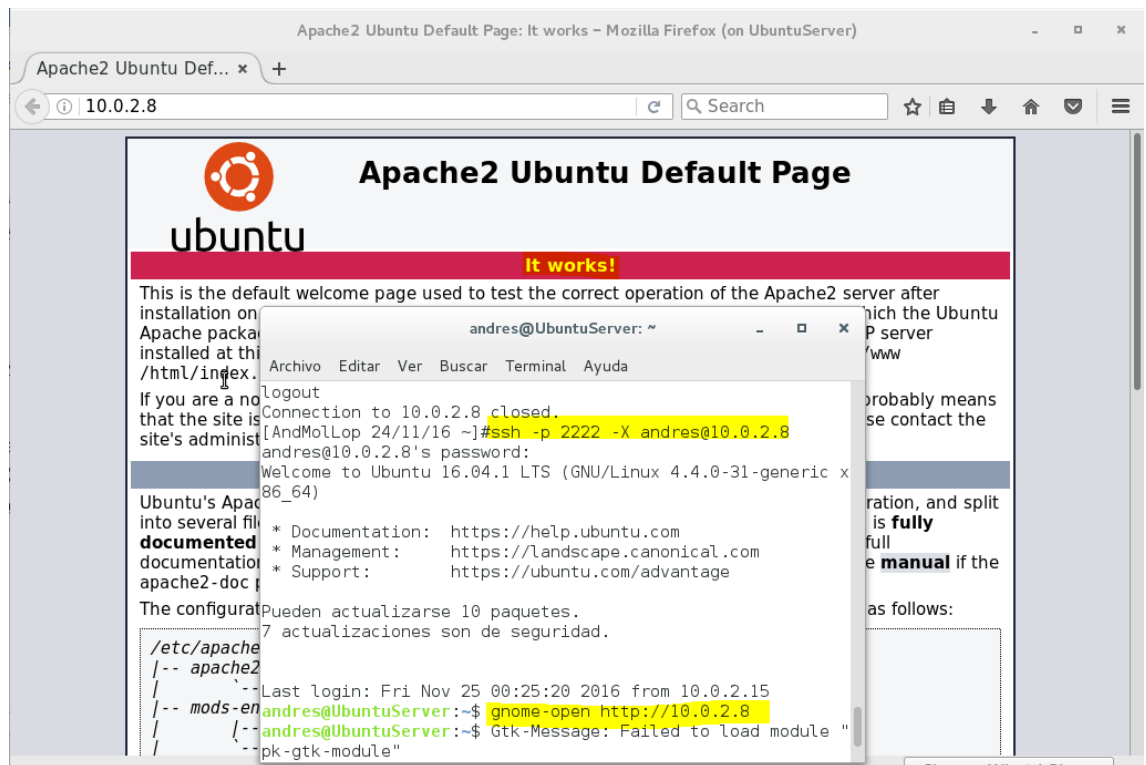


Ilustración 31. Comprobación del funcionamiento de apache2 de manera remota

6º) Para instalar MySQL ejecutamos *apt-get install mysql-server*. Y durante la instalación nos pedirá que introduzcamos una contraseña para el root de MySQL.

7º) Por último, vamos a instalar PHP con unos paquetes extras que hacen que se pueda comunicar con MySQL y pueda funcionar el código en Apache. Para ello ejecutamos el comando *apt-get install php libapache2-mod-php php-mcrypt php-mysql*.

8º) Ahora vamos a hacer que el servidor de Apache mire primero el archivo *index.php* en lugar de *index.html*. Para ello vamos al archivo de configuración de apache *dir.conf* situado en */etc/apache2/mods-enabled*, y al abrirlo con el editor, vemos que hay una lista de directorios index, pues lo que tenemos que hacer es llevar el *index.php* al primero de la lista, de manera que quede como en la Ilustración 32.

```
GNU nano 2.5.3 Archivo: /etc/apache2/mods-enabled/dir.conf
<IfModule mod_dir.c>
    DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Ilustración 32. Archivo *dir.conf* para decirle a Apache2 que mire primero *index.php*

9º) Una vez modificado el archivo *dir.conf*, reiniciamos el servicio de apache2 con *systemctl restart apache2*, para que se hagan efectivos los cambios.



10º) Para finalizar, comprobamos que nuestro sistema está correctamente configurado para PHP , para ello, creamos un archivo llamado *info.php* en el directorio */var/www/html*, y el contenido del archivo ponemos:

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

Y luego nos conectamos otra vez al servidor como mostré en el paso 5 con la *Ilustración 32*, solo que ahora vamos a <http://10.0.2.8/info.php>, con lo que obtenemos los datos del servidor y vemos que php funciona correctamente con el servidor. Yo muestro mi resultado de esta página en la *Ilustración 33*.

Posteriormente borraremos este archivo, para que no puedan acceder usuarios indeseados a esta información.

("How To Install Linux, Apache, MySQL, PHP (LAMP) stack on Ubuntu 16.04 | DigitalOcean", 2016)

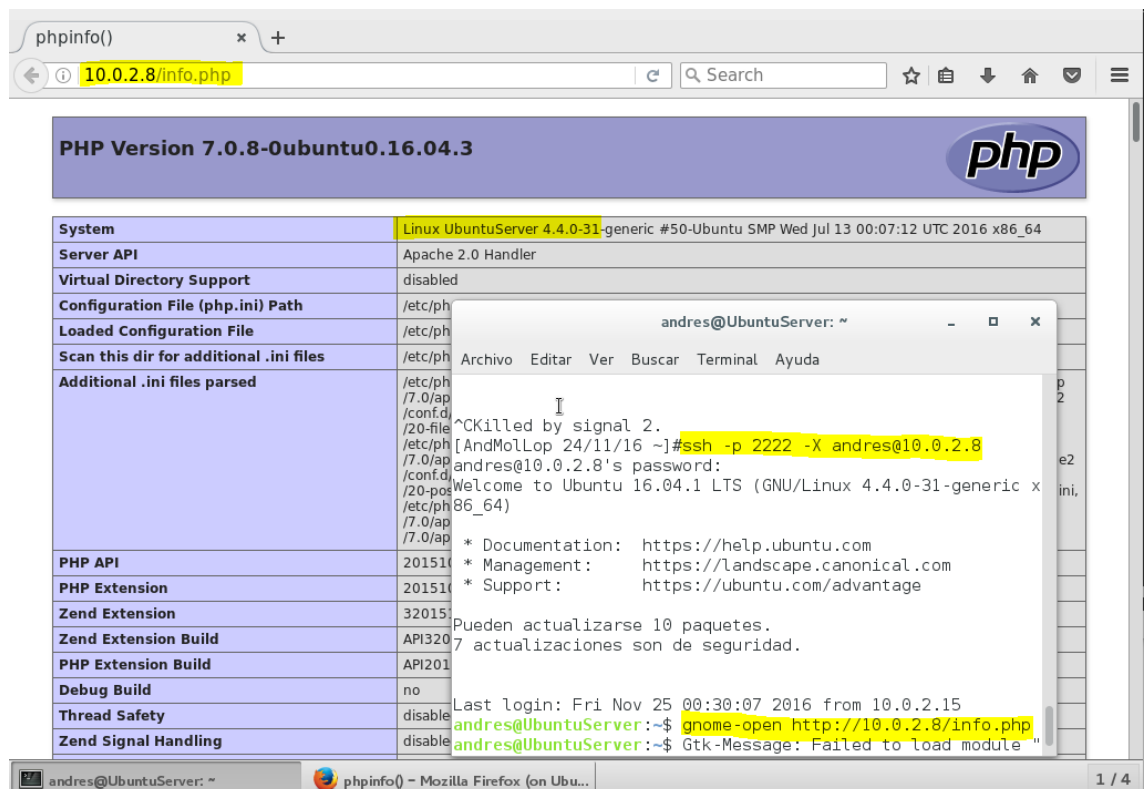


Ilustración 33. Visualización del archivo *info.php* desde el servidor apache de Ubuntu

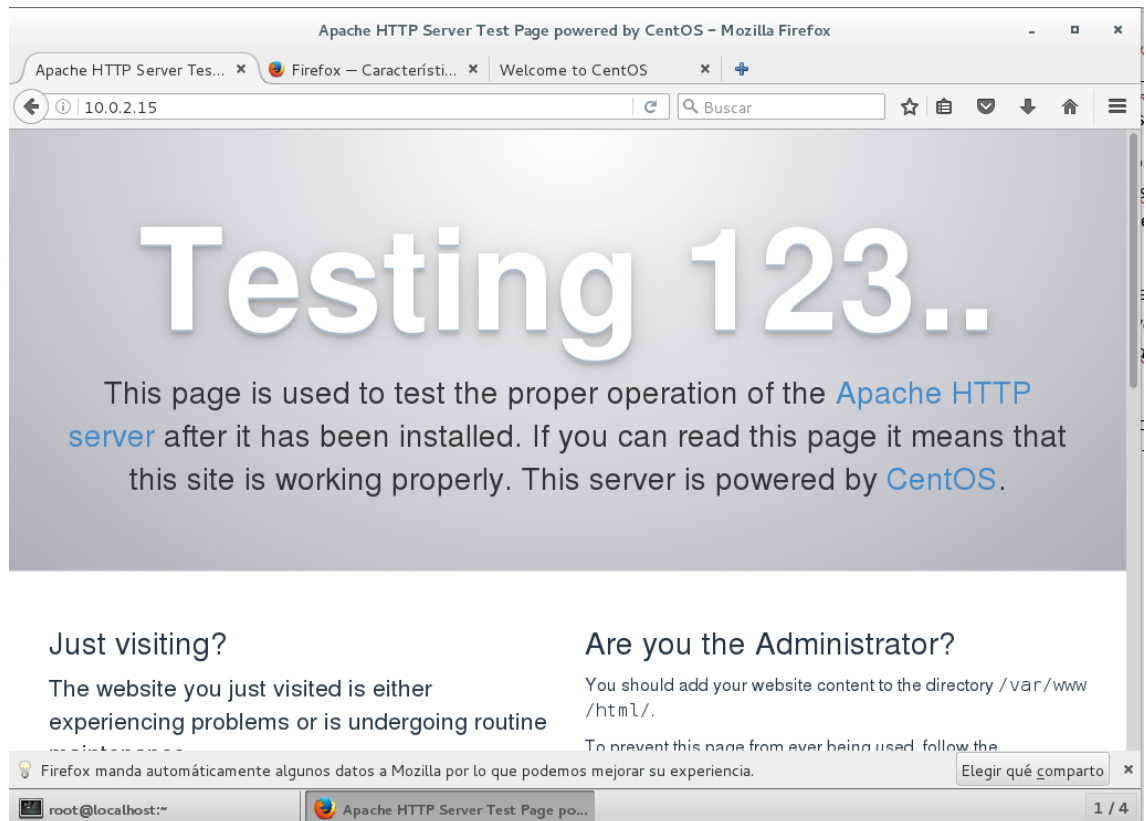
Ahora vamos a instalar estos mismos servicios en CentOS (con IP 10.0.2.15) pero se instalan de manera distinta, y por ello, ahora vamos a seguir estos pasos:

1º) Lo primero es instalar apache2, para ello vamos ejecutar *yum -y install httpd*. Y una vez instalado, iniciamos el servicio con *systemctl start httpd.service* y

hacemos que se inicie al arrancar el sistema ejecutando `systemctl enable httpd.service`

2º) Tenemos que abrir los puertos de apache2, para ello ejecutamos `firewall-cmd --permanent --add-service=http`, después `firewall-cmd --permanent --add-service=https`, y por último `firewall-cmd --reload`, para hacer efectivos los cambios.

3º) Y ahora para comprobar que apache está funcionando, abrimos el navegador y ponemos `http://nuestra_IP`. Yo muestro mi resultado en la *Ilustración 34*.



*Ilustración 34. Prueba de que apache funciona en CentOS*

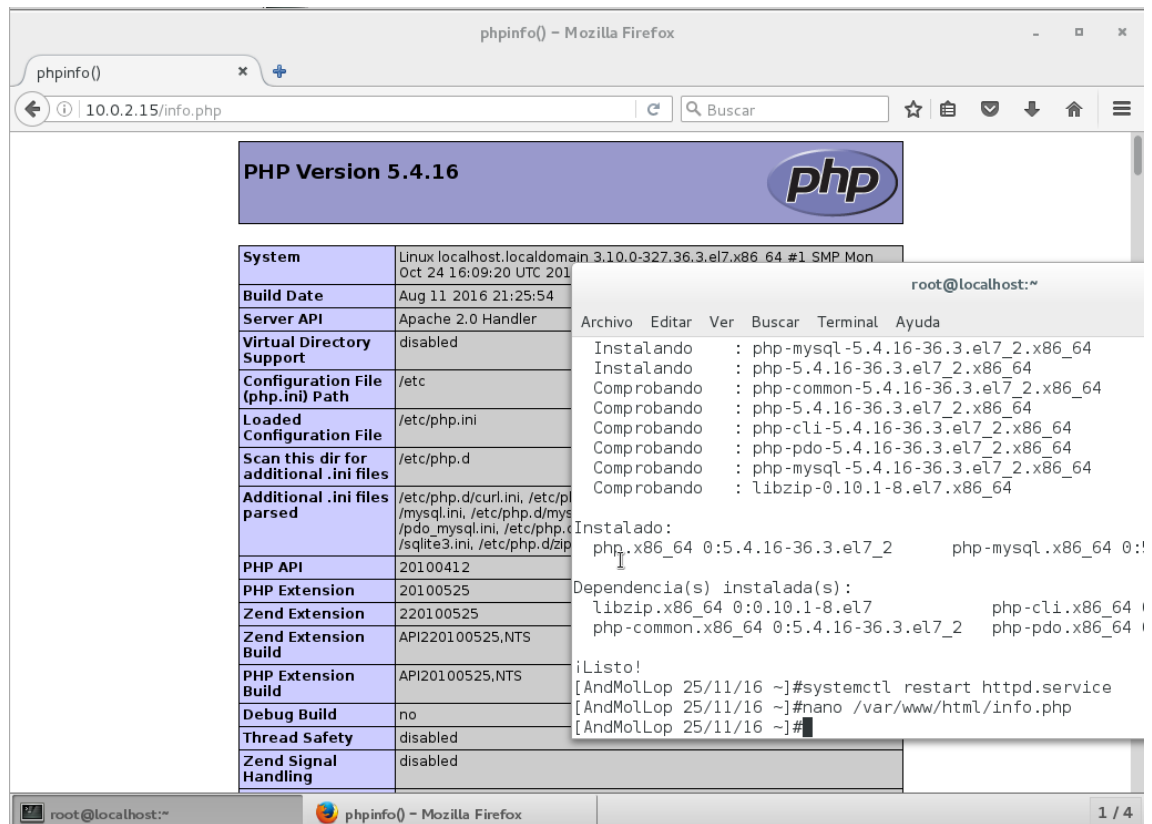
4º) Procedemos ahora a instalar MariaDB, para ello ejecutamos `yum -y install mariadb-server mariadb`. Y una vez instalado, activamos el servicio con `systemctl start mariadb.service`, y hacemos que se arranque al iniciar el sistema con `systemctl enable mariadb.service`.

5º) Para instalar PHP en este caso lo hacemos de la siguiente manera, `yum -y install php php-mysql`. Tras esto reiniciamos apache con `systemctl restart httpd.service` y ya estará funcionando correctamente.

6º) Para finalizar, comprobamos que realmente esté funcionando bien. Para esto creamos el mismo archivo que creamos en el último paso de la instalación en Ubuntu, y en el mismo sitio (archivo `info.php` en `/var/www/html`). Y accedemos a él mediante el servidor de apache, poniendo en el navegador

[http://nuestra\\_IP/info.php](http://nuestra_IP/info.php). Muestro en la *Ilustración 35* el resultado que me muestra esta página, que demuestra que todo está funcionando correctamente.

Posteriormente borramos el archivo *info.php* por motivos de seguridad.



The screenshot shows a web browser window titled 'phpinfo() - Mozilla Firefox' with the address bar displaying '10.0.2.15/info.php'. The main content area shows the 'PHP Version 5.4.16' header and a table of system and configuration details. A terminal window is overlaid on the right side of the page, showing the output of the 'phpinfo()' command. The terminal output includes the following information:

System	Linux localhost.localdomain 3.10.0-327.36.3.el7.x86_64 #1 SMP Mon Oct 24 16:09:20 UTC 2016
Build Date	Aug 11 2016 21:25:54
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/curl.ini, /etc/php.d/mysqli.ini, /etc/php.d/pdo_mysqli.ini, /etc/php.d/sqlite3.ini, /etc/php.d/zip.ini
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extension Build	API220100525,NTS
PHP Extension Build	API20100525,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled

The terminal window also shows the output of the 'phpinfo()' command, including the PHP version, system information, and configuration details. The terminal output is as follows:

```

root@localhost:~# phpinfo()
PHP Version 5.4.16

System
Linux localhost.localdomain 3.10.0-327.36.3.el7.x86_64 #1 SMP Mon Oct 24 16:09:20 UTC 2016
Build Date
Aug 11 2016 21:25:54
Server API
Apache 2.0 Handler
Virtual Directory Support
disabled
Configuration File (php.ini) Path
/etc
Loaded Configuration File
/etc/php.ini
Scan this dir for additional .ini files
/etc/php.d
Additional .ini files parsed
/etc/php.d/curl.ini, /etc/php.d/mysqli.ini, /etc/php.d/pdo_mysqli.ini, /etc/php.d/sqlite3.ini, /etc/php.d/zip.ini
PHP API
20100412
PHP Extension
20100525
Zend Extension
220100525
Zend Extension Build
API220100525,NTS
PHP Extension Build
API20100525,NTS
Debug Build
no
Thread Safety
disabled
Zend Signal Handling
disabled
  
```

*Ilustración 35. Visualización de info.php desde el servidor apache de CentOS*

("Install Apache, PHP And MySQL On CentOS 7 (LAMP)", 2016)

## **10º Cuestión: Realice la instalación usando GUI o PowerShell y compruebe que el servicio está funcionando accediendo a la MV a través de la anfitriona.**

Lo primero que vamos a hacer es cambiar la red la máquina virtual para que vaya por adaptador puente, para que tanto ella como la máquina anfitriona estén conectadas a la misma red, y una vez esto vamos a dejar en la máquina virtual la IP estática, para ello vamos a abrir un Símbolo del sistema, desde Inicio, y vamos a ejecutar el comando `ipconfig /all`, el cual nos dirá nuestra IP, nuestra máscara de subred, nuestra puerta de enlace predeterminada y los servidores DNS. Tal y como muestro en la *Ilustración 36*.

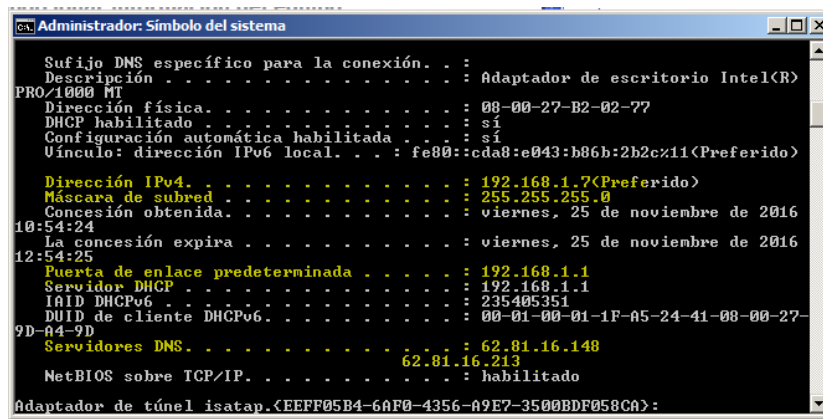


Ilustración 36. Visualización con `ipconfig /all` de la conexión de red establecida

Una vez sabemos esto, vamos a Inicio > Panel de control > Redes e Internet > Centro de Redes y Recursos Compartidos. Una vez aquí, en el lateral izquierdo le damos a cambiar configuración del adaptador, y ahí le damos doble click a la conexión que nos salga, con lo que deberíamos llegar a una ventana como la que muestro en la *Ilustración 37*.

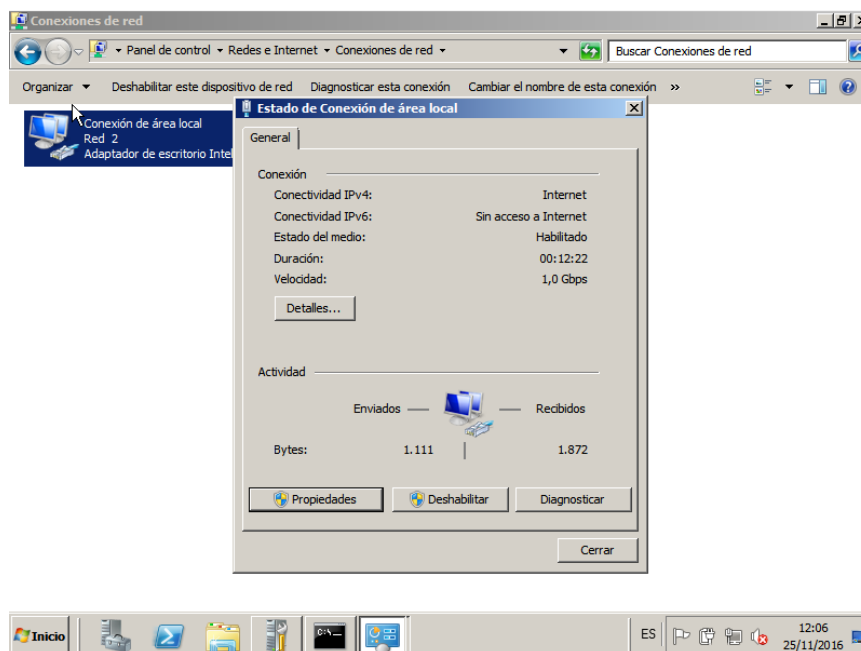


Ilustración 37. Ventana de estado de conexión de la red

En esta ventana le damos a Propiedades, y en la nueva que se nos abre, seleccionamos Protocolo de Internet versión 4(TCP/Ipv4) y le damos a propiedades. Tras esto deberíamos llegar a la ventana que muestro en la *Ilustración 38*.

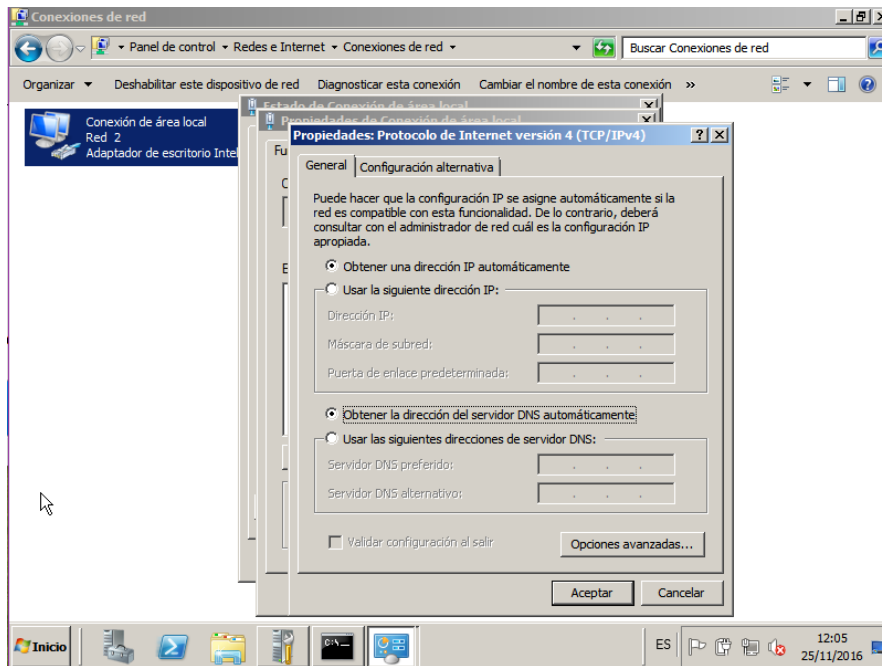


Ilustración 38. Ventana de propiedades del protocolo de internet versión 4

En esta ventana le damos a usar la siguiente dirección IP, y rellenamos los datos con los que nos salieron en el Símbolo del sistema. En mi caso quedarían como muestro en la Ilustración 39.

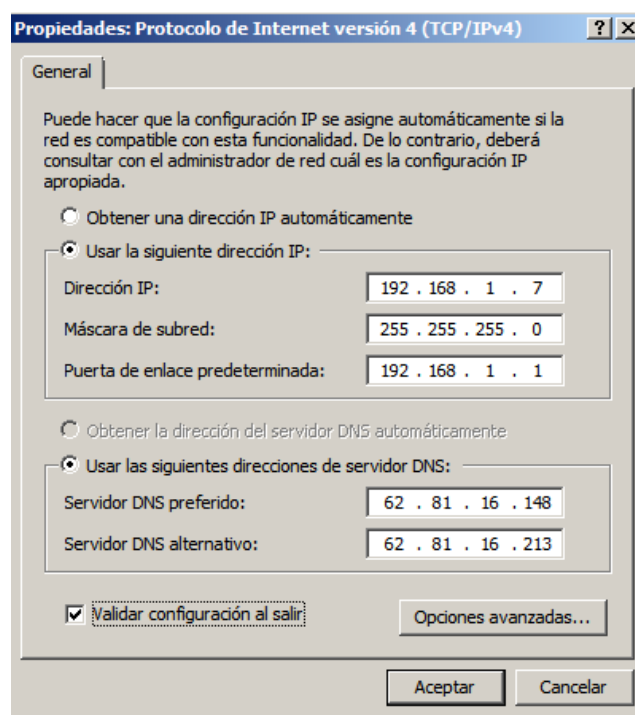


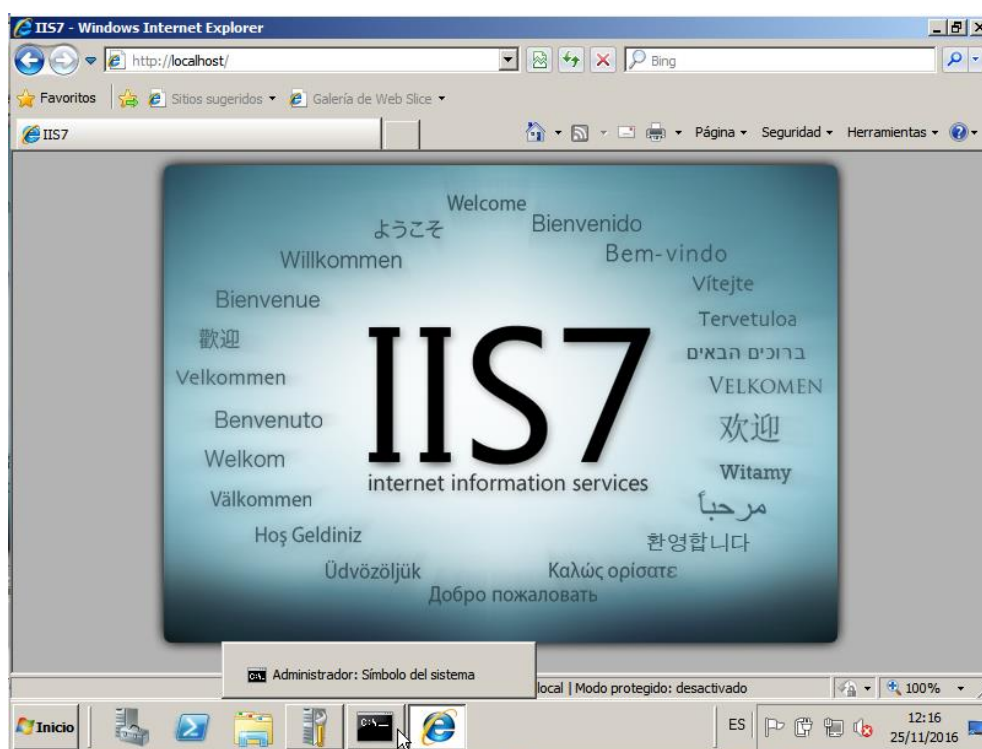
Ilustración 39. Configuración de la IP estática

Y tras aceptar la configuración se habrá hecho efectiva. Acto seguido vamos a Tareas de configuración inicial y en la sección 2 le damos a Descargar e instalar

actualizaciones, y en la ventana que se nos abre le damos a que busque instale las actualizaciones. Una vez termine este proceso, ya podemos pasar a instalar IIS.

Para ello volvemos a Tareas de configuración inicial y en la sección 3 le damos a agregar roles, en la ventana que nos sale le damos a Servidor Web (IIS), y a siguiente hasta que llegamos a Servicios de rol, en el cual dejamos los que hay, y además, marcamos lo mencionados en el guion de la práctica. Luego le damos a siguiente y a instalar, y con esto ya tendremos instalado el servidor web IIS.

Por último, para comprobar que funciona, abrimos el navegador y ponemos en la barra de direcciones `http://localhost`, con lo que, si se nos abre la página que muestro en la *Ilustración 40*, significa que está funcionando correctamente.



*Ilustración 40. Acceso a IIS desde la máquina virtual*

Y para finalizar, si todo ha ido bien, podemos ir a la máquina anfitriona, abrir el navegador y poner la IP que tiene la máquina virtual, y nos tiene que cargar la página también, tal y como muestro en la *Ilustración 41*.



Ilustración 41. Acceso a IIS desde la máquina anfitriona desde el navegador Microsoft Edge

("Configurar una dirección IP estática", 2016)

### **11ª Cuestión: Muestre un ejemplo de uso del comando.**

Como ejemplo de uso del comando *patch*, voy a crear un archivo de C que simplemente imprima por pantalla "Hola mundo!". Luego voy a copiar el archivo para duplicarlo y voy a modificar la copia, para que después de decir "Hola mundo!" también diga "Adiós!" y arreglar los fallos de sintaxis. Muestro estos ficheros y su contenido en la *Ilustración 42*.

```
[AndMolLop 25/11/16 ~]#ls
anaconda-ks.cfg Descargas Escritorio hola.c Música Público
a.out Documentos hola-adios.c Imágenes Plantillas Vídeos
[AndMolLop 25/11/16 ~]#cat hola.c
#include<stdio.h>
main(void){
    printf("Hola mundo!\n");
}
[AndMolLop 25/11/16 ~]#cat hola-adios.c
#include<stdio.h>
int main(void){
    printf("Hola mundo!\n");
    printf("Adios!\n");
    return 0;
}
[AndMolLop 25/11/16 ~]#_
```

Ilustración 42. Ficheros hola.c y su modificación hola-adios.c

Ahora uso el comando *diff -u hola.c hola-adios-c > hola.patch* para crear un parche llamado *hola.patch* y miro el contenido del parche con *cat hola.patch*. Muestro el resultado en la *Ilustración 43*.

```

[AndMolLop 25/11/16 ~]#diff -u hola.c hola-adios.c > hola.patch
[AndMolLop 25/11/16 ~]#ls
anaconda-ks.cfg  Documentos  hola.c      Música      Vídeos
a.out           Escritorio  hola.patch  Plantillas
Descargas       hola-adios.c  Imágenes   Público
[AndMolLop 25/11/16 ~]#cat hola.patch
--- hola.c      2016-11-25 14:59:29.568487914 +0100
+++ hola-adios.c 2016-11-25 14:59:16.441462877 +0100
@@ -1,4 +1,6 @@
#include<stdio.h>
main(void){
+int main(void){
+    printf("Hola mundo!\n");
+    printf("Adios!\n");
+    return 0;
+}
[AndMolLop 25/11/16 ~]#_

```

Ilustración 43. Creación del parche y visualización del contenido para ver los cambios que va a realizar

Ahora para aplicar el parche, ejecutamos `patch < hola.patch`, el parche ya sabe sobre que archivo tiene que aplicar los cambios y de que archivo mirarlos, tal y como hemos visto en la *Ilustración 43*. Una vez el parche se ha aplicado, visualizamos el contenido del archivo `hola.c`, el cual ha tenido que ser modificado debido al parche. Muestro el resultado de aplicar el parche en la *Ilustración 44*.

```

[AndMolLop 25/11/16 ~]#patch < hola.patch
patching file hola.c
[AndMolLop 25/11/16 ~]#cat hola.c
#include<stdio.h>
int main(void){
    printf("Hola mundo!\n");
    printf("Adios!\n");
    return 0;
}
[AndMolLop 25/11/16 ~]#_

```

Ilustración 44. Aplicación del parche y resultado

Si ahora comparamos el contenido de `hola.c` con el que tenía antes, comparando la *Ilustración 44* con la *Ilustración 42*, vemos que el contenido ha cambiado como decía el parche.

Por último, si compilamos `hola.c` con `gcc hola.c -o hola` y lo ejecutamos veremos que muestra los dos mensajes. Muestro el resultado de la ejecución en la *Ilustración 45*.

```

[AndMolLop 25/11/16 ~]#gcc hola.c -o hola
[AndMolLop 25/11/16 ~]#./hola
Hola mundo!
Adios!
[AndMolLop 25/11/16 ~]#_

```

Ilustración 45. Resultado de la ejecución una vez aplicado el parche

("HowTo Apply a Patch File To My Linux / UNIX Source Code", 2016)



**12ª Cuestión: Realice la instalación de esta aplicación y pruebe a modificar algún parámetro de algún servicio. Muestre las capturas de pantalla pertinentes así como el proceso de instalación.**

Para instalar Webmin, lo voy a hacer mediante un repositorio, para ello añado el nuevo repositorio a yum mediante `nano /etc/yum.repos.d/webmin.repo`, y en el contenido de este escribo lo que muestro en la *Ilustración 46*.

```
GNU nano 2.3.1      Fichero: /etc/yum.repos.d/webmin.repo
[Webmin]
name=Webmin Distribution Neutral
#baseurl=http://download.webmin.com/download/yum
mirrorlist=http://download.webmin.com/download/yum/mirrorlist
enable=1_
```

*Ilustración 46. Contenido de webmin.repo*

Una vez creado el repositorio, nos descargamos e importamos las claves con las que se instalan los paquetes de este nuevo repositorio, para ello ejecutamos `rpm --import http://www.webmin.com/jcameron-key.asc`.

Luego actualizamos los repositorios con `yum check-update`, y una vez actualizados instalamos webmin con `yum -y install webmin`. Ya instalado iniciamos el servicio y hacemos que se arranque automáticamente con `chkconfig webmin on` y `service webmin start`. Muestro en la *Ilustración 47* la instalación de webmin, que también nos instala las dependencias que necesita, y la iniciación del servicio.

```
-----
Total                               3.3 MB/s | 28 MB  00:08
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Instalando      : perl-Net-SSLeay-1.55-3.el7.x86_64                1/2
Operating system is CentOS Linux
  Instalando      : webmin-1.820-1.noarch                          2/2
Webmin install complete. You can now login to https://localhost.localdomain:10000/
as root with your root password.
  Comprobando     : perl-Net-SSLeay-1.55-3.el7.x86_64                1/2
  Comprobando     : webmin-1.820-1.noarch                          2/2

Instalado:
  webmin.noarch 0:1.820-1

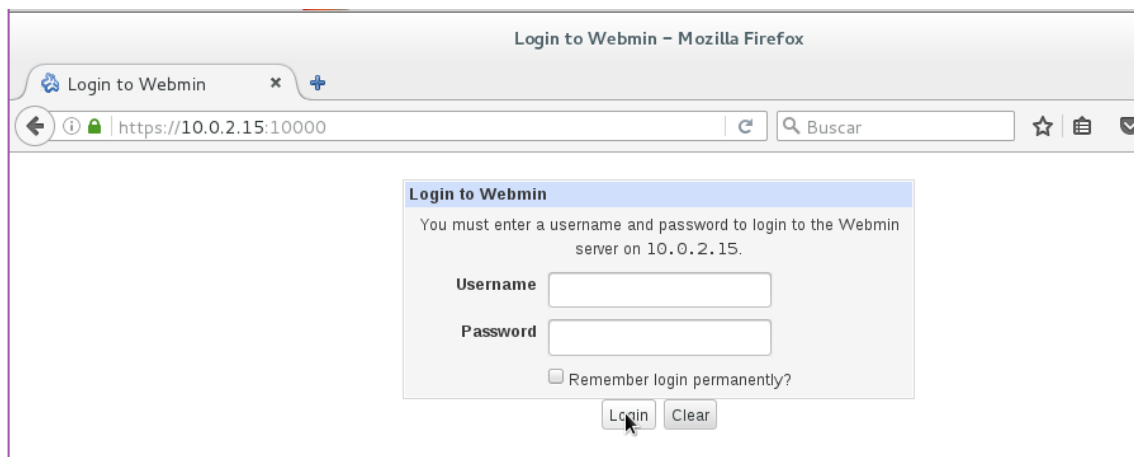
Dependencia(s) instalada(s):
  perl-Net-SSLeay.x86_64 0:1.55-3.el7

¡Listo!
[AndMolLop 25/11/16 ~]#chkconfig webmin on
[AndMolLop 25/11/16 ~]#service webmin start
[AndMolLop 25/11/16 ~]#_
```

*Ilustración 47. Instalación de Webmin y sus dependencias e iniciación del servicio*

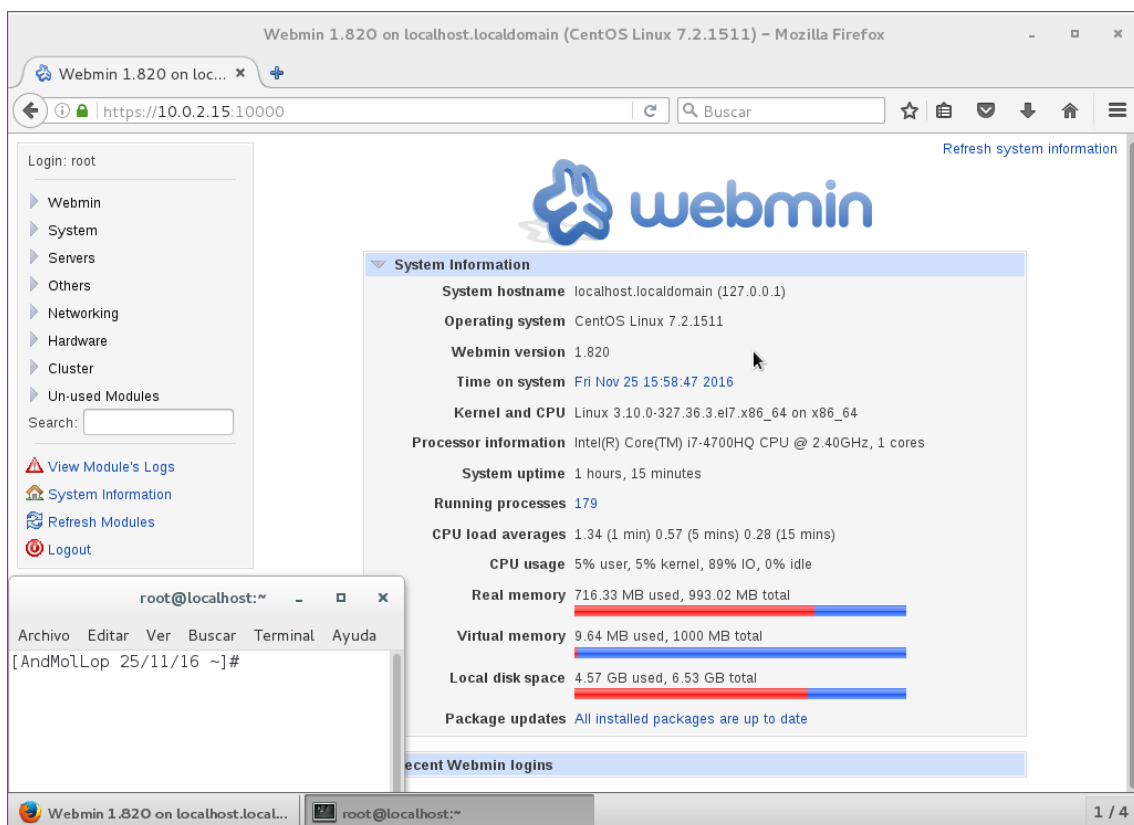
Por último, como Webmin utiliza el puerto 10000, lo habilitamos en el cortafuegos con `firewall-cmd --permanent --add-port=10000/tcp` y reiniciamos el cortafuegos `firewall-cmd --reload`.

Ya para finalizar, en el modo gráfico, abrimos el navegador y vamos a `https://nuestra_IP:10000`, y accedemos con nuestra cuenta de root. Muestro mi ventana de acceso en la *Ilustración 48*.



*Ilustración 48. Página de acceso a Webmin*

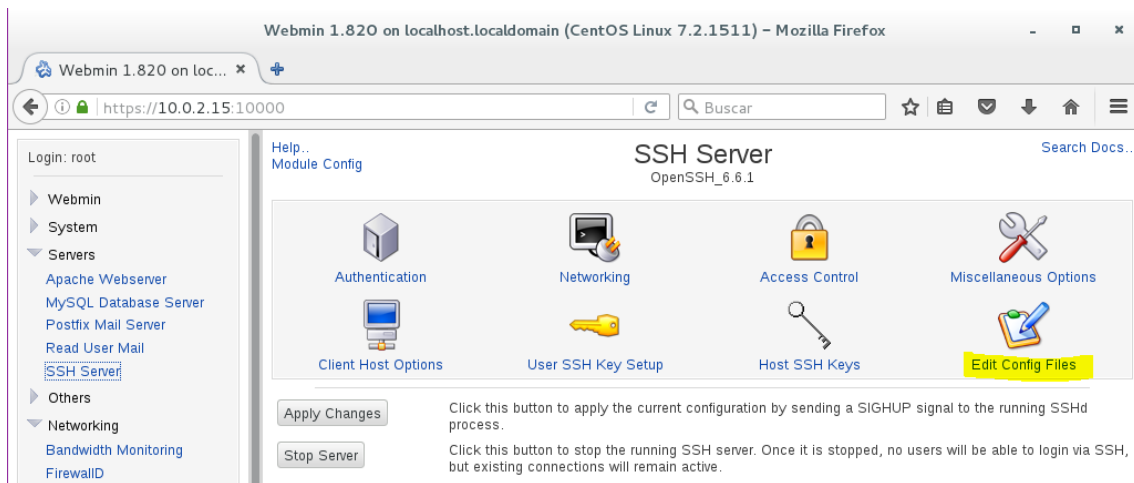
Una vez hemos accedido con nuestra cuenta de root, debemos ver algo parecido a lo que yo muestro en la *Ilustración 49*, en la cual se ve mi configuración del sistema.



*Ilustración 49. Página de inicio de Webmin*

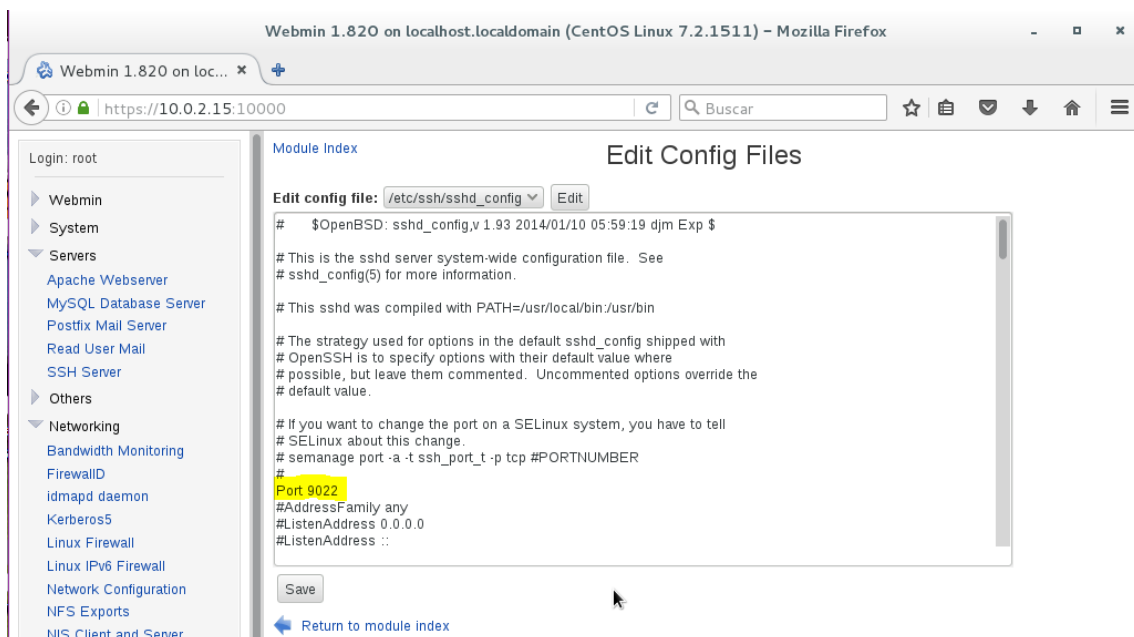
Y ahora por ejemplo para ver que funciona bien, voy a cambiar el puerto por el cual se establece la conexión ssh, para ello en la barra lateral izquierda le damos a

Servers y en el menú que se nos despliega le damos a SSH server, con lo cual llegamos a la página que muestro en la *Ilustración 50*.



*Ilustración 50. Opciones de SSH server desde Webmin*

En esta página le damos Edit Config Files, y veremos el archivo `sshd_config`, en el que cambiamos el Port que tengamos, por, por ejemplo 9022, y guardamos dándole a Save, lo que nos devolverá a la página anterior, donde le damos a Apply Changes. Muestro este cambio en la *Ilustración 51*.



*Ilustración 51. Modificación del puerto de ssh desde Webmin*

Ahora tenemos que agregar el puerto al cortafuegos, para ello le damos en la barra izquierda a Networking y ahí a FirewallD, con lo que se nos abrirá la página que muestro en la *Ilustración 52*.

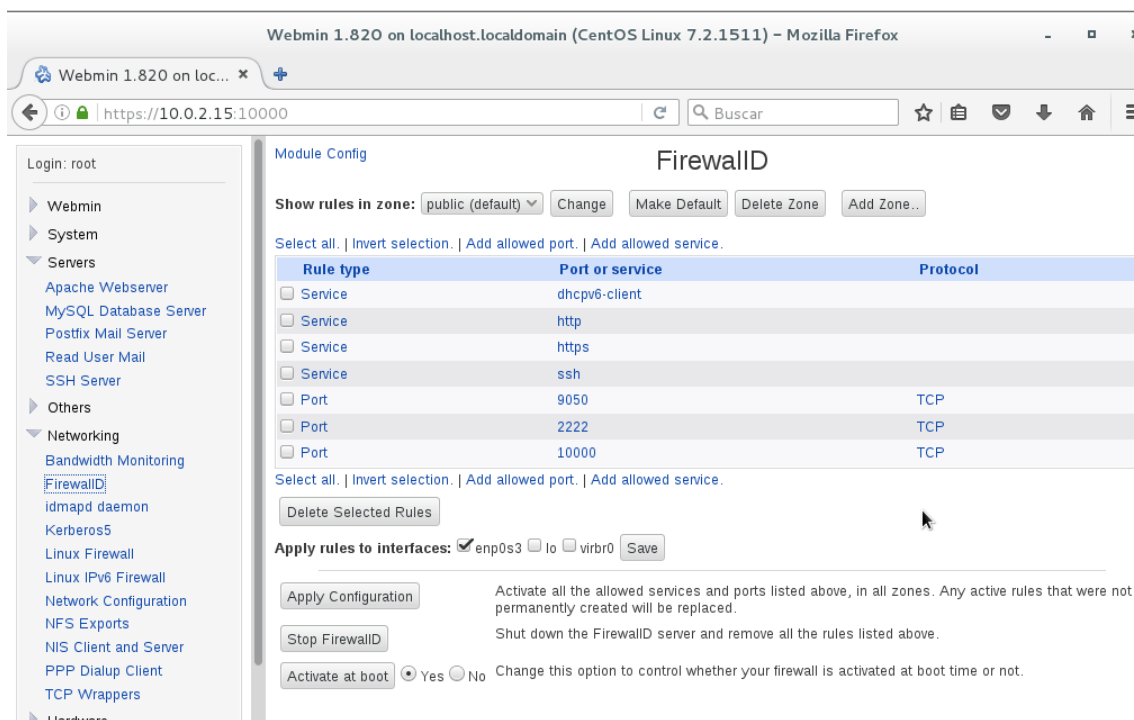


Ilustración 52. Cortafuegos desde Webmin

Una vez aquí, le damos a add allowed port, con lo que se nos abre la página de la *Ilustración 53*, y aquí creamos el puerto que hemos le hemos puesto a ssh.

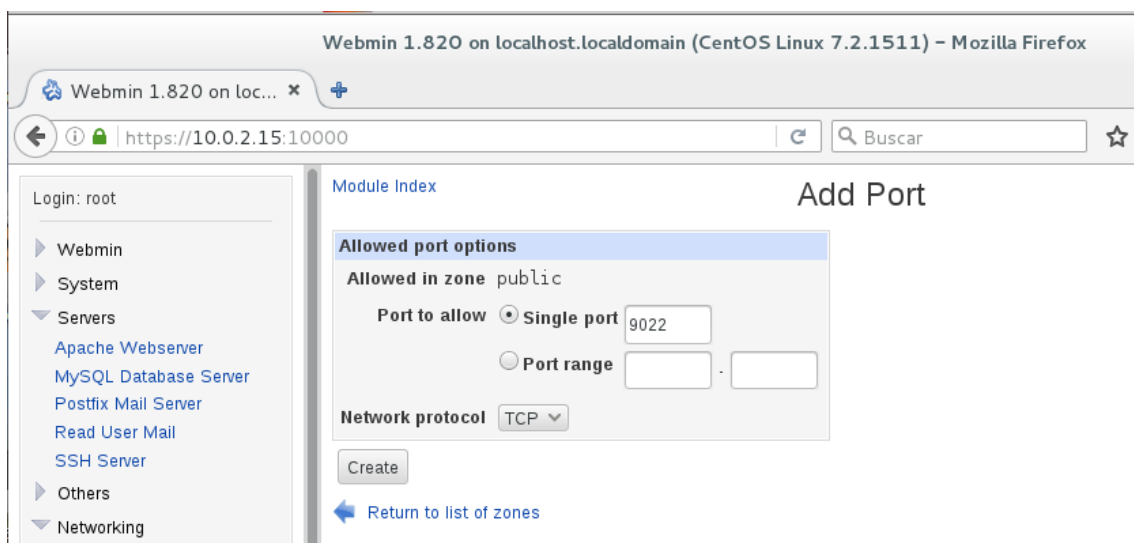


Ilustración 53. Añadir puerto a FirewallID desde Webmin

Una vez añadido al cortafuegos, le damos a Apply Configuración, y con esto ya estará cambiado el puerto por el cual se accede a ssh, para finalizar habrá que usar el comando `semanage port -a -t ssh_port_t -p tcp 9022` para que se haga efectivo y funcione.

Para comprobar que el cambio se ha hecho correctamente accedo desde Ubuntu, y como muestro en la *Ilustración 54* el cambio de puerto se ha realizado satisfactoriamente, y por el 2222 ya no se puede establecer la conexión.

```

AndMolLop 25/11/16:~$ ssh -p 2222 andresCentOS@10.0.2.15
ssh: connect to host 10.0.2.15 port 2222: Connection refused
AndMolLop 25/11/16:~$ ssh -p 9022 andresCentOS@10.0.2.15
The authenticity of host '10.0.2.15:9022 ([10.0.2.15]:9022)' can't be established.
ECDSA key fingerprint is SHA256:mx1cRh5CCbQI/Pu101lu0jp+LRD1Bi8SCUT9M4c774E.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.15:9022' (ECDSA) to the list of known hosts.
Last login: Fri Nov 25 16:26:58 2016 from 10.0.2.8
[AndMolLop CentOS 25/11/16 ~]# ls
prueba.txt
[AndMolLop CentOS 25/11/16 ~]#

```

Ilustración 54. Acceso a CentOS desde Ubuntu mediante SSH por el puerto 9022

("How to install Webmin on CentOS 7", 2016)

**13ª Cuestión: Instale phpMyAdmin, indique cómo lo ha realizado y muestre algunas capturas de pantalla. Configure PHP para poder importar BDs de hasta 25MiB (en vez de los 8MiB de límite por defecto). Indique cómo ha realizado el proceso y muestre capturas de pantalla.**

Para instalar phpMyAdmin vamos a ejecutar *apt-get install phpmyadmin php-mbstring php-gettext*, lo cual nos lanzará el instalador que muestro en la *Ilustración 55*, y en el que seleccionamos apache2.

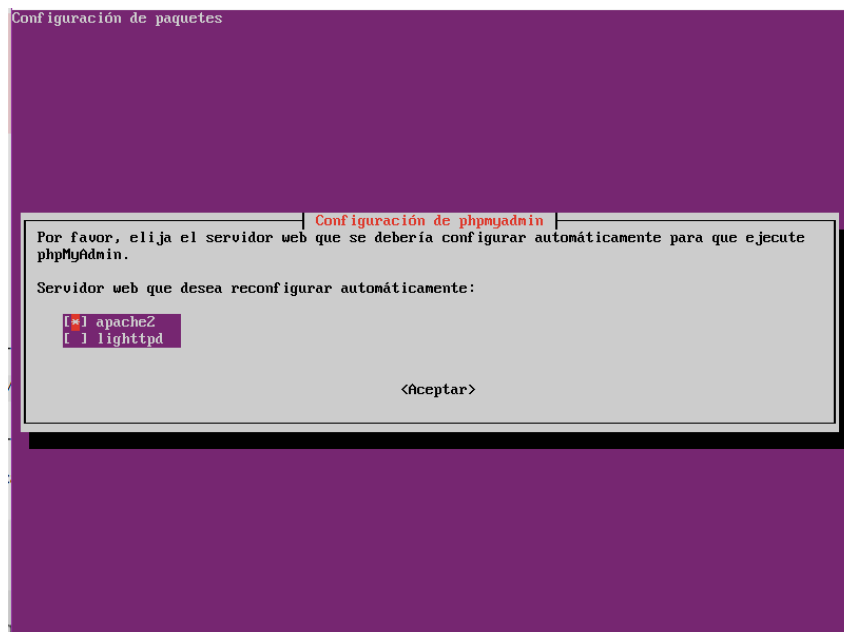
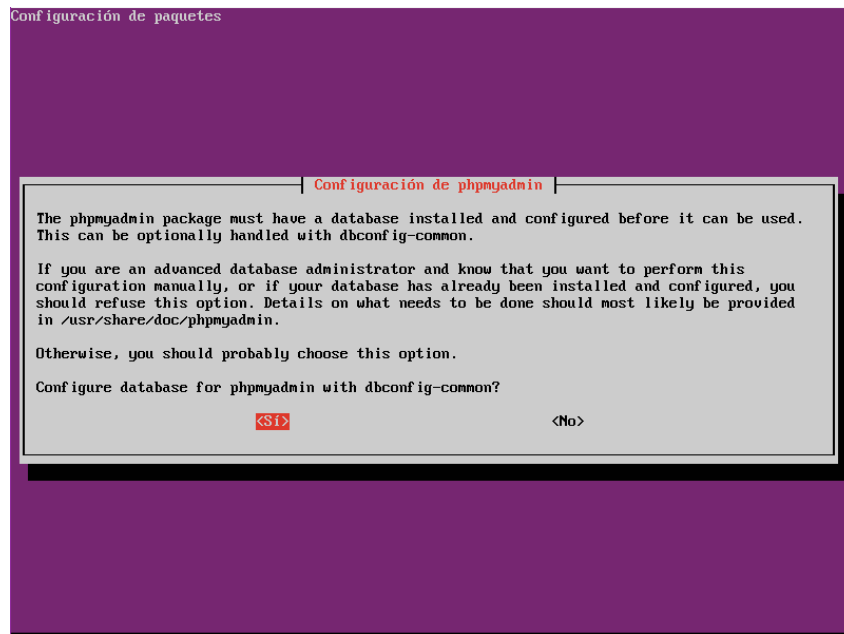


Ilustración 55. Instalador de phpMyAdmin

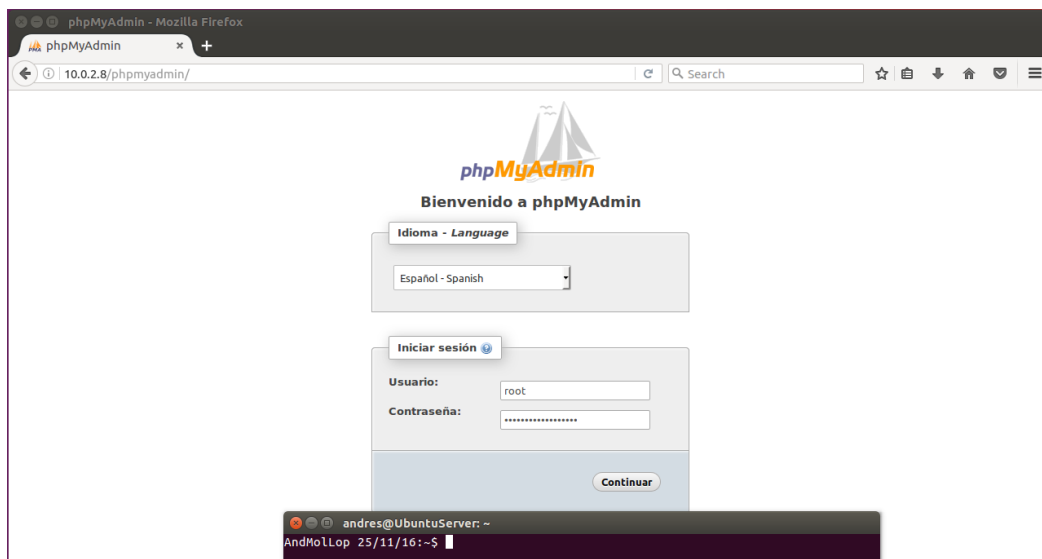
En la siguiente ventana, que la muestro en la *Ilustración 56*, le damos a Sí.



*Ilustración 56. Configuración de la base de datos para phpMyAdmin*

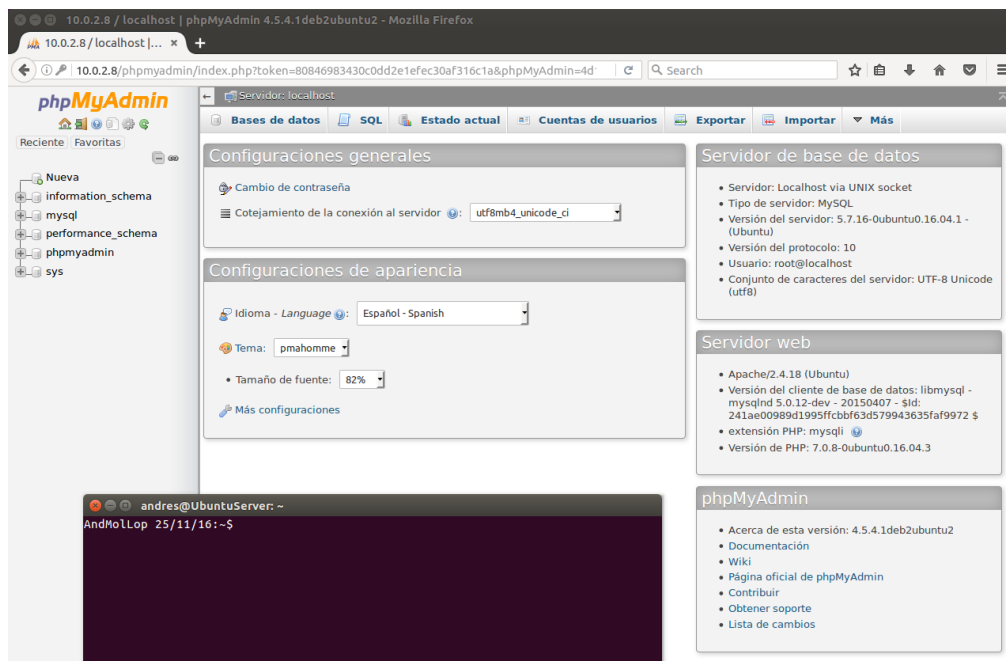
Y nos pide que ingresemos una contraseña para el usuario root de phpMyAdmin y la confirmemos. Una vez instalado, tenemos que habilitar las extensiones de PHP mcrypt y mbstring, para ello ejecutamos *phpenmod mcrypt* y *phpenmod mbstring*, y luego reiniciamos el servicio de apache2 con *systemctl restart apache2*.

Y con esto ya podemos acceder a phpMyAdmin poniendo en el navegador *http://nuestra\_IP/phpmyadmin*. Como muestro en la *Ilustración 57* yo accedo de manera activando la interfaz gráfica de Ubuntu Server.



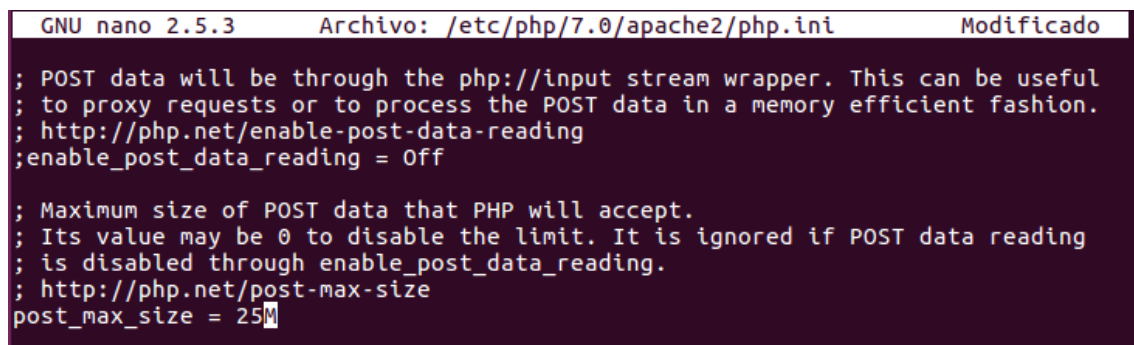
*Ilustración 57. Acceso a phpMyAdmin de manera remota desde CentOS*

En el usuario ponemos root, y en la contraseña ponemos la que le dimos al root al instalar mysql-server. Y una vez accedamos llegaremos a la página que muestro en la *Ilustración 58*.



*Ilustración 58. Página al acceder a phpMyAdmin*

Ahora vamos a modificar el tamaño de bases de datos que podemos importar a 25MiB, para ello vamos a abrir el archivo de configuración *php.ini* situado en */etc/php/7.0/apache2*. Y una vez en este archivo vamos a modificar el parámetro *post\_max\_size* a 25M, tal y como muestro en la *Ilustración 59*.



*Ilustración 59. Tamaño máximo de subida de PHP*

Una vez modificado el archivo, reiniciamos apache2 para que se hagan efectivos los cambios, y con esto ya se podrán importar bases de datos de hasta 25 MiB.

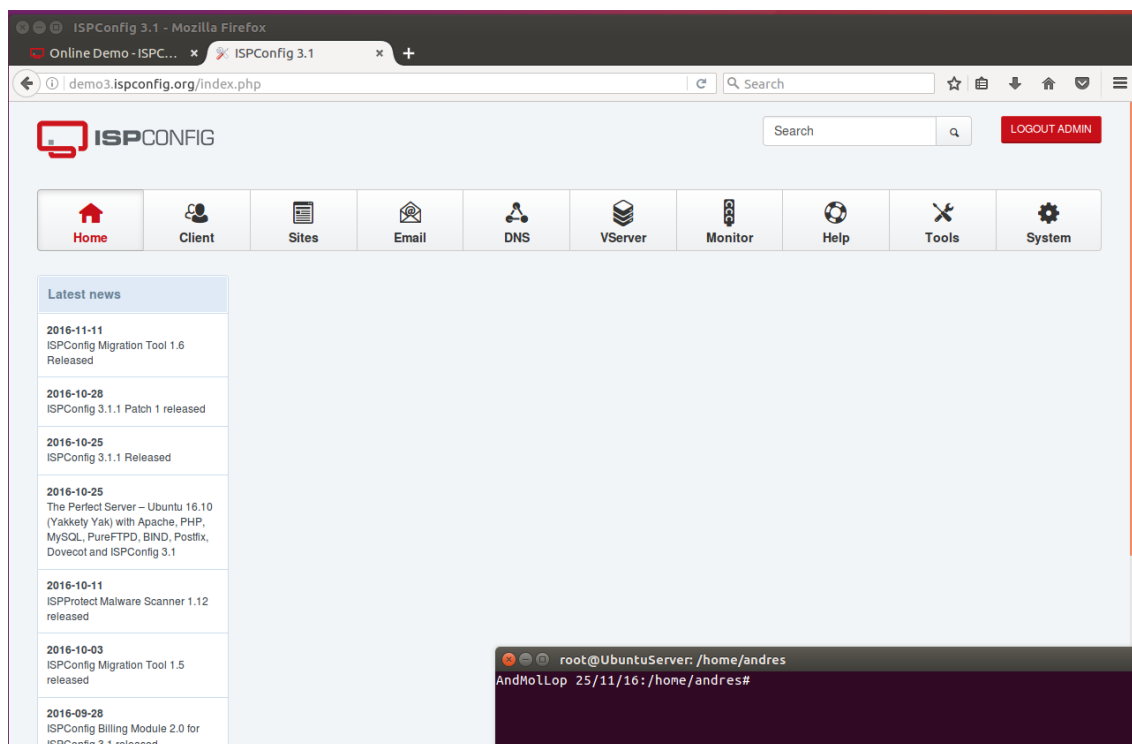
("How To Install and Secure phpMyAdmin on Ubuntu 16.04 | DigitalOcean", 2016)

("Aumentar el límite de upload en PHP", 2016)

**14º Cuestión: Visite al menos una de las webs de los software mencionados y pruebe las demos que ofrecen realizando capturas de pantalla y comentando qué está realizando.**

He decidido probar ISPConfig, para ello accedemos su web y le damos a demo, hay nos dan el usuario y contraseña de administrado, y el enlace por el cual tenemos que acceder.

Una vez accedemos llegamos a la página que muestro en la *Ilustración 60*, donde se nos muestran las distintas opciones que podemos usar.



*Ilustración 60. Página inicial de ISPConfig.*

Desde aquí ya podemos, por ejemplo, ver los clientes que tiene el servidor, editarlos, añadir nuevos clientes, etc...como muestro en la *Ilustración 61*.



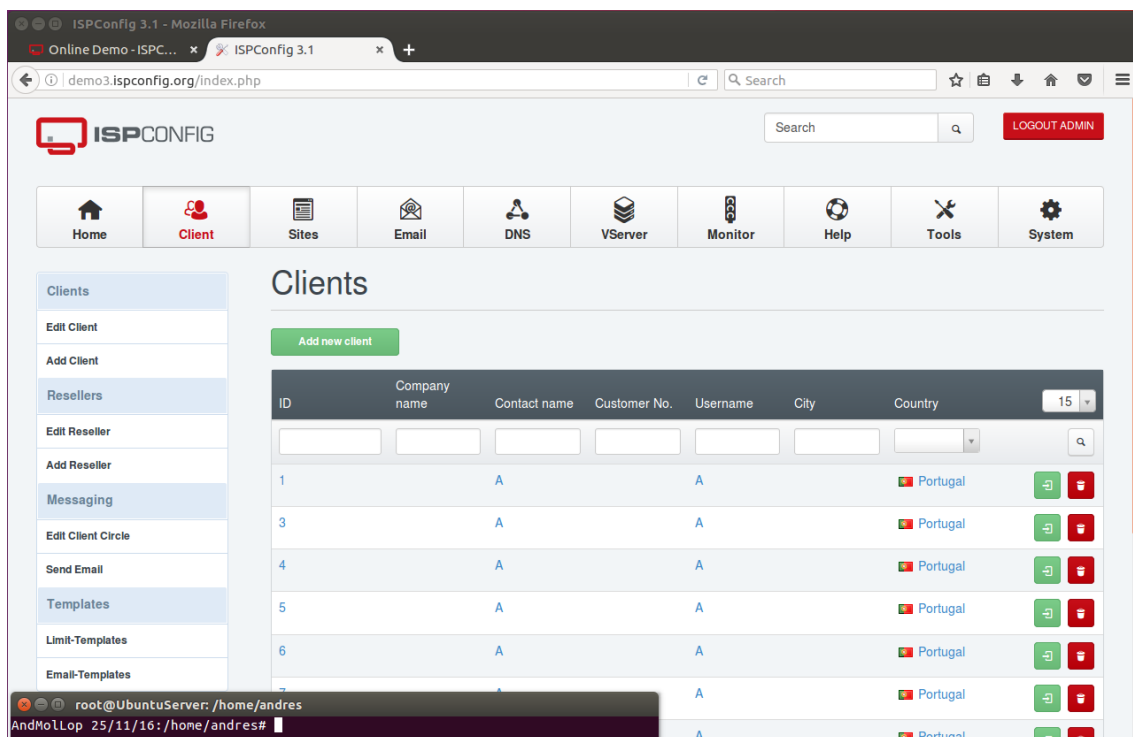


Ilustración 61. Administración de clientes en el servidor con ISPconfig

Por ejemplo, vamos a probar a añadir un nuevo cliente, para ello le damos a Add new client, y rellenamos una serie de campos, que muestro en la *Ilustración 62*.

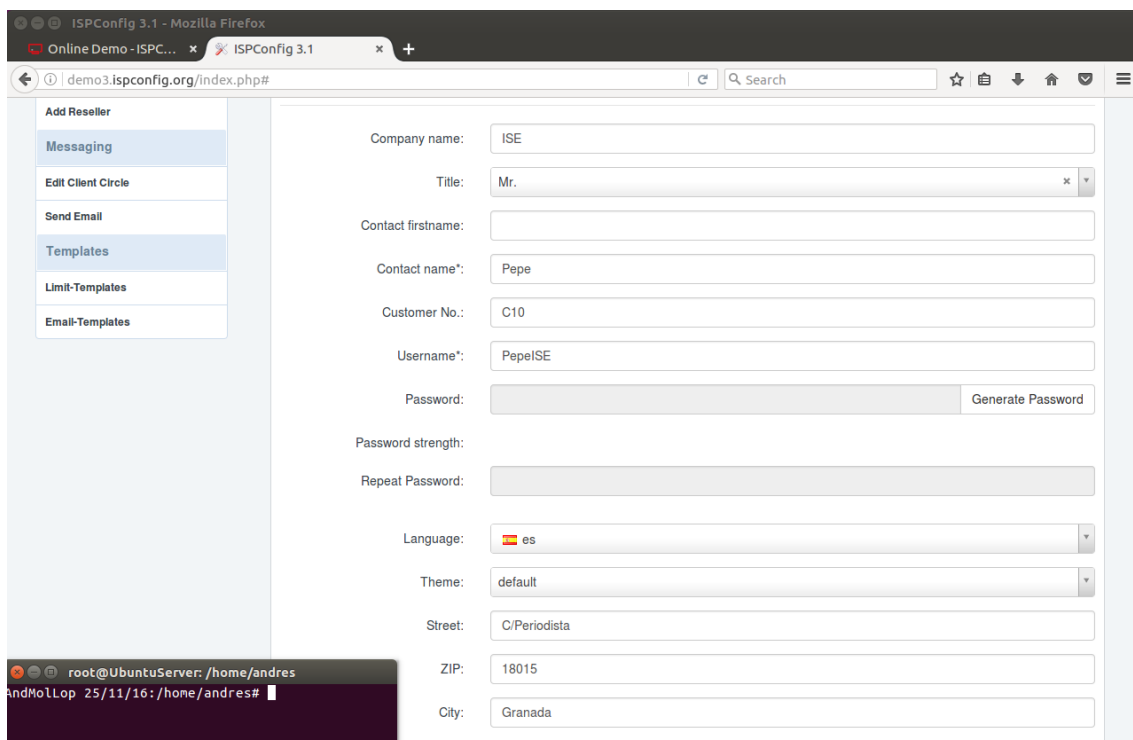


Ilustración 62. Añadir nuevo cliente al servidor con ISPconfig

Luego le damos a Save, y como vemos en la *Ilustración 63*, el cliente ha sido añadido correctamente

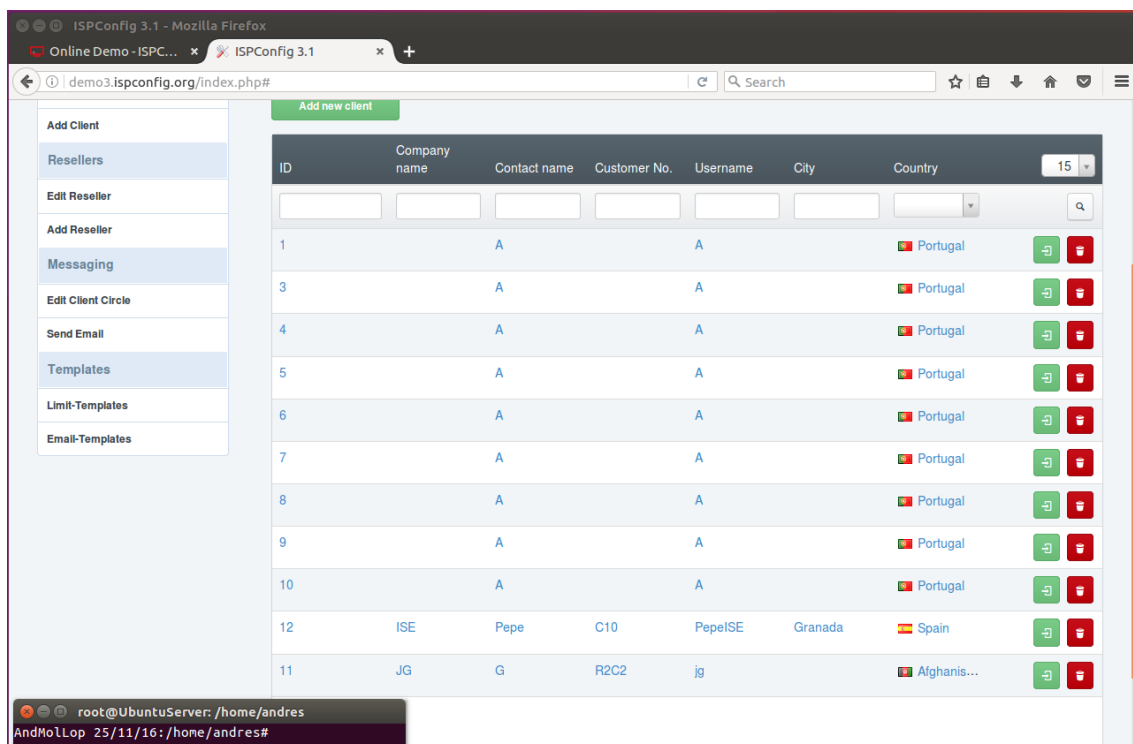


Ilustración 63. Cliente añadido al servidor con ISPconfig

Y esto es una de las cosas que podemos administrar con ISPconfig, el cual tiene una demo bastante reducida, pero tiene una implementación muy intuitiva.

("ISPConfig Hosting Control Panel", 2016)

## 15ª Cuestión:

### a) Ejecute los ejemplos de find, grep.

Mi resultado de ejecutar el ejemplo de grep, lo muestro en la *Ilustración 64*.

```
root@UbuntuServer: /home/andres
AndMolLop 25/11/16:~$ ps -Af | grep firefox
andres  32398 31613  0 18:27 pts/2    00:00:00 grep --color=auto firefox
AndMolLop 25/11/16:~$
```

Ilustración 64. Uso del comando grep

Y el resultado de ejecutar el ejemplo de find, lo muestro en la *Ilustración 65*.

```

root@UbuntuServer: /home/andres
AndMolLop 25/11/16:~$ ls Documentos/
adios.pdf  hola.pdf
AndMolLop 25/11/16:~$ ls PDFs/
AndMolLop 25/11/16:~$ find /home/andres/Documentos -name '*.pdf' -exec cp {} ~/PDFs \;
AndMolLop 25/11/16:~$ ls PDFs/
adios.pdf  hola.pdf
AndMolLop 25/11/16:~$

```

Ilustración 65. Uso del comando find

**b) Escriba el script que haga uso de sed para cambiar la configuración de ssh y reiniciar el servicio.**

He creado un script que cambie el parámetro que permite conectarse como root de no a sí, y que luego reinicie el servicio para que se haga efectivo el cambio. Muestro este script en la *Ilustración 66*.

```

AndMolLop 25/11/16:/home/andres# cat allowrootssh.sh
#!/bin/bash
sed -i 's/PermitRootLogin no/PermitRootLogin yes/' /etc/ssh/sshd_config
service ssh restart
AndMolLop 25/11/16:/home/andres#

```

Ilustración 66. Script que automatiza el cambio de la configuración de ssh para que si se pueda conectar como root

**c) Muestre un ejemplo de uso para awk.**

Para el ejemplo de awk he creado un fichero con una serie de números dispuestos en dos columnas, y luego he ejecutado awk haciendo que compruebe que los números de la primera columna son mayores que la segunda, y en caso de que lo sean, que escriba entre medio de cada número “es mayor que” y en caso de que no lo sea, que escriba “no es mayor que”. Muestro el fichero con los números y la ejecución de awk que hace lo dicho anteriormente en la *Ilustración 67*.

```

root@UbuntuServer: /home/andres
AndMolLop 25/11/16:~$ cat numeros.txt
2 5
8 3
19 68
54 78
9 1
AndMolLop 25/11/16:~$ awk '{if( $1 > $2 ) {print $1 " es mayor que " $2;} else {print $1
" no es mayor que " $2;}}' numeros.txt
2 no es mayor que 5
8 es mayor que 3
19 no es mayor que 68
54 no es mayor que 78
9 es mayor que 1
AndMolLop 25/11/16:~$ █

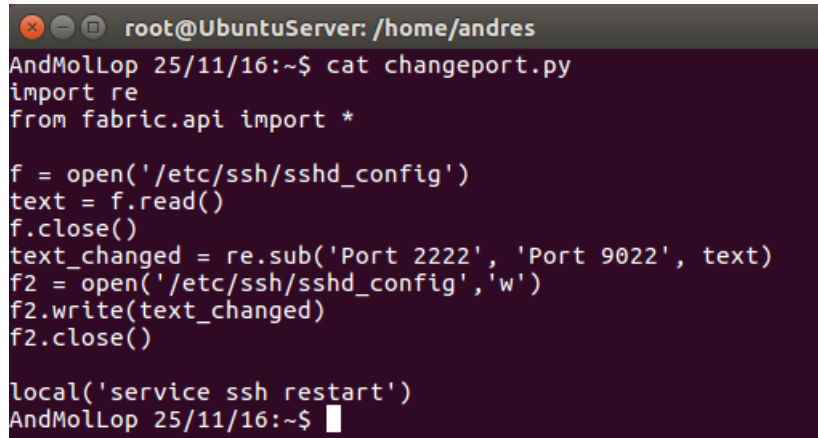
```

Ilustración 67. Uso del comando awk sobre el fichero numeros.txt

("AWK paso a paso... y sin usar el ratón | El rincón de Linux", 2016)

## 16ª Cuestión: Escriba el script para cambiar el acceso a ssh usando PHP o Python.

He creado un script en Python que lo que hace es cambiar el puerto de ssh por otro, está pensado para que el puerto que haya de serie sea el 2222 que es el que yo tengo, y que el puerto que se va a poner sea el 9022, el cual debe abrirse en el cortafuegos manualmente. En la *Ilustración 68* muestro el script que he creado.



```
root@UbuntuServer: /home/andres
AndMolLop 25/11/16:~$ cat changeport.py
import re
from fabric.api import *

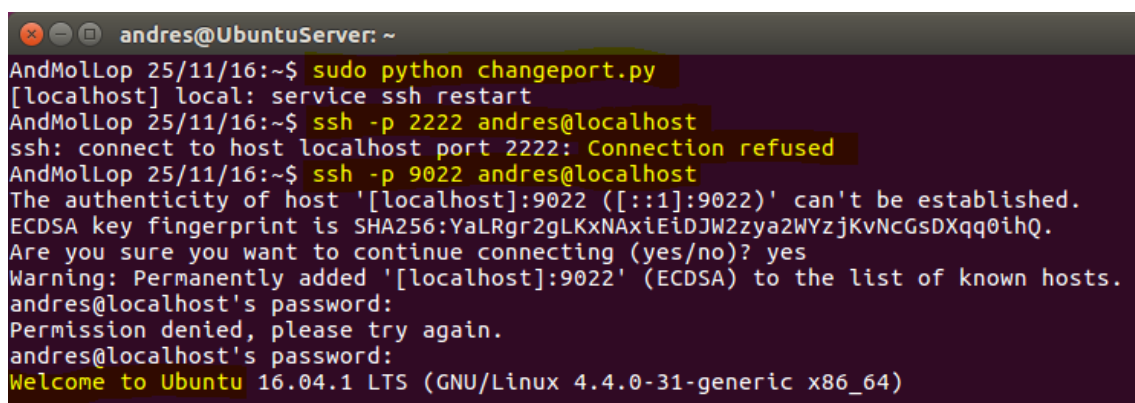
f = open('/etc/ssh/sshd_config')
text = f.read()
f.close()
text_changed = re.sub('Port 2222', 'Port 9022', text)
f2 = open('/etc/ssh/sshd_config', 'w')
f2.write(text_changed)
f2.close()

local('service ssh restart')
AndMolLop 25/11/16:~$
```

*Ilustración 68. Script en Python para cambiar puerto de ssh*

Para ejecutarlo hay que usar *sudo python changeport.py* ya que se requiere de tocar archivos y de ejecutar ordenes que necesitan privilegios de administrador. Básicamente el script lo que hace es almacenar en la variable `text` el archivo `sshd_config`, y luego en la variable `text_changed` vuelve a guardar todo el texto, pero cambiando la cadena `Port 2222` por `Port 9022`, una vez hecho esto, escribe `text_changed` en el archivo `sshd_config`, y por último, reinicia el servicio de ssh para que sea efectivo.

En la *Ilustración 69* demuestro que funciona correctamente, ejecutando el script, y posteriormente intentando conectarme por el puerto 2222 que era el que tenía, y al ver que falla, conecto por el puerto 9022 que es el que me ha puesto el script y ya sí conecta correctamente.



```
andres@UbuntuServer: ~
AndMolLop 25/11/16:~$ sudo python changeport.py
[localhost] local: service ssh restart
AndMolLop 25/11/16:~$ ssh -p 2222 andres@localhost
ssh: connect to host localhost port 2222: Connection refused
AndMolLop 25/11/16:~$ ssh -p 9022 andres@localhost
The authenticity of host '[localhost]:9022 (:::1):9022)' can't be established.
ECDSA key fingerprint is SHA256:YaLRgr2gLKxNAXiEiDJW2zya2WYzjKvNcGsDXqq0ihQ.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[localhost]:9022' (ECDSA) to the list of known hosts.
andres@localhost's password:
Permission denied, please try again.
andres@localhost's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-31-generic x86_64)
```

*Ilustración 69. Cambio de puerto de ssh ejecutando el script de Python y conexión ssh establecida*

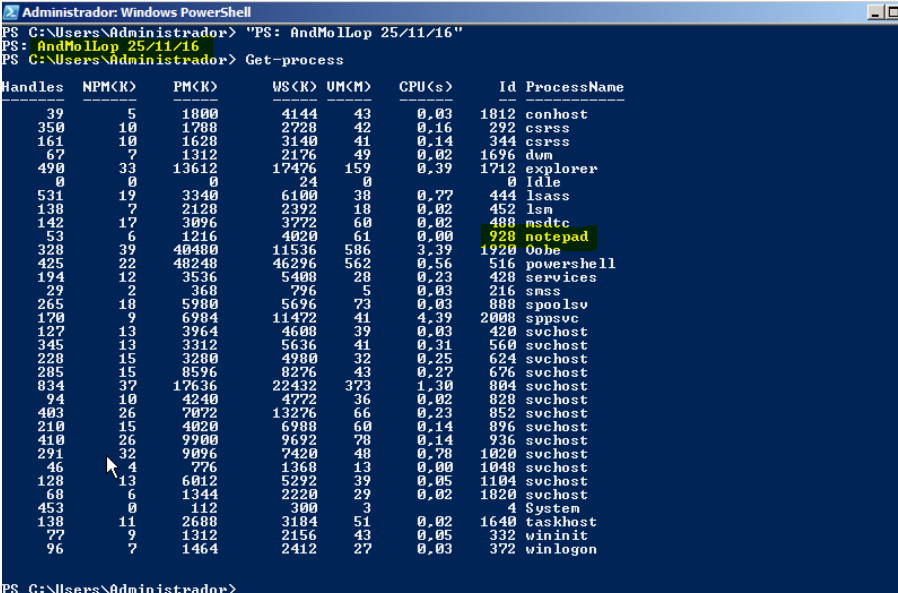
("7.2. re — Regular expression operations — Python 2.7.12 documentation", 2016)

("Systems Administration — The Hitchhiker's Guide to Python", 2016)

("How to use Fabric in Python", 2016)

## **17º Cuestión: Abra una consola de Powershell y pruebe a parar un programa en ejecución, realice capturas de pantalla y comente lo que se muestra.**

Para cerrar un programa concreto, voy a abrir el bloc de notas, además de la consola Powershell. Una vez en la consola, ejecuto *Get-process*, que nos muestra los procesos que hay funcionando y nos da sus IDs. En la *Ilustración 70* podemos ver los procesos y entre ellos el notepad que es el que yo he abierto. (También se puede ver que no he podido cambiar el PS porque no me dejaba acceder a \$profile).



Handles	NPM(K)	PM(K)	WS(K)	UM(M)	CPU(s)	Id	ProcessName
39	5	1800	4144	43	0.03	1812	conhost
350	10	1788	2728	42	0.16	292	csrss
161	10	1628	3140	41	0.14	344	csrss
67	7	1312	2176	49	0.02	1696	dm
490	33	13612	17476	159	0.39	1712	explorer
0	0	0	24	0	0	0	Idle
531	19	3340	6100	38	0.77	444	lsass
138	7	2128	2392	18	0.02	452	lsn
142	17	3096	3772	60	0.02	488	msdtc
53	6	1216	4820	61	0.00	928	notepad
328	39	40480	11536	586	3.39	1920	OObe
425	22	48248	46296	562	0.56	516	powershell
194	12	3536	5408	28	0.23	428	services
29	2	368	796	5	0.03	216	smss
265	18	5980	5696	73	0.03	888	spoolsv
170	9	6984	11472	41	4.39	2088	sppsvc
127	13	3964	4600	39	0.03	428	svchost
345	13	3312	5636	41	0.31	560	svchost
228	15	3280	4980	32	0.25	624	svchost
285	15	8596	8276	43	0.27	676	svchost
834	37	17636	22432	373	1.30	804	svchost
94	10	4240	4772	36	0.02	828	svchost
493	26	7072	13276	66	0.23	852	svchost
210	15	4820	6988	60	0.14	896	svchost
410	26	9900	9692	70	0.14	936	svchost
291	32	9096	7420	48	0.78	1020	svchost
46	4	776	1368	13	0.00	1048	svchost
128	13	6012	5292	39	0.05	1104	svchost
68	6	1344	2220	29	0.02	1820	svchost
453	0	112	300	3	0	4	System
138	11	2688	3184	51	0.02	1640	taskhost
77	9	1312	2156	43	0.05	332	wininit
96	7	1464	2412	27	0.03	372	winlogon

*Ilustración 70. Visualización de procesos activos*

Ahora que sabemos la ID del proceso notepad que es el bloc de notas, usamos *Stop-process ID\_proceso*. Y como vemos en la *Ilustración 71* el proceso ha sido detenido con éxito, ya que ya no aparece en *Get-process*.

```

Administrador: Windows PowerShell
PS C:\Users\Administrador> "PS: AndMolLop 25/11/16"
PS: AndMolLop 25/11/16
PS C:\Users\Administrador> Stop-process 928
PS C:\Users\Administrador> Get-process

```

Handles	NPM(K)	PM(K)	WS(K)	UM(M)	CPU(s)	Id	ProcessName
39	5	1800	4144	43	0.08	1812	conhost
355	10	1788	2728	42	0.16	292	csrss
157	10	1628	3132	41	0.16	344	csrss
67	7	1312	2176	49	0.02	1696	dwm
491	33	13632	18136	159	0.39	1712	explorer
0	0	0	24	0	0	0	Idle
535	19	3340	6100	38	0.81	444	lsass
138	17	2128	2392	18	0.02	452	lsn
142	17	3096	3772	60	0.02	488	msdtc
337	39	40476	10592	586	3.47	1920	Oobe
494	23	49628	47876	563	0.63	516	powershell
192	12	3484	5392	27	0.23	428	services
29	2	368	796	5	0.03	216	smss
265	18	5980	5696	73	0.03	888	spoolsv
171	8	6932	11460	41	4.61	2008	sppsvc
127	13	3964	4608	39	0.03	420	svchost
345	13	3316	5652	41	0.31	560	svchost
229	15	3332	5000	33	0.25	624	svchost
294	16	8648	8300	43	0.28	676	svchost
839	37	17692	22456	374	1.30	804	svchost
94	10	4352	4812	37	0.02	828	svchost
426	28	7232	13364	68	0.25	852	svchost
235	15	4020	6988	60	0.14	896	svchost
411	26	9900	9756	78	0.14	936	svchost
292	32	9096	7412	48	0.78	1020	svchost
46	4	776	1368	13	0.00	1048	svchost
128	12	5960	5276	39	0.05	1104	svchost
68	6	1344	2220	29	0.02	1820	svchost
455	0	112	300	3	0.00	4	System
137	11	2688	3184	51	0.02	1640	taskhost
77	9	1312	2156	43	0.05	332	wininit
96	7	1464	2412	27	0.03	372	winlogon

```

PS C:\Users\Administrador>

```

Ilustración 71. Detención del proceso notepad desde Powershell

Ahora por ejemplo pruebo a parar explorer, y en la *Ilustración 72* vemos que se detiene, pero automáticamente vuelve a ejecutarse, aunque con una ID distinta.

```

Administrador: Windows PowerShell
PS C:\Users\Administrador> "PS: AndMolLop 25/11/16"
PS: AndMolLop 25/11/16
PS C:\Users\Administrador> Stop-process 1712
PS C:\Users\Administrador> Get-process

```

Handles	NPM(K)	PM(K)	WS(K)	UM(M)	CPU(s)	Id	ProcessName
39	5	1800	4168	43	0.11	1812	conhost
364	10	1788	2728	42	0.16	292	csrss
166	10	1628	3136	41	0.16	344	csrss
67	7	1312	2252	49	0.02	1696	dwm
458	32	13092	23784	133	0.19	1532	explorer
0	0	0	24	0	0	0	Idle
545	19	3436	6600	39	0.83	444	lsass
143	7	2240	2508	18	0.02	452	lsn
142	17	3096	3772	60	0.02	488	msdtc
333	40	40536	14400	586	3.56	1920	Oobe
520	23	49628	47884	563	0.69	516	powershell
194	12	3588	5428	28	0.23	428	services
29	2	368	796	5	0.03	216	smss
265	18	5980	5696	73	0.03	888	spoolsv
170	9	7024	11552	41	4.72	2008	sppsvc
127	13	3964	4608	39	0.03	420	svchost
347	14	3368	5724	42	0.36	560	svchost
231	15	3336	5012	33	0.25	624	svchost
283	15	8108	7980	44	0.31	676	svchost
841	38	17812	23228	375	1.30	804	svchost
94	10	4352	4812	37	0.02	828	svchost
421	27	7240	13456	68	0.27	852	svchost
226	15	4020	7212	60	0.16	896	svchost
410	26	9916	9772	78	0.14	936	svchost
291	32	9096	7412	48	0.78	1020	svchost
46	4	776	1368	13	0.00	1048	svchost
128	13	6016	5296	39	0.05	1104	svchost
68	6	1344	2220	29	0.02	1820	svchost
454	0	112	300	3	0.00	4	System
137	11	2688	3200	51	0.02	1640	taskhost
77	9	1312	2156	43	0.05	332	wininit
96	7	1552	3144	27	0.03	372	winlogon

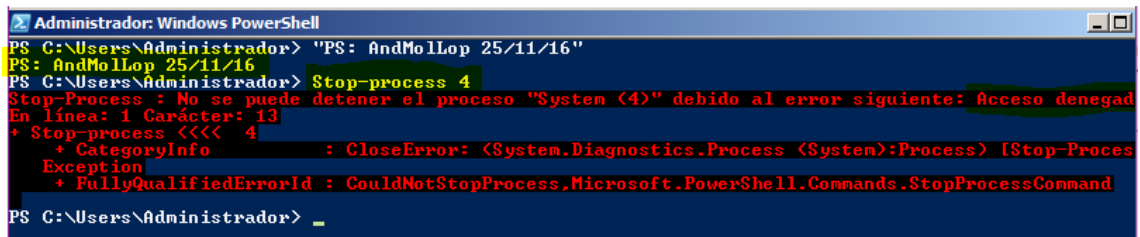
```

PS C:\Users\Administrador>

```

Ilustración 72. Detención del proceso explorer desde Powershell

Y, por último, intento detener el proceso System, y como vemos en la *Ilustración 73*, el propio sistema deniega el acceso directamente.



```
Administrador: Windows PowerShell
PS C:\Users\Administrador> "PS: AndMolLop 25/11/16"
PS: AndMolLop 25/11/16
PS C:\Users\Administrador> Stop-process 4
Stop-Process : No se puede detener el proceso "System (4)" debido al error siguiente: Acceso denegado
En línea: 1 Carácter: 13
+ Stop-process <<<< 4
+ ~~~~~
+ CategoryInfo          : CloseError: (System.Diagnostics.Process (System):Process) [Stop-Process]
+ Exception             :
+ FullyQualifiedErrorId : CouldNotStopProcess,Microsoft.PowerShell.Commands.StopProcessCommand
PS C:\Users\Administrador> _
```

*Ilustración 73. Intento de detención de System desde Powershell y denegación de la acción*

("Using the Stop-Process Cmdlet", 2016)

## **Bibliografía:**

- Cuestión 1:
  - *yum(8) - Linux manual page*. (2016). *Man7.org*. Retrieved 17 November 2016, from <http://man7.org/linux/man-pages/man8/yum.8.html>
  - *10. Usando yum con un servidor Proxy*. (2016). *Docs.fedoraproject.org*. Retrieved 17 November 2016, from [https://docs.fedoraproject.org/es-ES/Fedora Core/4/html/Software Management Guide/sn-yum-proxy-server.html](https://docs.fedoraproject.org/es-ES/Fedora%20Core/4/html/Software%20Management%20Guide/sec-yum-proxy-server.html)
  - *8.4.5. Adding, Enabling, and Disabling a Yum Repository*. (2016). *Access.redhat.com*. Retrieved 19 November 2016, from [https://access.redhat.com/documentation/en-US/Red Hat Enterprise Linux/6/html/Deployment Guide/sec-Managing Yum Repositories.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Deployment_Guide/sec-Managing_Yum_Repositories.html)
- Cuestión 2:
  - *Ubuntu Manpage: apt - command-line interface*. (2016). *Manpages.ubuntu.com*. Retrieved 19 November 2016, from <http://manpages.ubuntu.com/manpages/xenial/man8/apt.8.html>
  - Zamphirópolis, J. (2016). *Configurar proxy para apt-get aptitude en Ubuntu y Debian*. *Rapido-facil.blogspot.com.es*. Retrieved 19 November 2016, from <http://rapido-facil.blogspot.com.es/2011/09/configurar-proxy-para-apt-get-aptitude.html>
  - *Añadir repositorios externos - Guía Ubuntu*. (2016). *Guia-ubuntu.com*. Retrieved 19 November 2016, from [http://www.guia-ubuntu.com/index.php/A%C3%B1adir repositorios externos](http://www.guia-ubuntu.com/index.php/A%C3%B1adir%20repositorios%20externos)
- Cuestión 3:
  - *UFW - Community Help Wiki*. (2016). *Help.ubuntu.com*. Retrieved 19 November 2016, from <https://help.ubuntu.com/community/UFW>
  - Woerner, T. (2016). *firewall-cmd. firewalld*. Retrieved 20 November 2016, from <http://www.firewalld.org/documentation/man-pages/firewall-cmd.html>
  - *How To Set Up a Firewall Using Firewalld on CentOS 7 | DigitalOcean*. (2016). *Digitalocean.com*. Retrieved 20 November 2016, from <https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-using-firewalld-on-centos-7>
  - *3.8.13.5.7. Open Ports in the Firewall using the CLI*. (2016). *Docs.fedoraproject.org*. Retrieved 20 November 2016, from [https://docs.fedoraproject.org/en-US/Fedora/19/html/Security Guide/sec-Open Ports in the firewall-CLI.html](https://docs.fedoraproject.org/en-US/Fedora/19/html/Security_Guide/sec-Open_Ports_in_the_firewall-CLI.html)
  - *Port Scanning Basics*. (2016). *Nmap.org*. Retrieved 23 November 2016, from <https://nmap.org/book/man-port-scanning-basics.html>



- Cuestión 4:
  - Alvarez, M. (2016). *¿Qué es Telnet y SSH?*. DesarrolloWeb.com. Retrieved 23 November 2016, from <http://www.desarrolloweb.com/articulos/telnet-ssh-protocolo-red.html>
- Cuestión 5:
  - *OpenBSD manual pages*. (2016). Man.openbsd.org. Retrieved 23 November 2016, from <http://man.openbsd.org/OpenBSD-current/man1/ssh.1>
- Cuestión 6:
  - ssh-copy-id, 3. (2016). *3 Steps to Perform SSH Login Without Password Using ssh-keygen & ssh-copy-id*. Thegeekstuff.com. Retrieved 24 November 2016, from <http://www.thegeekstuff.com/2008/11/3-steps-to-perform-ssh-login-without-password-using-ssh-keygen-ssh-copy-id>
- Cuestión 7:
  - *Configura un servidor SSH en Ubuntu para acceder a tu equipo de forma remota*. (2016). RedesZone. Retrieved 24 November 2016, from <http://www.redeszone.net/gnu-linux/servidor-ssh-en-ubuntu/>
  - *Cambiar puerto de SSH en CentOS 7 - Blog de ADW.es*. (2016). Blog de ADW.es. Retrieved 24 November 2016, from <http://blog.adw.es/cambiar-puerto-de-ssh-en-centos-7/>
- Cuestión 8:
  - *HowTo: Restart SSH Service under Linux / UNIX*. (2016). Cyberciti.biz. Retrieved 24 November 2016, from <https://www.cyberciti.biz/faq/howto-restart-ssh/>
- Cuestión opcional 2:
  - *HOWTO fail2ban spanish - Fail2ban*. (2016). Fail2ban.org. Retrieved 24 November 2016, from [http://www.fail2ban.org/wiki/index.php/HOWTO\\_fail2ban\\_spanish](http://www.fail2ban.org/wiki/index.php/HOWTO_fail2ban_spanish)
- Cuestión 9:
  - *How To Install Linux, Apache, MySQL, PHP (LAMP) stack on Ubuntu 16.04 / DigitalOcean*. (2016). Digitalocean.com. Retrieved 25 November 2016, from <https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-16-04>
  - *Install Apache, PHP And MySQL On CentOS 7 (LAMP)*. (2016). Howtoforge.com. Retrieved 25 November 2016, from [https://www.howtoforge.com/apache\\_php\\_mysql\\_on\\_centos\\_7\\_lamp](https://www.howtoforge.com/apache_php_mysql_on_centos_7_lamp)
- Cuestión 10:
  - *Configurar una dirección IP estática*. (2016). Technet.microsoft.com. Retrieved 25 November 2016, from [https://technet.microsoft.com/es-es/library/ff710457\(v=ws.10\).aspx](https://technet.microsoft.com/es-es/library/ff710457(v=ws.10).aspx)
- Cuestión 11:
  - *HowTo Apply a Patch File To My Linux / UNIX Source Code*. (2016). Cyberciti.biz. Retrieved 25 November 2016, from <https://www.cyberciti.biz/faq/apply-patch-file-using-patch-command/>

- Cuestión 12:
  - *How to install Webmin on CentOS 7.* (2016). *lintut.com - Linux Howto's Guide*. Retrieved 25 November 2016, from <http://lintut.com/how-to-install-webmin-on-centos-7/>
- Cuestión 13:
  - *How To Install and Secure phpMyAdmin on Ubuntu 16.04* | *DigitalOcean*. (2016). *Digitalocean.com*. Retrieved 25 November 2016, from <https://www.digitalocean.com/community/tutorials/how-to-install-and-secure-phpmyadmin-on-ubuntu-16-04>
  - *Aumentar el límite de upload en PHP.* (2016). *Cloudi.ng Knowledge Base*. Retrieved 25 November 2016, from <https://clouding.io/kb/aumentar-el-limite-de-upload-en-php/>
- Cuestión 14:
  - *ISPConfig Hosting Control Panel.* (2016). *ISPConfig*. Retrieved 25 November 2016, from <http://www.ispconfig.org/>
- Cuestión 15:
  - *AWK paso a paso... y sin usar el ratón* | *El rincón de Linux.* (2016). *Linux-es.org*. Retrieved 25 November 2016, from <http://www.linux-es.org/node/31>
- Cuestión 16:
  - *7.2. re — Regular expression operations — Python 2.7.12 documentation.* (2016). *Docs.python.org*. Retrieved 25 November 2016, from <https://docs.python.org/2/library/re.html>
  - *Systems Administration — The Hitchhiker's Guide to Python.* (2016). *Docs.python-guide.org*. Retrieved 25 November 2016, from <http://docs.python-guide.org/en/latest/scenarios/admin/>
  - *How to use Fabric in Python.* (2016). *Python For Beginners*. Retrieved 25 November 2016, from <http://www.pythonforbeginners.com/systems-programming/how-to-use-fabric-in-python/>
- Cuestión 17:
  - *Using the Stop-Process Cmdlet.* (2016). *Technet.microsoft.com*. Retrieved 25 November 2016, from <https://technet.microsoft.com/en-us/library/ee177004.aspx>