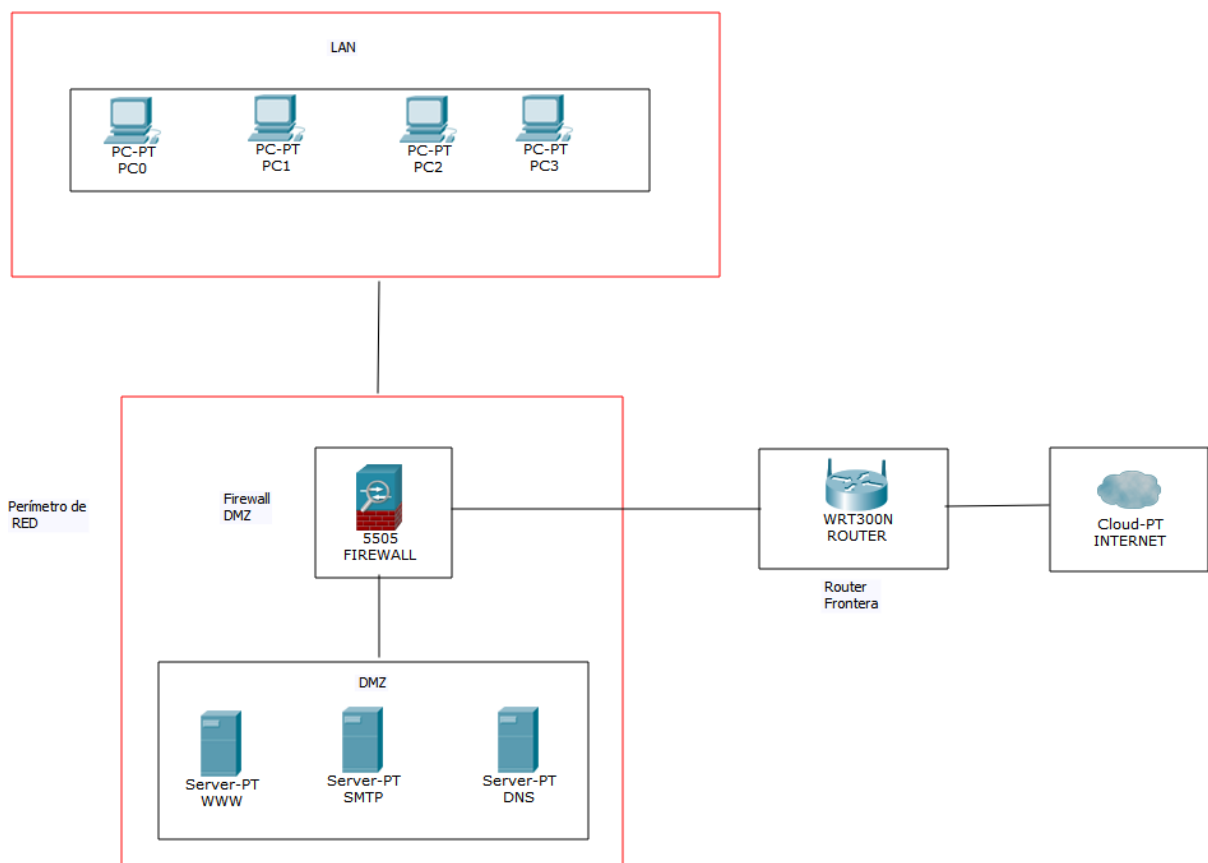


1. Dissenya una xarxa on hi hagi els següents elements:

- a. Router frontera.
- b. Perímetre de xarxa.
- c. DMZ.
- d. LAN.
- e. Firewall.



El **Perímetre de RED** se combina con el uso de las zonas **DMZ**. La zona **DMZ** será accesible desde la **WAN** pero no a la red **LAN**. El acceso o comunicación desde la WAN a la LAN tendrá que ser “autorizado” mediante el **Firewall** que *controla los accesos, aceptandolos o denegando los*.

2. Utilitza Hamachi per demostrar el concepte d'accés remot.

El primer paso es descargar el software. Para ellos nos dirigimos a la dirección web **<http://hamachi.uptodown.com/windows>**.

Nos permite dos tipos diferente de descarga, **con gestión** y **sin gestión**. Emplearemos la segunda opción ya que nos permite formar redes VPN en modo malla (entre otras).

Una vez descargado los instalamos en los equipos que queramos gestionar además del equipo desde el cual vamos a tener acceso a los mismo.

El siguiente paso es la configuración. Podemos crear una red virtual mediante **Red** → **Crear una nueva red** ó directamente clicamos en el botón **Crear una nueva red** como muestro en las capturas de abajo:



Rellenamos los campos de la nueva red que queremos crear:

ID de red: → “SAD-TAREA3”

Contraseña: → “nuestracontraseña”

Confirmar contraseña: → “repetimos nuestracontraseña”

Crear una red

Crear nueva red de cliente (?)

ID de red:
Se utiliza para ubicar la red y unirse a ella.

Contraseña:
Se utiliza para restringir el acceso a la red.

Confirmar contraseña:

Iniciar sesión para crear una nueva red gestionada (?)

Las redes gestionadas pueden administrarse centralmente en la web y ofrecer funcionalidad avanzada, como redes de puerta de enlace o topología de redes de

El siguiente paso es ir al siguiente host y unirlo a la nueva red que acabamos de crear. Para unir un nuevo equipo a la red, una vez abierto el software de **Hamachi** clicamos en **Unirse a una red existente** ó **Red** → **Unirse a una red existente...** . Introducimos los datos de la red y pulsamos en unirse:

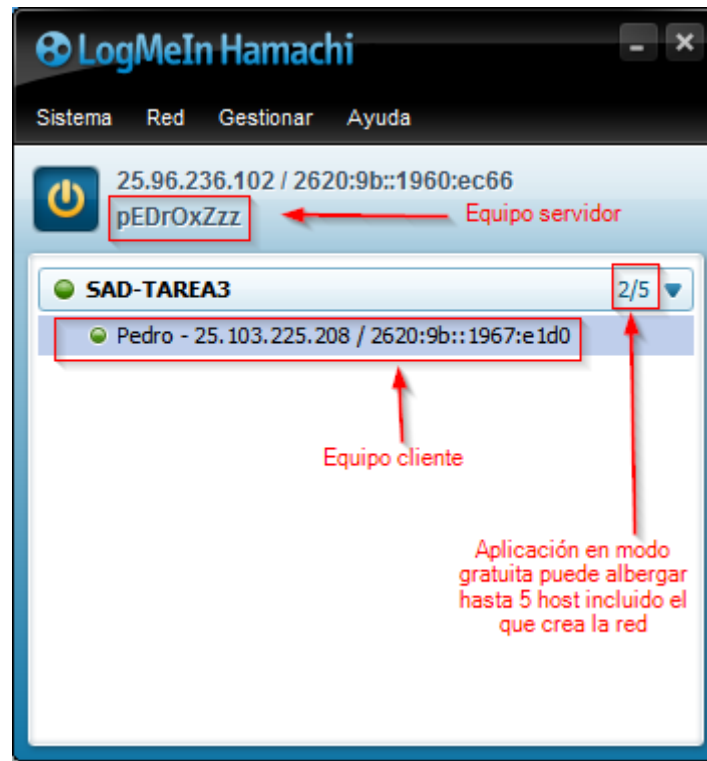


Unirse a una red

ID de red:

Contraseña:
Dejar en blanco si se desconoce.

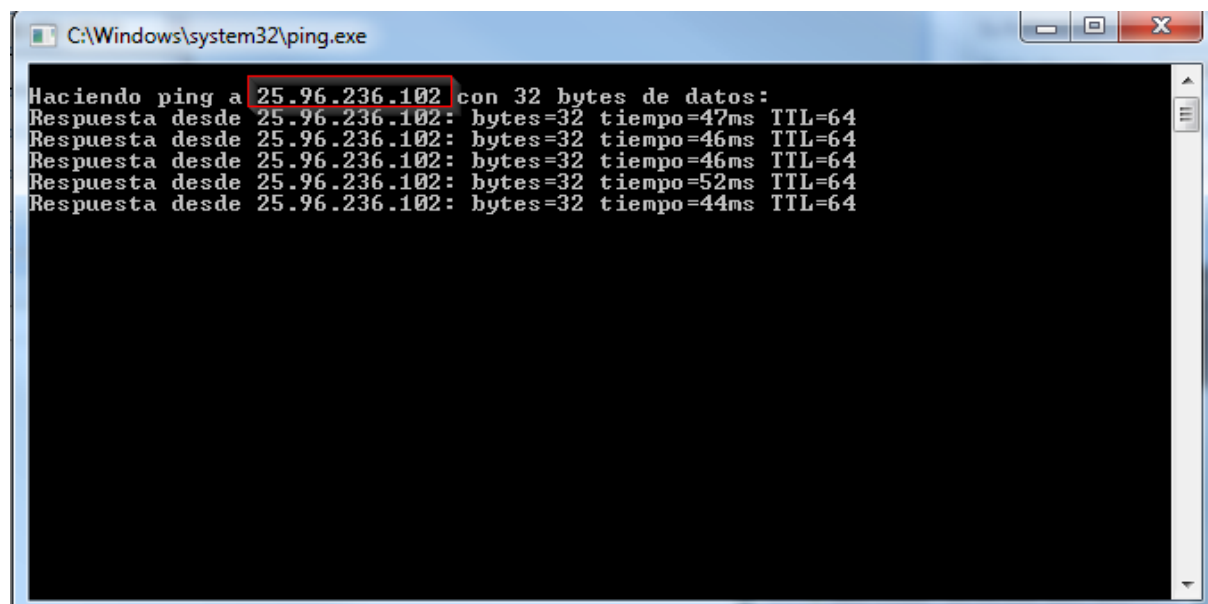
Con cada equipo que añadimos, lo podremos ver en el listado:



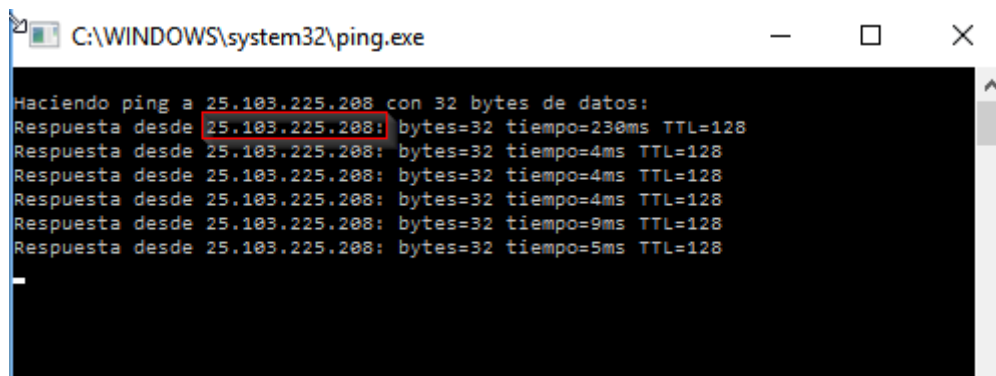
Realizaremos una serie de pruebas para ver si tenemos conexión entre los equipos:

Podemos hacer un ping desde el software de **hamachi** para comprobar que hay conexión:

Desde el cliente:



Desde el servidor:

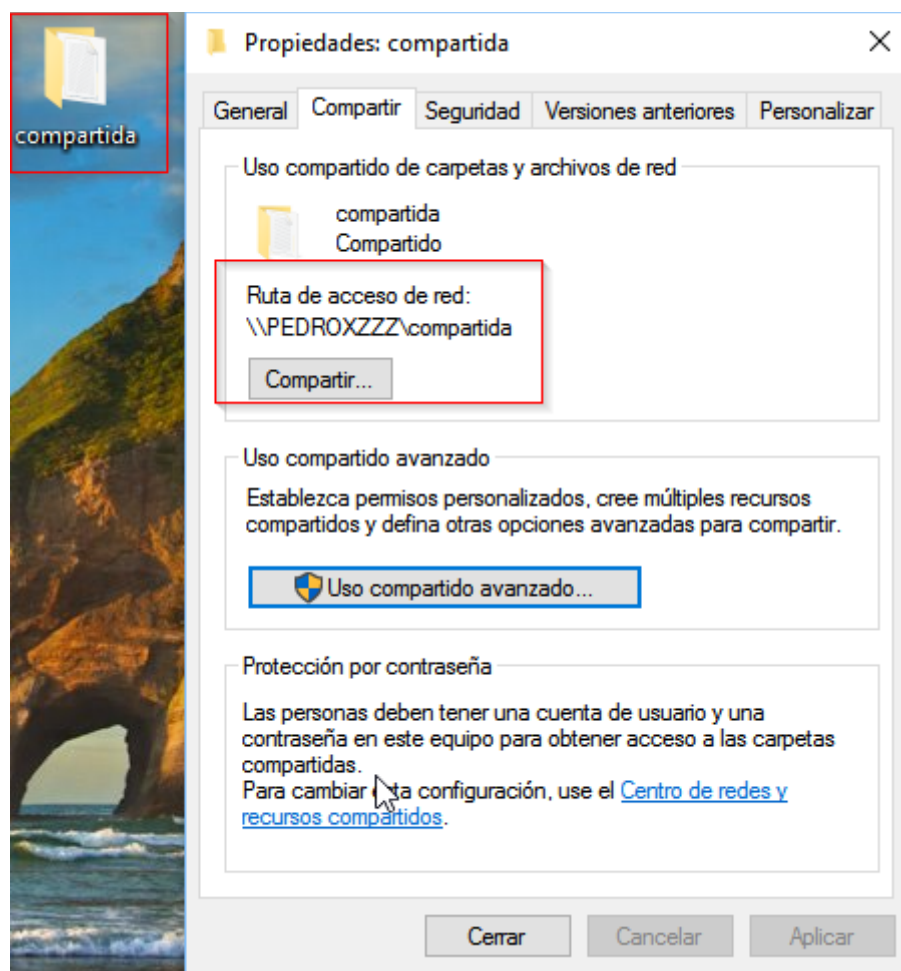


```
C:\WINDOWS\system32\ping.exe

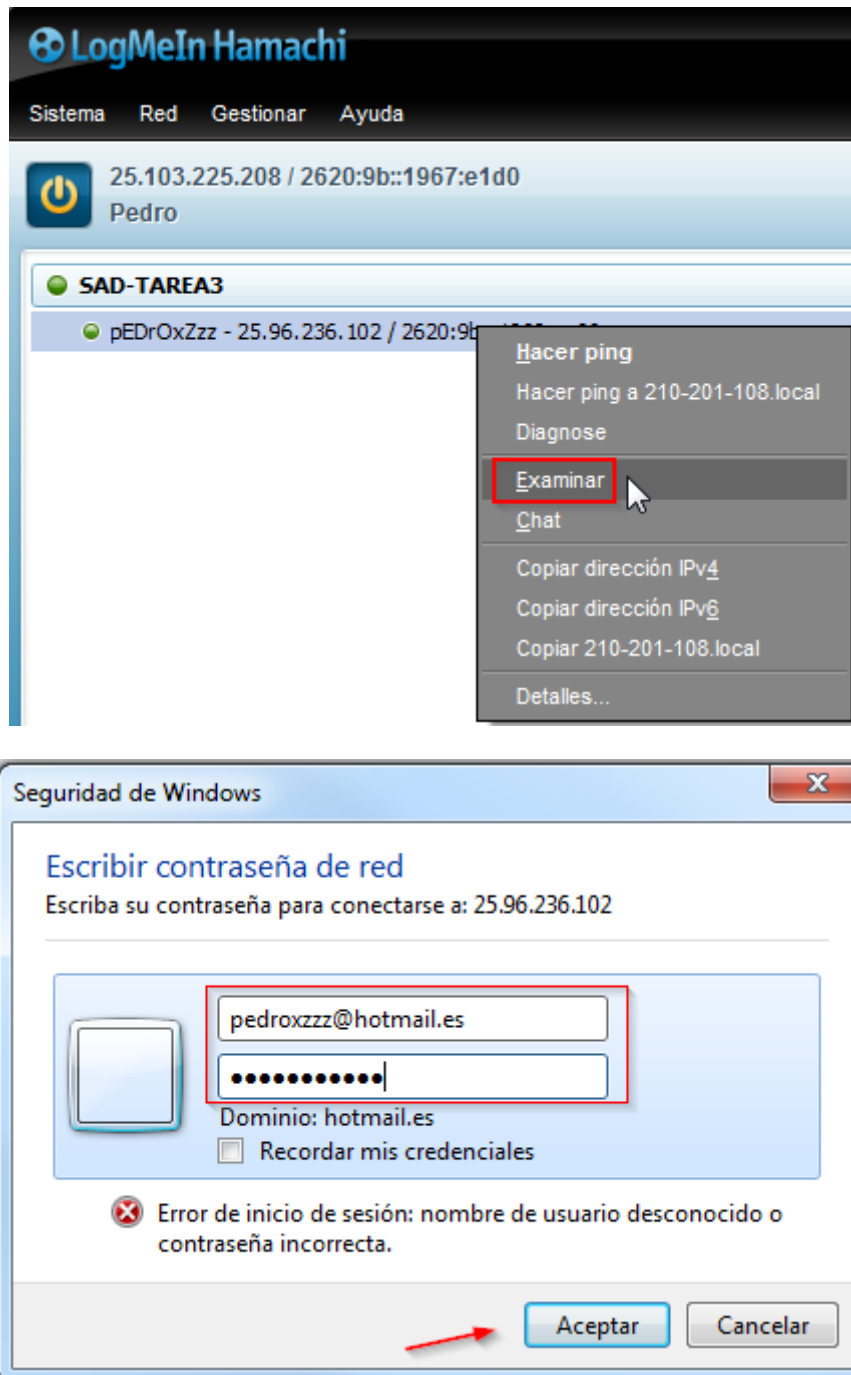
Haciendo ping a 25.103.225.208 con 32 bytes de datos:
Respuesta desde 25.103.225.208: bytes=32 tiempo=230ms TTL=128
Respuesta desde 25.103.225.208: bytes=32 tiempo=4ms TTL=128
Respuesta desde 25.103.225.208: bytes=32 tiempo=4ms TTL=128
Respuesta desde 25.103.225.208: bytes=32 tiempo=4ms TTL=128
Respuesta desde 25.103.225.208: bytes=32 tiempo=9ms TTL=128
Respuesta desde 25.103.225.208: bytes=32 tiempo=5ms TTL=128
```

Desde la aplicación, también podremos acceder a los archivos/carpetas “compartidos/as”, de los host conectados a la misma red.

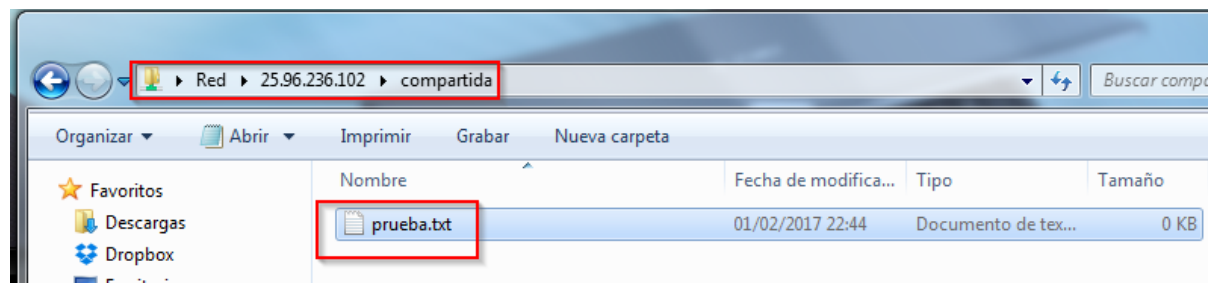
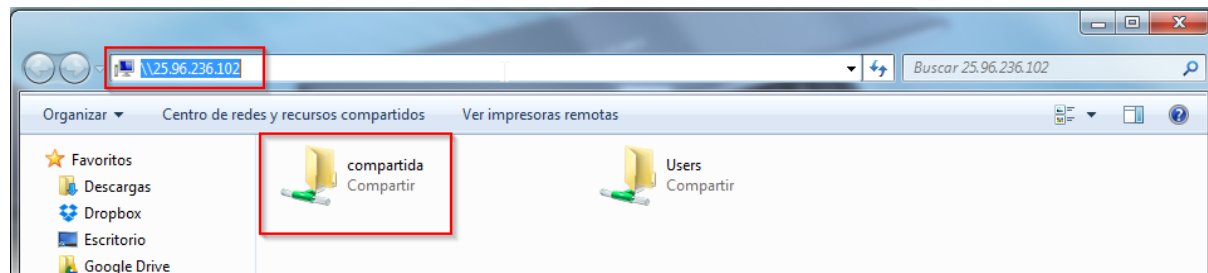
Para hacer la prueba, compartiremos una carpeta en el equipo que actuará como “servidor”:



Ahora nos vamos al equipo cliente y sobre el host clicamos con el botón derecho del ratón y seleccionamos “**Examinar**”:

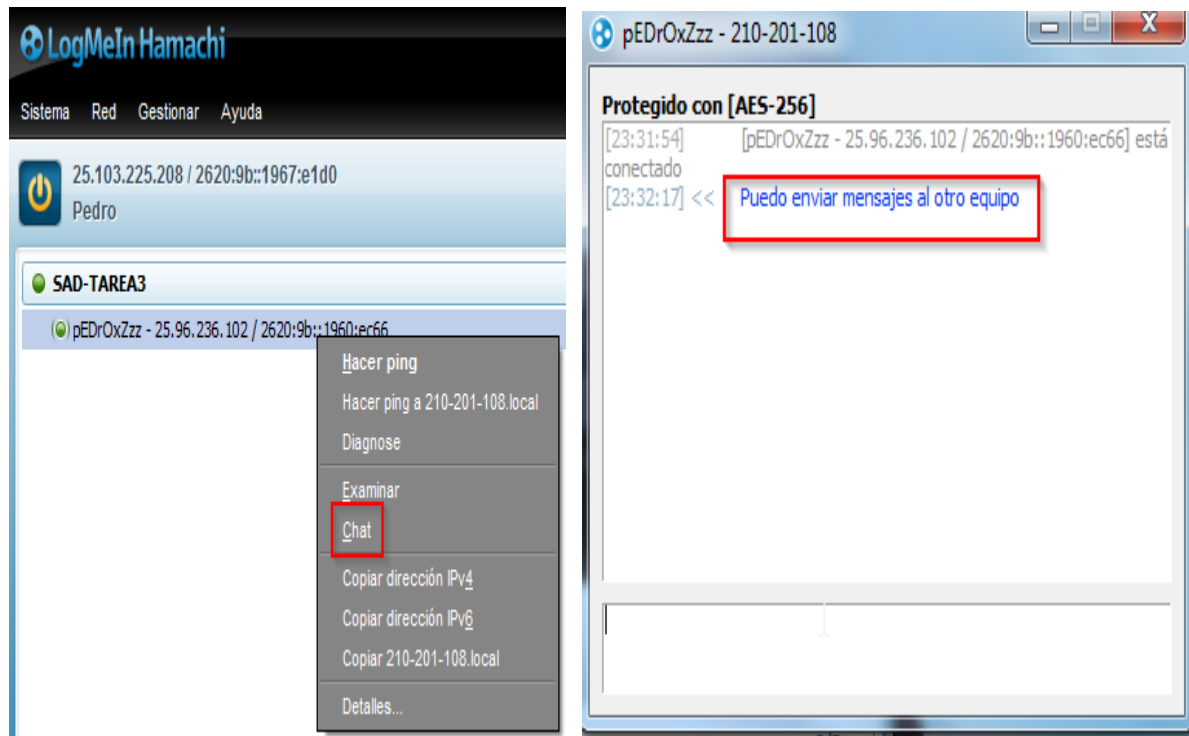


Tras introducir las credenciales de acceso al equipo remoto, podemos ver cómo es detectado por nuestro sistema como un nuevo elemento de la misma red:



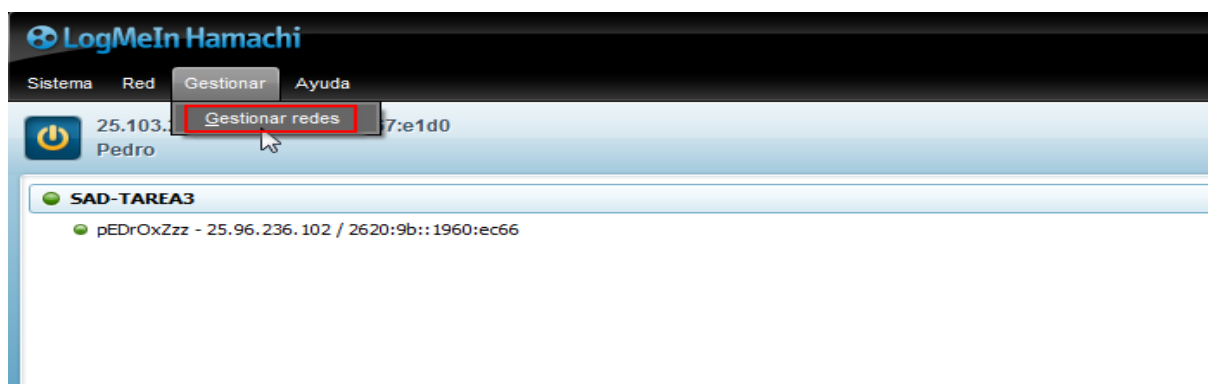
En este punto ya podemos compartir archivos entre las carpetas compartidas y si se da los privilegios oportunos, podremos crear, borrar, mover, etc...

También existe la posibilidad de enviar mensajes entre los hosts de la misma red:

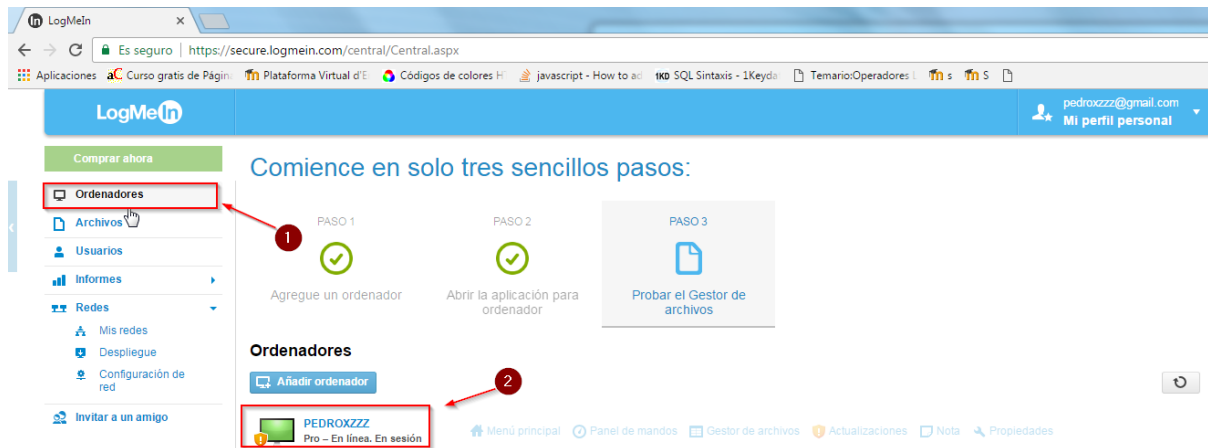


El punto más interesante bajo mi punto de vista, es el acceso remoto mediante **web**. Se puede controlar remotamente la máquina como si fuera tu propio escritorio, además de otras muchas opciones que comentaré brevemente.

Para acceder a la configuración remota por web tenemos que ir al menú de la aplicación **Gestionar** → **Gestionar redes**



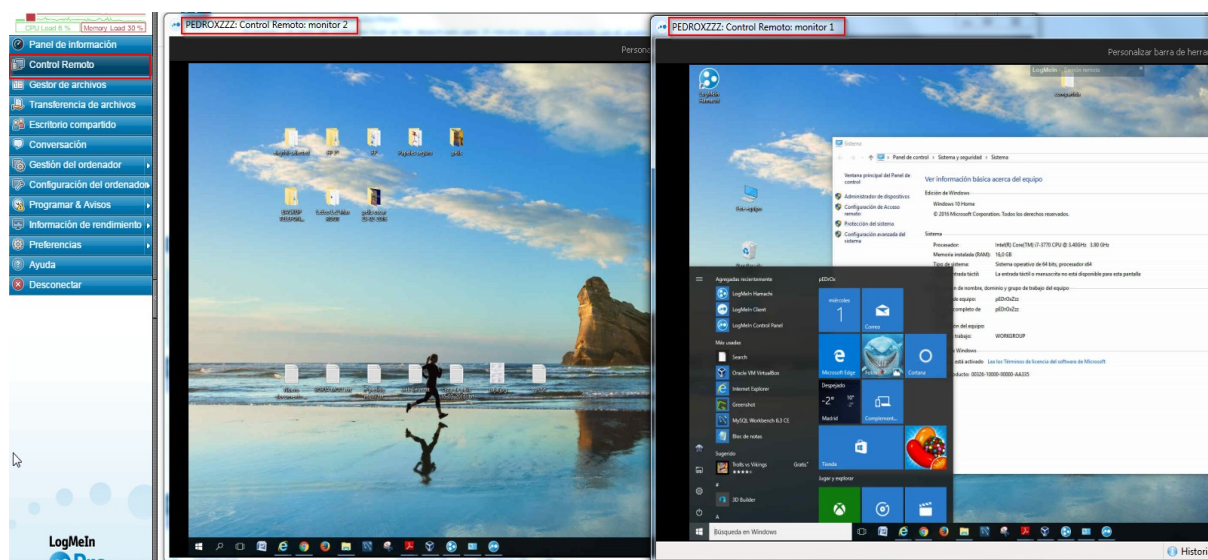
Al seleccionar **Gestionar redes**, automáticamente se abrirá el explorador y nos pedirá el **usuario** y **contraseña** del equipo que queremos controlar. Una vez dentro nos vamos al panel izquierdo de la web y seleccionamos **Ordenadores**:



Seleccionamos el equipo (tiene que estar en línea) y nos volverá a pedir el **usuario** y la **contraseña** del equipo que queremos controlar. Lo introducimos:



Para acceder al escritorio del equipo remoto, en la parte izquierda de la ventana clicamos en **Control Remoto**:



Podemos observar que se han abierto dos ventanas, eso es debido a que dispongo dos monitores. Con esta opción tenemos control total sobre el equipo y podemos trabajar con él como si estuviéramos físicamente delante del mismo, aunque con algunos desfases debidos a la conexión.

Otra opción muy interesante es la del **Panel de información**. Esta opción nos permite ver toda la información referente al equipo que estamos administrando, como por ejemplo el S.O. que tiene instalado, el tráfico de red, los discos físicos del equipo, etc...

Panel de información
Está conectado remotamente a: PEDROXZZZ
Ha iniciado sesión como: pEDROXZZz/Pedro
El teclado y el ratón del ordenador host se han desactivado para 10 minutos [Iniciar conversación con el usuario](#)

Información del sistema
Windows 10 x64 Edition 10.0 (montaje: 14393)
CPU: Intel(R) Core(TM) i7-3770 CPU 7 en 3392 MHz (x 8)
Memoria física: 29% (Total: 16.326,53 MB)
Uso de la memoria: 29% (Total: 18.758,53 MB)
Arrancado por última vez: Hace 5 días, 4 horas, 7 minutos
Usuario interactivo: PEDROXZZZ/Pedro

Tráfico de la red
Gráfico de tráfico de red. Nivel de detalle: 2 segundos. Máx. entrante: 690 kbit/s, Máx. saliente: 1038 kbit/s. Botones: Aplicar, Actualizar.

Unidades de disco
Tabla de unidades de disco:

Unidad	Tamaño	Libre	Uso
C:\	113.670,99 MB	15.441,82 MB	87%
D:\	244.194,99 MB	36.089,14 MB	86%
E:\	1.907.726,99 MB	441.255,78 MB	77%
G:\	1.907.725,99 MB	1.284.174,11 MB	33%

Tareas programadas
Tabla de tareas programadas:

Nombre	Ejecutado por última vez	Siguiente ejecución	Estado
Installation	01/02/2017 23:07		
XblGameSaveTask	01/02/2017 23:03		
RunFullMemoryDiag...	01/02/2017 23:03		
SilentCleanup	01/02/2017 23:03		
ProcessMemoryDiag...	01/02/2017 23:03		

También tenemos la opción de gestionar los archivos del equipo pudiendo moverlos, copiarlos, renombrarlos, eliminarlos, etc..

PEDROXZZZ: Gestor de archivos
Este ordenador: PEDRO | Remoto: PEDROXZZZ

Este ordenador: PEDRO
C:\Users\Bernat

Nombre	Tamaño	Modificado
.dia		15/11/2016 21:19:11
.editix		09/03/2016 21:05:44
.oracle_jre_usage		23/07/2016 18:56:55
.sysdb		29/02/2016 16:41:01
amaya		09/05/2016 14:54:46
Contacts		15/10/2015 2:37:51
Desktop		01/02/2017 23:20:04
Documents		30/04/2016 19:41:51
Downloads		30/04/2016 20:38:41
Dropbox		01/02/2017 19:36:49
Favorites		15/10/2015 2:37:51
Google Drive		29/03/2016 18:40:00

Remoto: PEDROXZZZ
C:\Users\Pedro

Nombre	Tamaño	Modificado
.dia		14/11/2016 21:22:56
.editix		26/02/2016 21:06:12
.fop		23/11/2016 19:16:12
.gem		16/07/2015 18:28:24
.gkrellm2		22/12/2016 21:05:20
.gstreamer-0.10		05/11/2015 21:11:06
.oracle_jre_usage		24/02/2016 19:59:49
.ssh		10/07/2015 20:03:10
.sysdb		24/02/2016 19:59:50
.VirtualBox		01/02/2017 21:14:08
3D Objects		14/12/2015 3:36:18
amaya		14/11/2015 0:37:50

3. Realiza una connexió amb SSH i descriu detalladament el procés que has seguit, incloent captures de pantalla.

Para este punto de la tarea, utilizaremos dos máquinas Ubuntu 16.04 para realizar la conexión por ssh. Utilizaremos una máquina “**cliente**” y otra como “**servidor**”.

Es muy recomendable configurar la red como estática aunque no imprescindible. Mi elección es configurarla como estática.

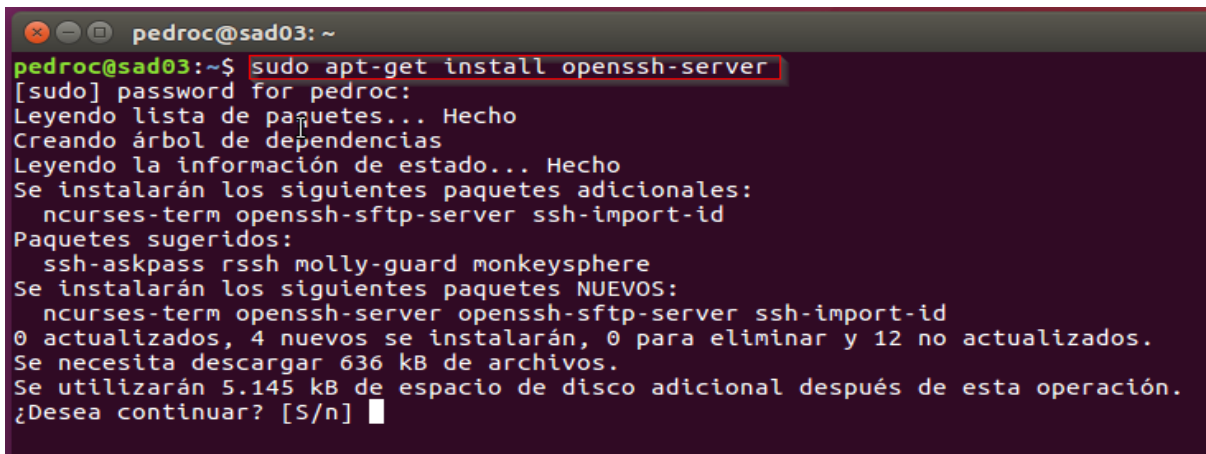
Máquina cliente:

IP: 192.168.2.52/24

Máquina servidor:

IP: 192.168.2.51/24

Una vez configurado la red como estática procedemos a la instalación de **openssh-server** para crear el túnel. Lo instalaremos tanto en la máquina servidor como en la cliente. Para ello abrimos un terminal y ejecutamos la orden **sudo apt-get install openssh-server**:



```
pedroc@sad03: ~  
pedroc@sad03:~$ sudo apt-get install openssh-server  
[sudo] password for pedroc:  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Se instalarán los siguientes paquetes adicionales:  
  ncurses-term openssh-sftp-server ssh-import-id  
Paquetes sugeridos:  
  ssh-askpass rssh molly-guard monkeysphere  
Se instalarán los siguientes paquetes NUEVOS:  
  ncurses-term openssh-server openssh-sftp-server ssh-import-id  
0 actualizados, 4 nuevos se instalarán, 0 para eliminar y 12 no actualizados.  
Se necesita descargar 636 kB de archivos.  
Se utilizarán 5.145 kB de espacio de disco adicional después de esta operación.  
¿Desea continuar? [S/n]
```

Una vez instalado pasamos a su configuración en la máquina que actuará como servidor. Abrimos un terminal y editamos el archivo de configuración de **openssh**. El archivo en cuestión se llama **sshd_config** y se encuentra en la ruta **/etc/ssh/**. Al final del archivo escribimos la línea **AllowUsers nombreusuario** (como nombre de usuario utilizaremos el que pusimos durante la instalación del S.O (**pedros**), podría ser cualquier otro que estuviera creado en el equipo servidor. Para crear un usuario no tenemos más que abrir un terminal es escribir la orden **sudo adduser nombreusuario**. Por seguridad también podemos cambiar el puerto de escucha de ssh que por defecto es el **22** al **5252** aunque este paso lo realizaremos más adelante por comodidad:

```
pedros@sad03: ~
GNU nano 2.5.3 Archivo: /etc/ssh/sshd_config Modificado
# be allowed through the ChallengeResponseAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via ChallengeResponseAuthentication may bypass
# the setting of "PermitRootLogin without-password".
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and ChallengeResponseAuthentication to 'no'.
UsePAM yes
AllowUsers pedros
```

Reiniciamos ssh para que los cambios tengan efecto:

`sudo /etc/init.d/ssh restart/reload` ó `sudo service ssh restart/reload`

```
pedros@sad03: ~
pedros@sad03:~$ sudo nano /etc/ssh/sshd_config
pedros@sad03:~$ sudo /etc/init.d/ssh restart
[ ok ] Restarting ssh (via systemctl): ssh.service.
pedros@sad03:~$
```

Ahora nos vamos a la máquina cliente y generamos el par de claves o la clave asimétrica que posteriormente tendremos que pasarla al servidor y así ahorrarnos tener que introducir claves constantemente. Para ello abrimos un terminal y ejecutamos la orden **`ssh-keygen`**:

```

pedroc@sad03: ~
pedroc@sad03:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/pedroc/.ssh/id_rsa): sad03
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in sad03.
Your public key has been saved in sad03.pub.
The key fingerprint is:
SHA256:Gs31gqM30FLerZ40Rxd1yA09u/frQTvs6l8lkBLZ7w pedroc@sad03
The key's randomart image is:
+---[RSA 2048]---+
|      .ooB == |
|      . oo*.==|
|      oo .+. =|
|      + =+... E|
|      o So.+o .+|
|      = .o. . o+|
|      o o   .o+|
|      . .   .++|
|      oo o |
+-----[SHA256]-----+
pedroc@sad03:~$

```

Una vez generadas el par de claves tenemos que copiar la clave pública al servidor. Para ello abrimos un terminal y ejecutamos:

scp nombreclave.pub usuarioservidor@ipservidor:/carpeta

```

root@sad03: /home/pedroc
root@sad03:/home/pedroc# scp sad03.pub pedros@192.168.2.51:~/.ssh/
The authenticity of host '192.168.2.51 (192.168.2.51)' can't be established.
ECDSA key fingerprint is SHA256:U7/htJ+vFiQfwbHhZRUaiM0kIKB+l86DTMQadzqEZKE.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.2.51' (ECDSA) to the list of known hosts.
pedros@192.168.2.51's password:
sad03.pub                                100% 394      0.4KB/s   00:00
root@sad03:/home/pedroc#

```

Ahora si nos vamos al servidor a la carpeta “.ssh” observamos que se ha copiado la clave pública:

```

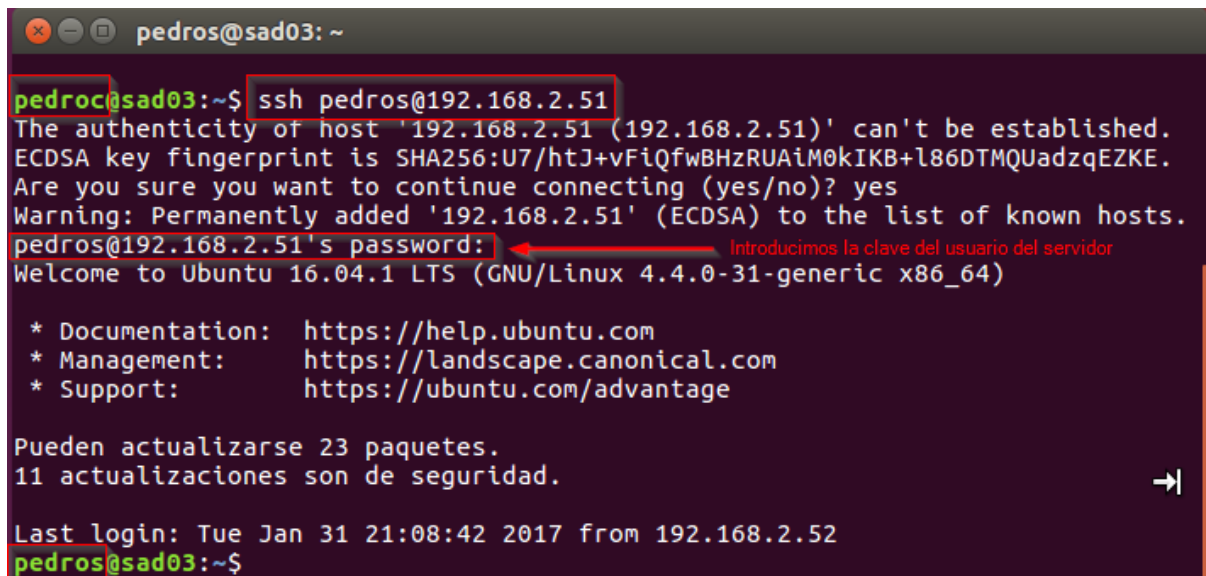
pedros@sad03: ~/.ssh
pedros@sad03:~/.ssh$ ls
id_rsa  id_rsa.pub  sad03.pub
pedros@sad03:~/.ssh$

```

El siguiente paso es copiar esa clave pública en el fichero donde se almacenan las claves autorizadas. Este paso lo podemos hacer

directamente en el servidor o conectándonos desde el cliente al servidor. En mi caso he elegido la segunda opción. Para ello abrimos un terminal en la máquina cliente y nos conectamos por **ssh** al equipo servidor con la siguiente orden:

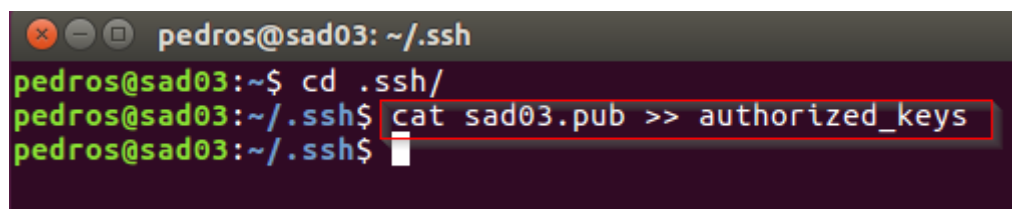
ssh usuario servidor@ipServidor



```
pedros@sad03: ~  
pedros@sad03:~$ ssh pedros@192.168.2.51  
The authenticity of host '192.168.2.51 (192.168.2.51)' can't be established.  
ECDSA key fingerprint is SHA256:U7/htJ+vFiQfWBHzRUAiM0kIKB+l86DTMQUadzqEZKE.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '192.168.2.51' (ECDSA) to the list of known hosts.  
pedros@192.168.2.51's password: ← Introducimos la clave del usuario del servidor  
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 23 paquetes.  
11 actualizaciones son de seguridad.  
  
Last login: Tue Jan 31 21:08:42 2017 from 192.168.2.52  
pedros@sad03:~$
```

Ahora para almacenar la clave pública en el fichero **authorized_keys** ejecutamos desde la conexión remota al servidor la siguiente orden:

Cat archivoclavepublica.pub >> authorized_keys



```
pedros@sad03: ~/.ssh  
pedros@sad03:~$ cd .ssh/  
pedros@sad03:~/.ssh$ cat sad03.pub >> authorized_keys  
pedros@sad03:~/.ssh$
```

Cerramos la sesión ssh y volvemos a conectar. Esta vez nos pedirá la clave pública en vez de la clave del usuario del servidor:


```

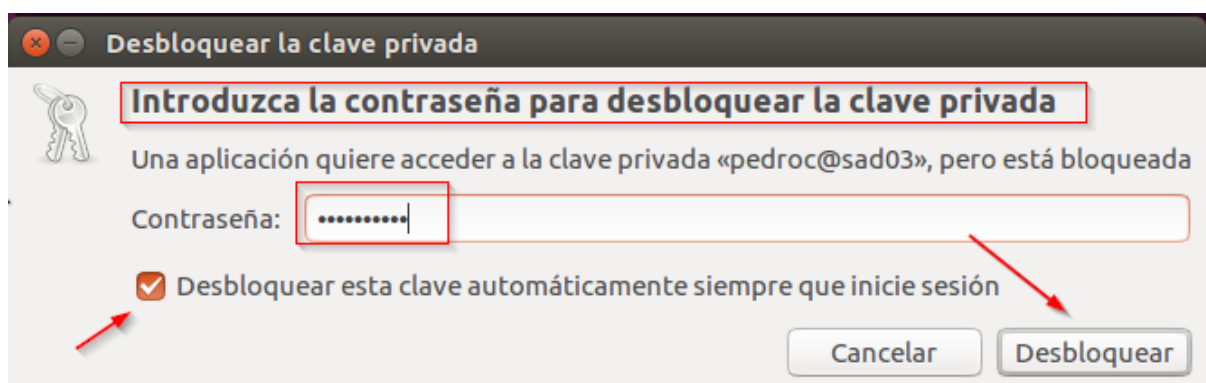
pedros@sad03: ~
pedros@sad03:~$ cd .ssh/
pedros@sad03:~/.ssh$ cat sad03.pub >> authorized_keys
pedros@sad03:~/.ssh$ sudo /etc/init.d/ssh reload
[sudo] password for pedros:
[ ok ] Reloading ssh configuration (via systemctl): ssh.service.
pedros@sad03:~/.ssh$ exit
logout
Connection to 192.168.2.51 closed.
pedros@sad03:~$ ssh pedros@192.168.2.51
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 23 paquetes.
11 actualizaciones son de seguridad.

Last login: Tue Jan 31 21:29:41 2017 from 192.168.2.52
pedros@sad03:~$

```



Una vez introducida la clave, ya no tendremos que volver a escribirla, siempre y cuando utilicemos el mismo equipo cliente que es el que contiene las claves. Incluso podemos retirar del archivo de configuración de **openssh** la línea "AllowUsers pedros" que añadimos al final del mismo. Ya que estamos cambiaremos el puerto de escucha y haremos unas comprobaciones que demuestre que estamos conectados:

```

pedros@sad03: ~/.ssh
GNU nano 2.5.3 Archivo: /etc/ssh/sshd_config Modificado
# Package generated configuration file
# See the sshd_config(5) manpage for details

# What ports, IPs and protocols we listen for
Port 5252
# Use these options to restrict which interfaces/protocols sshd will bind to

```

```

pedros@sad03: ~/.ssh
GNU nano 2.5.3 Archivo: /etc/ssh/sshd_config Modifica
# be allowed through the ChallengeResponseAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via ChallengeResponseAuthentication may bypass
# the setting of "PermitRootLogin without-password".
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and ChallengeResponseAuthentication to 'no'.
UsePAM yes
AllowUsers pedros

```

Eliminamos esta línea

...y reiniciamos el servicio: **sudo /etc/init.d/ssh restart/reload**

```

pedros@sad03: ~
pedroc@sad03:~$ ssh pedros@192.168.2.51 -p 5252
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 23 paquetes.
11 actualizaciones son de seguridad.

Last login: Tue Jan 31 21:30:08 2017 from 192.168.2.52
pedros@sad03:~$
pedros@sad03:~$

```

Podemos observar que ya no pide autenticación

Una vez conectados al servidor ejecutamos **ifconfig** y lo comparamos con el mismo comando desde otro terminal. Podemos observar que las direcciones IP son las del **servidor 192.168.2.51** y **cliente 192.168.2.52**:

```

pedros@sad03: ~
pedros@sad03:~$ ifconfig
enp0s3  Link encap:Ethernet direcciónHW 08:00:27:99:b6:c8
        Direc. inet:192.168.2.51 Difus.:192.168.2.255 Másc:255.255.255.0
        Dirección inet6: fe80::789c:5c56:6ac4:d04c/64 Alcance:Enlace
        ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
        Paquetes RX:3473 errores:0 perdidos:0 overruns:0 frame:0
        Paquetes TX:685 errores:0 perdidos:0 overruns:0 carrier:0
        colisiones:0 long.colaTX:1000
        Bytes RX:523635 (523.6 KB) TX bytes:91376 (91.3 KB)

pedroc@sad03: ~
pedroc@sad03:~$ ifconfig
enp0s3  Link encap:Ethernet direcciónHW 08:00:27:d3:93:04
        Direc. inet:192.168.2.52 Difus.:192.168.2.255 Másc:255.255.255.0
        Dirección inet6: fe80::730c:71ef:39c0:c690/64 Alcance:Enlace
        ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
        Paquetes RX:3242 errores:0 perdidos:0 overruns:0 frame:0
        Paquetes TX:1033 errores:0 perdidos:0 overruns:0 carrier:0
        colisiones:0 long.colaTX:1000
        Bytes RX:506677 (506.6 KB) TX bytes:123446 (123.4 KB)

```


Ahora desde el cliente conectado al servidor vamos a crear un archivo de texto en su escritorio y después comprobaremos si se ha creado:

The first screenshot shows a terminal window with the command `sudo nano /home/pedros/Escritorio/archivoComprobacion.txt` being entered. The second screenshot shows the nano editor interface with the file path `/home/pedros/Escritorio/archivoComprobacion.txt` and a status bar indicating the connection. The third screenshot shows a file manager window displaying the created file `archivoComprobacion.txt` and a terminal window showing the output of the `ifconfig` command, highlighting the IP address `192.168.2.51`.

```

pedros@sad03: ~
pedros@sad03:~$ sudo nano /home/pedros/Escritorio/archivoComprobacion.txt
pedros@sad03:~$

GNU nano 2.5.3 Archivo: /home/pedros/Escritorio/archivoComprobacion.txt Modificado
Estoy conectado por SSH desde el cliente con IP 192.168.2.52 al servidor con IP 192.168.2.51

archivoComprobacion.txt
archivoComprobacion.txt [Solo lectura]
Abrir Guardar
Estoy conectado por SSH desde el cliente con IP 192.168.2.52 al servidor con IP 192.168.2.51
pestaña: 8 Ln 1, Col 1 INS

pedros@sad03: ~
pedros@sad03:~/.ssh$ cd
pedros@sad03:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,MULTICAST> mtu 1500
        Link encap:Ethernet direcciónHW 08:00:27:99:b6:c8
        Direc. inet:192.168.2.51 Difus.:192.168.2.255 Másc:255.255.255.0

```

4. Descriu com utilitzar SSHFS en un equip amb Linux Ubuntu per muntar un directori remot en el sistema local.

Lo primero que necesitamos es tener instalado **OpenSSH-Server**, pero como vamos a utilizar las máquinas “cliente-servidor” del anterior punto esto ya lo tenemos.

Lo siguiente será instalar el paquete **sshfs** en el equipo **cliente**. Para ellos ejecutamos la siguiente orden en un terminal:

sudo apt-get install sshfs

The screenshot shows a terminal window with the command `sudo apt-get install sshfs` being entered.

```

pedroc@sad03: ~
pedroc@sad03:~$ sudo apt-get install sshfs

```

Ahora nos vamos al equipo **servidor** y creamos el directorio que utilizaremos como directorio remoto e introduciremos un archivo de prueba en formato “txt”:

```
root@sad03: /home/pedros/Documentos
root@sad03: /home/pedros/Documentos# mkdir directorioREMOTO
root@sad03: /home/pedros/Documentos#
```

...Creamos el archivo de prueba con el nombre **archivoREMOTO.txt**:

```
root@sad03: /home/pedros/Documentos/directorioREMOTO
root@sad03: /home/pedros/Documentos# ls
directorioREMOTO
root@sad03: /home/pedros/Documentos# cd directorioREMOTO/
root@sad03: /home/pedros/Documentos/directorioREMOTO# touch archivoREMOTO.txt
root@sad03: /home/pedros/Documentos/directorioREMOTO# ls
archivoREMOTO.txt
root@sad03: /home/pedros/Documentos/directorioREMOTO# nano archivoREMOTO.txt
Lo editamos
```

```
GNU nano 2.5.3 Archivo: archivoREMOTO.txt Modificado
Tarea para SAD03 por Pedro Antonio Ruiz Martínez
```

Volvemos al equipo cliente y creamos la carpeta en el cliente:

```
root@sad03: /home/pedroc/Documentos
root@sad03: /home/pedroc/Documentos# mkdir directorio-Cli-REMOTO
root@sad03: /home/pedroc/Documentos# LS
El programa «LS» no está instalado. Puede instalarlo escribiendo:
apt install sl
root@sad03: /home/pedroc/Documentos# ls
directorio-Cli-REMOTO
root@sad03: /home/pedroc/Documentos#
```

Lo siguiente es montar la carpeta remota en el cliente. Para este paso hay que saber la IP del equipo servidor para añadirla posteriormente, pero si recordamos en el punto 3 de la tarea, le otorgo la IP fija **192.168.2.51**.

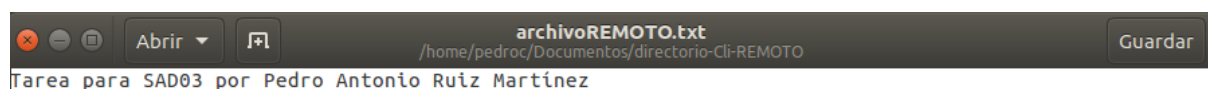
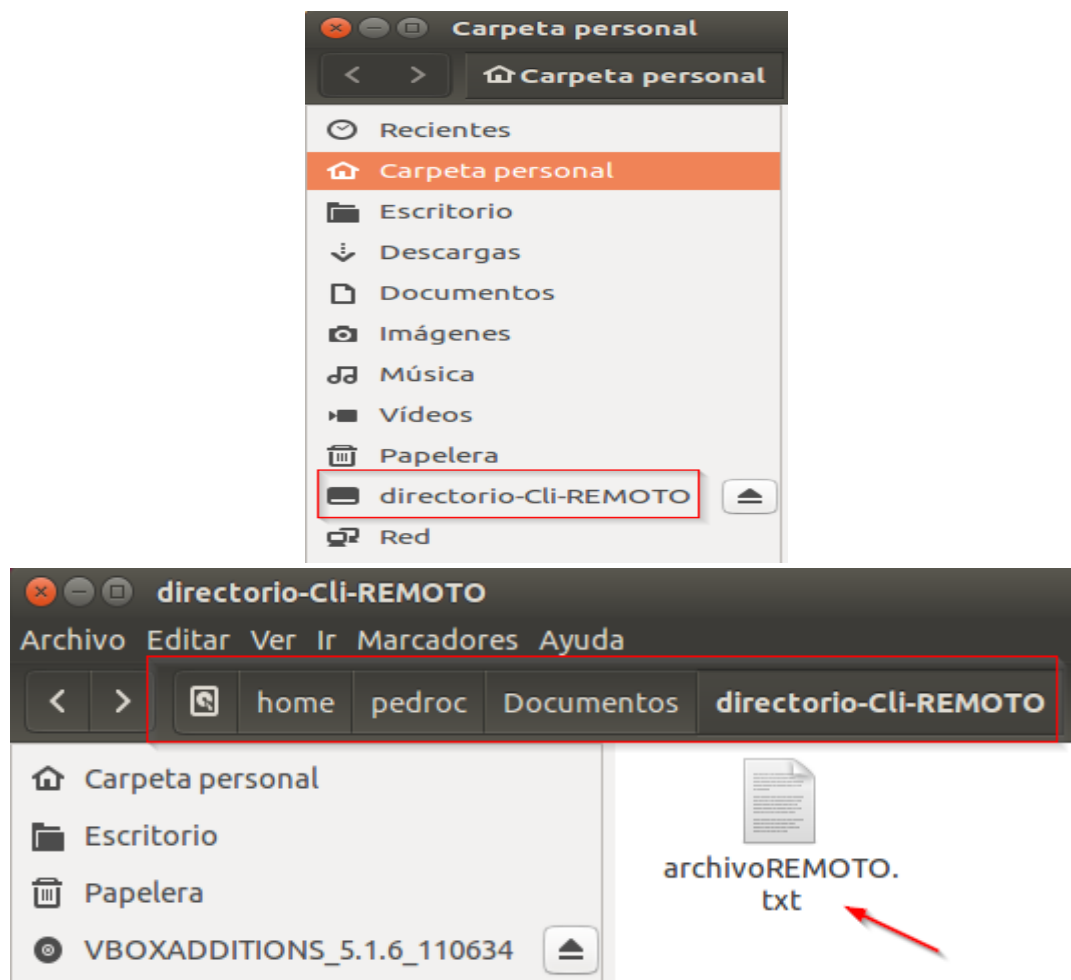
Sabiendo la IP del servidor que contiene la carpeta que deseamos montar en remoto, ejecutaremos la siguiente orden en un terminal desde el equipo **cliente**:

sshfs pedros@192.168.2.51:/home/pedros/Documentos/directorioREMOTO /home/pedroc/Documentos/directorio-Cli-REMOTO

```
root@sad03: /home/pedroc/Documentos
root@sad03: /home/pedroc/Documentos# sshfs pedros@192.168.2.51:/home/pedros/Documentos/directorioREMOTO /home/pedroc/Documentos/directorio-Cli-REMOTO/
pedros@192.168.2.51's password:
root@sad03: /home/pedroc/Documentos#
```

Ahora solo nos queda hacer la prueba de funcionamiento de la carpeta compartida. Para ello, ejecutaré "nautilus" como superusuario en el equipo cliente y accederé a la ruta del directorio donde hemos montado la carpeta remota. Una vez dentro veremos si contiene el archivo con nombre **archivoREMOTO.txt** que creamos en el servidor:

```
root@sad03: /home/pedroc/Documents/directorio-Cli-REMOTO
root@sad03: /home/pedroc/Documents/directorio-Cli-REMOTO# ls
archivoREMOTO.txt
root@sad03: /home/pedroc/Documents/directorio-Cli-REMOTO#
```



Cómo podemos ver en las capturas anteriores, el cliente está dentro de su carpeta local creada anteriormente y puede ver el contenido de la carpeta del servidor.