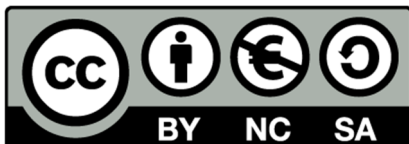

TEMA 3

Gestión de usuarios, NFS y LDAP

NFS



P. Ruíz

<http://somebooks.es/sistemas-operativos-red-2a->

1. [Introducción](#)
2. [Instalar NFS en un servidor Ubuntu 14.04 LTS](#)
 - Comprobar la instalación
3. [Instalar NFS en un cliente con Ubuntu 14.04 LTS](#)
4. [Configurar el servidor NFS](#)
 - Crear las carpetas a compartir
 - Exportar el contenido de las carpetas
 - Iniciar/reiniciar el servicio NFS
 - Crear un archivo en una carpeta compartida
5. [Acceder a la carpeta compartida con NFS desde un cliente con Ubuntu 14.04](#)
 - Crear el punto de montaje para las carpetas compartidas
 - Realizar el montaje de las carpetas compartidas y comprobarlo
 - Crear archivos en las carpetas compartidas
 - Montar automáticamente las carpetas compartidas al iniciar el cliente.
6. [Problemas con los permisos en NFS](#)

3.1. Introducción



Las siglas *NFS* provienen del inglés *Network File System* (*Sistema de archivos de red*). Se trata de un protocolo, implementado en 1984 por la empresa *Sun Microsystems*, que se utiliza en redes de área local para crear un sistema de archivos distribuido.

Sun Microsystems desarrolló *NFS* como un estándar abierto y lo incluyó en la publicación *Request for Comments* (*RFC*) para que cualquiera pudiese implementarlo.

RCF es una publicación de la *Internet Engineering Task Force* (*IETF*)

El objetivo de *NFS* es que varios usuarios (o programas) de una red local puedan acceder a archivos y directorios compartidos como si fuesen locales. De esta forma, se puede centralizar la capacidad de almacenamiento de la red, pudiendo ser más reducida en los clientes.

Para lograrlo, la instalación de *NFS* se divide en dos partes:

- Un equipo que actúa como servidor y que almacena los archivos compartidos.
- Uno o varios equipos que actúan como clientes y cuyos usuarios accederán a los archivos compartidos por el servidor como si fuesen locales.

El protocolo *NFS* actúa en el nivel de aplicación del *modelo OSI*.

Actualmente, el protocolo *NFS* está incluido en la mayoría de las distribuciones *Linux*, y en las diferentes versiones del sistema operativo *OSX* de *Apple*.

En los sistemas *Microsoft*, la situación es algo más confusa: Para las versiones *2000*, *2003* y *XP*, había que descargar e instalar el paquete *Windows Services for UNIX*. En *Windows 8* ya se incluye de fábrica, pero sólo en la versión *Enterprise edition* y, hasta el momento de escribir este documento, no existe forma de instalarlo en otras versiones más modestas. De hecho, los usuarios que actualizan desde las ediciones *Windows 7 Ultimate* o *Windows 7 Enterprise* a la edición *Windows 8 Pro*, no podrán seguir usando *NFS*.

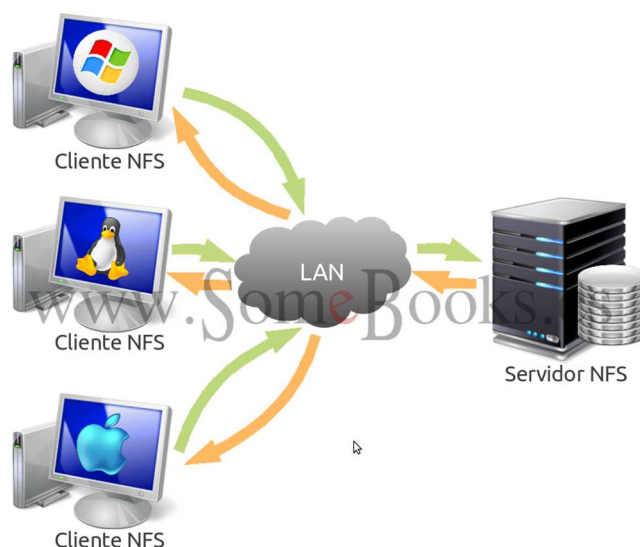
Una alternativa para esta situación será recurrir a software de terceros (esta es la propuesta que seguiremos aquí).

Cuando implementamos NFS en una red local obtenemos las siguientes ventajas:

- Al facilitar el acceso centralizado a la información, se evita la duplicidad de la misma en diferentes puntos de la red.
- De forma predeterminada, obliga a que todas las operaciones de escritura relacionadas con una actualización concluyan antes de continuar (incluida la actualización de la estructura de directorios). Así se asegura la integridad de los datos.
- Permite almacenar todo el perfil de los usuarios en el servidor (su directorio /home), de modo que cualquier usuario podrá acceder a sus datos desde cualquier lugar de la red.
- Permite compartir dispositivos de almacenamiento completos (como unidades ópticas, discos externos, memorias flash, etc), lo que facilita la reducción de costes en este tipo de dispositivos a la vez que mejora su aprovechamiento.
- Desde la versión 4, se incluyen características de seguridad *Kerberos* y *Listas de Control de Acceso (ACL – Access Control List)*, entre otras.

Como hemos dicho más arriba, para que NFS funcione en una red, debemos instalar dos componentes: uno en el servidor, que se encargará de ofrecer los recursos compartidos y otro en cada cliente que pretenda utilizar estos recursos.

OA continuación veremos cómo se instala NFS en un servidor *Ubuntu 14.04 LTS*. Después, veremos cómo se realiza la instalación en los equipos cliente, tanto en *Ubuntu 14.04 LTS* como en *Windows 8.1*. De esta forma, cuando terminemos la instalación, estaremos listos para comenzar a configurar nuestro sistema de archivos de red.



3.2. Instalar NFS en un servidor Ubuntu 14.04 LTS

El paquete que nos permite compartir directorios NFS se llama **nfs-kernel-server**, por lo que será el primer paquete a instalar. Además, necesitaremos el paquete **nfs-common**, que contiene los programas que nos permitirán usar NFS, tanto en el lado cliente como en el lado servidor. Entre ellos se encuentran los comandos **lockd**, **statd**, **showmount** y **nfsstat**.

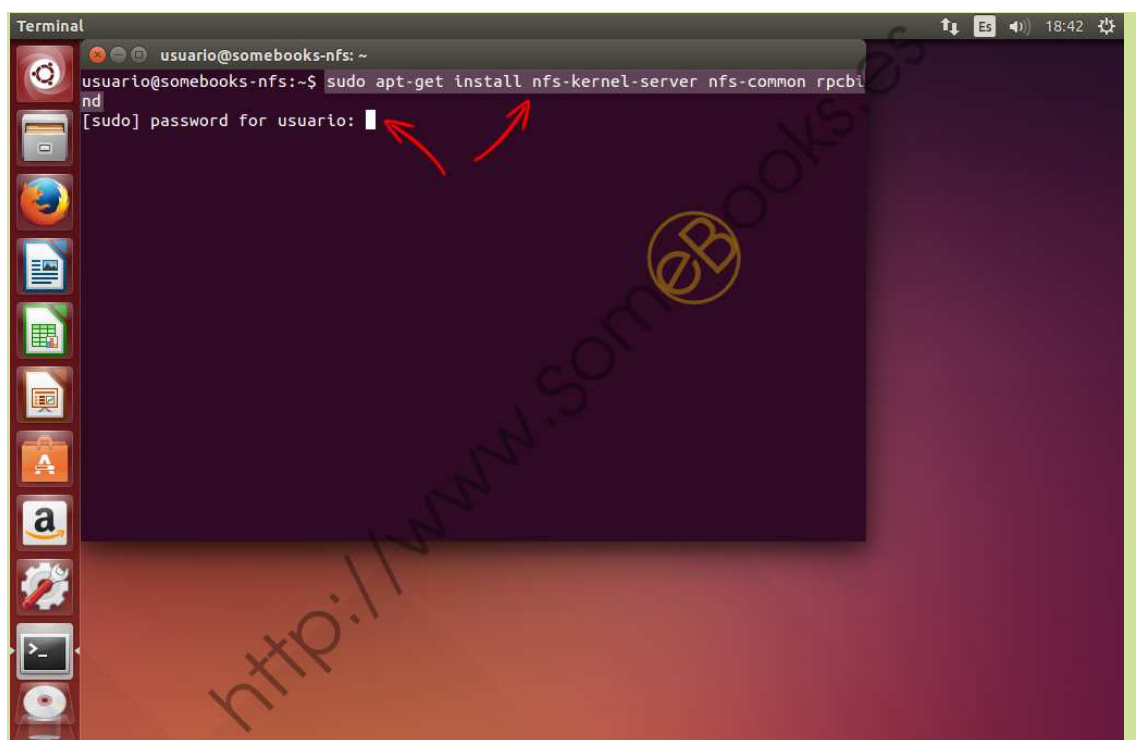
Por último, necesitaremos el paquete **rpcbind**, un servicio que convierte los identificadores de programa *RPC (Remote Procedure Call)* en direcciones universales

La instalación en el servidor es tan sencilla como abrir una ventana de terminal y ejecutar el comando que instala los paquetes anteriores:

```
sudo apt-get install nfs-kernel-server nfs-common rpcbind
```

1

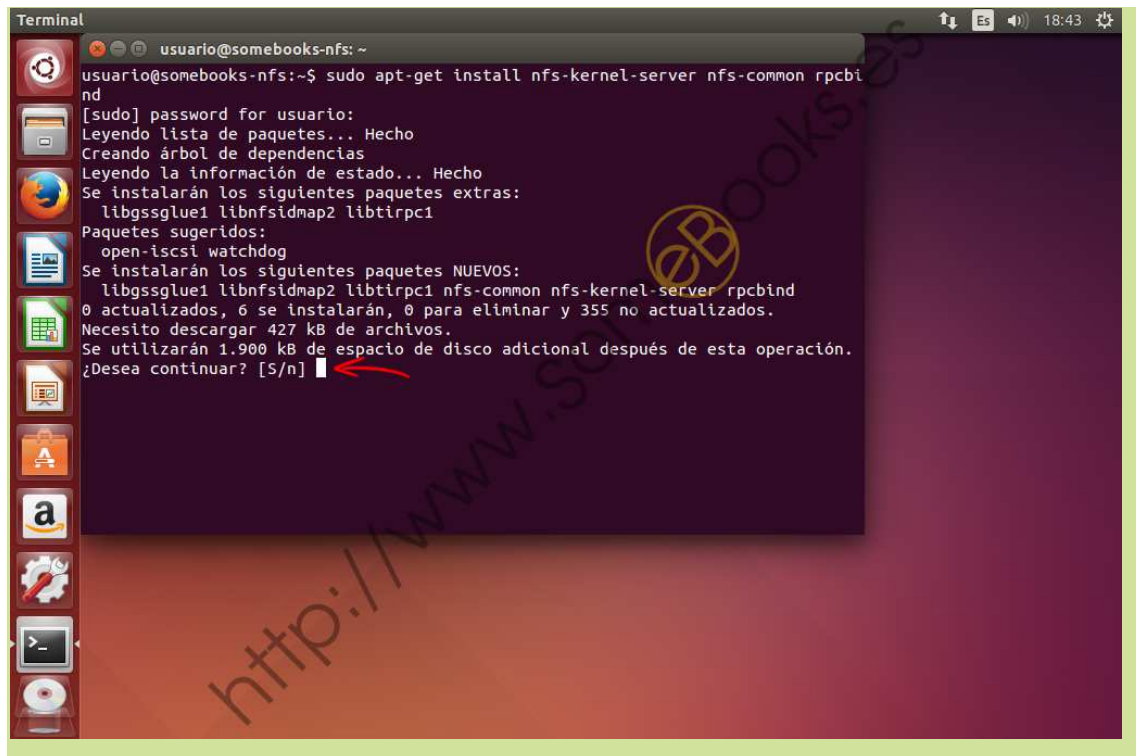
Lógicamente, para instalar deberemos escribir la contraseña de administración.



El sistema nos informa de otros paquetes, que necesitaremos añadir, para que la instalación sea satisfactoria (dependencias) y qué espacio de almacenamiento deberá utilizar en el disco.

2

Para continuar, deberemos pulsar la tecla **S** y, a continuación, la tecla **Intro**.



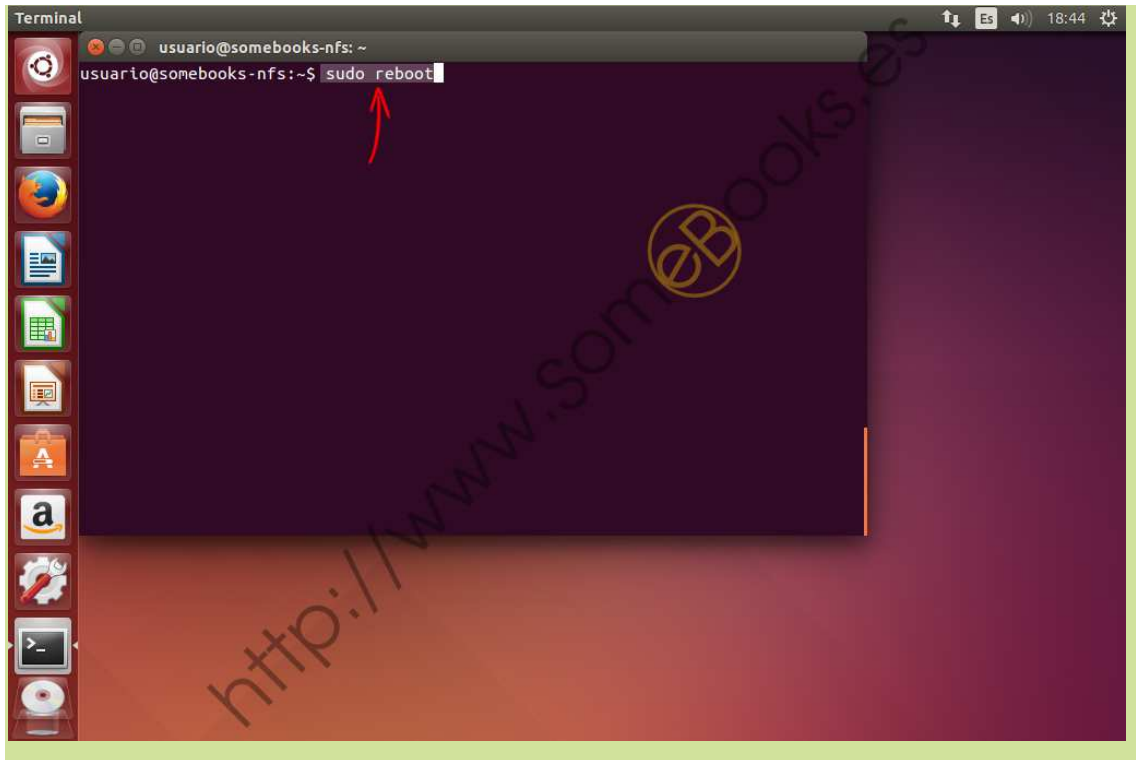
```
Terminal
usuario@somebooks-nfs: ~
usuario@somebooks-nfs:~$ sudo apt-get install nfs-kernel-server nfs-common rpcbind
[sudo] password for usuario:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  libgssglue1 libnfsidmap2 libtirpc1
Paquetes sugeridos:
  open-iscsi watchdog
Se instalarán los siguientes paquetes NUEVOS:
  libgssglue1 libnfsidmap2 libtirpc1 nfs-common nfs-kernel-server rpcbind
0 actualizados, 6 se instalarán, 0 para eliminar y 355 no actualizados.
Necesito descargar 427 kB de archivos.
Se utilizarán 1.900 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

La forma más sencilla de poner en funcionamiento NFS es reiniciando el equipo. Lo podemos hacer de muchas formas pero, ya que estamos usando la terminal, bastará con escribir la siguiente orden:

```
sudo reboot
```

3

Escribimos la orden y pulsamos la tecla **Intro**.



Comprobar la instalación

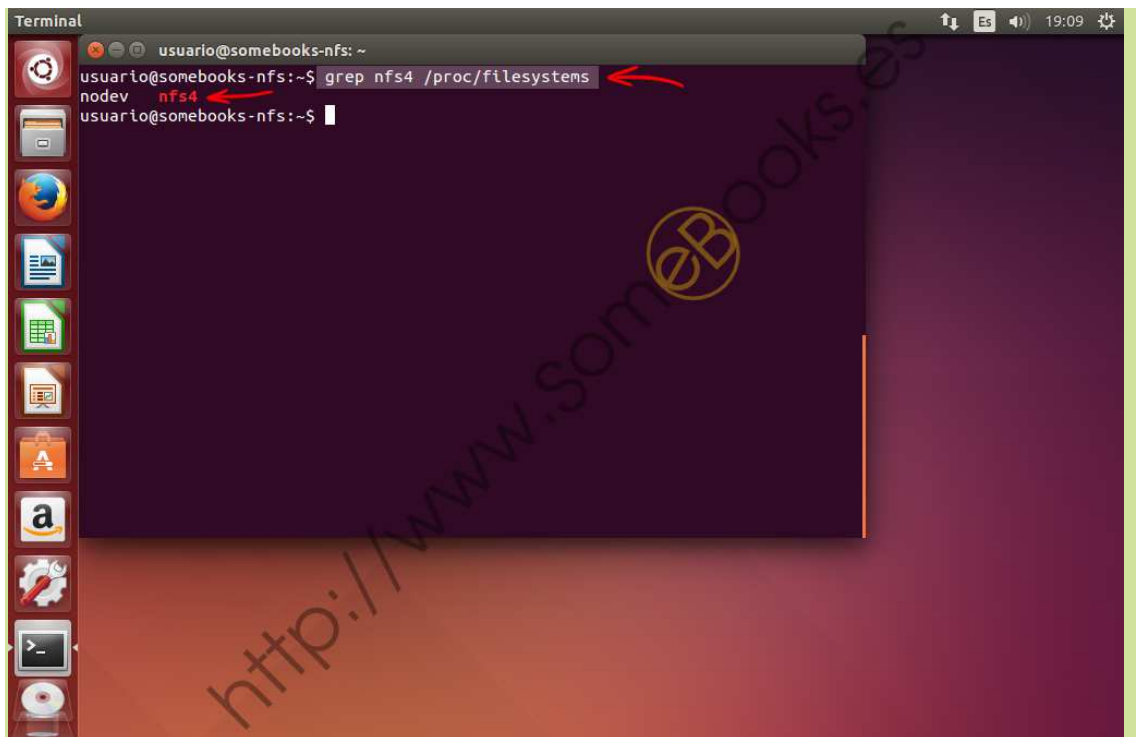
El archivo `/proc/filesystems` contiene los sistemas de archivos soportados por el núcleo de Ubuntu.

Una vez que se haya completado de nuevo el arranque del sistema, podemos comenzar por comprobar si el núcleo de *Ubuntu 14.04 LTS* contiene ahora soporte para *NFS*. Lo lograremos con la siguiente orden:

```
grep nfs4 /proc/filesystems
```

1

En particular, nos hemos asegurado de que disponemos de soporte para la versión 4 (nfs4)



```
Terminal
usuario@somebooks-nfs: ~
usuario@somebooks-nfs:~$ grep nfs4 /proc/filesystems
nodev nfs4
usuario@somebooks-nfs:~$
```


3.3. Instalar NFS en un cliente con Ubuntu 14.04 LTS

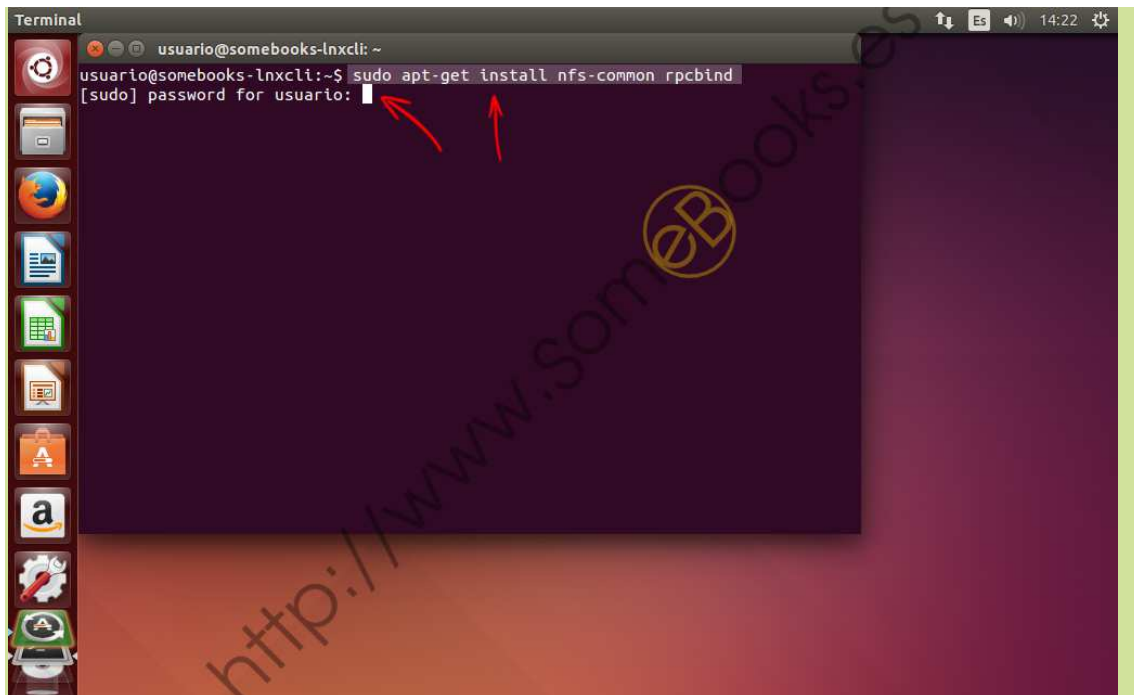
Si la instalación en el servidor parece fácil, en los clientes no lo es menos. En el caso particular de un cliente con *Ubuntu 14.04 LTS*, sólo tendremos que instalar los paquetes **nfs-common** y **rpcbind**.

Para lograrlo, ejecutamos el siguiente comando, también desde la terminal:

```
sudo apt-get install nfs-common rpcbind
```

1

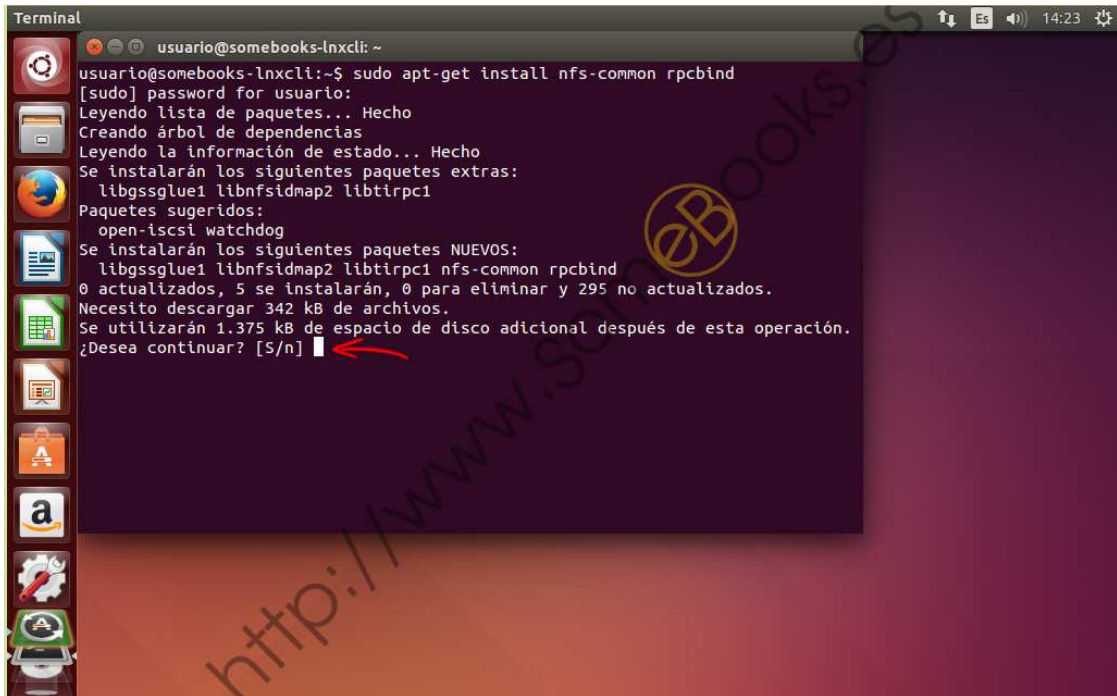
De nuevo, para instalar, deberemos escribir la contraseña de administración.



Como antes, *apt-get* nos informa de las dependencias que habrá que añadir para que la instalación sea correcta y qué espacio de almacenamiento necesitará en el disco.

2

Para continuar, deberemos pulsar la tecla **s** y, a continuación, la tecla **Intro**.

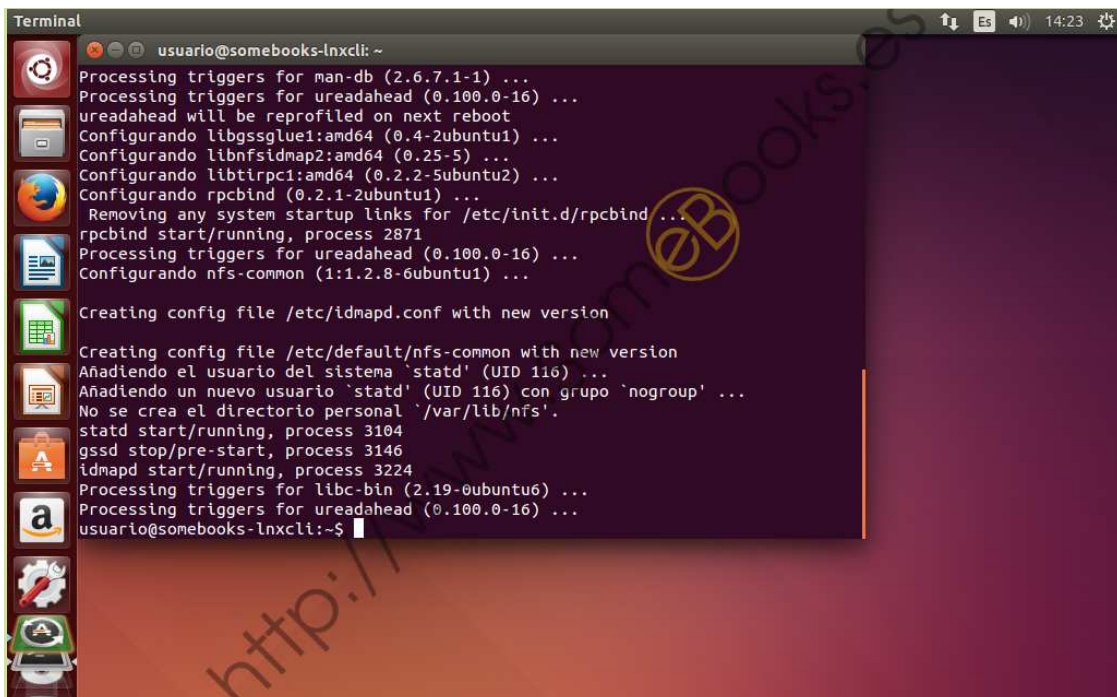


```
Terminal
usuario@somebooks-lnxcli: ~
usuario@somebooks-lnxcli:~$ sudo apt-get install nfs-common rpcbind
[sudo] password for usuario:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  libgssglue1 libnfsidmap2 libtirpc1
Paquetes sugeridos:
  open-iscsi watchdog
Se instalarán los siguientes paquetes NUEVOS:
  libgssglue1 libnfsidmap2 libtirpc1 nfs-common rpcbind
0 actualizados, 5 se instalarán, 0 para eliminar y 295 no actualizados.
Necesito descargar 342 kB de archivos.
Se utilizarán 1.375 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Después, *apt-get* se limita a informarnos del avance de la instalación y, finalmente, nos muestra el prompt del sistema.

3

De esta forma, comprobamos que la instalación ha sido satisfactoria.



```
Terminal
usuario@somebooks-lnxcli: ~
Processing triggers for man-db (2.6.7.1-1) ...
Processing triggers for ureadahead (0.100.0-16) ...
ureadahead will be reprofiled on next reboot
Configurando libgssglue1:amd64 (0.4-2ubuntu1) ...
Configurando libnfsidmap2:amd64 (0.25-5) ...
Configurando libtirpc1:amd64 (0.2.2-5ubuntu2) ...
Configurando rpcbind (0.2.1-2ubuntu1) ...
Removing any system startup links for /etc/init.d/rpcbind ...
rpcbind start/running, process 2871
Processing triggers for ureadahead (0.100.0-16) ...
Configurando nfs-common (1:1.2.8-6ubuntu1) ...
Creating config file /etc/idmapd.conf with new version
Creating config file /etc/default/nfs-common with new version
Añadiendo el usuario del sistema 'statd' (UID 116) ...
Añadiendo un nuevo usuario 'statd' (UID 116) con grupo 'nogroup' ...
No se crea el directorio personal '/var/lib/nfs'.
statd start/running, process 3104
gssd stop/pre-start, process 3146
idmapd start/running, process 3224
Processing triggers for libc-bin (2.19-0ubuntu6) ...
Processing triggers for ureadahead (0.100.0-16) ...
usuario@somebooks-lnxcli:~$
```

3.4. Configurar el servidor NFS

Vamos a suponer que necesitamos compartir con los clientes dos carpetas del servidor: `/home` y `/compartido`. Los pasos que deberemos seguir son estos:

- Crear la carpeta o las carpetas que aún no existan.
- Exportar el contenido de las carpetas
- Reiniciar el servicio NFS
- Crear un archivo en una de las carpetas compartidas para comprobar que es accesible desde los clientes

Crear las carpetas a compartir

La creación de una carpeta para compartir es tan sencilla como utilizar la interfaz gráfica o el comando `mkdir` en la consola (aquí nos inclinaremos por la segunda opción).

Además, como no queremos que los usuarios experimenten problemas para acceder a los datos, también cambiaremos el nombre del usuario y grupo propietarios de la carpeta, para que no sean propiedad de nadie, y los permisos de acceso, para que todos los usuarios dispongan de todos los permisos sobre ella.

Los comandos a ejecutar serían estos:

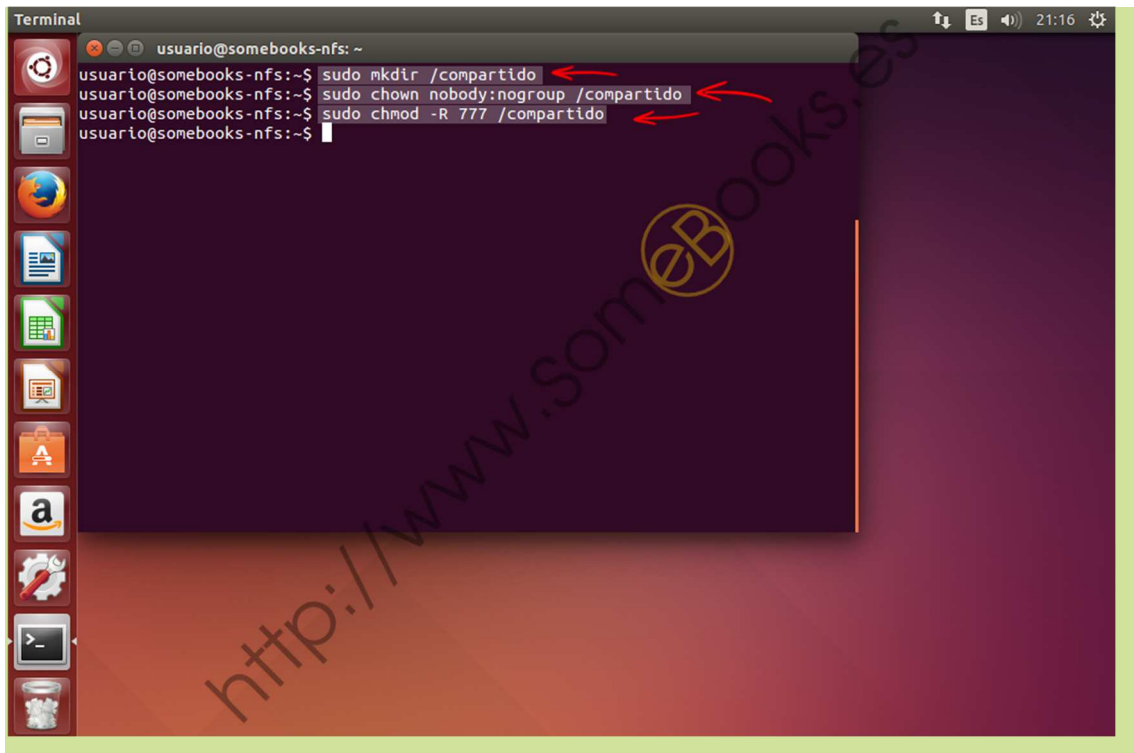
```
sudo mkdir /compartido

sudo chown nobody:nogroup /compartido

sudo chmod -R 777 /compartido
```

1

■ Creamos la carpeta y cambiamos sus permisos



Esto en cuanto a la carpeta `/compartido`. Por otra parte, a la carpeta `/home` sólo accederemos como *superusuario* y a cada una de sus subcarpetas con el usuario correspondiente. Por este motivo, no cambiaremos su propiedad.

Más adelante hablaremos de la cuestión de los permisos utilizando *NFS*. De momento, nos interesa que todo sea lo más sencillo posible.

Exportar el contenido de las carpetas

Después de esto, debemos editar el archivo `/etc/exports`. Este es el archivo donde se indican a *NFS* las carpetas que vamos a compartir (exportar, en la terminología *NFS*). Cada carpeta exportada debe estar en una línea diferente de este archivo, aunque una línea muy larga puede continuarse en la línea siguiente poniendo al final una barra invertida `"\`.

Las líneas tienen el siguiente formato:

```
ruta cliente_1(opciones) cliente_2(opciones) ...
```

Observa que cada cliente tiene dos partes:

No hay espacios en la definición de un cliente, sólo entre un cliente y otro, y entre el primero y la ruta de la carpeta.

- La primera identifica al ordenador cliente (en el ejemplo lo hemos representado como cliente_1, cliente_2, ...). Podemos usar cualquiera de las siguientes opciones:
 - Una dirección IP o un nombre DNS.
 - Caracteres comodín para representar todo el nombre del cliente o una parte. Podemos utilizar los comodines '?' (para representar un carácter cualquiera y '*' (para representar cualquier conjunto de caracteres. No se recomienda utilizarlos con direcciones IP.
 - Intervalos de direcciones IP. Por ejemplo, 192.168.1.0/30, que permitirá acceder a las primeras 30 direcciones a partir de 192.168.1.0
 - netgroup. Cuando dispongamos de un servidor NIS en la red, podremos agrupar los usuarios en grupos . En este caso, podremos utilizar el grupo con el formato @nombre_de_grupo.
- La segunda será una lista de opciones para compartir. Entre las opciones que podemos utilizar, se encuentran las siguientes:
 - *ro*(read-only): La carpeta compartida será de sólo lectura. Es la opción predeterminada.
 - *rw* (read-write): El usuario podrá realizar cambios en el contenido de la carpeta compartida.
 - *wdelay*: El servidor NFS no escribe en el disco si espera otra solicitud de forma inminente. Así se reducen los accesos a disco y mejora el rendimiento. Es la opción predeterminada, pero sólo funciona cuando usamos la opción *sync*.
 - *no_wdelay*: Deshabilita la característica anterior.
 - *root_squash*: Evita que los usuarios con privilegios administrativos los mantengan, sobre la carpeta compartida, cuando se conectan remotamente. En su lugar, se les trata como a un usuario remoto más. Es la opción predeterminada.

- *no_root_squash*: Deshabilita la característica anterior.
- *sync*: Evita responder peticiones antes de escribir los cambios pendientes en disco. Es la opción predeterminada.
- *async*: Deshabilita la característica anterior. Mejora el rendimiento a cambio de que exista el riesgo de corrupción en los archivos o, incluso, en todo el sistema de archivos, si se produjese una interrupción del fluido eléctrico o un bloqueo del sistema.
- *subtree_check*: Cuando el directorio compartido es un subdirectorío de un sistema de archivos mayor, NFS comprueba los directorios por encima de éste para verificar sus permisos y características. Es la opción predeterminada.
- *no_subtree_check*: Deshabilita la característica anterior, lo que hace que el envío de la lista de archivos sea más rápido, pero puede reducir la seguridad.

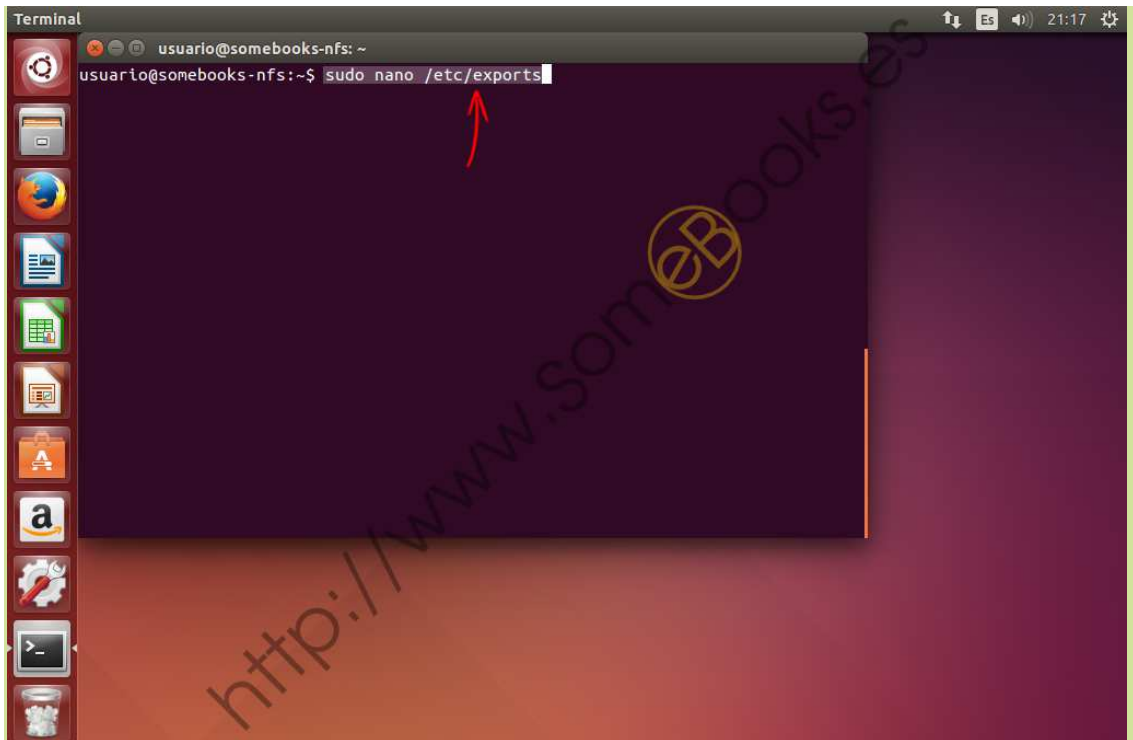
En nuestro caso, incluiremos estas dos nuevas líneas en el archivo `/etc/exports`:

```
/home *(rw, sync, no_root_squash, no_subtree_check)
/compartido *(rw, sync, no_subtree_check)
```

Y para lograrlo, recurriremos, como es habitual, al editor de textos nano:

1

Abrimos el editor con privilegios de administración.

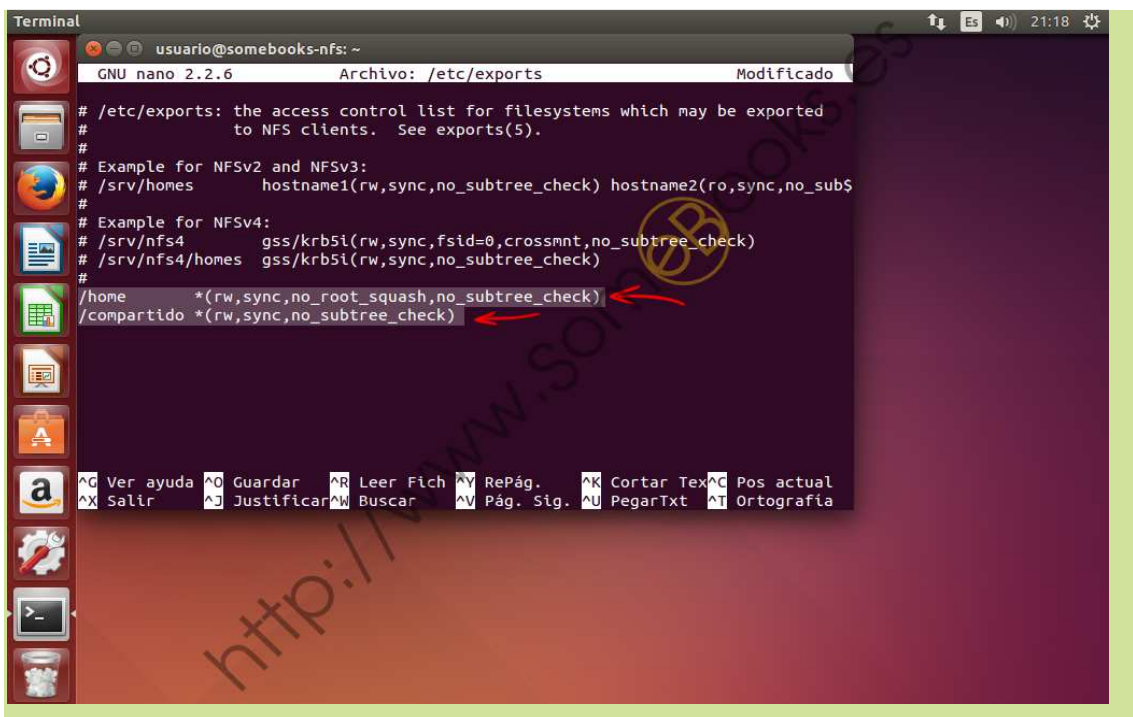


A terminal window titled 'Terminal' with the prompt 'usuario@somebooks-nfs: ~'. The command 'sudo nano /etc/exports' is entered at the prompt. A red arrow points to the command. The terminal background is dark purple with a large, faint watermark 'http://www.somebooks.es' and a circular logo.

Una vez abierto el archivo, incluimos las líneas que indicamos más arriba:

2

Para salir, guardando los cambios, usamos la combinación de teclas **Ctrl + X**.



A terminal window titled 'Terminal' with the prompt 'usuario@somebooks-nfs: ~'. The GNU nano 2.2.6 editor is open, showing the file '/etc/exports'. The editor interface includes a top bar with 'GNU nano 2.2.6', 'Archivo: /etc/exports', and 'Modificado'. The file content is displayed, showing comments and NFS export examples. The last two lines are highlighted with a red background: '/home *(rw,sync,no_root_squash,no_subtree_check)' and '/compartido *(rw,sync,no_subtree_check)'. A red arrow points to the end of the second line. The bottom of the screen shows a status bar with various keyboard shortcuts like '^G Ver ayuda', '^O Guardar', etc. The terminal background is dark purple with a large, faint watermark 'http://www.somebooks.es' and a circular logo.

Iniciar/reiniciar el servicio NFS

Siempre que hagamos cambios en el archivo `/etc/exports`, necesitaremos reiniciar el servicio para que éstos sean efectivos. Lo conseguiremos escribiendo la siguiente orden en la ventana de terminal:

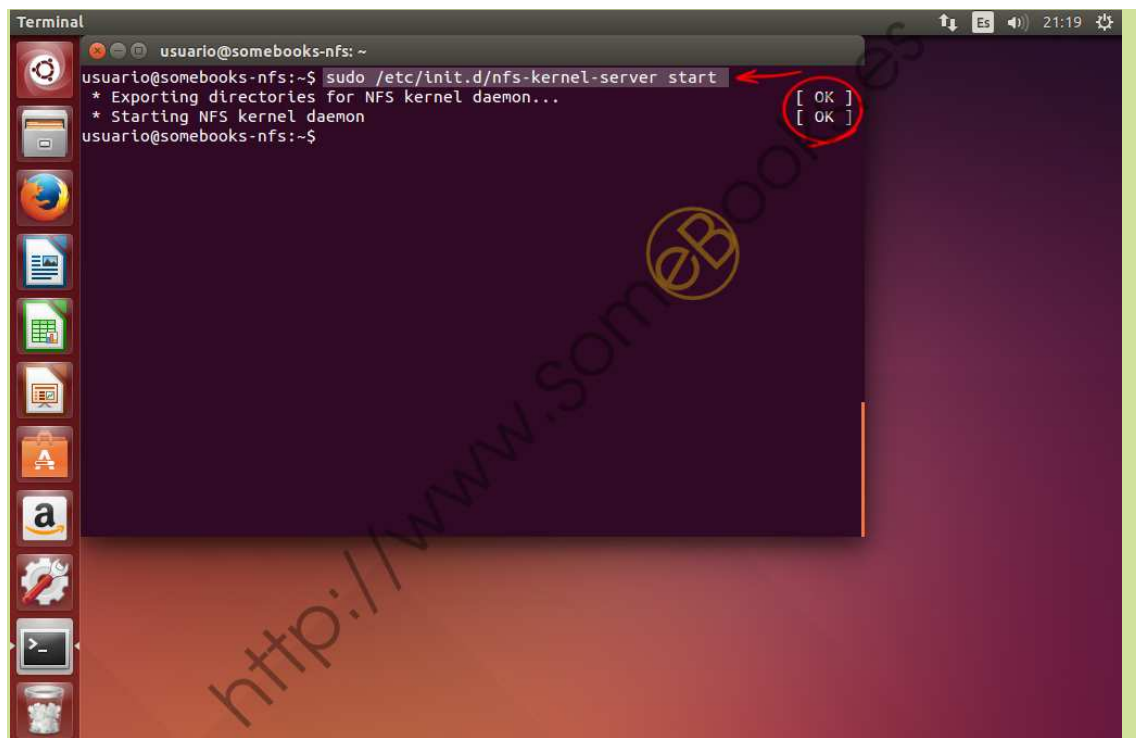
```
sudo /etc/init.d/nfs-kernel-server restart
```

En nuestro caso, como es la primera vez, en lugar de reiniciarlo, sencillamente lo iniciaremos. Es decir, la orden que escribiremos en la ventana de terminal es la siguiente:

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

1

Servicio iniciado.



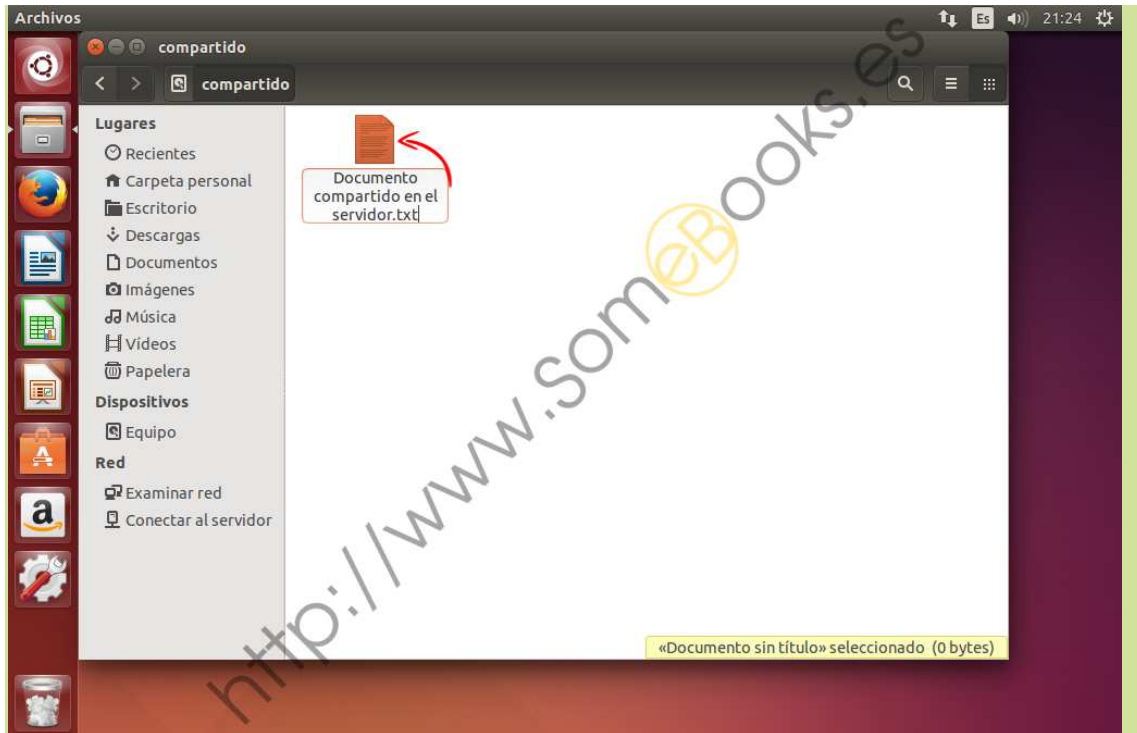
Crear un archivo en una carpeta compartida

Después de configurar el servidor, tendremos que hacer lo propio con los clientes. Sin embargo, para que podamos comprobar que todo es correcto después de completar esa la operación, sería conveniente disponer de algún contenido en una de las carpetas compartidas. Así comprobaremos que el contenido está accesible.

Por lo tanto, nos dirigimos a `/compartido` y creamos un archivo cualquiera.

1

En este caso, nos hemos decantado por el oportuno nombre de `Documento compartido en el servidor.txt`



3.5. Acceder a la carpeta compartida con NFS desde un cliente con Ubuntu 14.04

En principio, las operaciones realizadas en el punto anterior han conseguido ofrecer dos carpetas distintas del servidor para que puedan ser accedidas desde cualquiera de los equipos cliente de la red. Sin embargo, ahora tendremos que configurar los clientes para que ese acceso sea efectivo.

Comenzaremos por un cliente que incorpora el sistema operativo *Ubuntu 14.04*, en el que tendremos que realizar las siguientes operaciones para completar su configuración y asegurarnos de que todo funciona del modo adecuado:

- Crear el punto de montaje, en la estructura de directorios local, donde se montarán las carpetas compartidas.
- Realizar el montaje y comprobarlo.
- Crear archivos en las carpetas compartidas
- Conseguir que las carpetas compartidas se monten automáticamente al arrancar el cliente.

Una vez que tenemos claras las acciones que llevaremos a cabo, es hora de comenzar:

Crear el punto de montaje para las carpetas compartidas

Lo primero que tendremos que decidir es en qué lugar del árbol de directorios del equipo *cliente* se montarán las carpetas compartidas por el *servidor* (recuerda que en *GNU/Linux* sólo existe un árbol de directorios donde se van montando los diferentes volúmenes que necesitamos)

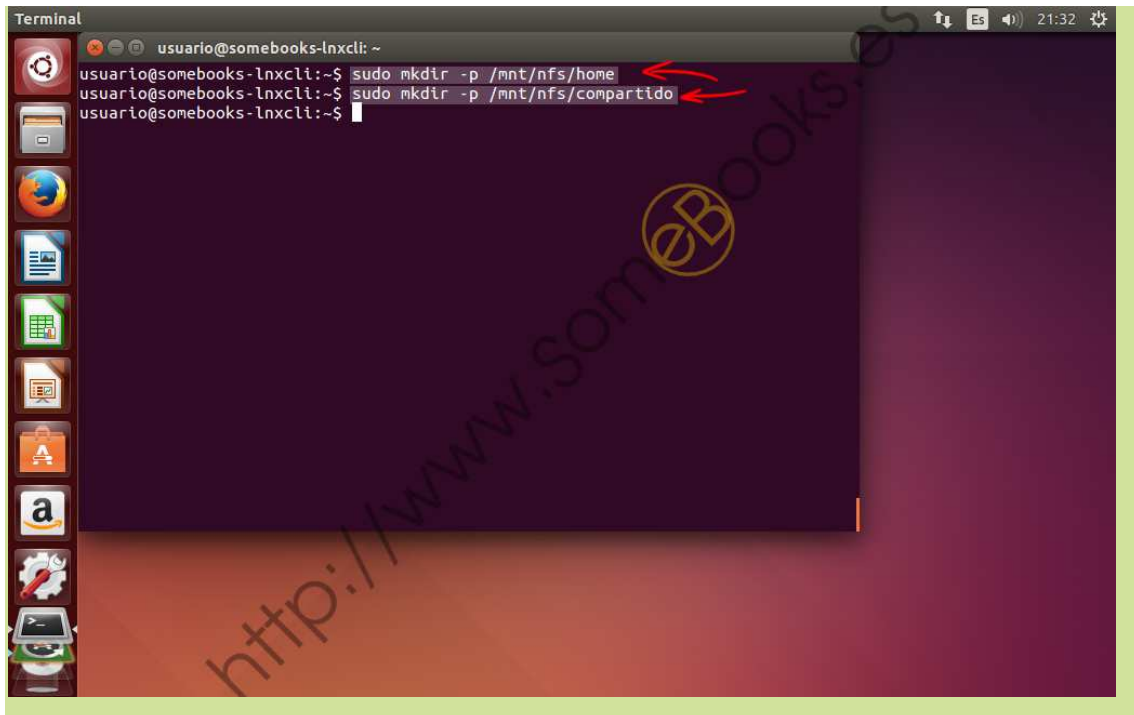
Para este ejemplo, crearemos una subcarpeta, dentro de `/mnt` llamada `nfs`. En su interior, reproduciremos la ruta original de las carpetas compartidas:

```
sudo mkdir -p /mnt/nfs/home
```

```
sudo mkdir -p /mnt/nfs/compartido
```

1

Creación de los puntos de montaje

A terminal window titled 'Terminal' showing a user at 'usuario@somebooks-lnxcli: ~'. The user enters two commands: 'sudo mkdir -p /mnt/nfs/home' and 'sudo mkdir -p /mnt/nfs/compartido'. Red arrows point to the 'home' and 'compartido' parts of the paths. The terminal has a dark background and a sidebar with application icons on the left. A watermark 'http://www.somebooks.es' is visible across the terminal area.

```
usuario@somebooks-lnxcli: ~  
usuario@somebooks-lnxcli:~$ sudo mkdir -p /mnt/nfs/home  
usuario@somebooks-lnxcli:~$ sudo mkdir -p /mnt/nfs/compartido  
usuario@somebooks-lnxcli:~$
```

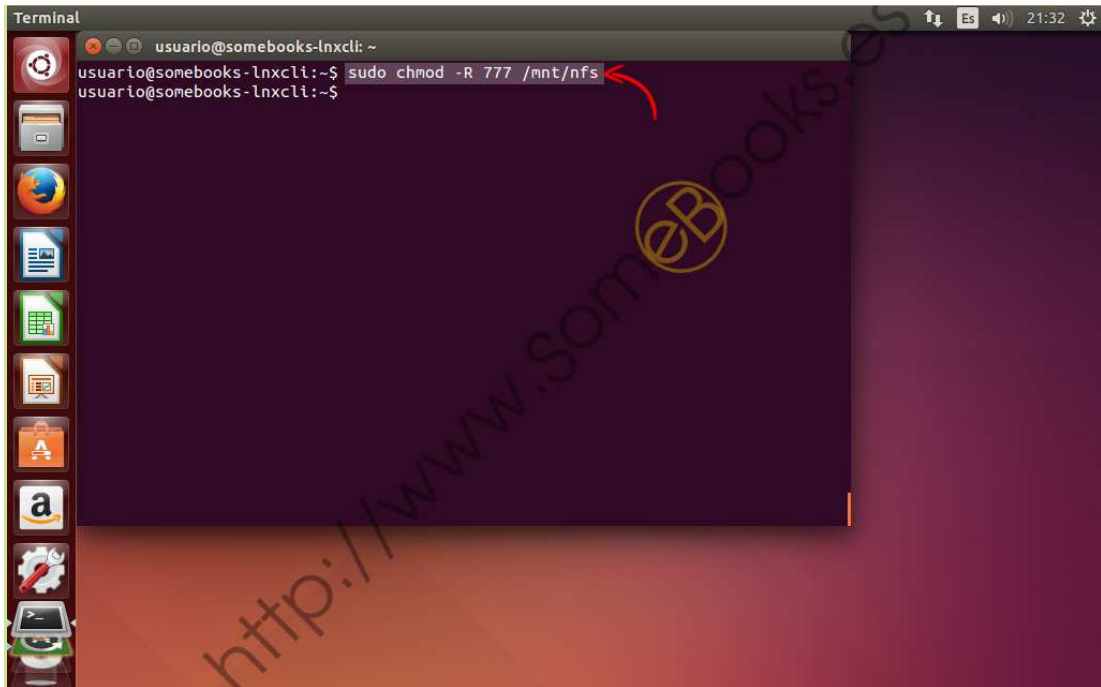
Quizás lo único destacable en la sintaxis anterior sea el uso del argumento **-p** (también podemos escribir **-parents**). Su cometido es doble: por un lado evitar que se produzca un error si alguna de las carpetas ya existiese (aunque este no es el caso); por el otro, crea automáticamente la parte de la estructura del árbol que sea necesaria. Es decir, en la primera orden, se crea la carpeta **home**, pero también la carpeta **nfs** (que aún no existía). En la segunda, sólo se crea la carpeta **compartido** (en realidad, en este segundo caso podríamos haber evitado el argumento).

Una última precaución que deberemos tener en cuenta es que, aunque hayamos dado permisos de escritura sobre las carpetas compartidas en la configuración *NFS* del servidor, no podremos escribir en ellas si no disponemos de permisos sobre los puntos de montaje donde se van a montar dichas carpetas en los clientes. Por lo tanto, antes de continuar, cambiaremos también sus permisos:

```
sudo chmod -R 777 /mnt/nfs
```

2

Asignamos permisos a la carpeta **nfs** y a todo su contenido.



Realizar el montaje de las carpetas compartidas y comprobarlo

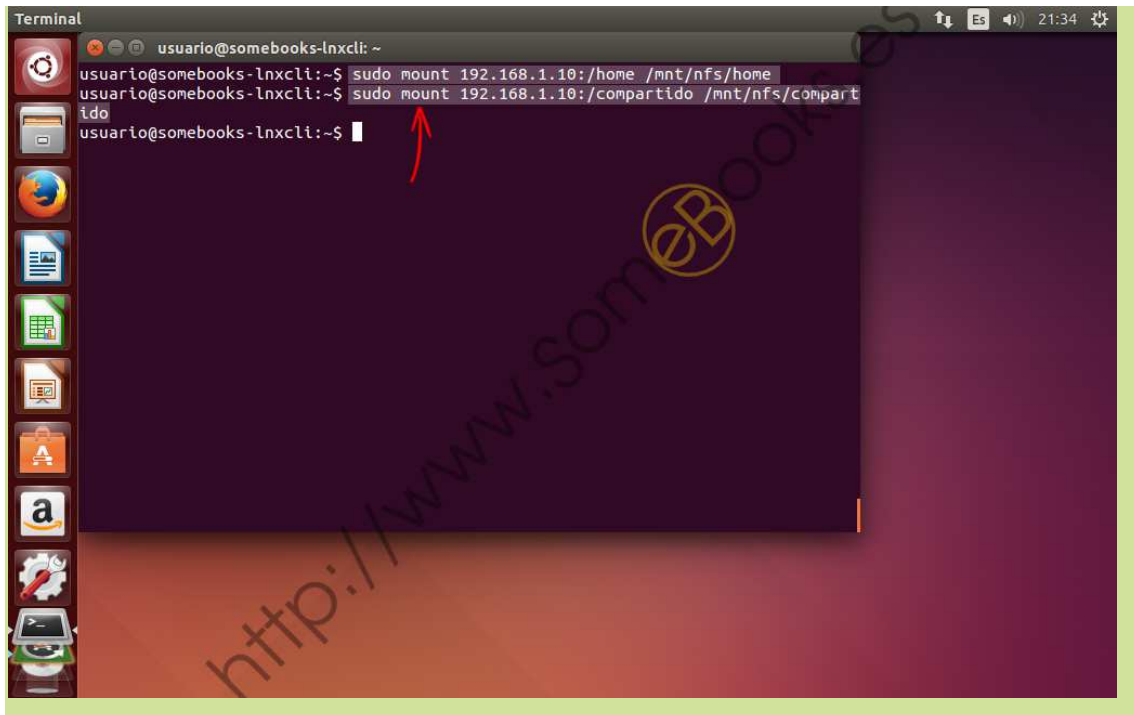
La siguiente tarea que debemos completar, será montar las carpetas compartidas por el servidor en el punto montaje que hemos creado en el apartado anterior. Para ello, como es natural, recurrimos al comando **mount**:

```
sudo mount 192.168.1.10:/home /mnt/nfs/home
```

```
sudo mount 192.168.1.10:/compartido /mnt/nfs/compartido
```

1

La ausencia de errores indica que el montaje se realiza con éxito.

A terminal window titled 'Terminal' with the prompt 'usuario@somebooks-lnxcli: ~'. It shows two successful 'mount' commands. The first command is 'sudo mount 192.168.1.10:/home /mnt/nfs/home' and the second is 'sudo mount 192.168.1.10:/compartido /mnt/nfs/compartido'. A red arrow points to the second command. The terminal has a dark purple background and a sidebar with application icons on the left. A watermark 'http://www.somebooks.es' is visible across the terminal area.

```
usuario@somebooks-lnxcli: ~
usuario@somebooks-lnxcli:~$ sudo mount 192.168.1.10:/home /mnt/nfs/home
usuario@somebooks-lnxcli:~$ sudo mount 192.168.1.10:/compartido /mnt/nfs/compartido
usuario@somebooks-lnxcli:~$
```

Para comprobar que todo ha ido bien, utilizaremos dos comandos: **df** y **mount** (aunque con cualquiera de ellos sería suficiente).

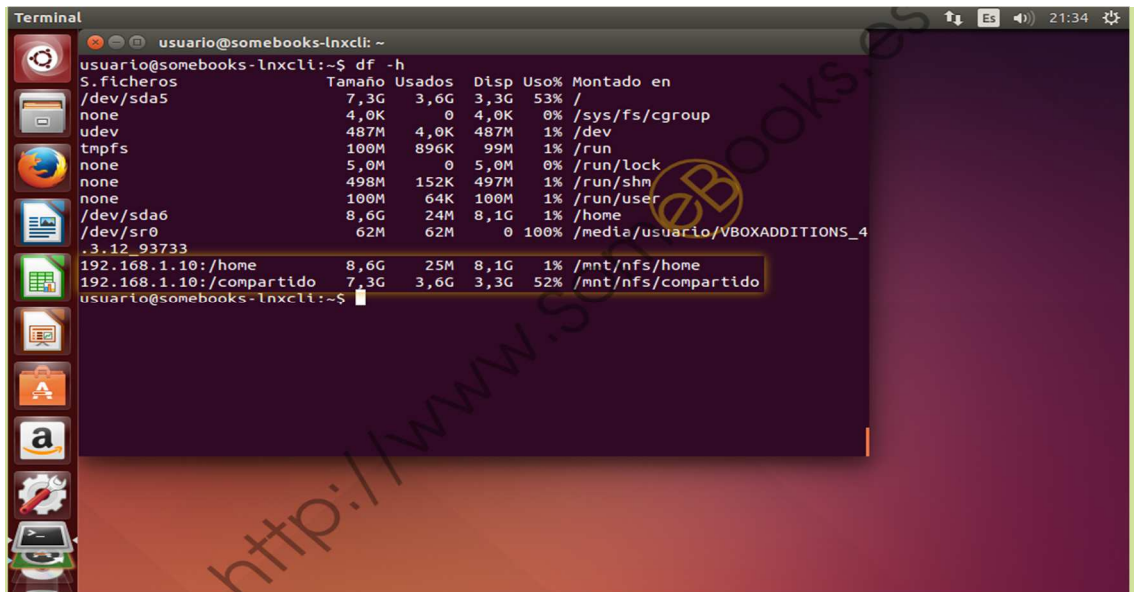
El comando **df** nos ofrece información sobre el espacio en disco utilizado y el que tenemos disponible en los sistemas de archivos que tengamos montados en estos momentos (de todos, salvo que indiquemos lo contrario en sus parámetros). Así pues, escribiremos lo siguiente en la terminal:

```
df -h
```

La opción **-h** hace que la salida sea más fácilmente comprensible para las personas (**-h** ó **-human-readable**).

2

Al final de la salida del comando, podemos comprobar que aparecen las carpetas que acabamos de montar.



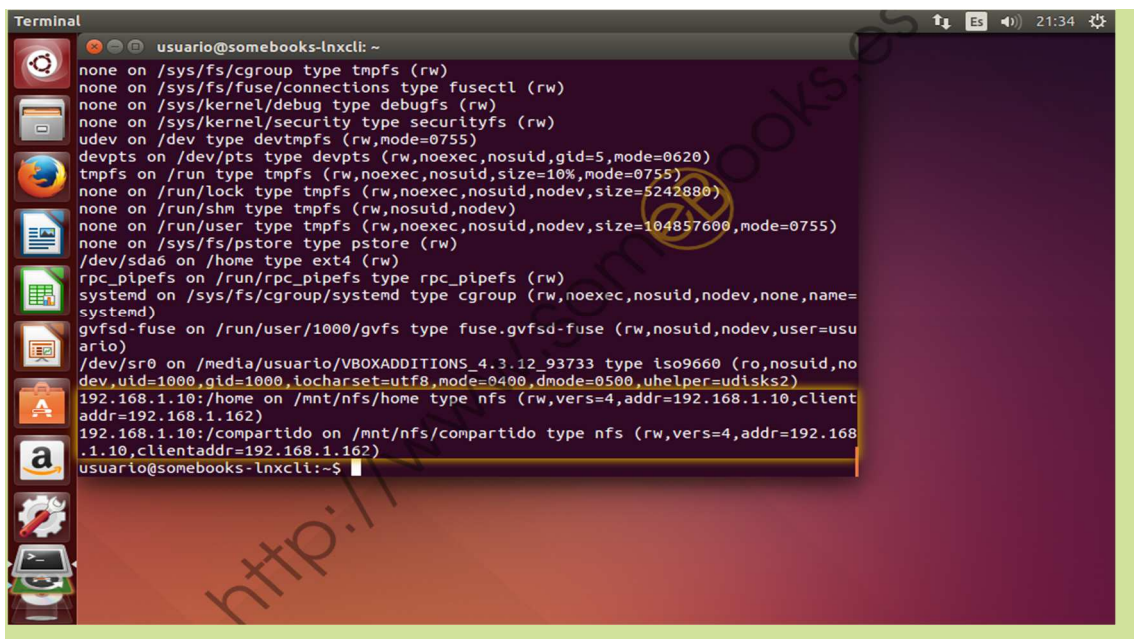
```
usuario@somebooks-lnxcli: ~  
usuario@somebooks-lnxcli:~$ df -h  
S.ficheros      Tamaño Usados  Disp Uso% Montado en  
/dev/sda5       7,3G  3,6G   3,3G  53% /  
none            4,0K    0   4,0K   0% /sys/fs/cgroup  
udev           487M   4,0K  487M   1% /dev  
tmpfs           100M  896K   99M   1% /run  
none            5,0M    0   5,0M   0% /run/lock  
none           498M  152K  497M   1% /run/shm  
none           100M   64K  100M   1% /run/user  
/dev/sda6       8,6G   24M   8,1G   1% /home  
/dev/sr0        62M   62M    0  100% /media/usuario/VBOXADDITIONS_4.3.12_93733  
192.168.1.10:/home      8,6G   25M   8,1G   1% /mnt/nfs/home  
192.168.1.10:/compartido 7,3G   3,6G   3,3G  52% /mnt/nfs/compartido  
usuario@somebooks-lnxcli:~$
```

Como hemos dicho antes, también podemos utilizar el comando `mount`, sin argumentos, para mostrar la lista de los volúmenes montados en este momento (en realidad, lo que muestra es el contenido del archivo `/etc/mtab`).

`mount`

3

También aquí, las dos últimas entradas representan las carpetas compartidas.



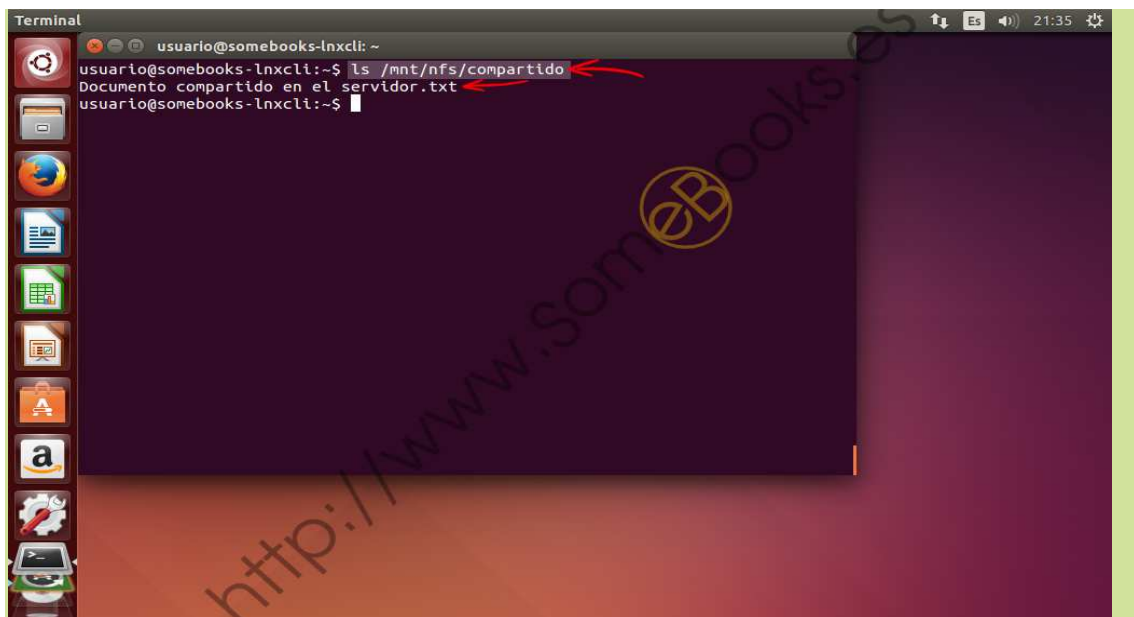
```
usuario@somebooks-lnxcli: ~  
usuario@somebooks-lnxcli:~$ mount  
none on /sys/fs/cgroup type tmpfs (rw)  
none on /sys/fs/fuse/connections type fusectl (rw)  
none on /sys/kernel/debug type debugfs (rw)  
none on /sys/kernel/security type securityfs (rw)  
udev on /dev type devtmpfs (rw,mode=0755)  
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=0620)  
tmpfs on /run type tmpfs (rw,noexec,nosuid,size=10%,mode=0755)  
none on /run/lock type tmpfs (rw,noexec,nosuid,nodev,size=5242880)  
none on /run/shm type tmpfs (rw,nosuid,nodev)  
none on /run/user type tmpfs (rw,noexec,nosuid,nodev,size=104857600,mode=0755)  
none on /sys/fs/pstore type pstore (rw)  
/dev/sda6 on /home type ext4 (rw)  
rpc_pipefs on /run/rpc_pipefs type rpc_pipefs (rw)  
systemd on /sys/fs/cgroup/systemd type cgroup (rw,noexec,nosuid,nodev,name=systemd)  
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,user=usuario)  
/dev/sr0 on /media/usuario/VBOXADDITIONS_4.3.12_93733 type iso9660 (ro,nosuid,nodev,uid=1000,gid=1000,iocharset=utf8,mode=0400,dmode=0500,uhelper=udisks2)  
192.168.1.10:/home on /mnt/nfs/home type nfs (rw,vers=4,addr=192.168.1.10,clientaddr=192.168.1.162)  
192.168.1.10:/compartido on /mnt/nfs/compartido type nfs (rw,vers=4,addr=192.168.1.10,clientaddr=192.168.1.162)  
usuario@somebooks-lnxcli:~$
```

La ventaja de este último método es que disponemos de información complementaria, como los permisos o las direcciones IP tanto del cliente como del servidor).

Una última comprobación sería obtener el contenido de las carpetas compartidas:

4

El contenido de la carpeta `/compartido`.

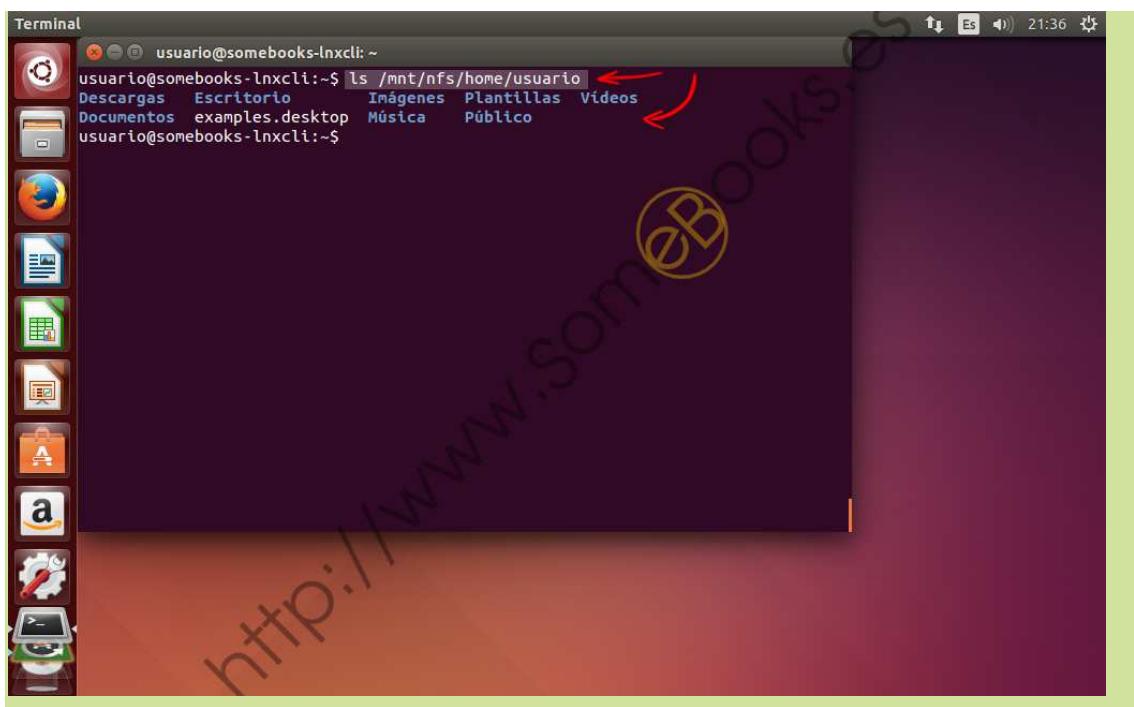


```
Terminal
usuario@somebooks-lnxcli: ~
usuario@somebooks-lnxcli:~$ ls /mnt/nfs/compartido
Documento compartido en el servidor.txt
usuario@somebooks-lnxcli:~$
```

A terminal window titled 'Terminal' with the prompt 'usuario@somebooks-lnxcli: ~'. The user enters the command 'ls /mnt/nfs/compartido'. The output is 'Documento compartido en el servidor.txt'. Red arrows point to the command and the output. A watermark 'http://www.somebooks.es' is visible across the terminal window.

5

El contenido de la carpeta de uno de los usuarios contenidos en `/home`.



```
Terminal
usuario@somebooks-lnxcli: ~
usuario@somebooks-lnxcli:~$ ls /mnt/nfs/home/usuario
Descargas  Escritorio  Imágenes  Plantillas  Videos
Documentos examples.desktop  Música    Público
usuario@somebooks-lnxcli:~$
```

A terminal window titled 'Terminal' with the prompt 'usuario@somebooks-lnxcli: ~'. The user enters the command 'ls /mnt/nfs/home/usuario'. The output lists several folders: 'Descargas', 'Escritorio', 'Imágenes', 'Plantillas', 'Videos', 'Documentos', 'examples.desktop', 'Música', and 'Público'. Red arrows point to the command and the output. A watermark 'http://www.somebooks.es' is visible across the terminal window.

Crear archivos en las carpetas compartidas

Para comprobar que todo funciona correctamente, crearemos un par de archivos vacíos (uno en cada carpeta) y después comprobaremos que se han creado.

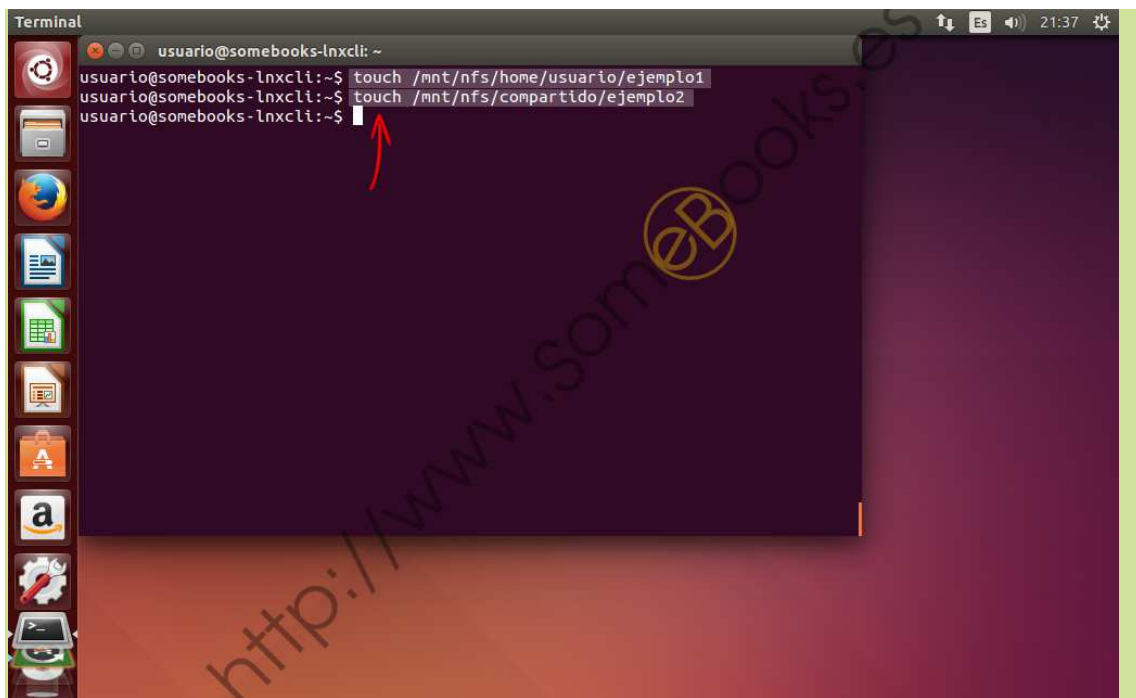
```
touch /mnt/nfs/home/ejemplo1
```

```
touch /mnt/nfs/compartido/ejemplo2
```

Si experimentas algún tipo de error durante la creación, puede estar ocasionado por los permisos de alguna de las carpetas montadas. Si es así, basta con volver a ejecutar la orden `chmod` que explicamos más arriba.

1

Creamos dos archivos de prueba.



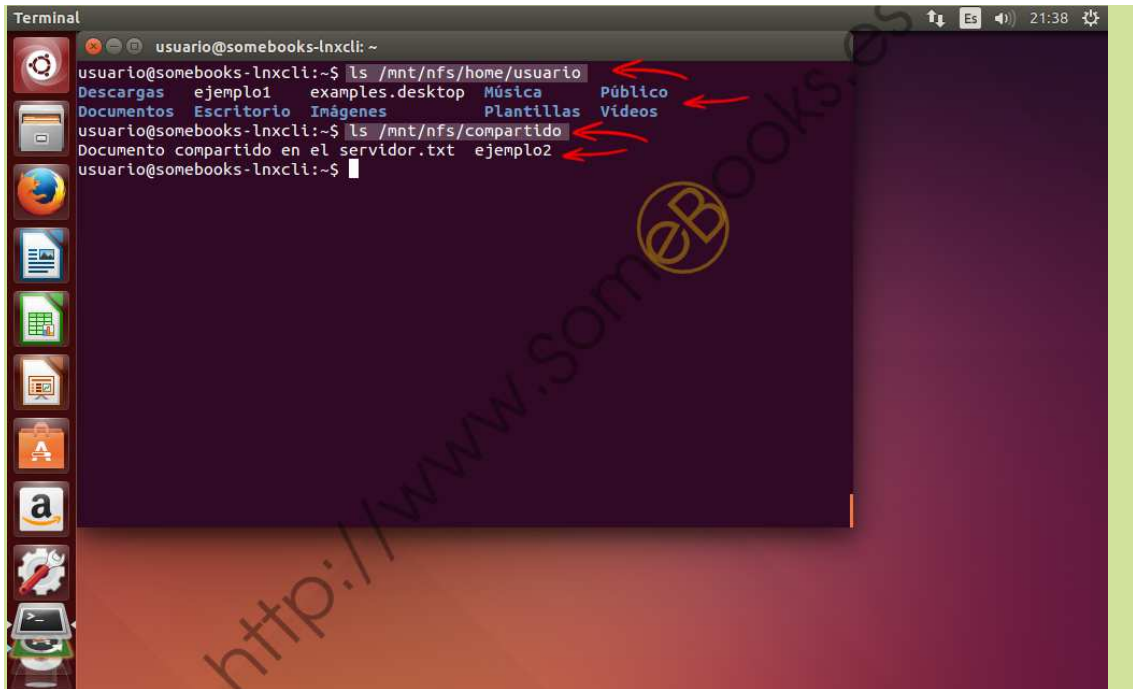
Y para comprobar que la creación se ha realizado con éxito, consultamos el contenido de ambas carpetas:

```
ls /mnt/nfs/home
```

```
ls /mnt/nfs/compartido
```


2

...Aunque, si la orden **touch** no ha devuelto errores, lo más probable es que todo haya ido bien.



Montar automáticamente las carpetas compartidas al iniciar el cliente.

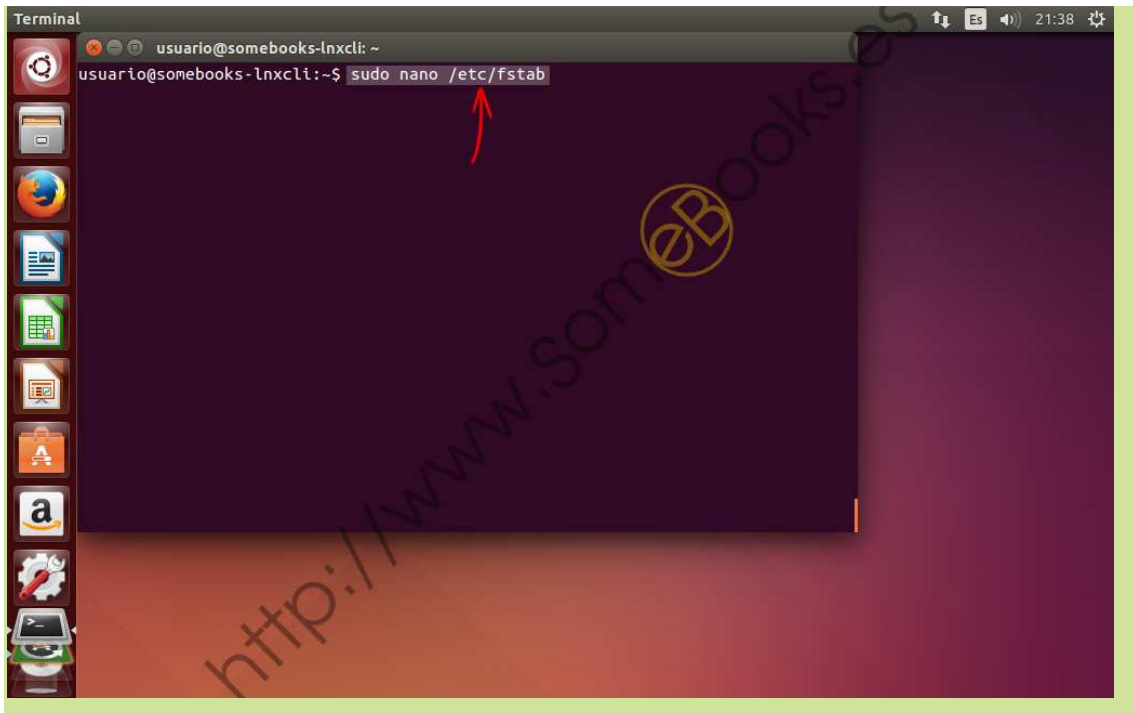
Los sistemas operativos de la familia *Unix*, como es el caso de *GNU/Linux*, disponen de un archivo `/etc/fstab` (file systems table) donde se guarda la información necesaria sobre los diferentes volúmenes que se montarán durante el arranque del sistema.

Cuando implementamos una estructura *NFS*, como la que estamos viendo aquí, lo más probable es que queramos que los clientes monten las carpetas compartidas durante el arranque del sistema. Como habrás podido deducir, esto se consigue modificando el archivo `/etc/fstab`. Y, como de costumbre, recurrimos al editor *nano* para conseguirlo:

```
sudo nano /etc/fstab
```

1

ejecutamos el editor nano para modificar el contenido de `/etc/fstab`.



Cada línea del archivo representa un volumen diferente y atiende al siguiente formato:

El estudio detallado de las *opciones* de montaje se escapan de los objetivos de este texto. Sin embargo, si estás interesado en profundizar, puedes recurrir, por ejemplo, al manual de *Ubuntu* (<http://manpages.ubuntu.com/manpages/hardy/es/man5/nfs.5.html>).

- *Dispositivo*: Referencia al volumen que vamos a montar. En nuestro caso, para la primera carpeta, su valor será `192.168.1.10:/home`.
- *Punto de montaje*: La carpeta donde se montarán los datos del volumen. Su valor para la primera carpeta será `/mnt/nfs/home`.
- *Sistema de archivos*: Indica el sistema de archivos utilizado en el volumen. En ambos casos, el valor será `nfs`.
- *Opciones*: Indica los parámetros que usará `mount` para montar el dispositivo. Estarán separadas por comas y no incluirán espacios. Para las dos carpetas, las opciones serán `auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo=1800`.

- *Frecuencia de respaldo:* Frecuencia con la que se utiliza la herramienta `dump` para respaldar (copiar) el sistema de archivos. Si su valor es cero, el volumen no se respalda.
- *Orden de revisión:* Orden en el que la herramienta `fsck` revisa el volumen en busca de posibles errores durante el proceso de inicio. Si su valor es cero, el volumen no se revisa.

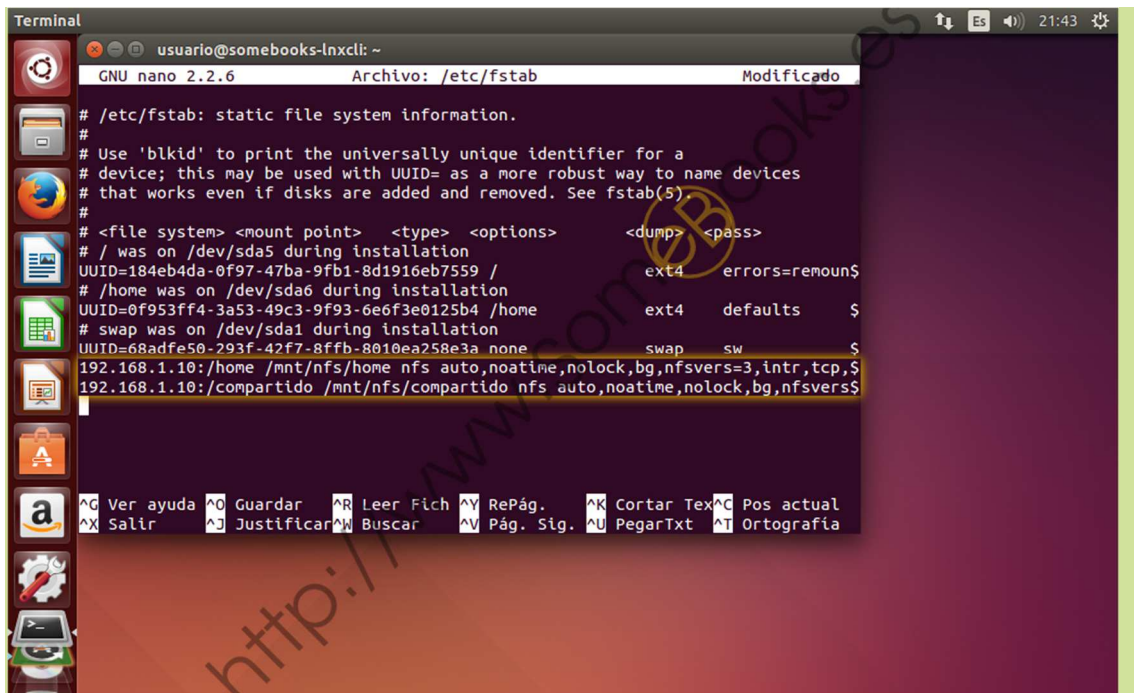
En definitiva, las líneas que debemos incluir son estas:

```
192.168.1.10:/home /mnt/nfs/home nfs
auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo=1800 0 0

192.168.1.10:/var/compartido /mnt/nfs/var/compartido nfs
auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo=1800 0 0
```

2

Aspecto final del archivo `/etc/fstab`.



The screenshot shows a terminal window with the nano text editor open to the file `/etc/fstab`. The editor's title bar indicates 'GNU nano 2.2.6' and 'Archivo: /etc/fstab'. The file content includes comments about using 'blkid' and a table of filesystem entries. The last two lines of the file are highlighted with a yellow box:

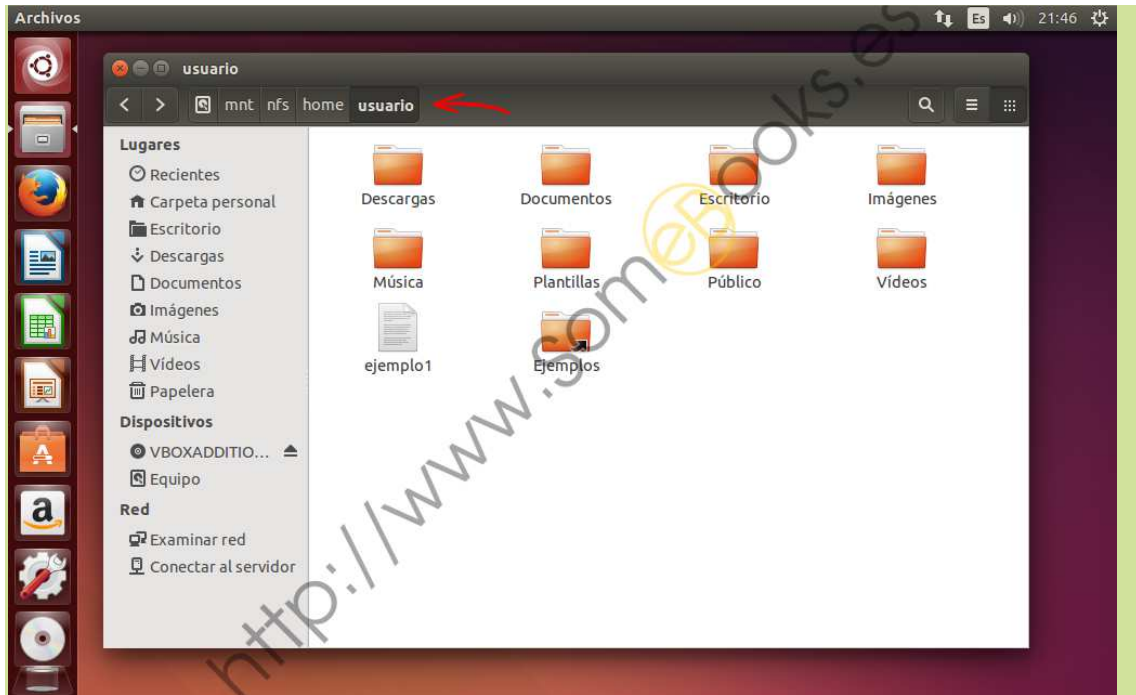
```
192.168.1.10:/home /mnt/nfs/home nfs auto,noatime,nolock,bg,nfsvers=3,intr,tcp,$
192.168.1.10:/compartido /mnt/nfs/compartido nfs auto,noatime,nolock,bg,nfsvers$
```

At the bottom of the terminal, there is a status bar with keyboard shortcuts for various actions like 'Ver ayuda', 'Guardar', 'Leer Fich', etc.

Después de esto, sólo quedará grabar los cambios y reiniciar para comprobar que todo funciona correctamente.

3

Después de reiniciar, comprobamos que seguimos teniendo acceso a los datos compartidos.

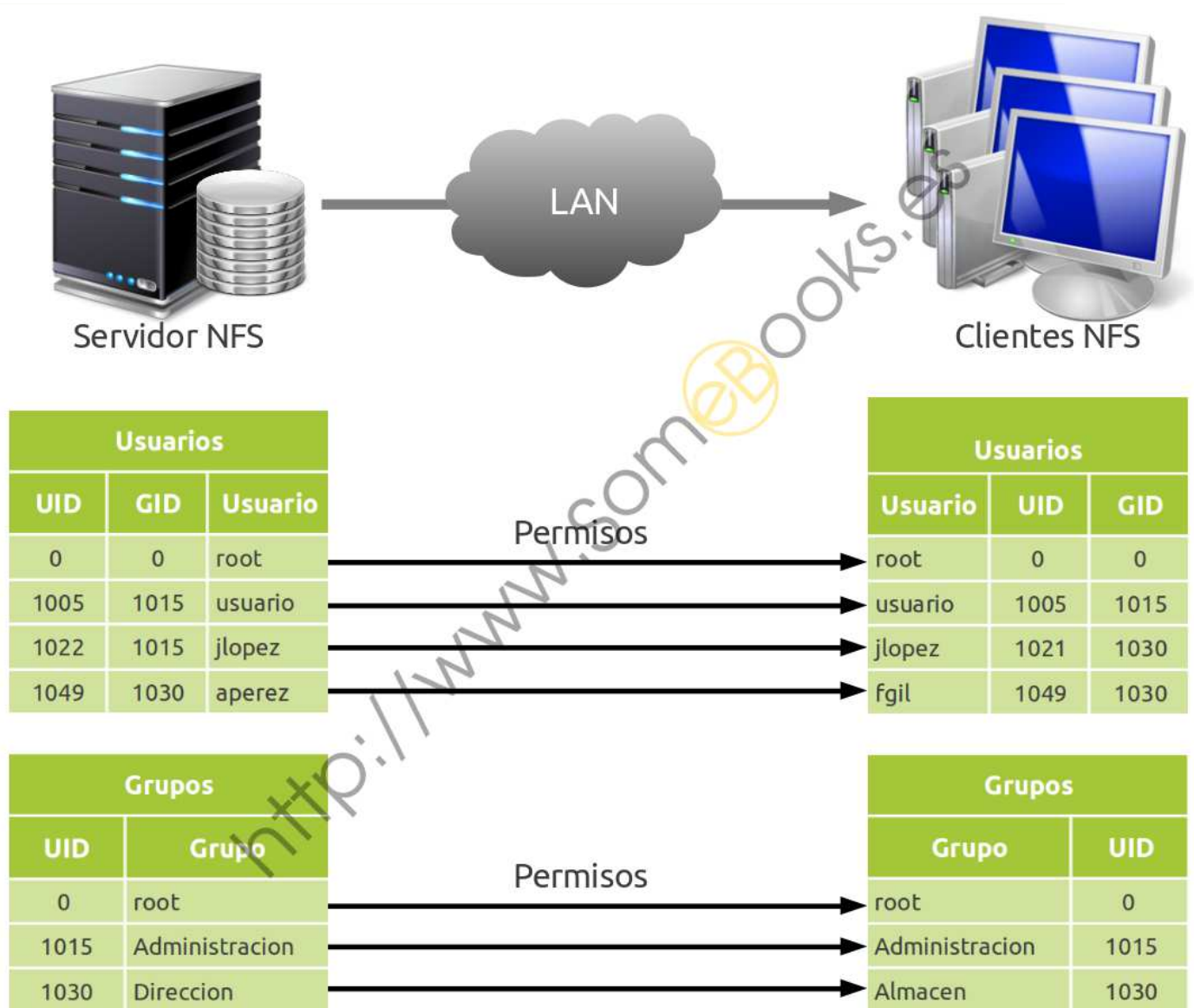


3.6. Problemas con los permisos en NFS

Uno de los problemas que plantea el uso de *NFS* es que no permite validar a los usuarios que tratan de acceder a una carpeta compartida. En realidad, el servidor *NFS* envía al cliente los permisos de cada archivo y subcarpeta que encuentre dentro de la carpeta compartida. Además, también se envía el *UID* del usuario propietario y el *GID* de su grupo principal.

El problema es que, cuando existan usuarios y/o grupos en los *equipos cliente* que tengan asignado el mismo *UID* o dispongan del mismo *GID* para su grupo principal, estos *usuarios locales* asumirán los permisos que tenían los usuarios del equipo servidor sobre el contenido de las carpetas compartidas.

Dicho así, parece un poco lioso, pero vamos a tratar de explicarlo usando el siguiente ejemplo:



Este ejemplo ilustra varias posibles situaciones:

- Los usuarios *root* y *usuario* existen tanto en el servidor como en los clientes y tienen los mismos valores *UID* y *GID*, por lo que podrán utilizar los elementos compartidos sin ningún problema.
- El usuario *jlopez* existe tanto en el servidor como en los clientes, sin embargo, como no coinciden ni sus *UID* ni sus *GID*, el usuario de un *equipo cliente* no podrá usar los datos compartidos que pertenezcan a la misma cuenta del *equipo servidor*. Sin embargo, sí que podrá beneficiarse de los permisos heredados del grupo cuyo *GID* es 1030, aunque se trate de grupos diferentes (en el servidor corresponde con el grupo *Direccion* y en los clientes con el grupo *Almacen*).
- Igual de curioso resulta el caso del usuario *fgil*, que podrá acceder desde un equipo cliente a los datos compartidos que pertenezcan a la cuenta *aperez* del servidor, ya que, aunque el nombre de las cuentas es diferente, sus valores *UID* y *GID* son los mismos.

La opción *root_squash* es el valor predeterminado, por lo que no es necesario incluirlo de forma explícita

Afortunadamente, si estamos usando la opción *root_squash* en las definiciones de las carpetas compartidas que incluimos en el archivo */etc/exports*, el superusuario del equipo cliente no tendrá plenos poderes sobre los archivos compartidos por el servidor. Lógicamente, sí que mantendrá los privilegios sobre sus propios archivos.

Como vemos, el tratamiento que hace *NFS* de las cuentas de usuarios y grupos se aleja bastante de lo deseable. Sin embargo, los problemas que hemos ilustrado más arriba desaparecen cuando combinamos el uso de *NFS* con la autenticación centralizada de *LDAP*, pero esto no lo aprenderemos hasta el próximo capítulo.