# Servidor de archivos: Network File System (NFS)

**Introducción**

NFS es un popular protocolo desarrollado por SUN, utilizado para compartir volúmenes entre maquinas dentro de una red de manera transparente, mas comúnmente utilizado entre sistemas basados sobre UNIX. Es útil y fácil de utilizar, sin embargo no en vano es apodado *cariñosamente* como *"No File Security"*. NFS no utiliza un sistema de contraseñas como el que tiene SAMBA, solo una lista de control de acceso determinada por direcciones IP o nombres. Es por esto que es importante que el administrador de la red local o usuario entienda que un servidor NFS puede ser un verdadero e inmenso agujero de seguridad si este no es configurado apropiadamente e implementado detrás de un corta-fuegos o firewall.

Todas las operaciones sobre ficheros son síncronas. Esto significa que la operación sólo retorna cuando el servidor ha completado todo el trabajo asociado para esa operación. En caso de una solicitud de escritura, el servidor escribirá físicamente los datos en el disco, y si es necesario, actualizará la estructura de directorios, antes de devolver una respuesta al cliente. Esto garantiza la integridad de los ficheros.

Decir que hemos usado la versión 3 del nfs, y que se está también ya la versión 4 que incluye seguridad Kerberos, trabaja con cortafuegos, permite ACLs y utiliza operaciones con descripción del estado.

**Requisitos**

Necesitaremos tener instalados en cada servidor los paquetes de nfs-utils y portmap o rpcbind. En la parte del cliente necesitaremos el paquete nfs-common.

**Opcinones**

Las opciones de exports son las siguientes (fichero configuración /etc/exports):

* (): Son las opciones que trae por defecto.
* ro: El directorio se exportará como solo lectura (opción por defecto).
* rw: El directorio se exportará como lectura/escritura.
* root\_squash: Los accesos desde el cliente con UID=0 (root) se convierten en el servidor en accesos con UID de un usuario anónimo (opción por defecto).
* sync: Modo de sincronización de datos. Se verificarán los datos al copiarlos, algo lento pero más seguro.
* async: Es el segundo modo de sincronización de datos. Es más veloz pero pueden haber pérdidas de datos en caso de caida de la conexión.
* no\_root\_squash: Se permite el acceso desde un UID = 0 sin conversión. Es decir, los accesos de root en el cliente se convierten en accesos de root en el servidor.
* all\_squash: Todos los accesos desde el cliente (con cualquier UID) se transforman en accesos de usuario anónimo.
* anonuid, anongid: Con las opciones root\_squash y all\_squash, el acceso anónimo se efectúa con el UID y GID primario del usuario denominado nobody si éste existe en el servidor (opción por defecto). Si queremos usar otro nombre, anonuid y anongid establecen respectivamente qué uid y gid tendrá la cuenta anónima que el servidor utilizará para acceder contenido del directorio.
* noaccess: Niega el acceso al directorio especificado, útil para evitar la recursión en la carpeta exportada.

Ahora procederemos a la instalación y configuración en cada S.O.

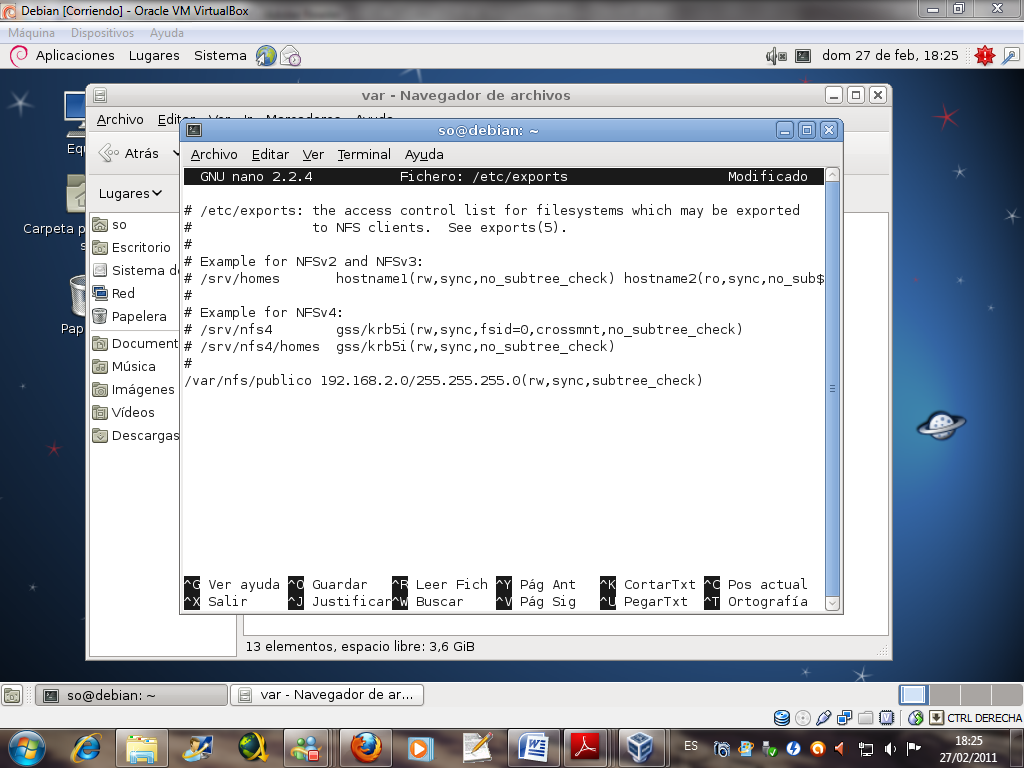
**Debian:**

1. Lo 1º que se repetirá en el resto de sistemas es comprobar que tenemos los paquetes nfs-utils y portmap.

**rpm –q nfs-utils portmap nfs-kernel-server nfs-common**

1. Editaremos el fichero “/etc/exports”, es en donde ponemos las rutas a compartir y sus opciones y a quien se comparte.

En el ejemplo el directorio que se comparte es el “/var/nfs/publico”, creado específicamente con mkdir y con los permisos adecuados, seguidamente d ela dirección se tiene que especificar la ip y su máscara de la red o de un equipo los cuales podrán acceder al recurso compartido, luego entre paréntesis se tienen q poner las opciones en este caso son: rw(escritura y lectura),sync(sincrono) y subtree\_check.

****

1. Configurar un nivel de seguridad com **portmap.** Esto se consigue editando los ficheros:

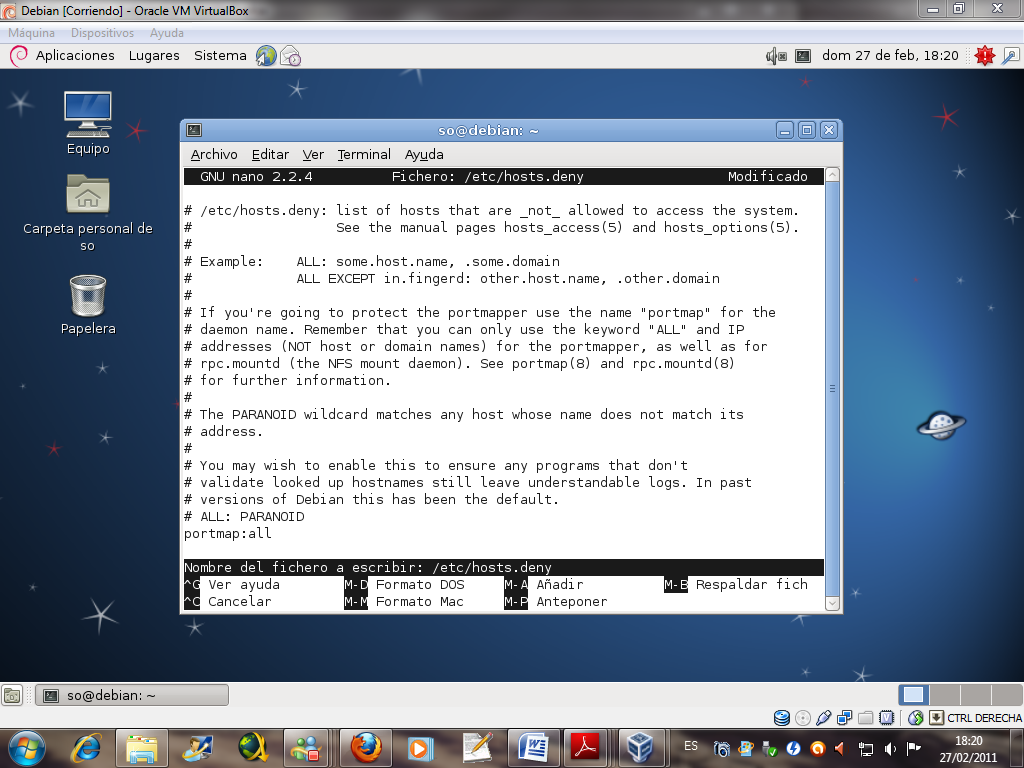
***/etc/hosts.allow***   
 ***/etc/hosts.deny***

Debemos especificar que direcciones IP o rango de direcciones IP pueden acceder a los servicios de portmap y quienes no pueden hacerlo.

**Hosts.allow -> portmap:192.168.2.2/255.255.255.0 (IP maquina cliente)**

****

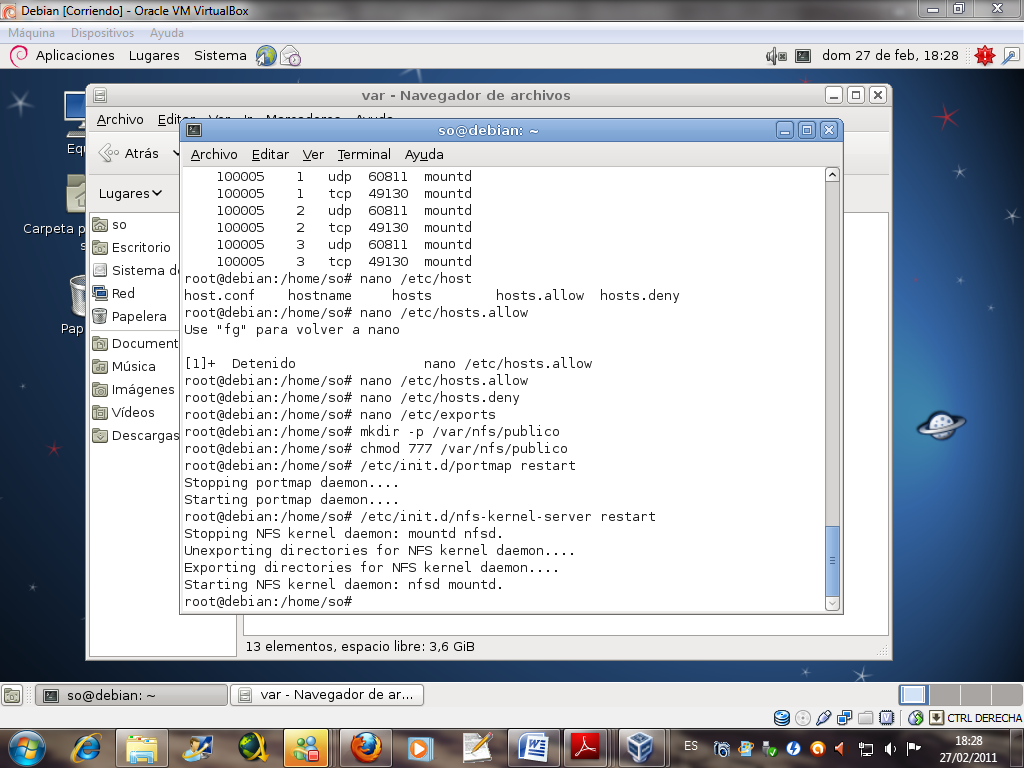
**Hosts.deny -> portmap:ALL**

****

1. Reiniciar demonios:

**/etc/init.d/portmap restart  
 /etc/init.d/nfs-kernel restart**

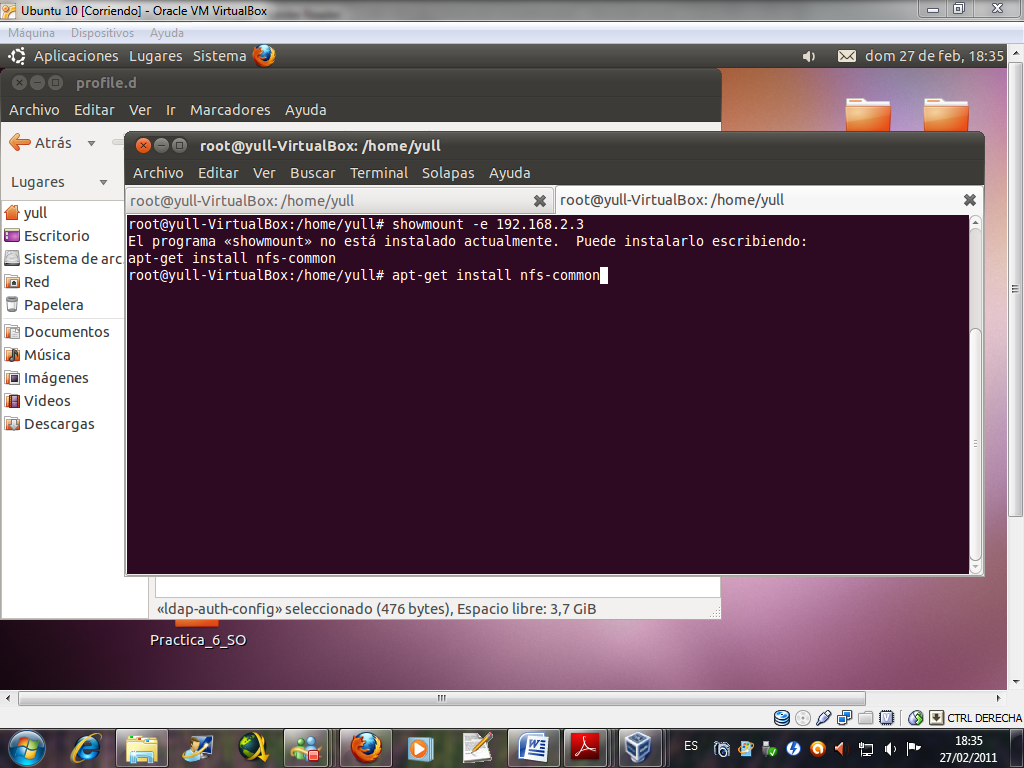
**SERVIDOR DEBIAN**

****

**CLIENTE UBUNTU**

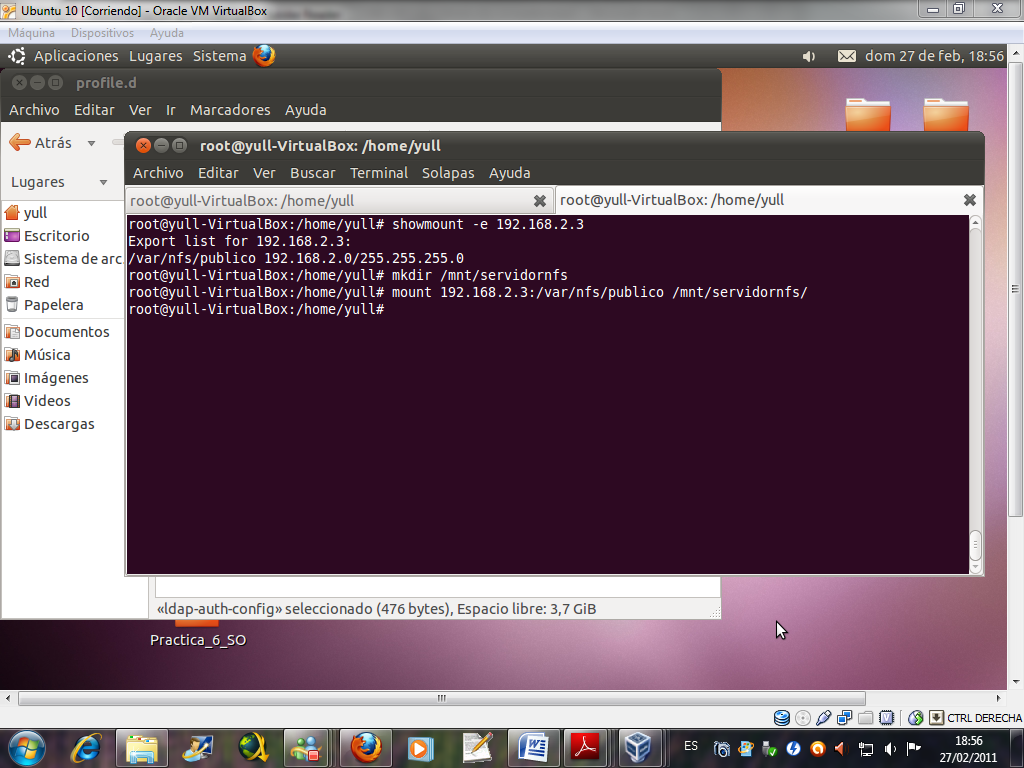
Exportamos la lista nfs del servidor para ver que si que esta y creamos carpeta para luego montar el recurso compartido en ella. Con el comando:

**Showmount –e 192.168.2.2** (ip servidor)

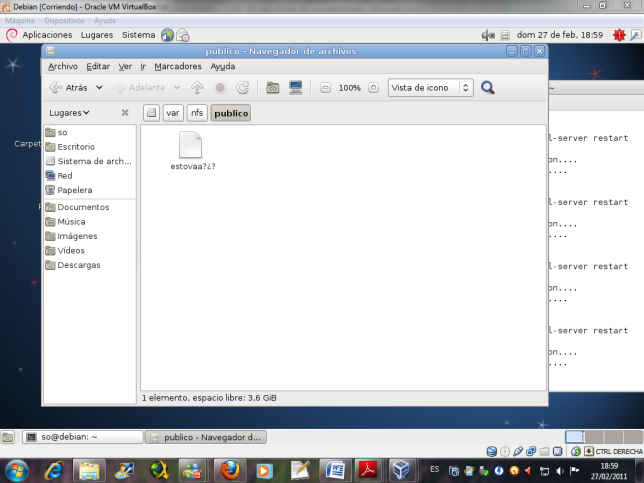


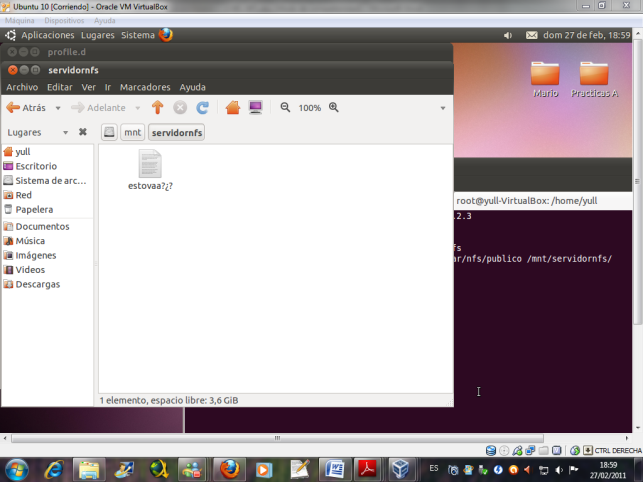
Montaremos el recurso, 1º creamos una carpeta donde montaremos el recurso con el comando mount donde 1º pondremos la ip del servidor luego la ruta q nos la proporciona el paso anterior y por último la dirección donde montaremos el recurso.

**mount 192.168.2.3:/var/nfs/publico /mnt/servidornfs/**



**Ejemplo de funcionamiento**





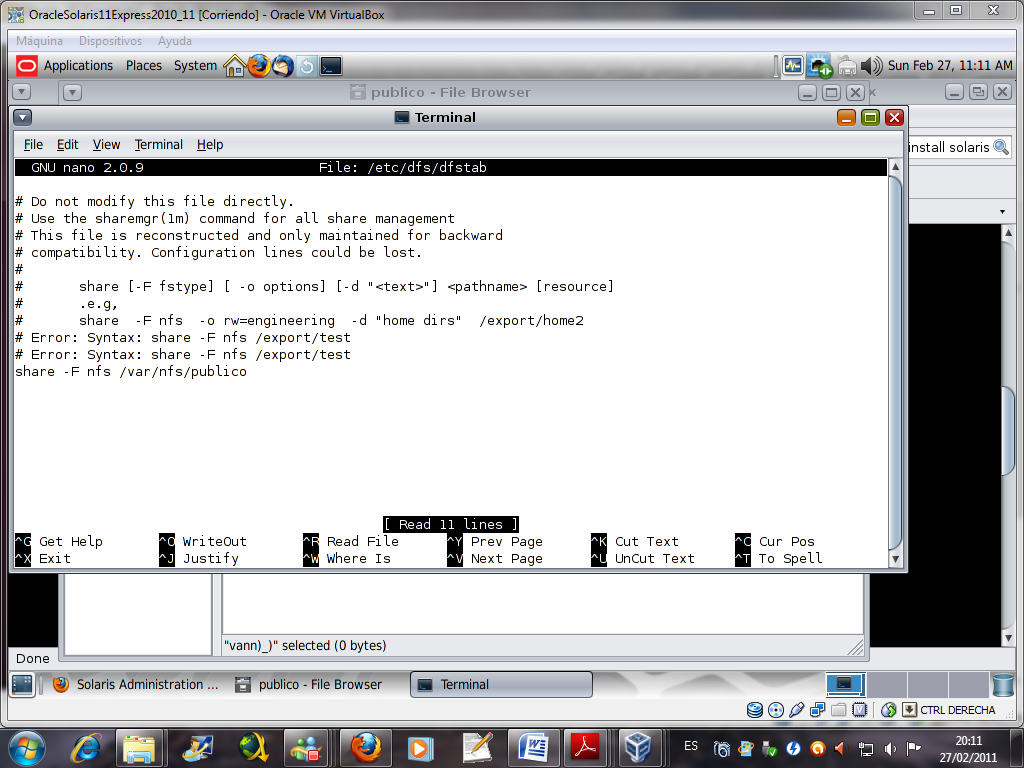
**SOLARIS 11 EXPRESSS**

Seguiremos los mismo pasos aproximadamente que en debían, 1º comprobamos la red y servicio nfs esta online. Y creamos el directorio a compartir.



En solaris el fichero donde pondremos al ruta del directorio a compartir se encuentra en **/etc/dfs/dfsta**, añadimosla siguiente linea:

**share -F nfs /var/nfs/public**

****

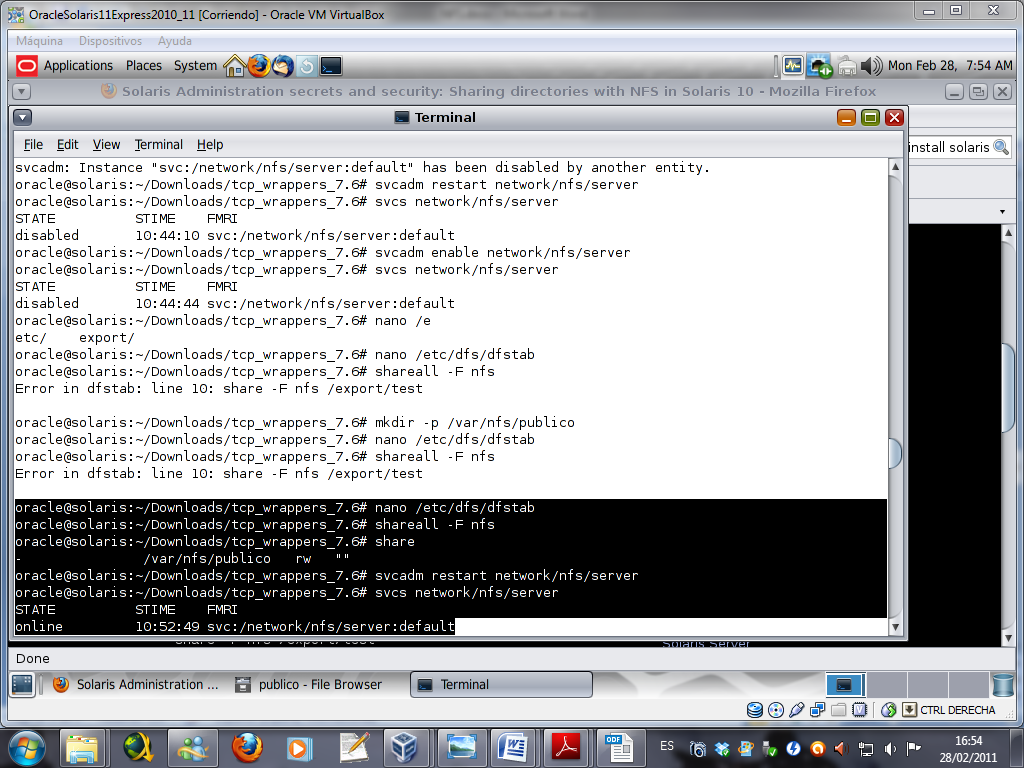
\*Si no ha sido creado el directorio aparecerá “disable” el estado

Comandos realizados:

* Estado server

**svcs network/nfs/server  
 offline**

**svcadm enable -r network/nfs/server  
svcadm enable -s network/nfs/server  
svcadm restart network/nfs/server**

****

* Reiniciar servicio

**svcadm restart network/nfs/server || svcs network/nfs/server**

**\*disable si no creas la carpeta y poner la ruta bien xD  
online**

* Modificar fichero

**editar /etc/dfs/dfstab**

**share -F nfs /export/test**

* Reiniciar servicio

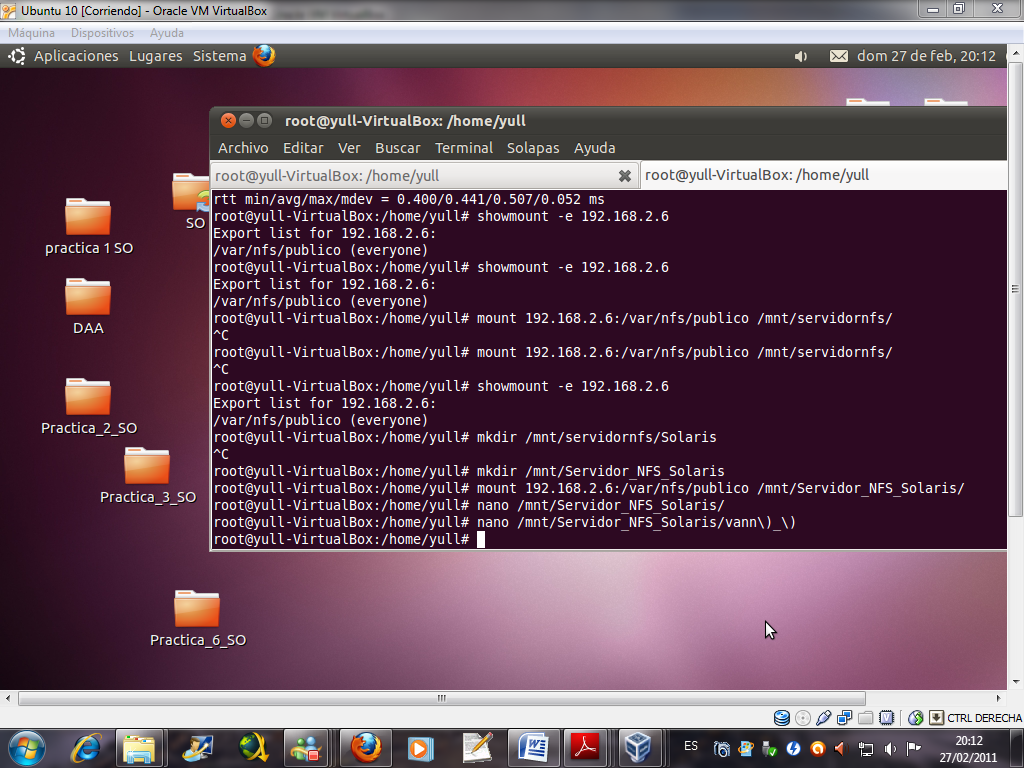
**svcadm restart network/nfs/server || svcs network/nfs/server**

**online**

**shareall -F nfs  
share**

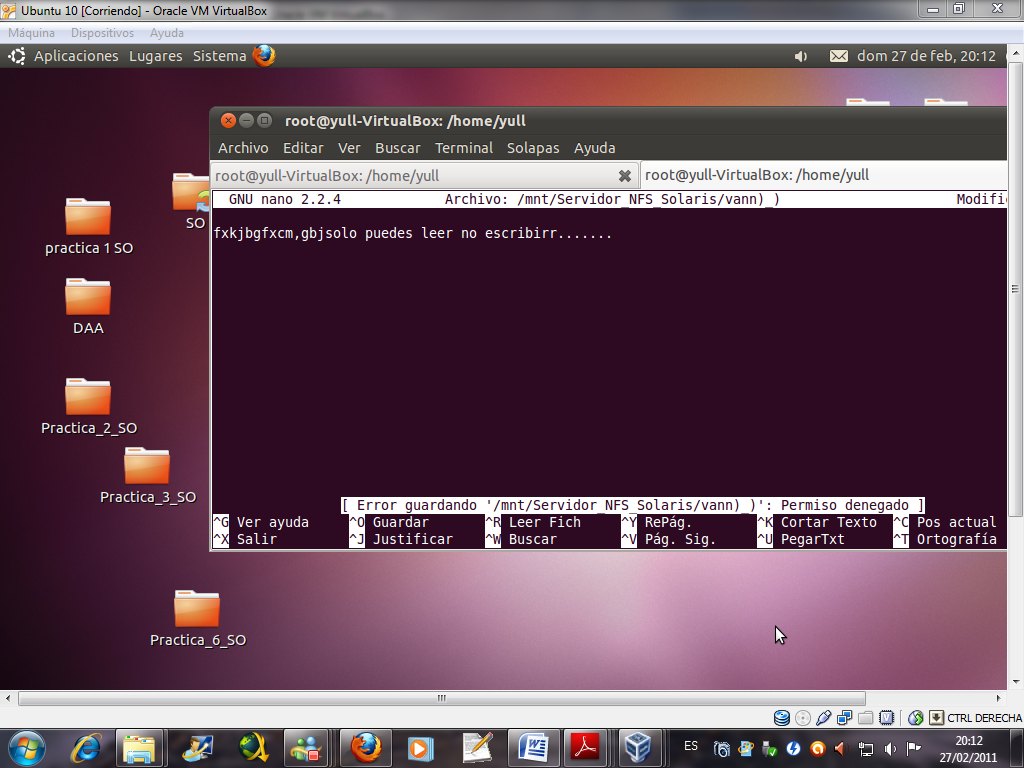
**CLIENTE**

* Crear carpeta y montar, como siempre.



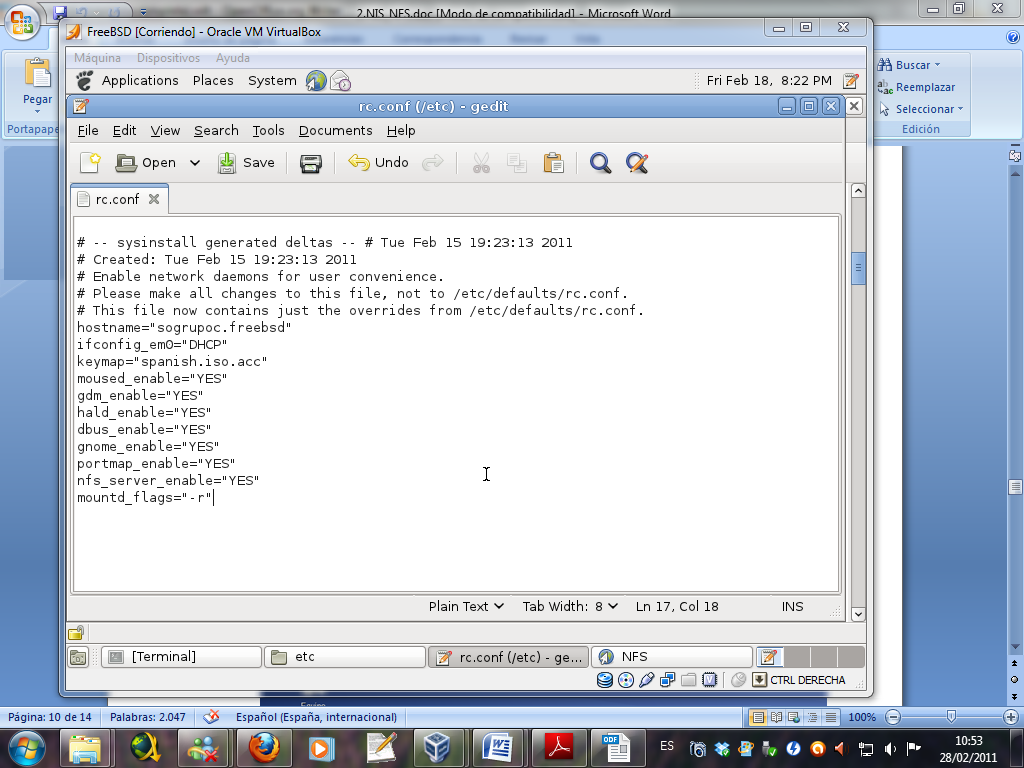
**TEST**

Abrir archivo, teniendo permisos de solo lectura



**Freebsd**

1. Configurara el archivo /etc/rc.conf



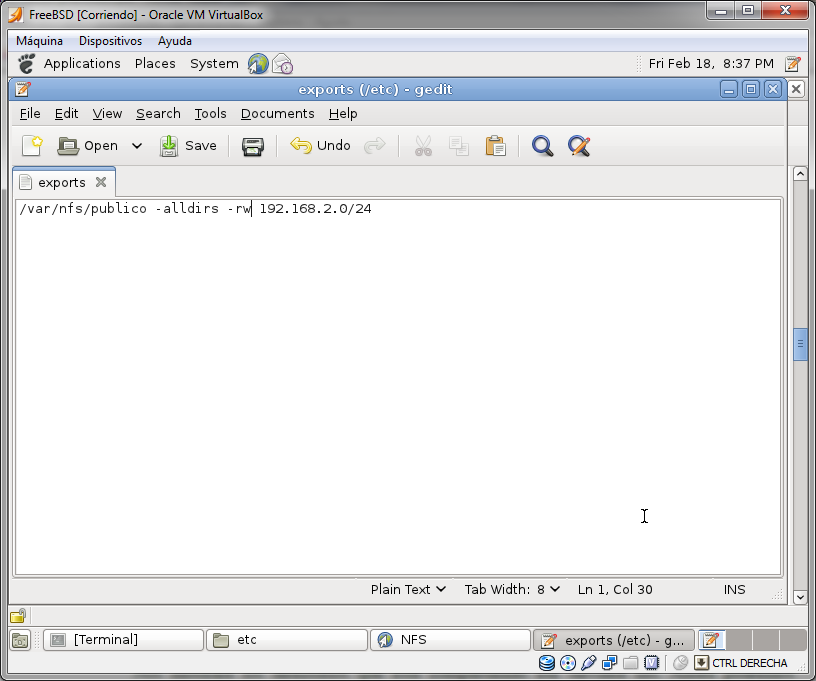
\* Foto errada, hay que poner rpcbind por portmap. Apartir de la version 5.X en FreeBSD utilizaremos el demonio **rpcbind** en vez de portmap.

**rpcbind\_enable="YES"**

**nfs\_server\_enable="YES"**

**mountd\_flags="-r"**

1. Crear el fichero “/etc/exports”, y añadir la ruta y las opciones adecuadas del directorio a compartir, dándonos cuenta que las opciones a diferencia d elos S.O. anteriores se escribían entre paréntesis y en FREEBSD se ponen delante d ela ip y con “–“.



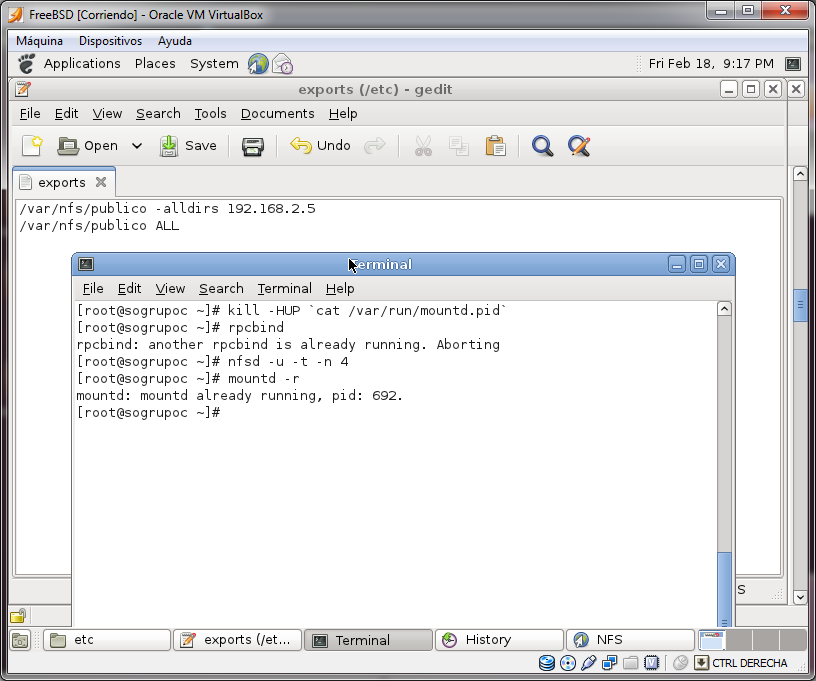
1. Posteriormente reiniciaremos el demonio mountd con, pondremos el servidor NFS en funcionamiento con:

**kill -HUP `cat /var/run/mountd.pid`**

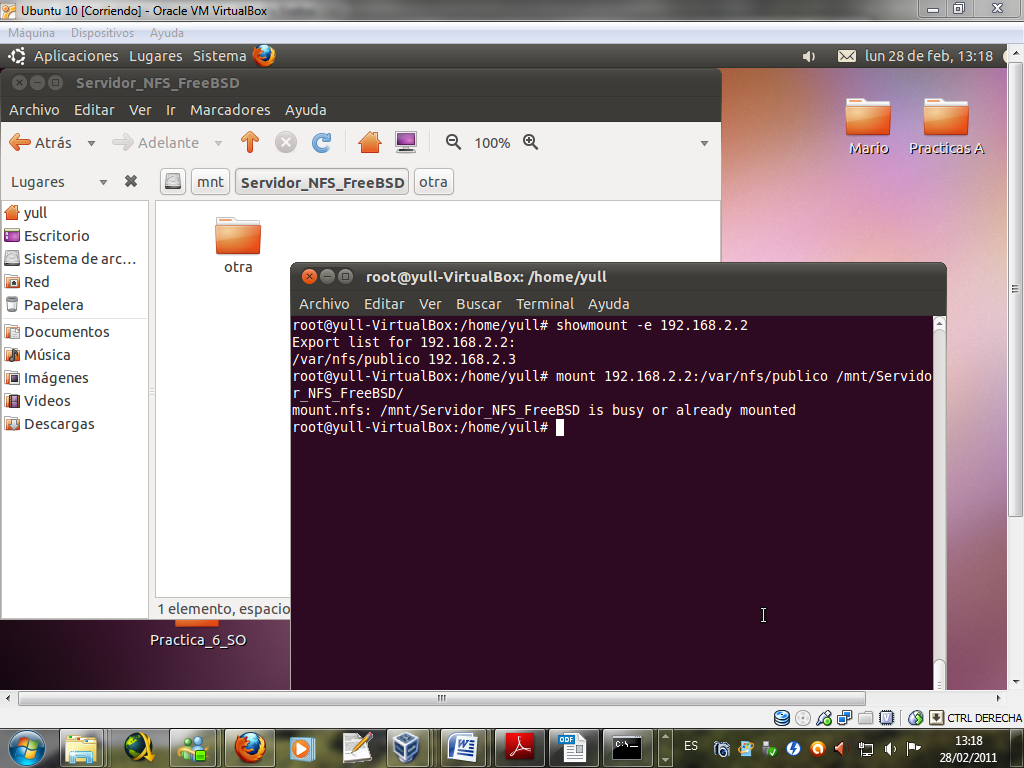
**rpcbind**

**nfsd -u -t -n 4**

**mountd -r**



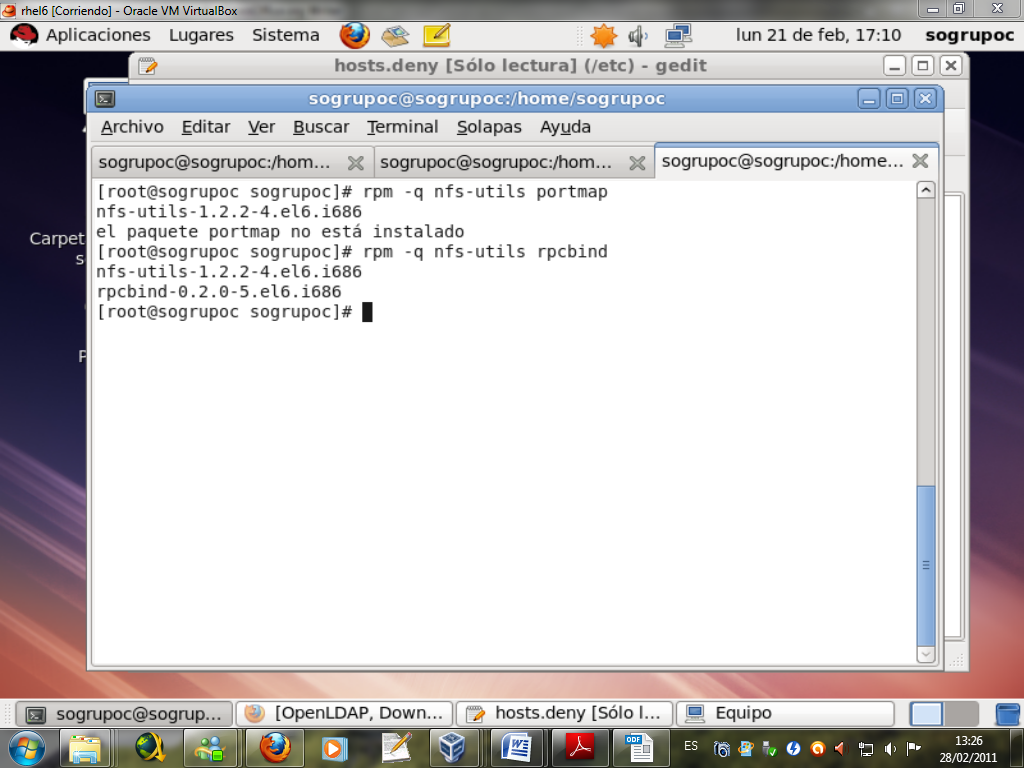
**Cliente & prueba**



**RED-HAT 6**

El primer paso es comprobar que tenemos instalados nfs-utils y portmap.

Portmap no está pero utilizaremos rpcbind como en Opensolaris

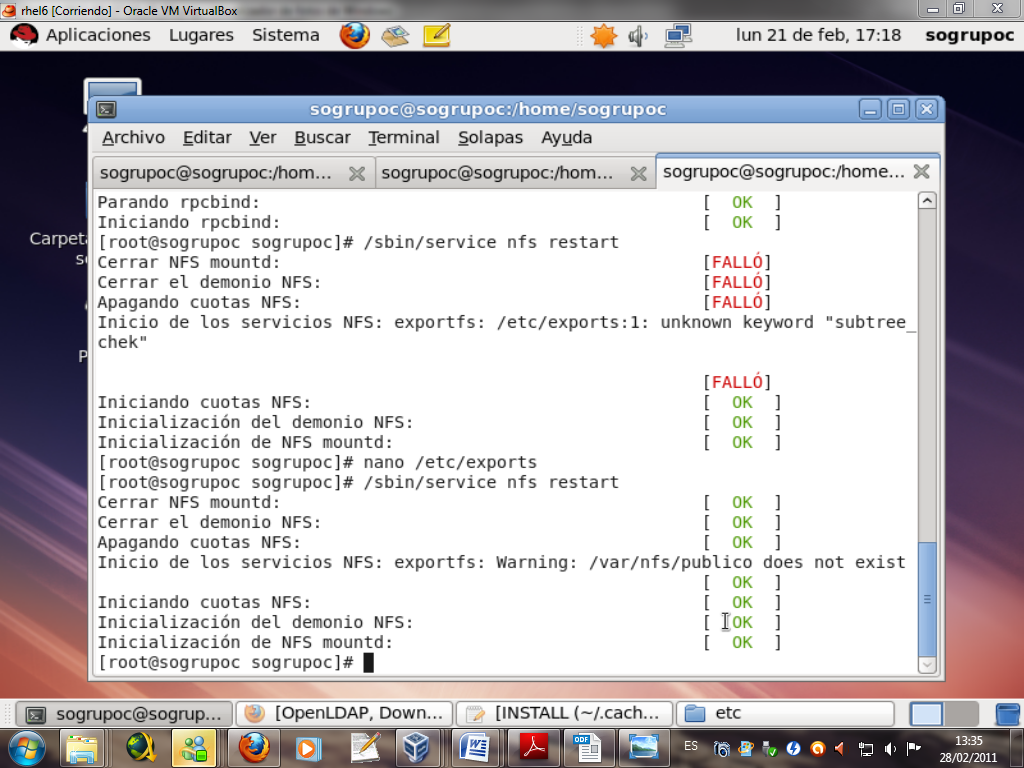


El segundo paso es configurar **/etc/exports , */etc/hosts.allow*** y ***/etc/hosts.deny***. igual que en Debian.

Reiniciar demonios y servicios:

**/sbin/service nfs restart**

**/sbin/service rpcbind restart**



\*Nota el fallo ese es debido a que se escribió mal el en archivo exports,se puso “subtree\_chek” en vez de “subtree\_check”.

Para que no nos de problemas el firewall, lo paramos:

**service iptables stop**

**Cliente & prueba**

Igual que siempre creamos carpeta comprábamos con “swhomount –e ip servidor” y montamos con el comando mount ip-servidor-recurso y ruta donde montar en el cliente:

**Mount 192.168.2.2:/var/nfs/publico /mnt/Servidor\_NFS\_RedHat**

