UNIVERSIDAD LUTERANA SALVADOREÑA

FACULTAD DE CIENCIAS DEL HOMBRE Y LA NATURALEZA

LICENCIATURA DE CIENCIAS DE LA COMPUTACIÓN

ING. MANUEL FLORES VILLATORO

SISTEMAS OPERATIVOS DE REDES

SISTEMA DE ARCHIVOS EN RED

INTEGRANTES		
NOMBRES	**Nº CARNET**	**PARTCIAPACIÓN (%)**
CLAROS PARADA , HERIBERTO DE JESÚS	CP02110925	34%
HERNÁNDEZ PACAS, MARTA ESMERALDA	HP01121473	33%
PEÑA NAVARRO , SARA OLINDA	PN01121384	33%

San Salvador, 18 de Octubre de 2014.

Índice de Contenido **Página**

Índice de Tablas

Índice de Ilustraciones

INTRODUCCIÓN

El documento tiene como propósito describir el proyecto que como grupo se implementara durante el ciclo, El proyecto elegido es el Sistema de Archivo de Red, que consiste en la implementación de un servidor de archivo que tenga soporte para archivos compartidos utilizando los protocolos NFS, SAMBA y FTP.

En una entorno informático es imprescindible disponer de un servicio que permita el acceso seguro a archivos remotos de forma trasparente. En muchas circunstancias hay necesidad de intercambiar información que garantice la seguridad y confidencialidad de la misma; Y NFS proporciona este servicio siguiendo la estructura cliente-servidor. El servidor NFS comparte una serie de directorios seleccionados con unas condiciones de seguridad concretas. El cliente NFS, si está autorizado para ello, puede 'montar' dichos directorios en su propio sistema de archivos pudiendo acceder a los archivos como si fueran locales y de esta forma compartir directorios, con las restricciones adecuadas, y pueden intercambiar archivos dentro de la red de área local.

También el protocolo Samba nos permite compartir archivos, directorios e impresoras entre los diferentes ordenadores de una red local (LAN). Una vez configurado el servidor podremos acceder fácilmente a los directorios compartidos (share) de otros ordenadores y almacenar archivos e intercambiar documentos entre diferentes máquinas como si estuvieran en nuestra propio PC, también podremos compartir las impresoras de la red con extrema facilidad e imprimir desde cualquier máquina aunque no estemos conectados directamente a la impresora. Además Samba permite autentificar a los usuarios mediante contraseñas facilitando un acceso seguro y regulado a los recursos de la red y asegurando la privacidad de los datos que se comparten.

La implementación clásica de un servicio FTP se basa en una arquitectura cliente/servidor. Es éste el encargado de alojar el sistema de directorios y archivos que serán accesibles por parte de aquél.

La idea del protocolo de FTP distribuido (o DFP) consiste en repartir un sistema de transferencia de ficheros entre varios servidores o hosts, coordinados de manera que, de cara a un cliente cualquiera, aparezcan como si se tratara de uno solo. Al adoptar este enfoque, inmediatamente, se adquieren varias ventajas.

OBJETIVOS

Objetivo General:

Mostar diferentes formas de compartir archivos a través de protocolos, permitiendo que un equipo pueda montar y trabajar con un sistema de archivos de otro equipo de red, como si fuera local.

Objetivos Específicos:

- ✓ Exponer los diferentes protocolos que nos permiten compartir archivos en red.
- ✓ Dar a conocer las formas de cómo trabajar con los protocolos SAMBA, FTP, NFS y SSH.
- ✓ Mostrar los elementos y las configuraciones necesarias en cada uno de los protocolos, para la compartición de archivos a través de la red.

MARCO TEÓRICO

Para poder implementar el proyecto debemos estudiar las diferentes formas de protocolos que se utilizan para este fin. Entre los diferentes protocolos se describen:

- ✓ **Samba**
- ✓ **FTP**
- ✓ **NFS**
- ✓ **SSH**
- ✓ **Redes de Almacenamiento NAS y NAN**

SAMBA

Es una implementación de código abierto del protocolo Server Message Block (SMB). Que Permite la interconexión de redes Microsoft Windows, Linux, UNIX y otros Sistemas Operativos juntos, permitiendo el acceso a archivos basados en Windows y compartir impresoras. El uso de Samba de SMB lo hace parecer como un servidor Windows a clientes Windows.

Breve Historia.

Samba fue desarrollado originalmente para Unix por Andrew Tridgell utilizando un sniffer o capturador de tráfico para entender el protocolo usando ingeniería inversa. El nombre viene de insertar dos vocales al protocolo estándar que Microsoft usa para sus redes, el SMB o server message block. En un principio Samba tomó el nombre de smbserver pero tuvieron que cambiarlo por problemas con una marca registrada. Tridgell buscó en el diccionario de su máquina Unix alguna palabra que incluyera las letras.

Características de Samba

Samba es una aplicación de servidor poderosa y versátil. Hasta los administradores bien empapados deben conocer sus habilidades y limitaciones antes de intentar una instalación y configuración.

Lo que Samba puede hacer:

- ✓ Sirve árboles de directorios e impresoras a clientes Linux, UNIX y Windows

- ✓ Asiste en la navegación de la red (con o sin NetBIOS).
- ✓ Autentifica las conexiones a dominios Windows.
- ✓ Proporciona resolución de nombres de Windows Internet Name Service (WINS).
- ✓ Actúa como un Controlador de Dominio Primario (Primary Domain Controller, PDC) estilo Windows.
- ✓ Actúa como un Backup Domain Controller (BDC) para un PDC basado en Samba.
- ✓ Actúa como un miembro servidor de dominio de Active Directory.
- ✓ Une un Windows NT/2000/2003 PDC.

Lo que Samba no puede hacer:

- ✓ Actúa como un Backup Domain Controller (BDC) para un Windows PDC (y viceversa)
- ✓ Actúa como un controlador de dominio de Active Directory

FTP ('Protocolo de Transferencia de Archivos')

FTP ('Siglas en inglés de File Transfer Protocol, Protocolo de Transferencia de Archivos') en informática: Es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP (Transmission Control Protocol), basada en la arquitectura **cliente-servidor**. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

Historia

FTP tiene sus orígenes en 1971, y aunque ha evolucionado con el paso de los años, es uno de los protocolos más antiguos que todavía están en uso. Hoy en día se usa principalmente en redes corporativas y la red más grande que existe, Internet. . La estructura general fue establecida en 1973. Fue modificado varias veces, añadiendo nuevos comandos y funcionalidades. Al final se publicó el RFC 959 en octubre de 1985, que es la que se utiliza actualmente.1

El servicio FTP es ofrecido por la capa de aplicación del modelo de capas de red TCP/IP al usuario, utilizando normalmente el puerto de red 20 y el 21. Un problema básico de FTP es que está pensado para ofrecer la máxima velocidad en la conexión, pero no la máxima seguridad, ya que todo el intercambio de información, desde el login y password del usuario en el servidor hasta la transferencia de cualquier archivo, se realiza en texto plano sin ningún tipo de cifrado, con lo que un posible atacante puede capturar este tráfico, acceder

al servidor y/o apropiarse de los archivos transferidos.

Utilidades del Puerto 20 y 21

El puerto 20 es el utilizado para el flujo de datos entre el cliente y el servidor y el puerto 21 para el flujo de control, es decir, para enviar las órdenes del cliente al servidor (FTPS).

El funcionamiento de FTP

El funcionamiento de FTP es sencillo. Una persona desde su ordenador invoca un programa cliente FTP para conectar con otro ordenador, que a su vez tiene instalado el programa servidor FTP. Una vez establecida la conexión y debidamente autenticado el usuario con su contraseña, se pueden empezar a intercambiar archivos de todo tipo.
Aunque no estemos familiarizados directamente con FTP, las opciones de que lo hayamos usado alguna vez son bastante grandes.

Por ejemplo:

Muchos de los enlaces de descarga que usas en Internet, son URLs que apuntan a un ordenador que está actuando como un servidor FTP: el navegador automáticamente hace la conexión y descarga lo correspondiente; es decir, que el protocolo FTP es el sistema encargado de transferir archivos más estable y fiable que hay en Internet. Esto significa que la descarga y subida de archivos que se hacen tendrán más opciones de completarse si errores de transferencia, y quedarán intactos después del envío.

NORMAS Y ESTANDARES DE FTP

Existen unas normas o estándares que permiten a FTP funcionar en casi cualquier medio. Estas especificaciones son las que hacen que plataformas independientes dentro de Internet puedan comunicarse entre sí. Por ejemplo a continuación algunas normas y estándares de FTP.

- ✓ FTP usa menos cabecera que otros mecanismos de transferencias de archivos, enviando menos paquetes en un sentido y en otro. La razón principal de esto es que FTP puede descargar ficheros en modo binario. Es decir; cuando descargas un fichero usando HTTP (Hyper Text Transfer Protocol), o envías/recibes un archivo añadido a un email, los datos primero se codifican en MIME (Multipurpose Internet Mail Extensions). Básicamente, esto significa que tu fichero es codificado como texto en la transmisión, y vuelve a convertirse en binario al final de la transferencia. Esta codificación aumenta considerablemente el tamaño de la cabecera.

- ✓ El propio protocolo TCP/IP, provee de un sistema de control y corrección de paquetes al ser recibidos en el destino. Si un paquete en la secuencia de envío se pierde, el ordenador que recibe los datos hace

una petición para el reenvió de datos. Esta es la razón de porque TCP/IP es tan fiable. Esto es una ventaja porque FTP funciona sobre el protocolo TCP/IP.

- ✓ Las más modernas versiones de FTP, permiten resumir las descargas que han quedado a medias. En el lado servidor, se incluyen unos marcadores que el cliente puede interpretar para saber desde donde tiene que seguir descargando el archivo. De este modo, en caso de fallo, no tenemos que volver a bajarnos un fichero entero otra vez.

NFS (Network File System)

Definición de NFS (Network File System):

Es un método de compartición de archivos entre máquinas de una red de tal forma que tenemos la impresión de trabajar en nuestro disco duro local. Red HatLinux puede trabajar como servidor o como cliente de NFS (o ambos), lo que implica que puede exportar sistemas de archivos a otros sistemas, así como montar los sistemas de archivos que otras máquinas exportan.

Breve Historia

Primer sistema comercial de archivos en red (Sun Microsystems, 1984) estándar, multiplataforma que permite acceder y compartir archivos en una red C/S heterogénea como si estuvieran en un sólo disco, [[Id_est|i.e.] montar un directorio de una máquina remota en una máquina local.

Una red tiene dos tipos de conexiones:

1. **Conexiones físicas:**

Permiten a las computadoras transmitir y recibir señales directamente. Las conexiones físicas están definidas por el medio empleado (pueden ser cables hasta satélites) para transmitir la señal, por la disposición geométrica de las computadoras (topología) y por el método usado para compartir información, desde textos, imágenes y hasta videos y sonidos.

2. **Conexiones Lógicas o Virtuales:**

Permiten intercambiar información a las aplicaciones informáticas, por ejemplo a un procesador de texto o cualquier tipo de software. Las conexiones lógicas son creadas por los protocolos de red y permiten compartir datos a través de la red entre aplicaciones correspondientes a computadoras de distinto tipo, algunas conexiones lógicas emplean software de tipo cliente-servidor y están destinadas principalmente a compartir archivos e impresoras.

Un sistema de archivo distribuido o sistema de archivos de red:

Es un sistema de archivos de computadoras que sirve para compartir archivos, impresoras y otros recursos como un almacenamiento persistente en una red de computadoras. El primer sistema de este tipo fue desarrollado en la década de 1970, y en 1985 Sun Microsystems creó el sistema de archivos de red NFS el cual fue ampliamente utilizado como sistema de archivos distribuido. Otros sistemas notables utilizados fueron el sistema de archivos Andrew (AFS) y el sistema *Server Message Block SMB*, también conocido como CIFS.

Metodología del sistema NFS

- ✓ Linux usa una combinación de soporte a nivel de kernel y demonios en continua ejecución para proporcionar la compartición de archivos NFS.
- ✓ El soporte NFS debe estar activo en el kernel.
- ✓ NFS usa Remote ProcedureCalls (RPC) para dirigir peticiones entre clientes y servidores.
- ✓ El uso de RPC implica que el servicio portmap debe estar disponible y activo. Paquete "nfs-utils".

Definición de RPC:

Es un mecanismo de comunicación entre procesos que permite que un programa que está ejecutándose en una máquina ejecute código localizado en un sistema remoto. Por ejemplo con RPC es posible hacer llamados a procedimientos localizados en otras máquinas.

Definición de portmap:

Demonio encargado de crear conexiones RPC con el servidor y de permitir o no el acces

SAN Y NAS

Con la creciente cantidad de información almacenada y por la necesidad de tener disponibles miles de datos, han surgido dos soluciones de almacenamiento; las redes SAN (Storage Area Network) y los sistemas NAS (Network Attached Storage).

SAN (Red de área de almacenamiento)

SAN (Red de área de almacenamiento): Es una red concebida para conectar servidores, matrices de discos y librerías de soporte. Principalmente, está basada en tecnología fibre channel y más recientemente en iSCSI. Su función es la de conectar de manera rápida, segura y fiable los distintos elementos que la conforman.

El SAN es un sistema de discos que se conecta a los servidores mediante redes de altísima velocidad (generalmente fibre channel).

Estructura de un SAN Las SAN.

Proveen conectividad de E/S a través de las computadoras host y los dispositivos de almacenamiento combinando los beneficios de tecnologías Fibber Channel y de las arquitecturas de redes brindando así una aproximación más robusta, flexible y sofisticada que supera las limitaciones de DAS empleando la misma interfaz lógica SCSI para acceder al almacenamiento.

Las SAN se componen de tres capas.

1. **Capa Host.**

Esta capa consiste principalmente en Servidores, dispositivos ó componentes (HBA, GBIC, GLM) y software (sistemas operativos).

2. **Capa Fibra:**

Esta capa la conforman los cables (Fibra óptica) así como los SAN Hubs y los SAN switches como punto central de conexión para la SAN.

3. **Capa almacenamiento:**

Esta capa la componen las formaciones de discos (Disk Arrays, Memoria Caché, RAIDs) y cintas empleados para almacenar datos.

La red de almacenamiento puede ser de dos tipos:

1. **Red Fibre Channel.**

La red Fibre Channel es la red física de dispositivos Fibre Channel que emplea Fibre Channel Switches y Directores y el protocolo Fibre Channel Protocol (FCP) para transporte (SCSI-3 serial sobre Fibre Channel).

1. **Red IP.**

Emplea la infraestructura del estándar LAN con hubs y/o switches Ethernet interconectados. Una SAN IP emplea iSCSI para transporte (SCSI-3 serial sobre IP).

¿Qué es iSCSI (Interfaz estándar de equipos pequeños de Internet)?

Es un método de conexión de dispositivos de almacenamiento a través de una red mediante TCP/IP. Puede usarse a través de una red de área local (LAN), una red de área extensa (WAN) o Internet.

Los dispositivos iSCSI son discos, cintas, CDs y otros dispositivos de almacenamiento de otro equipo en red al que se puede conectar. A veces, estos dispositivos de almacenamiento forman parte de una red denominada red de área de almacenamiento (SAN).

En la relación entre su equipo y el dispositivo de almacenamiento, su equipo se denomina iniciador porque inicia la conexión al dispositivo, que se denomina destino.

NAS (Network Attached Storage)

NAS (Network Attached Storage): Es una tecnología de almacenamiento dedicada a compartir la capacidad de almacenamiento de una computadora (Servidor) con ordenadores personales o servidores clientes a través de una red (normalmente TCP/IP), haciendo uso de un Sistema Operativo optimizado para dar acceso con los protocolos SAMBA, NFS, FTP o TFTP.

El NAS es un sistema de discos que se conecta a la red como cualquier otro dispositivo y se le asigna una dirección IP como un miembro más de la red.
Generalmente, los sistemas NAS son dispositivos de almacenamiento específicos a los que se accede desde los
equipos a través de protocolos de red (normalmente TCP/IP). También se podría considerar que un servidor Windows que comparte sus unidades por red es un sistema NAS, pero la definición suele aplicarse a sistemas específicos. Los protocolos de comunicaciones NAS son basados en ficheros por lo que el cliente solicita el fichero completo al servidor y lo maneja localmente, están por ello orientados a información almacenada en ficheros de pequeño tamaño y gran cantidad. Los protocolos usados son protocolos de compartición de ficheros como NFS, Microsoft Common Internet File System (CIFS).Muchos sistemas NAS cuentan con uno o más dispositivos de almacenamiento para incrementar su capacidad total. Normalmente, estos dispositivos están dispuestos en RAID (Redundant Arrays of Independent Disks) o contenedores de almacenamiento redundante.

Diferencias entra SAN y NAS

La mayor diferencia entre el SAN y el NAS es que el primero está conectado a los servidores mediante redes de altísima velocidad (normalmente canales de fibra) y el segundo está conectado a la red local, donde su desempeño depende de la velocidad de la misma.

En una SAN la información se almacena en la red SAN, y en el modelo NAS los clientes tienen que solicitar los archivos a los servidores para que éstos se los suministren.
Características SAN/NAS

	NAS	SAN
Tipo de datos	Archivos compartidos	Datos a nivel de bloque
Cableado utilizado	Ethernet LAN	Fibre channel dedicado
Clientes principales	Usuarios Finales	Servidores de aplicaciones
Acceso a disco	A través del dispositivo NAS	Acceso directo

SSH (Secure SHell : intérprete de órdenes segura)

SSH (o Secure SHell) es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse a un host remotamente. A diferencia de otros protocolos de comunicación remota tales como FTP o Telnet, SSH encripta la sesión de conexión, haciendo imposible que alguien pueda obtener contraseñas no encriptadas.

Breve Historia.

La primera versión del protocolo y el programa eran libres y los creó un finlandés llamado Tatu Ylönen, pero su licencia fue cambiando y terminó apareciendo la compañía SSH Communications Security, que lo ofrecía gratuitamente para uso doméstico y académico, pero exigía el pago a otras empresas. En el año 1997 (dos años después de que se creara la primera versión) se propuso como borrador en la IETF.

A principios de 1999 se empezó a escribir una versión que se convertiría en la implementación libre por excelencia, la de OpenBSD, llamada OpenSSH.

SFTP es un protocolo de transferencia de archivos que utiliza SSH para asegurar los comandos y los datos que se transfieren entre el cliente y el servidor. Los datos transferidos con FTP estándar no están cifrados, lo que los hace vulnerables a escuchas furtivas, interferencias o falsificaciones.

Con SFTP, los datos transferidos entre el cliente y el servidor están cifrados, lo que evita que usuarios no

autorizados tengan acceso a ellos. Debería usar SFTP cuando necesite transferir datos confidenciales o de carácter crítico entre un cliente y un servidor configurado para usar SSH en transferencias seguras.

Cómo funciona SFTP

Existen dos componentes básicos para la transferencia de archivos SFTP; validación del servidor y autenticación del cliente. Estos dos componentes usan claves públicas y privadas para autenticar la comunicación entre el cliente y el servidor. Se valida el servidor comparando su clave pública con las claves públicas almacenadas en el equipo cliente. La clave pública del servidor está habitualmente almacenada en un archivo llamado "known_hosts" en el servidor, y la clave pública del cliente está almacenada en un archivo cifrado en el equipo local.

SFTP vs. FTPS

SFTP y FTPS son dos protocolos completamente distintos.

- ✓ SFTP asegura las transmisiones con SSH, mientras que FTPS usa seguridad SSL
- ✓ El valor de puerto estándar para FTP es 21. El puerto predeterminado para SFTP 22

Características de SSH

El protocolo SSH proporciona los siguientes tipos de protección:

- ✓ Después de la conexión inicial, el cliente puede verificar que se está conectando al mismo servidor al que se conectó anteriormente.
- ✓ El cliente transmite su información de autenticación al servidor usando una encriptación robusta de 128 bits.
- ✓ Todos los datos enviados y recibidos durante la sesión se transfieren por medio de encriptación de 128 bits, lo cual los hacen extremamente difícil de descifrar y leer.
- ✓ El cliente tiene la posibilidad de reenviar aplicaciones desde el servidor, proporciona un medio seguro para usar aplicaciones gráficas sobre una red.

El protocolo SSH (Secure Shell): Es una herramienta que nos permite conectarnos a equipos remotos (Servidores en Producción) así mismo, nos da la capacidad de llevar a cabo tareas administrativas dentro del mismo como, activar o apagar servicios. Además de la conexión a otros equipos, SSH nos permite copiar datos de forma segura, gestionar claves RSA para no escribir claves al conectar a las máquinas y pasar los datos de cualquier otra aplicación por un canal seguro tunelizado mediante SSH. Una clave RSA (Sistema Criptografico con Clave Publica) es un algoritmo que genera un par de llaves de autenticacion, la publica y la

privada. La publica se distribuye en forma autenticada y la privada que generalmente es guardada en secreto por el propietario. El protocolo SSH (Secure Shell) esta implementado bajo el estándar TCP/IP, el cual a su vez se encuentra dividido en 5 secciones:

1. Nivel Físico.
2. Nivel De Enlace.
3. Nivel de Internet.
4. Nivel de Transporte.
5. Nivel de Aplicación.

La capa de aplicación es el nivel que los programas más comunes utilizan para comunicarse a través de una red con otros programas. Los procesos que acontecen en este nivel son aplicaciones específicas que pasan los datos al nivel de aplicación en el formato que internamente use el programa y es codificado de acuerdo con un protocolo estándar.

Los programas SSH se suministran normalmente en dos paquetes llamados generalmente:

1. **openssh-server =** El primero de ellos debe estar instalado necesariamente en la máquina remota a la que se quiere acceder.
2. **openssh-client.**: Este debe estar instalado en la máquina cliente (la mayoría de las distribuciones Linux lo instalan por omisión). Del lado del servidor, el firewall debe aceptar conexiones entrantes al puerto configurado para SSH.

El puerto por omisión es el 22, pero es posible cambiarlo por cualquier otro que el administrador considere conveniente.

INFORMACION SOBRE LA CONSTRUCCION DEL PROYECTO

A continuación se describes comandos utilizados para llevar a cabo el desarrollo práctico de cada uno de los protocolos:

Comandos utilizados para el protocolo samba

- ✓ **apt-get install** = Se utiliza para la instacion deprogramas y paquetes desde la terminal.
- ✓ **#adduser -system -no-create-home -uid 603 usuariosmb=** Procedemos a crear los usuarios, asignando nombre y número de identificación por usuarios.
- ✓ **#cat /etc/passwd | grep usuariosmb =** Comprobamos que los usuarios se añadieron correctamente.
- ✓ **# smbpasswd -a usuariosmb =** Con este comando le asignaremos el password a los usuarios

creados, nos pedirá que ingresemos un password y luego la confirmación del mismo, esto lo haríamos hasta terminar con todos los usuarios a crear.

- ✓ **# nano /etc/protocolo=** Editor para configuración del archivos.
- ✓ **#mkdir= C**rea directorios que se desea compartir.
- ✓ **#chmod 777=** Otorga permisos a los directories creados.
- ✓ **#testparm=** Muestra una información de acuerdo a la configuración ingresada.
- ✓ **# /etc/init.d/samba restart=** Reinicia el servicio de los programas después de una configuración.
- ✓ **cd /=** Se utiliza para entrar a carpetas o directorios.

Comandos utilizados para el protocolo ftp

- ✓ **#aptitude install gftp=** Se utiliza para la instalación de programas y paquetes desde la terminal.
- ✓ **#useradd -d /home/FTP -s /bin/false/usuario1=** Crearemos un usuario.
- ✓ **#chown -R usuario1 /home/FTP=** Dar al usuario el control de la carpeta. Que en este caso es el "usuario1"
- ✓ **#nano /etc/proftpd/proftpd.conf=** Configurar el documento del fichero **proftpd.conf.**
- ✓ **#/etc/init.d/proftpd restart=** Reinicia el **proftpd.**
- ✓ **#aptitude install gftp= I**nstalamos gftp.
- ✓ **#gftp=** Muestra la ventana en donde se colocara la contraseña, IP y Usuario para compartir archivos.

Comandos utilizados para el protocolo nfs.

- ✓ **#aptitude install nfs-kernel-server=** Instalamos el servidor NFS.
- ✓ **#mkdir= C**rea directorios que se desea compartir.
- ✓ **#chmod 777=** Otorga permisos a los directories creados.
- ✓ **#nano /etc/export=** Configura el archivo export para lo necesario.
- ✓ **# /etc/init.d/nfs-kernel-server restart=** Reiniciamos el servidor nfs.
- ✓ **#aptitude install nfs-common=** Instalacion del cliente nfs.
- ✓ **#mount -t nfs -o rw,nosuid 192.168.1.4:/home/nfs /home/nfs=** Permite montar el directorio compartido .
- ✓ **showmount -e ipservidor=** Verifica que directorios está compartiendo.

Comandos utilizados para el protocolo ssh.

- ✓ **# aptitude install ssh=** Instala ssh server.
- ✓ **#useradd -d /"Ruta del directorio home" -s /bin/bash "nombre del usuario"=** Creamos los usuarios que podrán conectarse al servidor.
- ✓ **#nano /etc/ssh/sshd_config=** Configua el archivo **sshd_config.**
- ✓ **#/etc/init.d/ssh restart=** Reiniciamos el servidor ssh.
- ✓ **#aptitude install sshfs=** Instala el paquete sshfs para el cliente.
- ✓ **#modprobe fuse= Comando para que** sistema cargue automáticamente **sshfs** al arrancar.
- ✓ **#mkdir= C**rea directorios que se desea compartir.
- ✓ **#chmod 777=** Otorga permisos a los directories creados.
- ✓ **$sshfs cliente3ssh@192.168.1.4:/home/ssh/cliente3ssh /home/herick/ssh=**
- ✓ **$sshfs cliente3ssh@192.168.1.4:/home/ssh/cliente3ssh /home/herick/ssh=** Este commando es utilizado para montar los archivos que queremos compartir. Clocando la dirección correcta.
- ✓ **$fusermount -u /home/herick/ssh=** Muestra el directorio compartido.

Requerimientos Preliminares para la Realización de las Prácticas de cada uno de los Protocolos.

1. Dos Computadoras (Servidor de Archivos y Cliente).
2. Sistema Operativo GNU/Linux (Debían).
3. Switch.
4. Cables UTP.

BUENAS PRÁCTICAS

- ✓ Debido a que NFS confía en la información que recibe de la red, es vital asegurar que sólo las máquinas que deban utilizarlo puedan conectarse a los varios servidores RPC necesarios. El firewall también debe bloquear falseado de IPs («IP spoofing») para prevenir que una máquina externa actúe como una interna y que el acceso a los puertos apropiados esté restringido a las máquinas que deban acceder a espacios compartidos por NFS.

- ✓ Es sumamente recomendable el uso de un demonio de sincronización horaria en todos los nodos de su red para mantener sincronizados los relojes de cliente y servidor. ¡Si no hay un ajuste preciso en los relojes de todos los nodos, el NFS puede presentar retrasos indeseados.

- ✓ Aplicar los permisos según sean necesarios a cada una de las carpetas para evitar que sean utilizadas maliciosamente.

- ✓ Restringir el acceso ftp y ssh a una pequeña lista de usuario.

- ✓ No exportar directorios con permisos de escritura salvo que se necesario.

- ✓ En ssh disminuir el logringrace tima para para evitar disminuir el tiempo en que se puedan realizar ataques de fuerza bruta.

DESCRIPCIÓN DEL PROYECTO

El proyecto realizado consiste: en la elaboración de un sistema de archivos en red, que permite que un equipo pueda montar y trabajar con un sistema de archivos de otro equipo de red, como si fuera local, es decir de otra manera compartiendo carpetas y su contenido así como discos duros. El sistema tiene soporte samba para poder interactuar con equipos GNU/Linux y permitir que estos tengan acceso a los archivos compartidos en la red, además de soporte SAMBA,FTP, SSH, y NFS para poder acceder desde cualquiera de esos protocolos.

Los sistemas de archivos en red son de mucha utilidad en diferentes maneras, ya que los escenarios en que se hace necesario el compartir archivos es muy amplio por ejemplo dentro del ámbito escolar el maestro puede necesitar compartir una guía de ejercicio con sus alumnos fácilmente lo puede poner en una carpeta y así todos los alumnos podrían ver el archivo, con este ejemplo podemos describir la funcionalidad que pretendemos tenga nuestro proyecto.

El Sistema instalo es el Sistema Operativo **Debían** con los Protocolos NFS, FTP, SSH, y SAMBA, de modo que compartiremos archivos. Pero para poder compartir un directorio en NFS se instalaron algunos paquetes como: **Nfs-common y nfs-kernel-server** y realizar al respectivas configuraciones del Servidor NFS y de esa manera compartir archivos y que usuarios podrán acceder a los archivos compartidos y que tipo de permiso tendrán.

DIAGRAMA DE RED

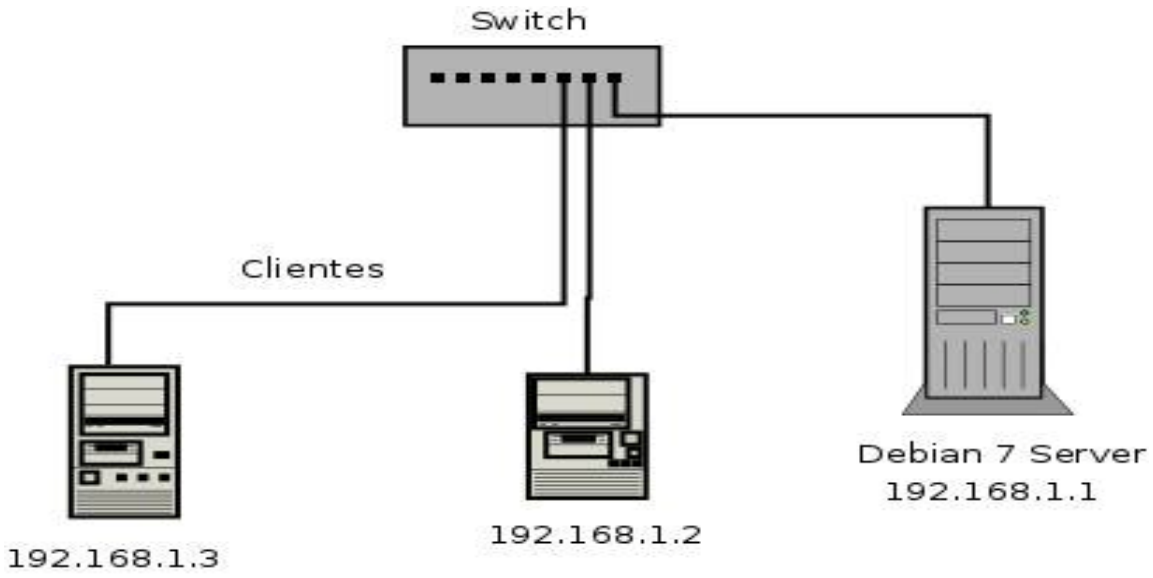
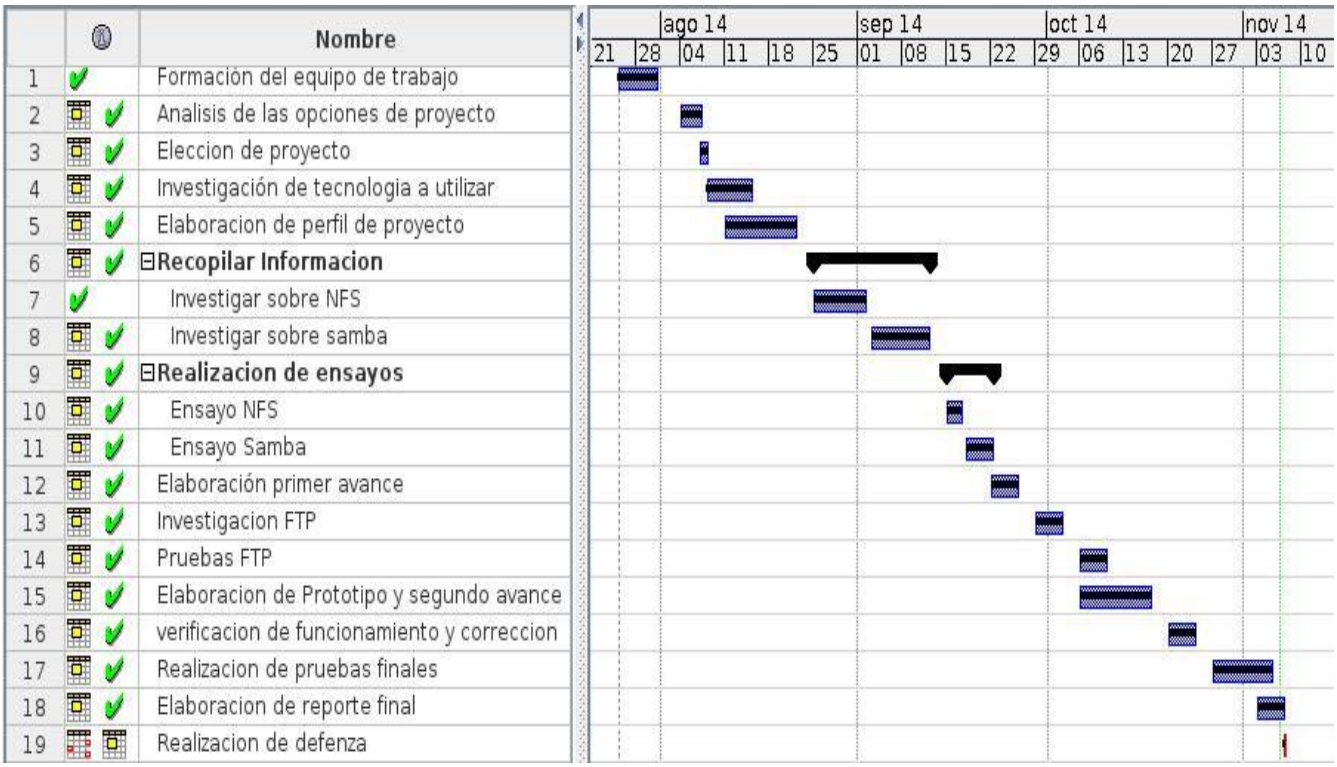


DIAGRAMA DE GANTT



VIABILIDAD O FACTIBILIDAD

Factibilidad Económica

Detalle	Cantidad	Costo
PC como servidor	1	$ 400.00
Clientes	2	$ 800.00
Switch	1	$ 40.00
S.O GNU/Linux (debían)	1	$ 0.00
Mano de Obra	3	$ 500.00
	Total	**$1740.00**

Observando los datos anteriores y valorando la importancia que aportara este proyecto en cualquier tipo de entidad que lo sugiera, es relativamente accesible y cómodo. Recompensando lo presupuestado resultados de productividad.

Factibilidad Operativa

El equipo operacional del proyecto es la ejecución y control del proyecto,

Factibilidad Técnica.

La implementación de un sistema de archivos en red es una opción acertada en un ambiente en que se requiera compartir archivos con grupos de personas de una manera constante y rápida además de que permitiría establecer permisos sobre dichos documentos compartidos

Factibilidad Legal

Al llevar a cabo este proyecto no estaremos faltando a ninguna ley o estándar que lo demande, en cuanto al Software a utilizar se usara GNU/LINUX libre y gratuito. No habrá ningún contrato de alguna licencia.

CONCLUSIÓN

Como equipo de trabajo concluimos que; Compartir archivos en red permite una forma más directa de interactuar con el cliente y servidor, pues de esta manera generamos un círculo una conexión donde la pc servidor es la madre generadora de información que los clientes desean y pueden obtener desde su pc.

Con la elaboración de este proyecto logramos describir y entender el funcionamiento de cada uno de los protocolos y las configuraciones necesarias de cada uno de ellos.

BILIOGRAFIA

- ✓ Debian 7.0 Wheezy - FTP Server - Install Vsftpd : Server World. (n.d.). Retrieved November 7, 2014, from http://www.server-world.info/en/note?os=Debian_7.0&p=ftp
- ✓ Hernando, V. (2012, November 4). SSHFS: Acceder localmente a un directorio remoto. Retrieved November 7, 2014, from http://claves-de-linux.blogspot.com/2012/11/sshfs-montar-directorio-remoto.html
- ✓ Instalación y Configuración de SSH (OpenSSH) en GNU/Linux (modo consola) - wiki de elhacker.net. (n.d.). Retrieved November 7, 2014, from http://wiki.elhacker.net/redes/administracion-de-redes-gnu-linux/instalacion-y-configuracion-de-ssh-openssh-en-gnu-linux-modo-consola
- ✓ Instalar servidor FTP y restringir permisos a los usuarios. (n.d.). Retrieved November 7, 2014, from http://rootear.com/ubuntu-linux/instalar-servidor-ftp
- ✓ Montar carpetas remotas con ssh y ftp. (n.d.). Retrieved November 7, 2014, from http://www.cambiatealinux.com/montar-carpetas-remotas-con-sshfs-y-ftp/
- ✓ Secure Shell (SSH) - Manuales. (n.d.). Retrieved November 7, 2014, from http://www.linuxparatodos.net/web/comunidad/base-de-conocimiento/-/wiki/Base+de+Conocimiento/Secure+Shell+(SSH)

ANEXOS

MANUAL PARA COMPARTIR ARCHIVOS EN RED CON SAMBA

Paso 1

- ✓ Como usuario root instalamos el servidor samba usando **apt-get install samba.**

```
herick@server:~$ su
Contraseña:
root@server:/home/herick# apt-get install samba
```

Paso 2

- ✓ Procedemos a crear los usuarios, asignando nombre y número de identificación por usuarios de la siguiente manera.
  #adduser -system -no-create-home -uid 603 usuariosmb

  Este comando se repite de acuerdo al número de usuarios que queremos crear, con la variante que el número (uid incrementaría de acuerdo al numero de usuarios que dese crear).

```
herick@server:~$ su
Contraseña:
root@server:/home/herick# adduser -system -no-create-home -uid 603 usuariosmb
```

Paso 3

- ✓ Comprobamos que los usuarios se añadieron correctamente.

  #cat /etc/passwd | grep usuariosmb

```
root@server:/home/herick# cat /etc/passwd | grep usuariosmb
usuariosmb:x:603:65534::/home/usuariosmb:/bin/false
root@server:/home/herick#
```

Paso 4

- ✓ Añadimos los usuarios a samba de la siguiente manera.

  # smbpasswd -a usuariosmb

Con este comando le asignaremos el password a los usuarios creados, nos pedirá que ingresemos un password y luego la confirmación del mismo, esto lo haríamos hasta terminar con todos los usuarios a crear.

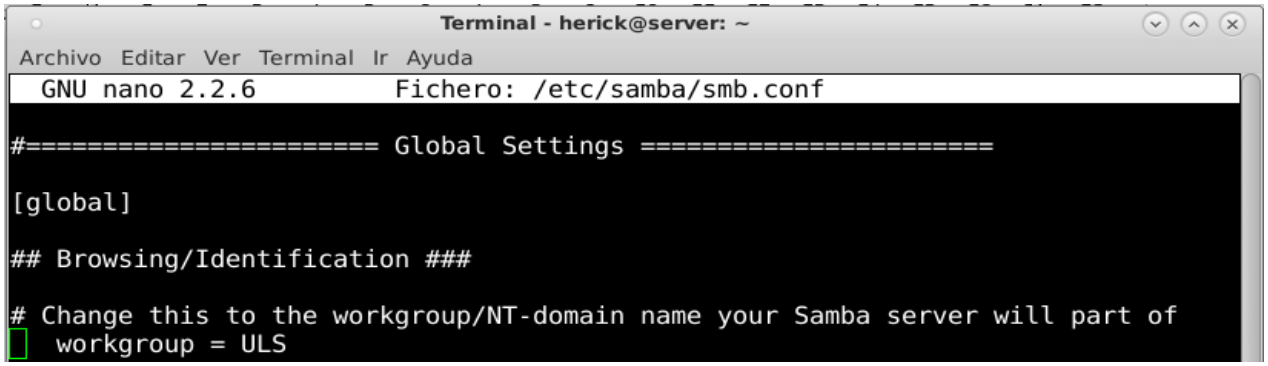
```
root@server:/home/herick# smbpasswd -a usuariosmb
New SMB password:
Retype new SMB password:
Added user usuariosmb.
root@server:/home/herick#
```

Paso 5

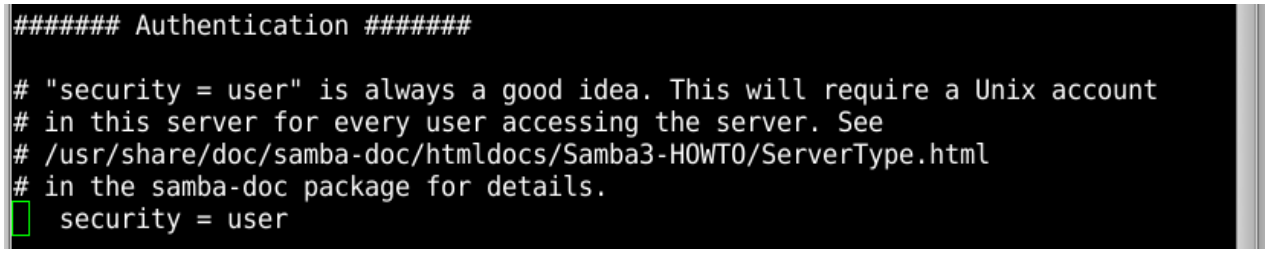
- ✓ Procedemos a la configuración del archivo de samba.

 # nano /etc/samba/smb.conf

(Procedemos a poner el nombre del grupo al que pertenece la PC para eso buscamos la línea Workgroup en la sección global).



(Habilitamos la autenticación del usuario des comentando la línea security de la sección Authentication).



(Procedemos a configurar el archivo con el recurso compartido).
Nos ubicamos en las últimas líneas para poder agregar nuestra configuración

Esta es la configuración que ingresaremos en nuestro archivo de samba

[Doc]----------->Nombre del recurso (como lo deban los clientes)
comment=Servidor samba---------->Comentarios
path=/homet/doc---------->Ruta del directorio a compartir
read only=yes---------->Permiso de lectura en general
browseable=yes---------->Visibilidad del recurso en la red
write list=usuariosmb---------->Lista de usuarios permitidos a escritura
valid users=usuariosmb---------->Lista de usuarios válidos para ingresar al recurso compartido
directory mask=0775---------->Permisos de escritura o lectura en recurso compartido
create mask=0644---------->Permisos de escritura o lectura para archivos

```
[DOC]
comment=Servidor samba
path=/home/doc
read only=yes
browseable=yes
write list=usuariosmb
valid users=usuariosmb
directory mask=0775
create mask=0644
```

Después de haber configurado los recursos compartidos Guardamos el archivo y lo cerramos (ctrl+o, enter y ctrl+x).

Paso 6

- ✓ Como no tenía creado el directorio que se desea compartir, se procede a crearlo y darle permisos.

#mkdir /home/doc

#chmod 777 /home/doc

```
root@server:/home/herick# nano /etc/samba/smb.conf
root@server:/home/herick# mkdir /home/doc
root@server:/home/herick# chmod 777 /home/doc
root@server:/home/herick#
```

Paso 7

- ✓ Proceder a hacer el testeo del archivo de configuración para ver si está correcto, ejecutamos el siguiente parámetro el cual nos mostrara una información de acuerdo a la configuración ingresada.

#testparm

```
root@server:/home/herick# testparm
Load smb config files from /etc/samba/smb.conf
rlimit_max: increasing rlimit_max (1024) to minimum Windows limit (16384)
Processing section "[homes]"
Processing section "[printers]"
Processing section "[print$]"
Processing section "[Osmany]"
Processing section "[DOC]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions
```

Luego de presionar ENTER nos mostrará las configuraciones que tiene dicho archivo, recuerde que si todo esta hecho bien, dará esta información al momento de ir dando ENTER.

```
[DOC]
        comment = Servidor samba
        path = /home/doc
        valid users = usuariosmb
        write list = usuariosmb
        create mask = 0644
        directory mask = 0775
root@server:/home/herick#
```

Paso 8

- ✓ Reiniciamos el servicio de samba de la siguiente manera.

 # /etc/init.d/samba restart

```
r Terminal Ir Ayuda
ome/herick# /etc/init.d/samba restart
g Samba daemons: nmbd smbd.
g Samba daemons: nmbd smbd.
ome/herick#
```

Hemos terminado con la configuración de samba y ahora procedemos a conectar los clientes

Paso 9.

- ✓ Instar smbsamba en la pc del cliente.

#aptitude install smbclient

Paso 10.

- ✓ En los equipos cliente que quieran tener acceso a este directorio compartido debe tener el mismo usuario y contraseña que hemos creado o conocerlas ya que al tratar de acceder le pedirá esos datos.

Paso 11.

- ✓ Colocar la contraseña y el usuario y aceptar, e inmediatamente podrá entrar al directorio que compartió.

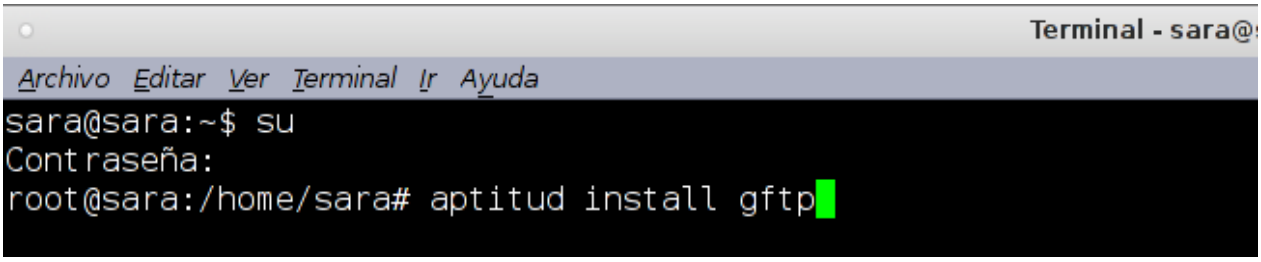
MANUAL PARA COMPARTIR ARCHIVOS EN RED CON FTP

Configuraciones en la PC del Servidor.

Paso 1

- ✓ Instalar FTP como usuario **root** digitamos en la terminal:

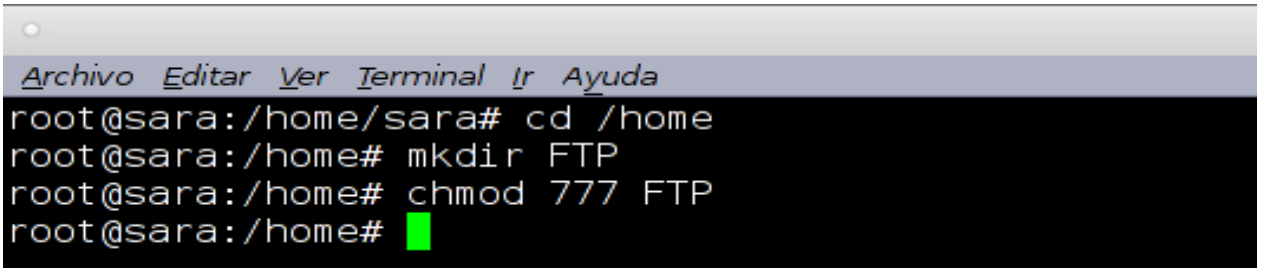
  aptitude install gftp



Paso 2

- ✓ Crear una carpeta y dar permisos.

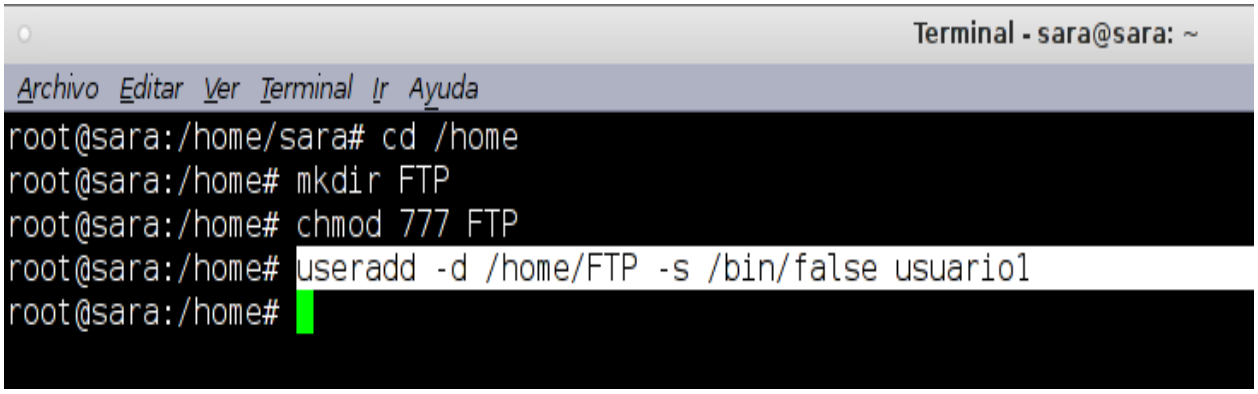
En este caso crearemos la carpeta en /home. Y para ello digitamos **cd /home** para entrar al home.



Paso 3

- ✓ Crearemos un usuario

  useradd -d /home/FTP -s /bin/false/usuario1

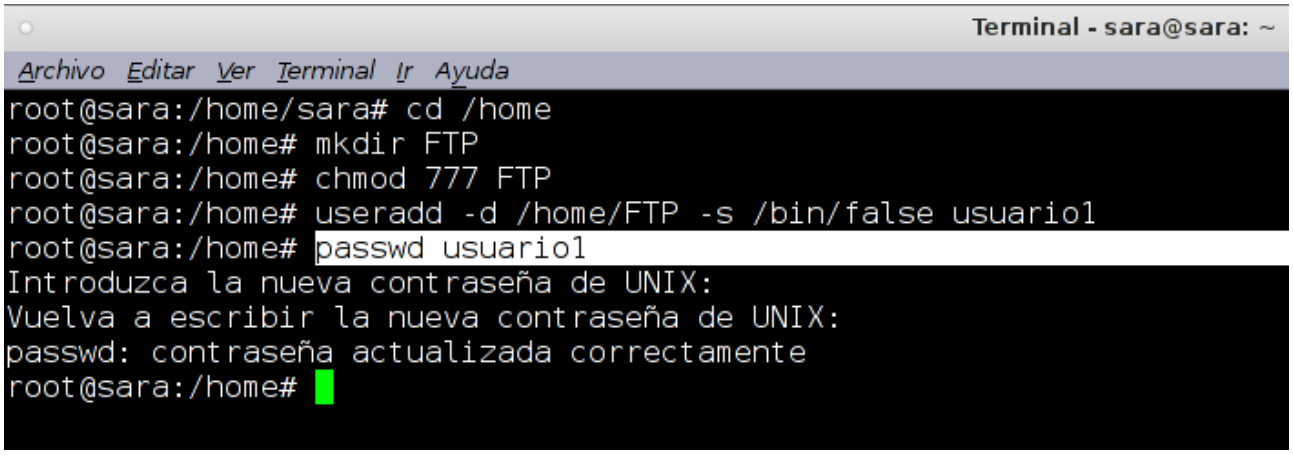


Paso 4

✓ Crear contraseña para el usuario1, digitamos el siguiente comando:

passwd usuario1

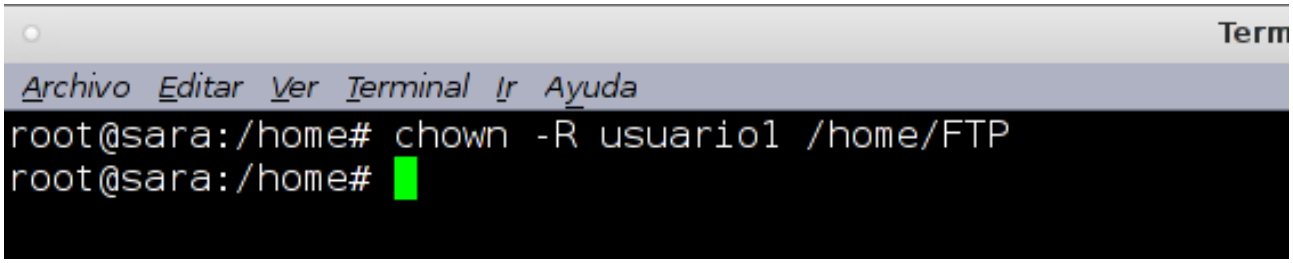
Introduzca la contraseña.



Paso 5

✓ Dar al usuario el control de la carpeta. Que en este caso es el "usuario1".

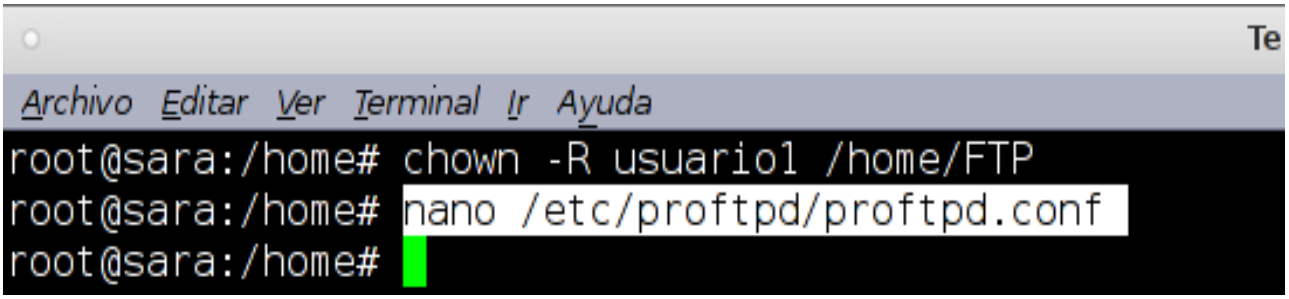
Chown -R usuario1 /home/FTP



Paso 6

✓ Configurar el documento del fichero **proftpd.conf,** digitando lo siguiente:

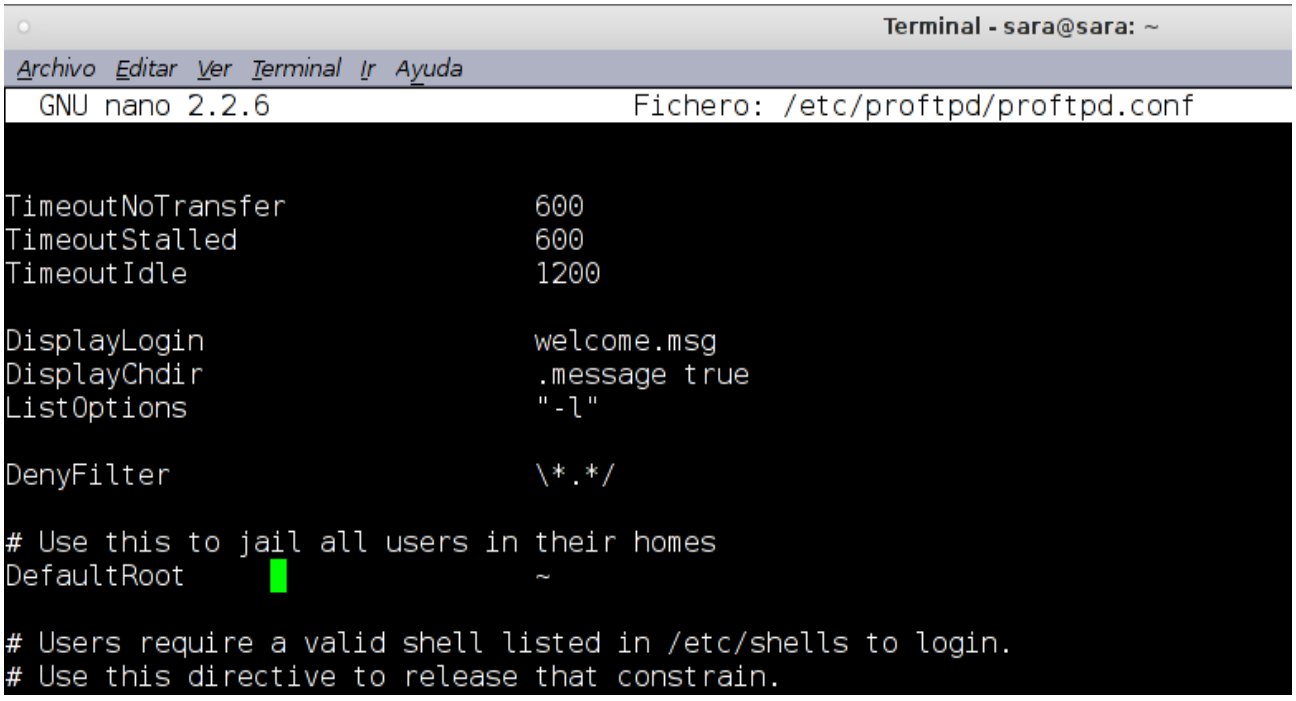
nano /etc/proftpd/proftpd.conf



Paso 7

✓ En el archivo **proftpd** realizamos los siguientes cambios.

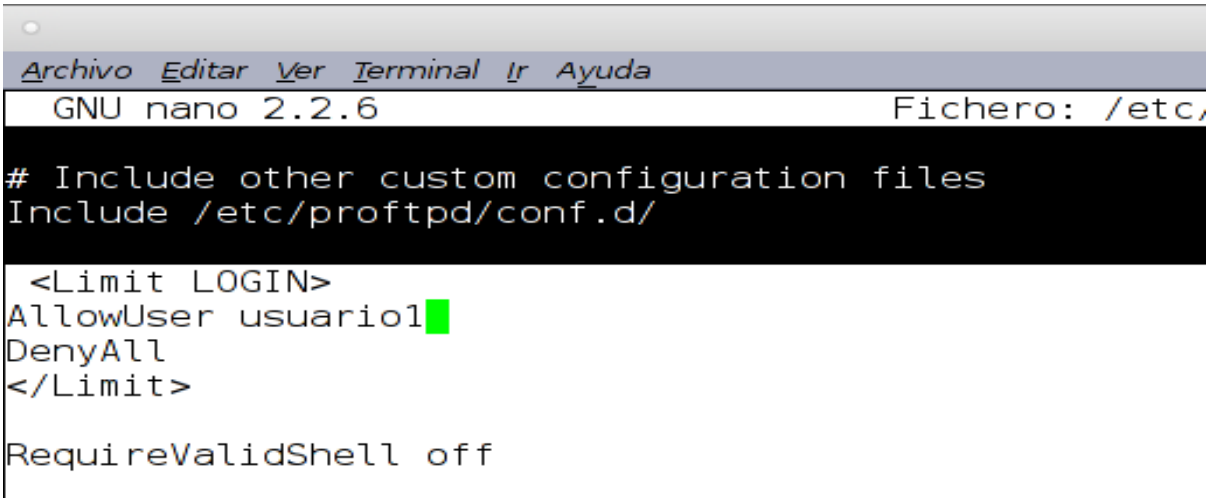
Des comentamos DefaultRoot ~



Paso 8

✓ Colocar al final de archivo lo siguiente:

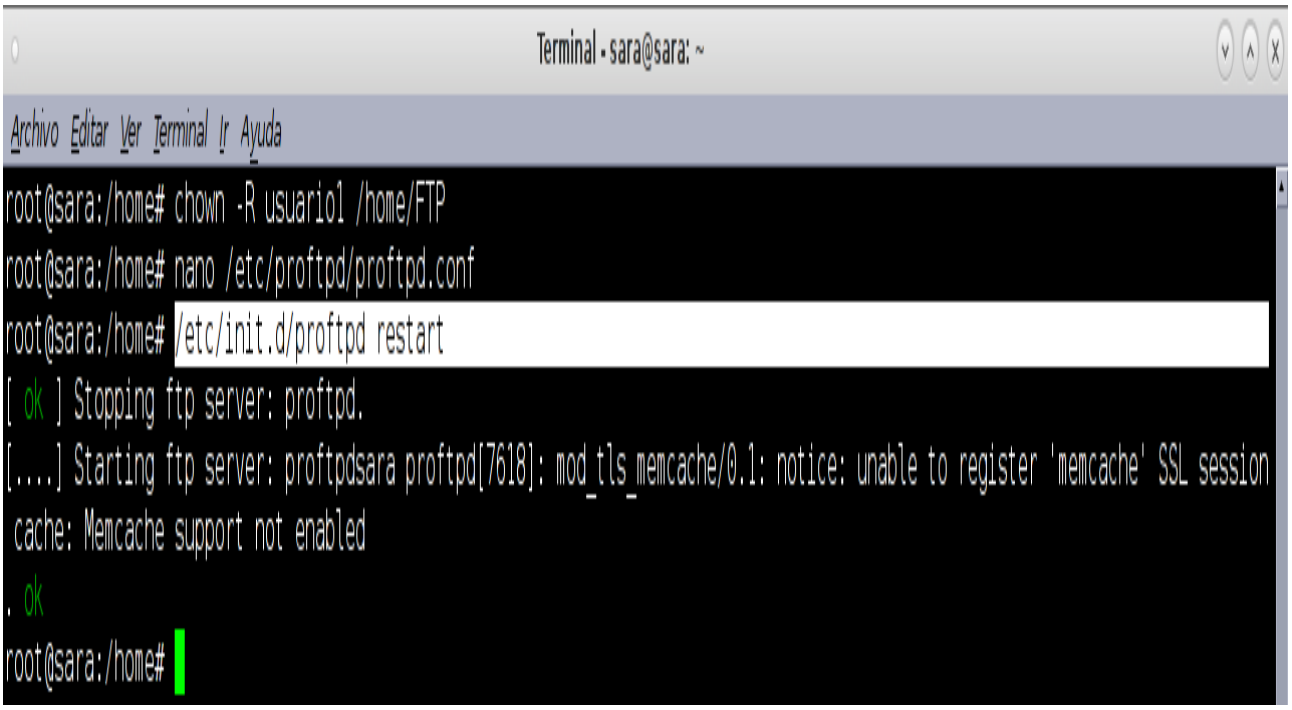
Guardamos la configuración con: ctrl+o, enter y ctrl+x



Paso 9

✓ Reiniciamos el **proftpd,** digitando lo siguiente:

/etc/init.d/proftpd restart

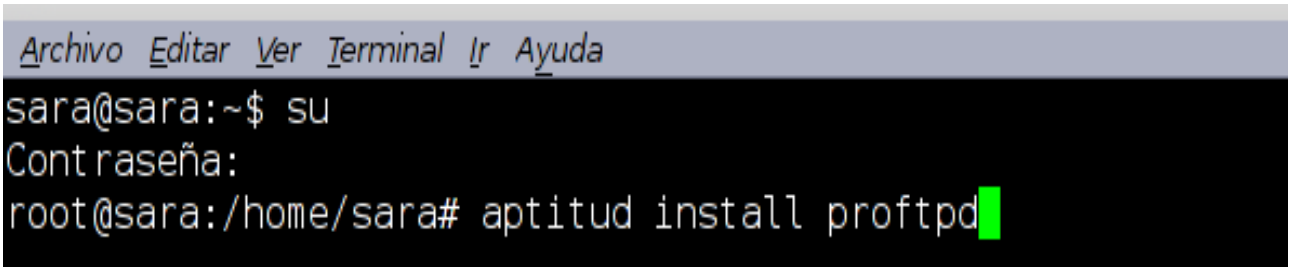


ENLA PC DEL CLIENTE HAREMOS LS SIGUIENTES CONFIGURACIONES

Para lograr compartir archivos en red con el protocolo ftp realizamos los siguientes pasos.

Paso 10

- ✓ En la PC del cliente instalamos gftp.
  Aptitude install gftp

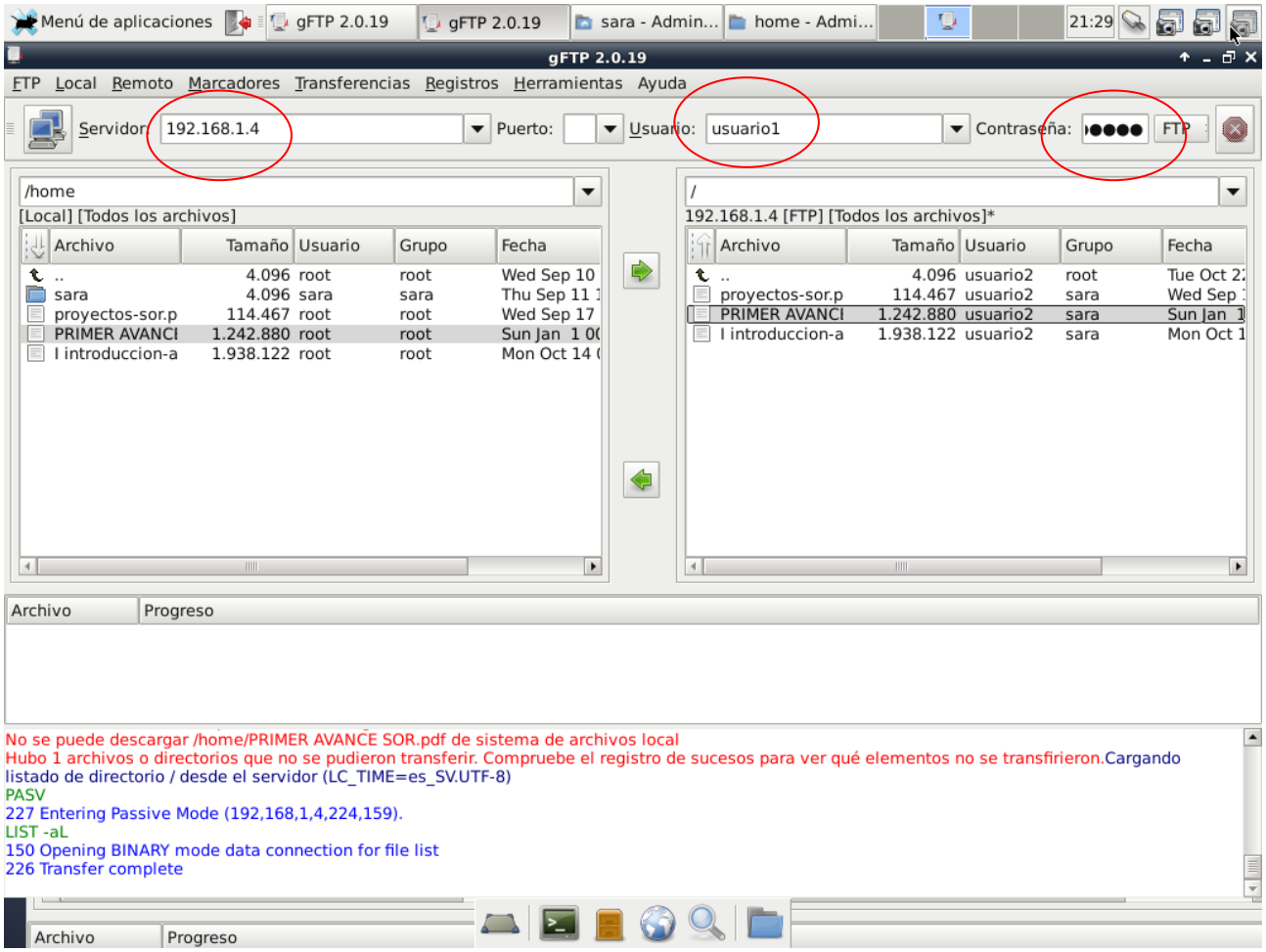


Paso 11

- ✓ Al tener instalado en la PC cliente el **gftp,** digitamos en la terminal "**gftp".**

Paso 12

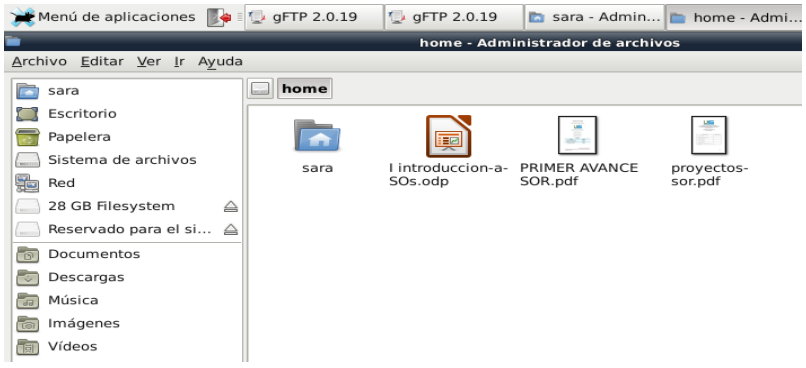
- ✓ Aparecerá una ventana donde se colocara la IP de la PC Servidor, el usuario que se creo y la contraseña.
  Mostrará los archivos que coloco en la carpeta FTP en el Servidor.



Paso 13

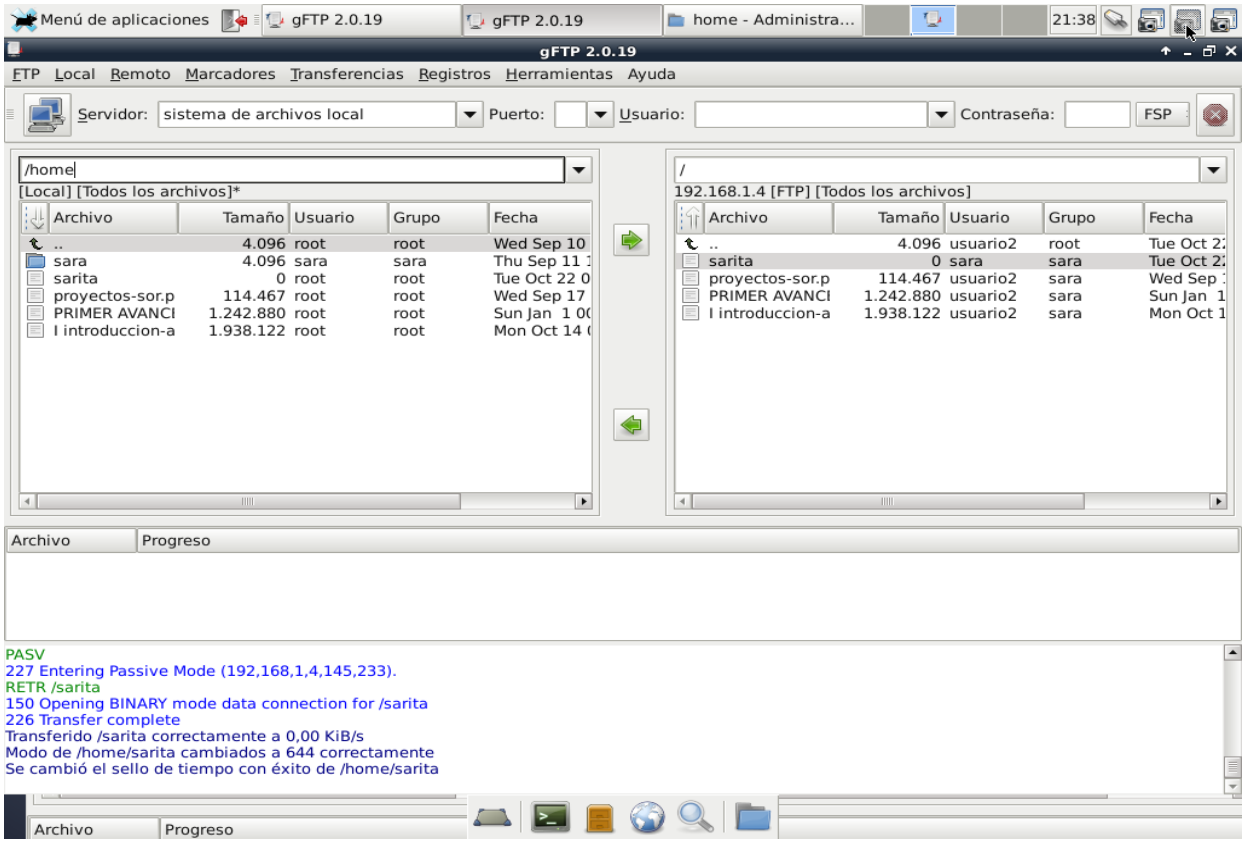
Los archivos colocados en la carpeta FTP existen tanto en el ordenador local como en el la PC Cliente.

Valla a la **carpeta /home** del la PC Cliente y podrá ver los archivos que contiene la carpeta NFS del PC Servidor.



Paso15

- ✓ Para compartir otro archivo solo debe colocarlo en la carpeta del servidor y actualizar en el la PC cliente dando clic en la imagen de la computadora en el lado superior izquierdo.



MANUAL PARA COMPARTIR ARCHIVOS EN RED CON EL PROTOCOLO NFS

Para compartir archivos en red seguiremos los siguientes pasos.

Paso 1.

- ✓ Instalamos el servidor NFS con

  #aptitude install nfs-kernel-server

```
root@server:/home/herick# aptitude install nfs-kernel-server
```

Paso 2.

- ✓ Creación del directorio a compartir y añadir a la lista de exportación.

Creamos el directorio en la ruta que deseemos y le otorgamos los permisos adecuados.

```
root@server:~# mkdir /home/nfs
root@server:~# chmod 777 /home/nfs
root@server:~# chown nobody:nogroup /home/nfs
```

Paso 3.

- ✓ Configuración del archivo export para agregar permisos.

Editamos el archivo /etc/exports y agregamos el directorio a la lista de exportacion y agregamos los permisos que consideremos más adecuados.

#nano /etc/export

```
 GNU nano 2.2.6            Fichero: /etc/exports                    Modificado

# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_c$
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#
/home/nfs *(rw,sync,no_root_squash,no_subtree_check)
```

Entre los permisos que podemos otorgar a un directorio se encuentran

-rw/ro= Exporta el directorio en modo lectura/escritura o sólo lectura.

- root_squash= Mapea los requerimientos del UID/GID 0 al usuario anónimo (por defecto usuario nobody con UID/GID 65534); es la opción por defecto.

- no_root_squash= No mapea root al usuario anónimo.

- all_squash= Mapea todos los usuarios al usuario anónimo.

- squash_uids/squash_gids= Especifica una lista de UIDs o GIDs que se deberían trasladar al usuario anónimo squash_uids= 0-15,20,25-50.

- anonuid/anongid= Fija el UID/GID del usuario anónimo (por defecto 65534)

- subtree_check/no_subtree_check= Si se exporta un subdirectorio (no un filesystem completo) el servidor comprueba que el fichero solicitado esté en el árbol de directorios exportado.

- sync modo síncrono= Requiere que todas las escrituras se completen antes de continuar; es opción por defecto.

- async modo asíncrono= No requiere que todas las escrituras se completen; más rápido, pero puede provocar pérdida de datos en una caída.

- secure= Los requerimientos deben provenir de un puerto por debajo de 1024

- insecure= Los requerimientos pueden provenir de cualquier puerto.

Paso 4.

- ✓ Reiniciamos el servidor usando.

  # /etc/init.d/nfs-kernel-server restart

```
root@server:~# /etc/init.d/nfs-kernel-server restart
[ ok ] Stopping NFS kernel daemon: mountd nfsd.
[ ok ] Unexporting directories for NFS kernel daemon....
[ ok ] Exporting directories for NFS kernel daemon....
[ ok ] Starting NFS kernel daemon: nfsd mountd.
root@server:~#
```

EN LA PC CLIENTE SEGUIREMOS LOS SIGUIENTES PASOS

Paso 5

- ✓ Instalacion del cliente nfs usamos el siguiente comando.

  #aptitude install nfs-common

Paso 6.

- ✓ Creamos el directorio donde montaremos la carpeta compartida y le damos los permisos para poder editarla.

```
root@server:/home# mkdir nfs
root@server:/home# chmod 777 nfs
root@server:/home#
```

Paso 7.

- ✓ Montamos el directorio compartido usando el siguiente comando

  #mount -t nfs -o rw,nosuid 192.168.1.4:/home/nfs /home/nfs

```
root@server:/home# mount -t nfs -o rw,nosuid 192.168.1.4:/home/nfs /home/nfs
root@server:/home#
```

Podemos verificar que directorios está compartiendo el servidor con el comando showmount -e ipservidor.

Paso 8.

- ✓ Montar automáticamente el directorio compartido.

Para montar automáticamente un directorio compartido con nfs editamos el archivo /etc/fstab agregando los datos correspondientes al recurso compartido.

```
/dev/sdd1       /media/usb0     auto    rw,user,noauto  0       0
192.168.1.4:/home/nfs /home/nfs nfs rw,nosuid 00
```

Así al reiniciar el cliente se montara automáticamente el directorio.

MANUAL PARA COMPARTIR ARCHIVOS EN RED CON SSH

Para compartir archivos en red con SSH haremos las siguientes configuraciones en la PC servidor.

Paso 1

- ✓ Instalar ssh server usando el comando.

  # aptitude install ssh

```
herick@server:~$ su
Contraseña:
root@server:/home/herick# aptitude install ssh
```

Paso 2

- ✓ Creación de los Usuarios.

Creamos los usuarios que podrán conectarse al servidor ssh usando el comando.

#useradd -d /"Ruta del directorio home" -s /bin/bash "nombre del usuario"

#passwd "usuario"

```
root@server:/home/herick# mkdir /home/ssh
root@server:/home/herick# mkdir /home/ssh/usuariossh
root@server:/home/herick# chmod 777 /home/ssh/usuariossh
root@server:/home/herick# useradd -d /home/ssh/usuariossh -s /bin/bash usuarioss
h
root@server:/home/herick# passwd usuariossh
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: contraseña actualizada correctamente
root@server:/home/herick#
```

Paso 3

- ✓ Configuraciones del archivo sshd_config.

Configuramos el archivo **sshd_config** que se encuentra en /etc/ssh haciendo las siguientes modificaciones:

#nano /etc/ssh/sshd_config

1. **PermitRootlogin NO "permite autenticarse como root"**

```
# Authentication:
LoginGraceTime 120
PermitRootLogin no
StrictModes yes
```

2. **X11Forwarding NO** "ejecucion de aplicaciones graficas".

```
X11Forwarding no
X11DisplayOffset 10
PrintMotd no
```

3. Si queremos delimitar los usuarios permitidos añadimos al final la línea AllowUser y agregamos los usuarios permitidos separados por espacio.

   AllowUsers usuariossh

```
#configuracion extra
AllowUsers usuariossh
```

4. Si queremos delimitar los grupos permitidos usamos AllowGoups.

Paso 4

- ✓ Al finalizar los las configuraciones reiniciamos el servidor ssh.

  #/etc/init.d/ssh restart

```
root@server:/home/herick# /etc/init.d/ssh restart
[ ok ] Restarting OpenBSD Secure Shell server: sshd.
root@server:/home/herick#
```

CONFIGURACIONES NECESARIAS PARA LA PC DEL CLIENTE

Para conectarse a un servidor ssh existen diversas formas ya sea usando un cliente sftp como filezila o montarla directamente en un directorio usando sshfs, que es la forma que utilizaremos.

Paso 5

- ✓ Instalación para la PC cliente usamos el comando:

  #aptitude install sshfs

```
root@server:/home/herick# apt-get install sshfs
```

Paso 6.

- ✓ Verificar el fichero modules.

Sshfs necesita fuse module para verificar si esta abrimos el fichero /etc/modules si no está simplemente lo agregamos al final para que el sistema lo cargue automáticamente al arrancar pero podemos utilizar el comando:

#modprobe fuse

```
herick@server:~$ su
Contraseña:
root@server:/home/herick# modprobe fuse
```

Paso 7.

- ✓ Verificar que el directorio pertenezca al grupo fuse.

El usuario que utilicemos para montar el directorio remoto debe pertenecer al grupo fuse.

```
root@server:/home/herick# usermod -G fuse -a herick
```

Paso 8.

- ✓ Crear los directorios que desea compartir.

Montamos el directorio remoto en una carpeta del sistema con los siguientes pasos:

1. creamos el directorio donde vamos a montar el directorio compartido.

```
root@server:/home# mkdir /home/ssh
```

2. Damos los permisos con chmod 777 al directorio ssh.

3. mapeamos la ruta del servidor remoto en este caso será:

   /home/ssh/cliente3ssh

4. para montarlo ejecutamos la instrucción siguiente con el usuario que agregamos al grupo fuse.

   $sshfs cliente3ssh@192.168.1.4:/home/ssh/cliente3ssh /home/herick/ssh

```
herick@server:~$ sshfs cliente3ssh@192.168.1.4:/home/ssh/cliente3ssh /home/heric
k/ssh
cliente3ssh@192.168.1.4's password:
herick@server:~$
```

5. Nos pedirá el password del usuario y al ponerlo montara el directorio remoto en la carpeta que hemos indicado.

Paso 9.

- ✓ Para desmontar el directorio compartido usamos el comando

  $fusermount -u /home/herick/ssh