



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			
N°	APELLIDOS	NOMBRES	CARNET
1	ACOSTA ARIAS	JOSUÉ ISAAC	AA01132424
2	BONILLA HENRÍQUEZ	ALEXI WALDIR	BH01132333
3	VEGA YANES	MIGUEL ALEJANDRO	VY01132301
4	VILLAFRANCO	NAHÚM DE JESÚS	V01121164

Indice

INTRODUCCIÓN.....	4
OBJETIVOS.....	5
GENERAL:	5
ESPECÍFICOS:.....	5
MARCO TEÓRICO	6
NFS (NETWORK FILE SYSTEM).....	6
CIFS O SAMBA	7
FTP ('PROCOLO DE TRANSFERENCIA DE ARCHIVOS').....	8
SAN Y NAS	9
NAS (NETWORK ATTACHED STORAGE).....	10
SSH (SECUREHELL: INTÉRPRETE DE ÓRDENES SEGURA)	14
DIFERENCIA SAN Y NAS.....	14
GUÍA DE INSTALACIÓN DE LOS DIFERENTES PROTOCOLOS.....	17
DESCRIPCIÓN DEL PROYECTO.	21
DIAGRAMA DE RED.	22
DIAGRAMA DE GANTT	22
VIABILIDAD Y FACTIBILIDAD.	23
CONCLUSIÓN	24
BIBLIOGRAFÍA	25
ANEXOS	26

INDICE DE TABLAS

VIABILIDAD Y FACTIBILIDAD. 23
DIFERENCIA SAN Y NAS 10

INDICE DE IMÁGENES

DIAGRAMA DE RED. 22
DIAGRAMA DE GANTT 22

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 un entorno informático es imprescindible disponer de un servicio que permita el acceso seguro a archivos remotos de forma transparente. 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 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 autenticar 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.

OBJETIVOS.

GENERAL:

-Mostrar 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 de forma local.

ESPECÍFICOS:

-Exponer los diferentes protocolos que nos permiten compartir archivos en red.

-Demostrar el funcionamiento de cómo trabajan los protocolos SAMBA, FTP, NFS.

-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:

- ✓ NFS
- ✓ CIFS OSAMBA
- ✓ FTP
- ✓ SSH

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, 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.

CIFS o 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.

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 deservidor 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 arboles de directorios e impresoras a clientes Linux, UNIX y Windows
- Asistente en la navegación de la red (con o sin NetBIOS)
- Autentifica las conexiones a dominios Windows
- Proporciona resolución de nombres de Windows Internet NameService (WINS)
- Actúa como un Controlador de Dominio Primario (PrimaryDomainController, PDC)

Estilo Windows

- Actúa como un BackupDomainController (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:

- Actuar como un BackupDomainController (BDC) para Windows PDC (y vice versa)
- Actuar 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.

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): Es una red concebida para conectar servidores, matrices de discos y librerías de soporte. Principalmente, está basada en tecnología fibrechannel 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 fibrechannel).

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 FibberChannel 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.

Diferencia entre ISCI Y Fiberchannel

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.

¿Qué es Raid?

Es un sistema de almacenamiento de datos en tiempo real que utiliza múltiples unidades de almacenamiento de datos (Discos duros o SSD) entre los que se distribuyen o replican los datos.

Niveles de RAID

RAID nivel 0:

También conocido como "Striping" o "Fraccionado". Los datos son divididos en pequeños segmentos y distribuidos entre los discos. No ofrece tolerancia a fallos, ya que no existe redundancia, esto quiere decir que un fallo en cualquiera de los discos rígidos puede ocasionar pérdida de información.

RAID nivel 1:

También conocido como "Mirroring" o "Espejado", funciona añadiendo discos rígidos paralelos a los discos rígidos principales existentes en la computadora. Los discos que fueron añadidos, trabajan como una copia del primero, así si el disco principal recibe datos, el disco anexado también los recibe, pues un disco rígido pasa a ser una copia prácticamente idéntica del otro. De esa forma, si uno de los discos rígidos presenta una falla, el otro inmediatamente puede asumir la operación y continuar el acceso a la información. La consecuencia es que la grabación de datos es más lenta, pues es realizada dos veces. Sin embargo, la lectura de esa información es más rápida, pues puede ser accedida de dos fuentes. Una aplicación muy común del RAID 1 es su uso en servidores de archivos.

RAID nivel 2:

Este tipo de RAID, adapta el mecanismo de detección de fallas en discos rígidos para funcionar en memoria. Así, todos los discos de la matriz están siendo "monitorizados" por el mecanismo. Actualmente, el RAID 2 es poco usado, ya que prácticamente todos los discos rígidos nuevos salen de fábrica con mecanismos de detección de fallas implantados.

RAID nivel 3:

En este nivel, los datos son divididos entre los discos de la matriz, excepto uno, que almacena información de paridad. Así, todos los bytes de los datos tienen su

paridad (aumento de 1 bit, que permite identificar errores) almacenada en un disco específico. A través de la verificación de esta información, es posible asegurar la integridad de los datos, en casos de recuperación por eso y por permitir el uso de datos divididos entre varios discos, el RAID 3 logra ofrecer altas tasas de transferencia y confianza en la información. Para usar el RAID 3, por lo menos 3 discos son necesarios

RAID nivel 4:

Este tipo de RAID, divide los datos entre los discos, siendo uno de esos discos exclusivo para paridad. La diferencia entre el nivel 4 y el nivel 3, es que en caso de falla de uno de los discos, los datos pueden ser reconstruidos en tiempo real a través de la utilización de la paridad calculada a partir de los otros discos, siendo que cada uno puede ser accedido de forma independiente.

El RAID 4 es el indicado para el almacenamiento de archivos grandes, donde es necesario asegurar la integridad de la información. Porque, en este nivel, cada operación de grabación requiere un nuevo cálculo de paridad, dando mayor confianza al almacenamiento (esta operación torna las grabaciones de datos más lentas).

RAID nivel 5:

Es muy semejante al nivel 4, excepto por que la paridad no está destinada a un único disco, sino a toda la matriz. Eso hace que la grabación de datos sea más rápida, pues no es necesario acceder a un disco de paridad en cada grabación. A pesar de eso, como la paridad es distribuida entre los discos, el nivel 5 tiene un poco menos de performance que el RAID 4. El RAID 5 es el nivel más utilizado y que ofrece resultados satisfactorios en aplicaciones no muy pesadas. Este nivel necesita de por lo menos 3 discos para funcionar.

RAID 0 + 1:

Es una combinación de los niveles 0 (Striping) y 1 (Mirroring), donde los datos son divididos entre los discos para mejorar el ingreso, pero también utilizan otros discos para duplicar la información. Así, es posible utilizar el buen ingreso del nivel 0 con la redundancia del nivel 1. Sin embargo, es necesario por lo menos 4 discos para montar un RAID de este tipo. Estas características hacen del RAID 0 + 1 el más rápido y seguro, sin embargo es el más caro de ser implementado.

Tipos de RAID:

Existen 2 tipos de RAID:

1- basado en hardware

2-basado en software.

Cada uno posee ventajas y desventajas.

Raid basado en hardware: es el más utilizado, pues no depende de un sistema operativo (estos ven al RAID como un único disco grande) y son bastante rápidos, lo que posibilita explorar íntegramente sus recursos. Su principal desventaja es ser caro.

El RAID basado en hardware, utiliza dispositivos denominados "controladores RAID", que pueden ser conectados en slots PCI de la placa madre de la computadora.

RAID basado en software: no es muy utilizado, pues a pesar de ser menos costoso, es más lento, posee más dificultades de configuración y depende del sistema operativo para tener una performance satisfactoria. Este tipo es dependiente del poder de procesamiento de la computadora en que es utilizado.

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	Fibrechannel dedicado
Clientes principales	Usuarios Finales	Servidores de aplicaciones
Acceso a disco	A través del dispositivo NAS	Acceso directo

SSH (SecureShell: intérprete de órdenes segura)

SSH (o SecureShell) 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 TatuYlönen, 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.

Diferencia entre SFTP vs. FTPS

SFTP y FTPS son dos protocolos completamente distintos.

SFTP asegura las transmisiones con SSH, mientras que FTPS usa seguridad SSL

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 extremadamente 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 Criptográfico con Clave Pública) es un algoritmo que genera un par de llaves de autenticación, la pública y la privada. La pública se distribuye en forma autenticada y la privada que generalmente es guardada en secreto por el propietario. El protocolo SSH (Secure Shell) está 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 del lado del servidor, el firewall debe aceptar conexión es entrantes al puerto configurado para SSH.

GUÍA DE INSTALACIÓN DE LOS DIFERENTES PROTOCOLOS

SMB/CIFS

Comandos utilizados para el protocolo samba

```
# apt-get install samba
```

Se utiliza para la instalación de Samba

```
#adduser -system -no-create-home -uid 603 lacrimosa
```

Procedemos a la creación de los usuarios, asignándoles nombre y número de identificación por usuario.

```
#cat /etc/passwd | grep lacrimosa
```

Comprobamos que los usuarios se añadieron de forma correcta.

```
#smbpasswd -a lacrimosa
```

Con este comando le asignaremos la contraseña a los usuarios que hemos creados.

```
# nano /etc/samba/smb.conf
```

Editor para configuración de los archivo de Samba.

```
#mkdir
```

Crea los directorios que se deseamos compartir.

```
#chmod 777
```

Otorga permisos root a los diferentes directorios a crear.

```
#testparm
```

Muestra una información de acuerdo a la configuración que se ha ingresado.

```
# /etc/init.d/samba restart
```

Reinicia el servicio de los programas después de una configuración.

```
#cd /
```

Se utiliza para ingresar a carpetas o directorios.

FTP

Comandos utilizados para el protocolo ftp

```
#apt-getinstallgftp
```

Le daremos al usuario el control de la carpeta. Que en este caso es el “lacrishare”

```
#nano /etc/proftpd/proftpd.conf
```

Configurar el documento del fichero proftpd.conf.

```
#/etc/init.d/proftpdrestart
```

Reinicia el proftpd.

NFS

Comandos utilizados para el protocolo nfs.

```
#aptitudeinstallnfs-kernel-server
```

Instalamos el servidor NFS.

```
mkdir
```

Crea los directorios que se desean compartir.

```
#chmod 777
```

Otorga permisos a los directorios creados.

```
#nano /etc/export
```

Configura el archivo export para lo necesario.

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

Reiniciamos el servidor nfs.

```
#-t nfs 192.168.60.101:/home/nahum/Publik /home/jorge/Escritorio/Share Nos  
permite montar el directorio compartido .
```

```
#showmount -e ipservidor
```

Nos muestra la carpeta que estamos compartiendo en NFS

SSH

Comandos utilizados para el protocolo ssh.

```
# apt-get install ssh
```

Instalamos el servidor ssh.

```
#useradd -d /"Ruta del directorio home" -s /bin/bash "nombre del usuario"=  
Creamos los usuarios que se podrán conectar al servidor.
```

#nano /etc/ssh/sshd_config= Configura el archivo sshd_config.

#aptitudeinstallsshfs= Instala el paquete sshfs para el cliente.

#modprobe fuse= Comando para que el sistema cargue automáticamente sshfs al arrancar.

ssh -p 22 nahum@192.168.60.101

Nos pedirá contraseña de root del servidor y podremos estar dentro de la terminal, como si fuera la nuestra.

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/ETHERNET.

DESCRIPCIÓN DEL PROYECTO.

El proyecto a realizar 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 proyecto debe permitir el 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 instalado es el Sistema Operativo Debian 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 también se hemos hecho configuraciones dentro de cada protocolo para realizar respectivas configuraciones del Servidor y cliente de esa manera compartir archivos y que usuarios podrán acceder a los archivos compartidos y que tipo de permiso tendrán.

DIAGRAMA DE RED.

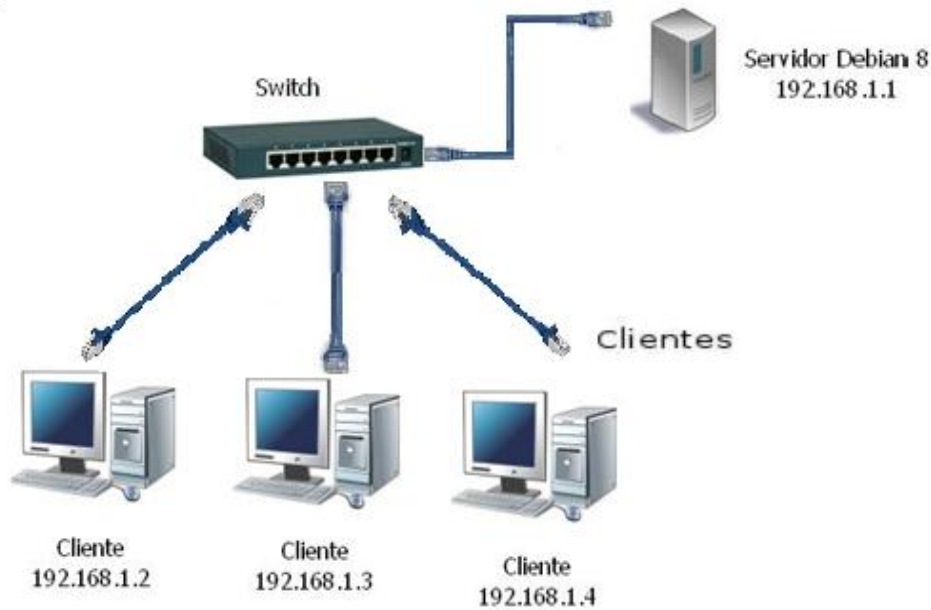


DIAGRAMA DE GANTT

Nombre de la tarea	Agosto 2015					Septiembre 2015				Octubre 2015				Noviembre 2015		
	Ago 1	Ago 8	Ago 15	Ago 22	Ago 29	Sep 5	Sep 12	Sep 19	Sep 26	Oct 3	Oct 10	Oct 17	Oct 24	Oct 31	Nov 7	Nov 14
1 - Formación de grupo de trabajo		█	█													
2 -Análisis de la opciones de proyecto		█	█													
3 -Eleccion del proyecto		█	█													
4 -Investigación de tecnología a utilizar para nuestro proyecto		█	█													
5 -Desarrollo y Entrega del Perfil del proyecto		█	█	█	█											
6 - Recopilacion de tecnologías a implementar				█	█	█	█									
7 - Investigacion sobre NFS				█	█											
8 - Investigacion de SAMBA					█	█										
9 Realizacion de ensayos con NFS y SAMBA						█	█									
10 Realizacion de prototipo y Entrega del primer Avance							█	█								
11 Investigacion FTP								█	█							
12 Pruebas FTP									█	█						
13 Elaboracion del prototipo y segundo Avance										█	█					
14 Pruebas de Correccion del Proyecto											█	█				
15 Elaboracion del Tercer Avance												█	█			
16 Realizacion de pruebas finales													█	█		
17 Elaboracion de reporte final														█	█	
18 Entrega de proyecto final y Defensa															█	█

VIABILIDAD Y FACTIBILIDAD.

Detalle	Cantidad	Costo
PC como servidor	1	\$ 200.00
Clientes	3	\$ 600.00
Switch	1	\$ 30.00
S.O GNU/Linux (debían)	1	\$ 0.00
Mano de Obra	3	\$ 100.00
	Total	\$915.00

Factibilidad Operativa:

El operacional del proyecto es la ejecución y control del proyecto

Factibilidad Técnica:

La implementación de un sistema de archivos en red es una opción adecuada en un ambiente que se requiere compartir archivos de una manera rápida y constante además que permitiría establecer permisos sobre los archivos compartidos.

CONCLUSIÓN

Como grupo de trabajo concluimos que el sistema de archivos es importante para internet porque por medio de eso se puede transferir archivos sin necesidad de copiar y pegar en una USB u otro dispositivo, es mejor porque por medio de una IP podemos hacerlo aun sin necesidad de internet si es forma local.

De manera que Compartir archivos en red 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 para ellos poder tener una conexión de archivo entre ellos sin necesidad de copiar y pegar.

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

BIBLIOGRAFÍA

Instalación y configuración de NFS

http://www.ite.educacion.es/formacion/materiales/85/cd/linux/m4/instalacion_y_configuracion_de_nfs.html

Instalación y configuración de Samba.

<http://www.driverlandia.com/instalar-samba-para-compartir-carpeta-en-linux-debian/>

<http://www.driverlandia.com/instalar-samba-para-compartir-carpeta-en-linux-debian/>

Instalación y Configuración de SSH(OpenSSH)en GNU/Linux(modos consola)-
wikideelhacker.net.

(n.d.).RetrievedNovember7,2014,from<http://wiki.elhacker.net/redes/administracion-de-redes-gnu-linux/instalacion-y-configuracion-de-ssh-openssh-en-gnu-linux-modos-consola>

Instalar servidor FTP y restringir permisos a los usuarios.(n.d.).

RetrievedNovember7,2014,from<http://rootear.com/ubuntu-linux/instalar-servidor-ftp>

ANEXOS

GUÍA DE CÓMO IMPLEMENTAR LOS DIFERENTES PROTOCOLOS.

MANUAL DE IMPLEMENTACIÓN DE SAMBA

Paso 1

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

```
root@debian:/home/nahum# apt-get install samba
```

Paso2

Procedemos a crear los usuarios, asignando nombre y número de identificación por usuarios de la siguiente manera.

```
#adduser-system-no-create-home-uid603 lacrimosa
```

```
root@debian:/home/nahum# adduser -system -no-create-home -uid 603 lacrimosa
```

Paso 3

Añadimos los usuarios a samba de la siguiente manera.

```
#smbpasswd-a lacrimosa
```

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@debian:/home/nahum# smbpasswd -a lacrimosa
New SMB password:
Retype new SMB password:
Added user lacrimosa.
root@debian:/home/nahum#
```

Paso 4

Procedemos a la configuración del archivo de samba.

```
# nano /etc/samba/smb.conf
```

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

```
##### Global Settings #####  
  
[global]  
  
## Browsing/Identification ##  
  
# Change this to the workgroup/NT-domain name your Samba server will part of  
workgroup = FileSystem  
  
##### Authentication #####
```

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

(Procedemos a configurar el archivo con el recurso compartido).

```
##### Authentication #####  
  
# Server role. Defines in which mode Samba will operate. Possible  
# values are "standalone server", "member server", "classic primary  
# domain controller", "classic backup domain controller", "active  
# directory domain controller".  
security= user  
  
# Most people will want "standalone sever" or "member server".  
# Running as "active directory domain controller" will require first  
# running "samba-tool domain provision" to wipe databases and create a  
# new domain.
```

Nos ubicamos en las últimas líneas para poder agregar nuestra configuración

Esta es la configuración que ingresaremos en nuestro archivo de samba

```
path = /home/nahum/Publik-smb
read only = no
security = user
null password = yes
case sensitive = no
comment = Carpeta Compartida
guest ok = yes
browseable = yes
umask = 000
```

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

Paso 5

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

```
#mkdir /home/nahum/Publik-smb
```

```
#chmod777 /home/nahum/Publik-smb
```

```
root@debian:/home/nahum# mkdir /home/nahum/Publik-smb
root@debian:/home/nahum# chmod 777 /home/nahum/Publik-smb/
root@debian:/home/nahum# █
```

Paso 6

Reiniciamos el servicio de samba de la siguiente manera.

```
#!/etc/init.d/smbarestart
```

```
root@debian:/home/nahum# /etc/init.d/samba restart
[ ok ] Restarting nmbd (via systemctl): nmbd.service.
[ ok ] Restarting smbd (via systemctl): smbd.service.
[ ok ] Restarting samba-ad-dc (via systemctl): samba-ad-dc.service.
root@debian:/home/nahum# █
```

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

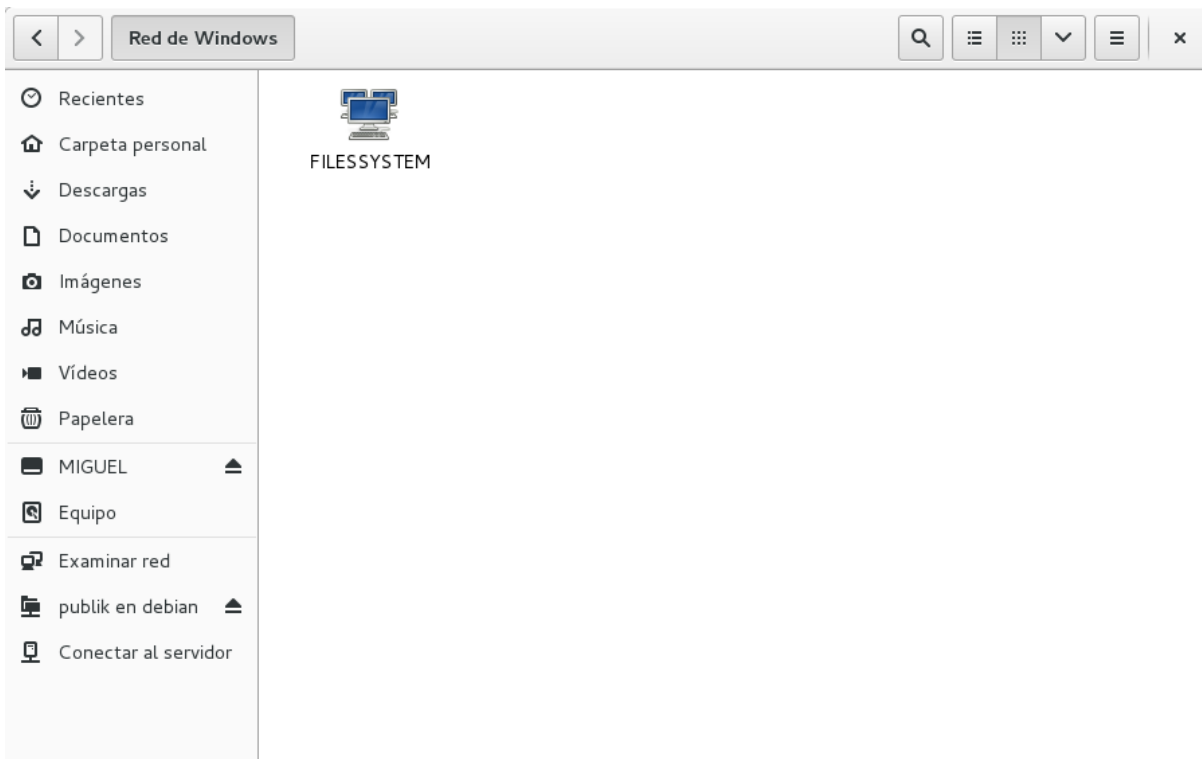
Paso 7.

Instalar smb samba en la pc del cliente.

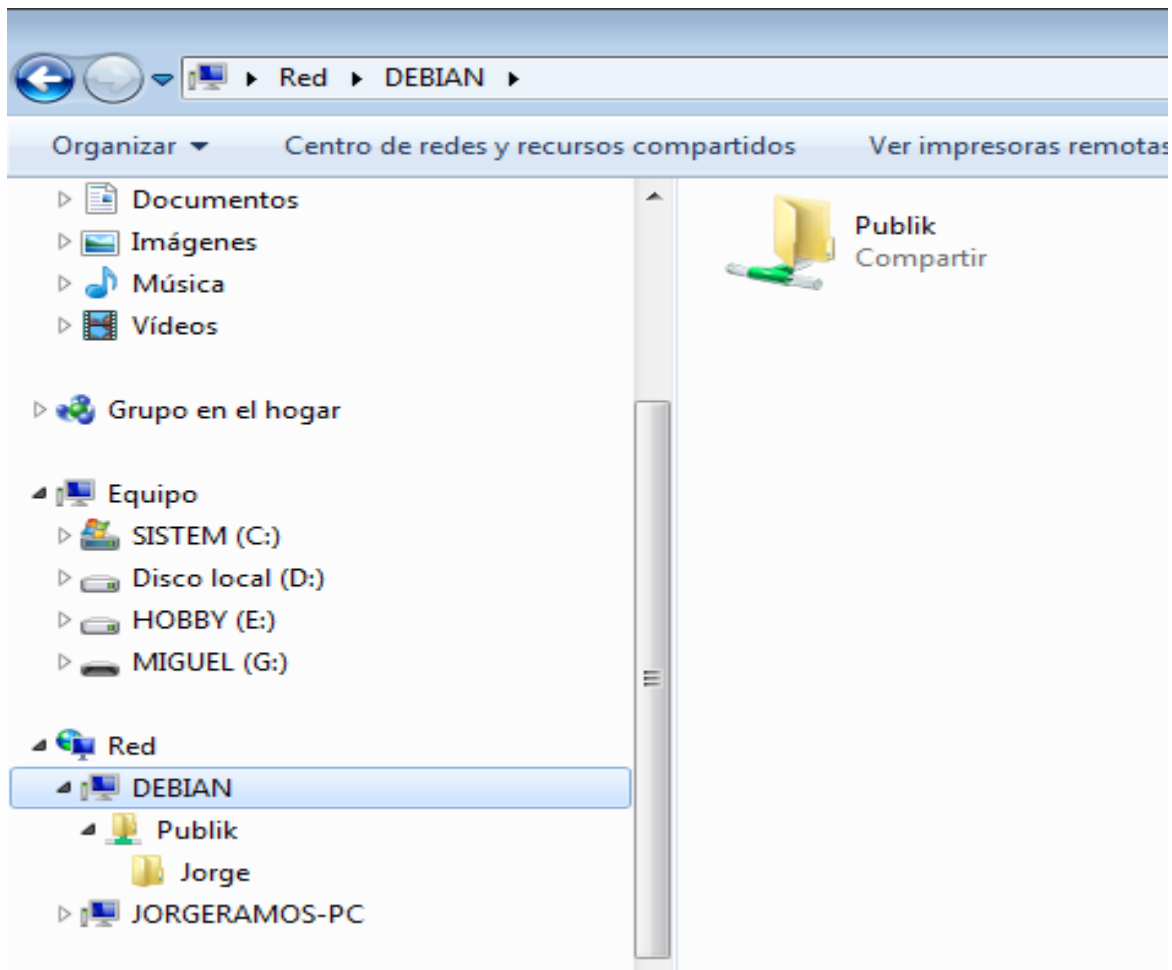
```
#aptitude install smbclient
```

Paso 8.

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.



También funciona con Windows intercambio de archivos Windows-Linux



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:

```
apt-get install gftp
```

```
root@debian:/home/nahum# apt-get install gftp
```

Paso 2

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

```
nano /etc/proftpd/proftpd.conf
```

Paso 3

En el archivo proftpd realizamos los siguientes cambios.

Des comentamos DefaultRoot ~

```
# Use this to jail all users in their homes
DefaultRoot ~

# Users require a valid shell listed in /etc/shells to login.
# Use this directive to release that constrain.
# RequireValidShell off
```

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

Paso 4

Reiniciamos el proftpd, digitando lo siguiente:

```
/etc/init.d/proftpd restart
```

Configuración de clientes FTP

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

Paso 5

En la PC del cliente instalamos gftp.

Apt-get install gftp

Paso 6

Abrimos un navegador o una carpeta y podremos establecer conexión

1- Poniendo la siguiente dirección en el navegador

```
ftp://nahum:villafranco@192.168.60.101
```


2- Abriendo una ventana y conectar con el navegador y podremos ver la carpeta de archivos del Servidor.

Conectarse al servidor

Detalles del servidor

Servidor: Puerto: - +

Tipo: ▼

Carpeta:

Detalles de usuario

Nombre de usuario:

Contraseña:

Recordar esta contraseña

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@debian:/home/nahum# apt-get 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.

```
#mkdir /home/nahum/Publik
```

```
#chmod 777 /home/nahaum/Publik
```

```
nahum@debian:~$ su
Contraseña:
root@debian:/home/nahum# mkdir Publik
root@debian:/home/nahum# chmod 777 Publik
root@debian:/home/nahum# █
```

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
# /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_sub$
#
# 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/nahum/Publik 192.168.60.0/24(rw,sync,no_root_squash)
```

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.
- 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@debian:/home/nahum# /etc/init.d/nfs-kernel-server restart
[ ok ] Restarting nfs-kernel-server (via systemctl): nfs-kernel-server.service.
root@debian:/home/nahum# █
```

EN LA PC CLIENTE SEGUIREMOS LOS SIGUIENTES PASOS

Paso 5

Instalacion del cliente nfs usamos el siguiente comando.

```
#apt-get install nfs-common y apt-get install portmap
```

```
root@debian:/home/jorge# apt-get install nfs-common && apt-get install portmap
```

Esto se instala los clientes que deseamos compartir, en este caso estamos usando el servidor como ejemplo pero solo es para mostrar.

Paso 6.

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

```
mkdir /home/jorge/Escritorio/Share
```

```
chmod 777 /home/jorge/Escritorio/Share
```

Paso 7.

Montamos el directorio compartido usando el siguiente comando

```
#mount -t nfs 192.168.60.101:/home/nahum/Publik /home/jorge/Escritorio/Share
```

```
root@debian:/home/jorge# mount -t nfs 192.168.60.101:/home/nahum/Publik /home/jorge/Escritorio/Share
```

Podemos verificar que directorio se está compartiendo del servidor ejecutamos 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.

`etc/fstab`

```
GNU nano 2.2.6          Fichero: /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>  <type> <options>          <dump> <pass>
# / was on /dev/sda7 during installation
UUID=afa4d5fe-b208-463c-b535-c79604d624c8 /          ext4      errors=remount-ro 0      1
# swap was on /dev/sda8 during installation
UUID=4d274e55-ff2e-4643-a519-9799f78b4fbe none        swap      sw          00
192.168.60.101:/home/nahum/Publik /home/jorge/Escritorio/Share nfs rw 0 0
```

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.

```
# apt-get install ssh
```

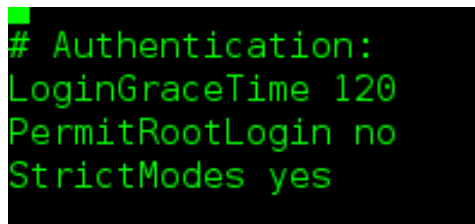
Paso 2

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
```

Paso 3

Al finalizar las configuraciones reiniciamos el servidor ssh.

```
#!/etc/init.d/ssh restart
```

CONFIGURACIONES NECESARIAS PARA LA PC DEL CLIENTE

Paso 4

Instalación para la PC cliente usamos el comando:

```
#apt-get install sshfs
```

Paso 5

Nos conectamos vía túnel por medio de terminal al servidor o también podemos hacerlo por interfaz, para hacerlo por vía túnel debemos escribir el siguiente comando en la terminal del cliente.

```
ssh -p 22 nahum@192.168.60.101
```

No pedirá contraseña de root del servidor y podremos estar dentro de la terminal, como si fuera la nuestra.

Modo interfaz, para poder ingresar al servidor de forma grafica

Conectarse al servidor

Detalles del servidor

Servidor: 192.168.60.101 Puerto: 22 - +

Tipo: SSH ▼

Carpeta: /

Detalles de usuario

Nombre de usuario: nahum

Contraseña: ●●●●●●●●●●

Recordar esta contraseña

Ayuda Cancelar Conectar

