



Universidad Nacional Autónoma de México

Facultad de Ingeniería

Semestre 2024-2

Redes de Datos Seguras

Grupo 6

Proyecto 3

Prof.: Ing. Edgar Martínez Meza

Brigada 3

- **Barrera Peña Víctor Miguel** - 315346219
- **Ramírez González José Miguel** - 420053583
- **Sánchez Manzano Mariana** - 411026622

Actividad

El proyecto deberá contar con las especificaciones dadas el primer día de clases en el documento impreso, aunado a lo que se especifica a continuación:

- Los equipos serán de dos o tres personas máximo.
- El proyecto contempla la creación de un servidor de autenticación y de archivos que permitirá conectarse desde un cliente con sistema operativo Windows o Linux del cual distribución, pero permitirá conectarse a ambos, aunque no de manera simultánea.
- El equipo servidor para la autenticación del cliente Linux deberá tener activos los servicios de NFS y NIS.
- El equipo servidor para la autenticación del cliente Windows deberá tener activo el servicio de SAMBA.
- El equipo cliente Linux deberá poder autenticarse desde el inicio del sistema operativo y automáticamente importar todos los archivos en la cuenta de usuario creada en el servidor.
- El equipo cliente Windows deberá poder autenticarse desde el inicio del sistema operativo y automáticamente importar todos los archivos en la cuenta de usuario creada en el servidor.
- El proyecto que se llevará a cabo es una simulación del laboratorio de computación salas "A" y "B" (Computación para Ingenieros). El equipo deberá entregar un informe por escrito que contenga todas las instalaciones que se llevaron a cabo, así como las modificaciones de configuración que se realizaron en cada uno de los archivos. La distribución de Linux que utilicen es independiente.

Introducción

1. ¿Qué es Samba?

Samba es un proyecto de software libre que implementa el protocolo de archivos compartidos de Windows para sistemas operativos de tipo UNIX. Básicamente, Samba permite que computadoras con GNU/Linux, Mac OS X o Unix se relacionen como servidores o clientes en redes basadas en

Windows. Aquí tienes algunos conceptos clave y pasos para configurar un servidor Samba:

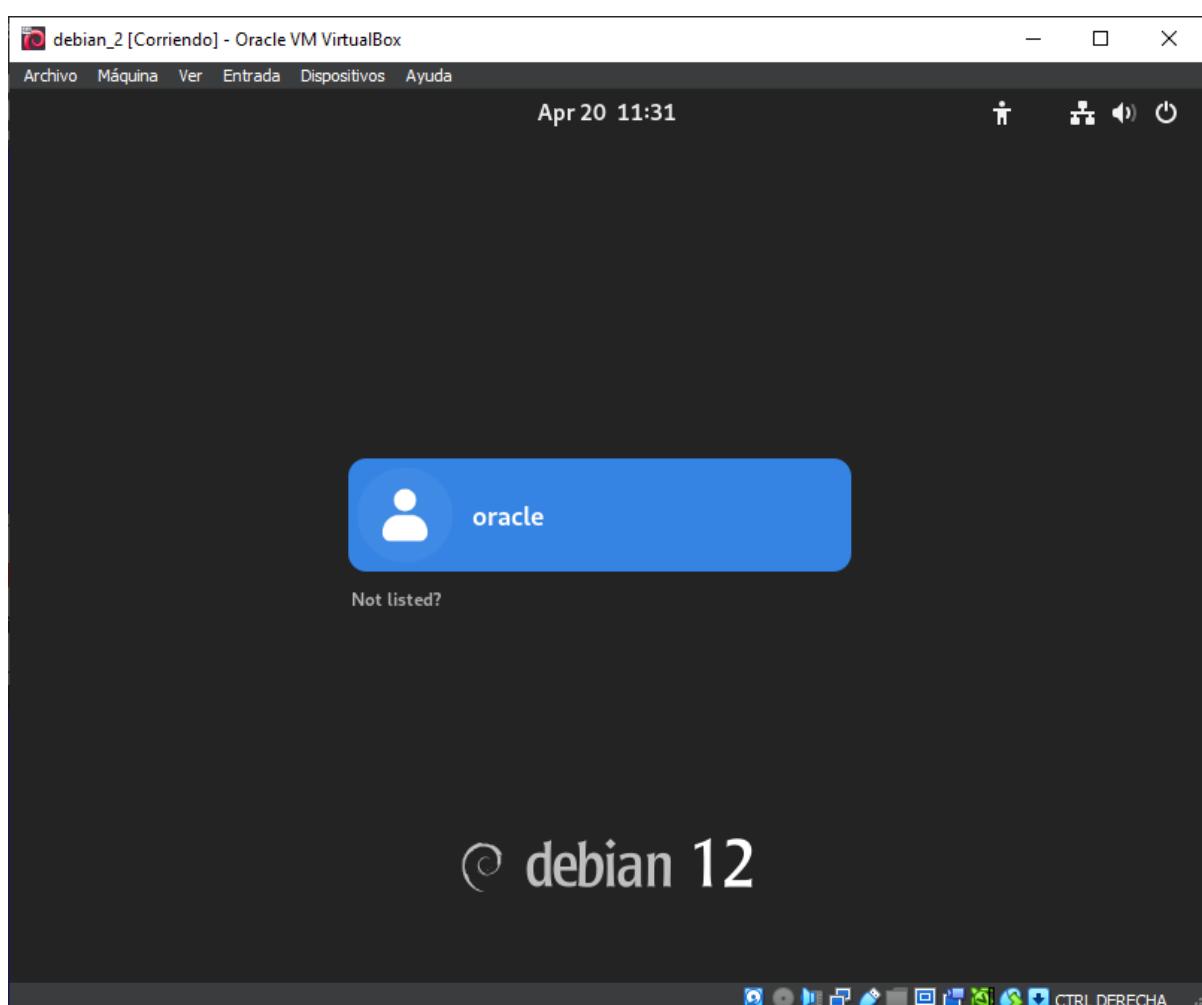
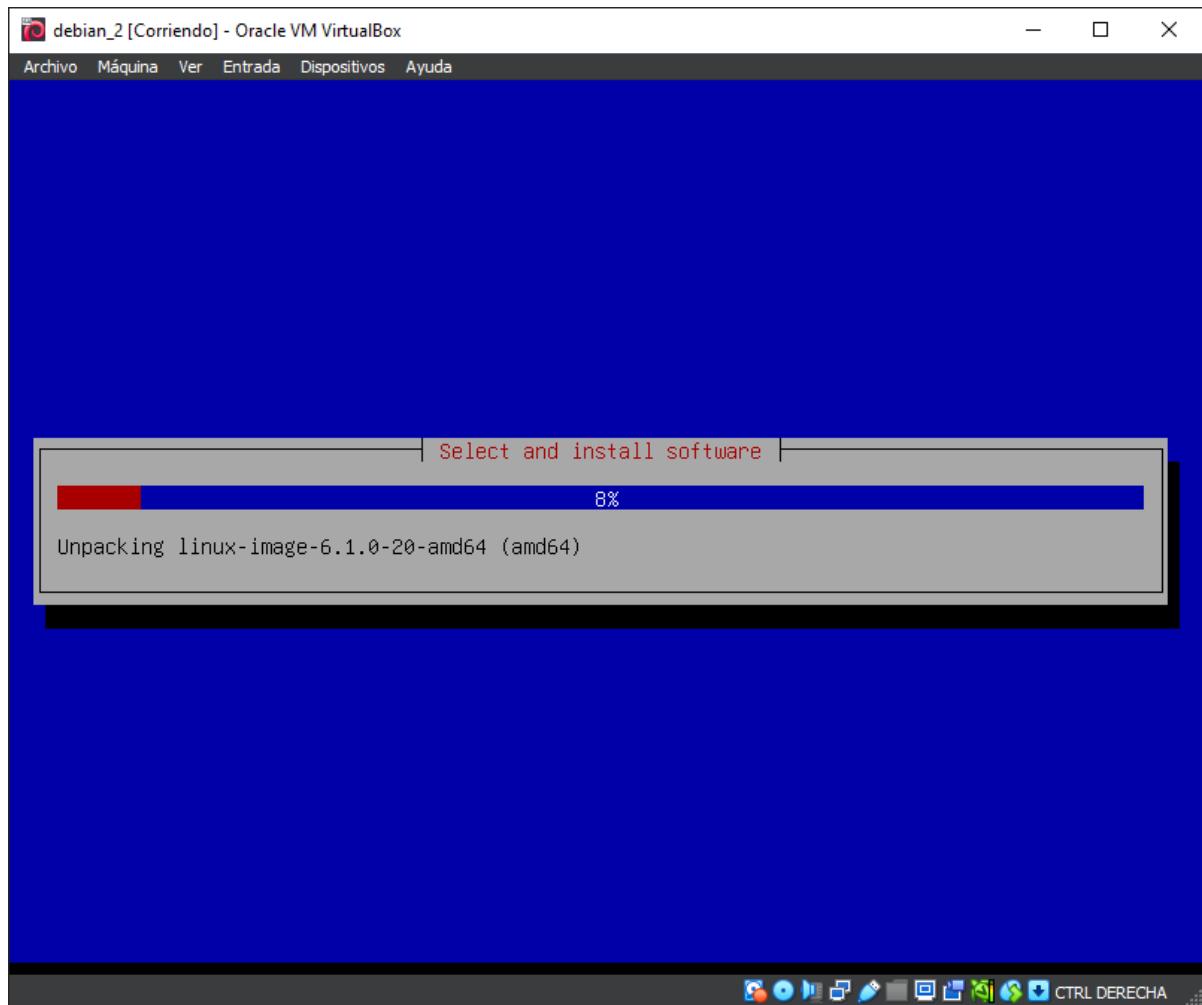
- Samba es una implementación del protocolo **SMB** (Server Message Block) creado por Microsoft. Su objetivo es permitir que sistemas no Microsoft (originalmente sistemas UNIX) intercambien archivos e impresoras con sistemas Windows.
- Samba implementa los protocolos **NetBIOS** y **SMB**, lo que le permite establecer comunicación entre máquinas Unix y productos Microsoft Windows.

2. Funcionamiento de Samba:

- Samba utiliza dos demonios (programas que se ejecutan en segundo plano):
 - **smbd**: Permite la compartición de archivos e impresoras sobre la red SMB, además de proporcionar verificación y autorización para el acceso de clientes SMB.
 - **nmbd**: Busca a través del servicio **Windows Internet Name Service (WINS)** y brinda ayuda mediante un visualizador.
- Estos demonios proporcionan los recursos compartidos a los clientes SMB en la red.

Desarrollo

Paso 1: instalar Debian



Paso 2: Instalar samba

sudo apt-get install samba

3. Configuración de Samba:

- La configuración de Samba en Linux se realiza mediante la edición de un solo archivo ubicado en /etc/samba/smb.conf.
- Aquí tienes un ejemplo básico de configuración:
 - [global]
 - workgroup = PRUEBAGROUP
 - server string = Samba %v
 - wins support = no
 - load printers = no
 - security = user
 - map to guest = bad user
 - guest ok = yes
 - Puedes personalizar esta configuración según tus necesidades.

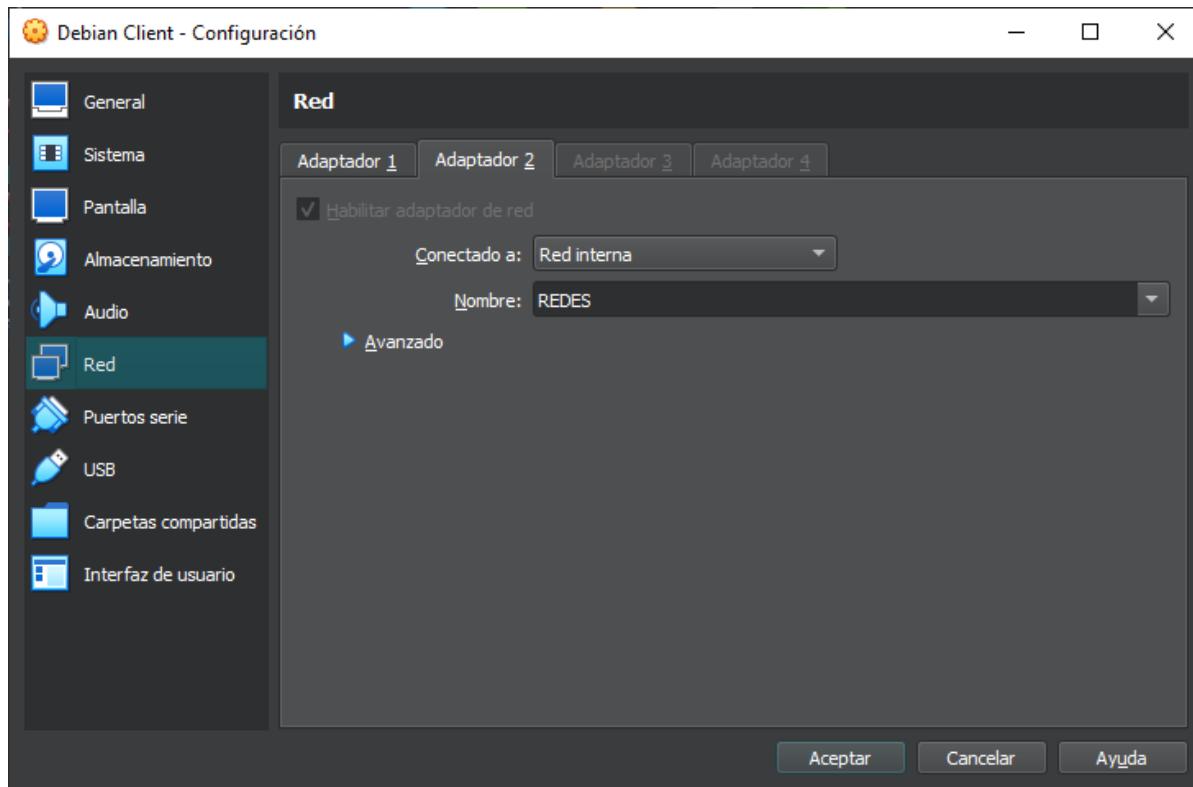
4. Proyecto de Simulación:

- Para tu proyecto, debes crear un servidor de autenticación y archivos que permita la conexión desde clientes con sistemas operativos Windows o Linux.
- El equipo servidor para la autenticación del cliente Linux debe tener activos los servicios de **NFS** y **NIS**.
- El equipo servidor para la autenticación del cliente Windows debe tener activo el servicio de **SAMBA**.
- Los equipos cliente (Linux y Windows) deben poder autenticarse desde el inicio del sistema operativo y automáticamente importar todos los archivos en la cuenta de usuario creada en el servidor.
- Al final, entrega un informe por escrito que detalle todas las instalaciones realizadas y las modificaciones de configuración en los archivos.

¡Buena suerte con tu proyecto!

[^1^23]{.underline}

Configuración de red de VirtualBox para una red interna



Configuración de red del servidor

```
server@debian-server:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:5a:68:05 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.35/24 brd 192.168.100.255 scope global enp0s3
        valid_lft forever preferred_lft forever
        inet6 2806:2f0:90a0:f873:a00:27ff:fe5a:6805/64 scope global dynamic mngtmpaddr
            valid_lft 259139sec preferred_lft 172739sec
        inet6 fe80::a00:27ff:fe5a:6805/64 scope link
            valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:03:32:89 brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.10/24 brd 10.10.10.255 scope global enp0s8
        valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:fe03:3289/64 scope link
            valid_lft forever preferred_lft forever
```

Configuración de red del cliente

```
cliente@cliente-nis:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:7c:58:d3 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.36/24 brd 192.168.100.255 scope global enp0s3
        valid_lft forever preferred_lft forever
        inet6 2806:2f0:90a0:f873:a00:27ff:fe7c:58d3/64 scope global dynamic mngtmpaddr
            valid_lft 259155sec preferred_lft 172755sec
        inet6 fe80::a00:27ff:fe7c:58d3/64 scope link
            valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:96:63:ee brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.11/24 brd 10.10.10.255 scope global enp0s8
        valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:fe96:63ee/64 scope link
            valid_lft forever preferred_lft forever
```

Capturas servidor NIS

Instalación y configuración del servidor NIS

```
server@debian-server:~$ sudo apt-get install rpcbind
```

```
[server@debian-server:~$ sudo service rpcbind start  
[sudo] password for server:
```

```
server@debian-server:~$ sudo apt-get install nis
```

```
server@debian-server:~$ sudo ypdomainname proy3
```

```
server@debian-server:~$ sudo ypdomainname
```

```
proy3
```

```
server@debian-server:~$ cat /etc/defaultdomain  
proy3
```

```
server@debian-server:~$ cat /etc/hosts
```

```
127.0.0.1      localhost
```

```
127.0.1.1      proy3
```

```
10.10.10.10    proy3
```

```
# Cliente NIS
```

```
10.10.10.11    cliente-nis
```

```
# The following lines are desirable for IPv6 capable hosts
```

```
::1      localhost ip6-localhost ip6-loopback
```

```
ff02::1 ip6-allnodes
```

```
ff02::2 ip6-allrouters
```

```
server@debian-server:~$ cat /etc/default/nis
#
# /etc/default/nis      Optional configuration settings for the NIS programs.
#
NISSERVER=true
NISCLIENT=false

# Location of the master NIS password file (for yppasswdd).
# If you change this make sure it matches with /var/yp/Makefile.
YPPWDDIR=/etc

# Do we allow the user to use ypchsh and/or ypchfn ? The YPCHANGEOK
# fields are passed with -e to yppasswdd, see it's manpage.
# Possible values: "chsh", "chfn", "chsh,chfn"
YPCHANGEOK=chsh

# NIS master server.  If this is configured on a slave server then ypinit
# will be run each time NIS is started.
NISMASTER=true

# Additional options to be given to ypserv when it is started.
YPSERVARGS=

# Additional options to be given to ypbind when it is started.
YPBINDARGS=

# Additional options to be given to yppasswdd when it is started. Note
# that if -p is set then the YPPWDDIR above should be empty.
YPPASSWDDARGS=

# Additional options to be given to ypxfrd when it is started.
YPXFRDARGS=
```

```
server@debian-server:~$ cat /etc/nsswitch.conf
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc-reference` and `info` packages installed, try:
# `info libc "Name Service Switch"` for information about this file.

passwd:      files nis winbind
shadow:      files nis
group:       files nis winbind

passwd:      compat winbind
group:       compat winbind

netgroup:    files nis
automount:   files nis

gshadow:     files systemd

hosts:       files mdns4_minimal [NOTFOUND=return] dns myhostname
networks:    files

protocols:   db files
services:    db files
ethers:      db files
rpc:         db files
```

```
server@debian-server:~$ cat /var/yp/securenets
#
# securenets      This file defines the access rights to your NIS server
#                  for NIS clients. This file contains netmask/network
#                  pairs. A clients IP address needs to match with at least
#                  one of those.
#
#                  One can use the word "host" instead of a netmask of
#                  255.255.255.255. Only IP addresses are allowed in this
#                  file, not hostnames.
#
# Always allow access for localhost, IPv4 and IPv6
255.0.0.0      127.0.0.0
host           ::1

# This lines gives access to everybody. PLEASE ADJUST!
#0.0.0.0          0.0.0.0
#:/:0
255.255.255.0  10.10.10.0
```

```
server@debian-server:~$ sudo nano /var/yp/Makefile
```

```
GNU nano 7.2          /var/yp/Makefile

MINUID=5000
MINGID=5000
MAXUID=5500
MAXGID=5500

# Should we merge the passwd file with the shadow file ?
# MERGE_PASSWD=true|false
MERGE_PASSWD=false

# Should we merge the group file with the gshadow file ?
# MERGE_GROUP=true|false
MERGE_GROUP=false
```

```
GNU nano 7.2          /var/yp/Makefile
ALL = passwd shadow group hosts rpc services netid protocols netgrp
```

Creación de usuario NIS

```
server@debian-server:~$ sudo groupadd --gid 5001 nispro
```

```
server@debian-server:~$ sudo adduser --uid 5001 --gid 5001 nispro
```

Contraseña: 1928

```
server@debian-server:~$ sudo service rpcbind start
server@debian-server:~$ sudo service ypserv start
server@debian-server:~$ sudo service ypxfrd start
server@debian-server:~$ sudo service yppasswdd start
```

```
server@debian-server:~$ sudo systemctl enable rpcbind
```

```
server@debian-server:~$ sudo systemctl enable ypserv
```

```
server@debian-server:~$ sudo systemctl enable ypxfrd
```

```
server@debian-server:~$ sudo systemctl enable yppasswdd
```

```
server@debian-server:~$ sudo /usr/lib/yp/ypinit -m
```

At this point, we have to construct a list of the hosts which will run NIS servers. debian-server is in the list of NIS server hosts. Please continue to add

the names for the other hosts, one per line. When you are done with the list, type a <control D>.

```
next host to add: debian-server
next host to add: proy3
next host to add:
```

The current list of NIS servers looks like this:

```
debian-server
proy3
```

```
Is this correct? [y/n: y] y
We need a few minutes to build the databases...
Building /var/yp/proy3/ypservers...
Running /var/yp/Makefile...
gmake[1]: Entering directory '/var/yp/proy3'
Updating passwd.byname...
Updating passwd.byuid...
Updating shadow.byname...
Updating group.byname...
Updating group.bygid...
Updating hosts.byname...
Updating hosts.byaddr...
Updating rpc.byname...
Updating rpc.bynumber...
Updating services.byname...
Updating services.byservicename...
Updating netid.byname...
Updating protocols.bynumber...
Updating protocols.byname...
Updating netgroup...
Updating netgroup.byhost...
Updating netgroup.byuser...
gmake[1]: Leaving directory '/var/yp/proy3'
```

debian-server has been set up as a NIS master server.

Now you can run ypinit -s debian-server on all slave server.

```
server@debian-server:~$ cd /var/yp
server@debian-server:/var/yp$ make
gmake[1]: Entering directory '/var/yp/proy3'
gmake[1]: Nothing to be done for 'all'.
gmake[1]: Leaving directory '/var/yp/proy3'
```

Capturas cliente NIS

```
client@cliente-nis:~$ sudo apt-get install nis
```

```
cliente@cliente-nis:~$ sudo ypdomainname proy3
cliente@cliente-nis:~$ sudo ypdomainname
proy3
```

```
cliente@cliente-nis:~$ cat /etc/defaultdomain
proy3
```

```
cliente@cliente-nis:~$ cat /etc/default/nis
#
# /etc/defaults/nis      Optional configuration settings for the NIS programs.
#
NISSERVER=false
NISCLIENT=true

# Location of the master NIS password file (for yppasswdd).
# If you change this make sure it matches with /var/yp/Makefile.
YPPWDDIR=/etc

# Do we allow the user to use ypchsh and/or ypchnfn ? The YPCHANGEOK
# fields are passed with -e to yppasswdd, see it's manpage.
# Possible values: "chsh", "chnfn", "chsh,chnfn"
YPCHANGEOK=chsh

# NIS master server.  If this is configured on a slave server then ypinit
# will be run each time NIS is started.
NISMASTER=false

# Additional options to be given to ypserv when it is started.
YPSERVARGS=

# Additional options to be given to ypbind when it is started.
YPBINDARGS=

# Additional options to be given to yppasswdd when it is started.  Note
# that if -p is set then the YPPWDDIR above should be empty.
YPPASSWDDARGS=

# Additional options to be given to ypxfrd when it is started.
YPXFRDARGS=
```

```
client@cliente-nis:~$ cat /etc/yp.conf
#
# yp.conf      Configuration file for the ypbind process. You can define
#               NIS servers manually here if they can't be found by
#               broadcasting on the local net (which is the default).
#
#               See the manual page of ypbind for the syntax of this file.
#
# IMPORTANT:   For the "ypserver", use IP addresses, or make sure that
#               the host is in /etc/hosts. This file is only interpreted
#               once, and if DNS isn't reachable yet the ypserver cannot
#               be resolved and ypbind won't ever bind to the server.
#
# ypserver ypserver.network.com
ypserver 10.10.10.10
```

```
client@cliente-nis:~$ cat /etc/hosts
127.0.0.1      localhost
127.0.1.1      cliente-nis
10.10.10.11    cliente-nis

# Servidor NIS
10.10.10.10    proy3

# The following lines are desirable for IPv6 capable hosts
::1      localhost ip6-localhost ip6-loopback
ff02::1  ip6-allnodes
ff02::2  ip6-allrouters
```

```
client@cliente-nis:~$ cat /etc/nsswitch.conf
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc-reference' and `info' packages installed, try:
# `info libc "Name Service Switch"' for information about this file.

passwd:        compat nis
shadow:        compat nis
group:         compat nis
gshadow:       files systemd

#hosts:         files mdns4_minimal [NOTFOUND=return] dns myhostname
hosts:         files dns nis
networks:      files

protocols:     db files
services:      db files
ethers:        db files
rpc:           db files

netgroup:      nis
```

```
client@cliente-nis:~$ sudo service ypbind start
```

```
client@cliente-nis:~$ sudo ypcat passwd
nispro:x:5001:5001:,:/home/nispro:/bin/bash
client_windows:x:5002:5002:/home/client_windows:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
```

Instalación y configuración de NFS

```
server@debian-server:~$ sudo apt-get install nfs-kernel-server nfs-common
```

```
server@debian-server:~$ cat /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_subtree_check)
#
# 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/nispro     10.10.10.11(rw,sync,root_squash,no_subtree_check)
```

```
server@debian-server:~$ sudo exportfs -av
exporting 10.10.10.11:/home/nispro
```

```
server@debian-server:~$ sudo service nfs-kernel-server start
```

```
server@debian-server:~$ sudo systemctl enable nfs-kernel-server
```

```
server@debian-server:~$ sudo service rpcbind start
```

```
server@debian-server:~$ sudo systemctl enable rpcbind start
```

Capturas cliente NFS

```
client@cliente-nis:~$ sudo apt-get install nfs-client
```

```
client@cliente-nis:~$ mkdir -p /home/nispro
```

```
client@cliente-nis:~$ sudo chmod 777 /home/nispro
```

```
client@cliente-nis:~$ sudo chown nobody:nogroup /home/nispro
```

```
client@cliente-nis:~$ ls -l /home
total 8
drwx----- 15 client client 4096 May  3 18:31 client
drwxrwxrwx 13 nobody nogroup 4096 May  5 15:01 nispro
```

```
client@cliente-nis:~$ mount -t nfs 10.10.10.10:/home/nispro /home/nispro
```

```
cliente@cliente-nis:~$ cat /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).
#
# systemd generates mount units based on this file, see systemd.mount(5).
# Please run 'systemctl daemon-reload' after making changes here.
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=4fe0c856-89b1-4314-b56d-1b4cfe8fc739 / ext4 errors=remount-ro 0 1
# swap was on /dev/sda5 during installation
UUID=c3f3d403-51d6-440d-bcbd-9e1e3f26fee9 none swap sw 0 0
/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0
10.10.10.10:/home/nispro /home/nispro
```

Filesystem	Size	Used	Avail	Use%	Mounted on
udev	3.8G	0	3.8G	0%	/dev
tmpfs	778M	1.5M	776M	1%	/run
/dev/sda1	14G	5.2G	7.9G	40%	/
tmpfs	3.8G	0	3.8G	0%	/dev/shm
tmpfs	5.0M	8.0K	5.0M	1%	/run/lock
10.10.10.10:/home/nispro	14G	5.4G	7.7G	42%	/home/nispro
tmpfs	778M	300K	778M	1%	/run/user/5001
tmpfs	778M	60K	778M	1%	/run/user/113
tmpfs	778M	52K	778M	1%	/run/user/1000

Login por consola desde el cliente

```
cliente@cliente-nis:~$ sudo login
cliente-nis login: nispro
Password:
Linux cliente-nis 6.1.0-20-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.85-1 (2024-04-11) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun May  5 14:49:17 CST 2024 on pts/1
nispro@cliente-nis:~$
```

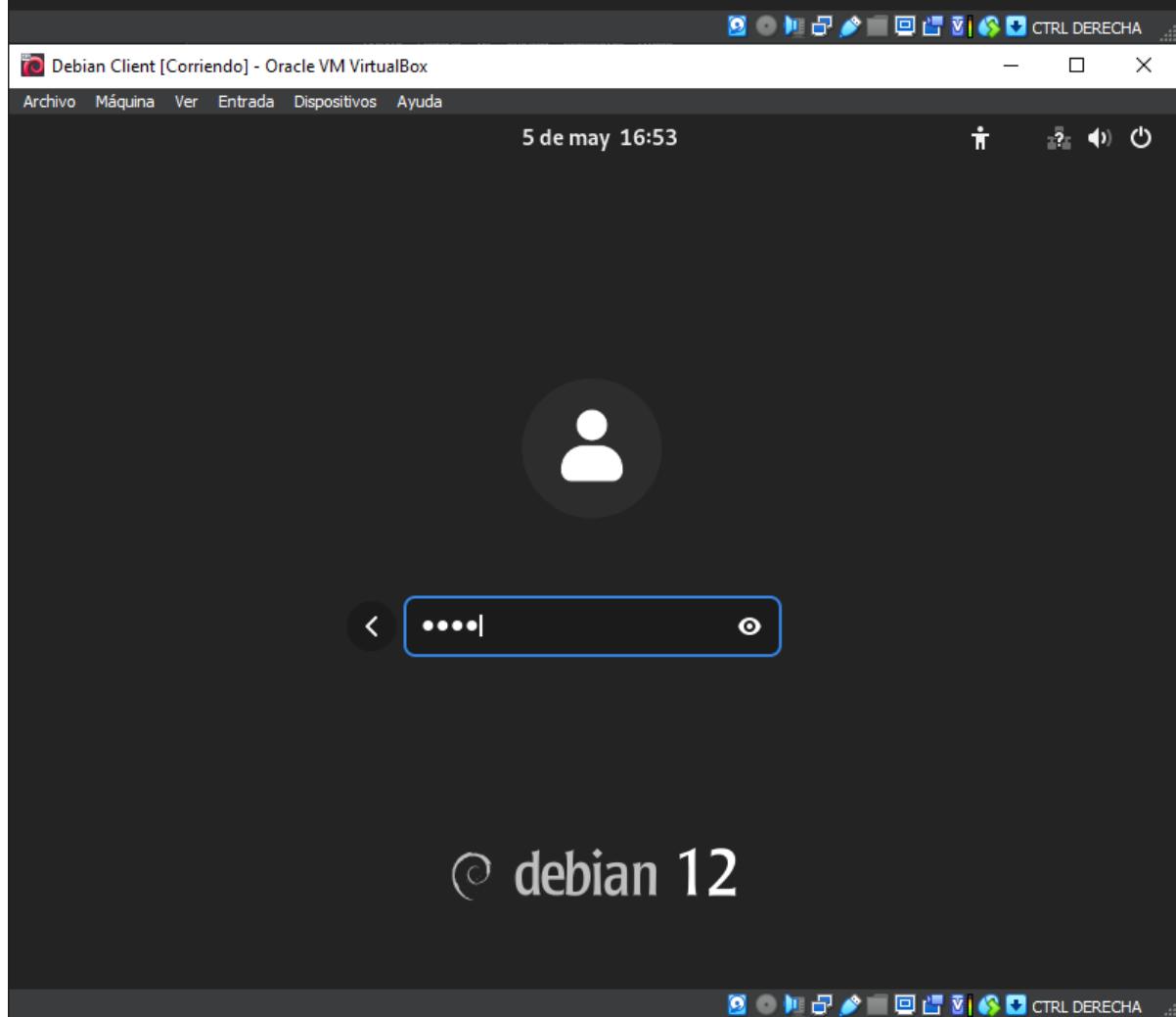
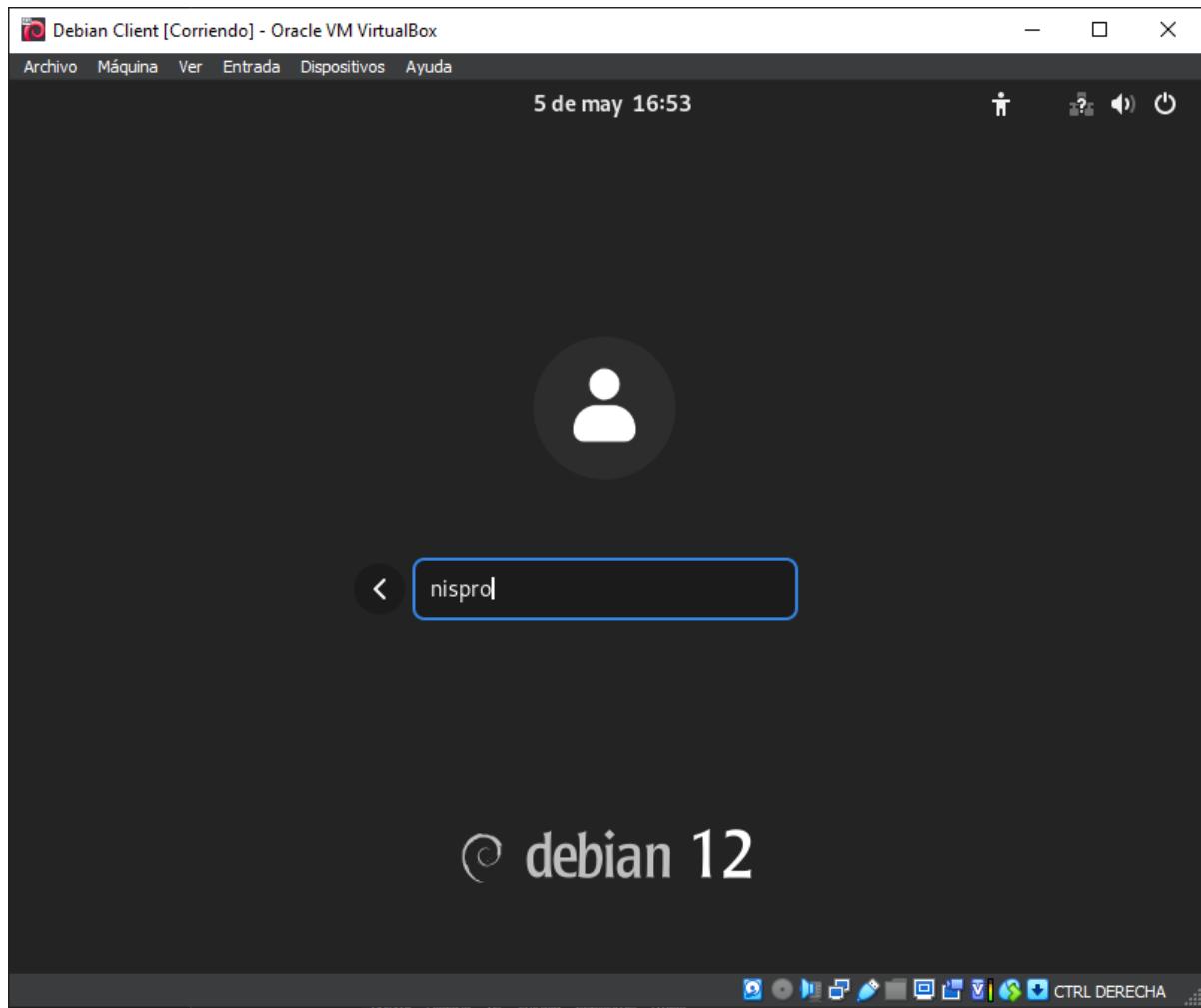
Creación de archivo desde el servidor

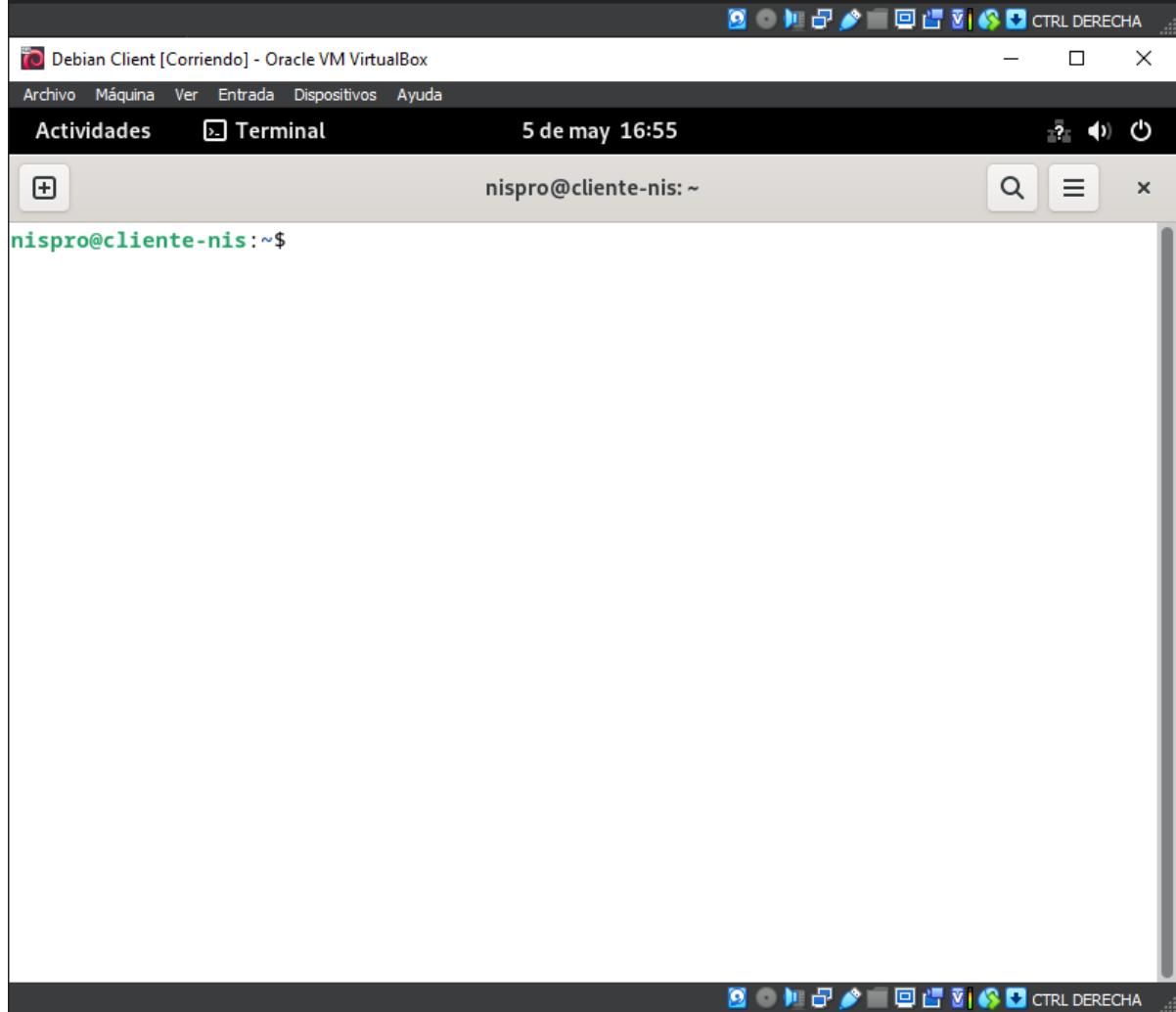
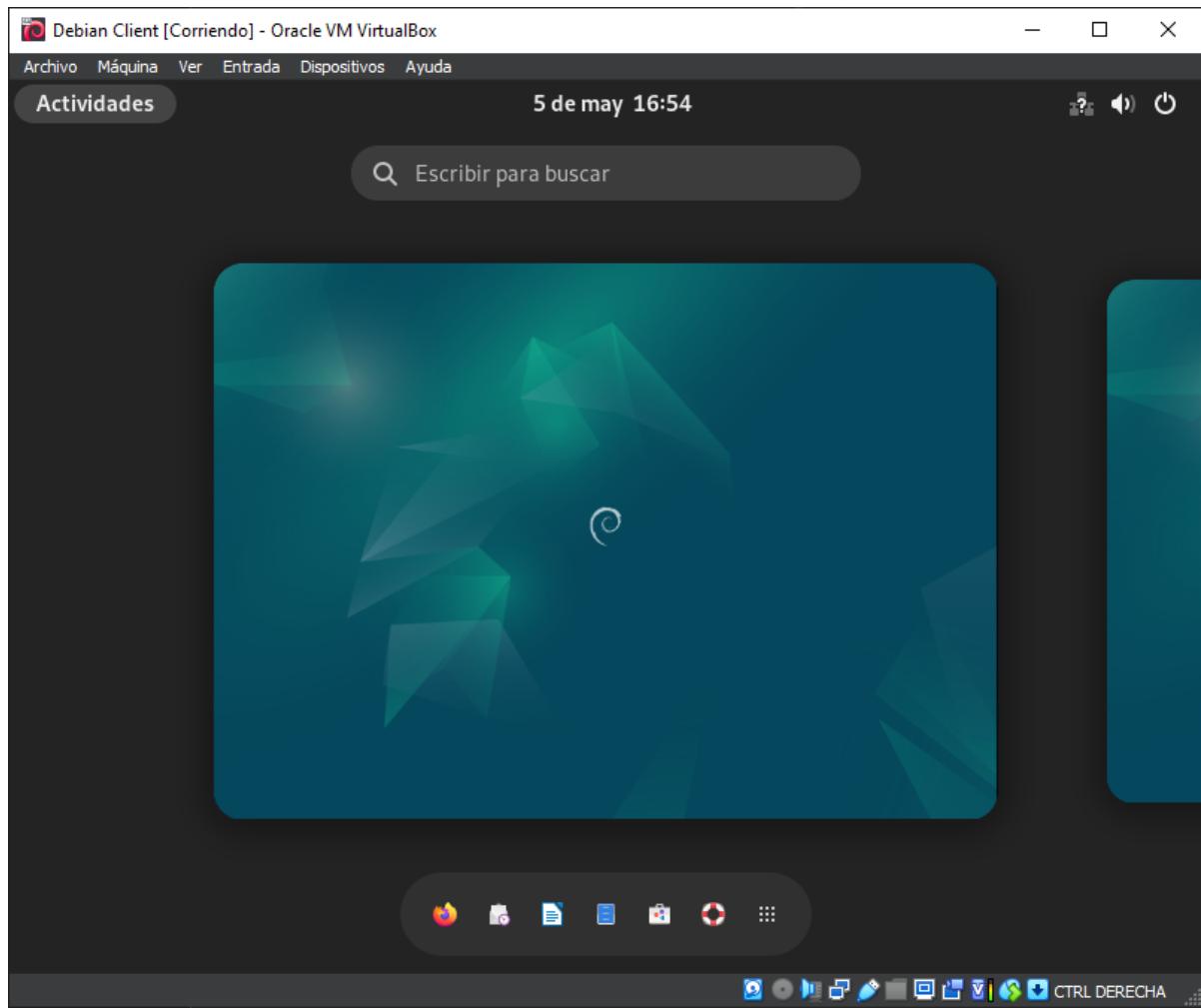
```
server@debian-server:~$ cat /home/nispro/test.txt
Create from server
```

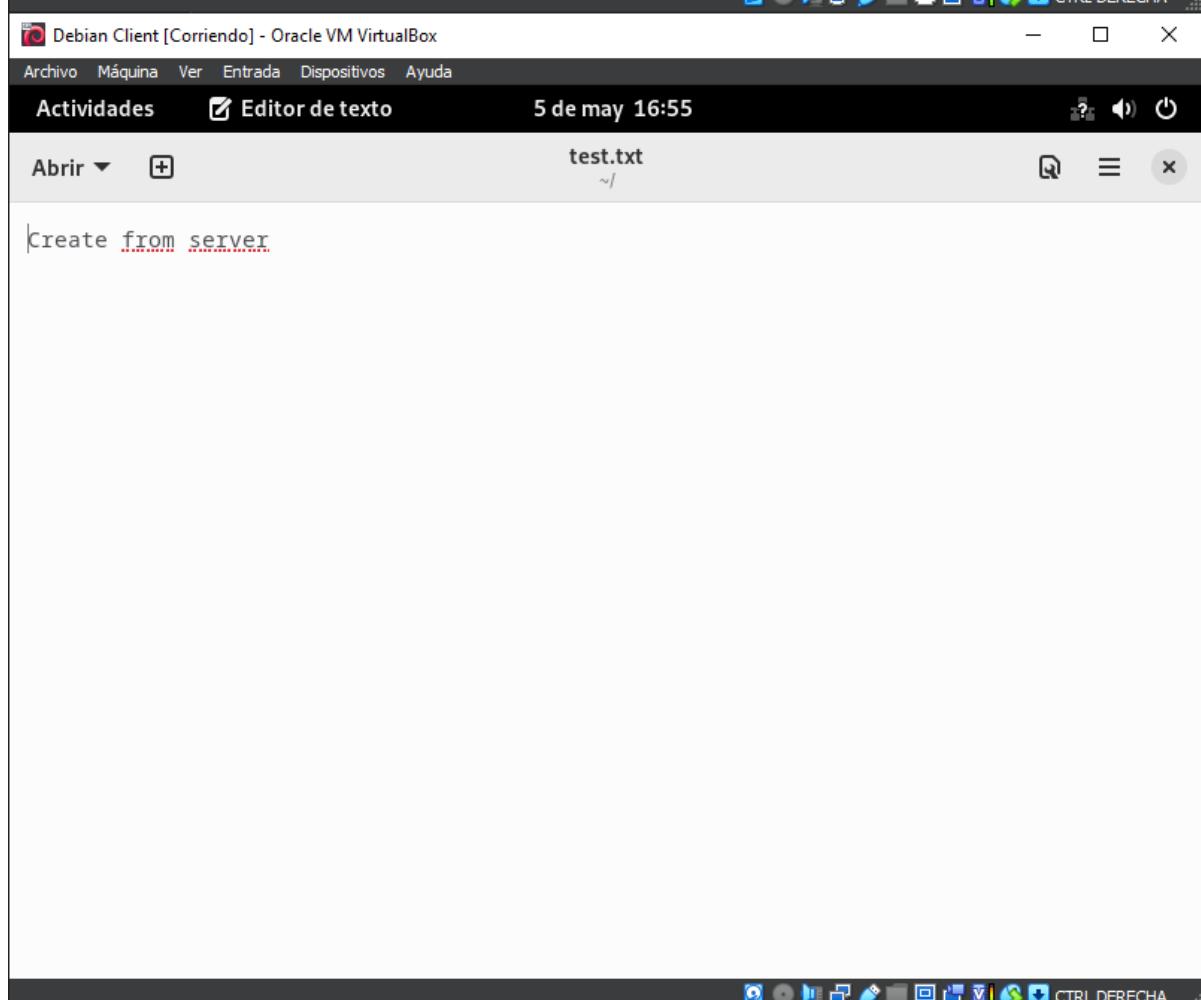
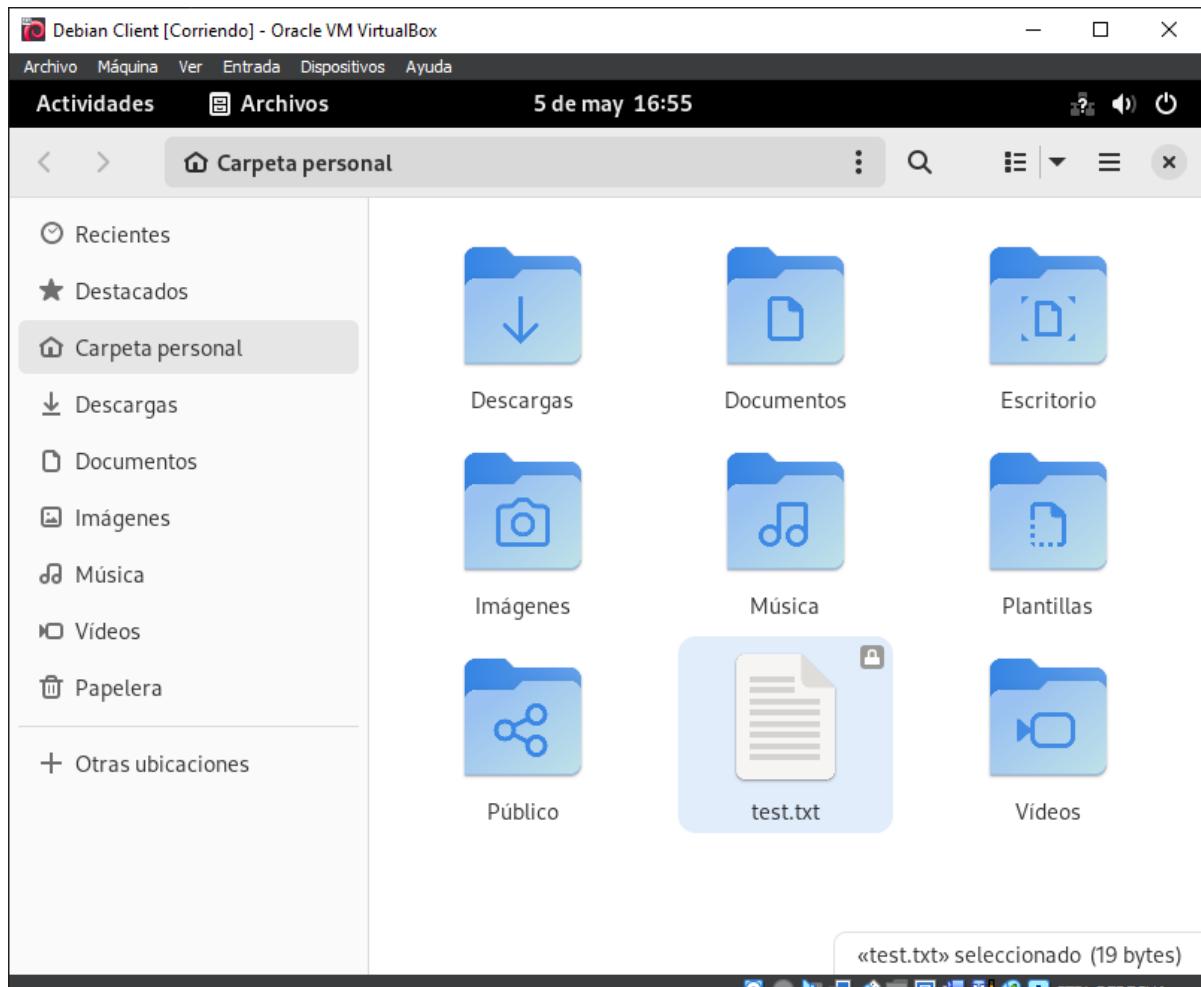
Vista del archivo desde el cliente

```
nispro@cliente-nis:~$ ls
Descargas Documentos Escritorio Imágenes Música Plantillas Público test.txt Vídeos
nispro@cliente-nis:~$ cat test.txt
Create from server
```

Login desde el sistema operativo







Configuración e instalación de samba en el
servidor

```
server@debian-server:~$ apt install samba smbclient attr winbind libpam-winbind libnss-winbind  
libpam-krb5 krb5-config krb5-user
```

Configurando la autenticaci

Cuando los usuarios intentan usar Kerberos y especifican un nombre principal o de usuario sin aclarar a qué dominio administrativo de Kerberos pertenece el principal, el sistema toma el reino predeterminado. El reino predeterminado también se puede utilizar como el reino de un servicio de Kerberos que se ejecute en la máquina local. Normalmente, el reino predeterminado es el nombre en mayúsculas del dominio del DNS local.

Reino predeterminado de la versión 5 de Kerberos:

```
PROY3
```

<Ok>

Configurando la autenticaci

Introduzca los nombres de los servidores Kerberos en el reino PROY3 de Kerberos, separados por espacios.

Servidores de Kerberos para su reino:

```
proy3
```

<Ok>

Configurando la autenticaci

Introduzca el nombre del servidor administrativo (cambio de contraseña) para el reino PROY3 de Kerberos.

Servidor administrativo para su reino de Kerberos:

```
proy3
```

<Ok>

```
server@debian-server:~$ sudo systemctl stop samba-ad-dc smbd nmbd winbind
```

```
server@debian-server:~$ sudo disable samba-ad-dc smbd nmbd winbind
```

```
server@debian-server:/etc/samba$ mv /etc/samba/smb.conf /etc/samba/smb.conf.default
```

```
server@debian-server:/etc/samba$ sudo samba-tool domain provision --use-rfc2307 --interactive
```

Realm: proy3.local

Domain [proy3]:

Server Role (dc, member, standalone) [dc]:

DNS backend (SAMBA_INTERNAL, BIND9_FLATFILE, BIND9_DLZ, NONE) [SAMBA_INTERNAL]:

DNS forwarder IP address (write 'none' to disable forwarding) [192.168.100.1]: 10.10.10.10

Administrator password:

Retype password:

Contraseña: 12345678As

```
server@debian-server:~$ cat /etc/samba/smb.conf
# Global parameters
[global]
    dns forwarder = 10.10.10.10
    netbios name = DEBIAN-SERVER
    realm = PROY3.LOCAL
    server role = active directory domain controller
    workgroup = PROY3
    idmap_ldb:use rfc2307 = yes

[sysvol]
    path = /var/lib/samba/sysvol
    read only = No

[netlogon]
    path = /var/lib/samba/sysvol/proy3.local/scripts
    read only = No
```

Samba-tool para crear un usuario

```
server@debian-server:~$ sudo samba-tool user add samba-client --password=12345678As
WARNING: Using password on command line is insecure. Please install the setproctitle python module.
New Password:
Retype Password:
User 'samba-client' added successfully
```

```
server@debian-server:~$ sudo samba-tool user list
Administrator
krbtgt
samba-client
Guest
```

```
server@debian-server:~$ mkdir /home/samba/client_windows
```

Modificación del archivo smb.conf

```
server@debian-server:~$ cat /etc/samba/smb.conf
# Global parameters
[global]
    dns forwarder = 10.10.10.10
    netbios name = DEBIAN-SERVER
    realm = PROY3.LOCAL
    server role = active directory domain controller
    workgroup = PROY3
    idmap_ldap:use rfc2307 = yes

[sysvol]
    path = /var/lib/samba/sysvol
    read only = No

[netlogon]
    path = /var/lib/samba/sysvol/proy3.local/scripts
    read only = No

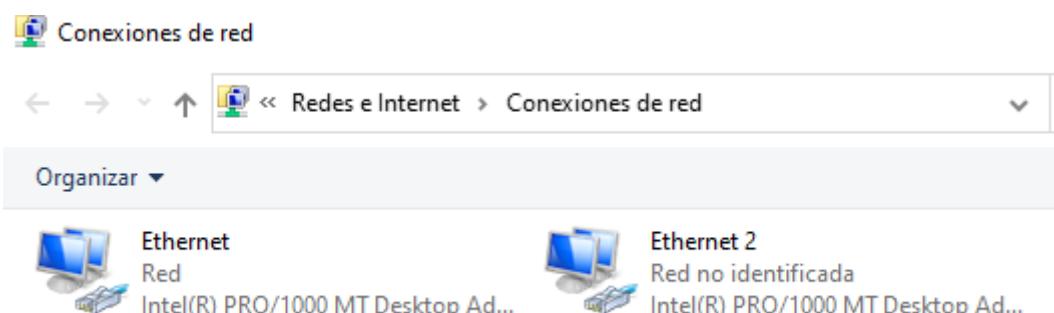
[client_windows]
    path = /home/samba/client_windows
    read only = no
    guest ok = no
    valid users = samba-client
```

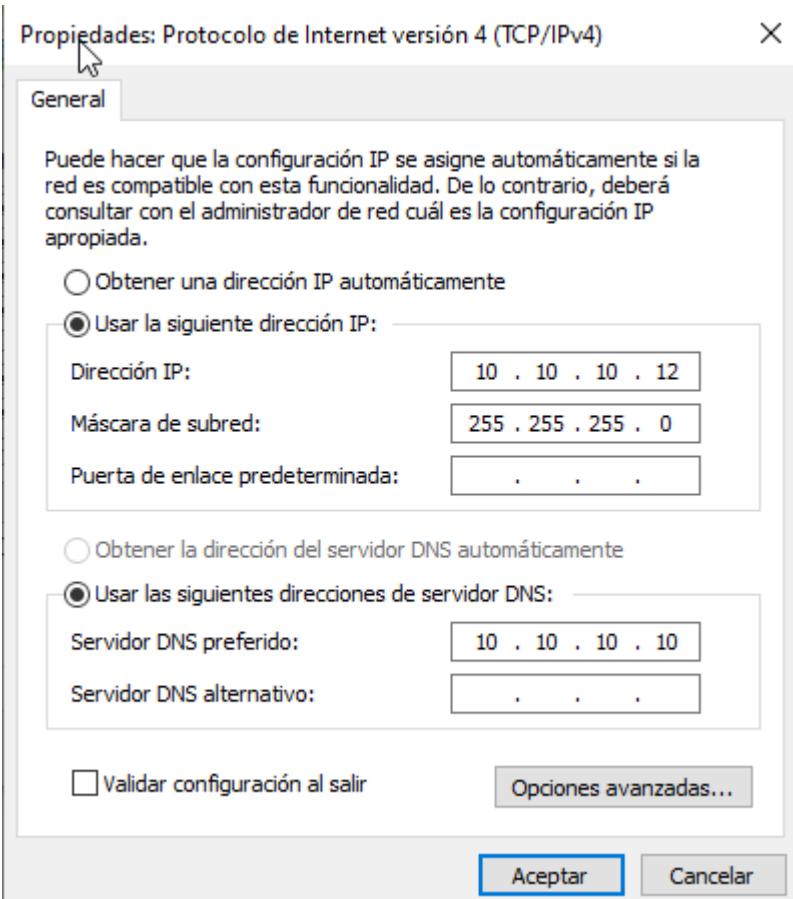
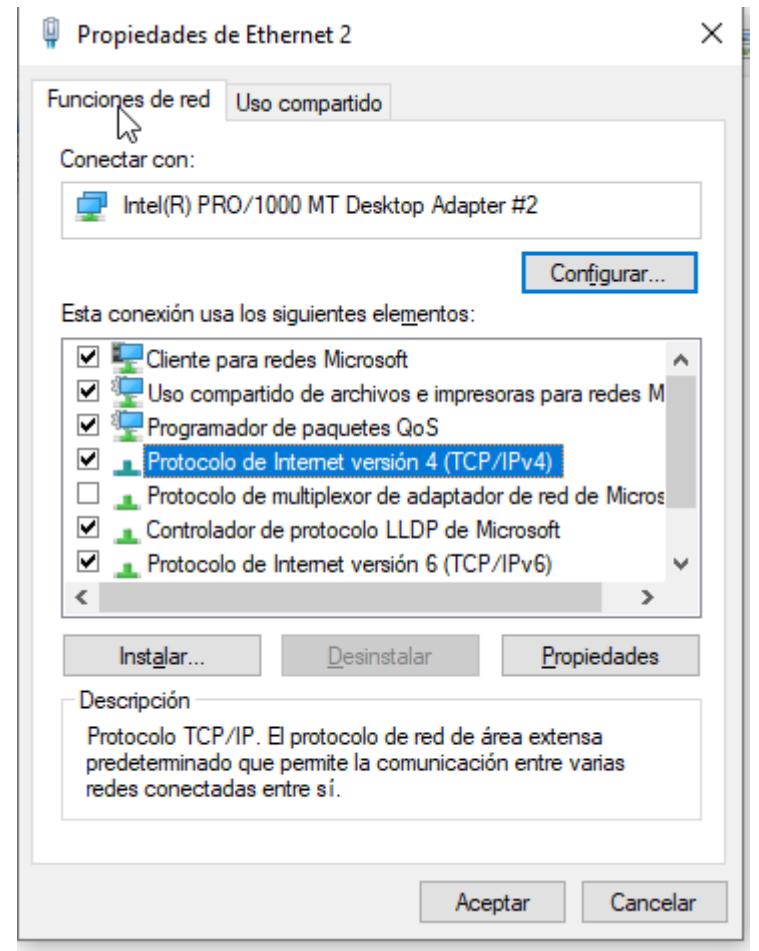
```
server@debian-server:~$ sudo systemctl unmask samba-ad-dc
```

```
server@debian-server:~$ sudo systemctl start samba-ad-dc
```

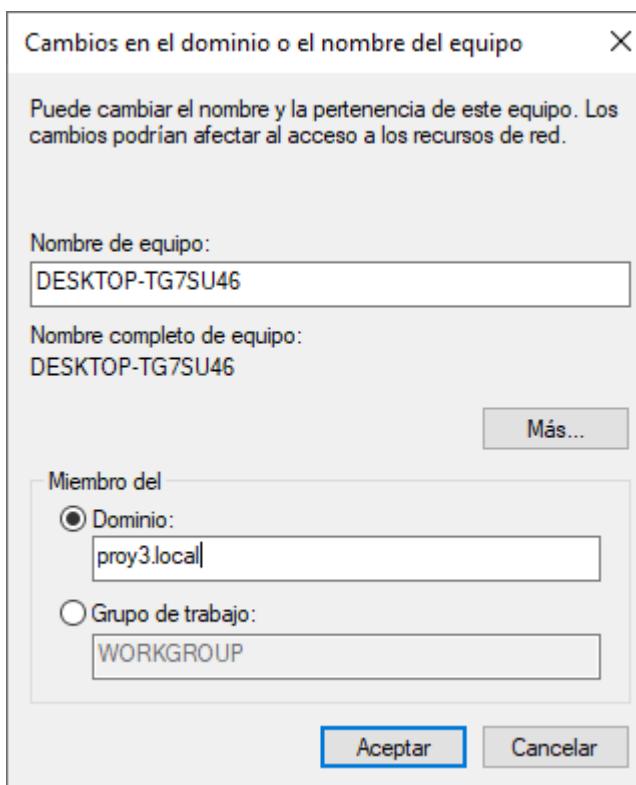
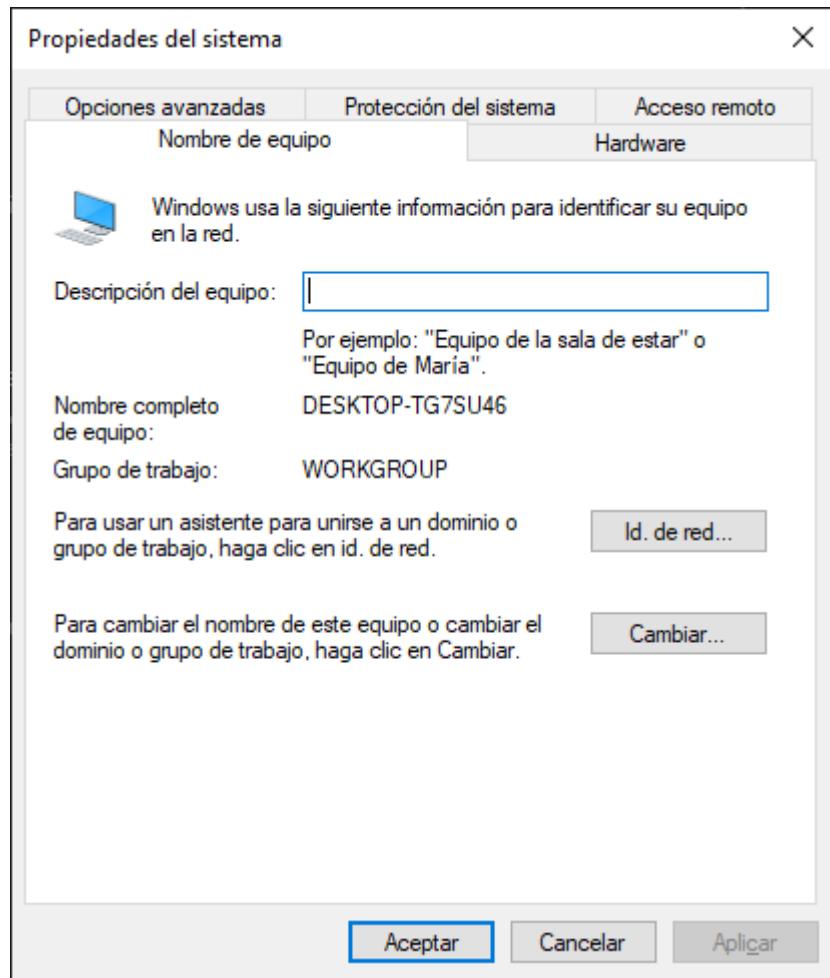
```
server@debian-server:~$ sudo systemctl enable samba-ad-dc
```

Configuración de samba en windows cliente





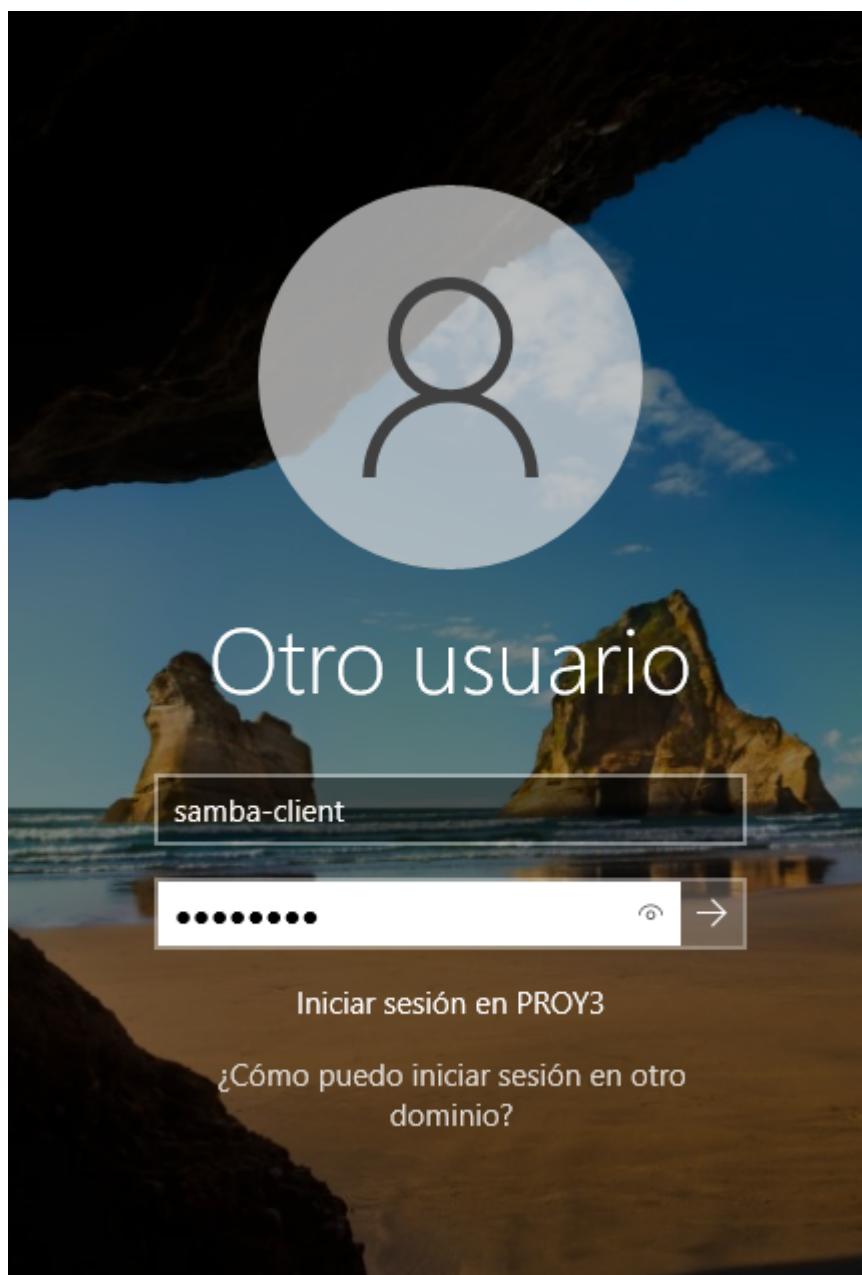
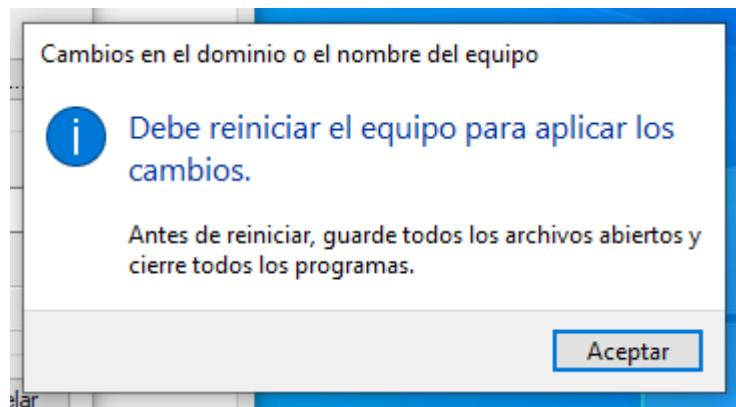
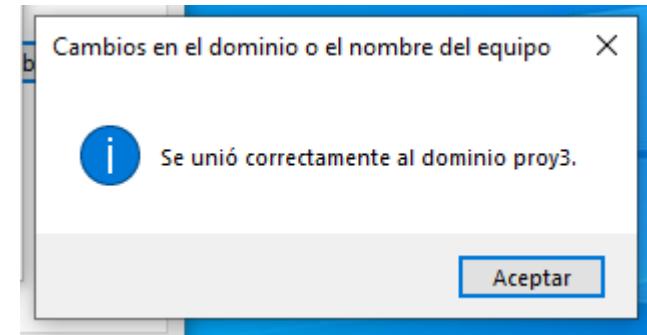
Configuración avanzada del sistema



Ingresa el usuario y contraseña del administrador de samba en el servidor:

- Administrator
- 12345678As

Reinicia el sistema

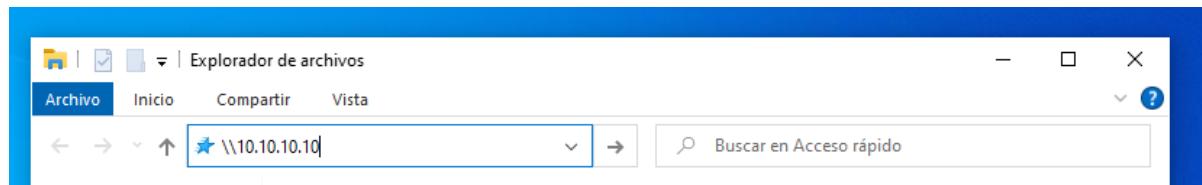


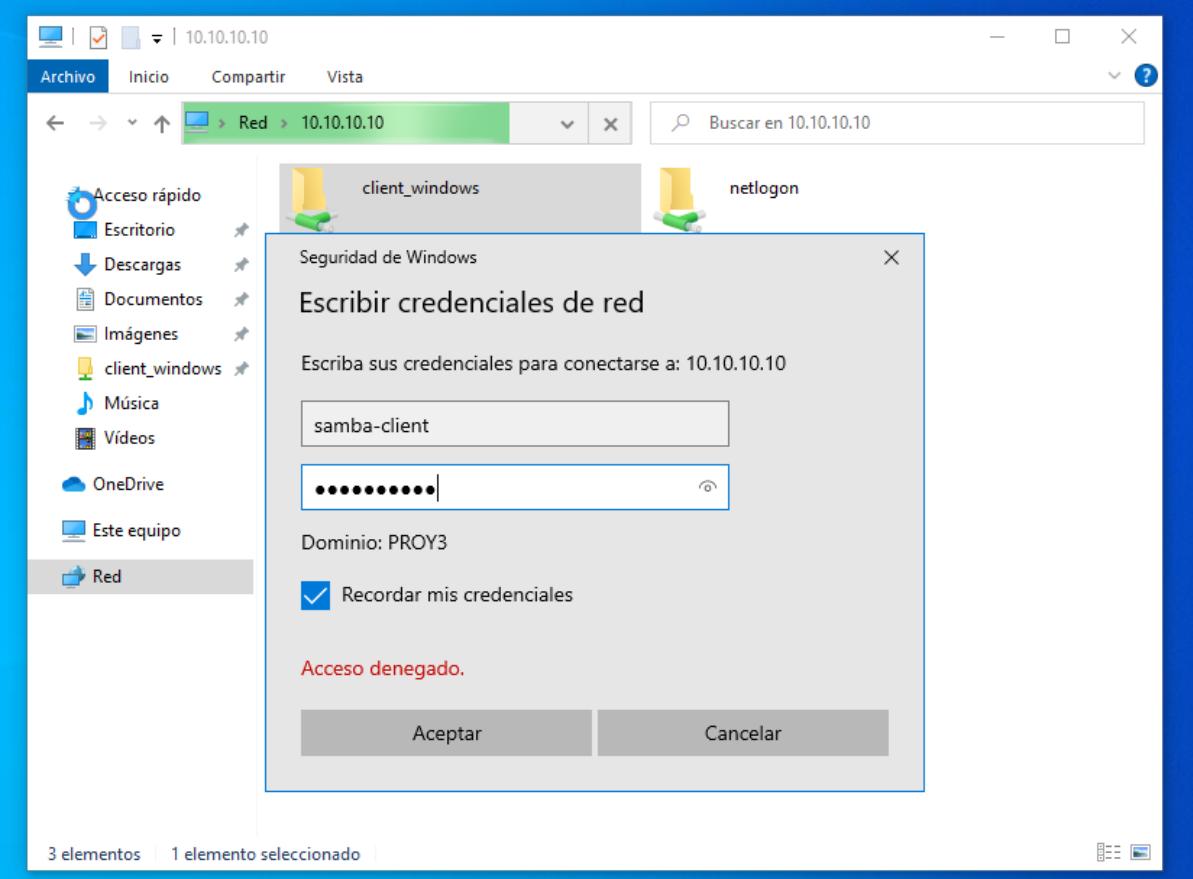
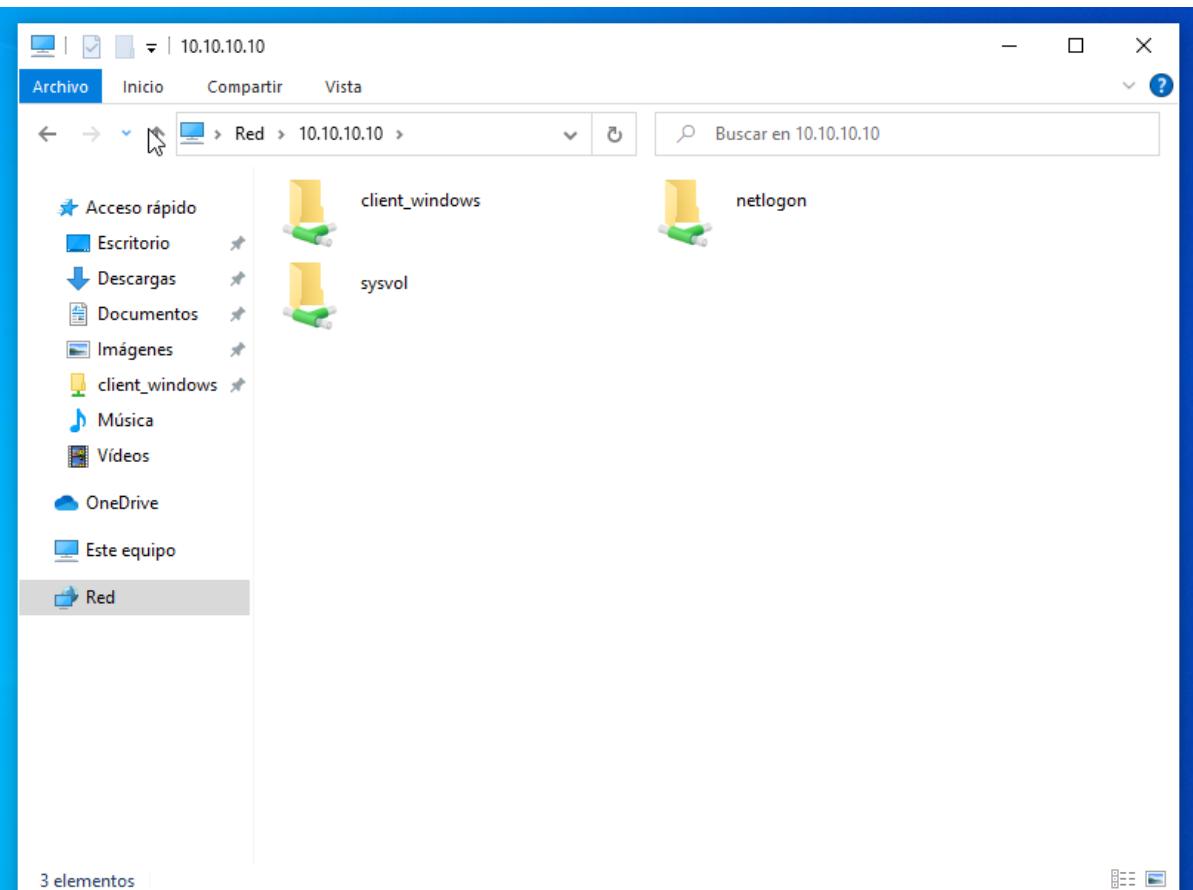
Creación de un archivo en el servidor

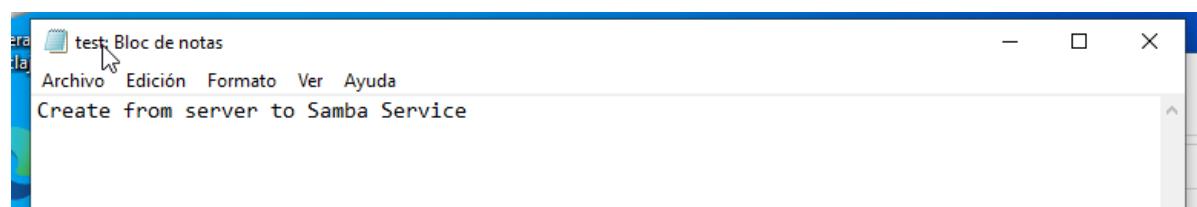
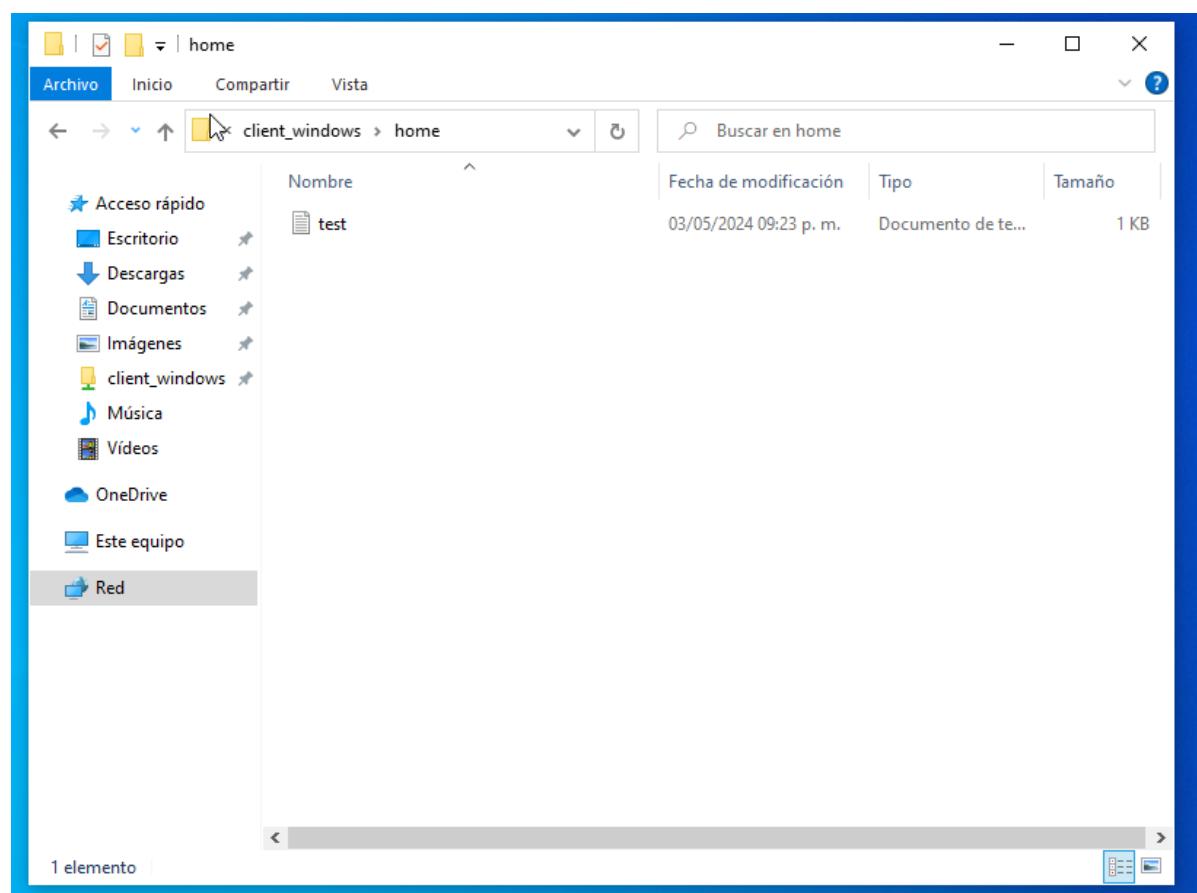
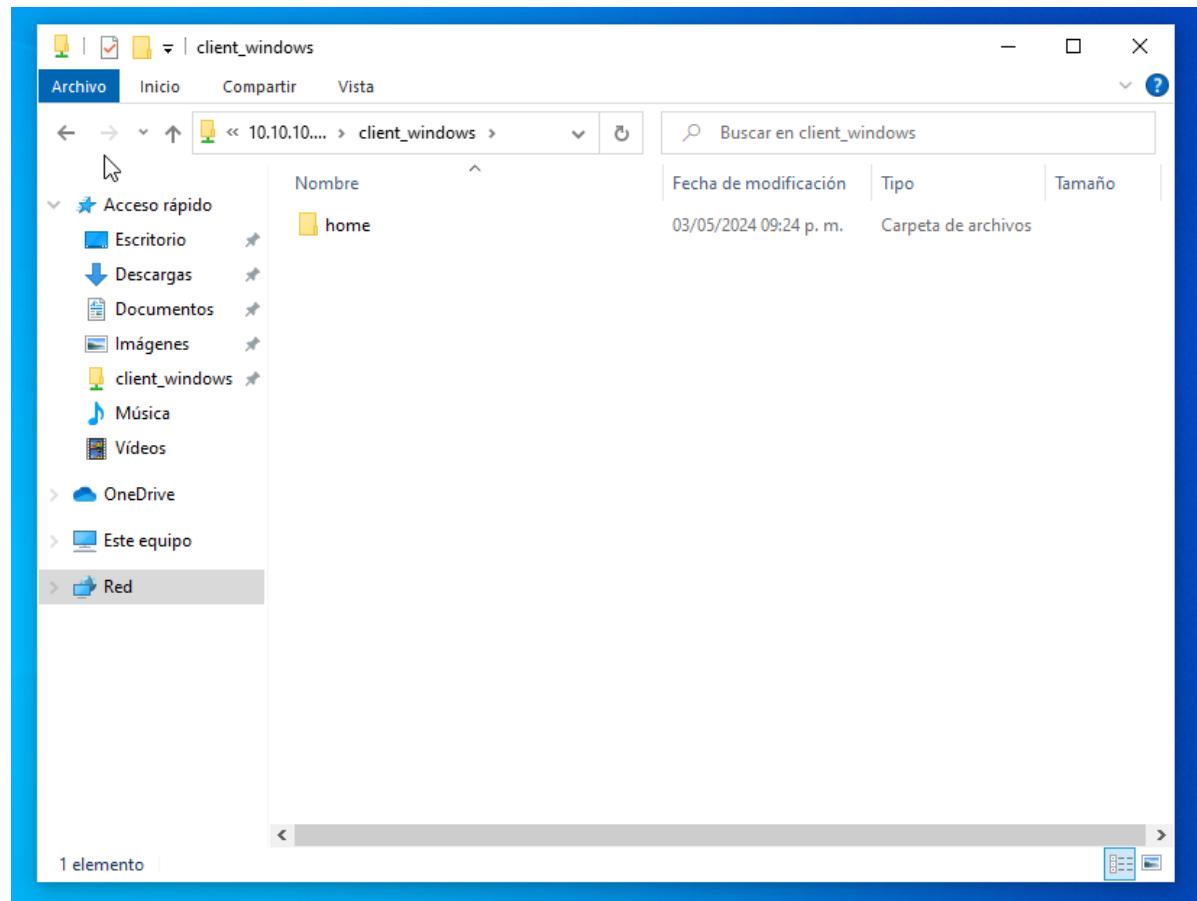
```
server@debian-server:~$ mkdir /home/samba/client_windows/home
```

```
server@debian-server:~$ cat /home/samba/client_windows/home/test.txt
Create from server to Samba Service
```

Vista del archivo desde windows







Conclusión

por Barrera Peña Víctor Miguel

La implementación del servidor Samba permitió cumplir con el objetivo de autenticar y compartir archivos entre sistemas Windows y Linux. La configuración de Samba se realizó correctamente, permitiendo a los clientes acceder a los recursos compartidos sin inconvenientes. Este éxito demuestra la capacidad de Samba para integrar sistemas operativos distintos en una red segura y funcional, facilitando la interoperabilidad en el entorno del proyecto.

por Ramírez González José Miguel

Se encontraron dificultades significativas en la configuración de los servicios NFS y NIS para la autenticación de clientes Linux. Los problemas de sincronización y autenticación retrasaron el proyecto, requiriendo ajustes adicionales y una comprensión más profunda de estos servicios. Aunque finalmente se lograron los objetivos, estos desafíos resaltan la complejidad de gestionar autenticaciones en sistemas Linux, evidenciando áreas de mejora en futuras implementaciones.

por Sánchez Manzano Mariana

El uso de Samba para la autenticación de clientes Windows fue efectivo, cumpliendo con los requisitos del proyecto. La configuración se realizó sin problemas, permitiendo que los clientes Windows accedieran a los archivos en el servidor de manera eficiente. Este éxito subraya la robustez de Samba como herramienta para la gestión de redes en entornos Windows, facilitando una integración sencilla y segura.

por

La configuración de la red interna en VirtualBox permitió simular eficazmente el entorno del laboratorio de computación. Se establecieron conexiones exitosas entre los servidores y clientes, validando la configuración de red y la interoperabilidad entre sistemas. Esta simulación demostró la viabilidad de la solución propuesta, asegurando que las configuraciones realizadas en VirtualBox puedan ser replicadas en un entorno real sin mayores inconvenientes.