



UA

Hito2Asorc

DanielAsensiRochDNI : 48776120C

December 19, 2021

Contents

1	Introducción	3
2	Configuración IP Estática	3
2.1	Debian	3
2.2	FreeBSD	3
2.3	Windows 10	3
3	Particionado de Discos	4
3.1	FREEBSD	4
3.2	Debian	4
3.3	Windows 10	4
4	Servicio SSH, SFTP y SCP	6
4.1	Debian	6
4.2	FreeBSD	6
4.3	Windows 10	7
4.4	Ejecución	8
5	VNC	9
5.1	FreeBSD	9
5.2	Debian	9
5.3	Windows 10	10
6	XRDP	11
6.1	Debian	11
6.2	FreeBSD	11
6.3	Windows 10	12
7	NFS	14
7.1	Debian	14
7.2	FreeBSD	14
7.3	Windows 10	15
8	SAMBA	17
8.1	Debian	17
8.2	FreeBSD	17
8.3	Windows 10	18
9	Cups servidor de impresión	20
9.1	FreeBSD	20
9.2	Windows 10	22
9.3	Debian	23
10	Base de datos y Web	24
10.1	Debian con MariaDb	24
10.2	FreeBSD y Postgres	26
10.3	Windows	27
11	TrueNAS + ISCSI	28
11.1	Instalación Máquina virtual	28
11.2	Debian	29
11.3	Windows 10	29

12 OwnCloud + Git	31
12.1 Debian	31
12.1.1 Owncloud	31
12.1.2 Git	32
13 DNS	33
13.1 FreeBSD	33
13.2 Windows	34
14 Web-Server	35
14.1 Freebsd	35

1 Introducción

En este Hito de la asignatura realizaremos la instalación de diferentes servicios útiles, que nos podrían servir en nuestro día a día como desarrolladores. Los sistemas operativos que utilizaremos en esta práctica serán FreeBSD, Debian, y Windows Server 2020.

Para realizar la instalación de los servicios y que el usuario final pueda disfrutar con seguridad de los mismos, primero deberemos realizar una configuración del sistema creando las particiones necesarias dentro del disco de nuestro sistema, para que un fallo no involucre al sistema completo.

2 Configuración IP Estática

2.1 Debian

Para configurar la Ip estática en Debian deberos acceder a la siguiente ruta y escribir las líneas correspondientes:

```
sudo nano /etc/networks/interfaces

auto enp0s8
iface enp0s8 inet static
    address 192.168.56.222
    netmask 255.255.255.0
```

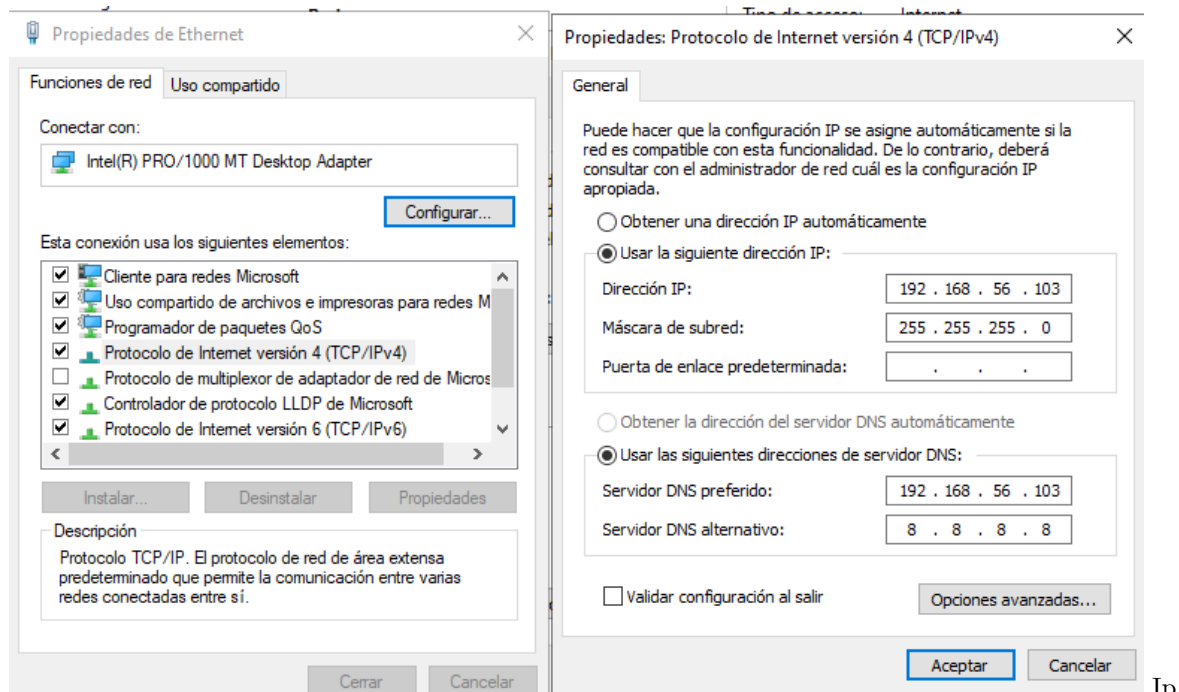
2.2 FreeBSD

En FreeBSD se realizará la configuración desde /etc/rc.conf:

```
ifconfig_em1="inet 192.168.56.221 netmask 255.255.255.0"
```

2.3 Windows 10

Para configurar la ip estática en windows 10 deberemos seguir los siguientes pasos:



Estática Windows 10

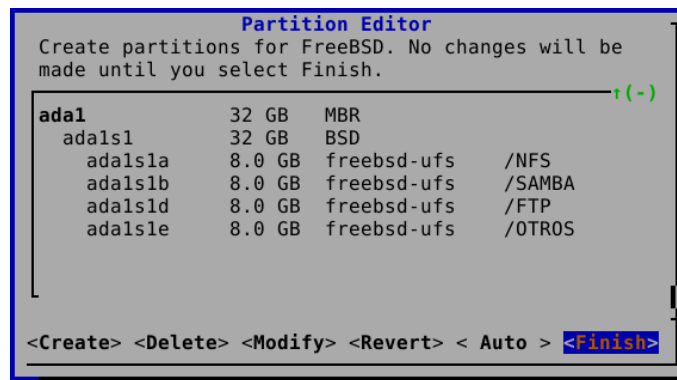
Ip

3 Particionado de Discos

En este apartado explicaré el como y el porque del particionado de los discos y mi elección de particiones para mayor seguridad.

3.1 FREEBSD

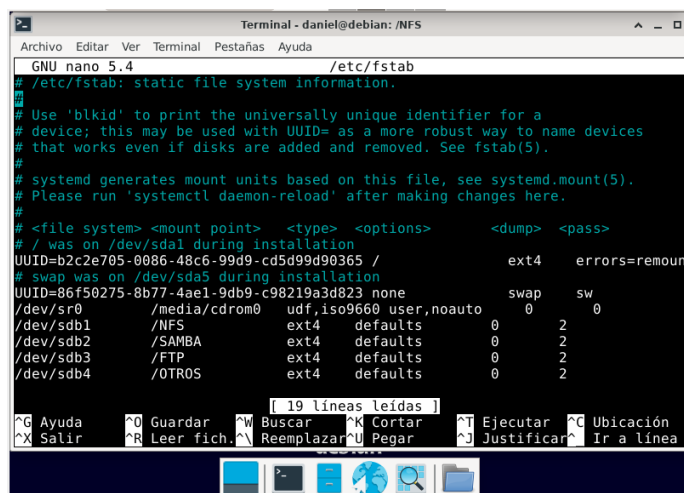
Para el particionado de discos en FreeBSD lo primero que haremos será reinstalar la máquina y instalar el sistema de archivos GPT con sus correspondientes particiones. Una vez esto este hecho el sistema de archivos quedará de la siguiente manera:



Particionado FreeBSD

3.2 Debian

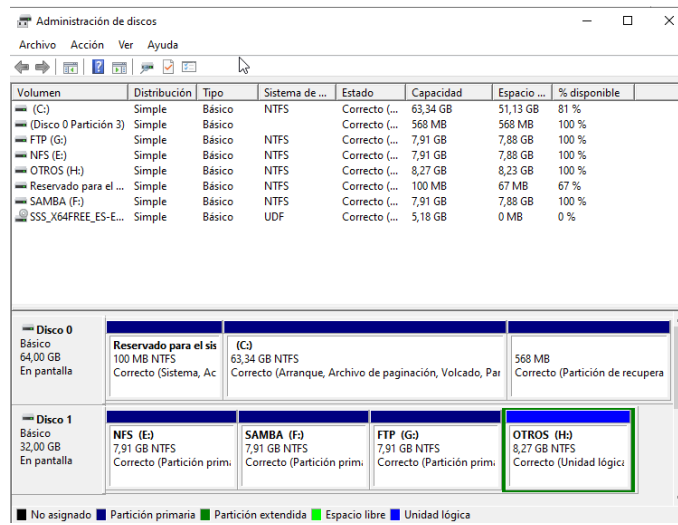
En Debian para el particionado de archivos utilizaremos la herramienta GParted, con ella lo que haremos será particionar el disco, después realizaremos la siguiente configuración en /etc/fstab



Particionado Debian

3.3 Windows 10

El particionado de windows lo realizaremos desde la herramienta de Crear y formatear discos y particiones del propio sistema operativo, quedando las particiones de la siguiente manera:



Particionado Windows10

4 Servicio SSH, SFTP y SCP

4.1 Debian

Lo primero que deberemos realizar en Debian será comprobar que tenemos el servicio instalado, si no fuera el caso deberíamos ejecutar el siguiente comando:

```
sudo apt install openssh-server
```

Una vez hecho esto pasaremos a editar el archivo de configuración del SSH que se encuentra en la siguiente ruta, añadiendo y modificando estas líneas:

```
/etc/ssh/sshd_config
Habilitaremos los Log:
SyslogFacility AUTH
LogLevel INFO

PermitRootLogin no

AllowUsers <usuario>

PermitEmptyPasswords no

ChallengeResponseAuthentication no

UsePam yes
```

Una vez hecho esto en el servidor pasaremos al cliente el cual tendrá que generar y guardar sus claves públicas en el servidor, para ello el cliente hará lo siguiente:

```
Generará la clave:
ssh-keygen -t rsa
La pasará al servidor:
ssh-copy-id -i nombreUsuario@ipServidor
```

Para ver los Log se usará:

```
sudo grep 'sshd' /var/log/auth.log
```

4.2 FreeBSD

Lo primero que deberemos realizar en FreeBSD será comprobar que tenemos el servicio instalado, si no fuera el caso deberíamos ejecutar el siguiente comando: Una vez hecho esto pasaremos a editar el archivo de configuración del SSH que se encuentra en la siguiente ruta, añadiendo y modificando estas líneas:

```
/etc/ssh/sshd_config
Habilitaremos los Log:
SyslogFacility AUTH
LogLevel INFO

PermitRootLogin no

AllowUsers <usuario>

PermitEmptyPasswords no
```

```
ChallengeResponseAuthentication no
```

```
PasswordAuthentication no
```

```
UsePam yes
```

```
/etc/rc.conf
```

```
sshd_enable = "YES"
```

Una vez hecho esto en el servidor pasaremos al cliente el cual tendrá que generar y guardar sus claves públicas en el servidor, para ello el cliente hará lo siguiente:

Generará la clave:

```
ssh-keygen -t rsa
```

La pasará al servidor:

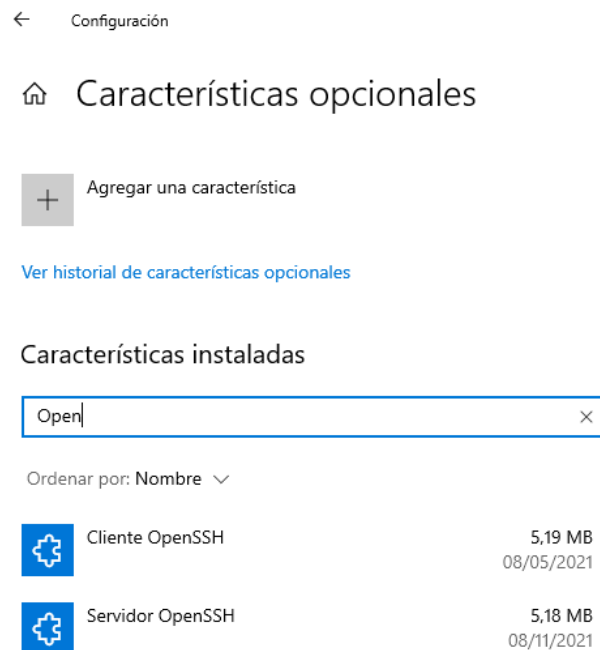
```
ssh-copy-id -i nombreUsuario@ipServidor
```

Para ver los Log se usará:

```
sudo grep 'sshd' /var/log/auth.log
```

4.3 Windows 10

Lo primero que haremos será instalar el servicio Ssh ya que este no viene instalado en Windows por defecto, para ello haremos lo siguiente:



Instalar OpenSsh

Una vez hecho esto en el servidor pasaremos al cliente el cual tendrá que generar y guardar sus claves públicas en el servidor, para ello el cliente hará lo siguiente:

Generará la clave:

```
ssh-keygen -t rsa
```

La pasará al servidor:

```
scp id_rsa.pub administrador@192.168.56.223:%programdata%/ssh
```


Una vez conectados escribiremos lo siguiente:

```
icacls administrators_authorized_keys /inheritance:r /grant "Administrador:F" /grant "SYSTEM:F"
```

```
type id_rsa.pub >> administrators_authorized_keys
```

Modificamos el archivo de configuración:

```
Habilitaremos los Log:  
SyslogFacility AUTH  
LogLevel INFO
```

```
PubkeyAuthentication yes
```

```
AllowUsers <usuario>
```

```
PermitEmptyPasswords no
```

Para ver los Log se usará:

```
%programdata%/ssh/logs
```

4.4 Ejecución

Para ejecutar SSH en todos los sistemas se usa el mismo comando:

```
ssh usuario@ip  
sftp usuario@ip  
  put <archivo>  
scp archivo usuario@ip
```

5 VNC

5.1 FreeBSD

Para la instalacion de VNC en Freebsd solo deberemos realizar lo siguiente.

```
sudo pkg install x11vnc
```

Y para ejecutar el servidor VNC solo deberemos escribir su comando de inicio en la terminal

```
x11vnc
```

Después en un cliente VNC como podría ser VNC viewer escribiremos la ip de nuestro Freebsd en mi caso.

```
192.168.56.221
```

5.2 Debian

Para la instalación de VNC en Debian solo deberemos realizar lo siguiente.

```
sudo apt install x11vnc
```

Y para ejecutar el servidor VNC solo deberemos escribir su comando de inicio en la terminal

```
x11vnc
```

Después en un cliente VNC como podría ser VNC viewer escribiremos la ip de nuestro Debian en mi caso.

```
192.168.56.222
```

Después crearemos el servicio dentro de Debian:

```
x11vnc -storepasswd /etc/x11vnc.pwd
```

Escribiremos la contraseña y la almacenaremos

Después modificaremos el siguiente archivo:

```
vi /etc/systemd/system/x11vnc.service
```

Escribimos:

```
[Unit]
```

```
Description=Start x11vnc at startup.
```

```
After=multi-user.target
```

```
[Service]
```

```
Type=simple
```

```
ExecStart=/usr/bin/x11vnc -auth guess -forever -loop -noxdamage -repeat -rfbauth /etc/x11vnc.pwd -r
```

```
[Install]
```

```
WantedBy=multi-user.target
```

Por ultimo ejecutamos y guardamos el servicio:

```
systemctl enable x11vnc
```

```
systemctl start x11vnc
```

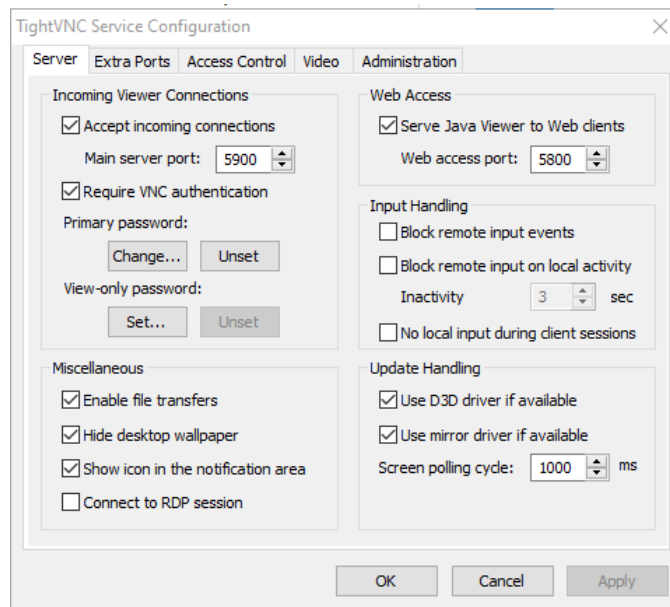
5.3 Windows 10

Para la instalación de VNC en windows 10 descargaremos el software de TightVNC desde su página oficial.

<https://www.tightvnc.com/download.php>

Ejecutaremos la instalación y seleccionaremos una instalación completa para contar con las características de servicio para windows 10.

Una vez hecho esto la configuración de TightVnc será la siguiente:



TightVnc en windows 10

Después en un cliente VNC como podría ser VNC viewer escribiremos la ip de nuestro Windows10 en mi caso.

192.168.56.106

6 XRDP

6.1 Debian

Para el entorno remoto de XRDP necesitaremos realizar la siguiente configuración:

```
sudo apt update

sudo apt install xfce4 xfce4-goodies xorg dbus-x11 x11-xserver-utils

sudo apt install xrdp

sudo systemctl status xrdp

sudo adduser xrdp ssl-cert

sudo systemctl restart xrdp

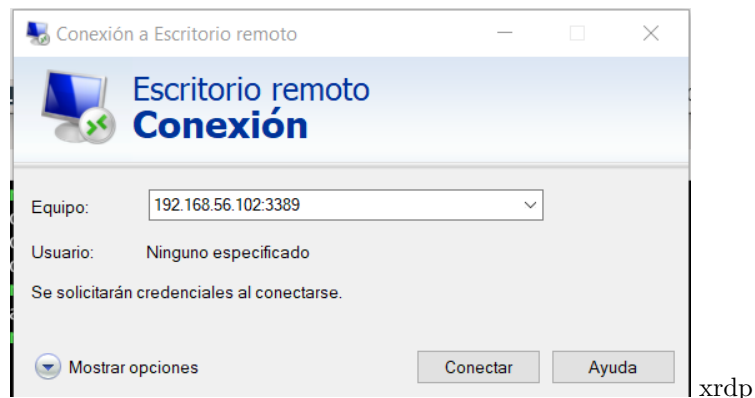
sudo apt install ufw

sudo ufw allow from 192.168.56.0/24 to any port 3389

sudo ufw allow from 192.168.100.0/24 to any port 3389

sudo ufw allow 3389
```

Lo probaremos en el escritorio remoto del host escribiendo lo siguiente:



192.168.56.102 Usuario: xrdp Contraseña: 270301

6.2 FreeBSD

Para el entorno remoto de XRDP necesitaremos realizar la siguiente configuración:

```
sudo apt update

sudo apt install xrdp

Modificamos el archivo /usr/local/etc/xrdp/startwm.sh

#!/bin/sh
#
```

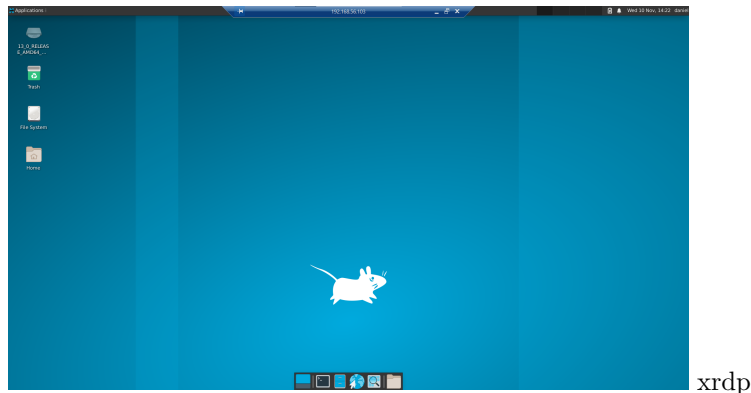
```
# This script is an example. Edit this to suit your needs.
# If ${HOME}/startwm.sh exists, xrdp-sesman will execute it instead of this.

#### set environment variables here if you want
export LANG=en_US.UTF-8

#### start desktop environment
# exec gnome-session
# exec mate-session
# exec start-lumina-desktop
# exec startkde
exec startxfce4
#exec xterm

Ejecutamos:
sysrc xrdp_enable="YES"
sysrc xrdp_sesman_enable="YES"
```

Lo probaremos en el escritorio remoto del host escribiendo lo siguiente:

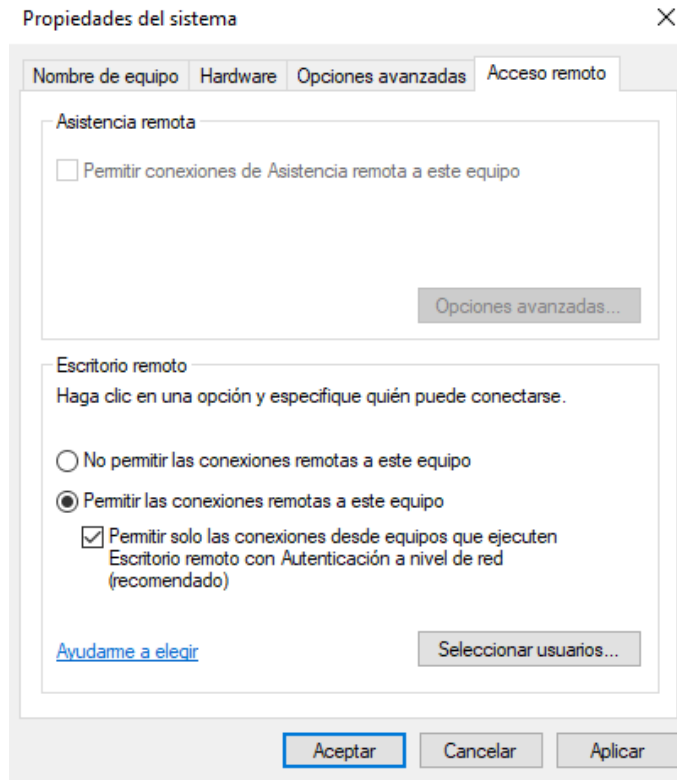


192.168.56.103 Usuario:daniel Contraseña: 270301

6.3 Windows 10

Solo deberemos realizar la siguiente configuracion:

Accedemos a Panel de Control / Sistema y Seguridad / Sistema / Configuración avanzada del sistema / Pestaña acceso remoto / Activamos Permitir las conexiones remotas a este equipo:



Rdp Windows 10

Para loguearse las credenciales son: Administrador 270301Dar*

7 NFS

7.1 Debian

Para instalar el servicio de NFS en debian lo que deberemos realizar será lo siguiente:

```
sudo apt install nfs-server
```

Añadiremos un usuario el cual se encargará de la gestión del directorio

```
sudo adduser nfsnobody
```

```
sudo chown nfsnobody:nfsnobody /NFS
```

```
sudo chmod 777 /NFS
```

Modificamos los archivos de configuración para que se puedan conectar y realizar cambios en los archivos añadiendo la siguiente linea

```
sudo nano /etc/exports
```

```
/NFS *(rw,sync,no_root_squash,no_all_squash,no_subtree_check)
```

Exportamos los cambios:

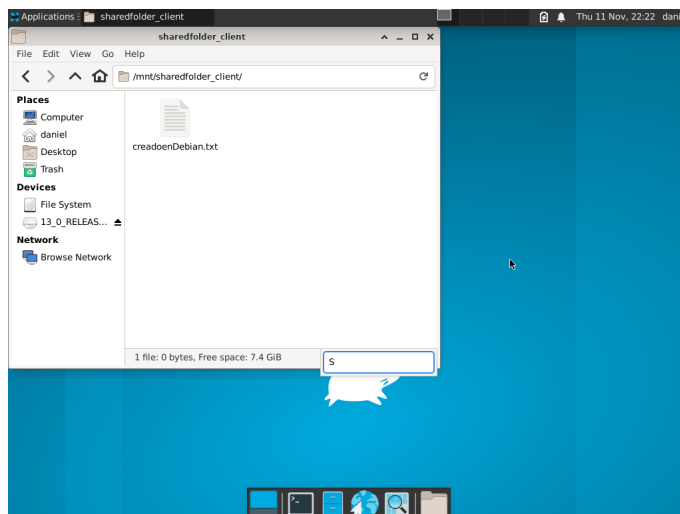
```
sudo exportfs -a
```

```
sudo service nfs-server start
```

Ahora desde el cliente crearemos la carpeta que hira conectada al servidor para ellos realizaremos lo siguiente:

```
sudo mkdir -p /mnt/sharedfolder_client
```

```
sudo mount 192.168.56.222:/NFS /mnt/sharedfolderNFSDebian
```



Carpeta Compartida con Debian

7.2 FreeBSD

El servidor NFS lo instalaremos de la siguiente manera en FreeBSD, lo primero que haremos será configurar la carpeta /etc/rc.conf añadiendo la siguientes lineas:

```
mountd_flags="-r"
rpcbind_enable="YES"
nfs_server_enable="YES"
nfs_server_flags="-u -t -n 4"
portmap_enable="YES"
```

Acto seguido añadiremos un nuevo fichero de configuración en la siguiente ruta:

```
/etc/exports
/NFS -alldirs -maproot=root 192.168.56.221
```

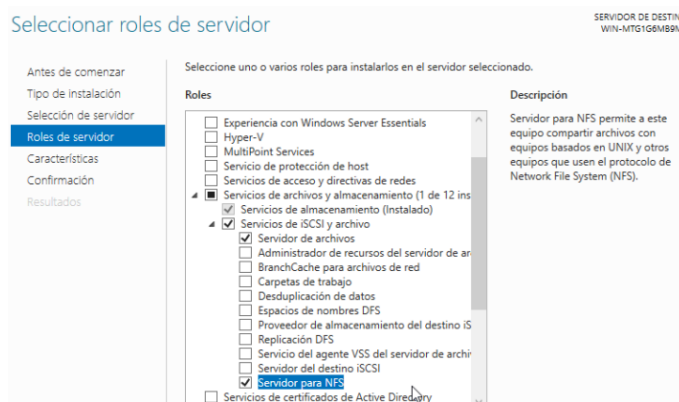
```
Reiniciamos el servicio para aplicar los cambios:
service nfsd restart
```

Desde el cliente realizaremos la siguiente configuración para poder añadir nuestra carpeta remota:

```
sudo mkdir /NFS-servidor-Free
mount -t nfs 192.168.56.221:/NFS /NFS-servidor-Free
```

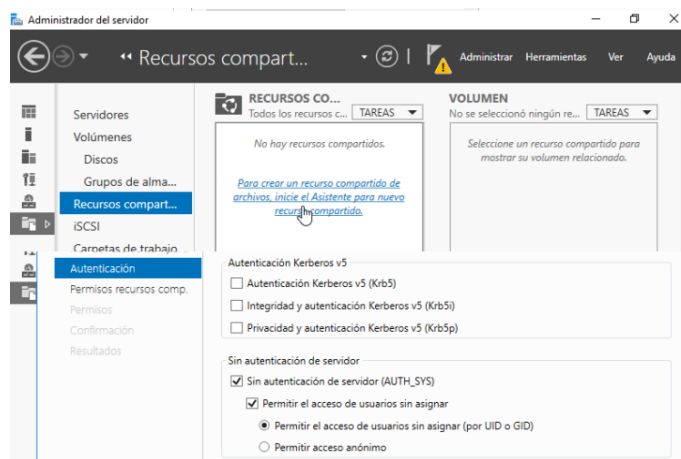
7.3 Windows 10

Para utilizar NFS desde windows le daremos un Rol nuevo a nuestro servidor para ello entraremos en la configuración del servidor y realizaremos las siguiente configuraciones:



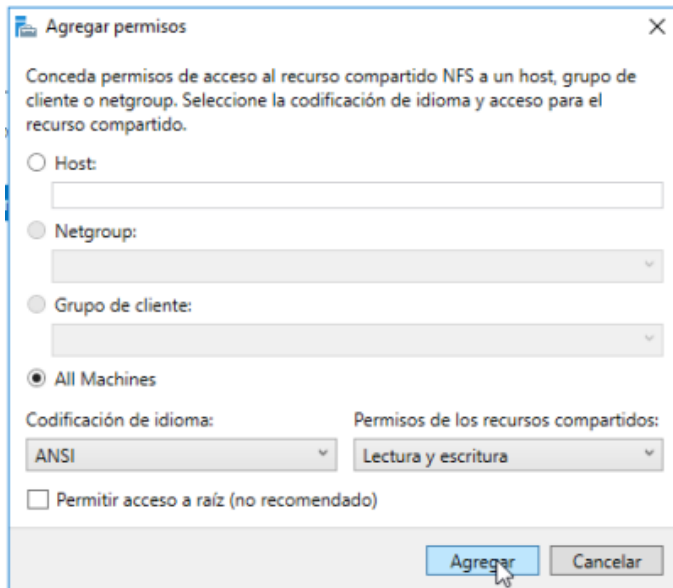
Seleccionamos el Rol de NFS

Seleccionamos los permisos y la carpeta donde irá nuestro servidor:



Selección de carpeta y permisos

Montamos el servidor y adjudicamos los permisos de lectura y escritura:



Montar y Permisos de lectura y escritura

En el cliente deberemos realizar lo siguiente para poder compartir archivos:

```
sudo mount -O vers=3 192.168.56.223:/nfs /mnt/sharedFolderWindows/
```

8 SAMBA

8.1 Debian

Para instalar el sistema de ficheros compartidos de Samba en Debian deberemos realizar los siguientes comandos:

```
sudo apt install samba samba-client samba-common
```

Modificamos el archivo de configuración de Samba alojado en `/etc/samba/smb.conf`, añadiendo las siguientes líneas:

```
[samba]
comment = Directorio compartido
Path = /SAMBA
browseable = yes
writeable = yes
guest ok = yes
```

Le damos los permisos a la carpeta de SAMBA y añadimos un nuevo usuario para samba:

```
chmod -R 777 /SAMBA
sudo adduser samba
```

Reiniciamos el servicio:

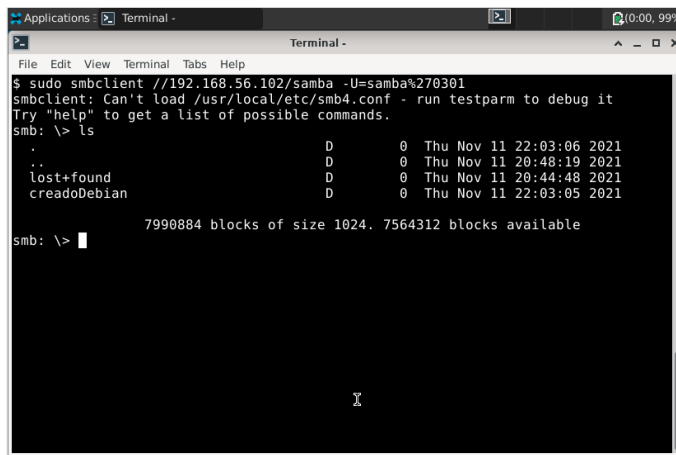
```
systemctl restart smbd
```

Le adjudicamos un contraseña:

```
sudo smbpasswd -a samba
```

Desde el cliente ejecutaremos lo siguiente para poder conectarnos:

```
smbclient //192.168.56.222/samba -U=samba%270301
```



Samba Debian de FreeBSD

8.2 FreeBSD

Para la instalación de SAMBA en FreeBSD lo primero que haremos será cambiar la carpeta `/etc/rc.conf` añadiendo las siguientes líneas:

```
samba_server_enable="yes"
```

Después instalaremos el paquete samba para poder utilizar los comandos y añadir un nuevo usuario al que llamaremos "sambauser" y añadiremos a un nuevo grupo "sambagroup":

```
sudo pkg install samba413
```

Adjudicamos una nueva contraseña al servidor de samba
`smbpasswd -a sambauser`

Añadimos la siguiente configuración a la carpeta `/usr/etc/local/smb4.conf`

```
[global]
netbios name = SERVERFREESAMBA
workgroup = WORKGROUP
host allow = 192.168.56.221 192.168.56.222 localhost
server string = samba
[samba]
comment = Directorio compartido
Path = /SAMBA/SAMBA
browseable = yes
writeable = yes
guest ok = yes
```

Creamos carpeta y adjudicamos permisos:

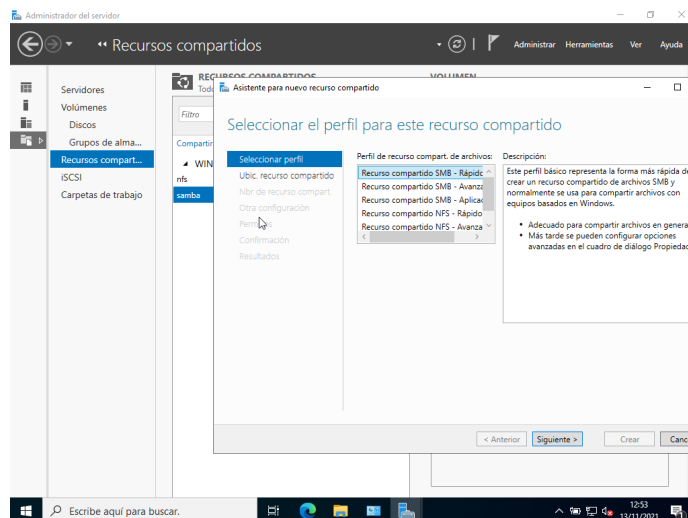
```
sudo chow sambauser:sambagroup /SAMBA/SAMBA/
sudo chmod 770 /SAMBA/SAMBA/
```

Desde el cliente haremos lo siguiente para conectarnos desde Debian:

```
sudo smbclient //192.168.56.221/SAMBA/SAMBA -U=sambauser%270301
```

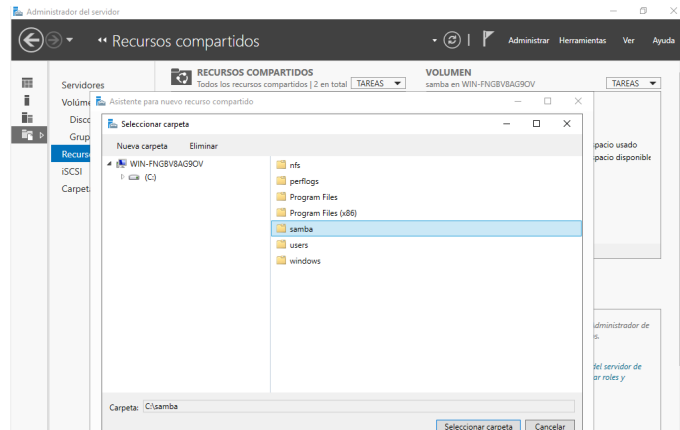
8.3 Windows 10

Para la configuración de Samba en windows 10 deberemos realizar la siguiente configuración del servidor, para ello previamente deberemos tener instalado las características de SAMBA es decir SMB:



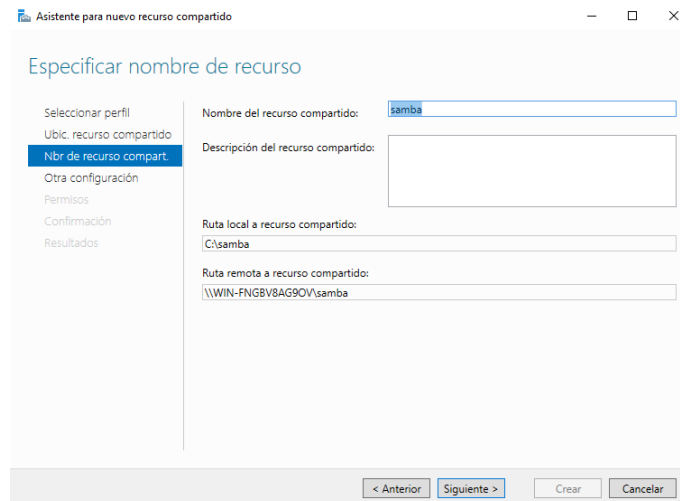
Creación del recurso compartido

Seleccionaremos la carpeta donde tendremos nuestro recurso compartido:



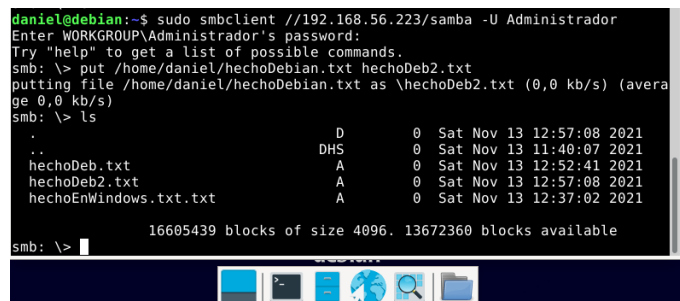
Creación del recurso compartido

Le adjudicaremos un nombre a nuestro recurso compartido para que pueda ser detectado por los equipos de nuestra red:



Creación del recurso compartido

Desde el cliente probaremos que detectamos la carpeta y que podemos coger y subir archivos de la misma



Creación del recurso compartido

Comando que deberemos poner en el cliente para conectarnos al servidor:

```
sudo smbclient //192.168.56.223/samba -U Administrador
```

9 Cups servidor de impresión

9.1 FreeBSD

Para instalar Cups en FreeBSD realizaremos los siguientes pasos:

```
pkg install cups cups-pdf cups-filters
```

Crearemos el archivo `/etc/devfs.rules` para guardar nuestra configuración de impresión:

```
[system=10]
add path 'unlpt*' mode 0660 group cups
add path 'ulpt*' mode 0660 group cups
add path 'lpt*' mode 0660 group cups
add path 'usb/X.Y.Z' mode 0660 group cups
```

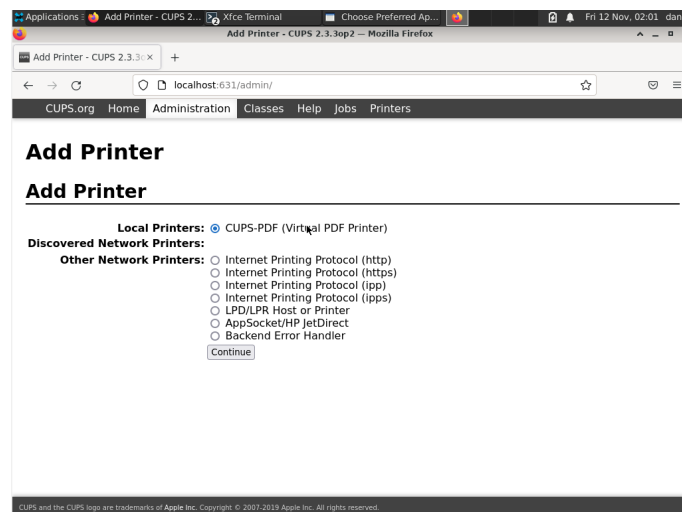
Modificaremos el archivo `/etc/rc.conf` para activar el servicio:

```
cupsd_enable="YES"
devfs_system_ruleset="system"
```

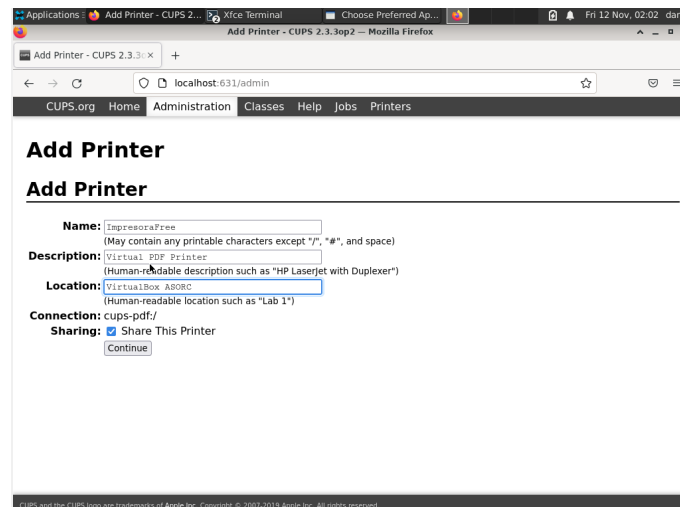
Reiniciamos los servicios:

```
service devfs restart
service cupsd restart
```

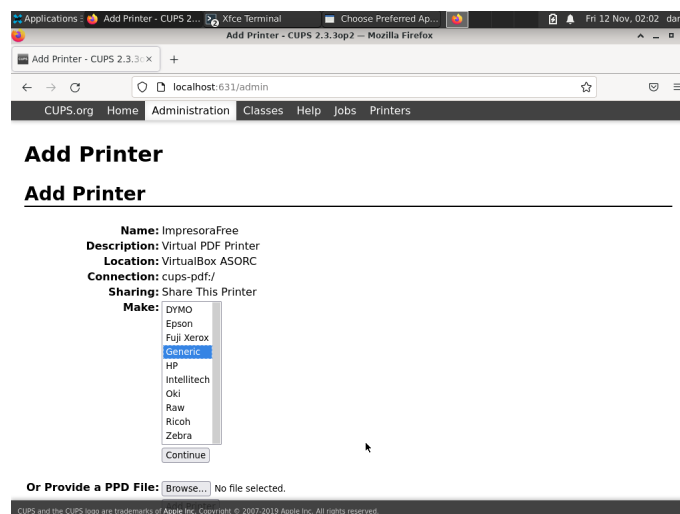
Accedemos desde nuestro buscador a la siguiente ruta para configurar una nueva impresora: <http://localhost:631>
Y configuramos lo siguiente:



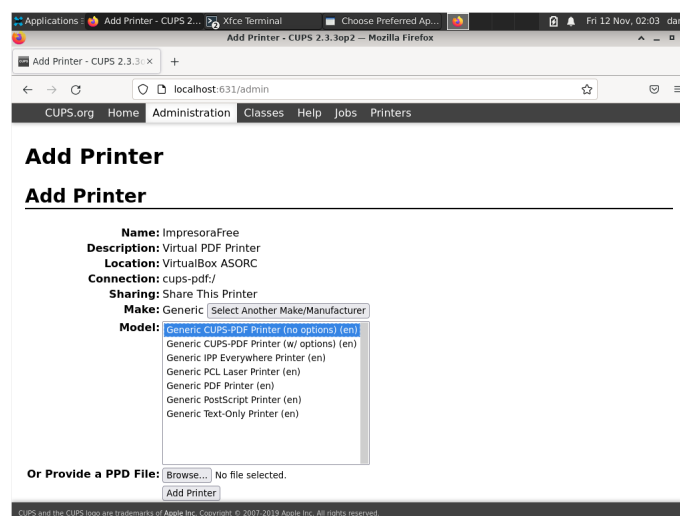
Cups configuración



Cups configuración

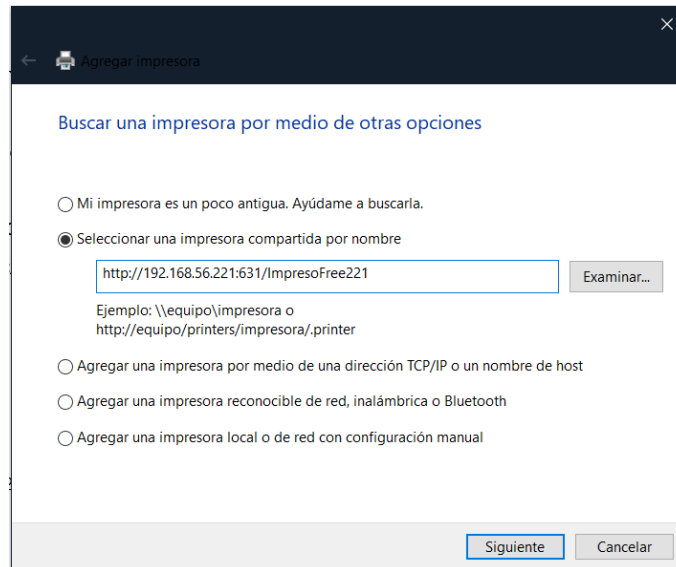


Cups configuración



Cups configuración

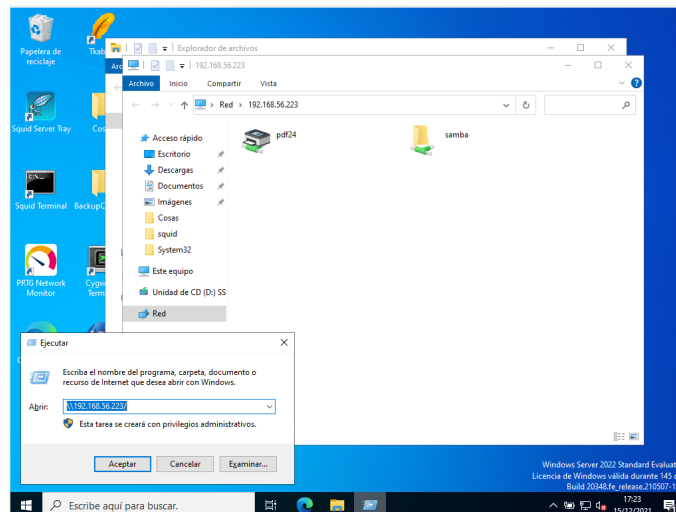
Para conectarnos por ejemplo desde nuestro Host realizaremos lo siguiente:



Cups conexión

9.2 Windows 10

Agregamos los roles de servidor de impresión en la administración del servidor un vez hecho esto, instalaremos pdf24, iremos a configuración de las impresoras y click derecho propiedades y compartir impresora, ahora desde el cliente tiene que ser un windows 10 server hacemos los siguiente:



Cups conexión

Ahora directamente podremos imprimir directamente en esa impresora.

9.3 Debian

Para instalar Cups en Debian realizaremos los siguientes pasos:

```
apt install cups cups-pdf cups-filters
```

Crearemos el archivo `/etc/devfs.rules` para guardar nuestra configuración de impresión:

```
[system=10]
add path 'unlpt*' mode 0660 group cups
add path 'ulpt*' mode 0660 group cups
add path 'lpt*' mode 0660 group cups
add path 'usb/X.Y.Z' mode 0660 group cups
```

Modificaremos el archivo `/etc/cups/cupsd.conf` para activar el servicio:

```
<Location />
  Order allow,deny
allow localhost
allow 192.168.56.0/24
</Location>
<Location /admin>
  Order allow,deny
allow localhost
allow 192.168.56.0/24
allow 192.168.0.0/24
</Location>
```

Reiniciamos los servicios:

```
service devfs restart
service cupsd restart
```

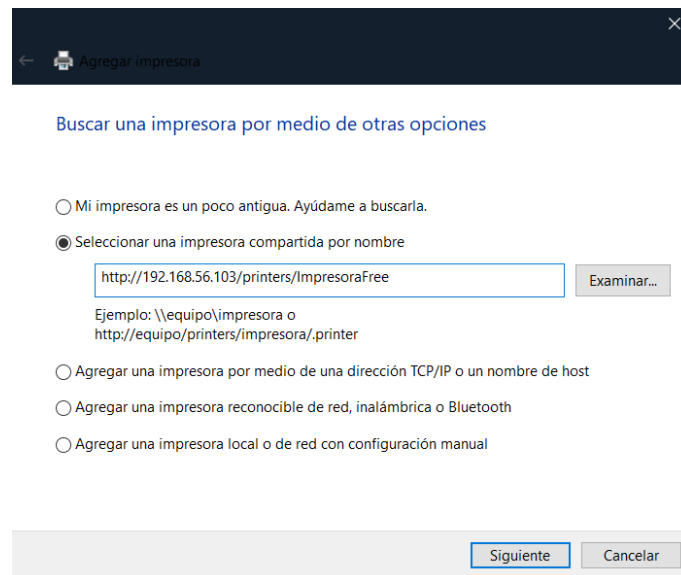
```
nano /etc/cups/cups-pdf-conf
```

```
Out ${HOME}
```

```
systemctl enable cups && systemctl restart cups
```

Accedemos desde nuestro buscador a la siguiente ruta para configurar una nueva impresora: <http://localhost:631>
Y configuramos igual que en FreeBSD

Para conectarnos por ejemplo desde nuestro Host realizaremos lo siguiente:



Cups conexión

Para ver que a funcionado todo si hemos realizado la impresión desde un client windows deberemos ir a la siguiente ruta:

```
/var/spool/cups-pdf/ANONYMOUS/
```

10 Base de datos y Web

10.1 Debian con MariaDb

Para la instalación y configuración de MariaDb y prueba con una página escrita en php he realizado la siguiente instalación y configuración:

```
sudo apt -y update
sudo apt -y install software-properties-common gnupg2
sudo apt -y upgrade
```

```
sudo apt-key adv --recv-keys --keyserver keyserver.ubuntu.com 0xF1656F24C74CD1D8
```

```
sudo add-apt-repository "deb [arch=amd64] http://mariadb.mirror.liquidtelecom.com/repo/10.5/debian "
```

```
sudo apt update
sudo apt install mariadb-server mariadb-client
```

```
sudo mysql_secure_installation
```

```
sudo mariadb
```

```
mysql -u miusuarioAdmin -p
```

```
sudo apt install php libapache2-mod-php php-mysql
```

```
sudo service apache2 restart
```

Iremos a la siguiente carpeta para poder realizar llamadas a MariaDb de forma remota:

```
sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

```
comentamos la linea:
#bind-address = 127.0.0.1
```

Utilizaremos el siguiente script en php para sacar los datos:

```
<?php

$hostname = "192.168.56.222";
$username = "daniel";
$password = "270301";
$db = "asorc";

$dbconnect=mysqli_connect($hostname,$username,$password,$db);

if ($dbconnect->connect_error) {
    die("Database connection failed: " . $dbconnect->connect_error);
}

?>

<table border="1" align="center">
<tr>
    <td>Reviewer Name</td>
    <td>Stars</td>
    <td>Details</td>
</tr>

<?php

$query = mysqli_query($dbconnect, "SELECT * FROM alumnos")
    or die (mysqli_error($dbconnect));

while ($row = mysqli_fetch_array($query)) {
    echo
    "<tr>
        <td>{$row['id']}</td>
        <td>{$row['nombre']}</td>
        <td>{$row['apellido']}</td>
    </tr>\n";
}

?>
```

Para ver la página web escribiremos en el buscador: <http://192.168.56.222/testdb.php>
Tendremos la siguiente salida

Reviewer Name	Stars	Details
1	daniel	asensi

MariaDb conexión

10.2 FreeBSD y Postgres

Para la instalación de Postgresql en FreeBSD deberemos realizar los siguiente pasos con las siguientes configuraciones:

```
sudo pkg install postgresql11-server postgresql11-client

sudo sysrc postgresql_enable=yes

/usr/local/etc/rc.d/postgresql initdb

/usr/local/etc/rc.d/postgresql start

sudo vim /var/db/postgres/data11/postgresql.conf
listen_addresses = '*'

service postgresql restart

passwd postgres

su - postgres

psql -c "alter user postgres with password 'StrongPassword'"

//Creamos las tablas que Deseemos
```

Pasamos a instalar Apache24 y Php para nuestro servidor web

```
pkg install php74
//Para saber nuestra configuración
echo '<?php phpinfo(); ?>' > /usr/local/www/apache24/data/info.php
pkg install mod_php74

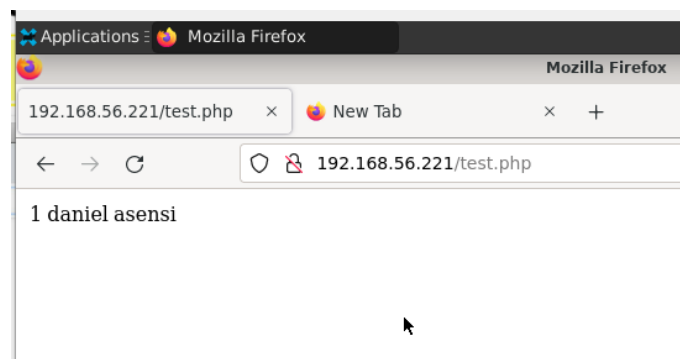
vim /usr/local/etc/apache24/Includes/php.conf
<IfModule dir_module>
DirectoryIndex index.php index.html
<FilesMatch "\.php$">
    SetHandler application/x-httpd-php
</FilesMatch>
<FilesMatch "\.phps$">
```

```
    SetHandler application/x-httpd-php-source
</FilesMatch>
</IfModule>
```

```
service apache2 restart
```

```
http://192.168.56.221/info.php
//Insalamos apache
sudo pkg install -y apache24
service apache24 onestatus
sudo service apache24 enable
sudo service apache24 start
```

Para acceder a nuestra pagina con la BD deberemos: www.webdaniel.com/test.php Se verá de la siguiente manera:

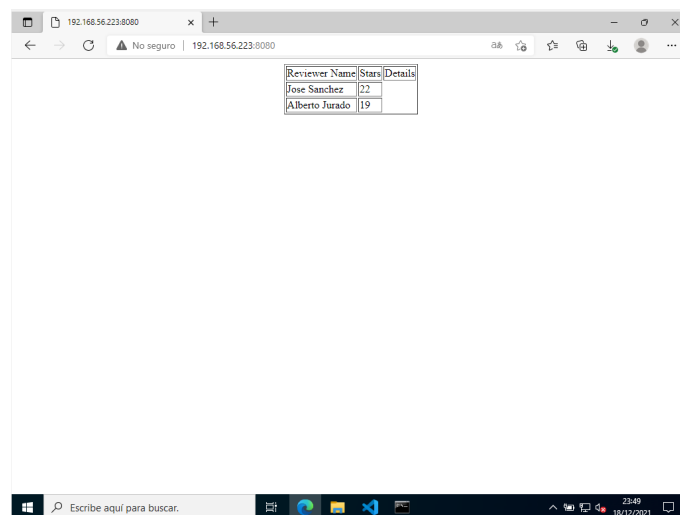


PostgreSql

10.3 Windows

Para el web server de windows instalaremos php desde su pagina oficial y crearemos un script, hecho esto desde mysql generaremos una tabla y este será el display tras ejecutar:

```
php -S 192.168.56.223:8080 212.php
```

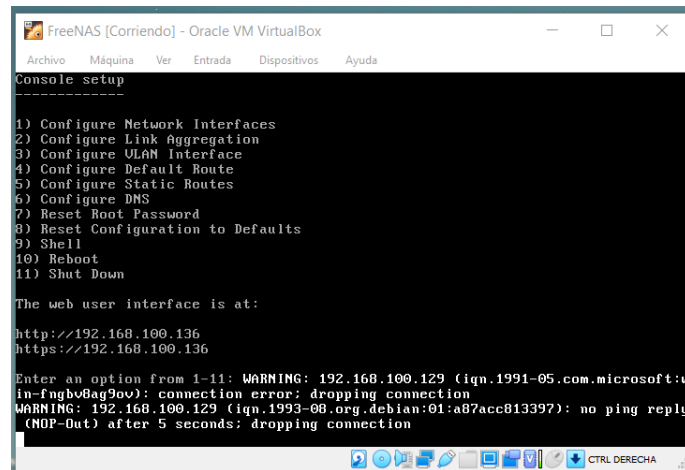


BDSERVER

11 TrueNAS + ISCSI

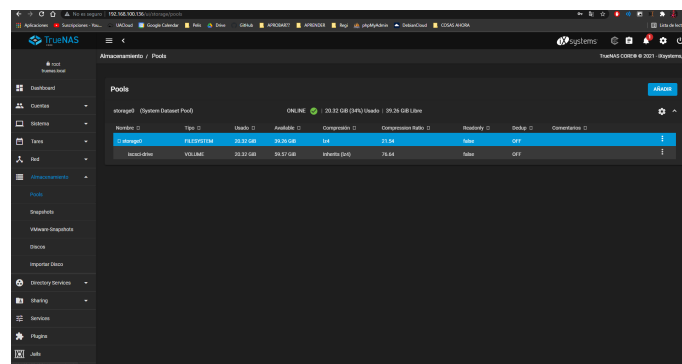
11.1 Instalación Máquina virtual

Lo primero que haremos para la instalación de TrueNas con ISCSI será instalar desde la iso de la página oficial en nuestro virtualbox una máquina virtual donde instalaremos los discos y a la que adjudicaremos una Ip para conectarnos, en mi caso para simplificar las cosas he usado el Adaptador Bridge ya que este no se prohíbe en la práctica, una vez realizada la instalación nos aparecerá una ventana como esta:



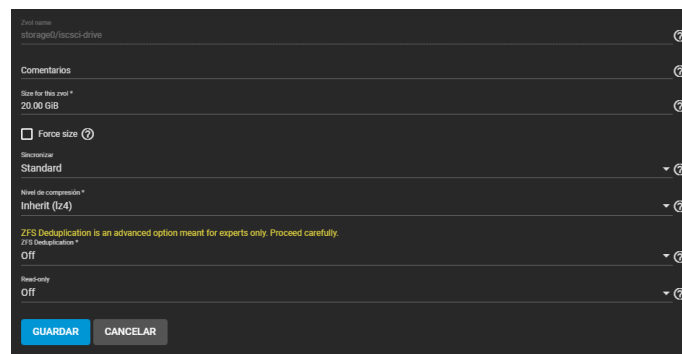
Ventana de Terminal

Desde esta ventana pondremos en el buscador nuestra Ip de TrueNas y nos loguearemos como Root:



Truenas Configuración

Añadiremos los discos duros y activaremos ISCSI para adjudicar a nuestros discos ZVOL donde procederemos con la configuración:



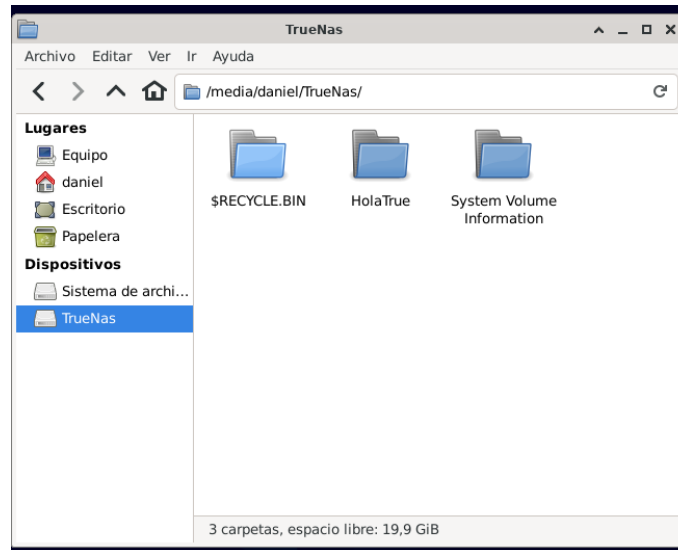
Ventana de Terminal

11.2 Debian

Para la configuración en debian de este servicio solo nos harán falta dos comandos:

```
sudo iscsiadm -m discovery -t st -p 192.168.56.108
```

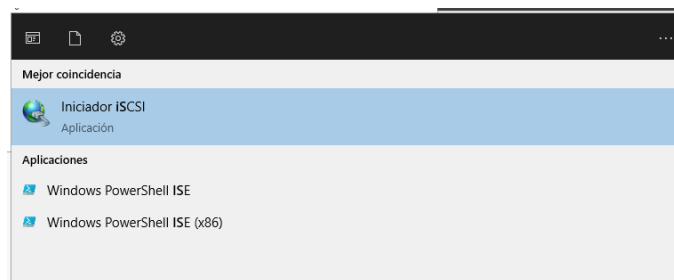
```
sudo iscsiadm -m node --targetname "iqn.2005-10.org.freenas.ctl:iscsi-drive-windows" --portal  
"192.168.56.108:3260" --login
```



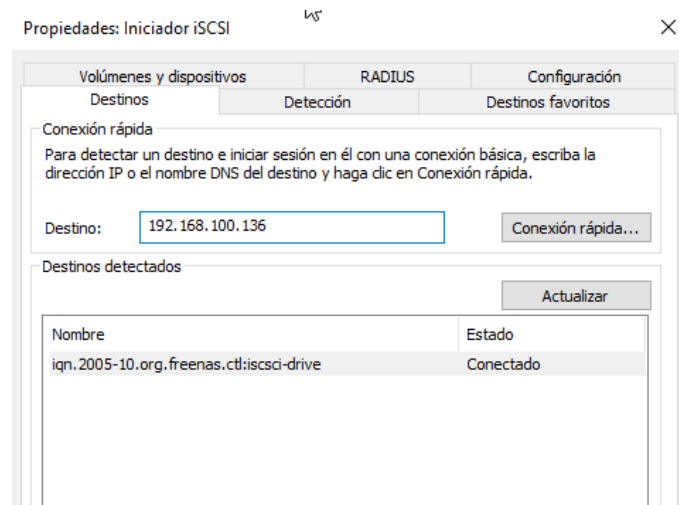
Ventana de Terminal

11.3 Windows 10

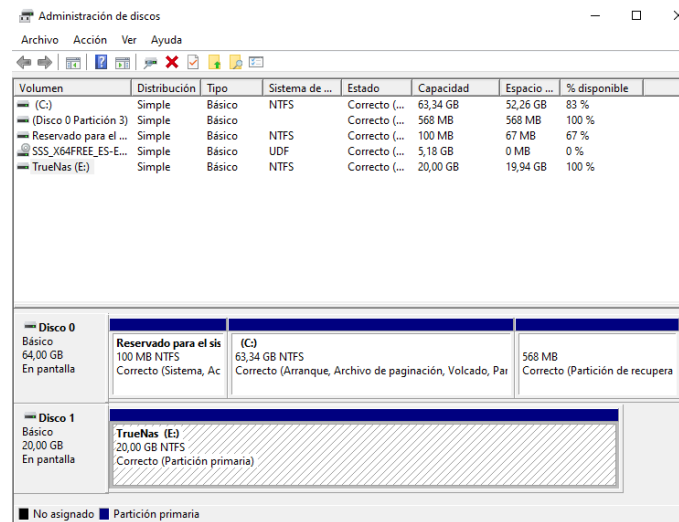
Para la configuración de TrueNas en windows seguiremos los siguientes pasos:



ISCSI Initiator



Selección del destino



Configuración del disco

12 OwnCloud + Git

12.1 Debian

12.1.1 Owncloud

Para la instalación de owncloud en nuestra máquina debian los primero que deberemos hacer será descargar Mysql con MariaDb y php en su última versión.

```
sudo apt-get install apache2 mariadb-server mariadb-client
```

```
sudo apt-get install php7.4 libapache2-mod-php7.4 php7.4-curl
php7.4-intl php7.4-json php7.4-gd php7.4-mbstring php7.4-mysql
php7.4-xml php7.4-zip
```

```
mysql -u root -p
MariaDB [(none)]> CREATE DATABASE owncloud;
MariaDB [(none)]> GRANT ALL ON ownclouddb.* to danielcloud@localhost IDENTIFIED BY '270301';
MariaDB [(none)]> FLUSH PRIVILEGES;
MariaDB [(none)]> EXIT;
```

Descargamos la ultima versión de OwnCloud:

```
sudo wget https://download.owncloud.org/community/owncloud-latest.tar.bz2
```

```
sudo mkdir /var/www/owncloud
```

```
sudo tar -xvf owncloud-latest.tar.bz2 --directory /var/www
```

```
sudo chown -R www-data:www-data /var/www/owncloud
```

```
sudo nano /etc/apache2/sites-available/owncloud.conf
Alias /owncloud "/var/www/owncloud/"
<Directory /var/www/owncloud/>
Options +FollowSymlinks
AllowOverride All
```

```
<IfModule mod_dav.c>
Dav off
</IfModule>
```

```
SetEnv HOME /var/www/owncloud
SetEnv HTTP_HOME /var/www/owncloud
</Directory>
```

```
sudo a2ensite owncloud
```

```
sudo a2enmod rewrite headers env mime unique_id dav
```

```
sudo systemctl restart apache2
```

Abrimos en nuestro buscador y buscamos los siguiente: 192.168.56.222/owncloud Nos logueamos con las credenciales que pusimos antes: Si saltará un error de MySql deberemos hacer lo siguiente

```
mysql -u root -p
SET GLOBAL innodb_read_only_compressed=OFF;
```


12.1.2 Git

El sistema de control de versiones de git se instalará de la siguiente manera en Debian y configuraremos su acceso con los siguientes pasos: Instalacion:

```
sudo apt install git
sudo mkdir /home/daniel/git
```

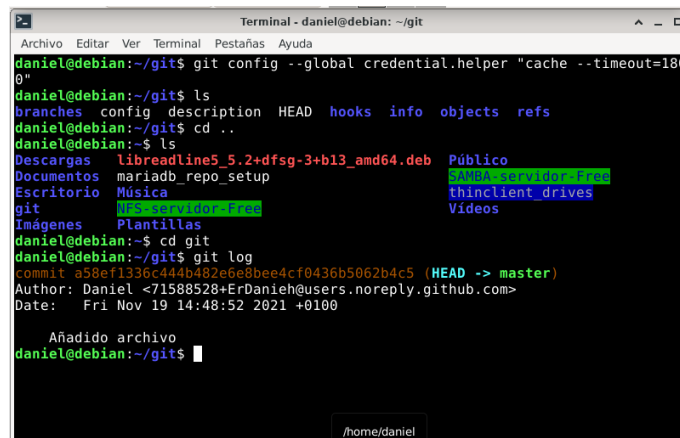
Inicializamos un repositorio de git vacio y lo configuramos con nuestro nombre y email:

```
git init --bare
git config --global user.name "Daniel"
git config --global user.email "danielasesi444@outlook.com"
git config --global credential.helper "cache --timeout=18000"
```

Desde el cliente en este caso será mi host escribimos en la gitbash:

```
git init
$ git commit -m "Añadido"
$ nano holaDeb.txt
$ git commit -m "Añadido archivo"
$ git remote add origin daniel@192.168.56.222:/home/daniel/git
$ git push origin master
```

Si desde el servidor hacemos un git log obtenemos lo siguiente:

A screenshot of a terminal window titled 'Terminal - daniel@debian: ~/git'. The terminal shows the following commands and output:
1. `git config --global credential.helper "cache --timeout=18000"`
2. `ls` output: `branches config description HEAD hooks info objects refs`
3. `cd ..`
4. `ls` output: `Descargas Libreadline5_5.2+dfsg-3+b13_amd64.deb Público`
`Documentos mariadb_repo_setup`
`Escritorio Música`
`git`
`Imágenes Plantillas`
5. `cd git`
6. `git log` output: `commit a58ef1336c444b482e6e8bee4cf0436b5062b4c5 (HEAD -> master)`
`Author: Daniel <71588528+ErDaniel@users.noreply.github.com>`
`Date: Fri Nov 19 14:48:52 2021 +0100`
`Añadido archivo`
The terminal prompt is `daniel@debian:~/git$`.
At the bottom of the terminal window, there is a small box containing the text `/home/daniel`.

Configuración del disco

13 DNS

13.1 FreeBSD

Lo primero que deberemos hacer para configurar nuestra DNS en freebsd será instalar la herramienta:

```
pkg install bind916
sysrc named_enable=yes
```

Configuramos nuestras zonas DNS de la siguiente manera:

```
nano /usr/local/etc/namedb/named.conf

listen-on      { 127.0.0.1;192.168.56.221; };

forwarders {
    8.8.8.8;8.8.4.4;
};

zone "danielasensi.com" {
    type master;
    file "/usr/local/etc/namedb/master/danielasensi.db";
    allow-query{any;};
};

nano /usr/local/etc/namedb/master/danielasensi.db

$TTL 3h
@ IN SOA danielasensi.com. webmaster.danielasensi.com. 42 1d 12h 1w 3h
    ; Serial, Refresh, Retry, Expire, Neg. cache TTL

                IN      NS      danielasensi.com.
                IN      A       192.168.56.221
ms-sv-05        IN      A       192.168.56.221
dns             IN      CNAME    ms-sv-05.danielasensi.com.
www             IN      CNAME    danielasensi.com.
```

```
daniel@:/usr/local/etc/namedb/master $ nslookup www.danielasensi.com
Server:      192.168.56.221
Address:     192.168.56.221#53

www.danielasensi.com canonical name = danielasensi.com.
Name:       danielasensi.com
Address:    192.168.56.221

daniel@:/usr/local/etc/namedb/master $ |
```

Prueba

Para la dns inversa utilizaremos el archivo reverse con la siguiente configuración:

[illegible]

Y en el archivo named.conf

```
/* Example of a slave reverse zone
*/
zone "56.168.192.in-addr.arpa" {
    type master;
    file "/usr/local/etc/namedb/master/reverse.db";
};
```

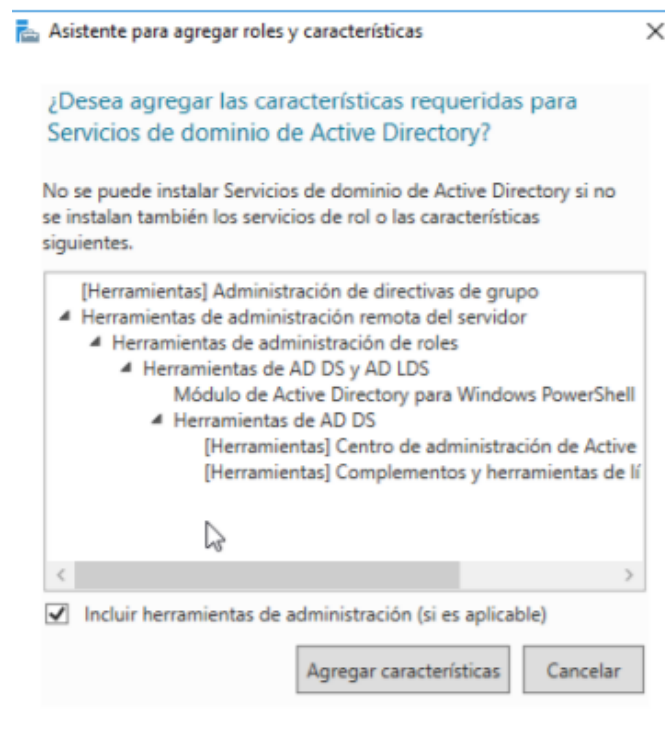
Ahora la probaremos:

```
root@:/usr/local/etc/namedb # nslookup 192.168.56.221
221.56.168.192.in-addr.arpa      name = 192.168.56.221.56.168.192.in-addr.arpa.
```

Prueba

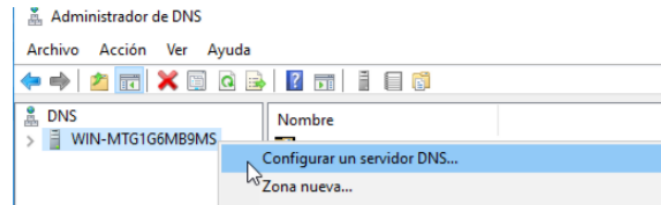
13.2 Windows

Accedemos a Administrador del servidor → Agregar roles y características → Instalación basada en características y roles → Seleccionar un servidor del grupo de servidores (habiendo seleccionado el nuestro propio) → Seleccionamos el checkbox de Servidor DNS y seguimos las instrucciones de instalación.



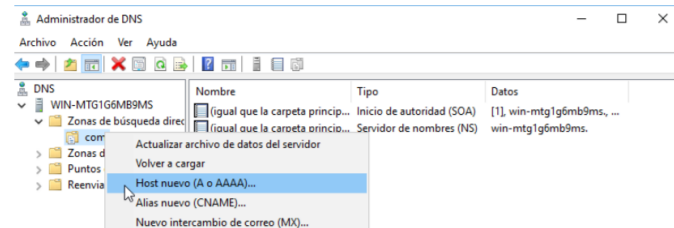
FreeBSD DNS funcionando

Desde el Administrador del servidor, nos vamos a Herramientas → DNS. Hacemos click derecho a nuestro servidor y entramos en Configurar un servidor DNS. Seleccionamos las opciones Crear una zona de búsqueda directa → Este servidor mantiene la zona → “com” como Nombre de la zona → “com.dns” como nombre de archivo → No admitir actualizaciones dinámicas → Agregamos los forwarders (193.145.233.5 DNS UA) • Creamos también una zona de búsqueda inversa. Todo siguiente, siguiente. Cuando nos pregunte por una IP, ponemos la nuestra.



Windows DNS Instalar

- Desde la vista de árbol de nuestro servidor, accedemos a la zona de búsqueda que acabamos de crear y con botón derecho seleccionamos Host nuevo:



Windows DNS Instalar

Desde allí, introducimos los datos necesarios para nuestro dominio a registrar. En mi caso danielasensidns.com

Para comprobar que funciona:

```
C:\Users\Administrador>nslookup srv.danielasensidns.com 192.168.56.223
Servidor: Unknown
Address: 192.168.56.223

Nombre: srv.danielasensidns.com
Address: 192.168.56.223

C:\Users\Administrador>ping srv.danielasensidns.com

Haciendo ping a srv.danielasensidns.com [192.168.56.223] con 32 bytes de datos:
Respuesta desde 192.168.56.223: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.56.223: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.56.223: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.56.223: bytes=32 tiempo<1m TTL=128

Estadísticas de ping para 192.168.56.223:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
              (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 0ms, Media = 0ms

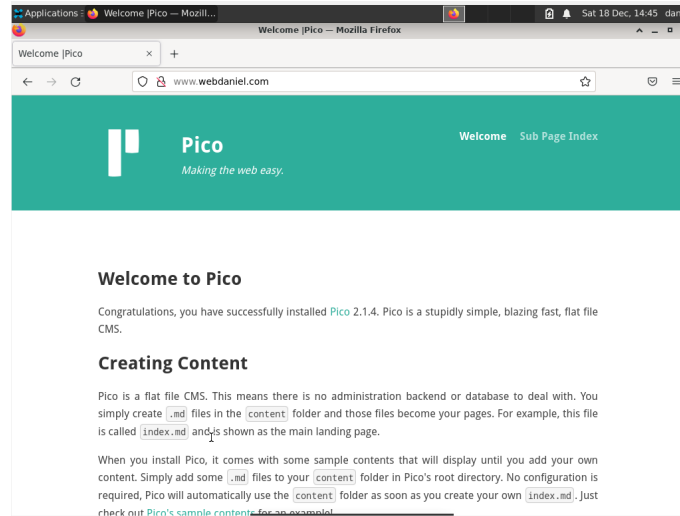
C:\Users\Administrador>
```

Windows DNS Instalar

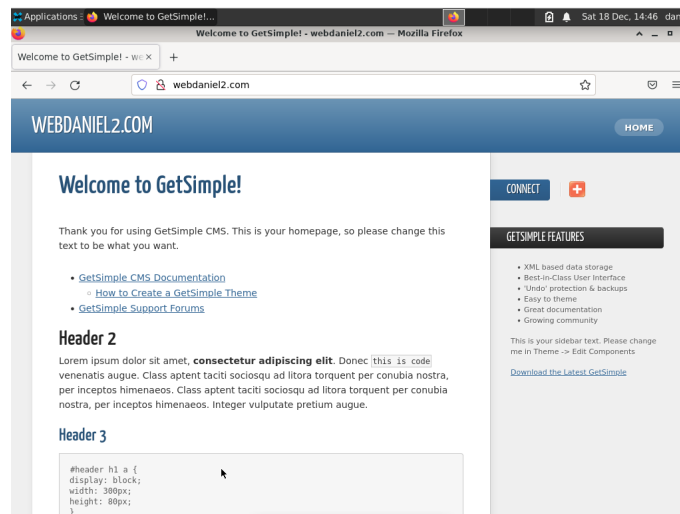
14 Web-Server

14.1 Freebsd

No voy a mostrar el proceso de instalación ya que este es tedioso y ya he explicado como instalar apache y php, solo crearemos nuevas dns para nuestros dominios como he hecho en el apartado anterior y meteremos las páginas en httpd.conf una vez hecho esto instalamos los modulos que pida php y los cms que queramos hecho esto vamos al navegador y ponemos las páginas web:



Web 1



Web 2