Relatório Projeto AS II

Tiago Carvalho nº2024180

16 de abril de 2025

1ºCibersegurança

Índice

Introdução

O presente trabalho prático teve como objetivo a implementação e configuração de um servidor Linux multifuncional, desempenhando o papel de servidor DNS, DHCP, FTP, SSH e ficheiros (Samba), com controlo de utilizadores, permissões e partilhas de rede. A atividade foi desenvolvida num ambiente virtualizado, com vista à simulação de um cenário empresarial, onde o servidor gere e fornece serviços a clientes numa rede interna. Através deste projeto, foi possível aplicar conhecimentos essenciais de administração de sistemas, nomeadamente em configuração de redes, segurança, gestão de utilizadores e automatização de tarefas via cron. O desafio proposto permitiu consolidar competências práticas essenciais na área de Sistemas de Informação.

Servidor Ubuntu

Passo 1 - Placas de Rede

O primeiro passo é configurar as placas de rede, para isso, começamos por instalar o net-tools (lembrando de dar apt update antes de qualquer instalação importante).

```
root@srv1:/home# apt install net-tools
```

Depois, abrimos o ficheiro de configuração das interfaces.

```
root@srv1:/home# nano /etc/network/interfaces
```

Configuramos a nossa placa de rede nat para utilizar dhcp e configuramos a placa de rede interna, definindo o seu IP e netmask.

```
  GNU nano 4.8                  /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d
# Interface enp0s3 - Configuração via DHCP
auto enp0s3
iface enp0s3 inet dhcp

# Interface enp0s8 - Configuração com IP fixo na rede 192.168.X.0/24
auto enp0s8
iface enp0s8 inet static
    address 192.168.31.1
    netmask 255.255.255.0
```

Depois disso, damos restart usando o seguinte comando, seguido de um comando parecido para verificar se o serviço está a correr.

```
tiago@srv1:~$ sudo systemctl restart networking
[sudo] password for tiago:
tiago@srv1:~$ sudo systemctl status networking
● networking.service - Raise network interfaces
     Loaded: loaded (/lib/systemd/system/networking.service; enabled; vendor p>
     Active: active (exited) since Wed 2025-04-16 19:01:10 UTC; 8s ago
       Docs: man:interfaces(5)
    Process: 2415 ExecStart=/sbin/ifup -a --read-environment (code=exited, sta>
   Main PID: 2415 (code=exited, status=0/SUCCESS)
      Tasks: 4 (limit: 3419)
     Memory: 2.5M
     CGroup: /system.slice/networking.service
             └─2440 /sbin/dhclient -1 -4 -v -i -pf /run/dhclient.enp0s3.pid -l>

abr 16 19:01:10 srv1 ifup[2440]: DHCPOFFER of 10.0.2.15 from 10.0.2.2
abr 16 19:01:10 srv1 ifup[2440]: DHCPREQUEST for 10.0.2.15 on enp0s3 to 255.25>
abr 16 19:01:10 srv1 dhclient[2440]: DHCPOFFER of 10.0.2.15 from 10.0.2.2
abr 16 19:01:10 srv1 dhclient[2440]: DHCPREQUEST for 10.0.2.15 on enp0s3 to 25>
abr 16 19:01:10 srv1 dhclient[2440]: DHCPACK of 10.0.2.15 from 10.0.2.2 (xid=0>
abr 16 19:01:10 srv1 ifup[2440]: DHCPACK of 10.0.2.15 from 10.0.2.2 (xid=0xede>
abr 16 19:01:10 srv1 ifup[2456]: RTNETLINK answers: File exists
abr 16 19:01:10 srv1 dhclient[2440]: bound to 10.0.2.15 -- renewal in 37799 se>
abr 16 19:01:10 srv1 ifup[2440]: bound to 10.0.2.15 -- renewal in 37799 second>
abr 16 19:01:10 srv1 systemd[1]: Finished Raise network interfaces.
```

Correndo o comando “ip a”, podemos verificar as interfaces de rede e podemos ver que a rede nat (enp0s3) está com o dhcp e a rede interna (enp0s8) com um novo IP

```
tiago@srv1:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default ql
en 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
       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:e7:14:95 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 86363sec preferred_lft 86363sec
    inet6 fd17:625c:f037:2:a00:27ff:fee7:1495/64 scope global dynamic mngtmpaddr nop
refixroute
       valid_lft 86366sec preferred_lft 14366sec
    inet6 fe80::a00:27ff:fee7:1495/64 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:fa:af:3a brd ff:ff:ff:ff:ff:ff
    inet 192.168.31.1/24 brd 192.168.31.255 scope global enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fefa:af3a/64 scope link
       valid_lft forever preferred_lft forever
```

definido.

Passo 2 - Hostname

Agora mudamos o hostname do nosso servidor. Para isso, editamos o ficheiro hostname e mudamos para SRV1.

```
tiago@srv1:~$ sudo nano /etc/hostname
```

```
  GNU nano 4.8                  /etc/hostname                       Modified
srv1


File Name to Write: /etc/hostname
^G Get Help      M-D DOS Format     M-A Append         M-B Backup File
^C Cancel        M-M Mac Format     M-P Prepend        ^T To Files
```

Salvando as alterações e saindo do ficheiro podemos usar este comando para verificar se o hostname foi realmente alterado.

```
tiago@srv1:~$ hostname
srv1
```

Após alterar o hostname, vamos identificar o nosso domínio. Para isso, vamos editar o ficheiro hosts, onde colocamos o nosso domínio e o respetivo hostname.

```
tiago@srv1:~$ sudo nano /etc/hosts
```

```
  GNU nano 4.8
127.0.0.1 localhost
127.0.1.1 srv1.istec.local srv1
```

Depois de salvar e sair podemos verificar com o comando abaixo se o nome do domínio foi alterado.

```
tiago@srv1:~$ hostname -f
srv1.istec.local
tiago@srv1:~$
```

Passo 3 - DHCP (Dynamic Host Configuration Protocol)

Para configurarmos o dhcp, primeiro instalamos o serviço do mesmo.

```
tiago@srv1:~$ sudo apt-get install isc-dhcp-server
```

Após a instalação, podemos editar o ficheiro de configuração das interfaces do dhcp.

```
tiago@srv1:~$ sudo nano /etc/default/isc-dhcp-server
```

Vamos comentar a linha de interfaces v6 e escrever o nome da nossa interface em v4.

```
# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPDv4_PID=/var/run/dhcpd.pid
#DHCPDv6_PID=/var/run/dhcpd6.pid

# Additional options to start dhcpd with.
#       Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="enp0s8"
#INTERFACESv6=""
```

Agora vamos editar o ficheiro de configuração do dhcp mas antes vamos levar o conteudo dele para um backup e começar do zero.

```
tiago@srv1:~$ sudo mv /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.conf.orig
```

Abrimos agora o ficheiro de configuração do dhcp.

```
tiago@srv1:~$ sudo nano /etc/dhcp/dhcpd.conf
```

Aqui definimos a nossa subnet colocando também a netmask. Dentro das chavetas, colocamos a range de IPs que o dhcp irá apanhar, ou seja, entre 192.168.31.2 e 192.168.31.254 sendo o 192.168.31.1 o IP do servidor. Como dns colocamos o IP do servidor e o dns da google e como domain name colocamos o nome que queremos. Abrimos mais umas chavetas para o cliente, colocamos o MAC address do cliente e definimos o IP dele para fixar no penúltimo da rede.

```
subnet 192.168.31.0 netmask 255.255.255.0
{
range dynamic-bootp 192.168.31.2 192.168.31.254;
option routers 192.168.31.1;
option domain-name-servers 192.168.31.1, 8.8.8.8;
option domain-name "istec.local";
}

host WINCLIENT
{
hardware ethernet 08:00:27:7E:21:1D;
fixed-address 192.168.31.254;
}
```

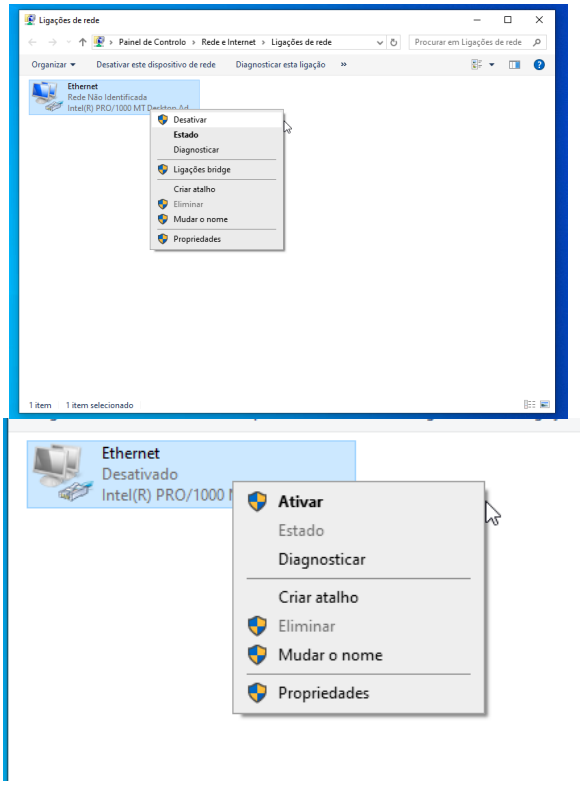
Para verificar o MAC address, ligamos a máquina cliente, vamos no cmd e corremos o comando ipconfig /all e verificando a linha do Physical Address.

```
C:\Users\client1>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : DESKTOP-RSG7DT3
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
   Physical Address. . . . . . . . . : 08-00-27-7E-21-1D
```

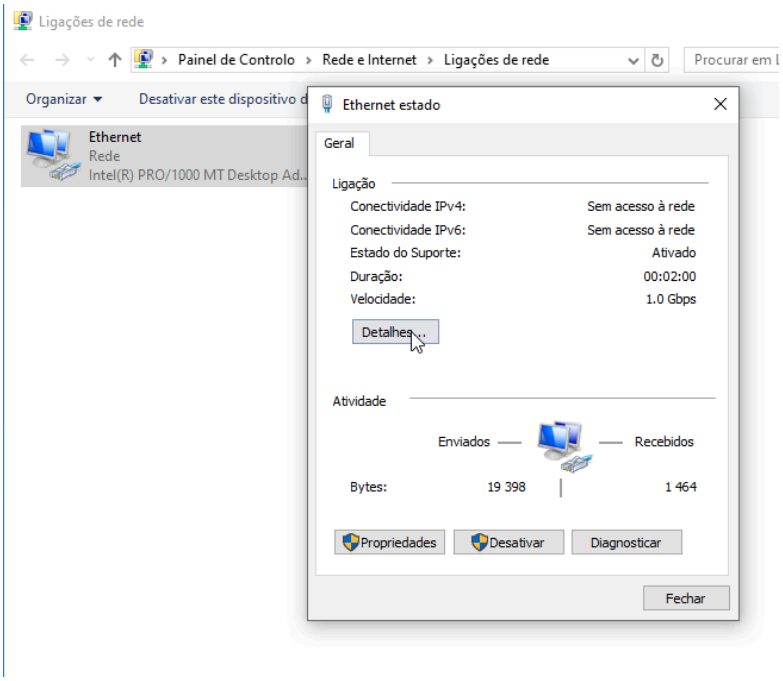
Depois disso, podemos guardar e sair do ficheiro, reiniciando de novo o serviço e verificando se ele está a funcionar.

```
tiago@srv1:~$ sudo systemctl restart isc-dhcp-server
tiago@srv1:~$ sudo systemctl status isc-dhcp-server
● isc-dhcp-server.service - ISC DHCP IPv4 server
     Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; ven
     Active: active (running) since Fri 2025-04-18 18:55:52 UTC; 1s ago
       Docs: man:dhcpd(8)
   Main PID: 4679 (dhcpd)
      Tasks: 4 (limit: 3419)
     Memory: 5.0M
     CGroup: /system.slice/isc-dhcp-server.service
             └─4679 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/

abr 18 18:55:52 srv1 sh[4679]: Wrote 0 new dynamic host decls to leases file.
abr 18 18:55:52 srv1 dhcpd[4679]: Wrote 0 leases to leases file.
abr 18 18:55:52 srv1 sh[4679]: Wrote 0 leases to leases file.
abr 18 18:55:53 srv1 dhcpd[4679]: Listening on LPF/enp0s8/08:00:27:fa:af:3a/19
abr 18 18:55:53 srv1 sh[4679]: Listening on LPF/enp0s8/08:00:27:fa:af:3a/192.1
abr 18 18:55:53 srv1 sh[4679]: Sending on   LPF/enp0s8/08:00:27:fa:af:3a/192.1
abr 18 18:55:53 srv1 sh[4679]: Sending on   Socket/fallback/fallback-net
abr 18 18:55:53 srv1 dhcpd[4679]: Sending on   LPF/enp0s8/08:00:27:fa:af:3a/19
abr 18 18:55:53 srv1 dhcpd[4679]: Sending on   Socket/fallback/fallback-net
abr 18 18:55:53 srv1 dhcpd[4679]: Server starting service.
lines 1-20/20 (END)
```

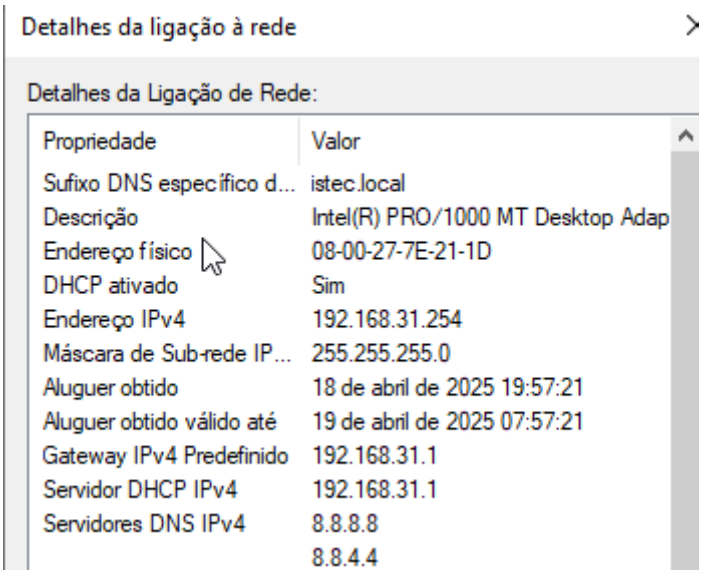
Depois disso, vamos na máquina cliente, abrimos o painel de controlo>rede e internet>ligações de rede e clicamos com o botão direito na nossa rede para desativar e voltar a ativar, reiniciando-a.



Depois que identifique a rede, clicamos duas vezes para ver o estado da rede e clicamos em detalhes.



E como podemos verificar já temos o nosso dns, dhcp e IP fixo para o penúltimo da rede.



Passo 4 - DNS (Domain Name System)

Agora precisamos de ter o nosso DNS a funcionar, conseguindo dar ping no cliente a partir do nome definido. Para isso vamos instalar o bind9.

```
tiago@srv1:~$ sudo apt install -y bind9 bind9utils bind9-doc
```

Agora, criamos um backup do ficheiro named.conf.options antes de começar a editar o mesmo.

```
tiago@srv1:~$ sudo cp /etc/bind/named.conf.options /etc/bind/named.conf.options.orig
inal
tiago@srv1:~$ sudo nano /etc/bind/named.conf.options
```

Definimos estas configurações.

```
        dnssec-validation auto;

        listen-on-v6 { any; };
        recursion yes;
        listen-on { 127.0.0.1; 192.168.31.1; };
        allow-transfer {none;};

        forwarders {
        192.168.31.1;
        };
};
```

Agora vamos em ficheiro named.conf.local, lembrando de dar restart e verificar o status do serviço sempre que alterarmos algum ficheiro.

Aqui definimos as nossas zonas de pesquisa direta e inversa, definindo um ficheiro para cada uma delas.

```
 GNU nano 4.8                  /etc/bind/named.conf.local                  Modified
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

// Zona direta para o dominio istec.local
        zone "istec.local" {
        type master;
        file "/etc/bind/db.istec.local";
        };

// Zona inversa para a rede 192.168.31.0/24
        zone "31.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/db.192";
        };
```

Agora fazemos backup do ficheiro de pesquisa direta.

```
tiago@srv1:~$ sudo cp /etc/bind/db.local /etc/bind/db.istec.local
```

Abrimos o mesmo.

```
tiago@srv1:~$ sudo nano /etc/bind/db.istec.local
```

Aqui colocamos o nome do nosso domínio bem como as entradas do servidor e cliente, colocando o IP de cada um.

```
 GNU nano 4.8                    /etc/bind/db.istec.local
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     localhost. root.localhost. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      srv1.istec.local.
srv1    IN      A       192.168.31.1
winclient IN    A       192.168.31.254
```

Depois de salvar damos restart ao serviço e verificamos se está a funcionar. Além

```
tiago@srv1:~$ sudo named-checkzone istec.local /etc/bind/db.istec.local
zone istec.local/IN: loaded serial 2
OK
```

disso, damos um checkzone para verificar o ficheiro que acabamos de editar e verificar se está ok.

Depois, vamos para o ficheiro de pesquisa inversa, fazendo backup e abrindo-o.

```
tiago@srv1:~$ sudo nano /etc/bind/db.192
tiago@srv1:~$ sudo cp /etc/bind/db.127 /etc/bind/db.192
```

Aqui faremos basicamente a mesma coisa, só que definimos ao contrário, colocando desta vez o último octeto do IP primeiro.

```
 GNU nano 4.8                    /etc/bind/db.192
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     localhost. root.localhost. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      srv1.istec.local.
1       IN      PTR     srv1.istec.local.
254     IN      PTR     winclient.istec.local.
```

Depois de salvar e sair, verificamos o ficheiro para ver se está ok também.

```
tiago@srv1:~$ sudo named-checkzone 0.31.168.192.in-addr.arpa /etc/bind/db.192
zone 0.31.168.192.in-addr.arpa/IN: loaded serial 1
OK
```

Mais uma vez damos restart e verificamos se o serviço está funcional.

```
tiago@srv1:~$ sudo systemctl status bind9
● named.service - BIND Domain Name Server
     Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: ena>
     Active: active (running) since Tue 2025-04-22 11:13:28 UTC; 6s ago
       Docs: man:named(8)
   Main PID: 3913 (named)
      Tasks: 6 (limit: 3419)
     Memory: 5.3M
     CGroup: /system.slice/named.service
             └─3913 /usr/sbin/named -f -u bind

abr 22 11:13:28 srv1 named[3913]: zone 127.in-addr.arpa/IN: loaded serial 1
abr 22 11:13:28 srv1 named[3913]: zone istec.local/IN: loaded serial 2
abr 22 11:13:28 srv1 named[3913]: zone 0.31.168.192.in-addr.arpa/IN: loaded serial 1
abr 22 11:13:28 srv1 named[3913]: zone localhost/IN: loaded serial 2
abr 22 11:13:28 srv1 named[3913]: all zones loaded
abr 22 11:13:28 srv1 named[3913]: zone istec.local/IN: sending notifies (serial 2)
abr 22 11:13:28 srv1 named[3913]: zone 0.31.168.192.in-addr.arpa/IN: sending notifi>
abr 22 11:13:28 srv1 named[3913]: running
abr 22 11:13:28 srv1 named[3913]: managed-keys-zone: Key 20326 for zone . is now tr>
abr 22 11:13:28 srv1 named[3913]: managed-keys-zone: Key 38696 for zone . is now tr>
lines 1-20/20 (END)
```

Agora para definirmos o dns permanentemente, temos que instalar o pacote do resolvconf.

```
tiago@srv1:~$ sudo apt install resolvconf
```

Após a instalação do mesmo, abrimos e editamos o ficheiro resolvconf.conf.d/head.

```
tiago@srv1:~$ sudo nano /etc/resolvconf/resolv.conf.d/head
```

Corremos o seguinte comando para verificar se tem algum erro ou está tudo bem.

```
tiago@srv1:~$ sudo resolvconf -u
```

Depois disso, e estar tudo verificado, vemos se o dns e a pesquisa inversa estão a funcionar.

```
tiago@srv1:~$ host 192.168.31.1
1.31.168.192.in-addr.arpa domain name pointer srv1.istec.local.
tiago@srv1:~$ host srv1.istec.local
srv1.istec.local has address 192.168.31.1
tiago@srv1:~$ nslookup srv1.istec.local
Server:         127.0.0.1
Address:        127.0.0.1#53

Name:   srv1.istec.local
Address: 192.168.31.1
```

Podemos também agora ir no cliente para dar um nslookup no dns do servidor e ver se reconhece.

```
C:\Users\client1>nslookup istec.local
Server:  srv1.istec.local
Address:  192.168.31.1

Name:    istec.local
```

Passo 5 - NAT (Network Address Translation)

Para conseguirmos ter internet no cliente a partir da rede interna dele, iremos editar o seguinte ficheiro.

```
tiago@srv1:~$ sudo nano /etc/sysctl.conf
```

Descomentamos a seguinte linha.

```
# Uncomment the next li
net.ipv4.ip_forward=1
```

Damos restart ao sistema.

```
tiago@srv1:~$ init 6
```

Corremos o seguinte comando.

```
tiago@srv1:~$ sudo iptables --table nat --append POSTROUTING --out-interface enp0s3
-j MASQUERADE
```

Agora, já devemos ter internet, porém para tornar esta definição permanente teremos de criar um ficheiro que inicie assim que o servidor o faça também. Criamos e editamos um ficheiro chamado script.sh no root.

```
tiago@srv1:~$ sudo nano /root/script.sh
```

Colocamos o comando anterior e salvamos.

```
  GNU nano 4.8                    /root/script.sh                    Modifi
#!/bin/sh
iptables --table nat --append POSTROUTING --out-interface enp0s3 -j MASQUERADE
```

Agora alteramos as permissões do script para poder haver execução.

```
tiago@srv1:~$ sudo chmod +x /root/script.sh
```

Depois, editamos o ficheiro rc.local e colocamos a localização do ficheiro criado anteriormente.

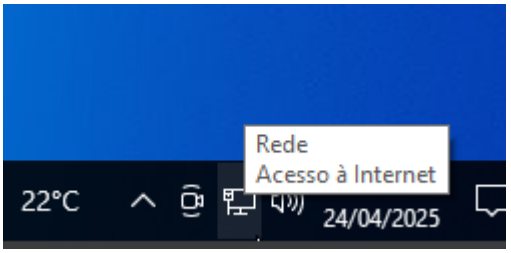
```
tiago@srv1:~$ sudo nano /etc/rc.local
```

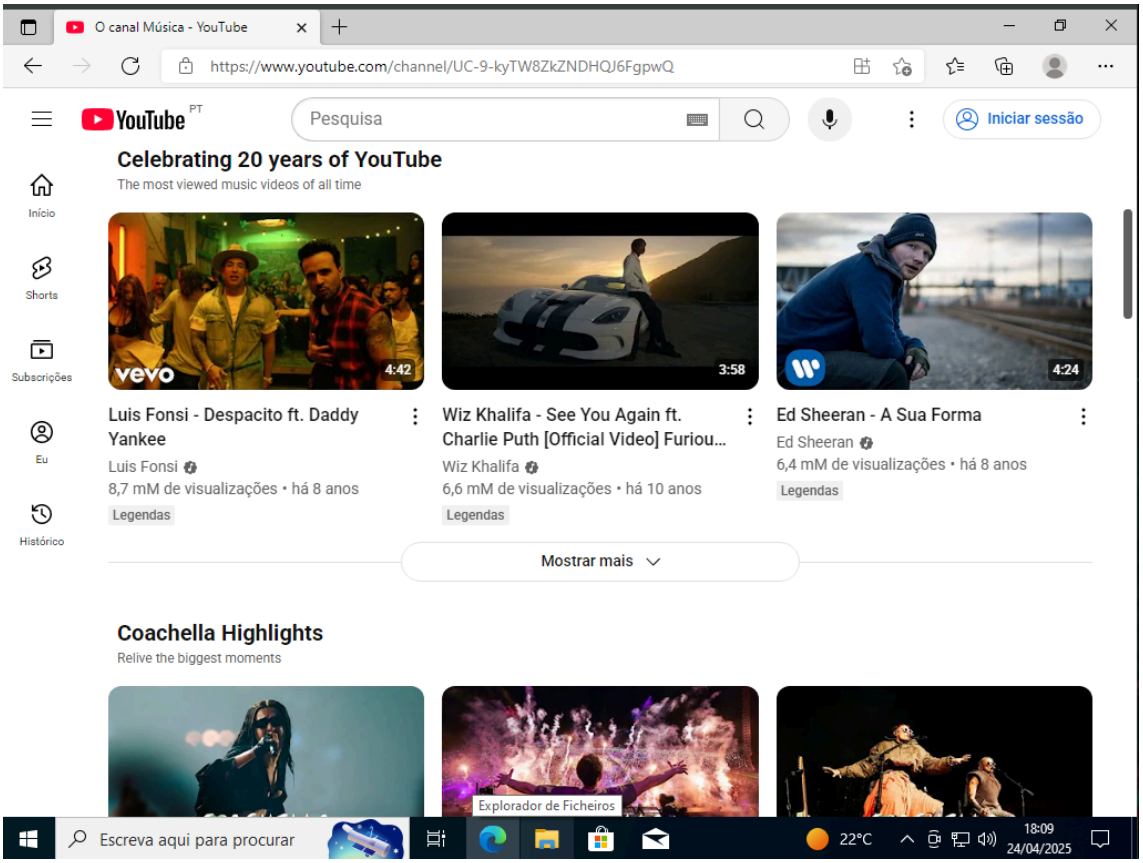
```
  GNU nano 4.8
#!/bin/sh -e

/root/script.sh

exit 0
```

Damos permissões de execução.

```
tiago@srv1:~$ sudo chmod +x /etc/rc.local
```

E podemos ir já verificar se realmente estamos com acesso à internet no cliente.





Passo 6 - Users e Grupos

Vamos agora criar alguns utilizadores e grupos. Usaremos o seguinte comando para criar utilizadores e faremos isto para os outros 6 (incluindo o user admin).

```
root@srv1:/home/tiago# adduser us1
Adding user `us1' ...
Adding new group `us1' (1002) ...
Adding new user `us1' (1002) with group `us1' ...
Creating home directory `/home/us1' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for us1
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
```

Adicionamos agora o grupos.

```
root@srv1:/home/tiago# groupadd grp1
root@srv1:/home/tiago# groupadd grp2
root@srv1:/home/tiago# groupadd grp3
```

Atribuímos os users aos devidos grupos.

```
root@srv1:/home/tiago# gpasswd -M us1,us2 grp1
root@srv1:/home/tiago# gpasswd -M us3,us4 grp2
root@srv1:/home/tiago# gpasswd -M us5,us6,us7 grp3
```

Tornamos o grp1 o grupo principal do us1 e podemos verificar com “id us1” que está certo.

```
tiago@srv1:~$ sudo usermod -g grp1 us1
```

```
tiago@srv1:~$ id us1
uid=1002(us1) gid=1009(grp1) groups=1009(grp1)
```

Passo 7 - Crontab e AT

Abrimos o crontab e colocamos os seguintes agendamentos. O professor pediu o último para dia 31 de abril mas como não existe, coloquei para dia 30.

```
tiago@srv1:~$ sudo crontab -e
```

```
  GNU nano 4.8                  /tmp/crontab.lM5ZcH/crontab                  Modi
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
0 0 * * 0 tar -czf /backup-admin.tar.gz /home/admin
0 0 1 1,2,8 * tail -n 10 /etc/passwd > /root/cp-passwd.dat
0 0 30 4 * cp /etc/*.conf /root/
```

Podemos verificar aqui que os agendamentos estão guardados.

```
tiago@srv1:~$ sudo crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
0 0 * * 0 tar -czf /backup-admin.tar.gz /home/admin
0 0 1 1,2,8 * tail -n 10 /etc/passwd > /root/cp-passwd.dat
0 0 30 4 * cp /etc/*.conf /root/
```

Agora colocamos o user admin na lista de permitidos do cron e do at.

```
tiago@srv1:~$ echo "admin" | sudo tee /etc/cron.allow > /dev/null
tiago@srv1:~$ echo "admin" | sudo tee /etc/at.allow > /dev/null
```

Podemos agora verificar que o admin tem permissão de usar os comandos, enquanto que os outros users não.

```
admin@srv1:~$ at now + 1 minute
warning: commands will be executed using /bin/sh
at> <EOT>
job 6 at Thu Apr 24 18:15:00 2025
```

```
tiago@srv1:~$ at now + 1 minute
You do not have permission to use at.
```

```
tiago@srv1:~$ crontab -e
You (tiago) are not allowed to use this program (crontab)
See crontab(1) for more information
```

Passo 8 - SSH (Secure Shell)

Vamos instalar o pacote de ssh server e ssh.

```
tiago@srv1:~$ sudo apt-get install openssh-server
```

```
tiago@srv1:~$ sudo apt install ssh
```

Agora, antes de editar o ficheiro de configuração do ssh, criamos um backup.

```
tiago@srv1:~$ sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config_original
```

```
tiago@srv1:~$ sudo nano /etc/ssh/sshd_config
```

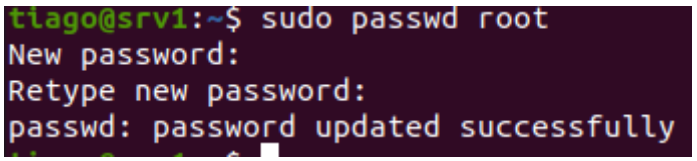
Definimos a porta que o ssh irá usar para 333.

```
Include /etc/ssh/sshd_co

Port 333
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

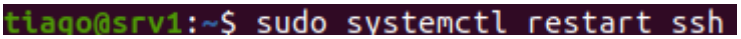
Descomentamos e mudamos a seguinte linha para permitir entrar numa conexão ssh via root.

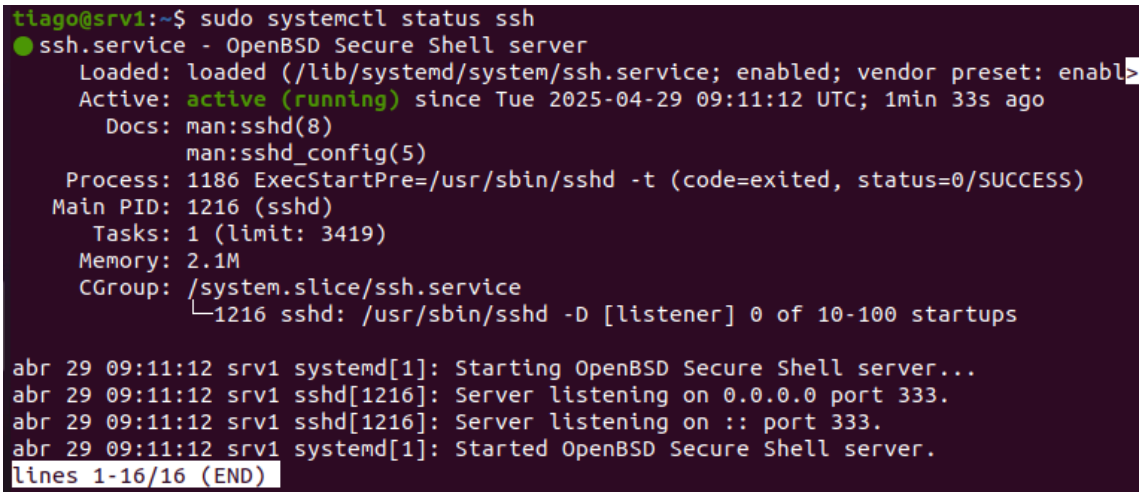
```
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Lembrando que teremos de definir uma password para o root para ser possível entrar com ele no ssh



Reiniciamos o serviço ssh, e verificamos se está a correr.





(A confirmação de funcionalidade estará nas configurações do windows cliente mais abaixo).

Passo 9 -FTP (File Transfer Protocol)

Vamos obter o pacote FTP.

```
tiago@srv1:~$ sudo apt-get install vsftpd
```

Agora editamos o ficheiro de configuração, mas primeiro criamos o backup do ficheiro.

```
tiago@srv1:~$ sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.orig
tiago@srv1:~$ sudo nano /etc/vsftpd.conf
```

Adicionamos a seguinte linha “listen_port=222”, definindo a porta desejada para ser utlizada na conexão ftp.

```
#
# Run standalone?  vsftp
# daemon started from an
listen=NO
listen_port=222
#
```

Descomentamos ou modificamos as seguintes linhas.

```
# Allow anonymous FTP? (D
anonymous_enable=NO
#
# Uncomment this to allow
local_enable=YES
#
# Uncomment this to enabl
write_enable=YES
#
```

Para limitar os users apenas à sua homefolder, descomentamos o seguinte.

```
# You may restrict local users to
# the possible risks in this befor
# chroot_list_enable below.
chroot_local_user=YES
allow_writeable_chroot=YES
#
```

Agora para colocarmos o admin a poder andar por todas as pastas no sistema, damos enable e definimos o caminho da lista onde colocaremos o user.

```
#chroot_local_user=YES
chroot_list_enable=YES
# (default follows)
chroot_list_file=/etc/vsftpd.chroot_list
#
```

Depois de salvar e sair, abrimos a lista chroot.

```
tiago@srv1:~$ sudo nano /etc/vsftpd.chroot_list
```

Escrevemos o nome do utilizador que queremos que tenha acesso aos ficheiros e pastas do sistema todo.

```
  GNU nano 4.8                    /etc/vsftpd.chroot_lis
admin
```

Agora, vamos criar um certificado ssl, para isso colocamos o seguinte comando definindo a localização do ficheiro com o certificado e com a chave.

```
tiago@srv1:~$ sudo openssl req -x509 -nodes -newkey rsa:2048 -keyout /etc/ssl/privat
e/vsftpd.key -out /etc/ssl/certs/vsftpd.pem -days 365
Generating a RSA private key
....+++++
.................................+++++
writing new private key to '/etc/ssl/private/vsftpd.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:PT
State or Province Name (full name) [Some-State]:Porto
Locality Name (eg, city) []:Porto
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Istec
Organizational Unit Name (eg, section) []:Istec
Common Name (e.g. server FQDN or YOUR name) []:Istec
Email Address []:istec@gmail.com
```

Modificamos as permissões dos dois ficheiros para apenas o dono conseguir ler e escrever.

```
tiago@srv1:~$ sudo chmod 600 /etc/ssl/certs/vsftpd.pem
tiago@srv1:~$ sudo chmod 600 /etc/ssl/private/vsftpd.key
```

Voltamos ao ficheiro de configuração do ftp

```
tiago@srv1:~$ sudo nano /etc/vsftpd.conf
```

Configuramos no final do ficheiro desta maneira, colocamos os caminhos da chave e do certificado.

```
rsa_cert_file=/etc/ssl/certs/vsftpd.pem
rsa_private_key_file=/etc/ssl/private/vsftpd.key
ssl_enable=YES
ssl_ciphers=HIGH
ssl_tlsv1=YES
ssl_sslv2=YES
ssl_sslv3=YES
force_local_data_ssl=YES
force_local_logins_ssl=YES
```

Agora damos restart no serviço e verificamos se está a correr.

```
tiago@srv1:~$ systemctl restart vsftpd
tiago@srv1:~$ systemctl status vsftpd
● vsftpd.service - vsftpd FTP server
     Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; vendor preset: en>
     Active: active (running) since Sun 2025-04-27 11:43:53 UTC; 8s ago
    Process: 2834 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty (code=exited, st>
   Main PID: 2835 (vsftpd)
      Tasks: 1 (limit: 3419)
     Memory: 844.0K
     CGroup: /system.slice/vsftpd.service
             └─2835 /usr/sbin/vsftpd /etc/vsftpd.conf

abr 27 11:43:53 srv1 systemd[1]: Starting vsftpd FTP server...
abr 27 11:43:53 srv1 systemd[1]: Started vsftpd FTP server.
lines 1-12/12 (END)
```

(A confirmação de funcionalidade estará nas configurações do windows cliente mais abaixo).

Passo 10 - Samba

Agora, instalamos o samba.

```
tiago@srv1:~$ sudo apt install samba
```

Criamos as pastas que iremos partilhar.

```
tiago@srv1:~$ sudo mkdir /share2
tiago@srv1:~$ sudo mkdir /home/share1
tiago@srv1:~$ sudo mkdir /general_share
```

Alteramos o grupo dono da pasta share1.

```
tiago@srv1:~$ sudo chown :grp1 /home/share1
```

Mudamos as permissões do share1 para permissões totais do dono e permissão de ler e executar para o grupo.

```
tiago@srv1:~$ sudo chmod 750 /home/share1
```

Damos permissão total para o us7.

```
tiago@srv1:~$ sudo setfacl -m u:us7:rwx /home/share1
```

Alteramos o grupo dono da pasta share2.

```
tiago@srv1:~$ sudo chown :grp2 /share2
```

Mudamos as permissões do share2 para permissões totais do dono e do grupo.

```
tiago@srv1:~$ sudo chmod 770 /share2
```

Damos permissões de ler e executar para o grp3.

```
tiago@srv1:~$ sudo setfacl -m g:grp3:rx /share2
```

Por fim, damos permissões totais a todos na pasta general_share.

```
tiago@srv1:~$ sudo chmod -R 777 /general_share
```

Entramos no ficheiro de configuração do samba.

```
tiago@srv1:~$ sudo nano /etc/samba/smb.conf
```

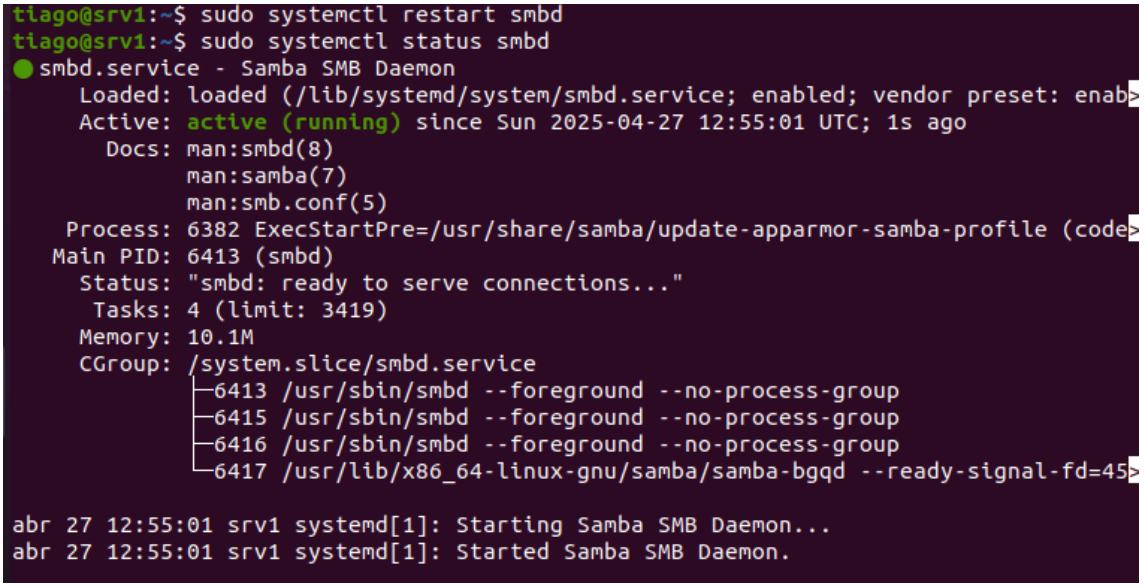
Fazemos as seguintes alterações em global.

```
[global]

unix charset = UTF-8
map to guest = Bad User


## Browsing/Identification ###

# Change this to the workgroup/NT-domain na
   workgroup = WORKGROUP

# server string is the equivalent of the NT
   server string = %h server (Samba, Ubuntu

#### Networking ####

# The specific set of interfaces / networks
# This can be either the interface name or
# interface names are normally preferred
;   interfaces = 127.0.0.1 enp0s8
```

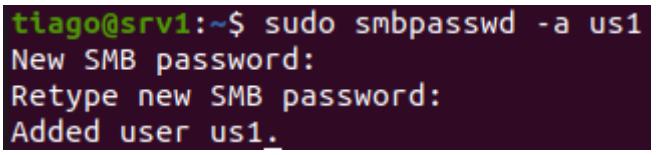
No fim do ficheiro adicionamos as seguintes configurações para as shares.

```
[homefolder]
   comment = Home Directories
   browseable = yes
   path = /home/%u
# By default, the home directori
# next parameter to 'no' if you
   read only = no

[share1]
path = /home/share1
browseable = yes
read only = yes
valid users = @grp1 us7
write list = us7

[share2]
path = /share2
browseable = no
valid users = @grp2 @grp3
read only = yes
write list = @grp2

[general_share]
path = /general_share
browseable = no
guest ok = yes
guest only = yes
writable = yes
force create mode = 777
force directory mode = 777
```

Agora damos restart ao samba e verificamos se está a correr.



Agora definimos a password para os users dentro do samba.



Comandos

Aqui temos o comando que procura todos os ficheiros em /home cujo dono seja o utilizador us1, guarde o ficheiro resultante num ficheiro owner-us1.dat em /root.

```
root@srv1:/home/tiago# find /home -user us1 > /root/owner-us1.dat
```

```
root@srv1:/home/tiago# cat /root/owner-us1.dat
/home/us1
/home/us1/.profile
/home/us1/.bashrc
/home/us1/a
/home/us1/.bash_logout
```

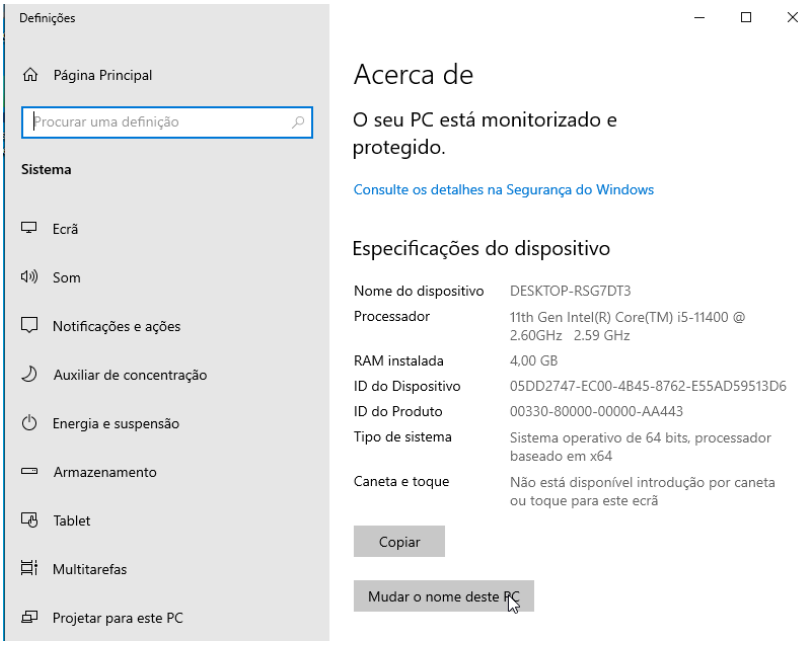
Aqui temos o comando que procura no conteúdo do ficheiro /etc/group todas as linhas que começam por “l”.

```
root@srv1:/home/tiago# grep "^l" /etc/group
lp:x:7:
list:x:38:
landscape:x:115:
lxd:x:117:tiago
lpadmin:x:125:
```

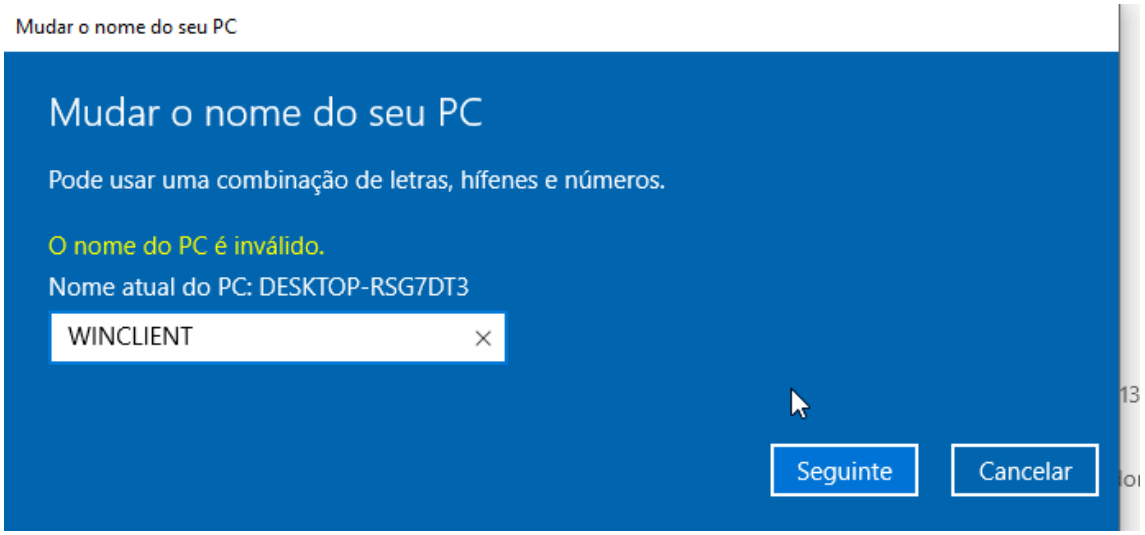
Cliente Windows

Passo 1 - Hostname

Vamos mudar o hostname e para isso vamos nas definições>Acerca de> mudar o nome deste PC.

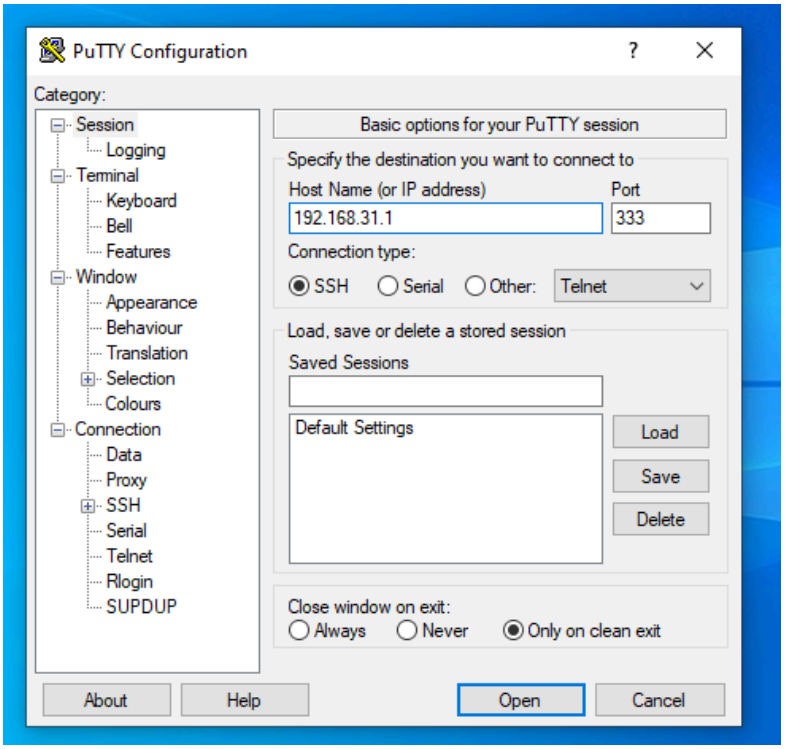


Agora mudamos o hostname.

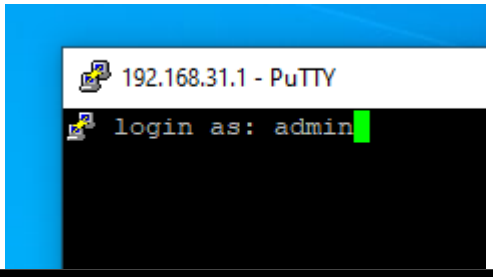


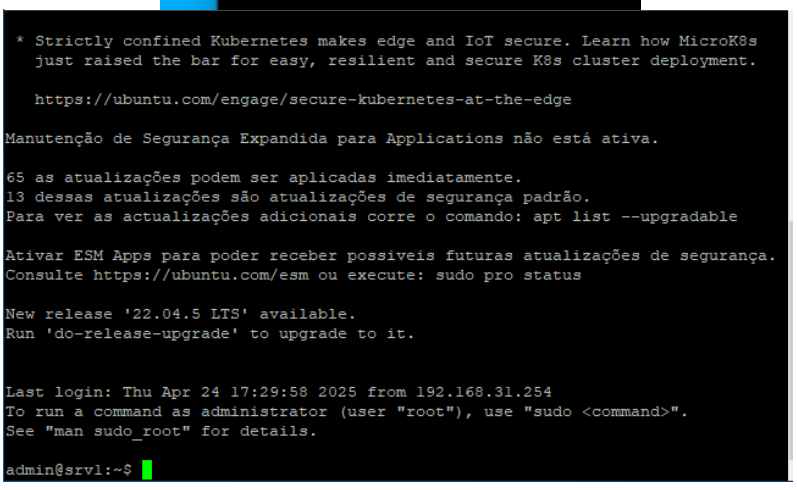
Passo 2 - SSH

Para a conexão ssh podemos usar o cmd ou instalar o puTTY. Ao abrir, colocamos o IP ou hostname do servidor e a porta para a qual destinamos a ligação, o tipo de conexão é ssh e clicamos para abrir.

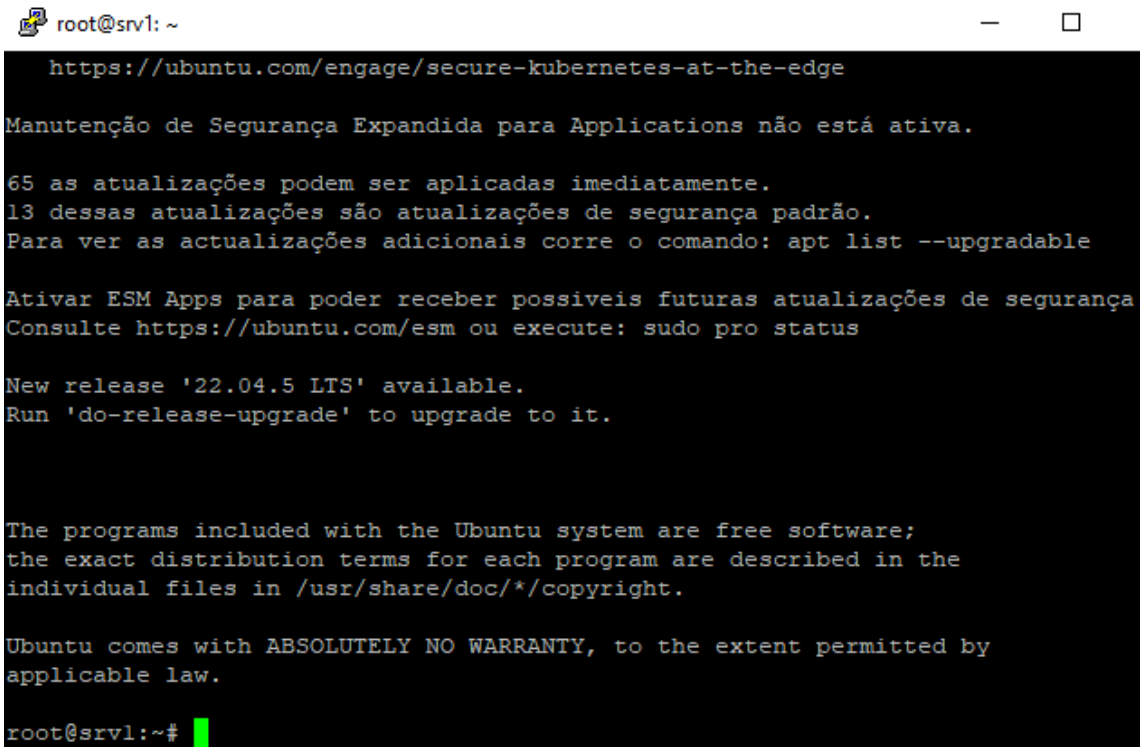


Damos login como admin.



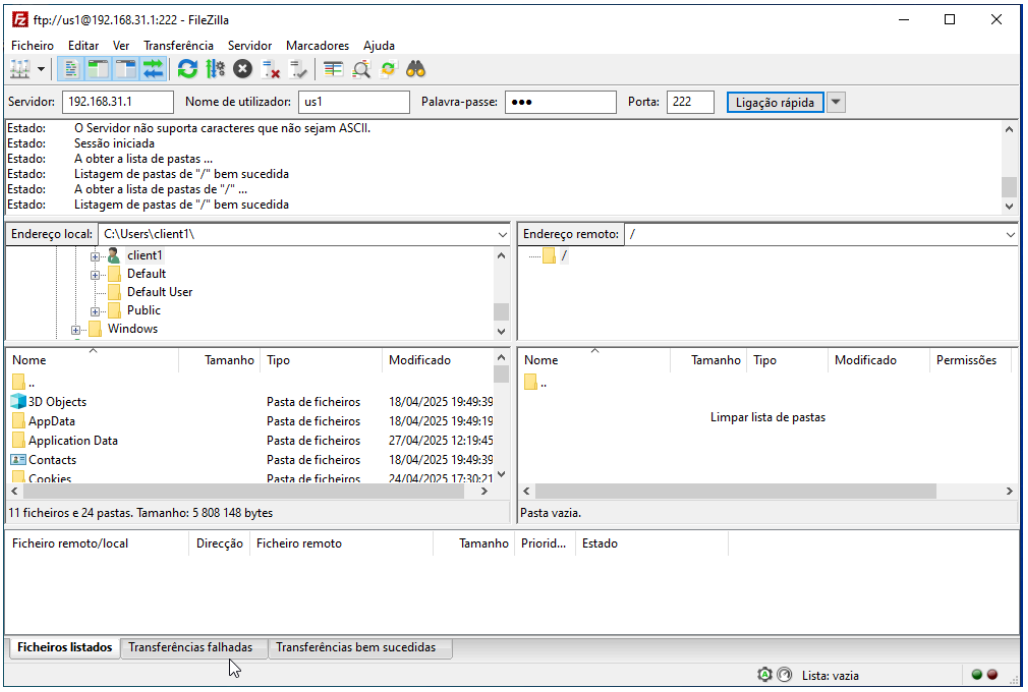


Podemos ver aqui que também é possível entrar como root.

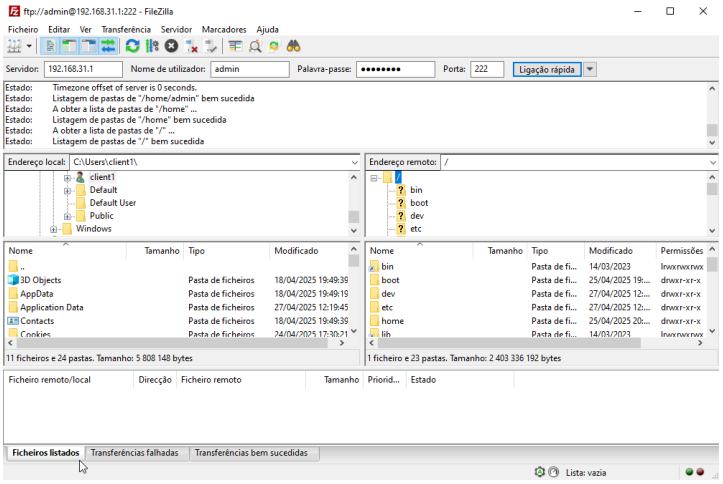


Passo 3 - FTP

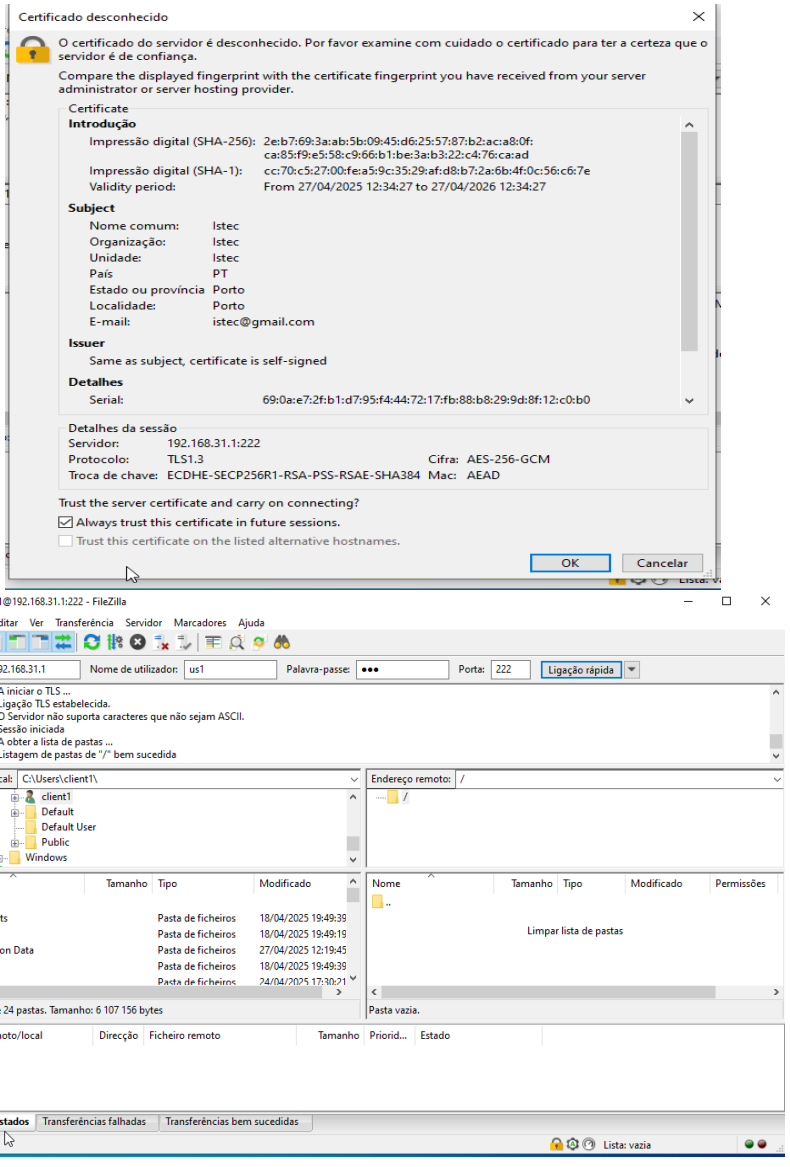
No FTP fazemos download do filezila. Abrindo o mesmo, colocamos o IP do servidor, o nome do utilizador, a password e a porta definida. Podemos já ver que o us1 tem apenas acesso à sua homefolder.



Aqui no admin podemos ver que tem acesso a todo o sistema.

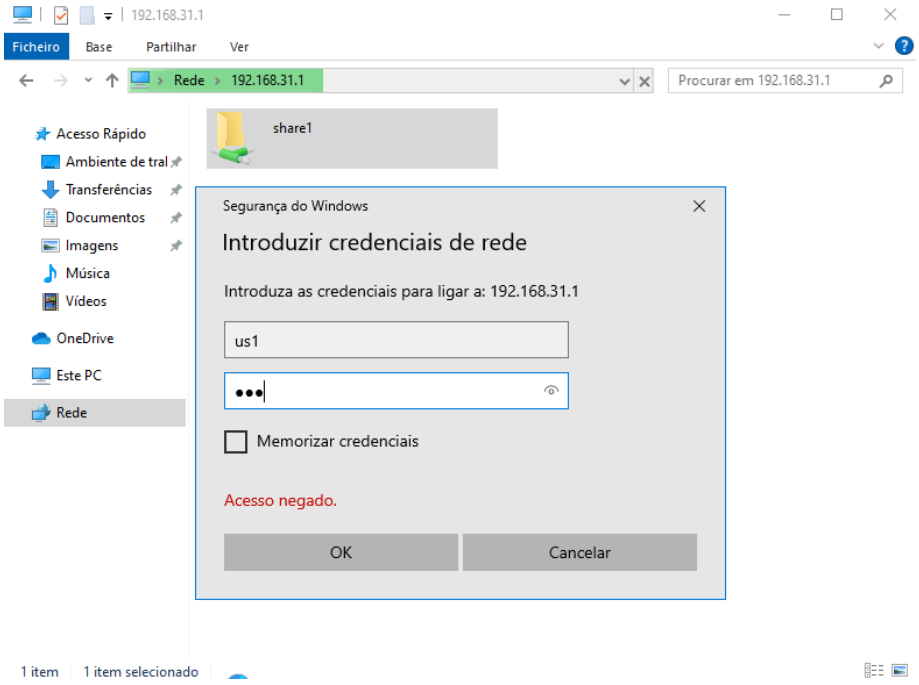


Podemos ver aqui o aviso quando eu adiciono o certificado.

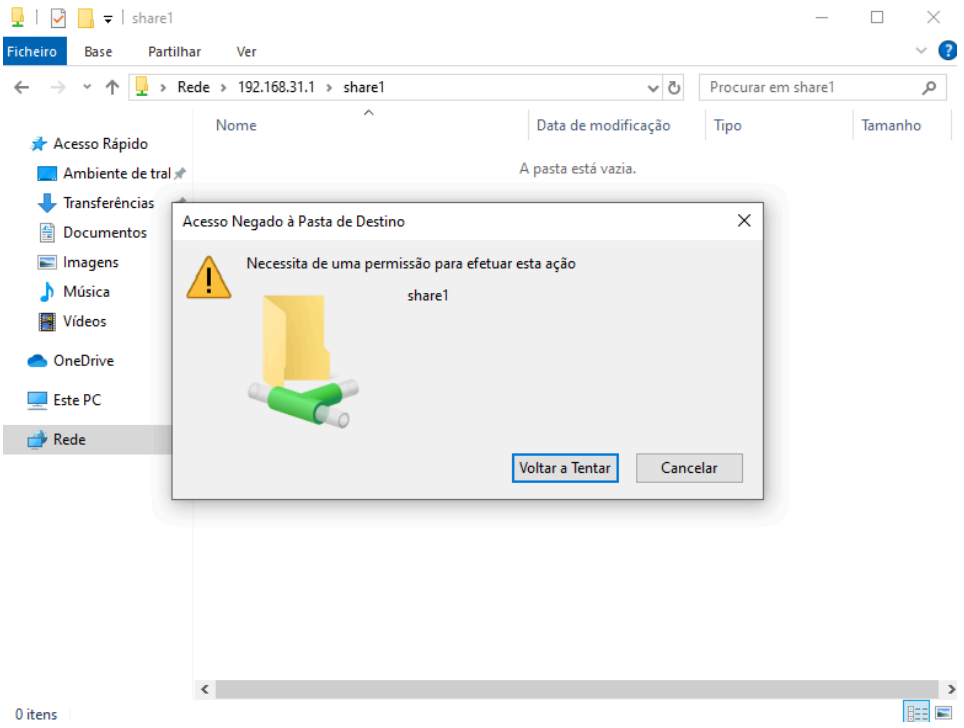


Passo 4 - Samba

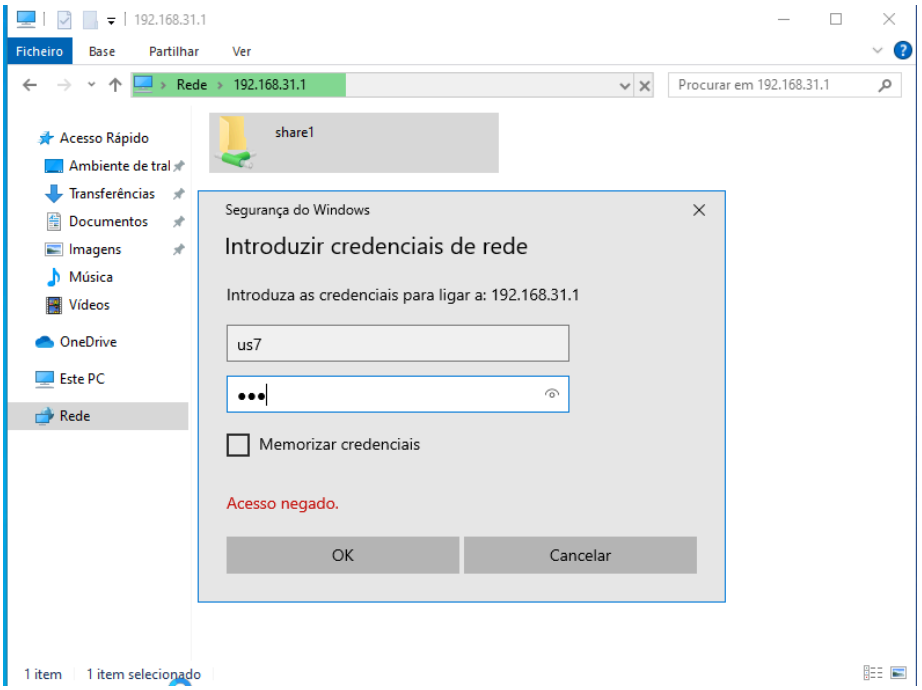
No cliente podemos ir no explorador e pesquisar o IP do server. Ao fazer isso, será pedido as credenciais.

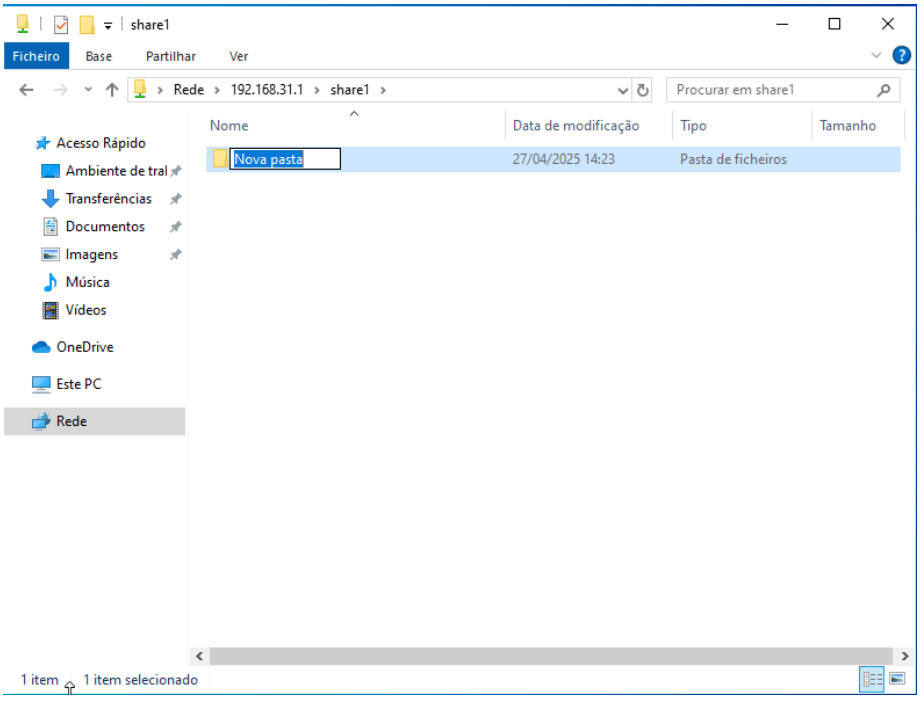


Com sessão no us1 do grp1, podemos ver que não dá para criar nada, apenas para ler

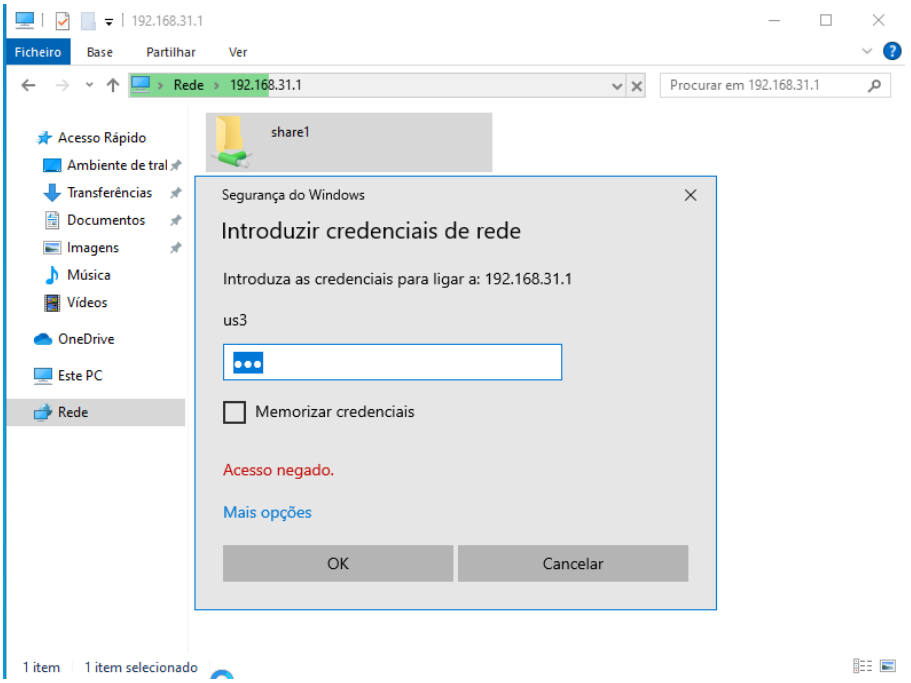


Agora, entrando com us7, podemos verificar que já consigo criar pastas.



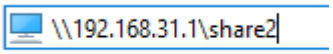


Podemos verificar que com um user de outro grupo não é possível entrar.

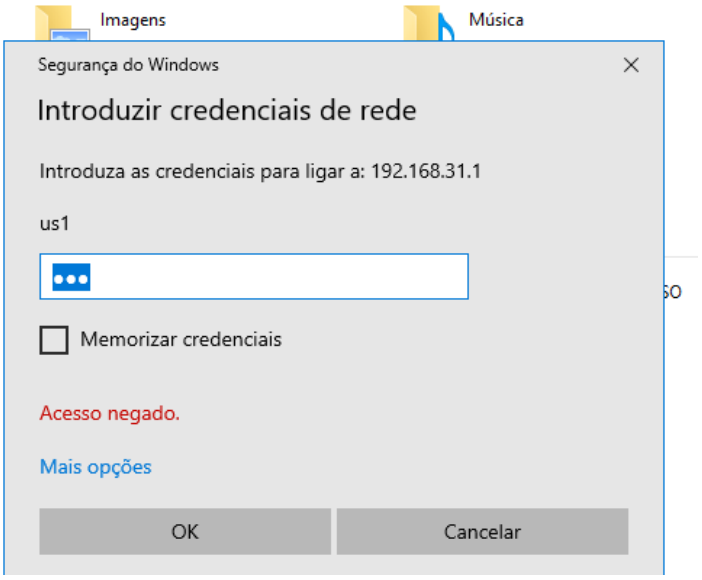


Agora,

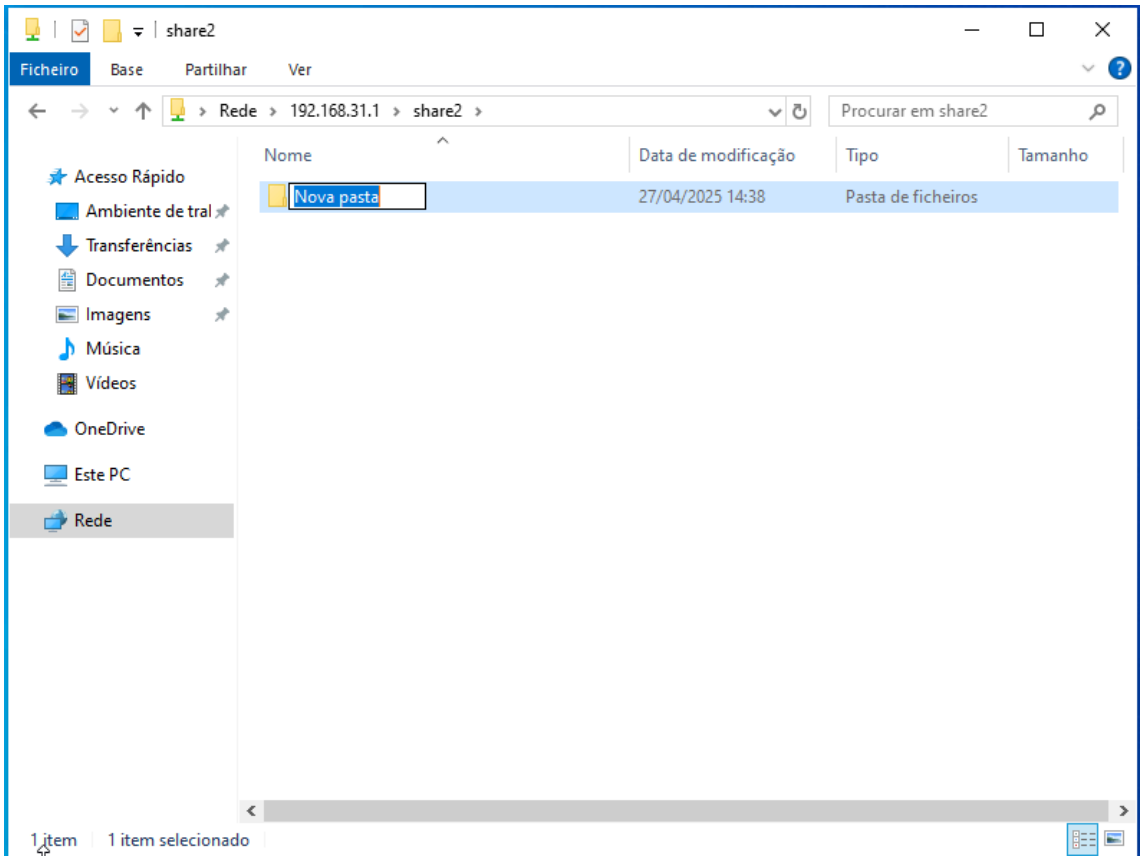
precisamos de ir diretamente às outras pastas pois colocamos elas escondidas.



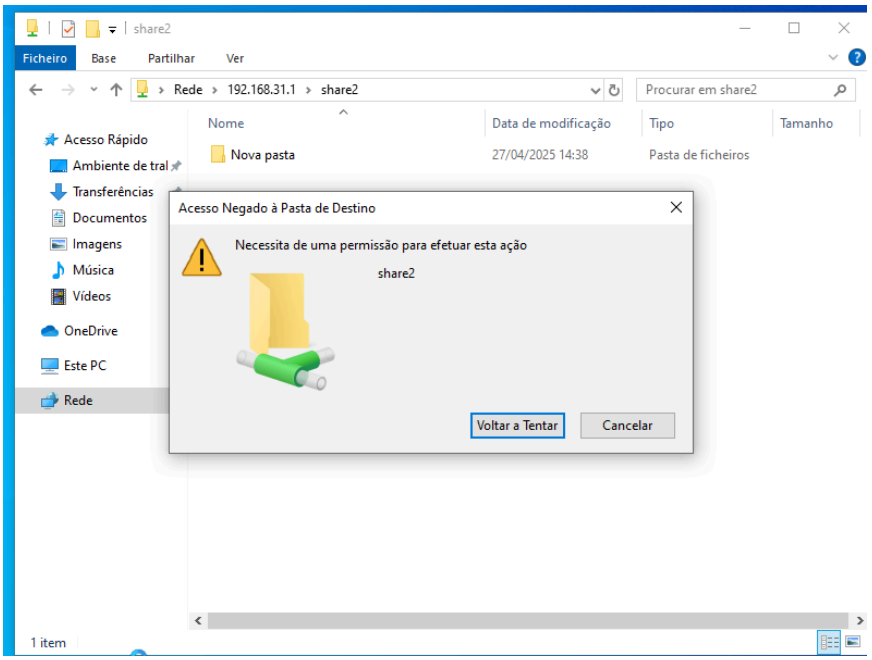
Podemos verificar que o us1 do grupo 1 não pode entrar.



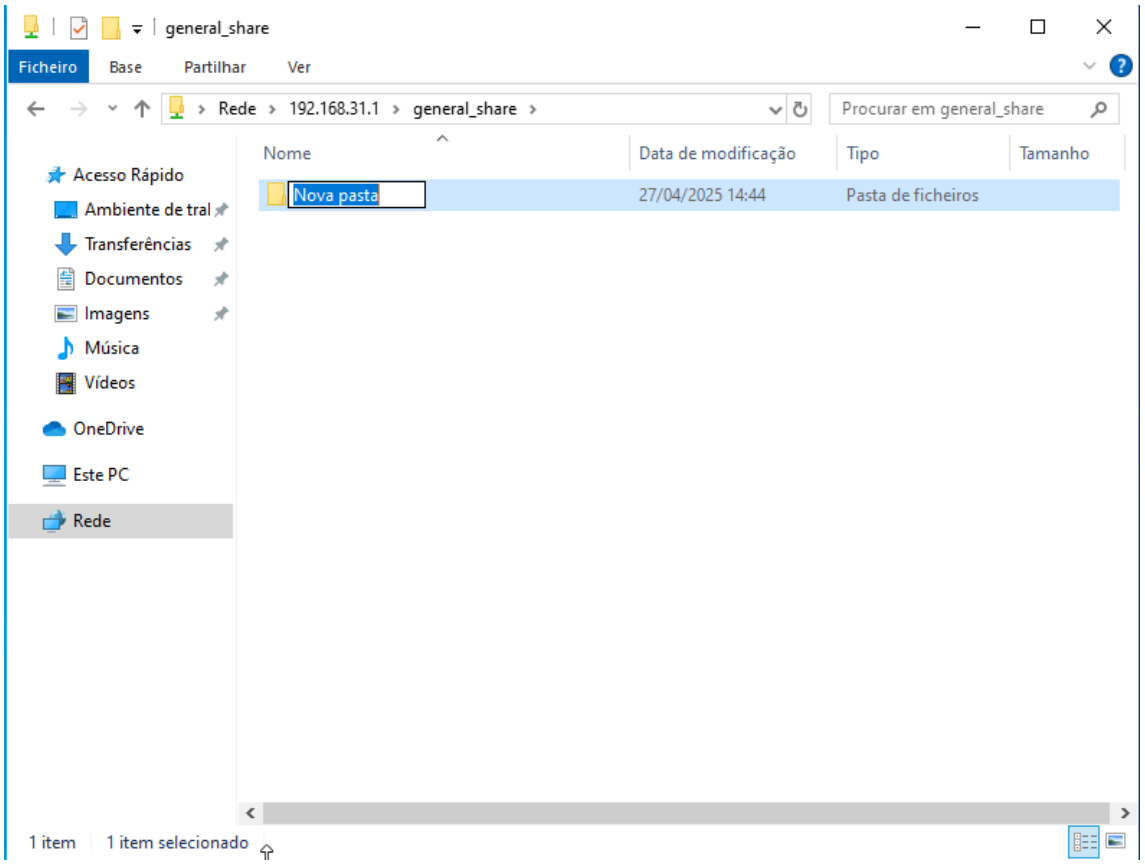
Com o us3 já podemos criar pastas.



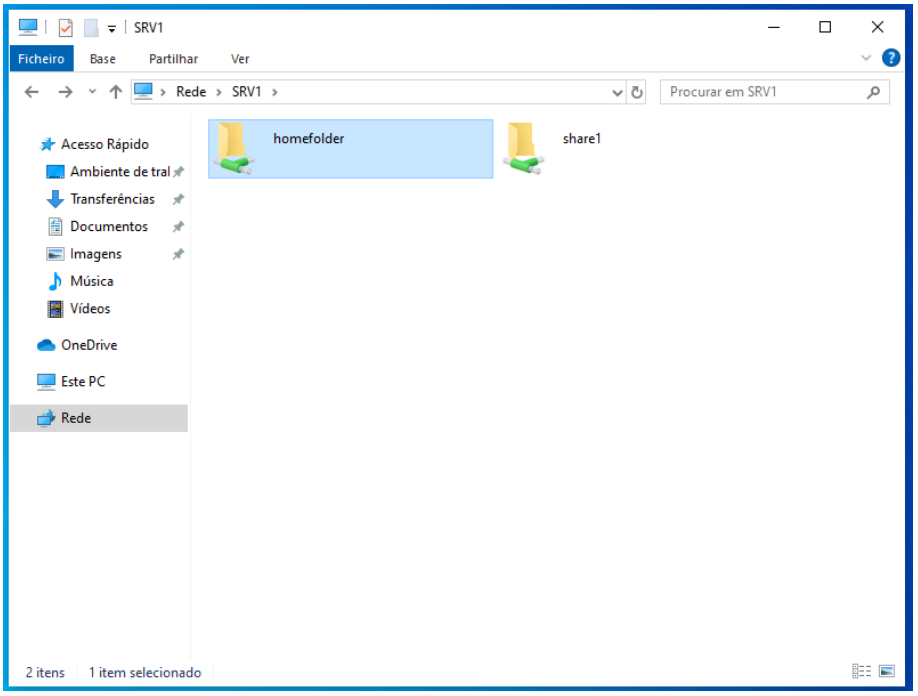
Enquanto que com um user do grupo 3 pode apenas ler.



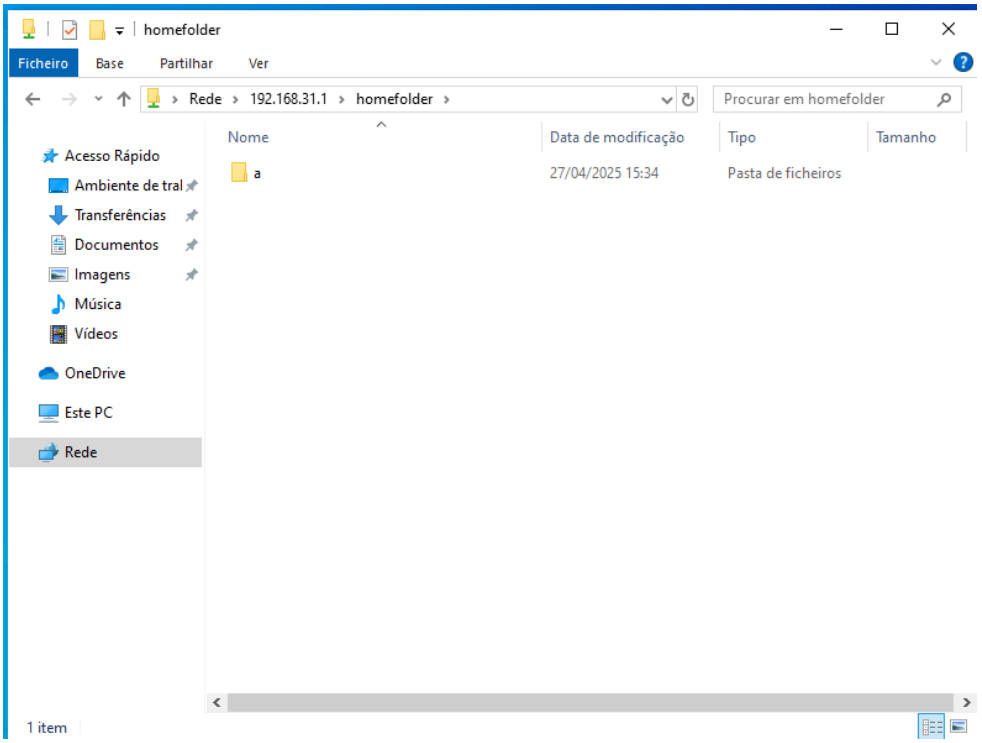
Depois disso temos a general_share que podemos fazer o que quisermos com qualquer user.



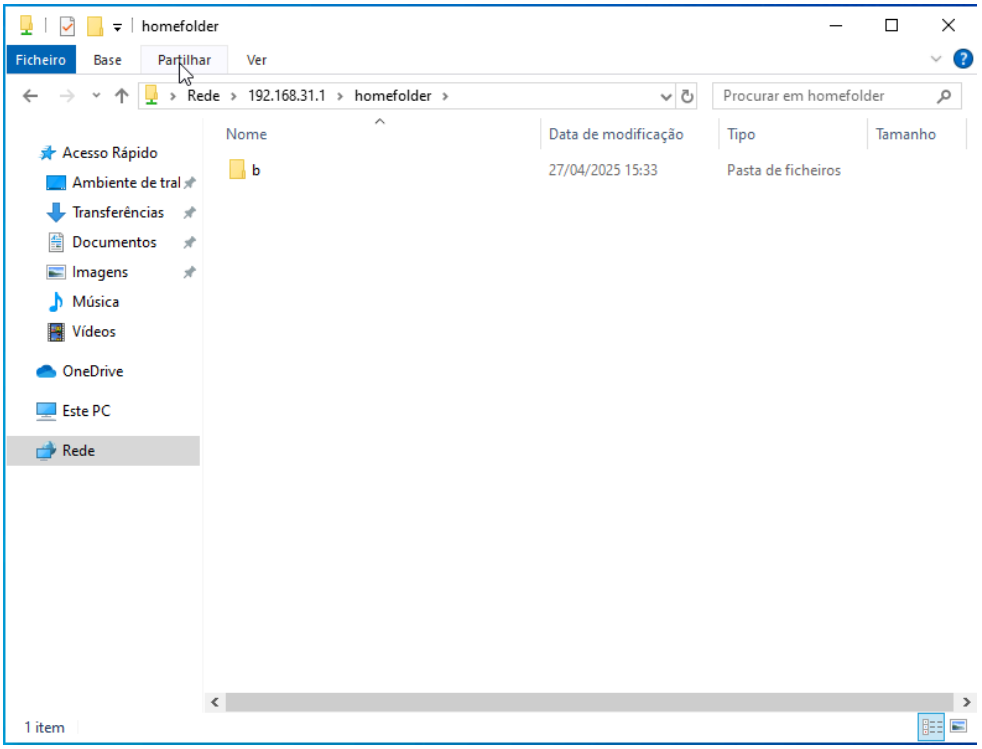
Por fim, temos a homefolder de cada user.



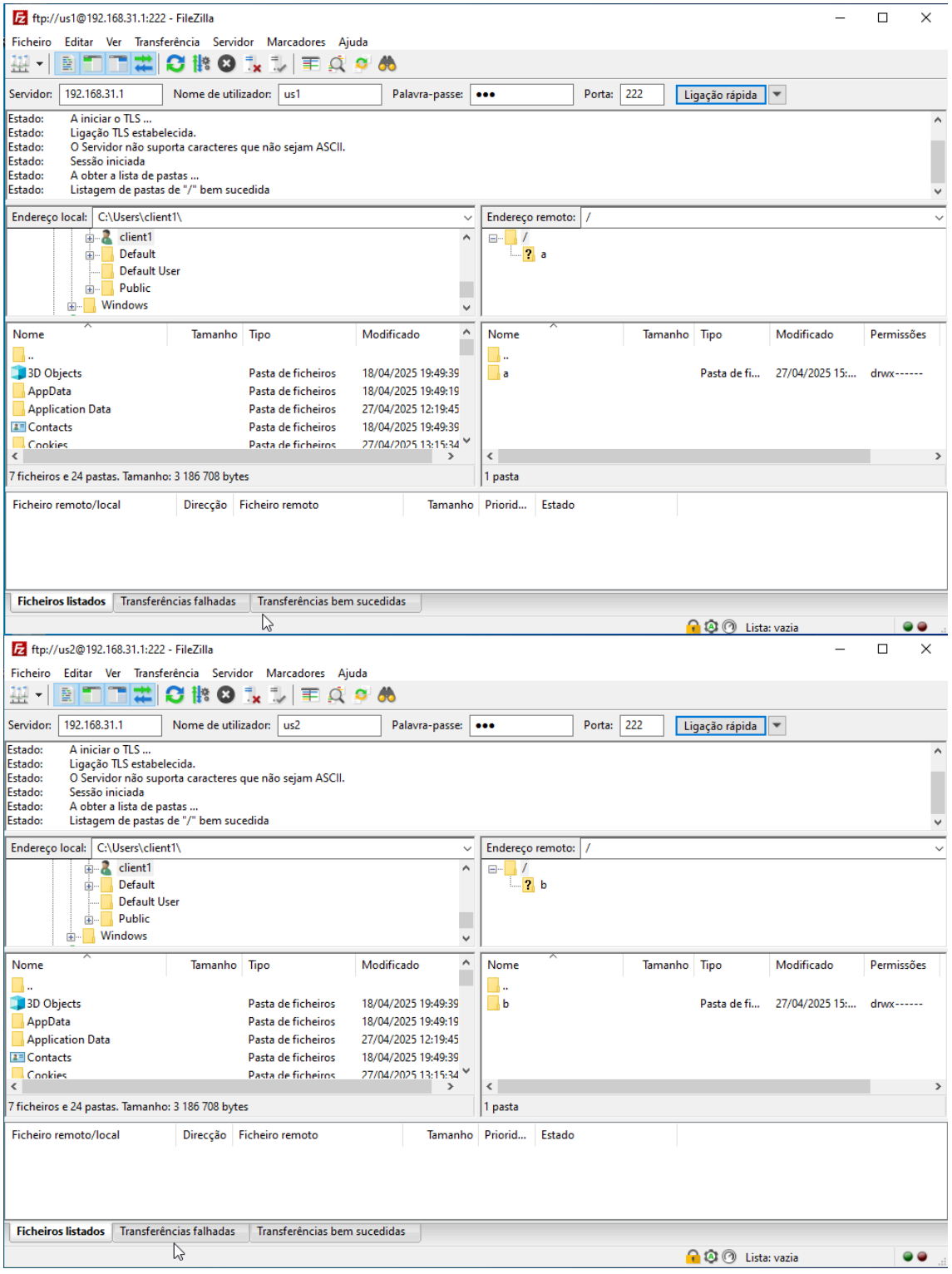
Aqui temos a homefolder do us1



Aqui temos a homefolder do us2



E aqui podemos confirmar que as prints anteriores são da homefolder do us1 e do us2.



Conclusão

A execução deste projeto permitiu compreender e aplicar conceitos fundamentais de administração de servidores em Linux, com especial foco na prestação de serviços de rede essenciais num contexto empresarial. Foi possível configurar com sucesso o DNS com zonas direta e inversa, atribuição dinâmica de IPs com DHCP, acesso remoto seguro com SSH em porta personalizada, serviço FTP com encriptação TLS e partilhas Samba com controlo de permissões avançado. A gestão de utilizadores e grupos, bem como a criação de tarefas automáticas com cron, reforçaram a importância da organização e segurança numa rede. O resultado final é um servidor funcional, robusto e adaptado às necessidades da infraestrutura simulada, pronto a ser integrado numa rede real de pequena ou média dimensão.