

Licenciatura em Engenharia Informática  
Tecnologias de Virtualização 2018/2019

# Trabalho Prático

Gonçalo Vicente 2172131

Filipe Almeida 2171302

Tiago Silva 2160789

## Trabalho Prático

### Índice

Introdução .....	4
Planeamento.....	5
Trabalho realizado .....	7
Ajustes do planeamento inicial .....	7
Funcionalidades .....	8
Conclusão.....	20

### Lista de Figuras

Figura 1 - Esquema de ligações do planeamento .....	6
Figura 2 - Esquema de ligações final.....	7
Figura 3 - Ligação do vSwitch0 (Rede WAN).....	8
Figura 4 - Ligação do vSwitch1 (Rede LAN).....	8
Figura 5 - Ligação dos novos switches criados .....	9
Figura 6 - Datastore com os ISOS.....	9
Figura 7 - IP's das diferentes redes.....	10
Figura 8 - Endereços MAC das diferentes interfaces.....	10
Figura 9 - Configurações de DHCP .....	11
Figura 10 - Regras de firewall das diferentes interfaces .....	11
Figura 11 - Atribuir IP estático ao servidor DNS .....	12
Figura 12 - Atribuir IP estático ao servidor Web .....	12
Figura 13 - Ping entre a PFSense e duas máquinas windows .....	13
Figura 14 - Ping entre a PFSense e o Ubuntu .....	13
Figura 15 - Ping entre a PFSense e Proxmox .....	13
Figura 16 - Ping entre o servidor DNS e a OPT1 .....	14
Figura 17 - Ping entre servidor DNS e a PFSense.....	14
Figura 18 - Ping entre servidor Web e a PFSense .....	14
Figura 19 - Ping entre o servidor Web e a OPT2.....	14
Figura 20 - Ping entre servidor DNS e a máquina virtual Windows .....	14
Figura 21 Ping entre servidor Web e a máquina virtual Windows.....	15
Figura 22 - Ligação das placas de rede no VirtualBox .....	15
Figura 23 - Esquema de rede do computador físico.....	15
Figura 24 - Ping entre o Ubuntu e a PFSense .....	16
Figura 25 - Ping entre o Windows e a PFSense .....	16
Figura 26 - Ping entre o Windows e o Proxmox .....	16
Figura 27 - Ping entre o Windows e o servidor DNS.....	16
Figura 28 - Composição do Proxmox .....	17
Figura 29 - Passos para receber IP via DHCP .....	17
Figura 30 – Ping entre o Proxmox e a PFSense.....	17

# **Introdução**

Este trabalho prático foi realizado no âmbito da disciplina de Tecnologias de Virtualização, do curso de Engenharia Informática da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria.

Este projeto tem como objetivo implementar sistemas de virtualização baseados em sistemas operativos, de forma a demonstrar as capacidades e potencialidades destas soluções, principalmente ao nível de capacidade de controlo e gestão de rede, capacidade da gestão de recursos e interligação de sistemas.

## Planeamento

Para a realização deste projeto iremos ter um cenário que será constituído por dois hypervisores do tipo 1 diferentes e um host. Os hypervisores a utilizar serão o ESXI e o Proxmox.

No hypervisor ESXI iremos criar duas máquinas virtuais, em que uma delas irá servir de servidor Web enquanto a outra irá servir de servidor DNS. Iremos também, criar uma outra máquina virtual para colocar uma PFSense de forma a existir uma firewall entre a Internet e o nosso sistema. Neste hypervisor, iremos ter uma rede NAT que irá estar em Load balancing com a Internet. Iremos ter ainda, três redes internas.

No hypervisor proxmox iremos implementar três servidores dentro do hypervisor, no entanto, estes servidores serão de DHCP, de partilha de ficheiros com Samba e iremos ter também um servidor com o FREENAS, que irá de servir de backups. Todos os servidores irão estar ligados a uma rede em modo bridge.

No host iremos ter um hypervisor do tipo 2. O VirtualBox para criar três máquinas virtuais com os sistemas operativos Windows (Windows 10) e Linux (Ubuntu). Os IP's destes sistemas operativos irão ser atribuídos por DHCP através da PFSense. As três máquinas virtuais irão estar ligadas através de uma rede bridge.

De forma a interligar o Host, o hypervisor Proxmox e o hypervisor ESXI iremos ligar os três a um switch físico.

De forma a ajudar a compreender o cenário que planeamos, realizámos um esquema que está representado na figura 1.

## Trabalho Prático

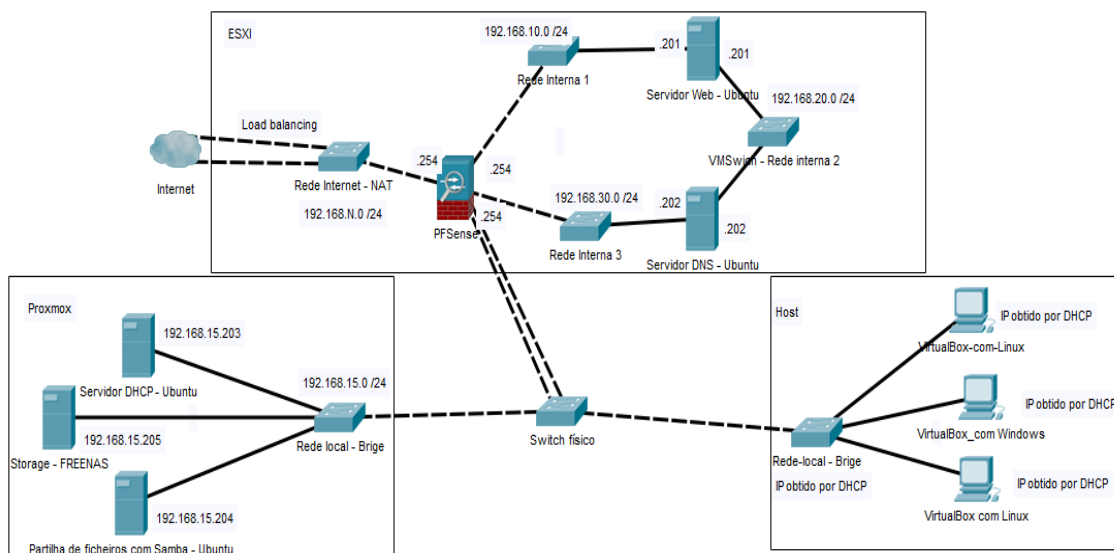


Figura 1 - Esquema de ligações do planeamento

## Trabalho realizado

### Ajustes do planeamento inicial

Inicialmente, tínhamos planeado implementar a solução que se encontra na figura 1, mas tivemos que fazer alguns ajustes a esse planeamento que tinha sido feito inicialmente.

Algo que nos apercebemos aquando da implementação deste projeto foi que não havia necessidade de termos o VMSwitch – Rede Interna 2, razão pela qual optamos por o retirar.

No esquema abaixo, podemos ver o esquema de ligações com os devidos ajustamentos.

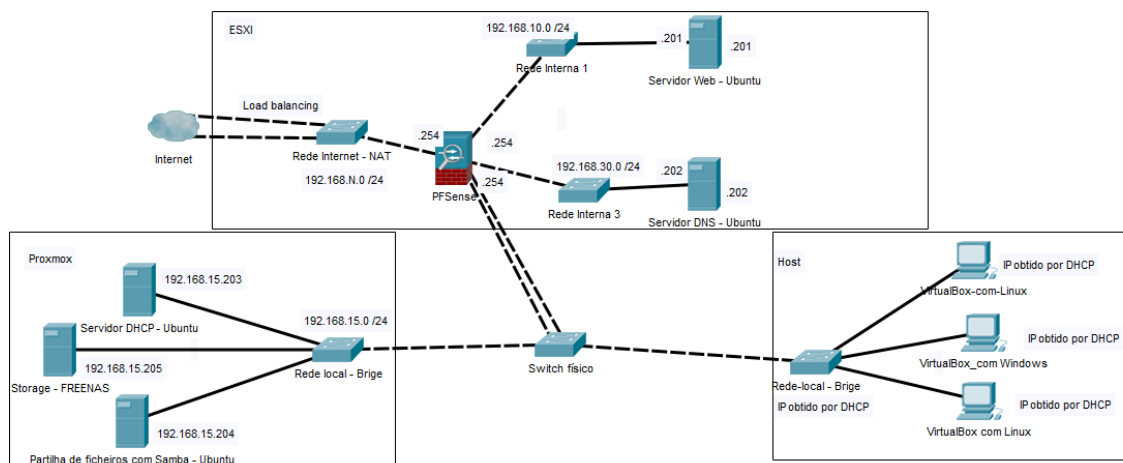


Figura 2 - Esquema de ligações final

## Trabalho Prático

### Funcionalidades

#### ESXI

No ESXi foram adicionados dois switches virtuais aos existentes

Nas figuras abaixo, podemos ver o esquema.

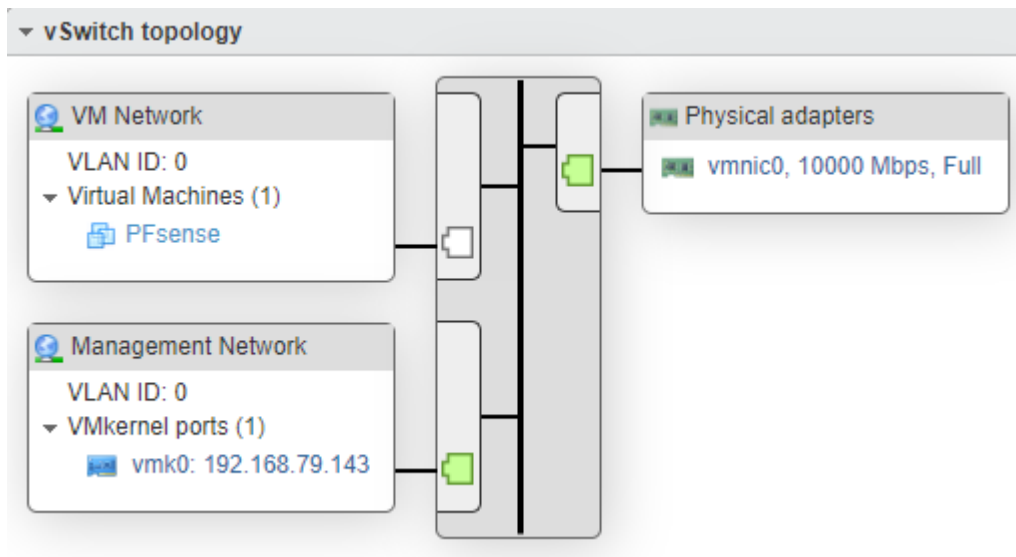


Figura 3 - Ligação do vSwitch0 (Rede WAN)

Nesta ligação, o vSwitch0 possui uma placa NAT e está ligada na porta WAN da PFSense, e é através da placa NAT que o cenário tem acesso à internet.

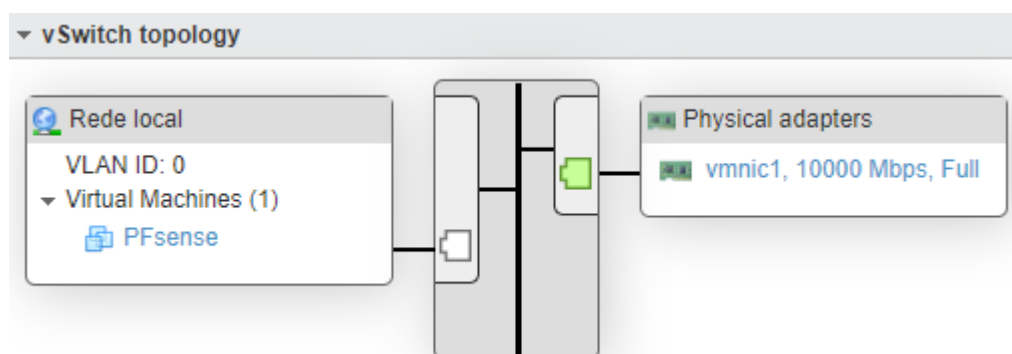


Figura 4 - Ligação do vSwitch1 (Rede LAN)

O vSwitch1 possui uma placa Bridge, este vSwitch permite a comunicação entre hosts e guests.



## Trabalho Prático

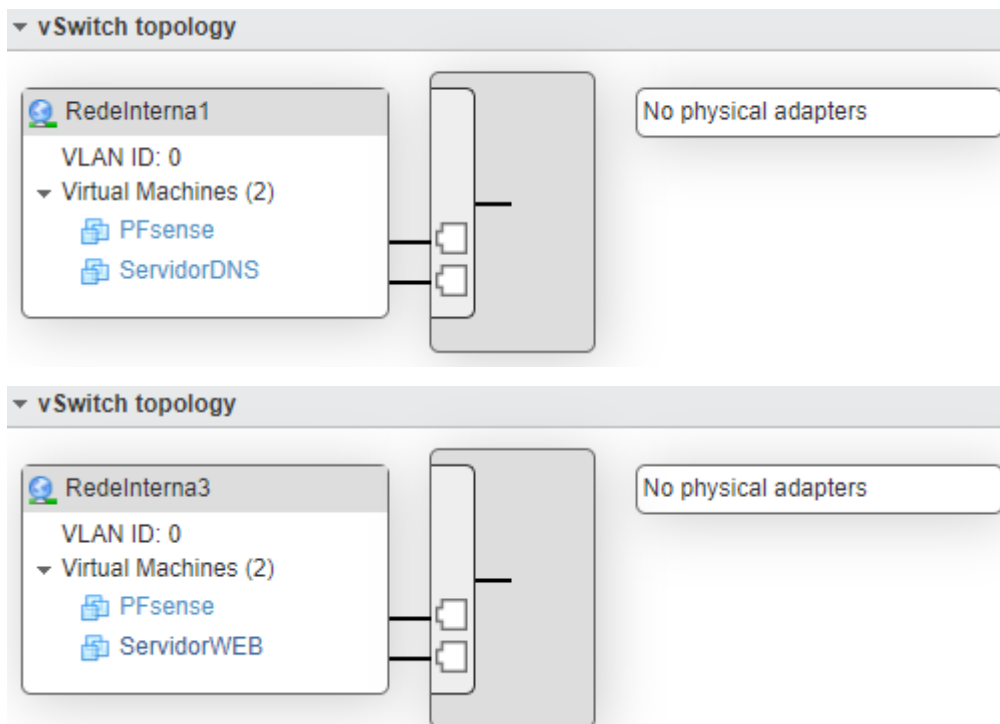


Figura 5 - Ligação dos novos switches criados

Foram adicionados estes dois novos vSwitchs, para interligar os servidores Web e DNS à PFsense, de forma a estarem numa rede diferente.

Para que fosse possível instalar os sistemas operativos para correr no ESXI, foi necessário carregar os ISOS para dentro da Datastore, no nosso caso a Datastore1.

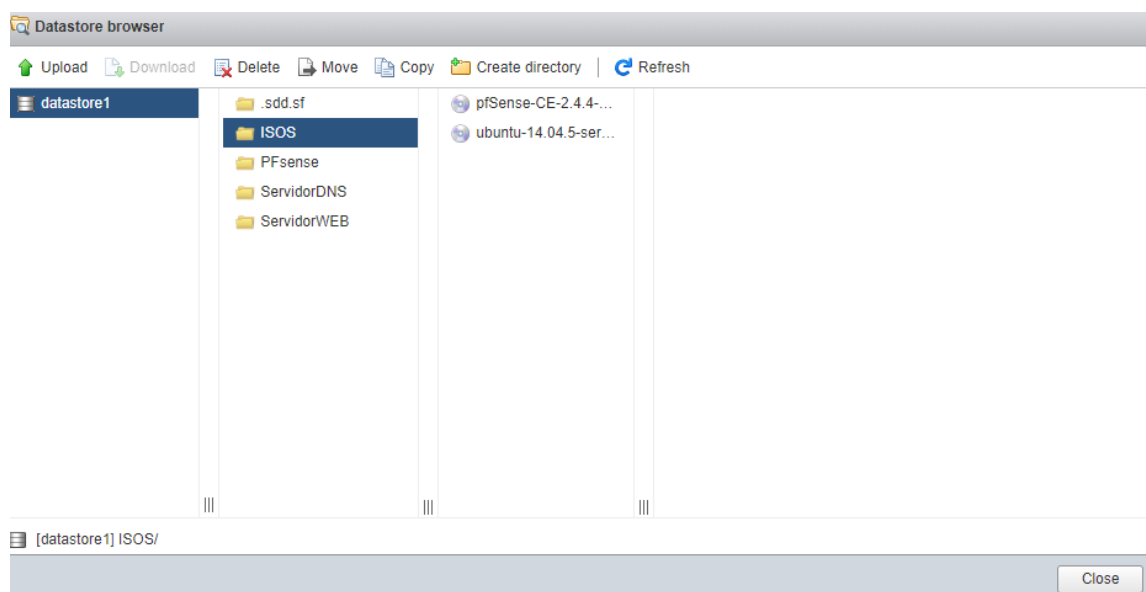


Figura 6 - Datastore com os ISOS

## Trabalho Prático

De seguida, foi adicionada uma nova máquina virtual ao hypervisor ESXI, onde foi instalada a PFSense e depois os dois servidores DNS e Web com os sistemas operativos Ubuntu server 16.04 LTS.

No servidor DNS foi escolhida esse serviço logo na instalação mas no servidor Web foi necessário instalar à posteriori o Apache.

Depois de instalada a PFSense, inserimos mais duas interfaces, a OPT1 e a OPT2 e atribuímos os IP's que pretendíamos. Depois inserimos o IP que definimos como interface de configuração, no nosso caso o 192.168.15.250, no browser para acedermos ao painel de configuração da mesma, onde ativamos o protocolo HTTPS, o DHCP e criamos 3 regras na firewall, uma para cada interface (LAN, OPT1 e OPT2).

```
WAN (wan)      -> em0      -> v4/DHCP4: 192.168.79.144/24
LAN (lan)      -> em1      -> v4: 192.168.15.250/24
OPT1 (opt1)    -> em2      -> v4: 192.168.10.1/24
OPT2 (opt2)    -> em3      -> v4: 192.168.30.1/24
```

Figura 7 - IP's das diferentes redes

Interface	Network port
WAN	em0 (00:0c:29:3d:03:1e)
LAN	em1 (00:0c:29:3d:03:28) <span>Delete</span>
OPT1	em2 (00:0c:29:3d:03:32) <span>Delete</span>
OPT2	em3 (00:0c:29:3d:03:3c) <span>Delete</span>

Figura 8 - Endereços MAC das diferentes interfaces

Era também pretendido que os sistemas operativos recebessem IP's por DHCP. Foi então necessário ativar o DHCP na PFSense e criar a gama que pretendíamos. Escolhemos então que a nossa PFSense iria atribuir IP's por DHCP desde o 192.168.15.100 até ao 192.168.15.241.

## Trabalho Prático

General Options	
Enable	<input checked="" type="checkbox"/> Enable DHCP server on LAN interface
BOOTP	<input type="checkbox"/> Ignore BOOTP queries
Deny unknown clients	<input type="checkbox"/> Only the clients defined below will get DHCP leases from this server.
Ignore denied clients	<input type="checkbox"/> Denied clients will be ignored rather than rejected. This option is not compatible with failover and cannot be enabled when a Failover Peer IP address is configured.
Ignore client identifiers	<input type="checkbox"/> If a client includes a unique identifier in its DHCP request, that UID will not be recorded in its lease. This option may be useful when a client can dual boot using different client identifiers but the same hardware (MAC) address. Note that the resulting server behavior violates the official DHCP specification.
Subnet	192.168.15.0
Subnet mask	255.255.255.0
Available range	192.168.15.1 - 192.168.15.254
Range	<div>192.168.15.100</div> <div>From</div> <div>192.168.15.241</div> <div>To</div>

Figura 9 - Configurações de DHCP

Na imagem abaixo, é possível ver as regras de firewall que tivemos que implementar para que as diferentes máquinas comunicassem entre si.

Firewall / Rules / LAN

Floating WAN LAN OPT1 OPT2

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	✓ 15 / 1.62 MIB	*	*	*	LAN Address	443 80	*	*		Anti-Lockout Rule	
<input type="checkbox"/>	✓ 1 / 234 B	IPv4 *	*	*	*	*	*	none			
<input type="checkbox"/>	✓ 0 / 0 B	IPv4 *	LAN net	*	*	*	*	none		Default allow LAN to any rule	
<input type="checkbox"/>	✓ 0 / 0 B	IPv6 *	LAN net	*	*	*	*	none		Default allow LAN IPv6 to any rule	

Add Add Delete Save Separator

Firewall / Rules / OPT1

Floating WAN LAN OPT1 OPT2

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✓ 0 / 0 B	IPv4 *	*	*	*	*	*	none			

Add Add Delete Save Separator

Firewall / Rules / OPT2

Floating WAN LAN OPT1 OPT2

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✓ 0 / 0 B	IPv4 *	*	*	*	*	*	none			

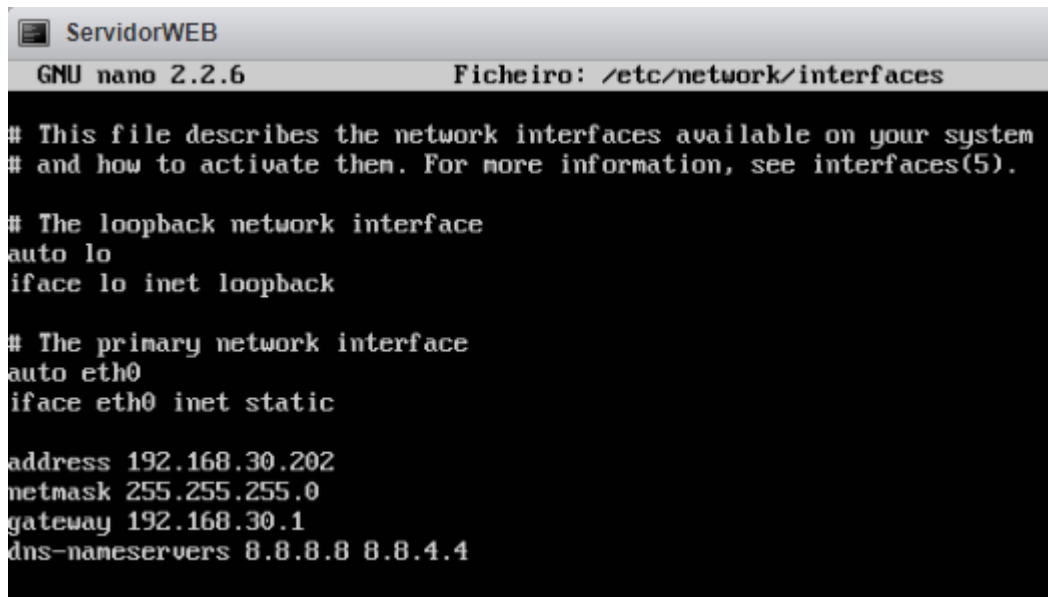
Add Add Delete Save Separator

Figura 10 - Regras de firewall das diferentes interfaces

## Trabalho Prático

Depois, foi ainda necessário configurar os servidores Web e DNS para que ficassem a comunicar com todos os outros sistemas operativos e com o hypervisor Proxmox.

Para tal, tivemos que ir ao ficheiro que se encontra em `/etc/network/interfaces` e adicionar o IP que pretendíamos, a máscara de rede, o gateway e o DNS. As imagens abaixo mostram esta situação.



```
GNU nano 2.2.6          Ficheiro: /etc/network/interfaces

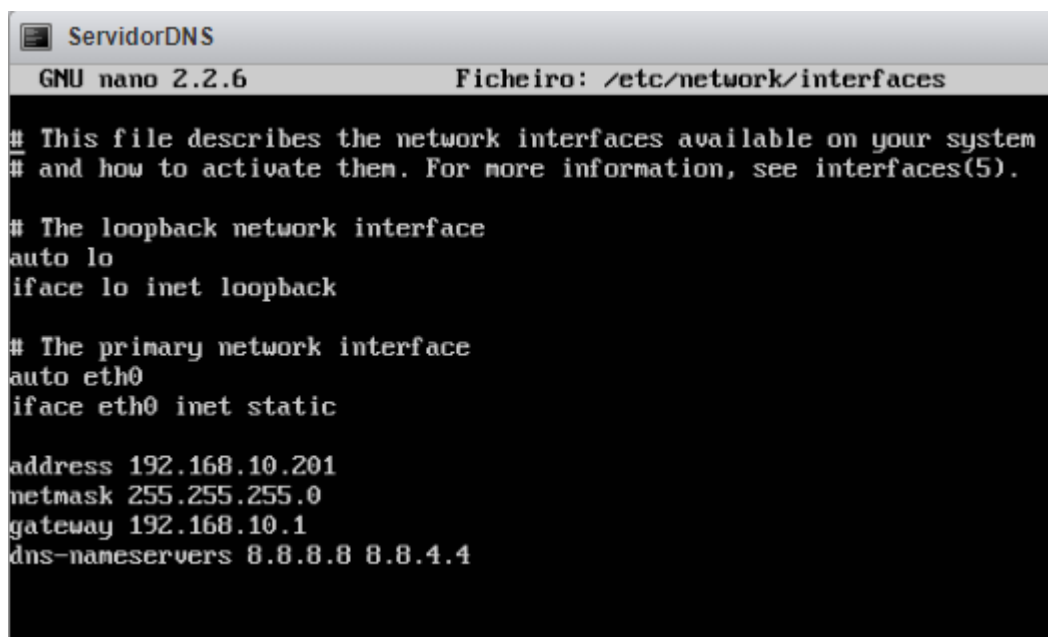
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static

address 192.168.30.202
netmask 255.255.255.0
gateway 192.168.30.1
dns-nameservers 8.8.8.8 8.8.4.4
```

Figura 12 - Atribuir IP estático ao servidor Web



```
GNU nano 2.2.6          Ficheiro: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static

address 192.168.10.201
netmask 255.255.255.0
gateway 192.168.10.1
dns-nameservers 8.8.8.8 8.8.4.4
```

Figura 11 - Atribuir IP estático ao servidor DNS

## Trabalho Prático

Posto tudo isto, era necessário verificar se era possível estabelecer comunicação com os diferentes sistemas operativos. Nas imagens abaixo é possível ver ping que foram efetuados entre a PFSense, os servidores que estavam no ESXI, o Proxmox e os sistemas operativos que estavam no Host.

```
[2.4.4-RELEASE][root@pfSense.localdomain]/root: ping 192.168.15.102
PING 192.168.15.102 (192.168.15.102): 56 data bytes
64 bytes from 192.168.15.102: icmp_seq=0 ttl=64 time=1.467 ms
64 bytes from 192.168.15.102: icmp_seq=1 ttl=64 time=1.924 ms
64 bytes from 192.168.15.102: icmp_seq=2 ttl=64 time=1.758 ms
64 bytes from 192.168.15.102: icmp_seq=3 ttl=64 time=2.048 ms
^C
--- 192.168.15.102 ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.467/1.799/2.048/0.218 ms
```

Figura 14 - Ping entre a PFSense e o Ubuntu

```
[2.4.4-RELEASE][root@pfSense.localdomain]/root: ping 192.168.15.103
PING 192.168.15.103 (192.168.15.103): 56 data bytes
64 bytes from 192.168.15.103: icmp_seq=0 ttl=128 time=1.464 ms
64 bytes from 192.168.15.103: icmp_seq=1 ttl=128 time=1.742 ms
64 bytes from 192.168.15.103: icmp_seq=2 ttl=128 time=1.437 ms
64 bytes from 192.168.15.103: icmp_seq=3 ttl=128 time=1.495 ms
^C
--- 192.168.15.103 ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.437/1.534/1.742/0.122 ms
```

```
[2.4.4-RELEASE][root@pfSense.localdomain]/root: ping 192.168.15.101
PING 192.168.15.101 (192.168.15.101): 56 data bytes
64 bytes from 192.168.15.101: icmp_seq=0 ttl=128 time=2.204 ms
64 bytes from 192.168.15.101: icmp_seq=1 ttl=128 time=1.481 ms
64 bytes from 192.168.15.101: icmp_seq=2 ttl=128 time=1.651 ms
64 bytes from 192.168.15.101: icmp_seq=3 ttl=128 time=1.177 ms
^C
--- 192.168.15.101 ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
```

Figura 13 - Ping entre a PFSense e duas máquinas windows

```
[2.4.4-RELEASE][root@pfSense.localdomain]/root: ping 192.168.15.105
PING 192.168.15.105 (192.168.15.105): 56 data bytes
64 bytes from 192.168.15.105: icmp_seq=0 ttl=64 time=1.473 ms
64 bytes from 192.168.15.105: icmp_seq=1 ttl=64 time=1.525 ms
64 bytes from 192.168.15.105: icmp_seq=2 ttl=64 time=2.164 ms
64 bytes from 192.168.15.105: icmp_seq=3 ttl=64 time=2.221 ms
^C
--- 192.168.15.105 ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.473/1.846/2.221/0.348 ms
```

Figura 15 - Ping entre a PFSense e Proxmox

## Trabalho Prático

```
serverdns@serverdns:~$ ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=0.462 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=0.407 ms
64 bytes from 192.168.10.1: icmp_seq=3 ttl=64 time=0.542 ms
64 bytes from 192.168.10.1: icmp_seq=4 ttl=64 time=0.406 ms
^C
--- 192.168.10.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2998ms
rtt min/avg/max/mdev = 0.406/0.454/0.542/0.057 ms
```

Figura 16 - Ping entre o servidor DNS e a OPT1

```
serverdns@serverdns:~$ ping 192.168.15.250
PING 192.168.15.250 (192.168.15.250) 56(84) bytes of data.
64 bytes from 192.168.15.250: icmp_seq=1 ttl=64 time=0.571 ms
64 bytes from 192.168.15.250: icmp_seq=2 ttl=64 time=0.464 ms
64 bytes from 192.168.15.250: icmp_seq=3 ttl=64 time=0.415 ms
64 bytes from 192.168.15.250: icmp_seq=4 ttl=64 time=0.410 ms
^C
--- 192.168.15.250 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
```

Figura 17 - Ping entre servidor DNS e a PFSense

```
serverdns@serverdns:~$ ping 192.168.15.101
PING 192.168.15.101 (192.168.15.101) 56(84) bytes of data.
64 bytes from 192.168.15.101: icmp_seq=1 ttl=127 time=3.26 ms
64 bytes from 192.168.15.101: icmp_seq=2 ttl=127 time=2.23 ms
64 bytes from 192.168.15.101: icmp_seq=3 ttl=127 time=1.93 ms
64 bytes from 192.168.15.101: icmp_seq=4 ttl=127 time=1.78 ms
^C
--- 192.168.15.101 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 1.783/2.305/3.266/0.579 ms
```

Figura 20 - Ping entre servidor DNS e a máquina virtual Windows

```
serverweb@serverweb:~$ ping 192.168.30.1
PING 192.168.30.1 (192.168.30.1) 56(84) bytes of data.
64 bytes from 192.168.30.1: icmp_seq=1 ttl=64 time=1.02 ms
64 bytes from 192.168.30.1: icmp_seq=2 ttl=64 time=0.357 ms
64 bytes from 192.168.30.1: icmp_seq=3 ttl=64 time=0.582 ms
64 bytes from 192.168.30.1: icmp_seq=4 ttl=64 time=0.866 ms
^C
--- 192.168.30.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 0.357/0.707/1.024/0.257 ms
serverweb@serverweb:~$
```

Figura 19 - Ping entre o servidor Web e a OPT2

```
serverweb@serverweb:~$ ping 192.168.15.250
PING 192.168.15.250 (192.168.15.250) 56(84) bytes of data.
64 bytes from 192.168.15.250: icmp_seq=1 ttl=64 time=0.579 ms
64 bytes from 192.168.15.250: icmp_seq=2 ttl=64 time=0.528 ms
64 bytes from 192.168.15.250: icmp_seq=3 ttl=64 time=0.471 ms
64 bytes from 192.168.15.250: icmp_seq=4 ttl=64 time=0.514 ms
^C
--- 192.168.15.250 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.471/0.523/0.579/0.038 ms
```

Figura 18 - Ping entre servidor Web e a PFSense

## Trabalho Prático

```
serverweb@serverweb:~$ ping 192.168.15.101
PING 192.168.15.101 (192.168.15.101) 56(84) bytes of data.
64 bytes from 192.168.15.101: icmp_seq=1 ttl=127 time=2.78 ms
64 bytes from 192.168.15.101: icmp_seq=2 ttl=127 time=2.76 ms
64 bytes from 192.168.15.101: icmp_seq=3 ttl=127 time=2.84 ms
64 bytes from 192.168.15.101: icmp_seq=4 ttl=127 time=2.59 ms
^C
--- 192.168.15.101 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3008ms
rtt min/avg/max/mdev = 2.599/2.749/2.842/0.090 ms
```

Figura 21 Ping entre servidor Web e a máquina virtual Windows

### Host

No Host estão instaladas 3 máquinas virtuais (2 com o sistema operativo windows 10 e 1 com Ubuntu 18.04) que simulam a parte do cliente.

Todas as máquinas virtuais fazem ligação do tipo bridge a partir duma placa de rede que faz ligação ao switch físico. Todas as máquinas obtêm endereços IPv4 a através de DHCP, que vem da PFSense.

Nas imagens abaixo podemos ver o esquema de ligação de rede do Host.



Figura 22 - Ligação das placas de rede no VirtualBox

De forma a que fosse possível adicionar uma placa em bridge ao computador físico, foi necessário criar uma ligação bridge entre a placa Ethernet e a placa Host-Only do hypervisor do tipo 2 que estamos a utilizar, no nosso caso VirtualBox.

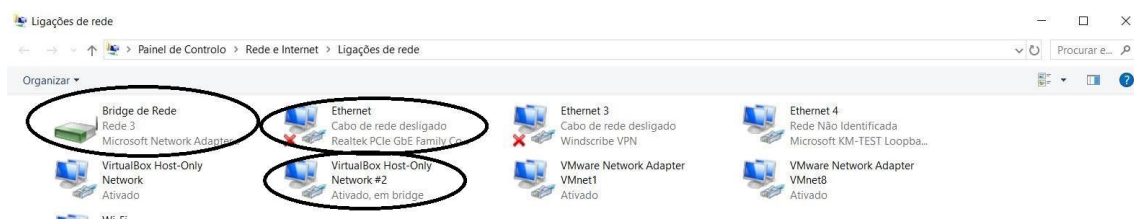


Figura 23 - Esquema de rede do computador físico



## Trabalho Prático

Posto tudo isto, era necessário verificar se era possível estabelecer comunicação com a PFSense. Nas imagens abaixo é possível ver pings que foram efetuados entre os diferentes sistemas operativos que fazem parte do Host, a PFSense e o Proxmox.

```
filipe@filipe-VirtualBox:~$ ping 192.168.15.250 -c 4
PING 192.168.15.250 (192.168.15.250) 56(84) bytes of data.
64 bytes from 192.168.15.250: icmp_seq=1 ttl=64 time=2.04 ms
64 bytes from 192.168.15.250: icmp_seq=2 ttl=64 time=2.69 ms
64 bytes from 192.168.15.250: icmp_seq=3 ttl=64 time=2.14 ms
64 bytes from 192.168.15.250: icmp_seq=4 ttl=64 time=2.51 ms

--- 192.168.15.250 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 2.049/2.352/2.697/0.271 ms
```

Figura 24 - Ping entre o Ubuntu e a PFSense

```
Pinging 192.168.15.250 with 32 bytes of data:
Reply from 192.168.15.250: bytes=32 time=1ms TTL=64
Reply from 192.168.15.250: bytes=32 time=1ms TTL=64
Reply from 192.168.15.250: bytes=32 time=1ms TTL=64
Reply from 192.168.15.250: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.15.250:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

Figura 25 - Ping entre o Windows e a PFSense

```
Pinging 192.168.15.105 with 32 bytes of data:
Reply from 192.168.15.105: bytes=32 time=3ms TTL=64
Reply from 192.168.15.105: bytes=32 time=1ms TTL=64
Reply from 192.168.15.105: bytes=32 time=2ms TTL=64
Reply from 192.168.15.105: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.15.105:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 3ms, Average = 1ms
```

Figura 26 - Ping entre o Windows e o Proxmox

```
Pinging 192.168.10.201 with 32 bytes of data:
Reply from 192.168.10.201: bytes=32 time=1ms TTL=63
Reply from 192.168.10.201: bytes=32 time=1ms TTL=63
Reply from 192.168.10.201: bytes=32 time=1ms TTL=63
Reply from 192.168.10.201: bytes=32 time=3ms TTL=63

Ping statistics for 192.168.10.201:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

Figura 27 - Ping entre o Windows e o servidor DNS



## Trabalho Prático

### Proxmox

No proxmox foram adicionadas duas máquinas virtuais. Uma das máquinas tinha o FREENAS e a outra era um servidor de Samba. Ambos estes serviços tinham como destino serem utilizados pelos sistemas operativos que se encontram no Host.

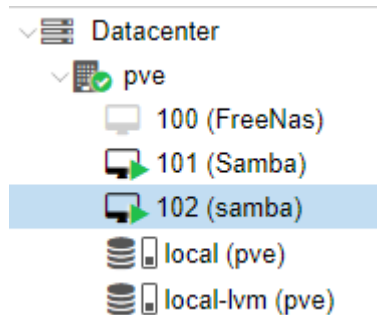


Figura 28 - Composição do Proxmox

Ao Proxmox, os IP's eram atribuídos por DHCP através da PFSense. Para tal, foi necessário colocar a placa de rede do Proxmox em modo bridge e as máquinas virtuais dentro deste também se encontravam em modo bridge, mas recebiam IP via DHCP.

Nas imagens abaixo, é possível ver os passos que fizemos de forma a obter IP's via DHCP.

enp0s3	Network Device	Yes	No	No				
vmbr0	Linux Bridge	Yes	Yes	No	enp0s3	192.168.15.105	255.255.255.0	192.168.15.250

```
root@pve:~# dhclient
root@pve:~# dhclient vmbr0
TNET: INK: answer: File: quiete

3: vmbr0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group
link/ether 08:00:27:63:6b:51 brd ff:ff:ff:ff:ff:ff
inet 192.168.15.105/24 brd 192.168.15.255 scope global vmbr0
    valid_lft forever preferred_lft forever
inet6 fe80::a00:27ff:fe63:6b51/64 scope link
    valid_lft forever preferred_lft forever
```

Figura 29 - Passos para receber IP via DHCP

Depois de já estarem a ser atribuídos IP's por DHCP era necessário verificar se já era possível fazer ping com a PFSense. Na imagem abaixo podemos ver que está a funcionar corretamente.

```
root@pve:~# ping 192.168.15.250
PING 192.168.15.250 (192.168.15.250) 56(84) bytes of data.
64 bytes from 192.168.15.250: icmp_seq=1 ttl=64 time=1.07 ms
64 bytes from 192.168.15.250: icmp_seq=2 ttl=64 time=2.29 ms
64 bytes from 192.168.15.250: icmp_seq=3 ttl=64 time=1.22 ms
64 bytes from 192.168.15.250: icmp_seq=4 ttl=64 time=1.35 ms
64 bytes from 192.168.15.250: icmp_seq=5 ttl=64 time=1.83 ms
```

Figura 30 – Ping entre o Proxmox e a PFSense

## Trabalho Prático

### Dificuldades encontradas

#### ESXI

Na realização deste projeto, deparamo-nos com algumas dificuldades no ESXI.

Uma das dificuldades foi o facto de, depois de já termos acedido ao painel de configuração da PFSense, deixarmos de conseguir voltar a entrar sem que tenhamos feito nenhuma alteração. Depois de muito tentar, mesmo com a ajuda do professor Nuno Veiga, conseguimos voltar a ter acesso ao painel de configuração da PFSense. Este problema tornou-se grave uma vez que, na altura dos acontecimentos, ainda não tínhamos adicionado as regras de firewall para as interfaces OPT1 e OPT2, e isso não é possível de fazer através da linha de comandos da PFSense, à qual sempre tivemos acesso.

Outra dificuldade que encontramos deveu-se ao facto de utilizarmos um switch que também fazia a função de router. Por defeito, a opção de DHCP está ativa, mas nós pretendíamos atribuir endereços IP's através da PFSense, o que iria criar um conflito. Sendo assim, foi necessário aceder ao painel de configuração do router através do IP 192.168.1.1 e desabilitar a opção do DHCP.

#### Host

No host, a dificuldade que tivemos deveu-se ao facto de que para serem feitas as ligações bridge a partir do VirtualBox é necessário associar uma ligação bridge entre a placa de rede física e uma placa de rede virtual implementada pelo VirtualBox, e então atribuir a cada uma das máquinas virtuais um network adapter do tipo HO (host-only), associada à placa de rede virtual que associámos à ligação bridge. Nas figuras 22 e 23 podemos ver esses passos que tivemos que fazer.

#### Proxmox

No proxmox foi onde tivemos mais dificuldades para a realização deste projeto. Inicialmente, não estamos a conseguir instalar o Proxmox devido a um erro que nos aparecia.

## Trabalho Prático

Resolvemos instalar uma máquina virtual com o Windows 10 Pro, de forma a utilizarmos o Hyper-V, a conselho do professor Nuno Veiga, mas devido a uns problemas que serão descritos mais à frente neste relatório, decidimos usar essa máquina virtual para aceder ao Proxmox, ao invés de ser no computador físico.

O problema que tivemos com o Hyper-V deu-se com o facto de não aparecer o serviço “Virtual Machine Management” na lista de serviços o que nos impossibilitava colocar em modo “Running”, que era essencial para podermos aceder às máquinas virtuais que iríamos instalar no interior do Hyper-V.

Em relação ao Proxmox tivemos mais um problema, desta feita que não conseguimos resolver. A partir da máquina virtual do Proxmox conseguíamos aceder a todas as outras máquinas, mas as máquinas virtuais que estavam a correr sobre o Proxmox não conseguiam ter acesso a todas as outras. Pensamos ter todas as configurações corretamente, mas mesmo assim não conseguimos fazer ping com as outras. Esta situação impossibilita-nos utilizar os serviços que instalamos.

## Conclusão

Por fim, com a realização deste projeto, podemos concluir que a virtualização é uma opção muito válida e a ter em conta, além de que se encontra em constante crescimento e inovação.

Contudo, nem em todas as situações é útil a virtualização, depende do contexto. No nosso caso, consideramos muito útil a virtualização, pois conseguimos poupar recursos físicos e utilizar apenas os recursos virtuais. Além disso, conseguimos ter capacidade de controlo e gestão de rede, capacidade da gestão de recursos e interligação de sistemas, tudo através da virtualização.