

Lab Week 1

Table of Contents

Atividade 1.1 - PREPARAÇÃO DE AMBIENTES	2
Atividade 1.2 - Conectividade e SSH	18

Atividade 1.1 - PREPARAÇÃO DE AMBIENTES

Objetivo

- Neste laboratório, preparei os ambientes que seriam utilizados nos próximos laboratórios.
- Instalei o Termux e executei o comando ping entre meu celular e a VM.
- Ao concluir este laboratório, fui capaz de fazer ping entre as VMs.

Detalhamento do Passo a Passo (VirtualBox no Fedora)

1. Instalação do VirtualBox

Para instalar o VirtualBox no Fedora, utilizei o pacote `.rpm` e o gerenciador de pacotes `dnf`.

2. Resolução de Conflitos com KVM (VERR_VMX_IN_VMX_ROOT_MODE)

Se ao tentar iniciar uma máquina virtual eu encontrei o erro `VERR_VMX_IN_VMX_ROOT_MODE`, isso indicou um conflito com o módulo KVM (Kernel-based Virtual Machine) que estava utilizando os recursos de virtualização de hardware. Para resolver, foi necessário descarregar os módulos do KVM.

Primeiro, verifiquei se os módulos KVM estavam carregados:

```
lsmod | grep kvm
```

Se a saída mostrou `kvm_intel`, `kvm` ou `irqbypass`, descarreguei-os na seguinte ordem:

```
sudo rmmod kvm_intel  
sudo rmmod irqbypass  
sudo rmmod kvm
```

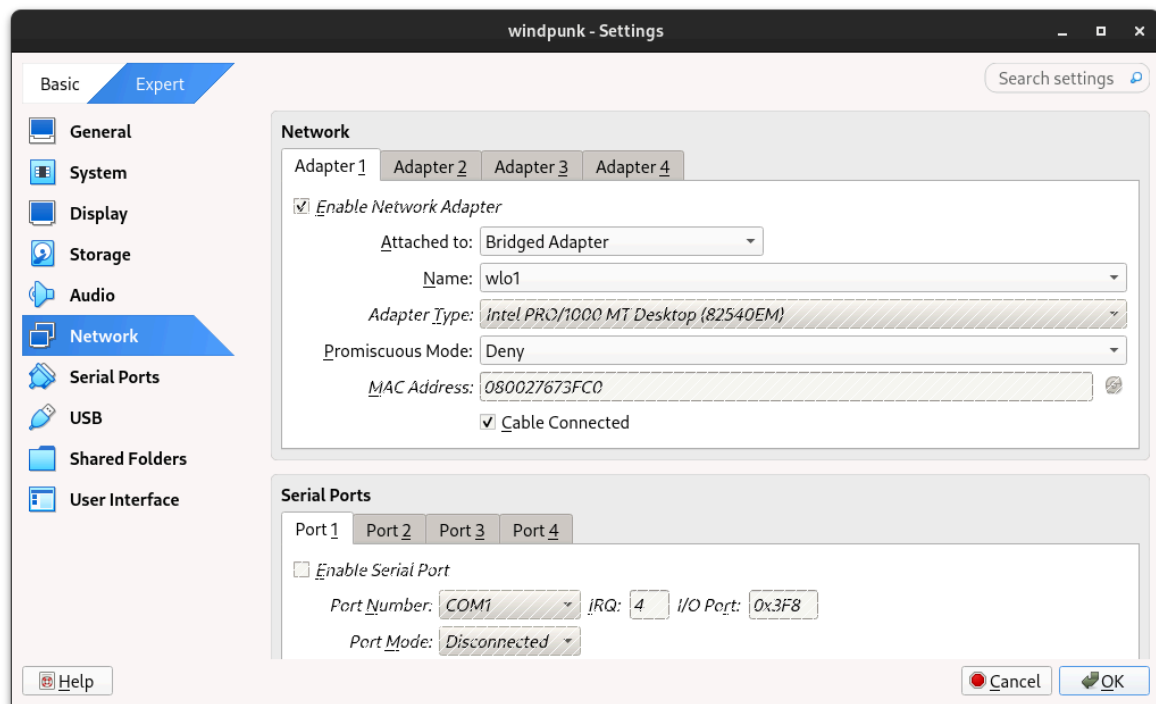
Após descarregar os módulos, tentei iniciar minha máquina virtual novamente.

3. Criação das Máquinas Virtuais (VMs)

Utilizando o VirtualBox no Fedora, criei duas máquinas virtuais:

VM1: Windows 10

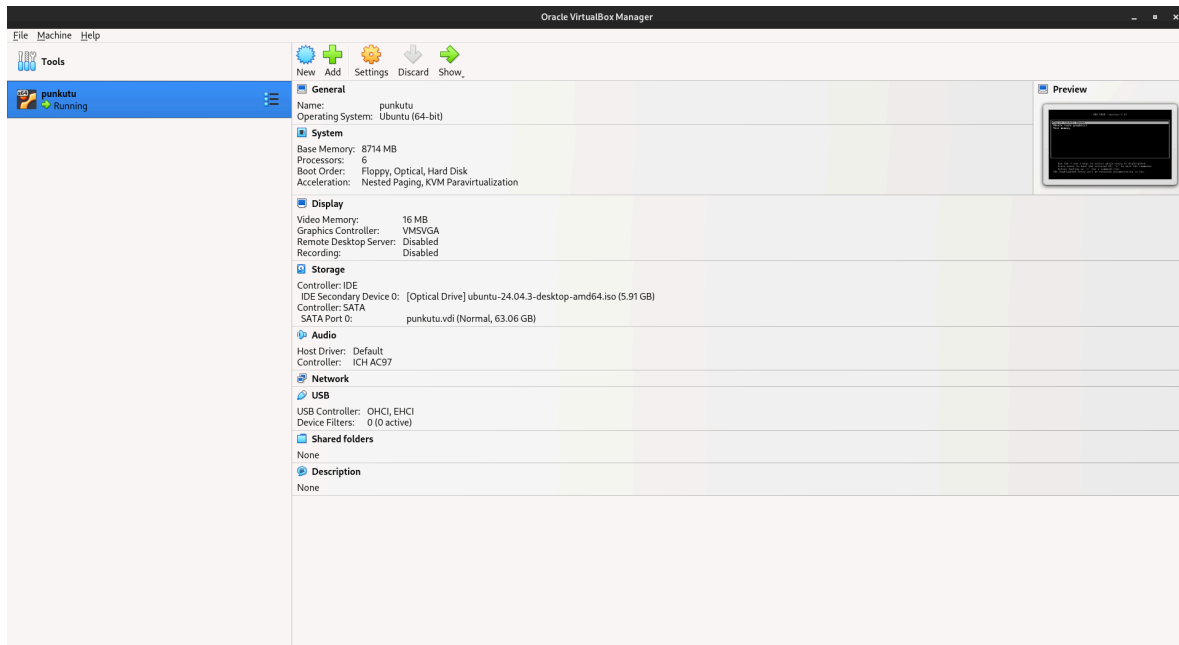
- Configurei as especificações de hardware conforme a necessidade (RAM, CPU, Disco).
- Instalei o sistema operacional Windows 10.



Captura de tela da máquina virtual Windows 10 após a instalação.

VM2: Ubuntu

- Configurei as especificações de hardware conforme a necessidade (RAM, CPU, Disco).
- Instalei o sistema operacional Ubuntu.

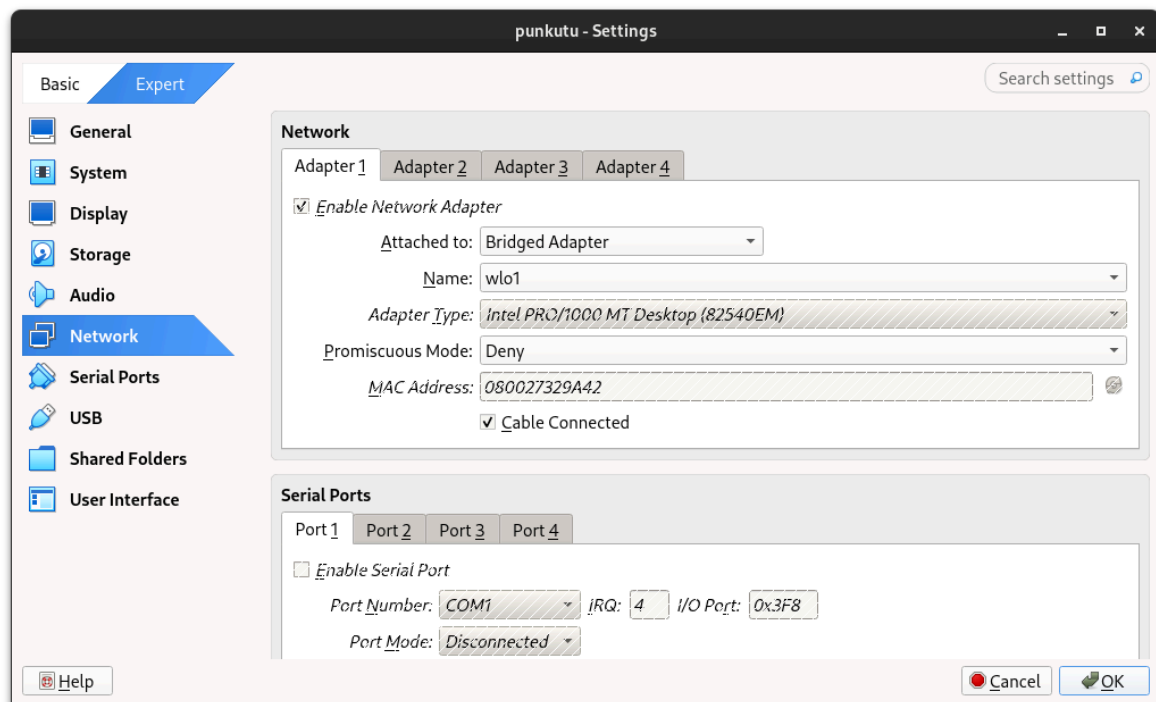


Captura de tela do VirtualBox mostrando as máquinas virtuais Windows 10 e Ubuntu criadas.

2. Configuração de Rede no Modo Ponte (Bridge)

Para ambas as VMs (Windows 10 e Ubuntu), fiz o seguinte:

- Acessei as configurações da VM no VirtualBox.
- Fui para a seção "Rede".
- No "Adaptador 1", selecionei "Placa em modo Bridge".
- Escolhi a interface de rede do meu Fedora que estava conectada à internet (ex: `enp0s3`, `wlp2s0`). Para identificar a interface correta, usei o comando `ip a` no terminal e procurei por uma interface com um endereço IP e status `UP` (ativa).



Configuração de rede em modo ponte (Bridge) no VirtualBox para a VM Ubuntu.

3. Configuração da Interface de Rede no Modo DHCP

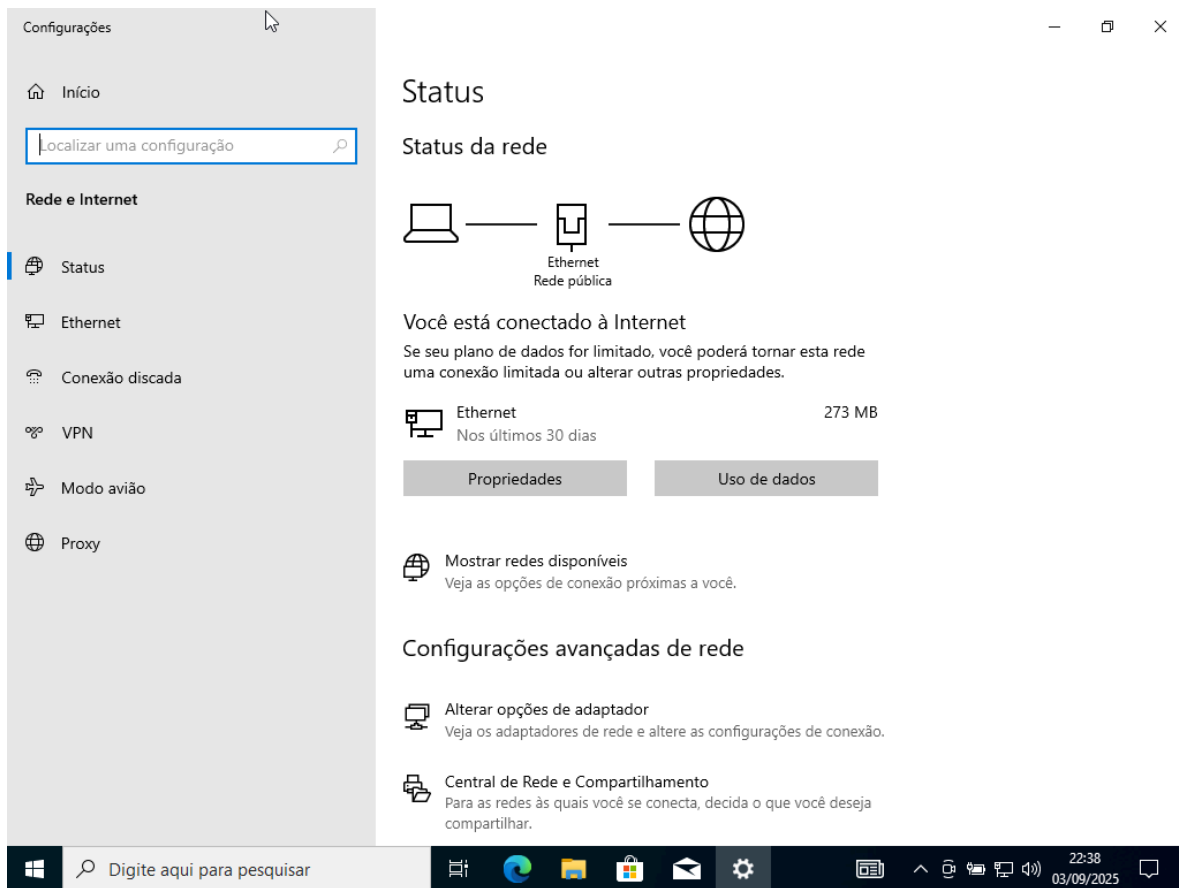
VM1: Windows 10

- No Windows 10, verifiquei se a configuração de rede estava como "Obter um endereço IP automaticamente" (DHCP). Geralmente, esta é a configuração padrão.

Para verificar e configurar a rede DHCP no Windows 10, segui estes passos:

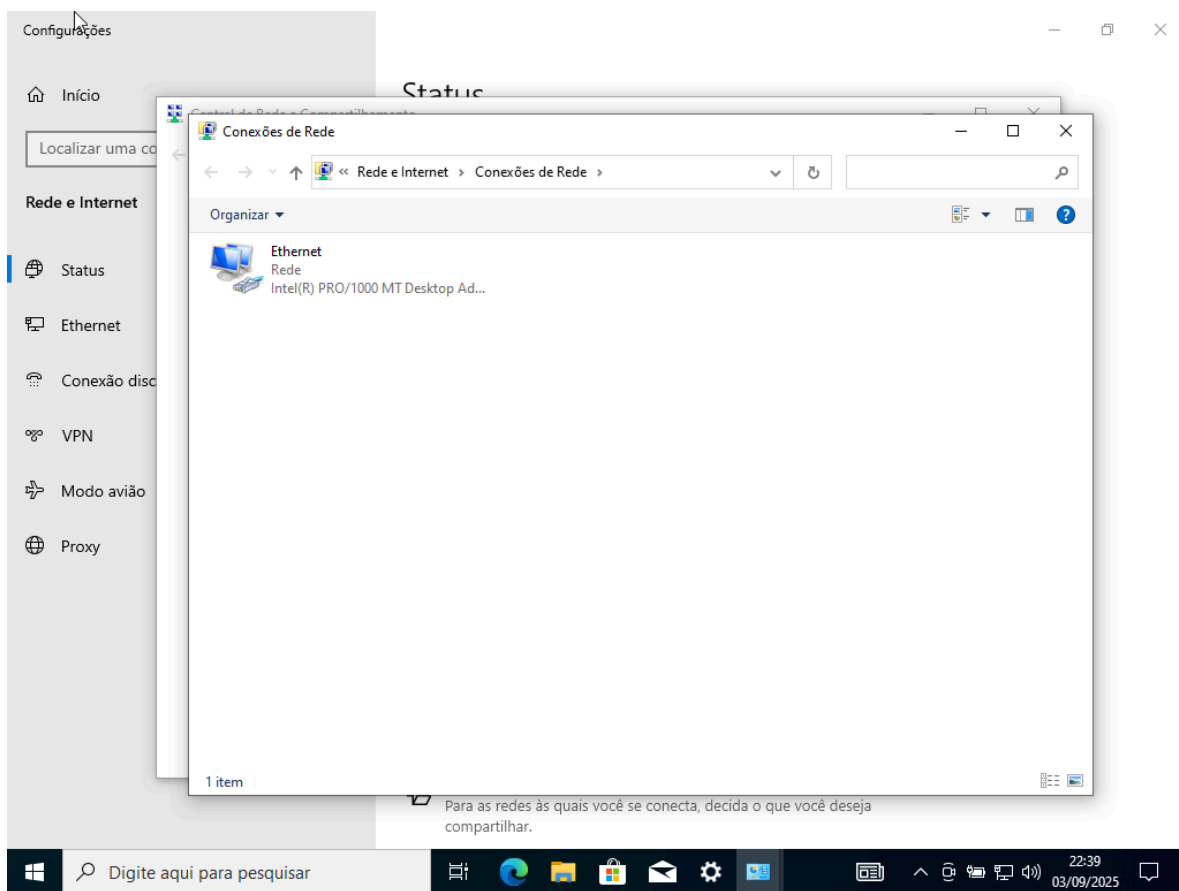
Via Interface Gráfica

1. **Abri as Configurações de Rede:** Cliquei com o botão direito do mouse no ícone de rede na barra de tarefas (canto inferior direito) e selecionei "Abrir Configurações de Rede e Internet".
2. **Acesse as Propriedades do Adaptador:** Na janela de Configurações, cliquei em "Alterar opções de adaptador" (ou "Central de Rede e Compartilhamento" -> "Alterar configurações do adaptador").



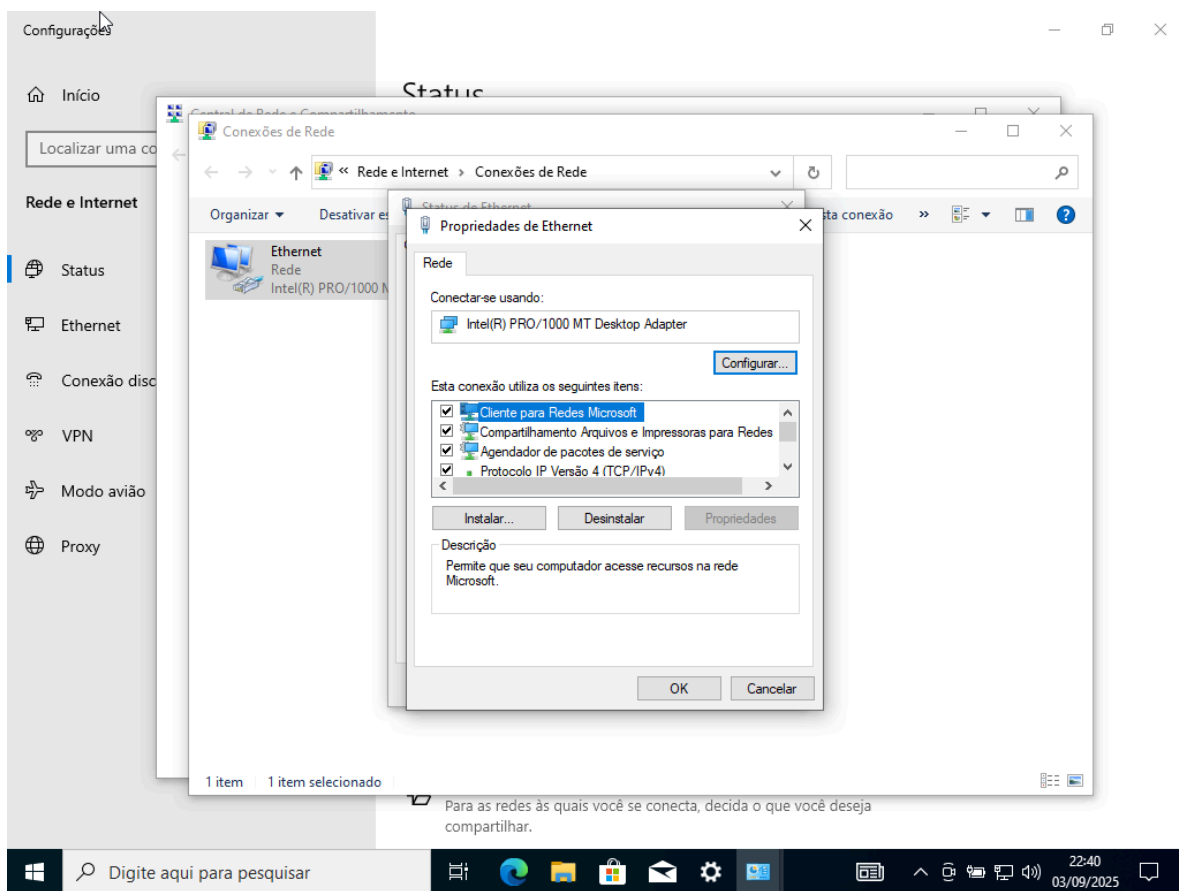
Captura de tela da janela "Conexões de Rede" no Windows 10.

3. **Selecionei o Adaptador de Rede:** Cliquei com o botão direito do mouse no adaptador de rede que estava usando (geralmente "Ethernet" ou "Wi-Fi") e selecionei "Propriedades".



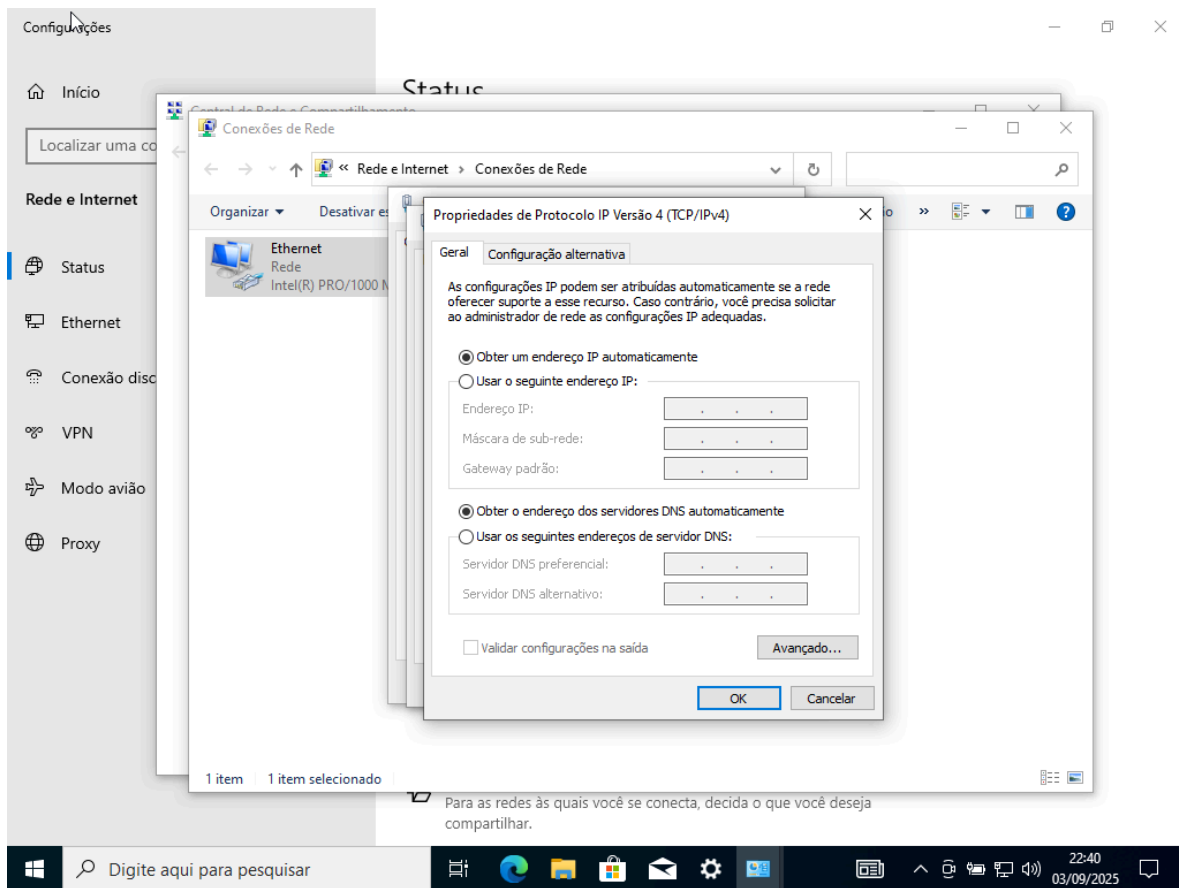
Captura de tela das propriedades do adaptador de rede no Windows 10.

4. **Configurei o Protocolo IPv4:** Na janela de Propriedades do adaptador, selecionei "Protocolo IP Versão 4 (TCP/IPv4)" e cliquei em "Propriedades".



Captura de tela das propriedades do Protocolo IP Versão 4 (TCP/IPv4) no Windows 10.

5. **Habilitei o DHCP:** Certifiquei-me de que as opções "Obter um endereço IP automaticamente" e "Obter o endereço do servidor DNS automaticamente" estavam selecionadas. Cliquei em "OK" para fechar as janelas.



Captura de tela das propriedades do Protocolo IP Versão 4 (TCP/IPv4) com DHCP habilitado no Windows 10.

i Verificação:

Após esses passos, confirmei que o Windows 10 estava configurado para obter um endereço IP via DHCP.

VM2: Ubuntu

- No Ubuntu, configurei a interface de rede para obter um endereço IP via DHCP.
- No Ubuntu, a configuração de rede via DHCP geralmente foi automática. Se necessário, verifiquei ou configurei através das "Configurações de Rede" do sistema ou, via linha de comando, utilizando `netplan`.

Para verificar e configurar a rede DHCP no Ubuntu, segui estes passos:

Via Linha de Comando (Netplan)

No Ubuntu, a configuração de rede via DHCP é gerenciada principalmente pelo `netplan` a partir da versão 17.10.

1. **Encontrei o arquivo de configuração:** Os arquivos de configuração do `netplan` estão em `/etc/netplan/`. No meu caso, utilizei o arquivo `50-cloud-init.yaml`. Eu listei os arquivos com:

```
ls /etc/netplan/
```

2. **Verifiquei o arquivo de configuração:** Abri o arquivo `.yaml` (por exemplo, `50-cloud-init.yaml`) com um editor de texto como `nano` ou `vim`.

```
sudo nano /etc/netplan/50-cloud-init.yaml
```

Certifiquei-me de que a configuração para minha interface de rede (ex: `enp0s3`) estava definida para DHCP. A linha `dhcp4: true` confirma isso.

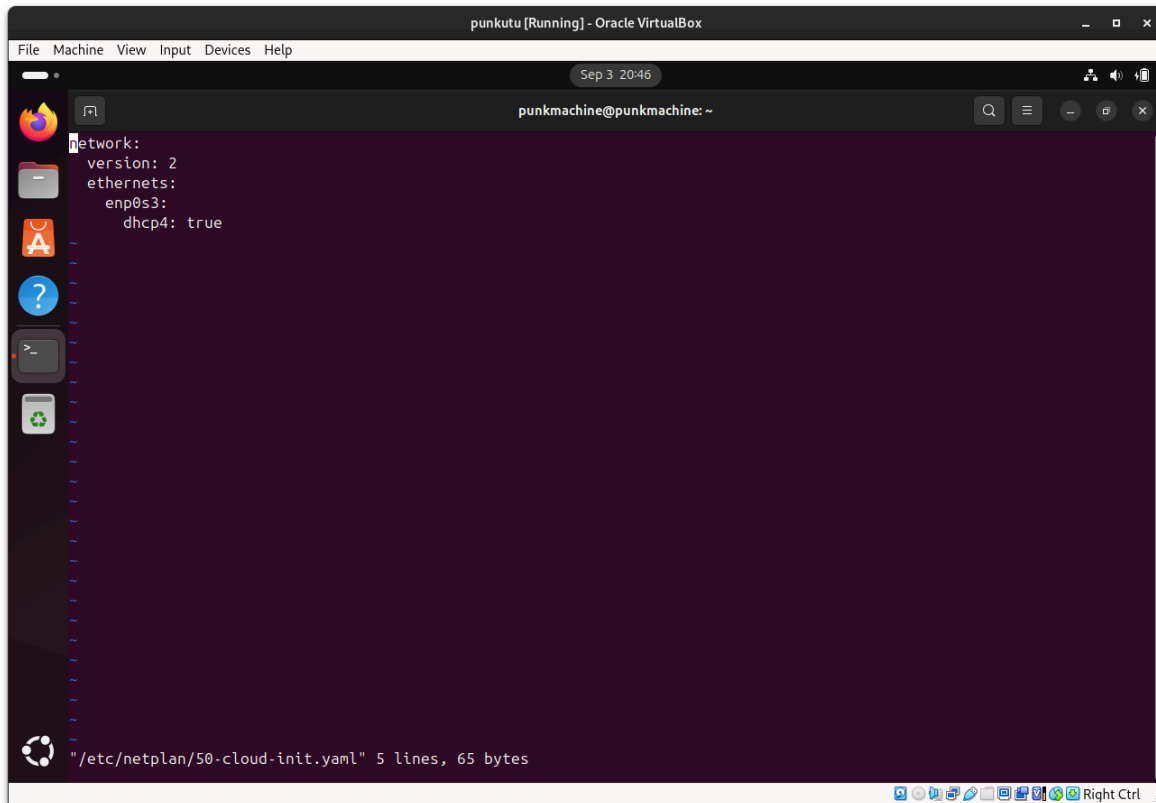
```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: true
```

3. **Apliquei as mudanças (se necessário):** Se eu fizesse alguma alteração, aplicaria as configurações com:

```
sudo netplan apply
```

Verificação:

Após a verificação, confirmei que a interface de rede já estava configurada para DHCP via `netplan`, especificamente no arquivo `50-cloud-init.yaml`.

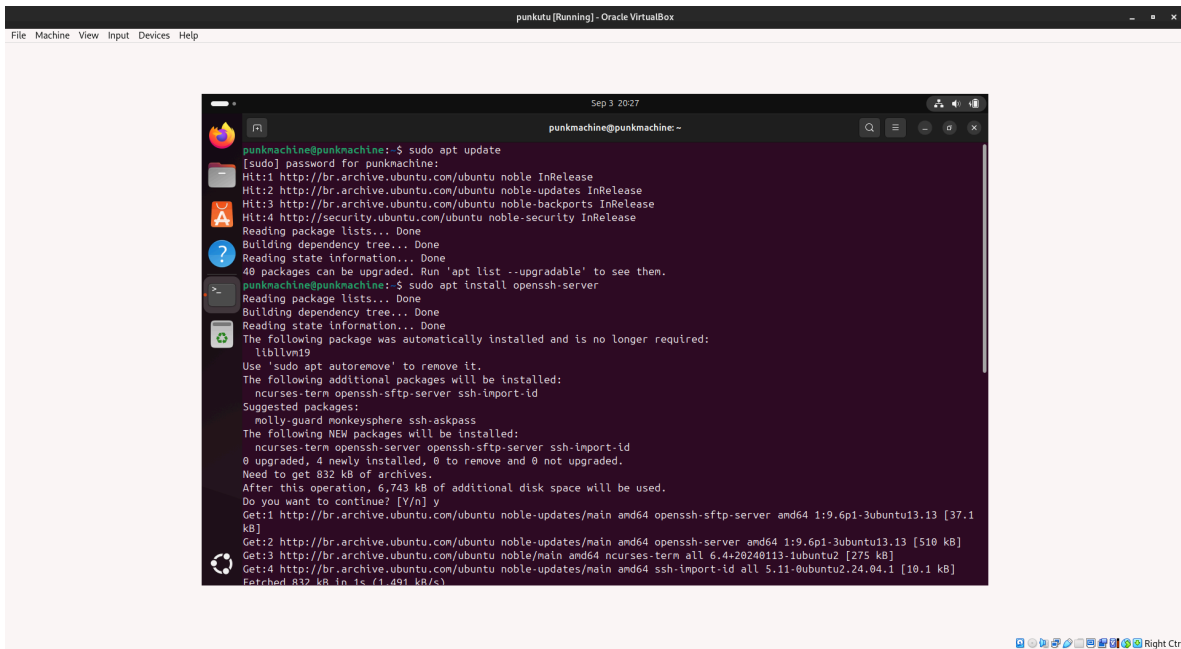


Captura de tela do arquivo de configuração Netplan (50-cloud-init.yaml) confirmando DHCP.

4. Configuração do Servidor SSH na VM2 (Ubuntu)

- No Ubuntu (VM2), instalei e configurei um servidor SSH.
- Abri o terminal.
- Instalei o pacote `openssh-server`:

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



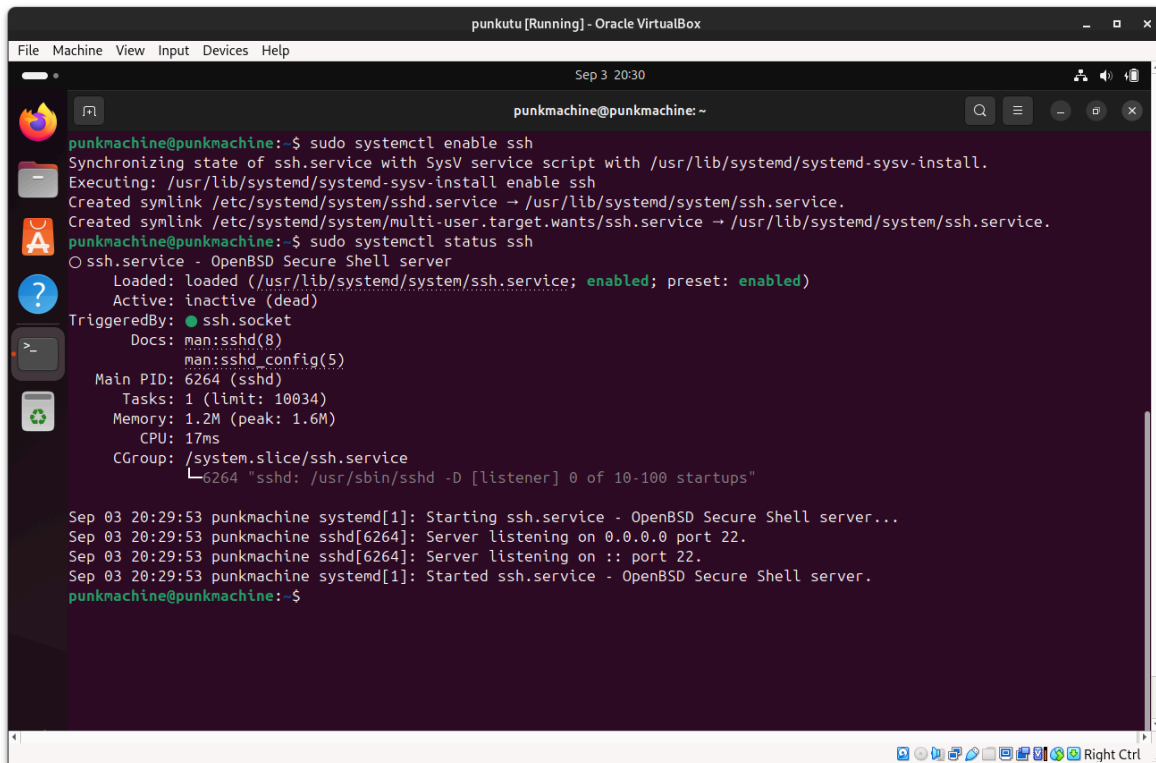
Captura de tela do terminal mostrando a instalação do servidor OpenSSH no Ubuntu.

- Iniciei o serviço SSH:

```
sudo systemctl start ssh
```

- Verifiquei o status do serviço:

```
sudo systemctl status ssh
```



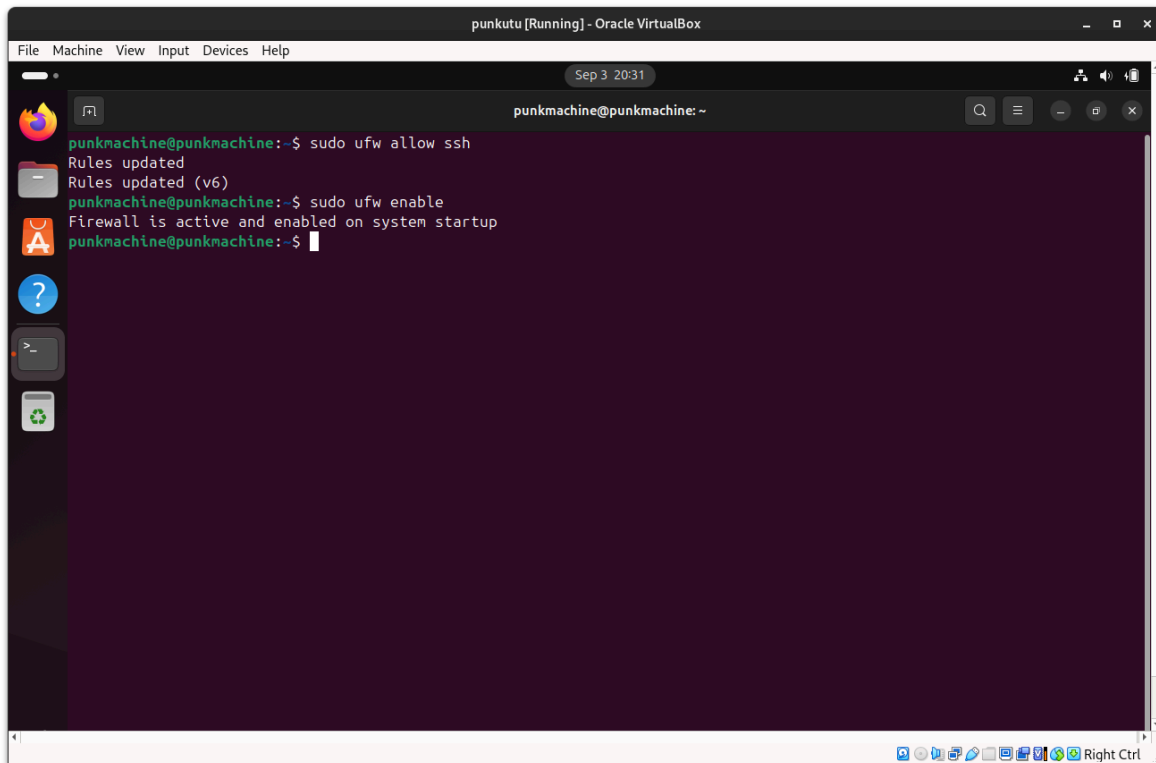
```
punkmachine@punkmachine:~$ sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
Created symlink /etc/systemd/system/ssh.service → /usr/lib/systemd/system/ssh.service.
punkmachine@punkmachine:~$ sudo systemctl status ssh
○ ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: inactive (dead)
 TriggeredBy: ● ssh.socket
    Docs: man:sshd(8)
          man:sshd_config(5)
   Main PID: 6264 (sshd)
     Tasks: 1 (limit: 10034)
    Memory: 1.2M (peak: 1.6M)
       CPU: 17ms
    CGroup: /system.slice/ssh.service
            └─6264 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Sep 03 20:29:53 punkmachine systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Sep 03 20:29:53 punkmachine sshd[6264]: Server listening on 0.0.0.0 port 22.
Sep 03 20:29:53 punkmachine sshd[6264]: Server listening on :: port 22.
Sep 03 20:29:53 punkmachine systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
punkmachine@punkmachine:~$
```

Captura de tela do terminal mostrando o status do serviço SSH no Ubuntu.

- Configurei o firewall (UFW) para permitir conexões SSH:

```
sudo ufw allow ssh
sudo ufw enable
```

A screenshot of a terminal window titled 'punkutu [Running] - Oracle VirtualBox'. The terminal shows the following commands and output:

```
punkmachine@punkmachine:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
punkmachine@punkmachine:~$ sudo ufw enable
Firewall is active and enabled on system startup
punkmachine@punkmachine:~$
```

The terminal has a dark purple background and a light-colored prompt. The window's title bar and menu bar are visible at the top.

Captura de tela do terminal mostrando a configuração do firewall UFW para permitir SSH no Ubuntu.

5. Configuração do Cliente SSH na VM1 (Windows 10)

VM1: Windows 10

- No Windows 10 (VM1), o cliente SSH já veio pré-instalado em versões mais recentes.

Para configurar e testar o cliente SSH no Windows 10, segui estes passos:

1. Verifiquei a Instalação do Cliente SSH:

- Abri o PowerShell ou o Prompt de Comando.
- Digitei `ssh` e pressionei Enter. Se o comando fosse reconhecido e mostrasse a ajuda do SSH, significava que o cliente estava instalado. Caso contrário, eu precisaria instalá-lo (geralmente via "Recursos Opcionais" nas Configurações do Windows).

```
Prompt de Comando
Microsoft Windows [versão 10.0.19045.2965]
(c) Microsoft Corporation. Todos os direitos reservados.

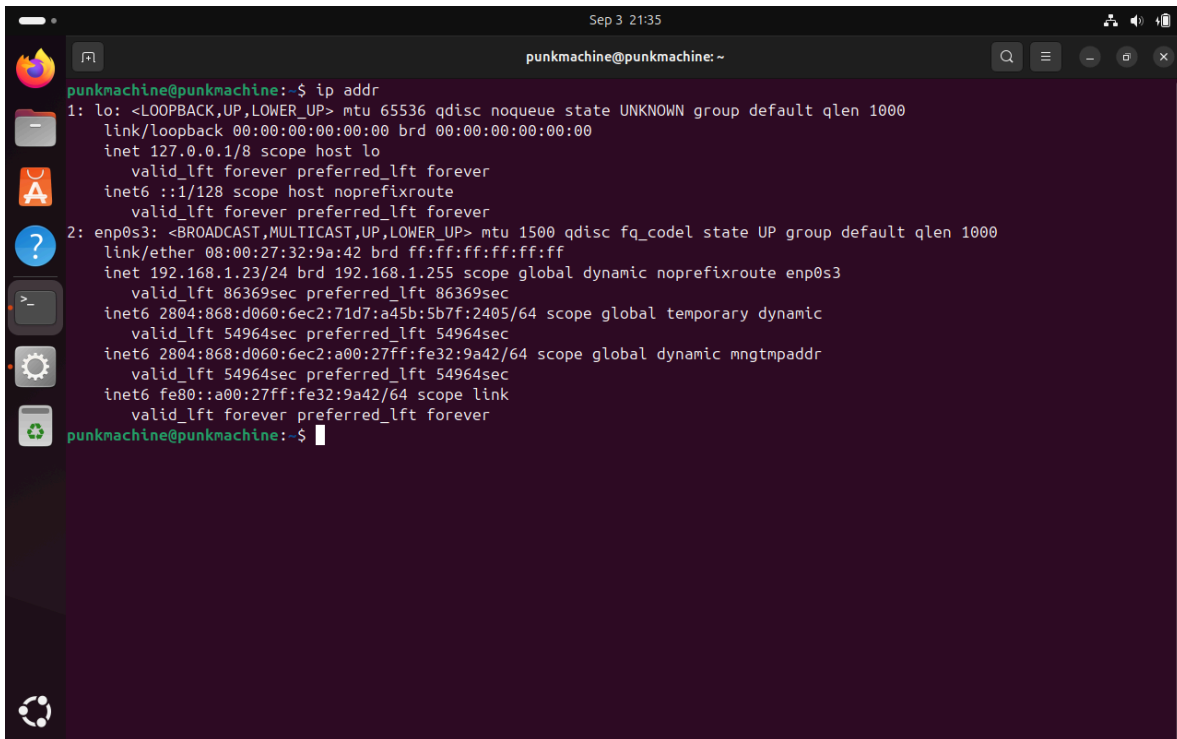
C:\Users\Punk da Silva>ssh
usage: ssh [-46AaCfGgKkMnQsTtVvXxYy] [-B bind_interface]
          [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
          [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
          [-i identity_file] [-J [user@]host[:port]] [-L address]
          [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
          [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
          [-w local_tun[:remote_tun]] destination [command]

C:\Users\Punk da Silva>
```

Captura de tela do PowerShell/CMD mostrando a ajuda do comando SSH, confirmando sua instalação.

2. Testei a Conectividade SSH com a VM2 (Ubuntu):

- No PowerShell ou Prompt de Comando, utilizei o comando `ssh` para me conectar à VM2.
- O comando que usei foi: `ssh usuario@IP_DA_VM2`
 - Substituí `usuario` pelo nome de usuário que configurei na VM2 (Ubuntu).
 - Substituí `IP_DA_VM2` pelo endereço IP da minha máquina virtual Ubuntu.



```
punkmachine@punkmachine:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:32:9a:42 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.23/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3
        valid_lft 86369sec preferred_lft 86369sec
    inet6 2804:868:d060:6ec2:71d7:a45b:5b7f:2405/64 scope global temporary dynamic
        valid_lft 54964sec preferred_lft 54964sec
    inet6 2804:868:d060:6ec2:a00:27ff:fe32:9a42/64 scope global dynamic mngtmpaddr
        valid_lft 54964sec preferred_lft 54964sec
    inet6 fe80::a00:27ff:fe32:9a42/64 scope link
        valid_lft forever preferred_lft forever
punkmachine@punkmachine:~$
```

Captura de tela do terminal Ubuntu mostrando o endereço IP da VM.

- Na primeira conexão, o sistema me perguntou se eu queria aceitar a chave de host da VM2. Digitei `yes` e pressionei Enter.
-
- Em seguida, me pediu a senha do usuário na VM2. Digitei a senha e pressionei Enter.

```
punkmachine@punkmachine: ~
Microsoft Windows [versão 10.0.19045.2965]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Punk da Silva>ssh
usage: ssh [-46AaCfGgKkMnqsTtVvXxYy] [-B bind_interface]
          [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
          [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
          [-i identity_file] [-J [user@]host[:port]] [-L address]
          [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
          [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
          [-w local_tun[:remote_tun]] destination [command]

C:\Users\Punk da Silva>ssh punkmachine@192.168.1.23
The authenticity of host '192.168.1.23 (192.168.1.23)' can't be established.
ECDSA key fingerprint is SHA256:yj7lyZs7aCmPDWRPUMatB/03GYDI/0wcH6vyoD2wf70.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.23' (ECDSA) to the list of known hosts.
punkmachine@192.168.1.23's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

11 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

punkmachine@punkmachine:~$
```

Captura de tela do PowerShell/CMD mostrando a conexão SSH bem-sucedida com a VM Ubuntu.

i Verificação:

Após inserir a senha corretamente, fui conectado ao terminal da VM2 (Ubuntu) via SSH, confirmando que a comunicação entre as VMs estava funcionando.

Atividade 1.2 - Conectividade e SSH

Objetivo

- Comprovar a conectividade no nível IP com outro dispositivo TCP/IP e testar o protocolo SSH.

A. TESTES NECESSÁRIOS PARA O LABORATÓRIO

1. Verificar o status da conexão entre VM1 e VM2 usando o comando ping.

Para verificar a conectividade entre a VM1 (Windows 10) e a VM2 (Ubuntu), utilizei o comando `ping`.

Passo a passo:

1. Na VM1 (Windows 10), abri o Prompt de Comando ou PowerShell.
 2. Na VM2 (Ubuntu), abri o Terminal.
 3. Em cada VM, obtive o endereço IP da outra VM.
- No Windows 10 (VM1), utilizei `ipconfig`.

```
Selecionar Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6

PS C:\Users\Punk da Silva> ipconfig

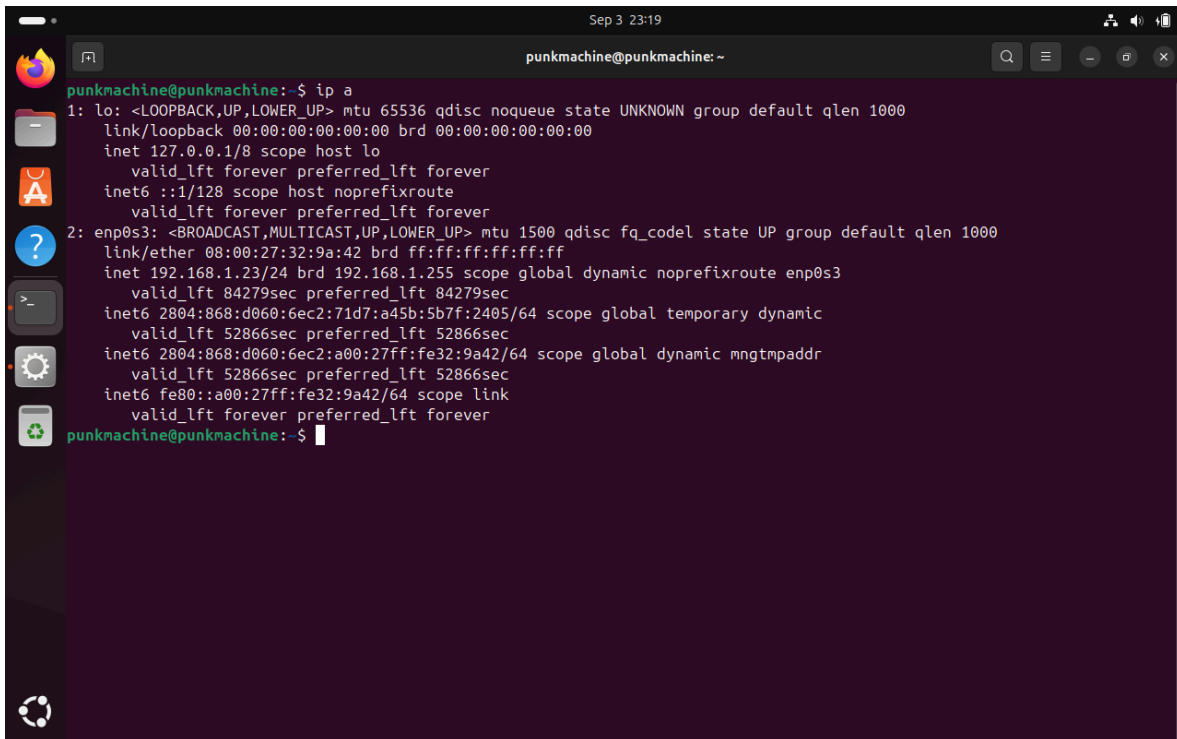
Configuração de IP do Windows

Adaptador Ethernet Ethernet:

    Sufixo DNS específico de conexão. . . . . : 
    Endereço IPv6 . . . . . : 2804:868:d060:6ec2:80cc:949f:eadf:4514
    Endereço IPv6 Temporário. . . . . : 2804:868:d060:6ec2:f51a:e112:cfec:26ac
    Endereço IPv6 de link local . . . . . : fe80::d15d:faff:b6a9:a8d4%5
    Endereço IPv4. . . . . : 192.168.1.24
    Máscara de Sub-rede . . . . . : 255.255.255.0
    Gateway Padrão. . . . . : fe80::1%5
                                192.168.1.1
PS C:\Users\Punk da Silva>
```

Resultado do comando ipconfig no Prompt de Comando do Windows, mostrando o endereço IPv4, máscara de sub-rede e gateway padrão.

- No Ubuntu (VM2), utilizei `ip a` ou `ifconfig`.

A terminal window titled 'punkmachine@punkmachine: ~' showing the output of the 'ip a' command. The output lists details for two network interfaces: 'lo' (loopback) and 'enp0s3' (Ethernet). The 'lo' interface has IP 127.0.0.1. The 'enp0s3' interface has IP 192.168.1.23 and several IPv6 addresses. The terminal window has a dark theme and a sidebar with application icons on the left.

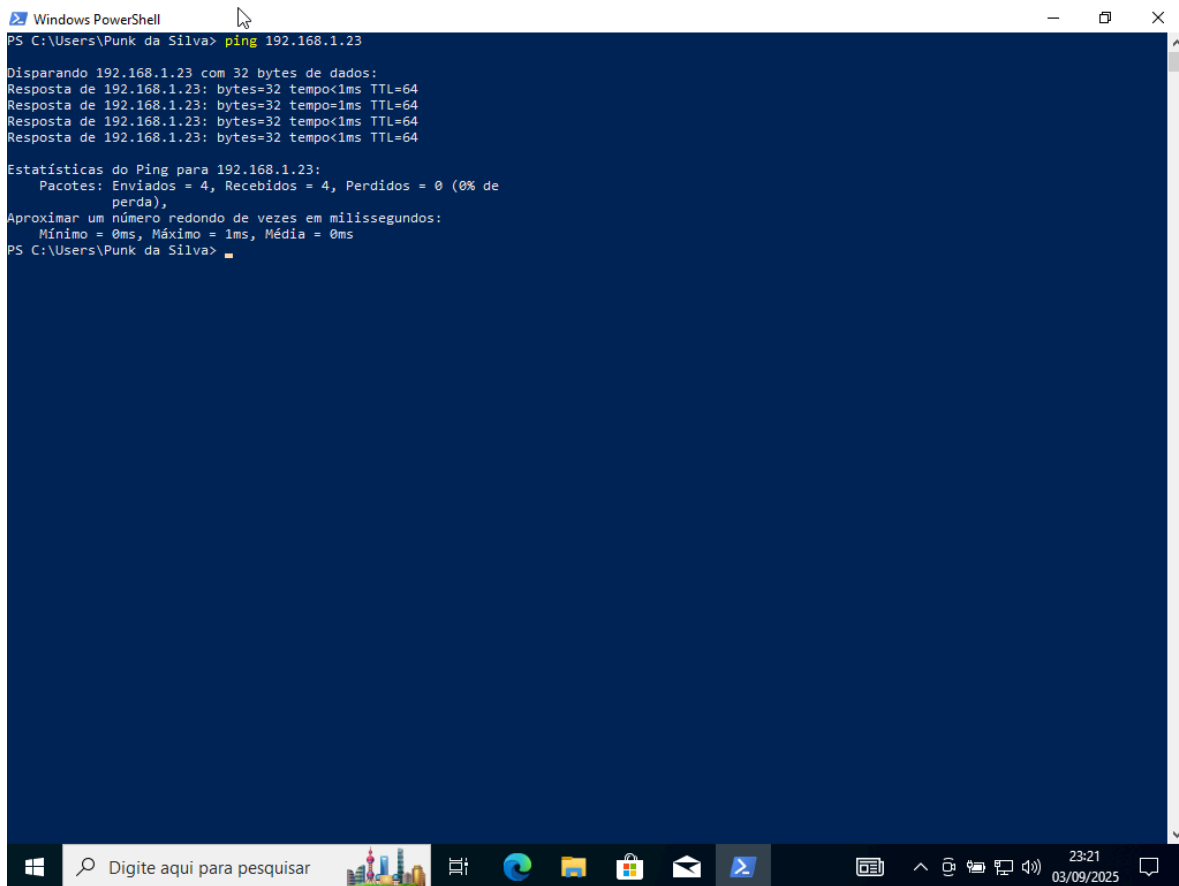
```
punkmachine@punkmachine:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:32:9a:42 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.23/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3
        valid_lft 84279sec preferred_lft 84279sec
    inet6 2804:868:d060:6ec2:71d7:a45b:5b7f:2405/64 scope global temporary dynamic
        valid_lft 52866sec preferred_lft 52866sec
    inet6 2804:868:d060:6ec2:a00:27ff:fe32:9a42/64 scope global dynamic mngtmpaddr
        valid_lft 52866sec preferred_lft 52866sec
    inet6 fe80::a00:27ff:fe32:9a42/64 scope link
        valid_lft forever preferred_lft forever
punkmachine@punkmachine:~$
```

Resultado do comando 'ip a' no terminal do Ubuntu, exibindo os detalhes das interfaces de rede e seus endereços IP.

4. Executei o comando `ping` de uma VM para a outra.

- Da VM1 (Windows 10) para a VM2 (Ubuntu):

```
ping <IP_DA_VM2>
```



```
Windows PowerShell
PS C:\Users\Punk da Silva> ping 192.168.1.23

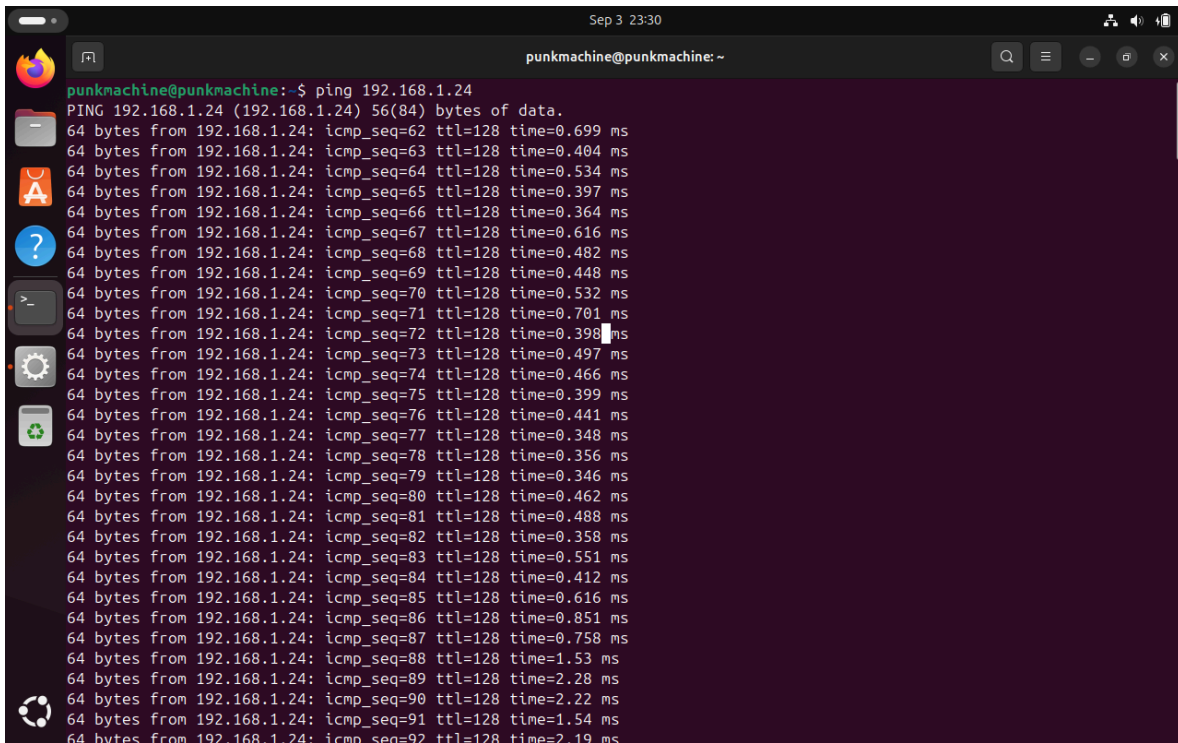
Disparando 192.168.1.23 com 32 bytes de dados:
Resposta de 192.168.1.23: bytes=32 tempo<1ms TTL=64
Resposta de 192.168.1.23: bytes=32 tempo<1ms TTL=64
Resposta de 192.168.1.23: bytes=32 tempo<1ms TTL=64
Resposta de 192.168.1.23: bytes=32 tempo<1ms TTL=64

Estatísticas do Ping para 192.168.1.23:
    Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de
    perda),
Aproximar um número redondo de vezes em milissegundos:
    Mínimo = 0ms, Máximo = 1ms, Média = 0ms
PS C:\Users\Punk da Silva>
```

Resultado do comando ping do Windows para um IP de uma máquina Linux, mostrando o recebimento de respostas.

- Da VM2 (Ubuntu) para a VM1 (Windows 10):

```
ping <IP_DA_VM1>
```



```
punkmachine@punkmachine:~$ ping 192.168.1.24
PING 192.168.1.24 (192.168.1.24) 56(84) bytes of data:
64 bytes from 192.168.1.24: icmp_seq=62 ttl=128 time=0.699 ms
64 bytes from 192.168.1.24: icmp_seq=63 ttl=128 time=0.404 ms
64 bytes from 192.168.1.24: icmp_seq=64 ttl=128 time=0.534 ms
64 bytes from 192.168.1.24: icmp_seq=65 ttl=128 time=0.397 ms
64 bytes from 192.168.1.24: icmp_seq=66 ttl=128 time=0.364 ms
64 bytes from 192.168.1.24: icmp_seq=67 ttl=128 time=0.616 ms
64 bytes from 192.168.1.24: icmp_seq=68 ttl=128 time=0.482 ms
64 bytes from 192.168.1.24: icmp_seq=69 ttl=128 time=0.448 ms
64 bytes from 192.168.1.24: icmp_seq=70 ttl=128 time=0.532 ms
64 bytes from 192.168.1.24: icmp_seq=71 ttl=128 time=0.701 ms
64 bytes from 192.168.1.24: icmp_seq=72 ttl=128 time=0.398 ms
64 bytes from 192.168.1.24: icmp_seq=73 ttl=128 time=0.497 ms
64 bytes from 192.168.1.24: icmp_seq=74 ttl=128 time=0.466 ms
64 bytes from 192.168.1.24: icmp_seq=75 ttl=128 time=0.399 ms
64 bytes from 192.168.1.24: icmp_seq=76 ttl=128 time=0.441 ms
64 bytes from 192.168.1.24: icmp_seq=77 ttl=128 time=0.348 ms
64 bytes from 192.168.1.24: icmp_seq=78 ttl=128 time=0.356 ms
64 bytes from 192.168.1.24: icmp_seq=79 ttl=128 time=0.346 ms
64 bytes from 192.168.1.24: icmp_seq=80 ttl=128 time=0.462 ms
64 bytes from 192.168.1.24: icmp_seq=81 ttl=128 time=0.488 ms
64 bytes from 192.168.1.24: icmp_seq=82 ttl=128 time=0.358 ms
64 bytes from 192.168.1.24: icmp_seq=83 ttl=128 time=0.551 ms
64 bytes from 192.168.1.24: icmp_seq=84 ttl=128 time=0.412 ms
64 bytes from 192.168.1.24: icmp_seq=85 ttl=128 time=0.616 ms
64 bytes from 192.168.1.24: icmp_seq=86 ttl=128 time=0.851 ms
64 bytes from 192.168.1.24: icmp_seq=87 ttl=128 time=0.758 ms
64 bytes from 192.168.1.24: icmp_seq=88 ttl=128 time=1.53 ms
64 bytes from 192.168.1.24: icmp_seq=89 ttl=128 time=2.28 ms
64 bytes from 192.168.1.24: icmp_seq=90 ttl=128 time=2.22 ms
64 bytes from 192.168.1.24: icmp_seq=91 ttl=128 time=1.54 ms
64 bytes from 192.168.1.24: icmp_seq=92 ttl=128 time=2.19 ms
```

Resultado do comando ping do Ubuntu para um IP de uma máquina Windows, mostrando o recebimento de respostas.

5. Observei os resultados para confirmar a troca de pacotes e a ausência de perdas.

2. Fazer ping nos seguintes sites: Gmail.com, www.facebook.com das 2 VMs.

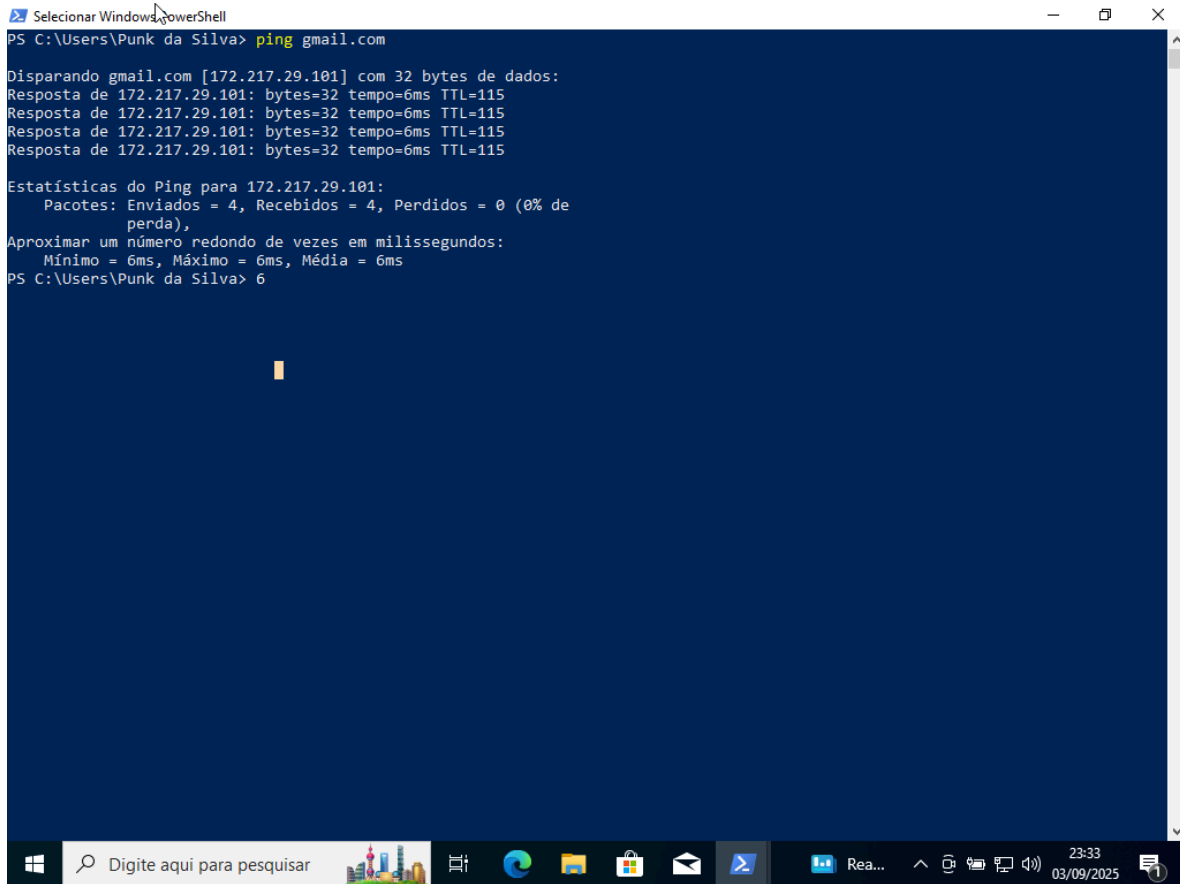
Para testar a conectividade externa de ambas as VMs, realizei pings para sites conhecidos.

Passo a passo:

1. Na VM1 (Windows 10), abri o Prompt de Comando ou PowerShell.
 2. Na VM2 (Ubuntu), abri o Terminal.
 3. Em cada VM, executei os comandos `ping` para os sites especificados.
- Na VM1 (Windows 10):

```
ping gmail.com
```

ping www.facebook.com

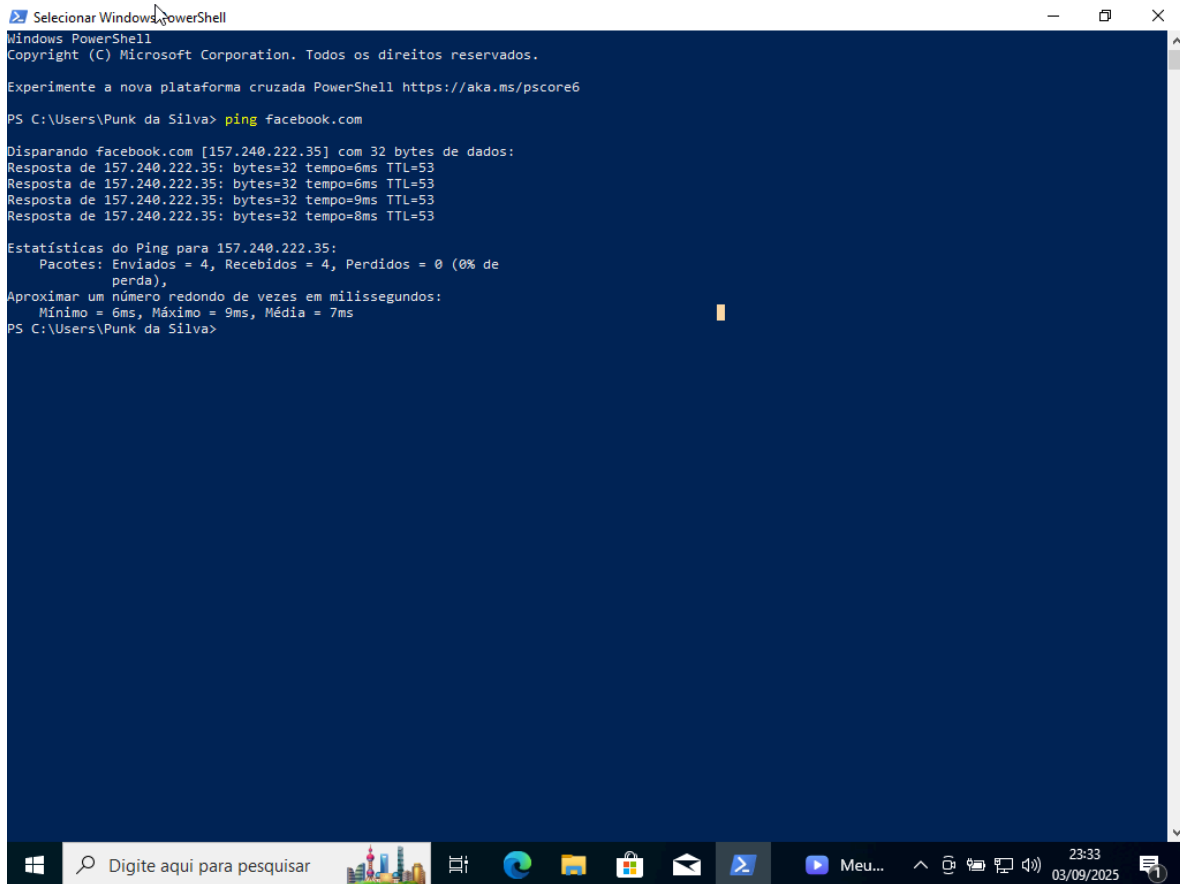


```
Selecionar Windows PowerShell
PS C:\Users\Punk da Silva> ping gmail.com

Disparando gmail.com [172.217.29.101] com 32 bytes de dados:
Resposta de 172.217.29.101: bytes=32 tempo=6ms TTL=115
Resposta de 172.217.29.101: bytes=32 tempo=6ms TTL=115
Resposta de 172.217.29.101: bytes=32 tempo=6ms TTL=115
Resposta de 172.217.29.101: bytes=32 tempo=6ms TTL=115

Estatísticas do Ping para 172.217.29.101:
    Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de perda),
Aproximar um número redondo de vezes em milissegundos:
    Mínimo = 6ms, Máximo = 6ms, Média = 6ms
PS C:\Users\Punk da Silva> 6
```

Resultado do comando 'ping gmail.com' no Windows, mostrando conectividade com o site.



The screenshot shows a Windows PowerShell window titled "Selecionar Windows PowerShell". The window has a dark blue background. The text inside the window is as follows:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6

PS C:\Users\Punk da Silva> ping facebook.com

Disparando facebook.com [157.240.222.35] com 32 bytes de dados:
Resposta de 157.240.222.35: bytes=32 tempo=6ms TTL=53
Resposta de 157.240.222.35: bytes=32 tempo=6ms TTL=53
Resposta de 157.240.222.35: bytes=32 tempo=9ms TTL=53
Resposta de 157.240.222.35: bytes=32 tempo=8ms TTL=53

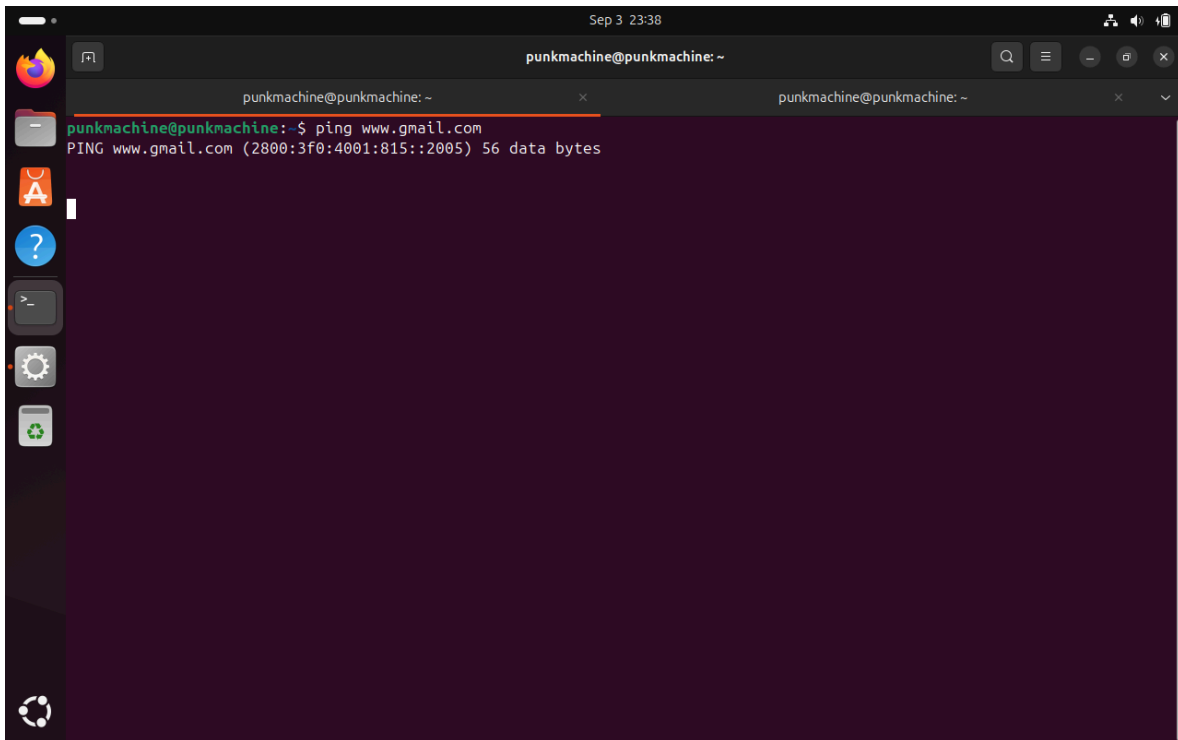
Estatísticas do Ping para 157.240.222.35:
    Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de
    perda),
Aproximar um número redondo de vezes em milissegundos:
    Mínimo = 6ms, Máximo = 9ms, Média = 7ms
PS C:\Users\Punk da Silva>
```

The Windows taskbar is visible at the bottom, showing the Start button, a search bar with the text "Digite aqui para pesquisar", and several application icons. The system tray on the right shows the time as 23:33 and the date as 03/09/2025.

Resultado do comando 'ping www.facebook.com' no Windows, mostrando conectividade com o site.

- Na VM2 (Ubuntu):

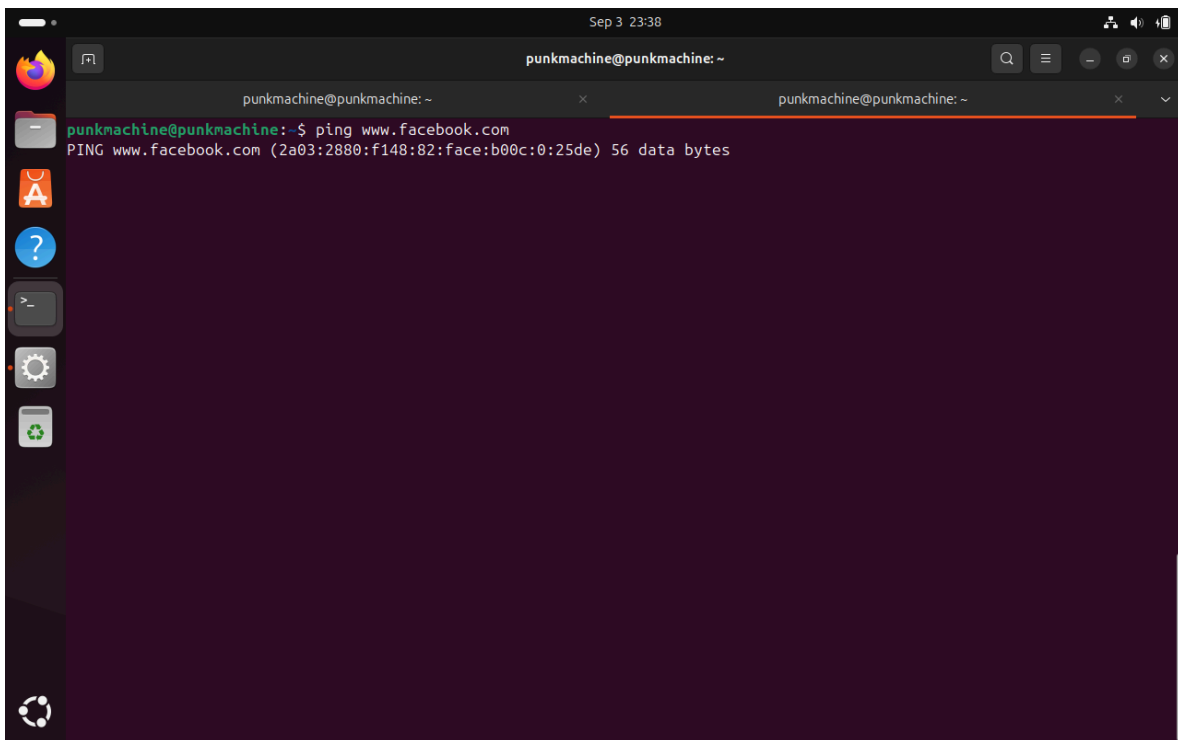
```
ping gmail.com
ping www.facebook.com
```



A terminal window on an Ubuntu system. The window title is 'punkmachine@punkmachine: ~'. The prompt is 'punkmachine@punkmachine:~\$'. The command 'ping www.gmail.com' has been entered. The output shows 'PING www.gmail.com (2800:3f0:4001:815::2005) 56 data bytes'. The terminal has a dark purple background. On the left, there is a vertical dock with icons for Firefox, the Dash, and several application launchers. The top of the window shows system status icons and the date 'Sep 3 23:38'.

```
punkmachine@punkmachine:~$ ping www.gmail.com
PING www.gmail.com (2800:3f0:4001:815::2005) 56 data bytes
```

Resultado do comando 'ping gmail.com' no Ubuntu, mostrando conectividade com o site.



A terminal window on an Ubuntu system, similar to the one above. The window title is 'punkmachine@punkmachine: ~'. The prompt is 'punkmachine@punkmachine:~\$'. The command 'ping www.facebook.com' has been entered. The output shows 'PING www.facebook.com (2a03:2880:f148:82:face:b00c:0:25de) 56 data bytes'. The terminal has a dark purple background. On the left, there is a vertical dock with icons for Firefox, the Dash, and several application launchers. The top of the window shows system status icons and the date 'Sep 3 23:38'.

```
punkmachine@punkmachine:~$ ping www.facebook.com
PING www.facebook.com (2a03:2880:f148:82:face:b00c:0:25de) 56 data bytes
```

Resultado do comando 'ping www.facebook.com' no Ubuntu, mostrando conectividade com o site.

4. Observei os resultados para verificar a resolução de nomes (DNS) e a conectividade com a internet.

3. Explicar as informações retornadas após o ping.

Após executar o comando `ping`, observei as seguintes informações:

- **Reply from <IP_ADDRESS> ou 64 bytes from <IP_ADDRESS>**: Indica que uma resposta foi recebida do endereço IP de destino.
- **bytes=<NUMBER>**: O tamanho do pacote de dados enviado e recebido (geralmente 32 ou 64 bytes).
- **time=<TIME>ms**: O tempo, em milissegundos, que levou para o pacote ir e voltar (latência). Tempos menores indicam uma conexão mais rápida.
- **TTL=<NUMBER> (Time To Live)**: O número de "saltos" (hops) que o pacote pode dar antes de ser descartado. Diminui a cada roteador que o pacote passa.
- **Packets: Sent = <NUMBER>, Received = <NUMBER>, Lost = <NUMBER>**: Um resumo da sessão de ping, mostrando quantos pacotes foram enviados, recebidos e perdidos. Perda de pacotes indica problemas na rede.
- **Approximate round trip times in milli-seconds::** Um resumo dos tempos mínimo, máximo e médio de ida e volta.

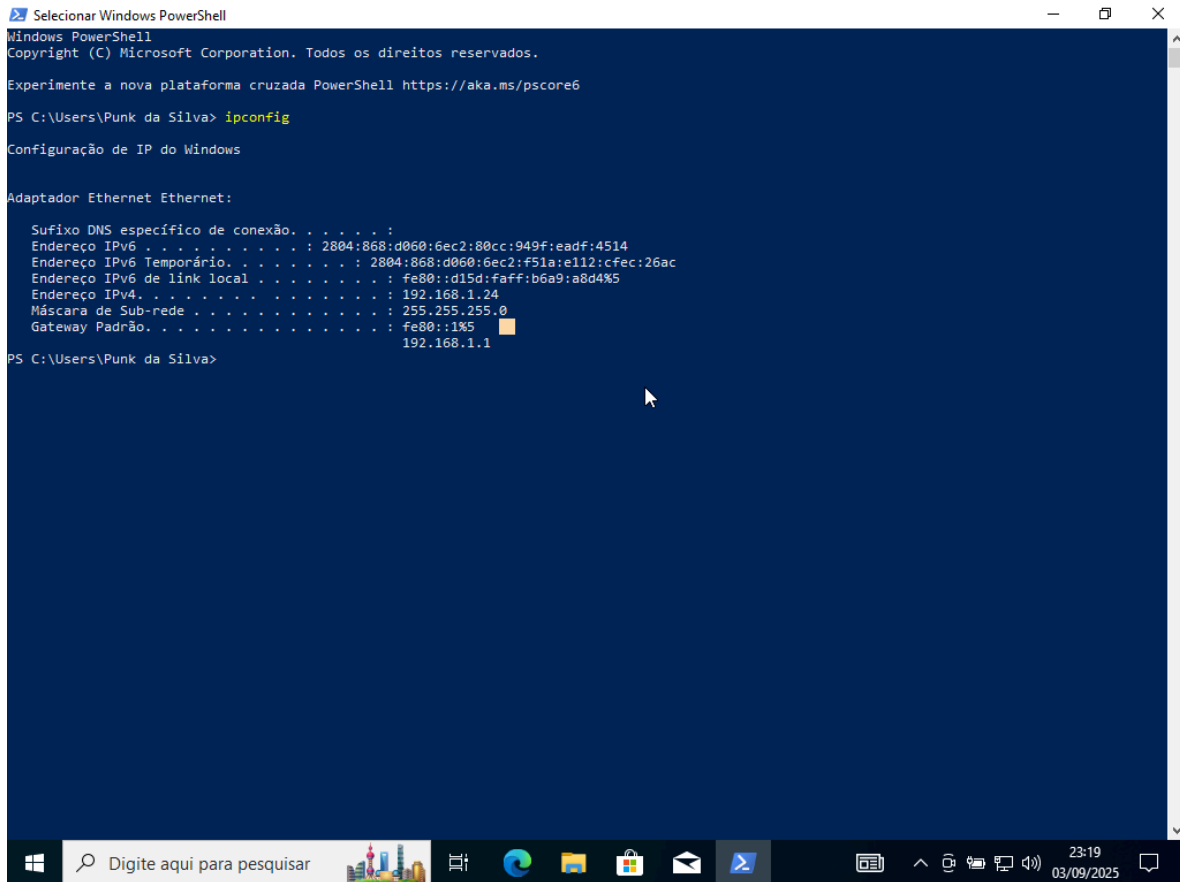
4. Executar o comando IPCONFIG e descrever as informações que ele retorna.

Executei o comando `ipconfig` no Windows 10 (VM1) para obter informações sobre a configuração de rede.

Passo a passo:

1. Na VM1 (Windows 10), abri o Prompt de Comando ou PowerShell.
2. Executei o comando:

```
ipconfig
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6

PS C:\Users\Punk da Silva> ipconfig

Configuração de IP do Windows

Adaptador Ethernet Ethernet:

    Sufixo DNS específico de conexão. . . . . : 
    Endereço IPv6 . . . . . : 2804:868:d060:6ec2:80cc:949f:eadf:4514
    Endereço IPv6 Temporário. . . . . : 2804:868:d060:6ec2:f51a:e112:cfec:26ac
    Endereço IPv6 de link local . . . . . : fe80::d15d:faff:b6a9:a8d4%5
    Endereço IPv4. . . . . : 192.168.1.24
    Máscara de Sub-rede . . . . . : 255.255.255.0
    Gateway Padrão. . . . . : fe80::1%5
                                192.168.1.1

PS C:\Users\Punk da Silva>
```

Resultado do comando ipconfig no Prompt de Comando do Windows, mostrando o endereço IPv4, máscara de sub-rede e gateway padrão.

3. Observei as seguintes informações retornadas:

- **Endereço IPv4:** O endereço IP atribuído à interface de rede.
- **Máscara de Sub-rede:** A máscara de sub-rede utilizada.
- **Gateway Padrão:** O endereço IP do roteador (gateway) que a VM utiliza para acessar outras redes.
- **Servidores DNS:** Os endereços IP dos servidores DNS configurados.

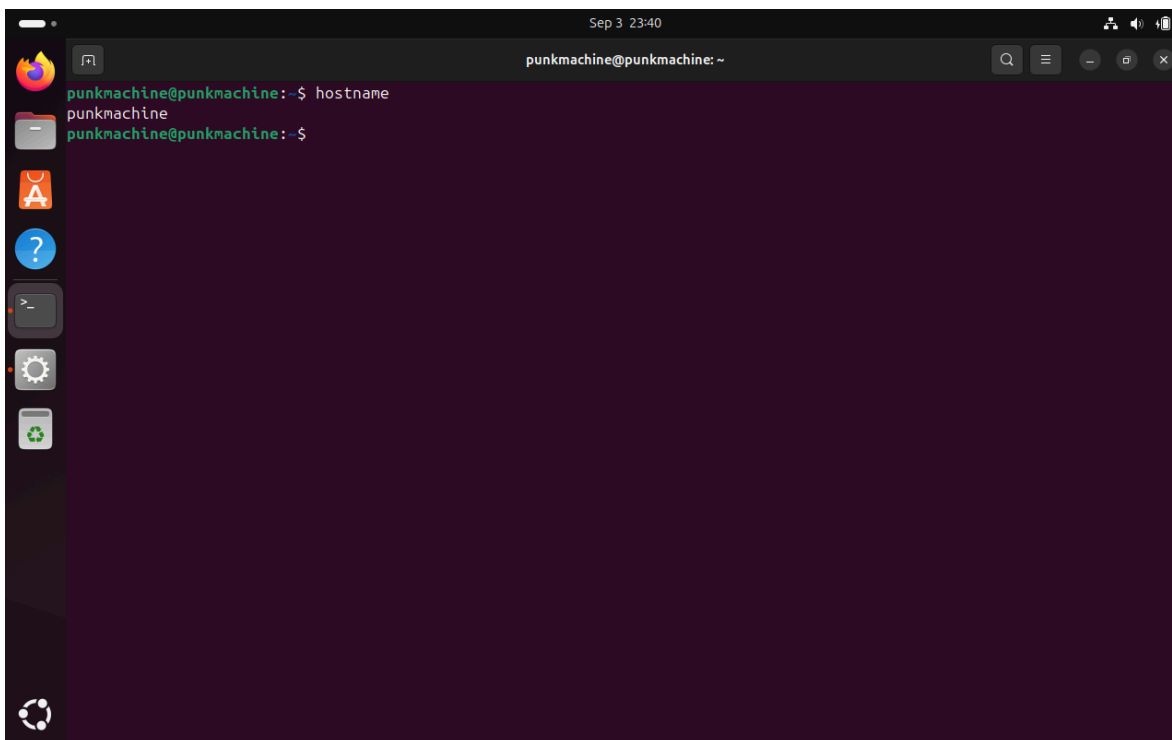
5. Executar o comando HOSTNAME para descrever as informações que ele retorna.

Executei o comando `hostname` no Ubuntu (VM2) para obter o nome do host da máquina.

Passo a passo:

1. Na VM2 (Ubuntu), abri o Terminal.
2. Executei o comando:

```
hostname
```



Resultado do comando 'hostname' no terminal do Ubuntu, exibindo o nome do host da máquina.

3. Observei que o comando retornou o nome configurado para a máquina, que é o identificador da VM na rede.

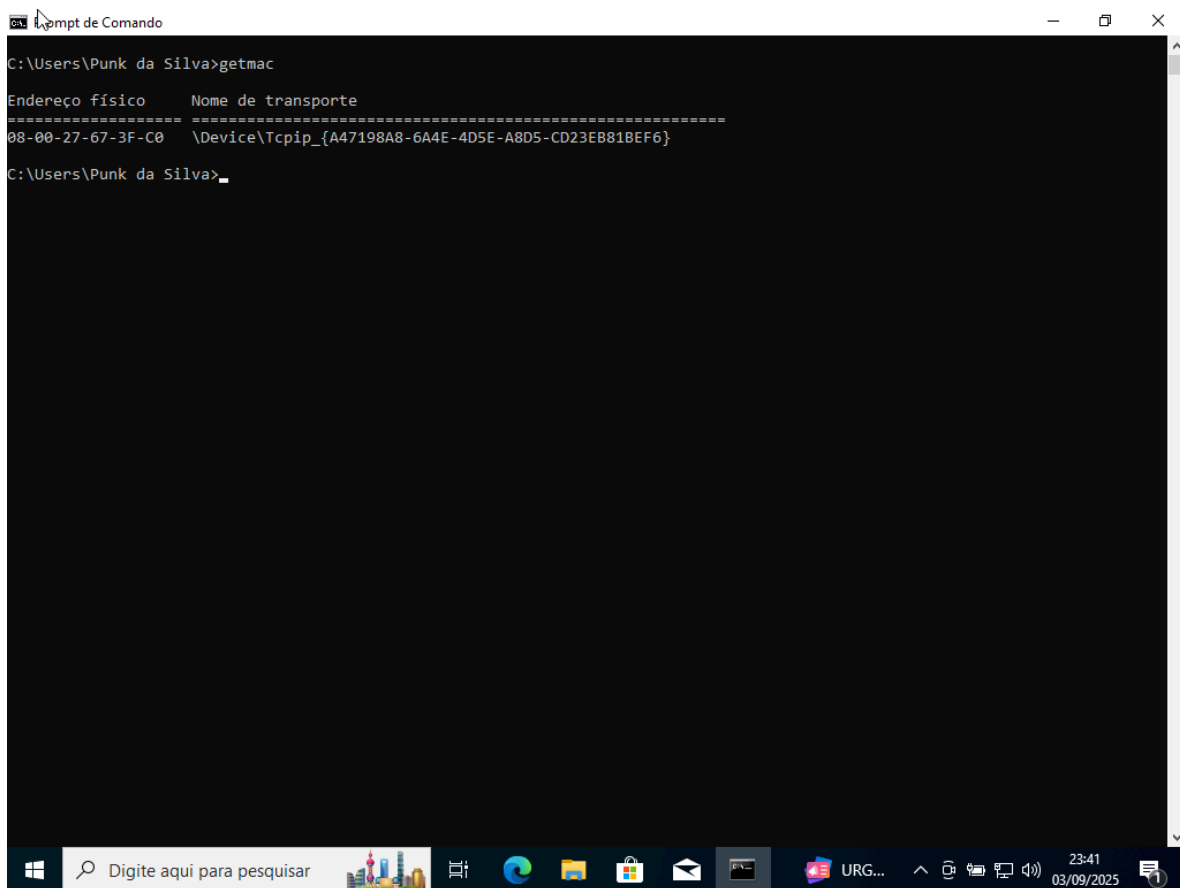
6. Executar o comando GETMAC e descrever as informações que ele retorna.

Executei o comando `getmac` no Windows 10 (VM1) para obter o endereço MAC (Media Access Control) das interfaces de rede.

Passo a passo:

1. Na VM1 (Windows 10), abri o Prompt de Comando ou PowerShell.
2. Executei o comando:

getmac



```
cmd [Prompt de Comando]
C:\Users\Punk da Silva>getmac

Endereço físico      Nome de transporte
-----
08-00-27-67-3F-C0    \Device\NPF_{A47198A8-6A4E-4D5E-A8D5-CD23EB81BEF6}
```

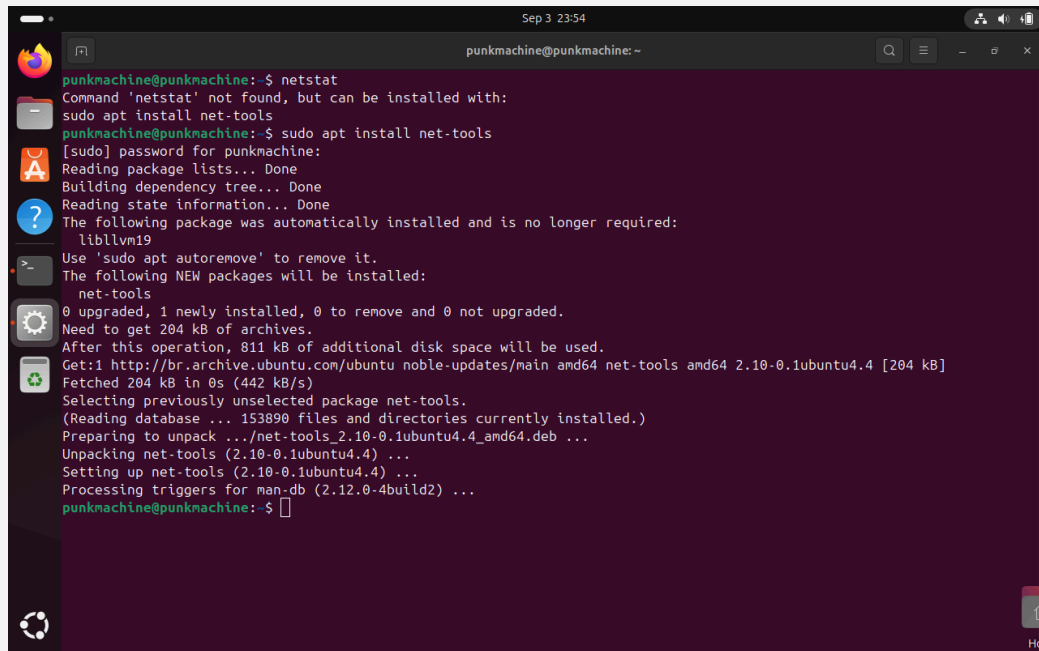
Resultado do comando 'getmac' no Windows, listando os endereços MAC das interfaces de rede.

3. Observei que o comando retornou o endereço físico (MAC) de cada adaptador de rede presente na VM, que é um identificador único de hardware.

7. Executar o comando NETSTAT e descrever as informações que ele retorna.

⚠ Nota: Em algumas versões mais recentes do Ubuntu, o comando `netstat` pode não estar instalado por padrão. Se você receber um erro de "comando não encontrado", pode instalá-lo com o seguinte comando:

```
sudo apt install net-tools
```

A terminal window titled 'punkmachine@punkmachine: ~' with a search bar and window controls. The terminal shows the following commands and output:

```
punkmachine@punkmachine:~$ netstat
Command 'netstat' not found, but can be installed with:
sudo apt install net-tools
punkmachine@punkmachine:~$ sudo apt install net-tools
[sudo] password for punkmachine:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 204 kB of archives.
After this operation, 811 kB of additional disk space will be used.
Get:1 http://br.archive.ubuntu.com/ubuntu noble-updates/main amd64 net-tools amd64 2.10-0.1ubuntu4.4 [204 kB]
Fetched 204 kB in 0s (442 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 153890 files and directories currently installed.)
Preparing to unpack .../net-tools_2.10-0.1ubuntu4.4_amd64.deb ...
Unpacking net-tools (2.10-0.1ubuntu4.4) ...
Setting up net-tools (2.10-0.1ubuntu4.4) ...
Processing triggers for man-db (2.12.0-4build2) ...
punkmachine@punkmachine:~$
```

Execução do comando 'sudo apt install net-tools' no terminal do Ubuntu para instalar o netstat.

Após instalar o `netstat` (devido ao erro de 'comando não encontrado'), executei-o no Ubuntu (VM2) para exibir as conexões de rede ativas, tabelas de roteamento e estatísticas de interface.

Passo a passo:

1. Na VM2 (Ubuntu), abri o Terminal.
2. Executei o comando:

```
netstat
```

```

punkmachine@punkmachine:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 punkmachine:bootpc      _gateway:bootps        ESTABLISHED

Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags     Type       State      I-Node  Path
unix    3      [ ]      STREAM     CONNECTED  33123    /run/user/1000/pulse/native
unix    3      [ ]      STREAM     CONNECTED  19597    /run/user/1000/bus
unix    3      [ ]      STREAM     CONNECTED  16751    /run/user/1000/bus
unix    3      [ ]      STREAM     CONNECTED  16692    /run/user/1000/bus
unix    3      [ ]      STREAM     CONNECTED  39975    /run/user/1000/bus
unix    3      [ ]      STREAM     CONNECTED  14178    /run/dbus/system_bus_socket
unix    2      [ ]      DGRAM      CONNECTED  17604    /run/dbus/system_bus_socket
unix    2      [ ]      DGRAM      CONNECTED  13265    /run/dbus/system_bus_socket
unix    3      [ ]      STREAM     CONNECTED  12436    /run/user/1000/bus
unix    3      [ ]      STREAM     CONNECTED  10038    /run/user/1000/bus
unix    3      [ ]      STREAM     CONNECTED  40030    /run/user/1000/bus
unix    3      [ ]      STREAM     CONNECTED  16806    /run/systemd/journal/stdout
unix    3      [ ]      STREAM     CONNECTED  14656    /run/systemd/journal/stdout
unix    3      [ ]      STREAM     CONNECTED  14891    /run/systemd/journal/stdout
unix    3      [ ]      STREAM     CONNECTED  19540    /run/user/1000/bus
unix    3      [ ]      STREAM     CONNECTED  13129    /run/user/1000/bus
unix    2      [ ]      DGRAM      CONNECTED  5416    /run/user/1000/at-spi/bus
unix    3      [ ]      STREAM     CONNECTED  41471    /run/user/1000/at-spi/bus
unix    3      [ ]      STREAM     CONNECTED  19638    /run/user/1000/at-spi/bus
unix    2      [ ]      DGRAM      CONNECTED  16494    /run/user/1000/at-spi/bus
unix    3      [ ]      STREAM     CONNECTED  11428    /run/user/1000/at-spi/bus
unix    3      [ ]      STREAM     CONNECTED  19612    /run/snapd-snap.socket
unix    3      [ ]      STREAM     CONNECTED  16758    /run/snapd-snap.socket
unix    3      [ ]      STREAM     CONNECTED  14721    /run/snapd-snap.socket
unix    3      [ ]      STREAM     CONNECTED  13154    /run/systemd/journal/stdout
unix    3      [ ]      STREAM     CONNECTED  14864    /run/systemd/journal/stdout
unix    3      [ ]      STREAM     CONNECTED  16690    /run/systemd/journal/stdout

```

Resultado do comando 'netstat' no Ubuntu, mostrando as conexões de rede ativas.

3. Observei que o comando retornou uma lista de conexões ativas (TCP, UDP), mostrando o endereço local, o endereço estrangeiro e o estado da conexão.

8. Executar o comando NETSTAT -an e descrever as informações que ele retorna.

Executei o comando `netstat -an` no Ubuntu (VM2) para obter uma visão mais detalhada das conexões de rede, incluindo portas numéricas.

Passo a passo:

1. Na VM2 (Ubuntu), abri o Terminal.
2. Executei o comando:

```
netstat -an
```



```
Sep 3 23:56
punkmachine@punkmachine: ~
punkmachine@punkmachine:~$ netstat -an
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.54:53           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp6       0      0 :::1:631                :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
udp        0      0 0.0.0.0:5353            0.0.0.0:*               LISTEN
udp        0      0 127.0.0.54:53           0.0.0.0:*               LISTEN
udp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
udp        0      0 192.168.1.23:68         192.168.1.1:67         ESTABLISHED
udp        0      0 0.0.0.0:49878           0.0.0.0:*               LISTEN
udp6       0      0 :::59051                :::*                    LISTEN
udp6       0      0 :::5353                 :::*                    LISTEN
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node      Path
unix   3      [ ]         STREAM     CONNECTED    33123       /run/user/1000/pulse/native
unix   3      [ ]         STREAM     CONNECTED    19597
unix   3      [ ]         STREAM     CONNECTED    16751
unix   3      [ ]         STREAM     CONNECTED    16692
unix   3      [ ]         STREAM     CONNECTED    39975
unix   3      [ ]         STREAM     CONNECTED    14178       /run/dbus/system_bus_socket
unix   2      [ ]         DGRAM      17604
unix   2      [ ]         DGRAM      13265
unix   3      [ ]         STREAM     CONNECTED    12436
unix   2      [ ACC ]       STREAM     LISTENING    8205       /run/systemd/io.systemd.syst
unix   3      [ ]         STREAM     CONNECTED    10038
unix   3      [ ]         STREAM     CONNECTED    40030       /run/user/1000/bus
unix   3      [ ]         STREAM     CONNECTED    16806       /run/systemd/journal/stdout
unix   3      [ ]         STREAM     CONNECTED    14656
unix   3      [ ]         STREAM     CONNECTED    14891
```

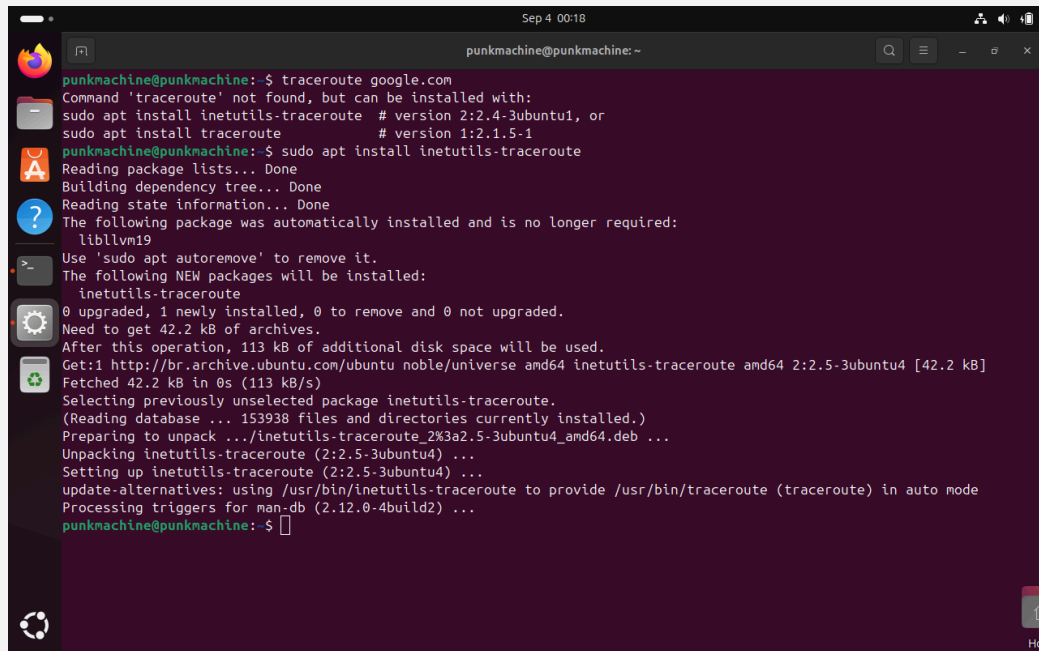
Resultado do comando 'netstat -an' no Ubuntu, exibindo todas as conexões e portas em formato numérico.

3. Observei que a opção `-a` mostrou todas as conexões e portas de escuta, e a opção `-n` exibiu endereços e números de porta em formato numérico, sem tentar resolver nomes de host ou nomes de serviço. Isso me deu uma visão clara de todas as portas abertas e conexões estabelecidas.

9. Executar o comando TRACEROUTE e descrever as informações que ele retorna (Linux).

⚠ Nota: Em algumas versões mais recentes do Ubuntu, o comando `traceroute` pode não estar instalado por padrão. Se você receber um erro de "comando não encontrado", pode instalá-lo com o seguinte comando:

```
sudo apt install traceroute
```

A screenshot of a terminal window titled 'punkmachine@punkmachine: ~'. The terminal shows the following sequence of commands and output:
1. Command: `traceroute google.com`
Output: `Command 'traceroute' not found, but can be installed with:
sudo apt install inetutils-traceroute # version 2:2.4-3ubuntu1, or
sudo apt install traceroute # version 1:2.1.5-1`
2. Command: `sudo apt install traceroute`
Output: `Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
libllvm19
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
inetutils-traceroute
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 42.2 kB of archives.
After this operation, 113 kB of additional disk space will be used.
Get:1 http://br.archive.ubuntu.com/ubuntu noble/universe amd64 inetutils-traceroute amd64 2:2.5-3ubuntu4 [42.2 kB]
Fetched 42.2 kB in 0s (113 kB/s)
Selecting previously unselected package inetutils-traceroute.
(Reading database ... 153938 files and directories currently installed.)
Preparing to unpack .../inetutils-traceroute_2%3a2.5-3ubuntu4_amd64.deb ...
Unpacking inetutils-traceroute (2:2.5-3ubuntu4) ...
Setting up inetutils-traceroute (2:2.5-3ubuntu4) ...
update-alternatives: using /usr/bin/inetutils-traceroute to provide /usr/bin/traceroute (traceroute) in auto mode
Processing triggers for man-db (2.12.0-4build2) ...
punkmachine@punkmachine: $`

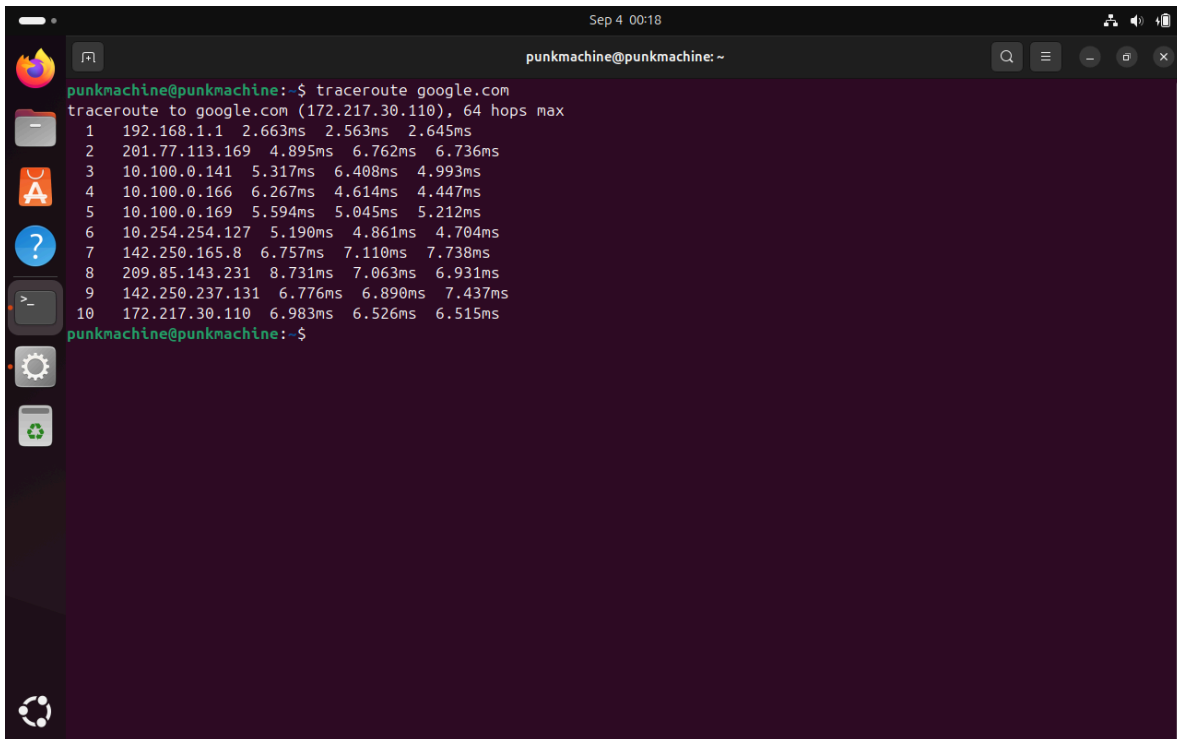
Execução do comando 'sudo apt install traceroute' no terminal do Ubuntu.

Após instalar o `traceroute` (devido ao erro de 'comando não encontrado'), executei-o no Ubuntu (VM2) para rastrear a rota que os pacotes percorrem até um destino.

Passo a passo:

1. Na VM2 (Ubuntu), abri o Terminal.
2. Executei o comando para um destino externo (por exemplo, `google.com`):

```
traceroute google.com
```



```
punkmachine@punkmachine:~$ traceroute google.com
traceroute to google.com (172.217.30.110), 64 hops max
 1  192.168.1.1  2.663ms  2.563ms  2.645ms
 2  201.77.113.169  4.895ms  6.762ms  6.736ms
 3  10.100.0.141  5.317ms  6.408ms  4.993ms
 4  10.100.0.166  6.267ms  4.614ms  4.447ms
 5  10.100.0.169  5.594ms  5.045ms  5.212ms
 6  10.254.254.127  5.190ms  4.861ms  4.704ms
 7  142.250.165.8  6.757ms  7.110ms  7.738ms
 8  209.85.143.231  8.731ms  7.063ms  6.931ms
 9  142.250.237.131  6.776ms  6.890ms  7.437ms
10  172.217.30.110  6.983ms  6.526ms  6.515ms
punkmachine@punkmachine:~$
```

Resultado do comando 'traceroute google.com' no Ubuntu, mostrando a rota de pacotes até o destino.

3. Observei que o comando listou todos os roteadores (hops) pelos quais os pacotes passaram até chegar ao destino, juntamente com o tempo de resposta para cada hop. Isso me ajudou a identificar possíveis gargalos ou problemas de rota na rede.

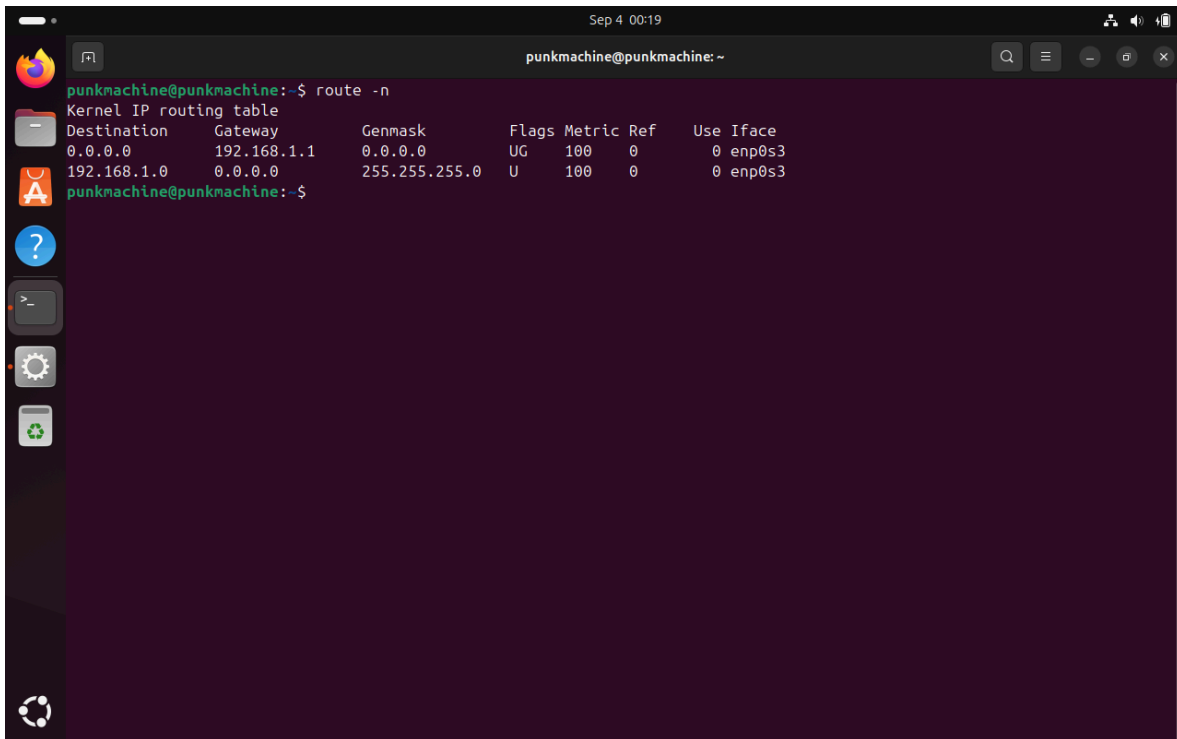
10. Executar o comando ROUTE e descrever as informações que ele retorna (Linux).

Executei o comando `route` no Ubuntu (VM2) para exibir e manipular a tabela de roteamento IP.

Passo a passo:

1. Na VM2 (Ubuntu), abri o Terminal.
2. Executei o comando:

```
route -n
```



```
punkmachine@punkmachine:~$ route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 192.168.1.1 0.0.0.0 UG 100 0 0 enp0s3
192.168.1.0 0.0.0.0 255.255.255.0 U 100 0 0 enp0s3
punkmachine@punkmachine:~$
```

Resultado do comando 'route -n' no Ubuntu, exibindo a tabela de roteamento IP.

3. Observei que o comando retornou a tabela de roteamento, mostrando as rotas conhecidas pelo sistema, incluindo o destino, gateway, máscara de sub-rede, interface e métrica. A opção `-n` exibiu os endereços numericamente.

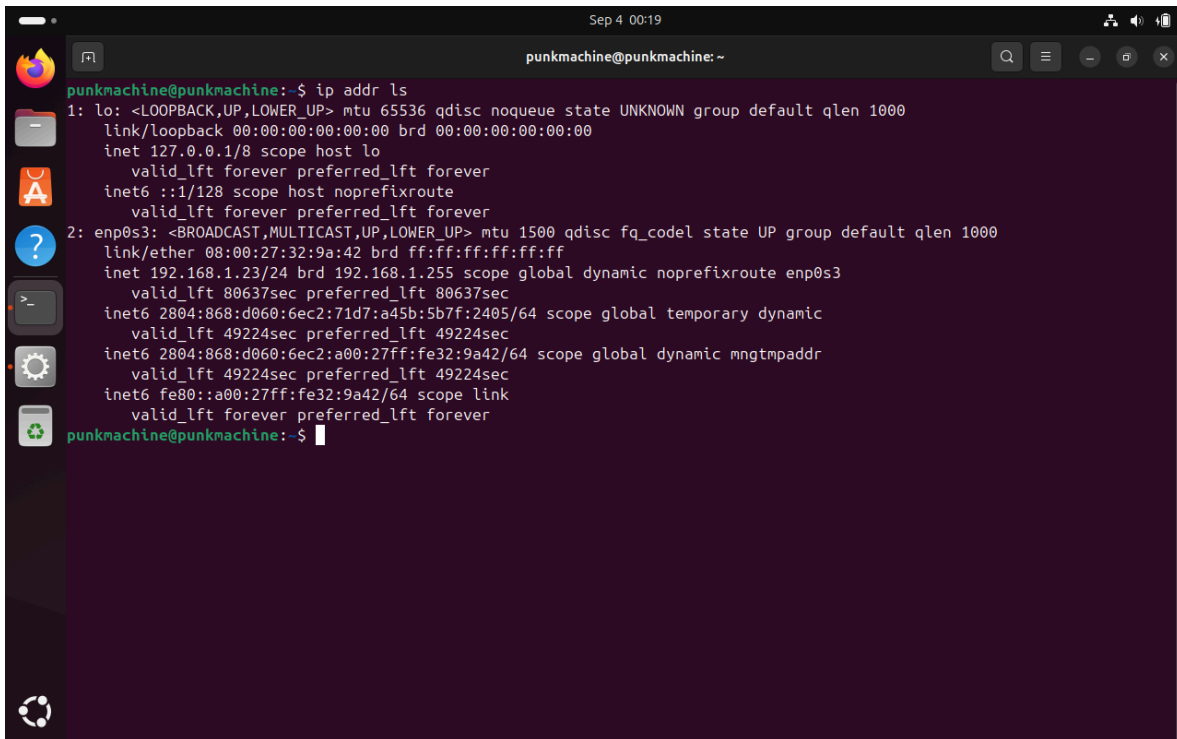
11. Executar o comando `ip addr ls` e descrever as informações que ele retorna (Linux).

Executei o comando `ip addr ls` no Ubuntu (VM2) para exibir informações detalhadas sobre os endereços IP e interfaces de rede.

Passo a passo:

1. Na VM2 (Ubuntu), abri o Terminal.
2. Executei o comando:

```
ip addr ls
```

A terminal window titled 'punkmachine@punkmachine: ~' showing the output of the 'ip addr ls' command. The output lists details for two network interfaces: 'lo' (loopback) and 'enp0s3' (ethernet).

```
punkmachine@punkmachine:~$ ip addr ls
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 08:00:27:32:9a:42 brd ff:ff:ff:ff:ff:ff
   inet 192.168.1.23/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3
       valid_lft 80637sec preferred_lft 80637sec
   inet6 2804:868:d060:6ec2:71d7:a45b:5b7f:2405/64 scope global temporary dynamic
       valid_lft 49224sec preferred_lft 49224sec
   inet6 2804:868:d060:6ec2:a00:27ff:fe32:9a42/64 scope global dynamic mngtmpaddr
       valid_lft 49224sec preferred_lft 49224sec
   inet6 fe80::a00:27ff:fe32:9a42/64 scope link
       valid_lft forever preferred_lft forever
punkmachine@punkmachine:~$
```

Resultado do comando 'ip addr ls' no Ubuntu, com detalhes sobre as interfaces de rede.

3. Observei que o comando retornou informações sobre todas as interfaces de rede, incluindo seus endereços IP (IPv4 e IPv6), máscaras de sub-rede, estado (UP/DOWN) e endereços MAC. É uma ferramenta mais moderna e abrangente que `ifconfig`.

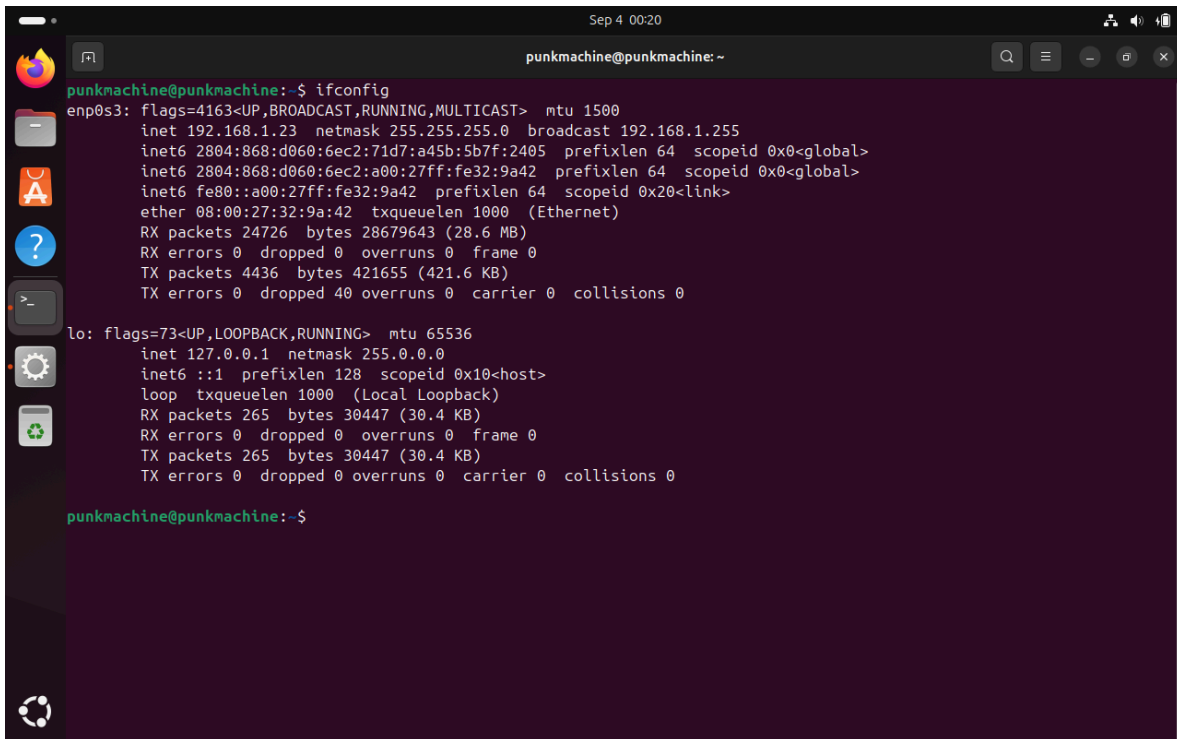
12. Executar o comando `ifconfig` e descrever as informações que ele retorna (Linux).

Executei o comando `ifconfig` no Ubuntu (VM2) para exibir e configurar interfaces de rede.

Passo a passo:

1. Na VM2 (Ubuntu), abri o Terminal.
2. Executei o comando:

```
ifconfig
```



```
punkmachine@punkmachine:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.23 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 2804:868:d060:6ec2:71d7:a45b:5b7f:2405 prefixlen 64 scopeid 0x0<global>
    inet6 2804:868:d060:6ec2:a00:27ff:fe32:9a42 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::a00:27ff:fe32:9a42 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:32:9a:42 txqueuelen 1000 (Ethernet)
    RX packets 24726 bytes 28679643 (28.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4436 bytes 421655 (421.6 KB)
    TX errors 0 dropped 40 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 265 bytes 30447 (30.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 265 bytes 30447 (30.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

punkmachine@punkmachine:~$
```

Resultado do comando 'ifconfig' no Ubuntu, mostrando a configuração das interfaces de rede.

3. Observei que o comando retornou informações sobre as interfaces de rede ativas, incluindo o endereço IP, máscara de sub-rede, endereço MAC, estatísticas de pacotes (enviados/recebidos) e erros. Embora ainda seja usado, `ip addr ls` é a ferramenta preferida em sistemas Linux modernos.

Usando Termux do seu celular

1. Executar o comando ping entre as VMs e o celular usando Termux.

Para verificar a conectividade entre as VMs e o celular via Termux, utilizei o comando `ping`.

Passo a passo:

1. No meu celular, abri o aplicativo Termux.
2. Obtive o endereço IP do meu celular no Termux.

ifconfig



Nota: Notei que o comando `ip a` não funcionou corretamente no Termux por falta de acesso root. Por isso, utilizei o `ifconfig`. O comando já estava disponível no meu Termux, mas caso não estivesse, a instalação seria feita com `pkg install net-tools`.

```
18:25 >_ ☀️ 📶 73%

- $ ifconfig
Warning: cannot open /proc/net/dev (Permission denied). Limited output.
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 1 (UNSPEC)

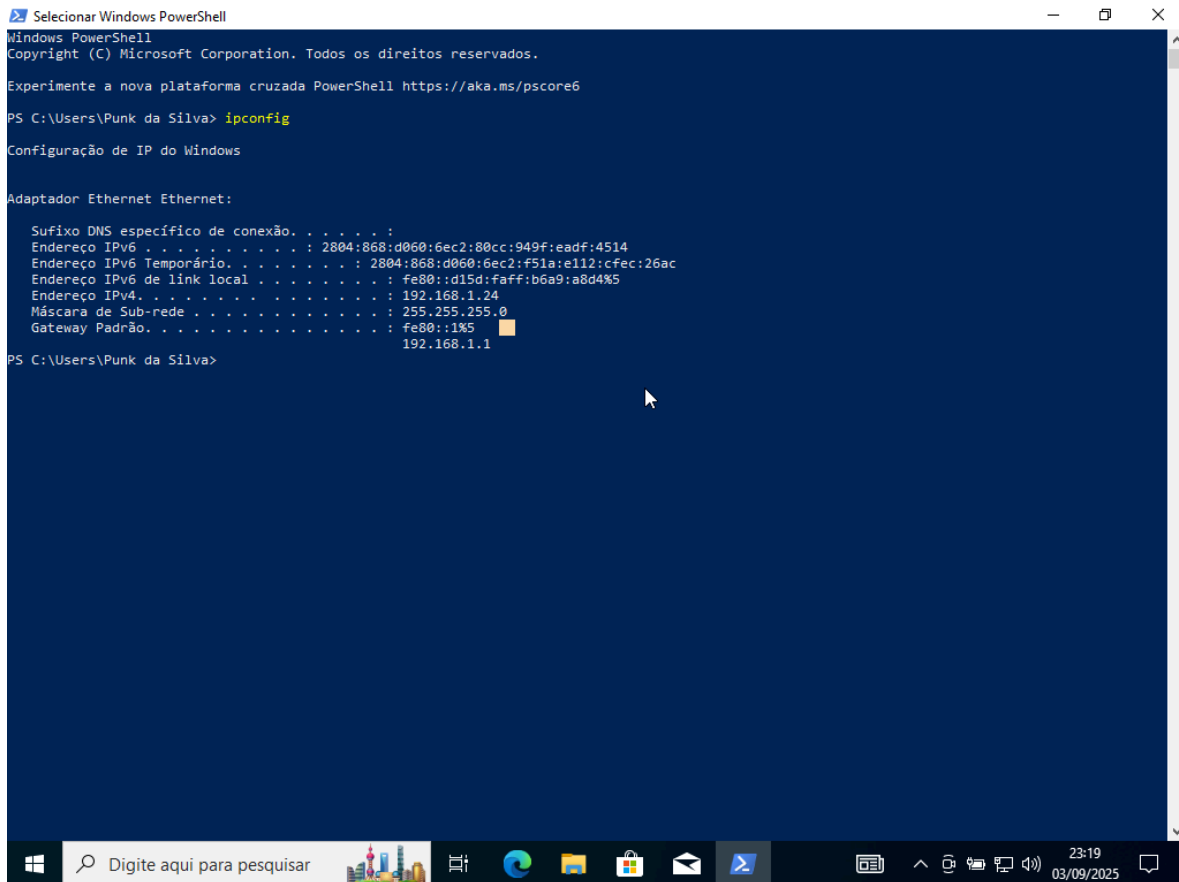
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.2 netmask 255.255.255.0 broadcast 192.168.1.255
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 1000 (UNSPEC)

- $ █
```

Resultado do comando 'ifconfig' no Termux, exibindo o endereço IP do celular.

3. Obtive os endereços IP da VM1 (Windows 10) e VM2 (Ubuntu).

- No Windows 10 (VM1), utilizei `ipconfig`.



```
Selecionar Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6

PS C:\Users\Punk da Silva> ipconfig

Configuração de IP do Windows

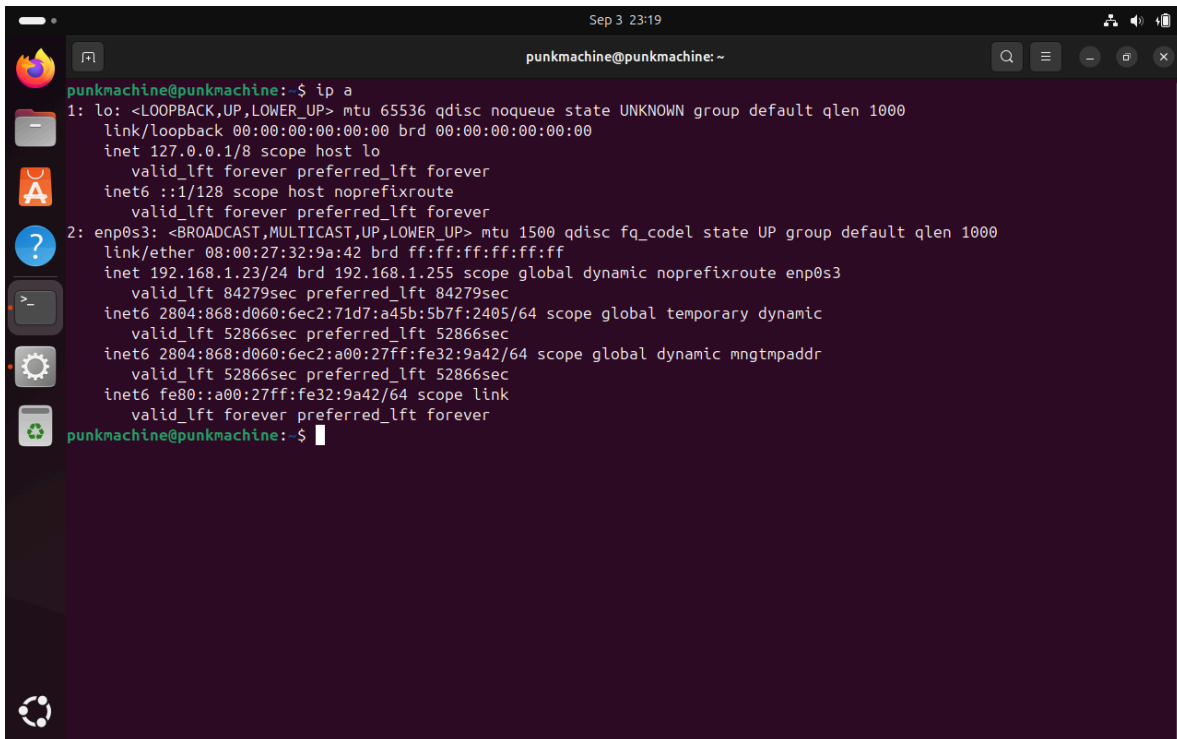
Adaptador Ethernet Ethernet:

    Sufixo DNS específico de conexão. . . . . : 
    Endereço IPv6 . . . . . : 2804:868:d060:6ec2:80cc:949f:eadf:4514
    Endereço IPv6 Temporário. . . . . : 2804:868:d060:6ec2:f51a:e112:cfec:26ac
    Endereço IPv6 de link local . . . . . : fe80::d15d:faff:b6a9:a8d4%5
    Endereço IPv4. . . . . : 192.168.1.24
    Máscara de Sub-rede . . . . . : 255.255.255.0
    Gateway Padrão. . . . . : fe80::1%5
                                192.168.1.1

PS C:\Users\Punk da Silva>
```

Resultado do comando `ipconfig` no Prompt de Comando do Windows, mostrando o endereço IPv4, máscara de sub-rede e gateway padrão.

- No Ubuntu (VM2), utilizei `ip a` ou `ifconfig`.

A terminal window titled 'punkmachine@punkmachine: ~' with a search bar and window controls. The terminal shows the output of the 'ip a' command. It lists details for the loopback interface 'lo' (127.0.0.1) and the ethernet interface 'enp0s3' (192.168.1.23). The output includes MTU, link type, MAC address, and various flags like 'LOOPBACK', 'UP', 'LOWER_UP', 'BROADCAST', 'MULTICAST', 'DYNAMIC', and 'PROMISC'. The prompt 'punkmachine@punkmachine:~\$' is visible at the bottom.

```
punkmachine@punkmachine:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:32:9a:42 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.23/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3
        valid_lft 84279sec preferred_lft 84279sec
    inet6 2804:868:d060:6ec2:71d7:a45b:5b7f:2405/64 scope global temporary dynamic
        valid_lft 52866sec preferred_lft 52866sec
    inet6 2804:868:d060:6ec2:a00:27ff:fe32:9a42/64 scope global dynamic mngtppaddr
        valid_lft 52866sec preferred_lft 52866sec
    inet6 fe80::a00:27ff:fe32:9a42/64 scope link
        valid_lft forever preferred_lft forever
punkmachine@punkmachine:~$
```

Resultado do comando 'ip a' no terminal do Ubuntu, exibindo os detalhes das interfaces de rede e seus endereços IP.

4. Executei o comando `ping` do Termux para as VMs.

- Do Termux para a VM1 (Windows 10):

```
ping <IP_DA_VM1>
```

```
19:08 >_ [icons] 70%
c64 bytes from 192.168.1.23: icmp_seq=942 ttl=64 time=59.2 ms
64 bytes from 192.168.1.23: icmp_seq=943 ttl=64 time=86.9 ms
64 bytes from 192.168.1.23: icmp_seq=944 ttl=64 time=65.0 ms
8263000000 5 08250097091 6 10043043192 2 1956504350
3 364 bytes from 192.168.1.23: icmp_seq=945 ttl=64 time=52.8 ms
/64 bytes from 192.168.1.23: icmp_seq=946 ttl=64 time=121 ms
/64 bytes from 192.168.1.23: icmp_seq=947 ttl=64 time=59.2 ms
64 bytes from 192.168.1.23: icmp_seq=948 ttl=64 time=56.9 ms
64 bytes from 192.168.1.23: icmp_seq=950 ttl=64 time=81.9 ms
64 bytes from 192.168.1.23: icmp_seq=951 ttl=64 time=79.8 ms
64 bytes from 192.168.1.23: icmp_seq=952 ttl=64 time=87.2 ms
/64 bytes from 192.168.1.23: icmp_seq=953 ttl=64 time=55.3 ms
64 bytes from 192.168.1.23: icmp_seq=954 ttl=64 time=84.0 ms
64 bytes from 192.168.1.23: icmp_seq=955 ttl=64 time=102 ms
64 bytes from 192.168.1.23: icmp_seq=956 ttl=64 time=70.0 ms
/64 bytes from 192.168.1.23: icmp_seq=957 ttl=64 time=58.1 ms
64 bytes from 192.168.1.23: icmp_seq=959 ttl=64 time=94.3 ms
64 bytes from 192.168.1.23: icmp_seq=960 ttl=64 time=14.1 ms
64 bytes from 192.168.1.23: icmp_seq=961 ttl=64 time=5.56 ms
64 bytes from 192.168.1.23: icmp_seq=962 ttl=64 time=5.26 ms
64 bytes from 192.168.1.23: icmp_seq=963 ttl=64 time=80.0 ms
64 bytes from 192.168.1.23: icmp_seq=964 ttl=64 time=102 ms
64 bytes from 192.168.1.23: icmp_seq=965 ttl=64 time=22.8 ms
^C
--- 192.168.1.23 ping statistics ---
965 packets transmitted, 874 received, 9% packet loss,
time 1143540ms
rtt min/avg/max/mdev = 4.624/95.405/1210.705/180.805 ms
, pipe 2
$ █
```

Resultado do comando ping do Termux para a VM com Windows.

- Do Termux para a VM2 (Ubuntu):

```
ping <IP_DA_VM2>
```

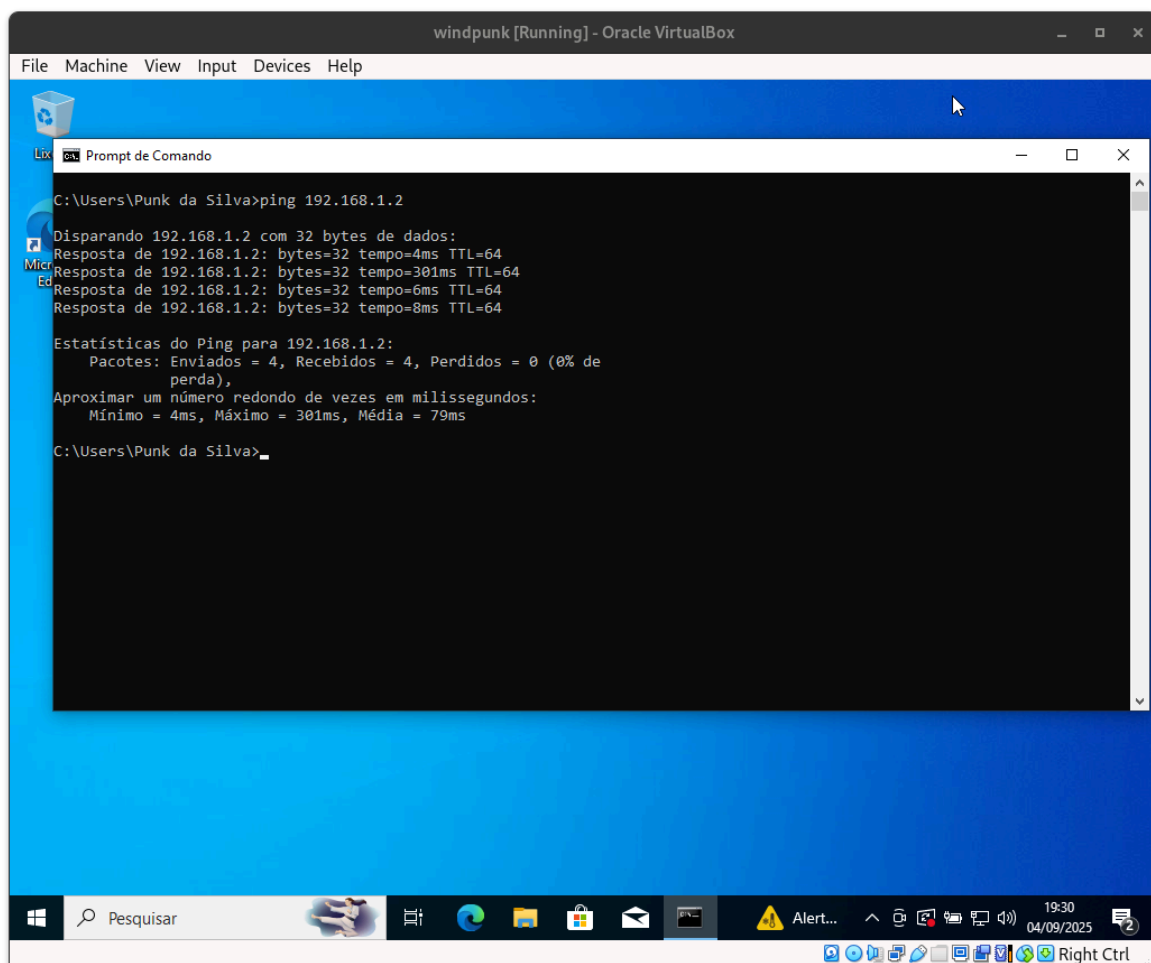
```
19:10 >_ [icons] 68%
141 ms
64 bytes from 192.168.1.24: icmp_seq=1571 ttl=128 time=
60.0 ms
64 bytes from 192.168.1.24: icmp_seq=1572 ttl=128 time=
89.2 ms
64 bytes from 192.168.1.24: icmp_seq=1573 ttl=128 time=
117 ms
64 bytes from 192.168.1.24: icmp_seq=1574 ttl=128 time=
125 ms
64 bytes from 192.168.1.24: icmp_seq=1575 ttl=128 time=
54.6 ms
64 bytes from 192.168.1.24: icmp_seq=1576 ttl=128 time=
73.5 ms
64 bytes from 192.168.1.24: icmp_seq=1577 ttl=128 time=
92.2 ms
64 bytes from 192.168.1.24: icmp_seq=1578 ttl=128 time=
120 ms
64 bytes from 192.168.1.24: icmp_seq=1579 ttl=128 time=
50.1 ms
64 bytes from 192.168.1.24: icmp_seq=1580 ttl=128 time=
56.6 ms
64 bytes from 192.168.1.24: icmp_seq=1581 ttl=128 time=
85.1 ms
64 bytes from 192.168.1.24: icmp_seq=1582 ttl=128 time=
103 ms
64 bytes from 192.168.1.24: icmp_seq=1583 ttl=128 time=
132 ms
/64 bytes from 192.168.1.24: icmp_seq=1584 ttl=128 time
=40.6 ms
64 bytes from 192.168.1.24: icmp_seq=1585 ttl=128 time=
70.0 ms
64 bytes from 192.168.1.24: icmp_seq=1586 ttl=128 time=
88.4 ms
64 bytes from 192.168.1.24: icmp_seq=1587 ttl=128 time=
117 ms
64 bytes from 192.168.1.24: icmp_seq=1588 ttl=128 time=
103 ms
64 bytes from 192.168.1.24: icmp_seq=1589 ttl=128 time=
34.0 ms
64 bytes from 192.168.1.24: icmp_seq=1590 ttl=128 time=
35.7 ms
64 bytes from 192.168.1.24: icmp_seq=1591 ttl=128 time=
78.2 ms
64 bytes from 192.168.1.24: icmp_seq=1592 ttl=128 time=
91.1 ms
^C
--- 192.168.1.24 ping statistics ---
1592 packets transmitted, 1441 received, 9% packet loss
, time 1945922ms
rtt min/avg/max/mdev = 4.569/99.914/1170.002/158.854 ms
, pipe 2
$ 8
```

Resultado do comando ping do Termux para a VM com Ubuntu.

5. Executei o comando `ping` das VMs para o Termux.

- Da VM1 (Windows 10) para o Termux:

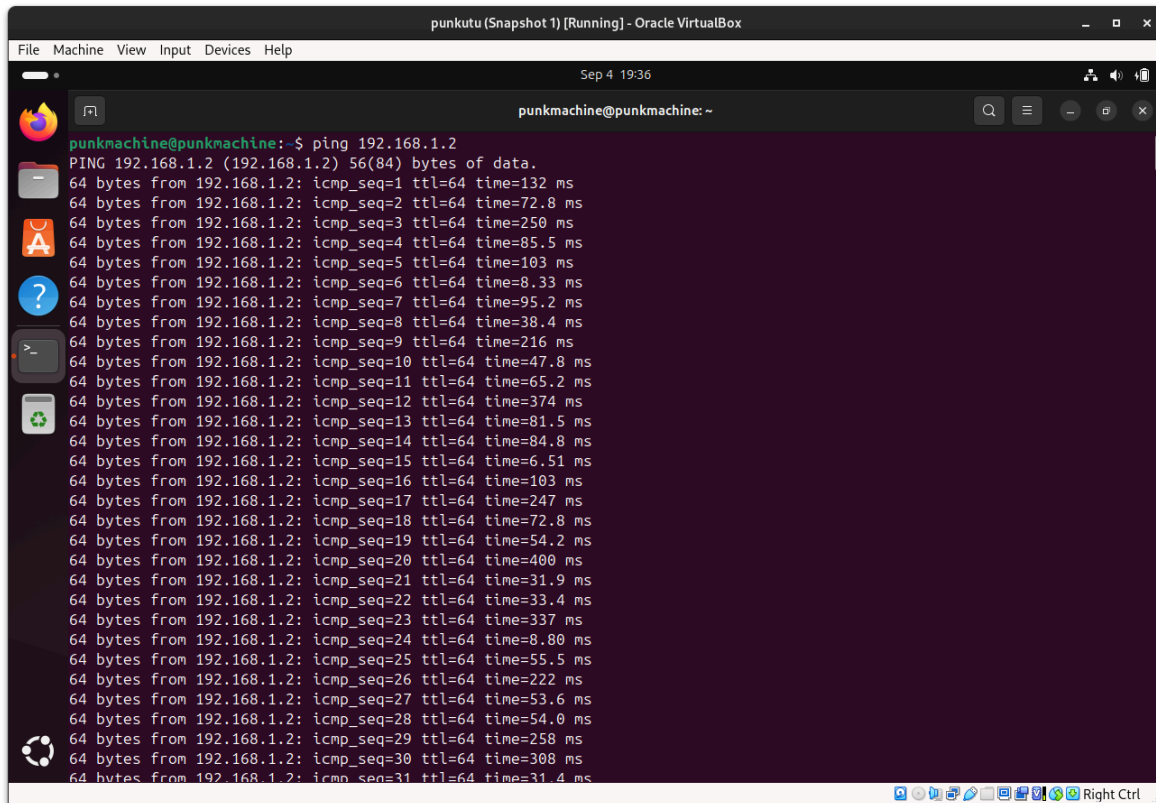
```
ping <IP_DO_CELULAR_TERMUX>
```



Resultado do comando ping da VM Windows para o celular com Termux.

- Da VM2 (Ubuntu) para o Termux:

```
ping <IP_DO_CELULAR_TERMUX>
```



Resultado do comando ping da VM Ubuntu para o celular com Termux.

6. Observei os resultados para confirmar a conectividade.

2. Executar o comando ls e descrever as informações que ele retorna.

Executei o comando `ls` no Termux para listar o conteúdo do diretório atual.

Passo a passo:

1. No meu celular, abri o aplicativo Termux.

2. Executei o comando:

```
ls
```

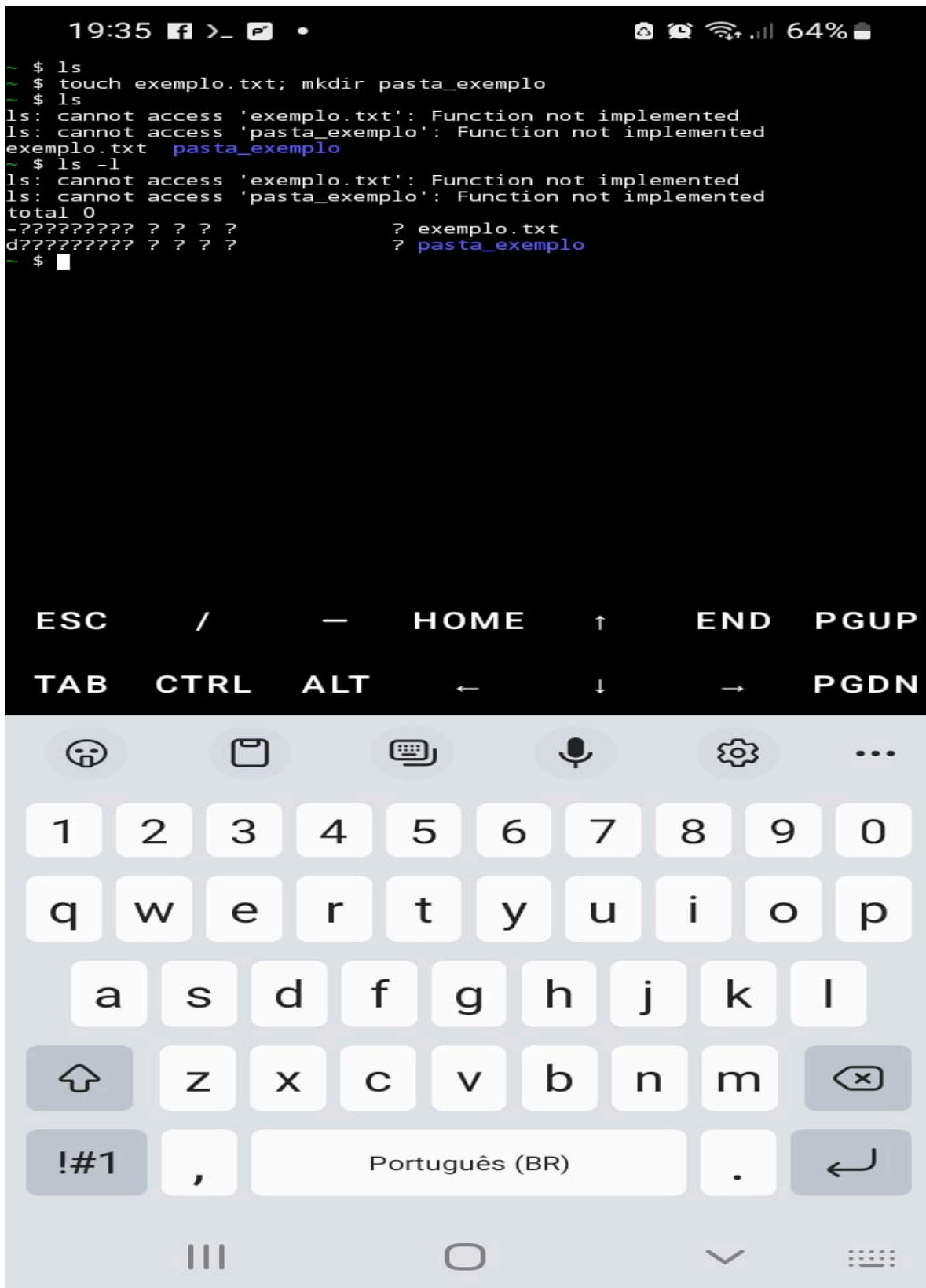
3. O comando não retornou nenhuma saída, indicando que o diretório inicial estava vazio.

4. Para testar, criei um arquivo chamado `exemplo.txt` e um diretório chamado `pasta_exemplo`:

```
touch exemplo.txt  
mkdir pasta_exemplo
```

5. Ao executar `ls` novamente, a saída foi a seguinte:

```
exemplo.txt  pasta_exemplo
```

Sequência de comandos no Termux: 'ls' em um diretório vazio, 'touch' e 'mkdir' para criar arquivo e pasta, e 'ls -l' para listar o conteúdo com detalhes.

6. Isso confirmou que o comando `ls` lista o conteúdo do diretório. Observei também que, usando `ls -l`, mais detalhes são exibidos, como permissões, proprietário, tamanho e data de modificação.