



UTILIZAÇÃO DE MÁQUINAS VIRTUAIS PARA SIMULAR O CLIENTE E O SERVIDOR WEB

Estudo de Caso - Development Tools & Cloud Computing



01 DE JULHO DE 2025
UNIFECAP

Sumário

Introdução	3
Criação das Máquinas Virtuais no VirtualBox	3
Máquina Virtual – Cliente	3
Máquina Virtual – Server.....	4
Instalação do Sistema Operacional.....	4
Usuário “cliente”:	4
Usuário “server”:	6
Configuração do WebServer.....	7
Instalação do Apache2.....	7
Verificação do status do Apache2.....	8
Criação da Página HTML com Mini Currículo	9
Testes realizados	10
Endereço IP da VM server	10
Verificação de Conectividade – Ping	10
Acesso à Página Web no Navegador (cliente)	11
Teoria na prática: Facebook e seus laboratórios de testes com Mac Minis	12
Conclusão.....	12
Minha Autoavaliação	12
O que foi feito bem?	12
O que pode ser melhorado?	12
Como posso aplicar esse aprendizado em situações futuras?	12
Bibliografias utilizadas	13

Introdução

Este relatório descreve o processo de criação e configuração de um ambiente virtualizado que simula o cenário de um cliente com dificuldades na implementação de um servidor web Linux. Foram utilizadas duas máquinas virtuais no VirtualBox, uma representando o cliente e outra representando o servidor web. O servidor foi configurado com Apache2 e substituí a página padrão por um mini currículo pessoal em HTML.

Criação das Máquinas Virtuais no VirtualBox

Máquina Virtual – Cliente

- Nome da VM: Cliente
- Sistema Operacional: Ubuntu Desktop 24.04.2 LTS
- Memória RAM: 4 GB
- HD: 20 GB
- Processador: 1 CPU
- Rede: Bridge
- Usuário: cliente



Máquina Virtual – Server

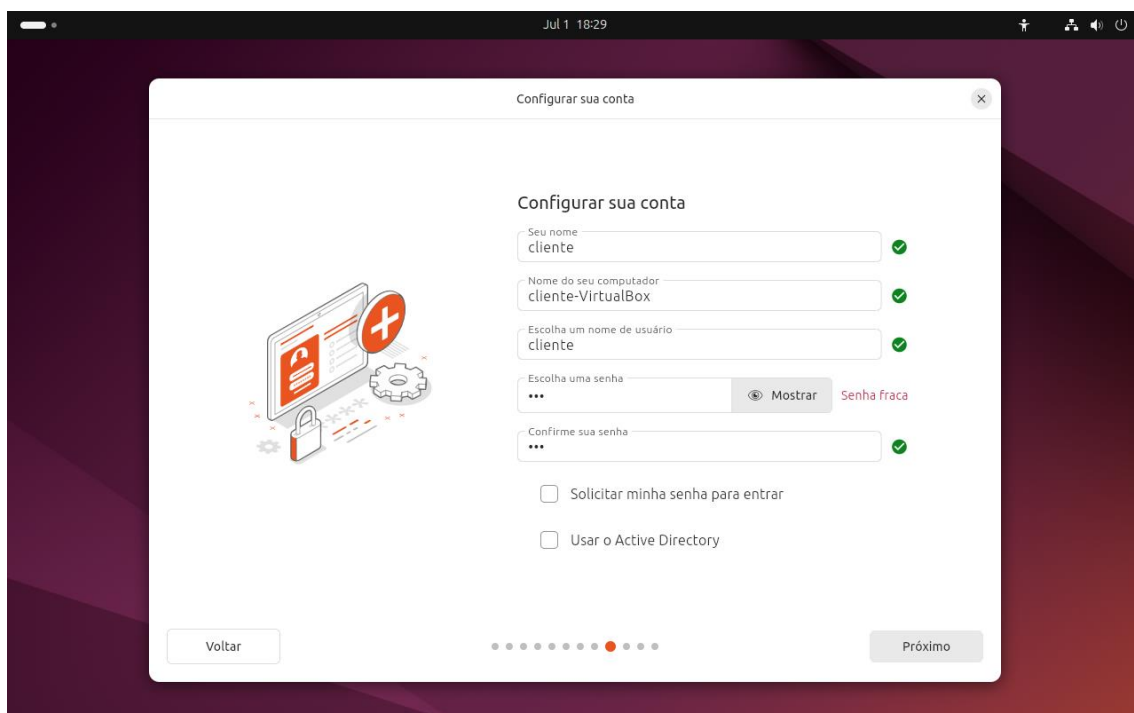
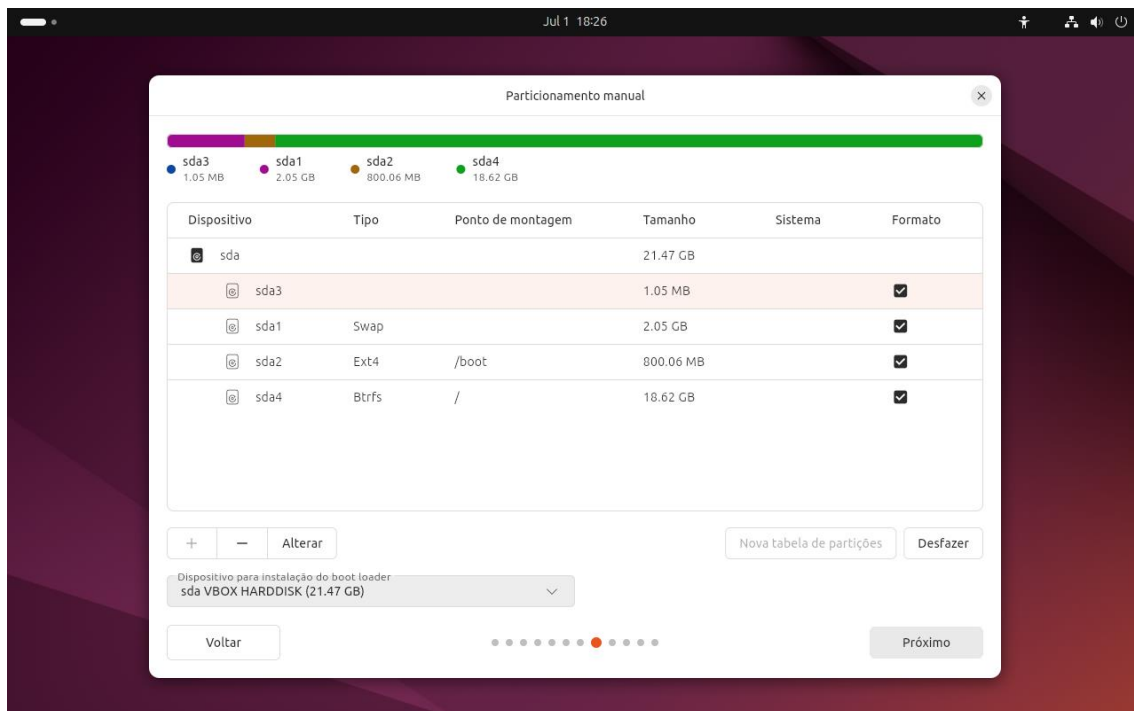
- Nome da VM: Server
- Sistema Operacional: Ubuntu Desktop 24.04.2 LTS
- Memória RAM: 4 GB
- HD: 40 GB
- Processador: 1 CPU
- Rede: Bridge
- Usuário: server

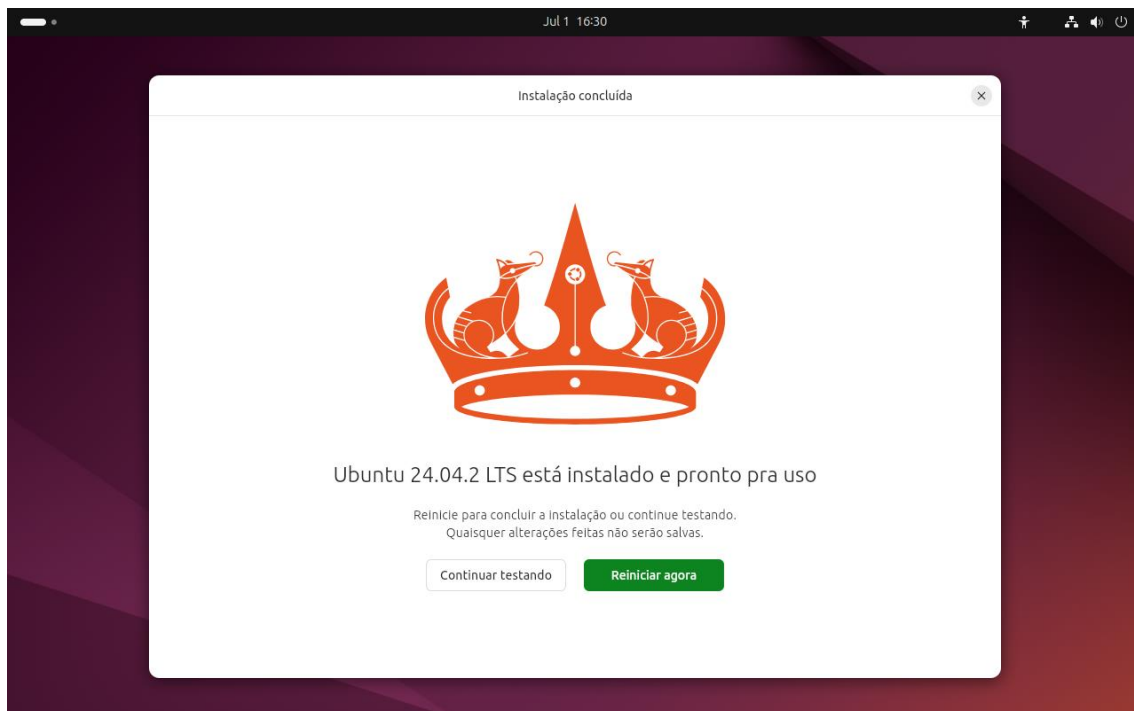


Instalação do Sistema Operacional

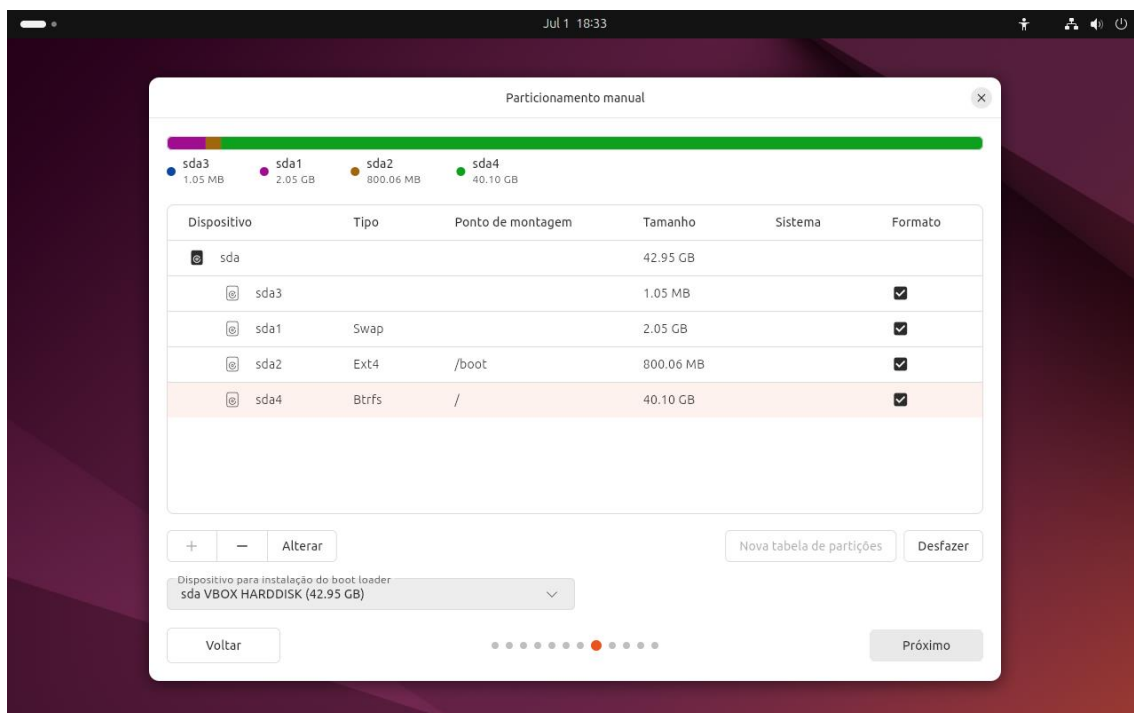
É importante lembrar que ambas as VMs foram iniciadas com a imagem ISO do Ubuntu Desktop 24.04.1 LTS e passaram pelo processo padrão de instalação, com a criação dos usuários especificados anteriormente.

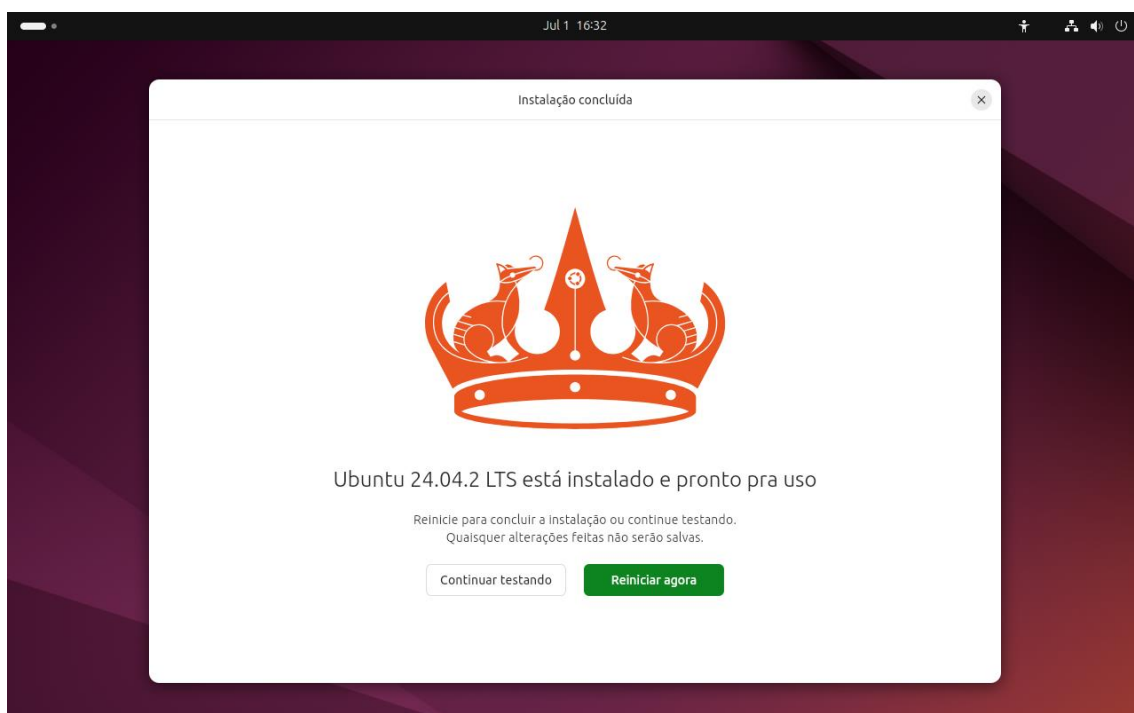
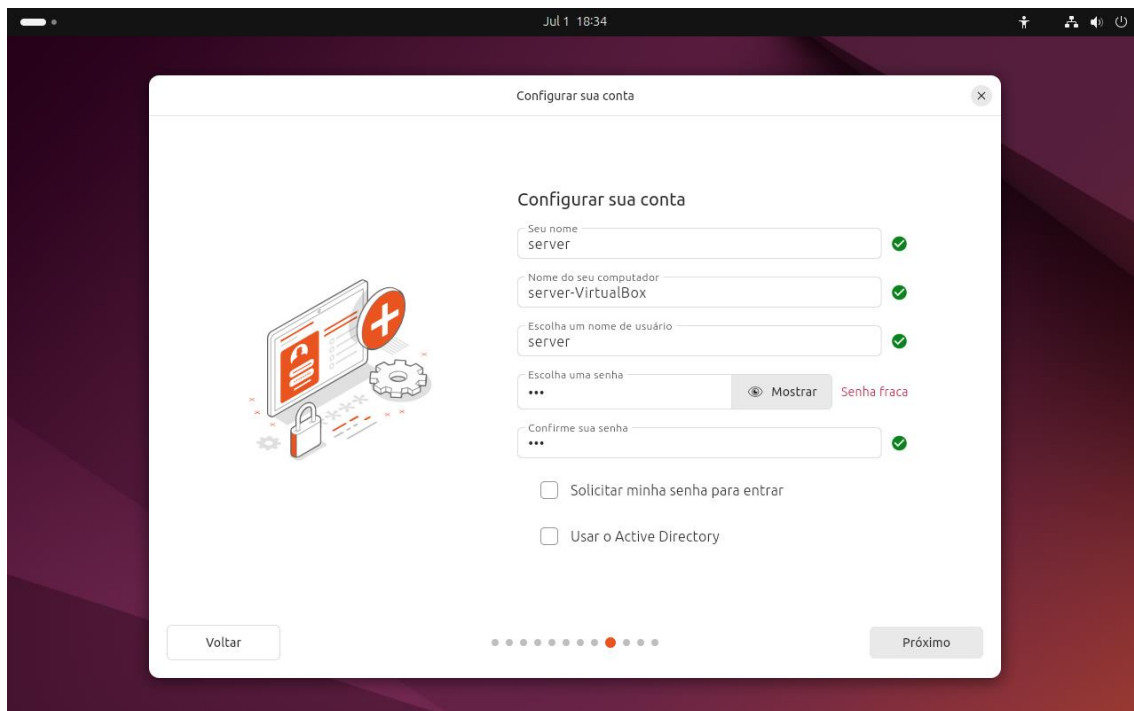
Usuário “cliente”:





Usuário “server”:





Configuração do WebServer

Instalação do Apache2

No terminal da máquina “server”, foi executado o comando “sudo apt install apache2 -y” para instalar o Apache.

```

server@server-VirtualBox: ~
server@server-VirtualBox:~$ sudo apt install apache2 -y
[sudo] senha para server:
Lendo listas de pacotes... Pronto
Construindo árvore de dependências... Pronto
Lendo informação de estado... Pronto
Os pacotes adicionais seguintes serão instalados:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3
  libaprutil1-ldap libaprutil1t64
Pacotes sugeridos:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
Os NOVOS pacotes a seguir serão instalados:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64
0 pacotes atualizados, 8 pacotes novos instalados, 0 a serem removidos e 105 não atualizados.
É preciso baixar 1.900 kB de arquivos.
Depois desta operação, 7.455 kB adicionais de espaço em disco serão usados.
Obter:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libapr1t64 amd64 1.7.2-3.1ubuntu0.1 [108 kB]
Obter:2 http://archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1t64 amd64 1.6.3-1.1ubuntu7 [91,9 kB]
Obter:3 http://archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.3-1.1ubuntu7 [11,2 kB]
Obter:4 http://archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-ldap amd64 1.6.3-1.1ubuntu7 [9.116 B]
Obter:5 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-bin amd64 2.4.58-1ubuntu8.6 [1.330 kB]
Obter:6 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-data all 2.4.58-1ubuntu8.6 [163 kB]
Obter:7 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-utils amd64 2.4.58-1ubuntu8.6 [97,2 kB]
Obter:8 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2 amd64 2.4.58-1ubuntu8.6 [90,2 kB]
Baixados 1.900 kB em 2s (796 kB/s)
A seleccionar pacote anteriormente não seleccionado libapr1t64:amd64.
(Lendo banco de dados ... 151649 ficheiros e diretórios atualmente instalados.)
A preparar para desempacotar .../0-libapr1t64_1.7.2-3.1ubuntu0.1_amd64.deb ...
A descompactar libapr1t64:amd64 (1.7.2-3.1ubuntu0.1) ...
A seleccionar pacote anteriormente não seleccionado libaprutil1t64:amd64.
A preparar para desempacotar .../1-libaprutil1t64_1.6.3-1.1ubuntu7_amd64.deb ...
A descompactar libaprutil1t64:amd64 (1.6.3-1.1ubuntu7) ...
A seleccionar pacote anteriormente não seleccionado libaprutil1-dbd-sqlite3:amd64

```

Verificação do status do Apache2

Após a instalação, utilizei o comando “sudo systemctl status apache2” para verificar o status do Apache.

```

server@server-VirtualBox: ~
server@server-VirtualBox:~$ sudo systemctl status apache2
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /usr/lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /usr/lib/systemd/system/apache-htcacheclean.service.
A processar 'triggers' para ufw (0.36.2-6) ...
A processar 'triggers' para man-db (2.12.0-4build2) ...
A processar 'triggers' para libc-bin (2.39-0ubuntu8.4) ...
server@server-VirtualBox:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: 
   Active: active (running) since Tue 2025-07-01 18:03:47 -03; 1min 37s ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 6538 (apache2)
      Tasks: 55 (limit: 4605)
     Memory: 5.5M (peak: 5.7M)
        CPU: 81ms
    CGroup: /system.slice/apache2.service
            └─6538 /usr/sbin/apache2 -k start
              └─6540 /usr/sbin/apache2 -k start
                └─6541 /usr/sbin/apache2 -k start

jul 01 18:03:47 server-VirtualBox systemd[1]: Starting apache2.service - The Ap
jul 01 18:03:47 server-VirtualBox apache2ctl[6537]: AH00558: apache2: Could not >
jul 01 18:03:47 server-VirtualBox systemd[1]: Started apache2.service - The Apa
lines 1-16/16 (END)

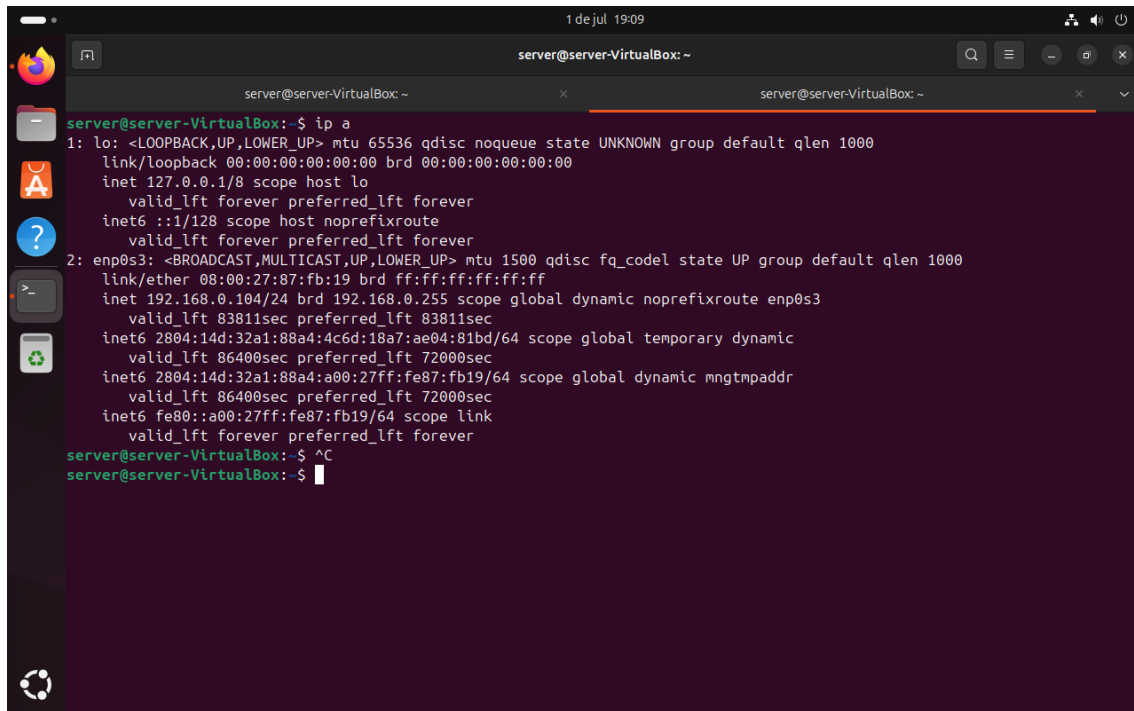
```


Testes realizados

Endereço IP da VM server

Utilizei o comando “ip a” no terminal da VM server para obter o endereço IP local e virtual.

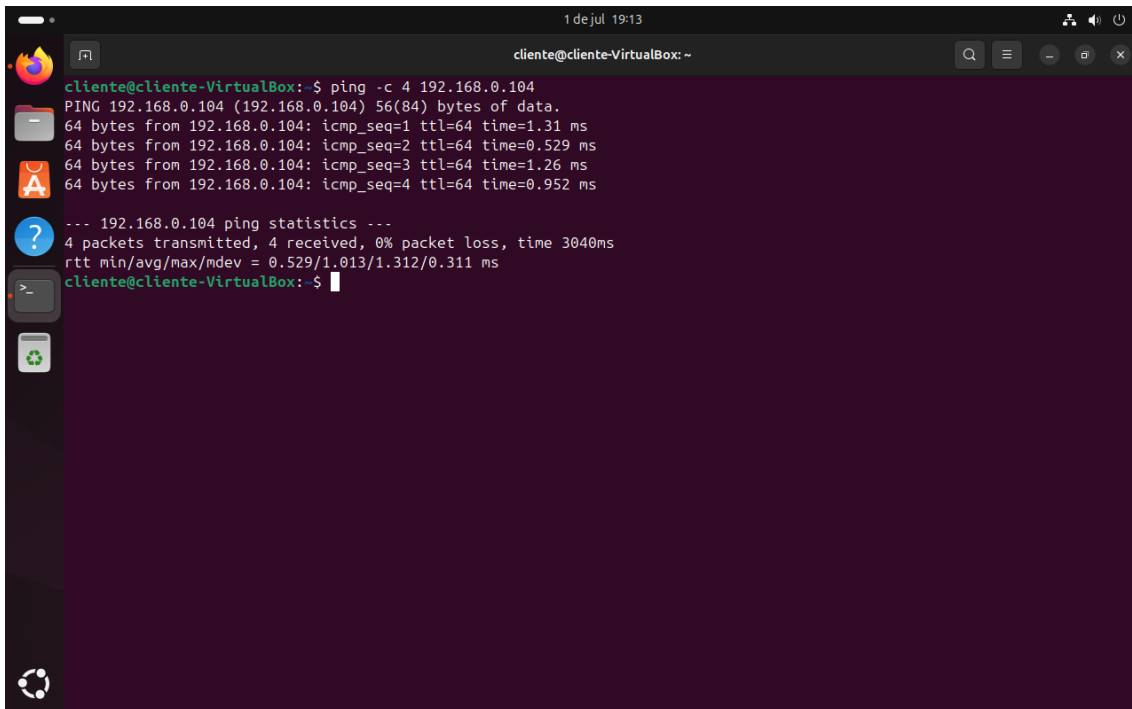
- Localhost: 127.0.0.1/8
- Rede virtual: 192.168.0.104/24



```
server@server-VirtualBox: ~  
server@server-VirtualBox:~$ 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 08:00:27:87:fb:19 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.0.104/24 brd 192.168.0.255 scope global dynamic noprefixroute enp0s3  
        valid_lft 83811sec preferred_lft 83811sec  
    inet6 2804:14d:32a1:88a4:4c6d:18a7:ae04:81bd/64 scope global temporary dynamic  
        valid_lft 86400sec preferred_lft 72000sec  
    inet6 2804:14d:32a1:88a4:a00:27ff:fe87:fb19/64 scope global dynamic mngtmpaddr  
        valid_lft 86400sec preferred_lft 72000sec  
    inet6 fe80::a00:27ff:fe87:fb19/64 scope link  
        valid_lft forever preferred_lft forever  
server@server-VirtualBox:~$ ^C  
server@server-VirtualBox:~$
```

Verificação de Conectividade – Ping

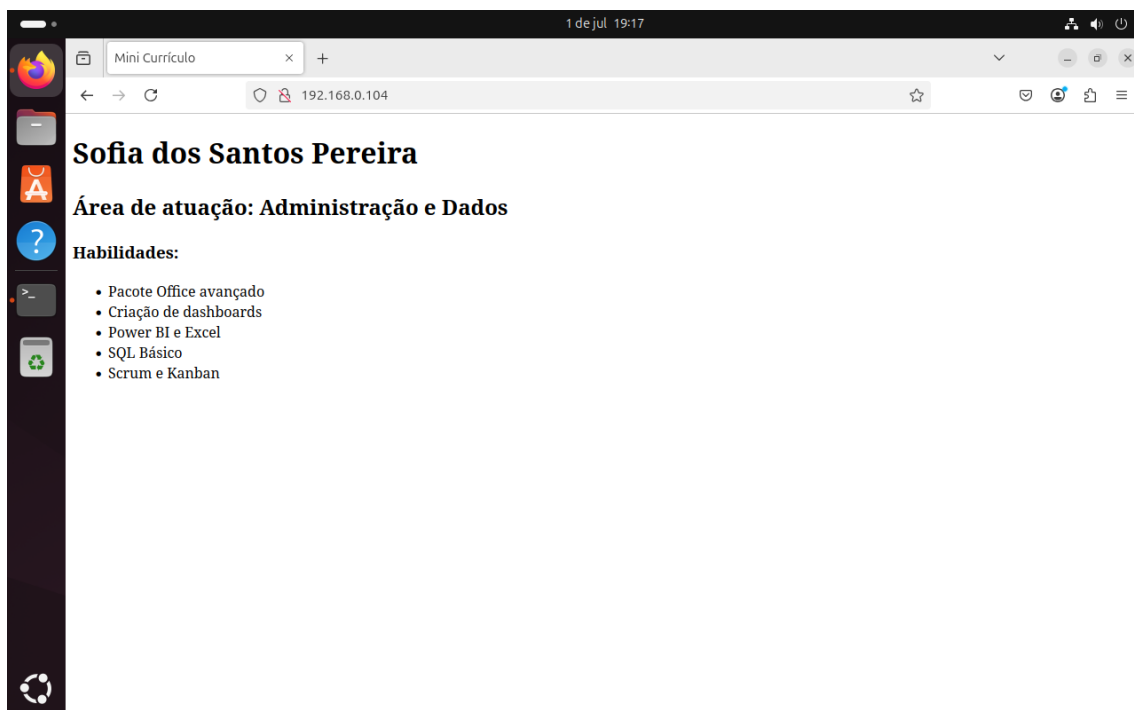
Após obter o IP do servidor, utilizei o comando “ping -c 4 192.168.0.104” na VM cliente para obter o Ping. O parâmetro “-c 4” limita a quantidade de pacotes gerados para apenas 4. Isso nos garante uma interface mais limpa no terminal, que é ideal para esse trabalho.



```
cliente@cliente-VirtualBox: ~  
cliente@cliente-VirtualBox:~$ ping -c 4 192.168.0.104  
PING 192.168.0.104 (192.168.0.104) 56(84) bytes of data.  
64 bytes from 192.168.0.104: icmp_seq=1 ttl=64 time=1.31 ms  
64 bytes from 192.168.0.104: icmp_seq=2 ttl=64 time=0.529 ms  
64 bytes from 192.168.0.104: icmp_seq=3 ttl=64 time=1.26 ms  
64 bytes from 192.168.0.104: icmp_seq=4 ttl=64 time=0.952 ms  
  
--- 192.168.0.104 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3040ms  
rtt min/avg/max/mdev = 0.529/1.013/1.312/0.311 ms  
cliente@cliente-VirtualBox:~$
```

Acesso à Página Web no Navegador (cliente)

Após a realização de todos os testes, foi possível acessar o Mini Currículo na VM cliente através do endereço IP do servidor (192.168.0.104).



Teoria na prática: Facebook e seus laboratórios de testes com Mac Minis

A Facebook opera extensas “fazendas” de Mac Mini e salas de servidores Linux para testar suas aplicações (como o app iOS), utilizando precisamente o que aprendemos até aqui: várias máquinas virtuais e servidores interconectados para replicar ambientes de produção e testes.

Eles usam milhares de Mac Minis rodando macOS para testar iOS/macOS apps – isso é necessário porque o iOS só pode ser testado em hardware Apple.

A gigante da tecnologia roda “racks and racks” de Mac Minis em data centers – uma rede local de máquinas que se comunicam entre si, similar à VM cliente acessando o servidor Apache nesse projeto. Com isso, eles conduzem testes contínuos de centenas de milhares de builds várias vezes por hora, garantindo qualidade mesmo em uma escala gigante.

Conclusão

A simulação do ambiente proposto foi realizada com sucesso. As máquinas virtuais foram configuradas corretamente, o servidor Apache2 está funcional e o usuário consegue acessar a página web personalizada hospedada na máquina servidora. Isso demonstra que o laboratório está adequado para reproduzir o ambiente do cliente real e fornecer o suporte necessário.

Minha Autoavaliação

O que foi feito bem?

O projeto cumpriu com êxito o objetivo de simular um ambiente cliente-servidor realista, refletindo uma boa compreensão sobre comunicação em redes locais, serviços web e uso de máquinas virtuais como laboratório seguro e isolado.

O que pode ser melhorado?

Melhorias podem ser aplicadas por meio da automação de tarefas rotineiras com shell script, uso de snapshots para controle de estado e configuração de IPs estáticos, o que reflete práticas reais de DevOps e administração de sistemas.

Como posso aplicar esse aprendizado em situações futuras?

Eu posso expandir o conhecimento adquirido nesse projeto para áreas como testes automatizados, deploys controlados, ambientes de staging, práticas de DevOps e gerenciamento de servidores físicos ou em nuvem, tornando-se essencial para uma atuação profissional na área de infraestrutura ou desenvolvimento web.

Bibliografias utilizadas

Facebook constrói supercomputadores com pilhas gigantes de Mac Minis:

- ✓ <https://www.wired.com/2013/12/apples-mac-mini-reborn/>

Servidor WEB HTTP (Indicado no relatório da Faculdade):

- ✓ <https://www.youtube.com/watch?v=k-AqjnUkec4>