

**ANHANGUERA EDUCACIONAL**  
**SCRN (ASA NORTE) 708/709 - BLOCO B, S/N, ASA NORTE - BRASÍLIA – DF**

**SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE**  
**SISTEMAS**

**AUTOR(ES):**  
**LUCAS LOPES RIBEIRO**

**SISTEMAS DISTRIBUÍDOS**  
**Sincronização relógios através de servidor NTP em sistema operacional Linux**  
**Criação de máquina virtual com sistema operacional GNU/Linux/Debian**  
**Containerização com Docker**  
**Segurança em sistemas distribuídos**

**BRASÍLIA**  
**2025**  
**Lucas Lopes Ribeiro**

## **SISTEMAS DISTRIBUÍDOS**

**Sincronização relógios através de servidor NTP em sistema operacional Linux**

**Criação de máquina virtual com sistema operacional GNU/Linux/Debian**

**Containerização com Docker**

**Segurança em sistemas distribuídos**

Portifólio de Sistemas Distribuídos apresentado ao(s) tutor(es):  
André Martins e Vinicius Mendes Gomes da Silva, do curso de  
Tecnologia em Análise e Desenvolvimento de Sistemas da  
instituição Anhanguera Educacional.

**BRASÍLIA**

**2025**

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>4</b>
<b>2 DESENVOLVIMENTO.....</b>	<b>5</b>
2.1 Sincronização relógios através de servidor NTP.....	5
2.2 Virtualização e Containerização.....	10
2.3 Conteneirização com Docker.....	16
2.4 Segurança em Sistemas Distribuídos.....	19
<b>3 CONCLUSÃO.....</b>	<b>21</b>



## **1 INTRODUÇÃO**

Este trabalho tem como objetivo abordar aspectos fundamentais dos sistemas distribuídos, destacando conceitos técnicos e práticos que garantem a funcionalidade e segurança dessas infraestruturas.

Inicialmente, será discutido o processo de sincronização de relógios utilizando o servidor NTP (Network Time Protocol) em sistemas operacionais Linux, um elemento crítico para manter a precisão temporal e o funcionamento coordenado entre sistemas distribuídos. Em seguida, será abordada a criação de máquinas virtuais utilizando o sistema operacional GNU/Linux/Debian, uma prática relevante para o aprendizado e implementação de ambientes virtuais robustos e configuráveis.

Adicionalmente, o trabalho explorará o conceito de containerização através da plataforma Docker mostrando como orquestrar o servidor web Apache em um clustersimples através do Docker.

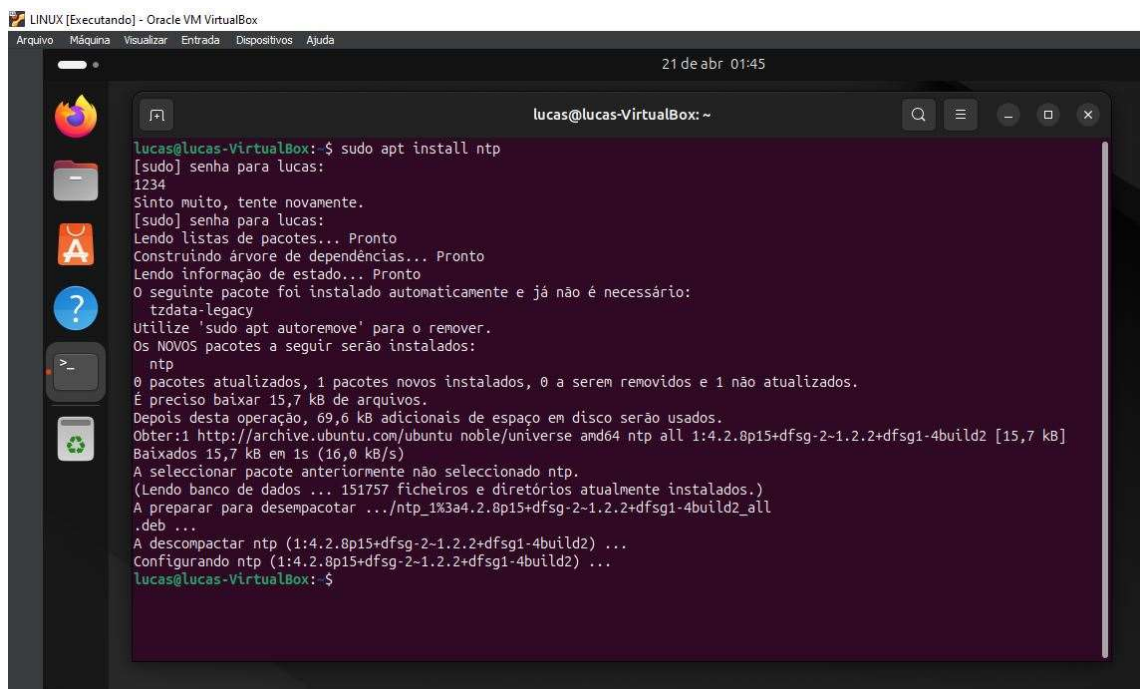
Por fim, será introduzido o tema de segurança em sistemas distribuídos, com foco no uso inicial de analisadores de protocolos, como o Wireshark, para a identificação de vulnerabilidades e monitoramento de redes.

## 2 DESENVOLVIMENTO

### Atividade 1: Sincronização relógios através de servidor NTP

Ao iniciar o trabalho do portfólio, foi instalada a versão mais atual do linux, que apresenta leves diferenças nas respostas do terminal, como por exemplo a atualização do comando **Systemctl** em vez do comando **Service**, visto que o Ubuntu adotou o **systemd** como gerenciador padrão dos serviços. Embora o comando **service** ainda funcione como uma interface mais simples e compatível com versões antigas, ele é apenas um “atalho” e está cada vez menos recomendado em sistemas modernos. Será possível notar sua utilização em breve.

Ao realizar o primeiro comando **sudo apt install ntp** o terminal retornou que o **ntp** tinha sido instalado e sua árvore criada.



```
LINUX [Executando] - Oracle VM VirtualBox
Arquivo  Máquina  Visualizar  Entrada  Dispositivos  Ajuda
21 de abr 01:45
lucas@lucas-VirtualBox: ~
lucas@lucas-VirtualBox:~$ sudo apt install ntp
[sudo] senha para lucas:
1234
Sinto muito, tente novamente.
[sudo] senha para lucas:
Lendo listas de pacotes... Pronto
Construindo árvore de dependências... Pronto
Lendo informação de estado... Pronto
O seguinte pacote foi instalado automaticamente e já não é necessário:
tzdata-legacy
Utilize 'sudo apt autoremove' para o remover.
Os NOVOS pacotes a seguir serão instalados:
ntp
0 pacotes atualizados, 1 pacotes novos instalados, 0 a serem removidos e 1 não atualizados.
É preciso baixar 15,7 kB de arquivos.
Depois desta operação, 69,6 kB adicionais de espaço em disco serão usados.
Obter:1 http://archive.ubuntu.com/ubuntu noble/universe amd64 ntp all 1:4.2.8p15+dfsg-2-1.2.2+dfsg1-4build2 [15,7 kB]
Baixados 15,7 kB em 1s (16,0 kB/s)
A seleccionar pacote anteriormente não seleccionado ntp.
(Lendo banco de dados ... 151757 ficheiros e diretórios atualmente instalados.)
A preparar para desempacotar .../ntp_1%3a4.2.8p15+dfsg-2-1.2.2+dfsg1-4build2_all
.deb ...
A descompactar ntp (1:4.2.8p15+dfsg-2-1.2.2+dfsg1-4build2) ...
Configurando ntp (1:4.2.8p15+dfsg-2-1.2.2+dfsg1-4build2) ...
lucas@lucas-VirtualBox:~$
```

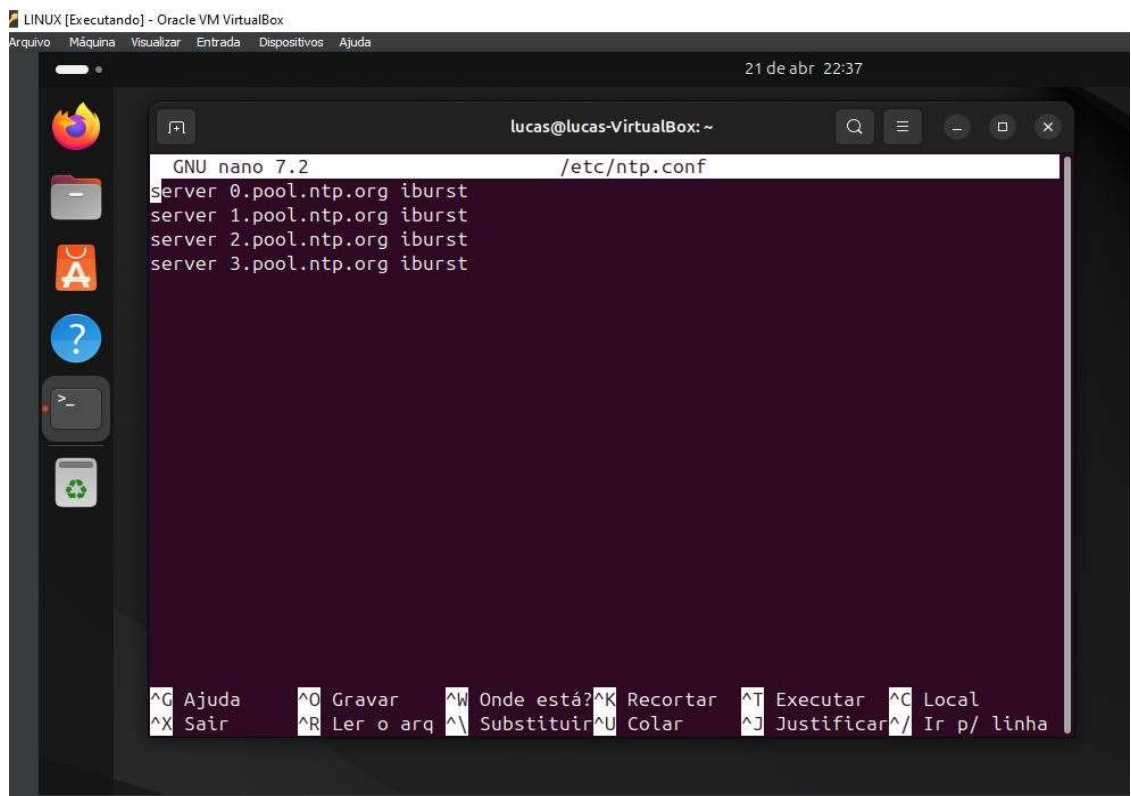
Entretanto, ao utilizar o segundo comando “`ls/etc/ntp.conf`” o terminal não retornou o conjunto de servidores abaixo, então houve a necessidade de escrevê-la manualmente e após isso, sua devida configuração, conforme orientação do tutorial do portfólio.

pool 0.ubuntu.pool.ntp.org iburst

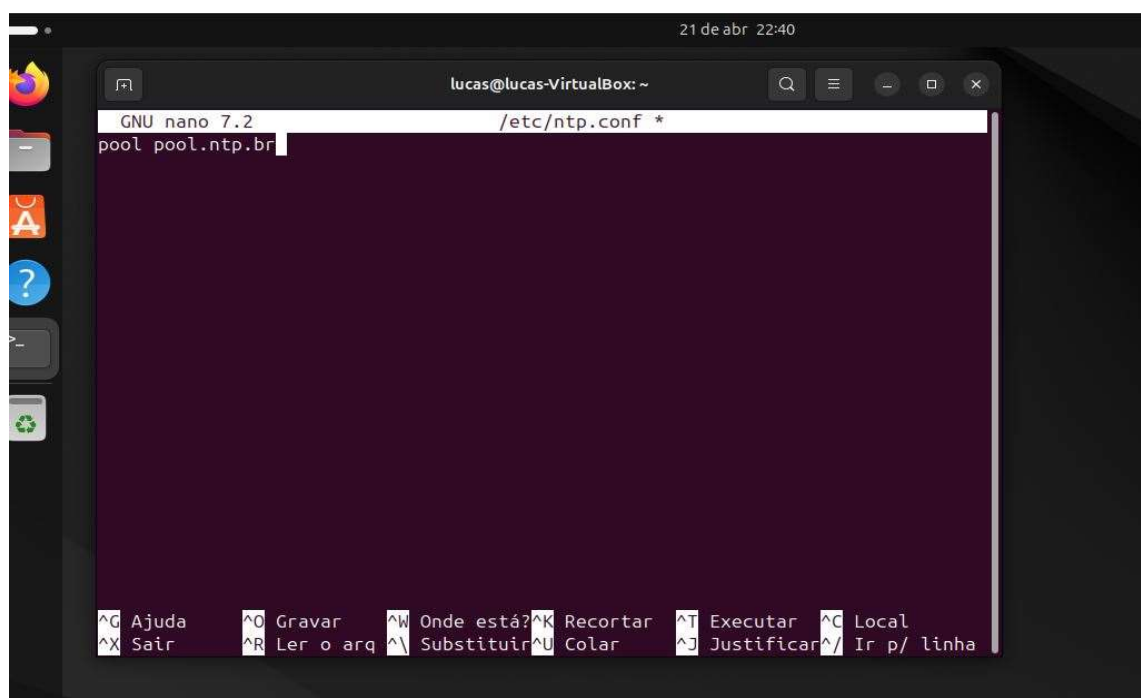
pool 1.ubuntu.pool.ntp.org iburst

pool 2.ubuntu.pool.ntp.org iburst

pool 3.ubuntu.pool.ntp.org iburst



Logo em seguida eu deletei a informação conforme orientação do trabalho e configurei novamente de acordo com a solicitação do portfólio com o comando: `pool.ntp.br`



O próximo passo seria utilizar o comando **"sudo service ntp restart"**. Contudo, utilizando o comando: `sudo systemctl daemon-reload`, as configurações do systemd serão carregadas além de incluir atualizações. Em seguida, devido a versão mais atual do Linux, invés de utilizar o comando: `"sudo service ntp restart"`, será utilizado o comando **"sudo systemctl restart ntpsec"**. Após isto, utilizando o comando **"systemctl status ntp"** o terminal apresenta a informação: **active (running)** indicando que esta tudo funcionando corretamente.



```

LINUX [Executando] - Oracle VM VirtualBox
Arquivo Máquina Visualizar Entrada Dispositivos Ajuda

21 de abr 23:11

lucas@lucas-VirtualBox: ~

lucas@lucas-VirtualBox:~$ sudo systemctl daemon-reload
lucas@lucas-VirtualBox:~$ sudo systemctl restart ntp
lucas@lucas-VirtualBox:~$ systemctl status ntp
● ntpsec.service - Network Time Service
   Loaded: loaded (/usr/lib/systemd/system/ntpsec.service; enabled; preset: e>
   Active: active (running) since Mon 2025-04-21 23:09:14 -03; 1min 35s ago
     Docs: man:ntpd(8)
   Process: 21436 ExecStart=/usr/libexec/ntpsec/ntp-systemd-wrapper (code=exit>
   Main PID: 21440 (ntpd)
     Tasks: 1 (limit: 5316)
    Memory: 10.5M (peak: 11.0M)
       CPU: 84ms
    CGroup: /system.slice/ntpsec.service
            └─21440 /usr/sbin/ntpd -p /run/ntpd.pid -c /etc/ntpsec/ntp.conf -g>

abr 21 23:09:18 lucas-VirtualBox ntpd[21440]: DNS: dns_check: processing 3.ubuntu>
abr 21 23:09:18 lucas-VirtualBox ntpd[21440]: DNS: Pool taking: 200.186.125.195
abr 21 23:09:18 lucas-VirtualBox ntpd[21440]: DNS: Pool taking: 200.160.7.186
abr 21 23:09:18 lucas-VirtualBox ntpd[21440]: DNS: Pool taking: 168.181.126.108
abr 21 23:09:18 lucas-VirtualBox ntpd[21440]: DNS: Pool taking: 162.159.200.1
abr 21 23:09:18 lucas-VirtualBox ntpd[21440]: DNS: dns_take_status: 3.ubuntu.po>
abr 21 23:09:19 lucas-VirtualBox ntpd[21440]: DNS: dns_probe: ntp.ubuntu.com, c>
abr 21 23:09:19 lucas-VirtualBox ntpd[21440]: DNS: dns_check: processing ntp.uba>
abr 21 23:09:19 lucas-VirtualBox ntpd[21440]: DNS: Server taking: 2620:2d:4000:>
abr 21 23:09:19 lucas-VirtualBox ntpd[21440]: DNS: dns_take_status: ntp.ubuntu.>
lines 1-22/22 (END)...skipping...
● ntpsec.service - Network Time Service

```

```

LINUX [Executando] - Oracle VM VirtualBox
Arquivo Máquina Visualizar Entrada Dispositivos Ajuda

21 de abr 23:11

lucas@lucas-VirtualBox: ~

abr 21 23:09:19 lucas-VirtualBox ntpd[21440]: DNS: dns_take_status: ntp.ubuntu.>
lines 1-22/22 (END)...skipping...
● ntpsec.service - Network Time Service
   Loaded: loaded (/usr/lib/systemd/system/ntpsec.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-04-21 23:09:14 -03; 1min 35s ago
     Docs: man:ntpd(8)
   Process: 21436 ExecStart=/usr/libexec/ntpsec/ntp-systemd-wrapper (code=exited, status=0/SUCCESS)
   Main PID: 21440 (ntpd)
     Tasks: 1 (limit: 5316)
    Memory: 10.5M (peak: 11.0M)
       CPU: 84ms
    CGroup: /system.slice/ntpsec.service
            └─21440 /usr/sbin/ntpd -p /run/ntpd.pid -c /etc/ntpsec/ntp.conf -g -N -u ntpsec:ntpsec

abr 21 23:09:18 lucas-VirtualBox ntpd[21440]: DNS: dns_check: processing 3.ubuntu.pool.ntp.org, 8, 101
abr 21 23:09:18 lucas-VirtualBox ntpd[21440]: DNS: Pool taking: 200.186.125.195
abr 21 23:09:18 lucas-VirtualBox ntpd[21440]: DNS: Pool taking: 200.160.7.186
abr 21 23:09:18 lucas-VirtualBox ntpd[21440]: DNS: Pool taking: 168.181.126.108
abr 21 23:09:18 lucas-VirtualBox ntpd[21440]: DNS: Pool taking: 162.159.200.1
abr 21 23:09:18 lucas-VirtualBox ntpd[21440]: DNS: dns_take_status: 3.ubuntu.pool.ntp.org=>good, 8
abr 21 23:09:19 lucas-VirtualBox ntpd[21440]: DNS: dns_probe: ntp.ubuntu.com, cast_flags:1, flags:20801
abr 21 23:09:19 lucas-VirtualBox ntpd[21440]: DNS: dns_check: processing ntp.ubuntu.com, 1, 20801
abr 21 23:09:19 lucas-VirtualBox ntpd[21440]: DNS: Server taking: 2620:2d:4000:1::41
abr 21 23:09:19 lucas-VirtualBox ntpd[21440]: DNS: dns_take_status: ntp.ubuntu.com=>good, 0
~
~
~

```

Concluída esta etapa, o próximo passo do portfólio é utilizar o comando "**ntpq -p**" para que seja possível consultar os servidores NTP manualmente e verificar o status da sincronização.

```

abr 21 23:09:19 lucas-VirtualBox ntpd[21440]: DNS: dns_take_status: ntp.ubuntu.
lines 1-22/22 (END)
lucas@lucas-VirtualBox:~$ ntpq -p
      remote           refid      st t when poll reach  delay  offset  jitter
=====
0.ubuntu.pool.n .POOL.          16 p   - 256    0  0.0000  0.0000  0.0001
1.ubuntu.pool.n .POOL.          16 p   - 256    0  0.0000  0.0000  0.0001
2.ubuntu.pool.n .POOL.          16 p   - 256    0  0.0000  0.0000  0.0001
3.ubuntu.pool.n .POOL.          16 p   - 256    0  0.0000  0.0000  0.0001
#prod-ntp-5.ntp1 79.243.60.50    2 u   33  64   77 276.4782 -52.9364 34.9301
+b.ntp.br        200.160.7.186   2 u   27  64   77 86.9803 -35.6300 16.5291
#lrtest1.ntp.ifs .LRTE.          1 u   26  64   77 83.1435 -46.8098 15.4173
+sa-north-1.clea 76.127.35.142   2 u   24  64   77 42.5333 -34.3523 14.8668
+e.st1.ntp.br    .ONBR.          1 u   23  64   77 42.0075 -34.9823 16.3207
#time100.stupi.s .PPS.           1 u   23  64   77 335.9259 -70.0337 32.5561
+gps.nu.ntp.br   .GPS.           1 u   20  64   77 42.2084 -28.6172 11.8220
+c.ntp.netplanet 200.160.7.193   2 u   23  64   77 66.8734 -48.6106 16.5107
+time.cloudflare 10.42.8.4       3 u   27  64   77 17.9462 -27.8900 15.1944
#a.ntp.netplanet 200.20.186.76   2 u   23  64   77 76.7379 -47.7809 15.2824
+any.time.nl     148.107.165.199 2 u   24  64   77 42.1229 -29.9127 13.9332
*a.st1.ntp.br    .ONBR.          1 u   24  64   77 42.1566 -29.3103 11.1819
+gps.jd.ntp.br   .GPS.           1 u   23  64   77 41.5591 -34.9691 16.5840
+gps.jd.ntp.br   .GPS.           1 u   21  64   77 42.1826 -34.1671 18.0599
#b.ntp.netplanet 200.20.186.76   2 u   24  64   77 77.0196 -47.2819 13.2266
#gru.clearnet.pw 232.14.36.233   3 u   20  64   77 48.1916 -29.6950 12.5505
+a.ntp.br        200.160.7.186   2 u   23  64   77 41.7232 -34.7643 17.9044

```

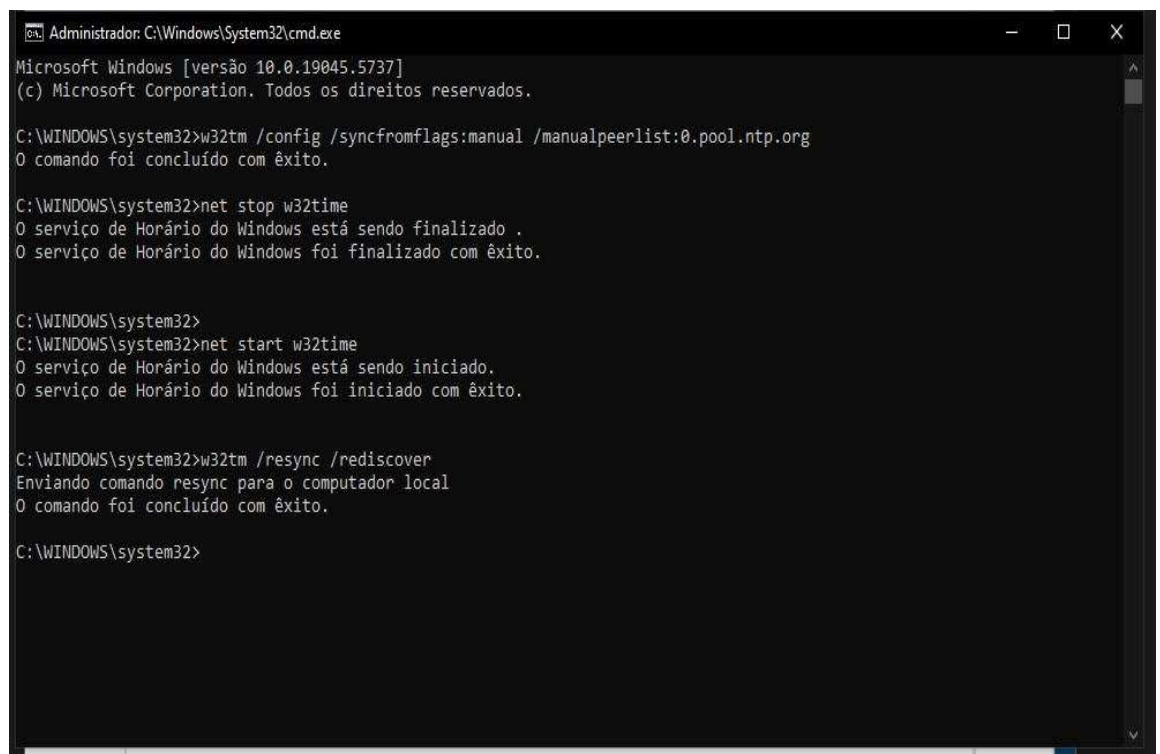
Por fim, abaixo está sendo configurado o sincronismo de data e hora do sistema operacional Windows 10 utilizando o protocolo NTP (Network Time Protocol), CMD do Windows 10. Aqui está uma explicação objetiva do que cada comando realiza:

Comando: **w32tm/config/syncfromflags:manual/manualpeerlist:0.pool.ntp.org**. Esse comando configura manualmente o Windows para sincronizar a data e hora com o servidor 0.pool.ntp.org. Ele aponta o sistema para buscar a referência de tempo nesse servidor NTP confiável.

Comandos: **net stop w32time** e **net start w32time**. Esses comandos reiniciam o serviço de horário do Windows (w32time). A reinicialização aplica as novas configurações feitas no passo anterior.

Comando: **w32tm /resync /rediscover**: Esse comando força uma sincronização imediata da data e hora com o servidor configurado, garantindo que as alterações funcionem corretamente.

Cada comando foi executado com sucesso, e o sistema agora está sincronizado com o servidor NTP especificado, garantindo precisão no relógio do computador.



```
Administrador: C:\Windows\System32\cmd.exe
Microsoft Windows [versão 10.0.19045.5737]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\WINDOWS\system32>w32tm /config /syncfromflags:manual /manualpeerlist:0.pool.ntp.org
O comando foi concluído com êxito.

C:\WINDOWS\system32>net stop w32time
O serviço de Horário do Windows está sendo finalizado .
O serviço de Horário do Windows foi finalizado com êxito.

C:\WINDOWS\system32>
C:\WINDOWS\system32>net start w32time
O serviço de Horário do Windows está sendo iniciado.
O serviço de Horário do Windows foi iniciado com êxito.

C:\WINDOWS\system32>w32tm /resync /rediscover
Enviando comando resync para o computador local
O comando foi concluído com êxito.

C:\WINDOWS\system32>
```

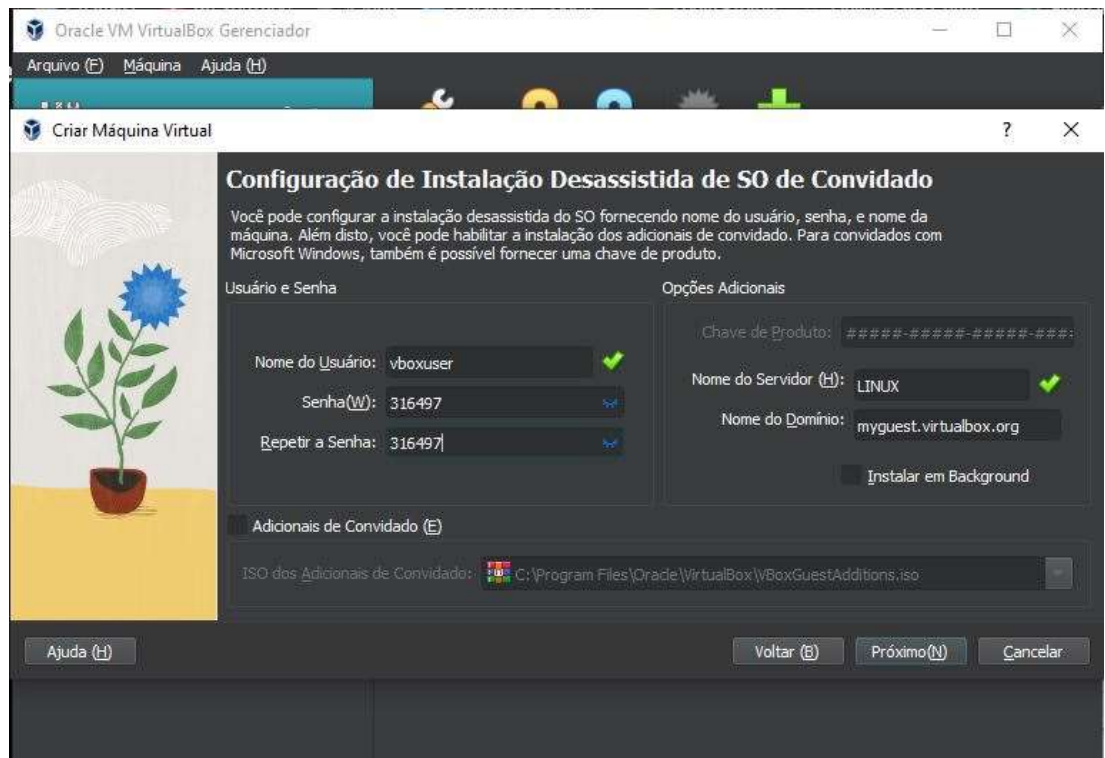
## Atividade 2 - Virtualização e Containerização:

A próxima atividade descrita no documento do portfólio, é a criação de uma máquina virtual no software **ViturtualBox** com sistema operacional GNU/LINUX/DEBIAN.

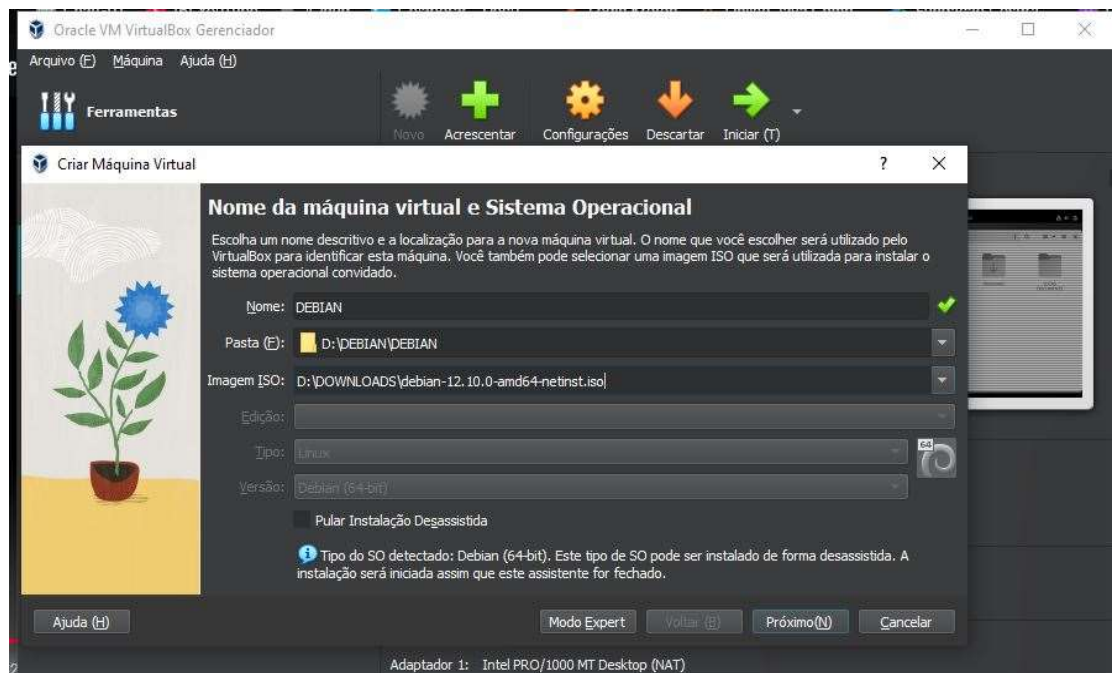
O VituralBox é um software de virtualização de código aberto que permite que os usuários criem máquinas virtuais em seus computadores, simulando sistemas operacionais completos dentro de um ambiente isolado.

Nesta etapa será descrita a criação, configuração da máquina virtual e seleção do arquivo ISO do sistema operacional. Nas imagens a seguir será possível identificar os procedimentos realizados.

### Criação da máquina virtual no software Virtual Box:

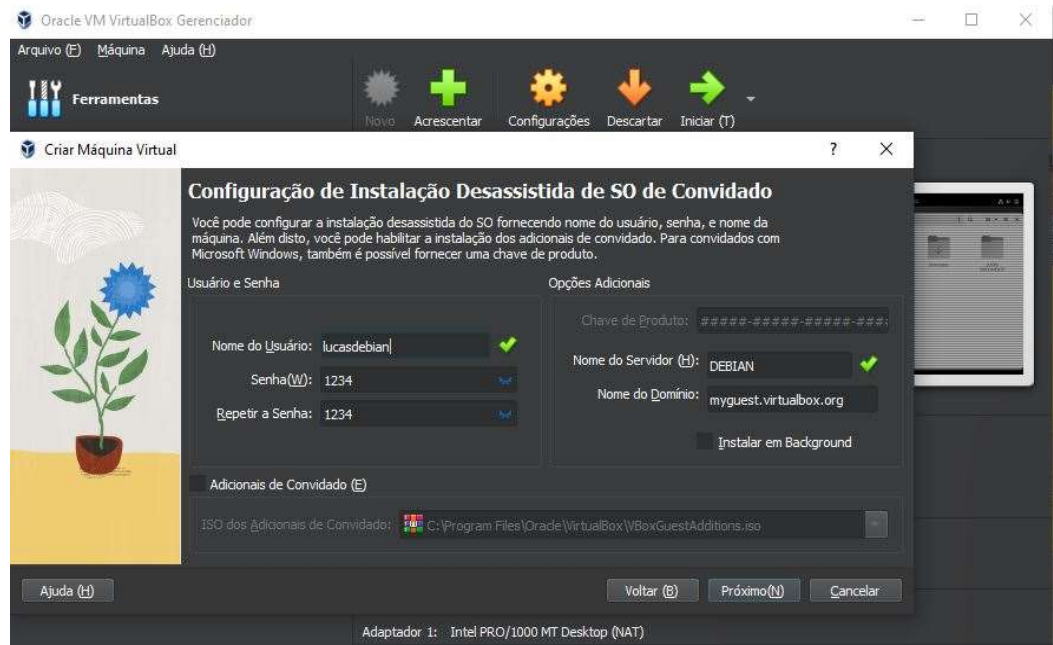


### Definindo nome da máquina virtual e sistema operacional:

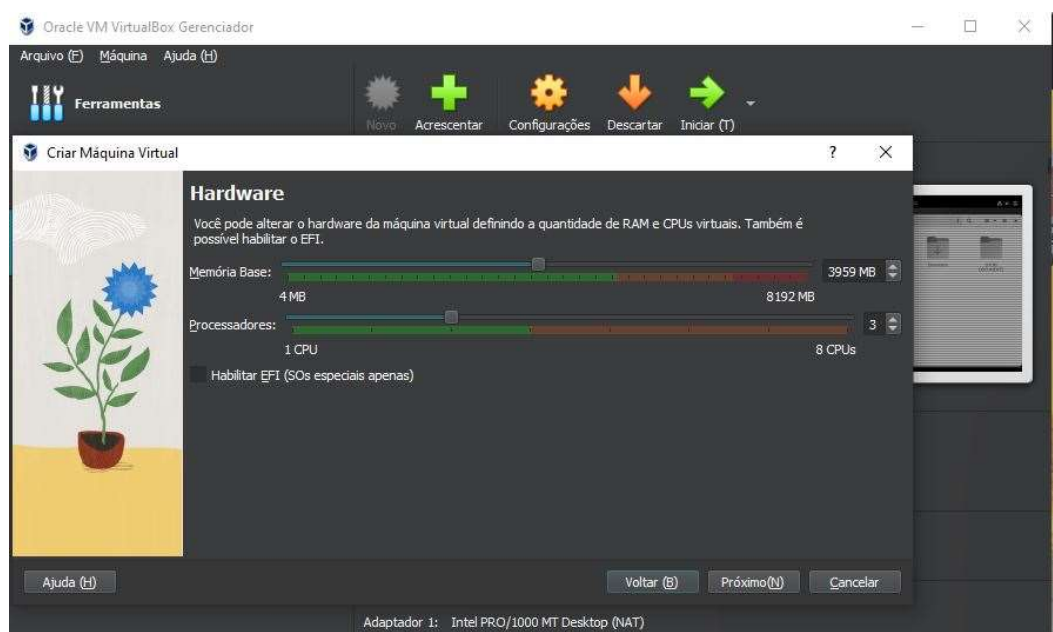


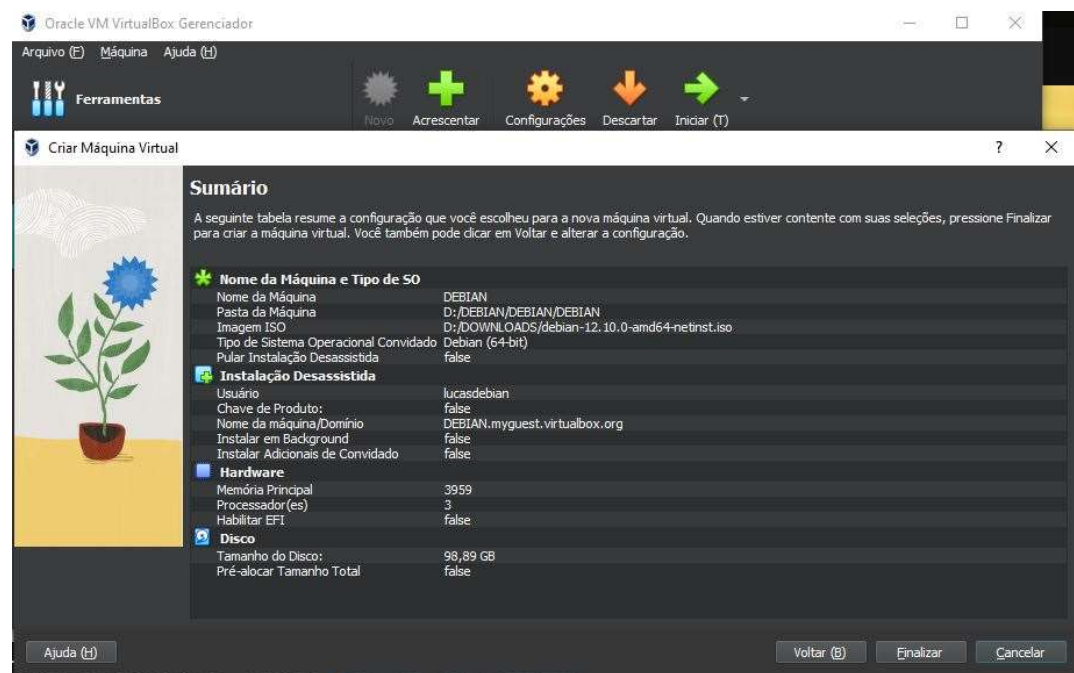
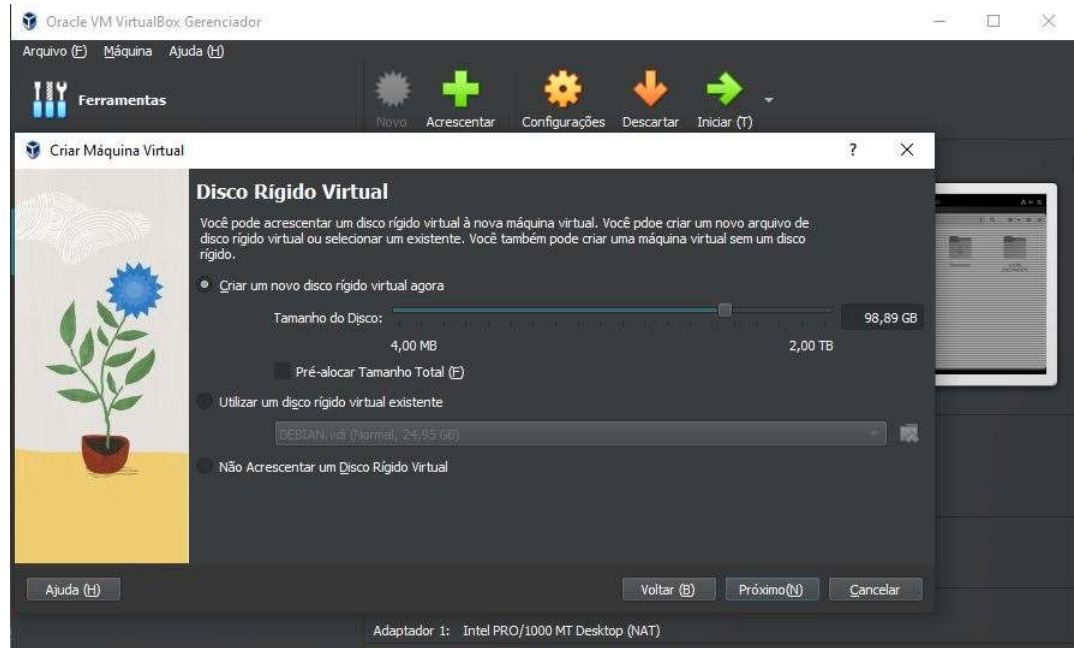


Nesta etapa é possível verificar as opções que definem as informações de endetada no sistema, como nome de usuário, nome do servidor, domínio e senha de acesso:



Configurações de memória da máquina e mais configurações:

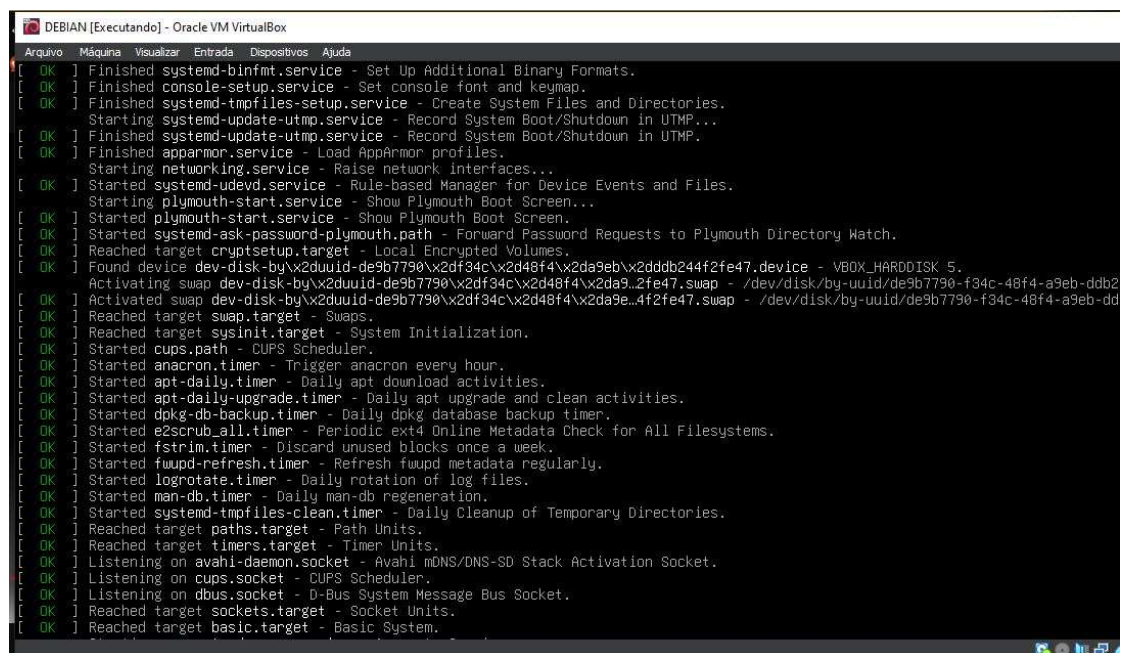




Na próxima imagem é possível ver as máquinas virtuais que existem no perfil do virtual box e suas configurações gerais:



Então, após criar e configurar a máquina com sistema operacional Debian, ela será iniciada, sendo possível ver o código de sua inicialização na imagem a seguir:



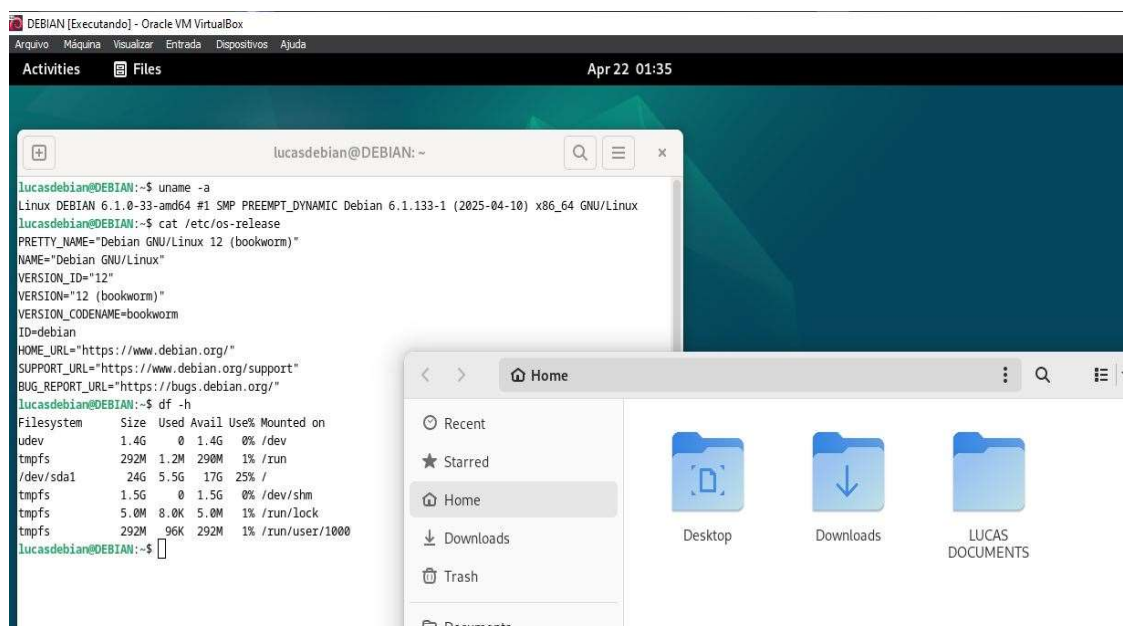
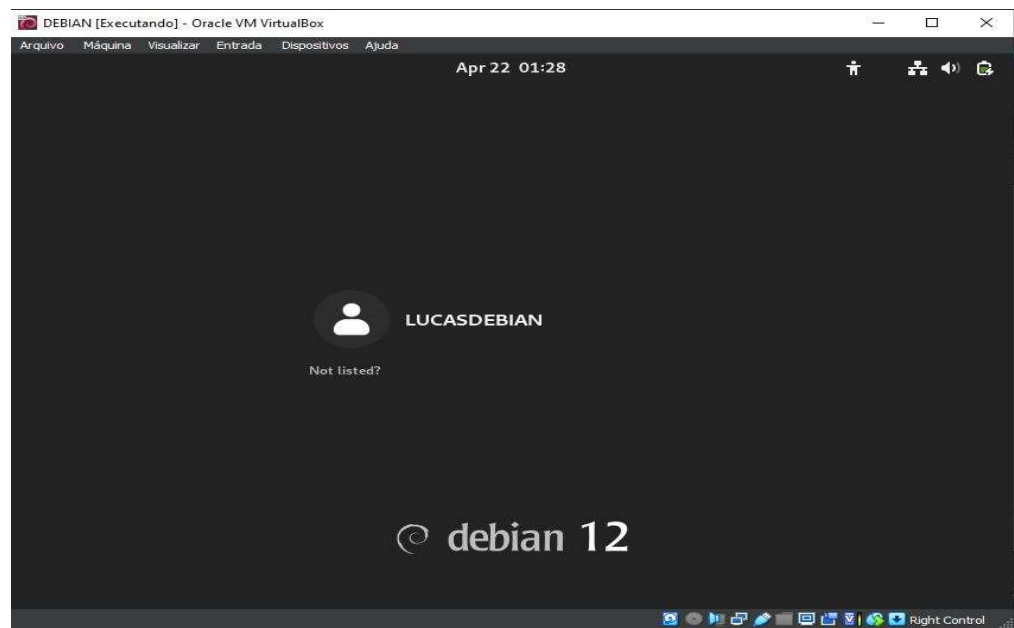
Agora é possível ver a tela de inicialização com o login do usuário e a logo sem seguida, a área de trabalho onde o terminal foi aberto e os comandos, **u-name -a** , **cat /etc/os-release** e **df -h**, afim de demonstrar

detalhes técnicos sobre o sistema.

Comando: **uname -a**, mostra informações detalhadas sobre o sistema, incluindo o kernel, nome do host, arquitetura e até outras especificações.

Comando: **cat /etc/os-release**: Exibe informações sobre a distribuição do sistema operacional, como nome, versão e codinome.

Comando **df -h**: mostra o uso de espaço em disco e das partições montadas, em formato legível para humanos, com KB, MB ou GB.



### Atividade 3 - Containerização com Docker:



A próxima tarefa do trabalho em questão é demonstrar conteneirização trabalhando com o Docker.

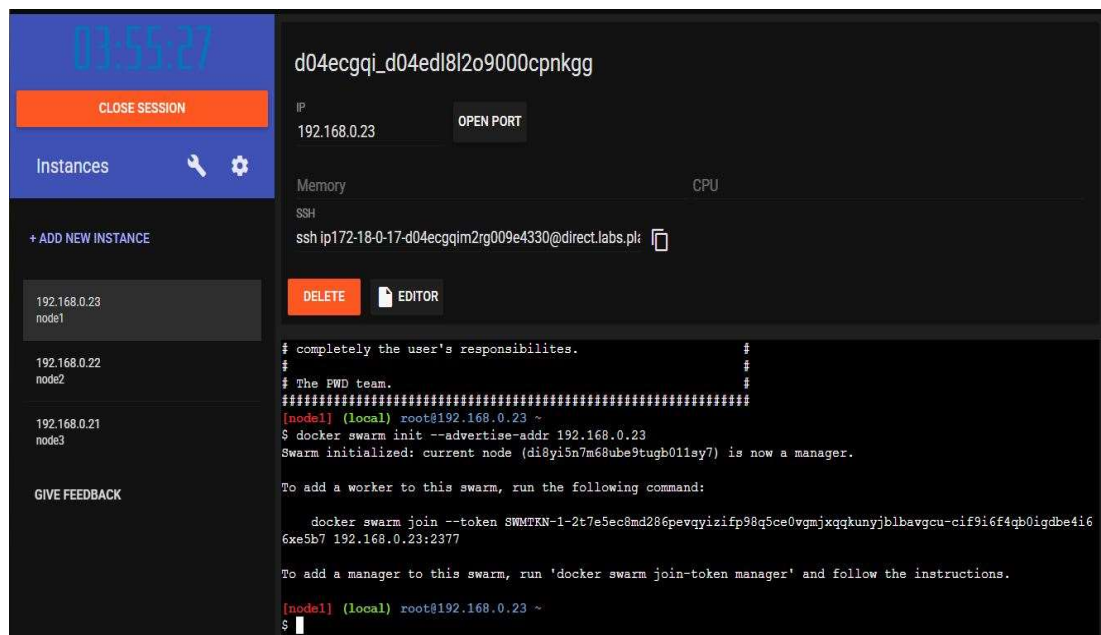
A conteneirização é uma técnica de virtualização leve que isola aplicações e suas dependências em “contêineres”, garantindo sua eficiência em vários ambientes. Por sua vez, o Docker é a plataforma que implementa a conteneirização, permitindo criar, gerenciar e executar os containers.

Depois de acessar o site do serviço Docker e passar pelas etapas de cadastro, será apresentado o ambiente de laboratório:

Seguindo o roteiro do portfólio, primeiro serão criadas três instâncias ou três nós que serão suficientes para analisarmos nosso cluster sem comprometer a usabilidade da plataforma de testes do Docker.

O comando define esse nó como manager do cluster.

**\$ docker swarm init --advertise-addr 192.168.0.23**



Com o resultado da saída, é necessário executar o código nas outras instâncias para "amarrar" os nós do cluster definindo-os como workers. Apresentado na imagem abaixo:

O comando: **docker swarm join --token SWMTKN-1-2t7e5ec8md286pevqyizifp98q5ce0vgmjxqqkunyjb1bavgcucif9i6f4qb0igdb e4i66xe5b7 192.168.0.23:2377**

```

+ ADD NEW INSTANCE

192.168.0.23
node1

192.168.0.22
node2

192.168.0.21
node3

GIVE FEEDBACK

ssh ip172-18-0-19-d04ecgqim2rg009e4330@direct.labs.pl:

[DELETE] [EDITOR]

#####
#                               #
#   WARNING!!!!                 #
#   This is a sandbox environment. Using personal credentials         #
#   is HIGHLY! discouraged. Any consequences of doing so are         #
#   completely the user's responsibilities.                             #
#                               #
#   The FWD team.                                                       #
#####
[node3] (local) root@192.168.0.21 ~
$ docker swarm join --token SWMTKN-1-2t7e5ec8md286pevqyizifp98q5ce0vgmjxqqkunyjblbavgcu-cif9i6f4qb0igdbe4i66xe5b7 192.168.0.23:2377
This node joined a swarm as a worker.
[node3] (local) root@192.168.0.21 ~
$

```

Uma vez que os nós estão criados e seus papéis definidos, o próximo comando será digitado para criar as 5 instâncias do servidor Apache, e logo em seguida outro comando será dado. Que informa em quais nós as 5 réplicas do serviço estão sendo executadas conforme tutorial do portfólio. Os comandos: `docker service create --name WEB --publish 80:80 --replicas=5 httpd` e `docker service ps WEB`.

```

+ ADD NEW INSTANCE

192.168.0.23
node1

192.168.0.22
node2

192.168.0.21
node3

GIVE FEEDBACK

SSH
ssh ip172-18-0-17-d04ecgqim2rg009e4330@direct.labs.pl:

[DELETE] [EDITOR]

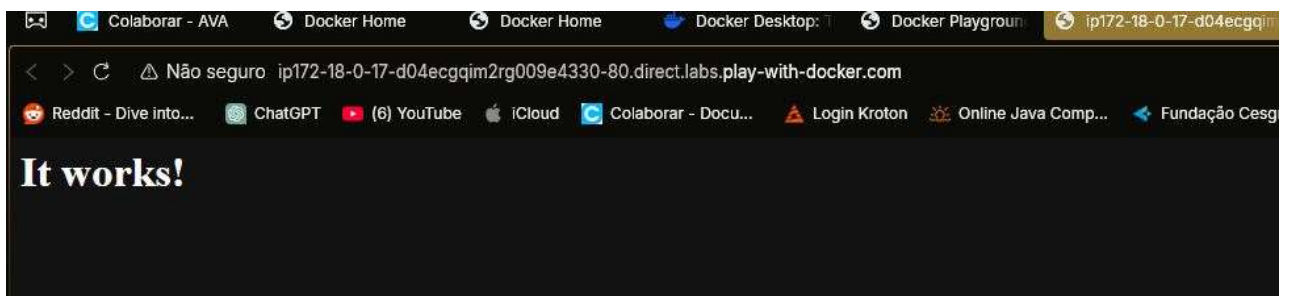
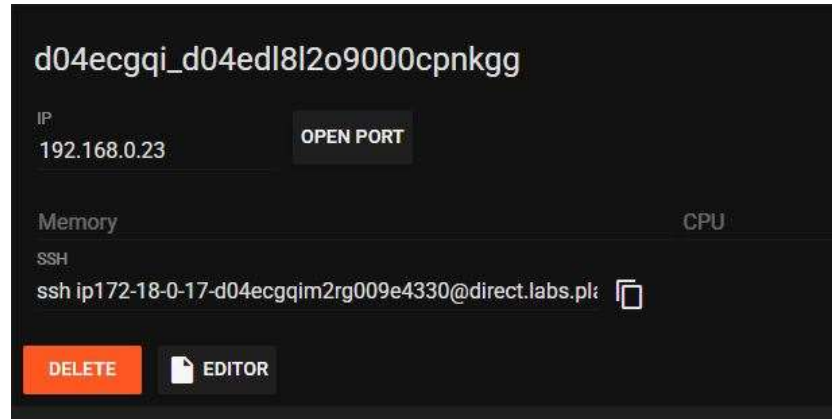
docker swarm join --token SWMTKN-1-2t7e5ec8md286pevqyizifp98q5ce0vgmjxqqkunyjblbavgcu-cif9i6f4qb0igdbe4i66xe5b7 192.168.0.23:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

[node1] (local) root@192.168.0.23 ~
$ docker service create --name WEB --publish 80:80 --replicas=5 httpd
ftphovh36ob5kblflxw7vb7u
overall progress: 5 out of 5 tasks
1/5: running
2/5: running
3/5: running
4/5: running
5/5: running
verify: Service ftphovh36ob5kblflxw7vb7u converged
[node1] (local) root@192.168.0.23 ~
$

```

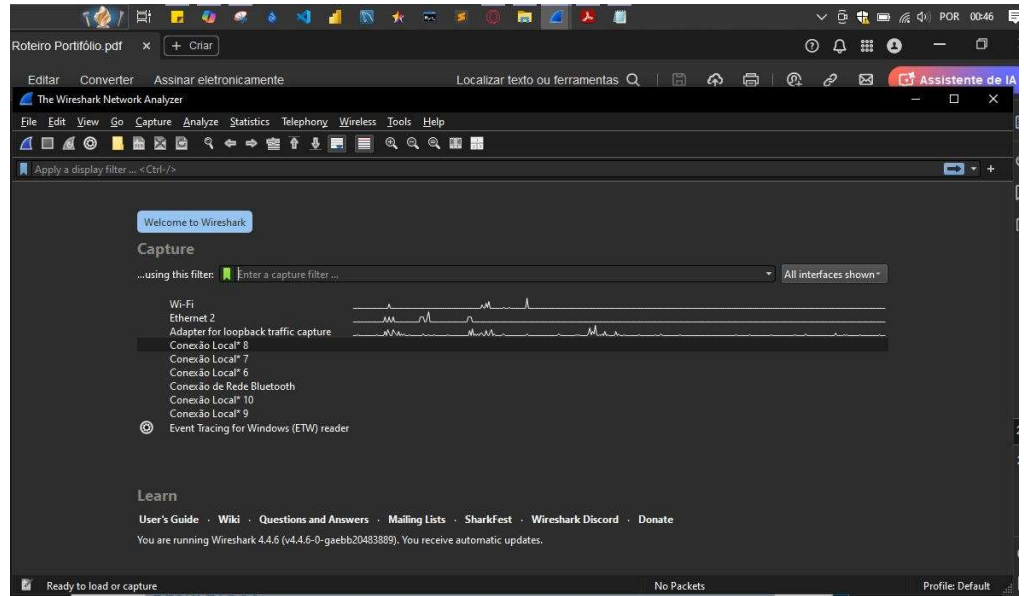
Então é pressionada a opção “Open Port” e uma nova aba será aberta informando que o serviço de configuração do cluster está informando corretamente.



#### Atividade 4 - Aplicações de Sistemas Distribuídos:

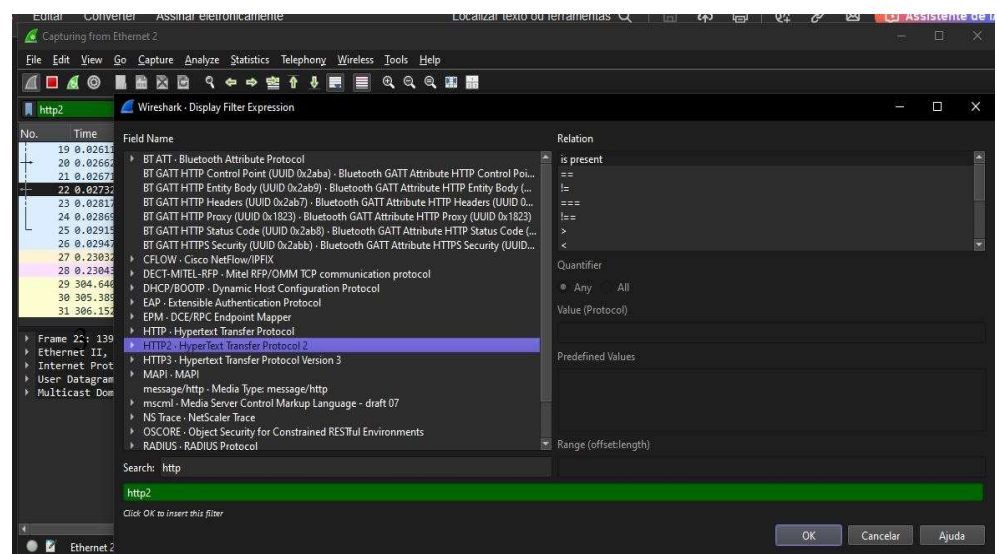
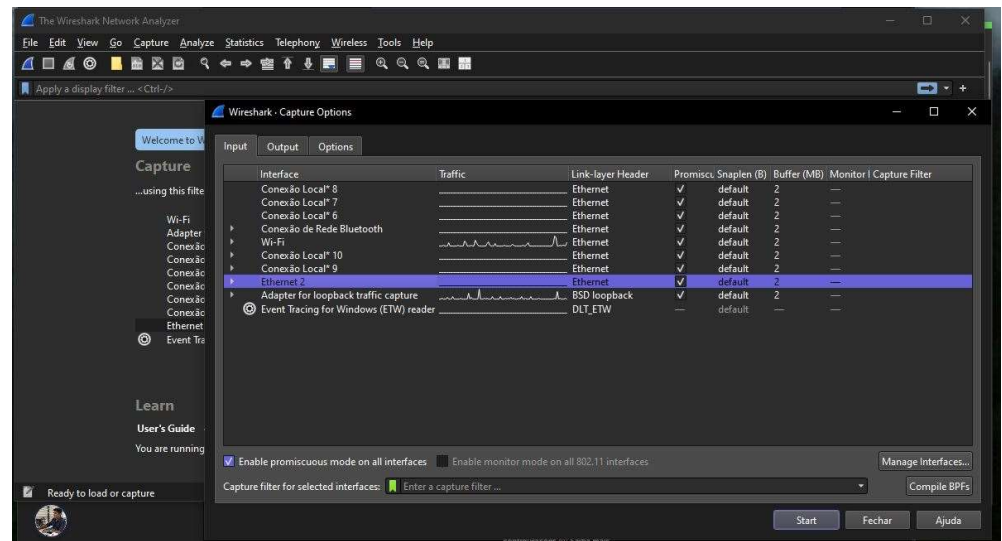
Concluída esta parte, a atividade final do portfólio é compreender como um analisador de pacotes funciona. No caso em questão será utilizado o software WireShark.

Após o download e instalação temos a tela abaixo, que mostra o painel inicial e suas abas. Inclusive, após as primeiras capturas de tráfego de rede o painel apresenta as ultimas redes consultadas:



De acordo com o tutorial do portfólio, é seguida a orientação de ir na aba de opções, selecionar CAPTURE e depois OPTION, será selecionada a interface Ethernet, configurando o software para capturar o tráfego de rede.

Apções capturar o trafego de rede, foi selecionado um filtro de busca "http" que é demosntrado na imagem abaixo entre outros.



### 3 CONCLUSÃO

Neste trabalho, foram realizados procedimentos utilizando sistemas e softwares específicos para explorar diferentes aspectos dos sistemas distribuídos. No sistema operacional Linux, configurou-se a sincronização de relógios através do servidor NTP utilizando comandos no terminal.

A criação de uma máquina virtual foi feita no software VirtualBox, usando o sistema GNU/Linux Debian como base para o ambiente virtual. Além disso, a plataforma Docker foi empregada para implementar a containerização, demonstrando seu uso prático no gerenciamento de aplicativos em contêineres.

Por fim, utilizou-se o Wireshark como ferramenta para análise de protocolos de rede, destacando sua aplicação na identificação e monitoramento de tráfego. Esses passos foram conduzidos com o objetivo de compreender e aplicar conceitos fundamentais dos sistemas distribuídos.