



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Instituto de Ciências Exatas e de Informática

Apresentação de projeto Solo Forte Agropecuária*

Bruno Alfeu Mendes de Araújo¹
Gabriel Amâncio de Oliveira¹
Guilherme de Souza Mendonça Silva¹
Isaac¹
Matheus Godinho Blaselbauer¹
Yan Guimarães Martins¹

Fabio Leandro Rodrigues Cordeiro²

Resumo

Este trabalho insere-se no contexto de planejamento e implementação de infraestrutura de redes de computadores para a empresa fictícia de médio porte Solo Forte Agropecuária, seguindo um fluxo desde o levantamento dos requisitos aos testes de conexão, para que as filiais e a sede da empresa se comuniquem corretamente e isto resolva as dores de problemas de conexão da empresa. Para isso são utilizados os serviços de rede HTTP, FTP, DNS, NFS, DHCP, Banco de Dados com PostgreSQL, AD(Active Directory) + GPO(Group Policy Object). A justificativa para tal implementação reside na necessidade de uma infraestrutura de TI robusta que suporte a expansão da empresa, otimize a logística e centralize a gestão de dados. O objetivo deste projeto é prototipar e configurar os serviços de rede fundamentais para a operação da Solo Forte Agropecuária, conhecendo os ambientes de nuvem(cloud) e físicos(on-premise) para a hospedagem dos serviços citados.. Como resultado, obteve-se a configuração funcional de todos os serviços propostos. Conclui-se que a arquitetura de rede híbrida proposta é viável e atende aos requisitos operacionais iniciais da empresa, demonstrando a correta implementação dos serviços de rede essenciais.

Palavras-chave: infraestrutura de redes; serviços de rede; DHCP; ambiente híbrido; virtualização.

*Artigo apresentado ao Instituto de Ciências Exatas e Informática da Pontifícia Universidade Católica de Minas

Gerais, campus Contagem, como pré-requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

¹Alunos do Programa de Graduação em Sistemas de Informação – xxx@sga.pucminas.br.

²Professor do Programa de Graduação em Sistemas de Informação – xxx@pucminas.br.

1 ETAPA 1- PROTOTIPAGEM E PLANEJAMENTO

Muitas empresas possuem problemas para pensar em como desenvolver uma boa arquitetura de rede e pensar nas melhores escolhas de acordo com suas necessidades, como por exemplo o número de dispositivos necessários para estruturar a rede, a topologia que virá a ser utilizada de acordo com suas necessidades, e como permitir uma comunicação eficiente de forma que não haja nenhum defeito crítico na rede. Todo o plano para se estruturar uma rede precisa ser bem montado e escalável de acordo com as principais necessidades empresariais, desde o levantamento dos requisitos ao teste e execução das conexões. Por isso, é essencial escolher uma boa metodologia a seguir para definir esses parâmetros e permitir que o produto final seja fiel à ideia da empresa.

Portanto, uma abordagem bem consolidada no contexto de redes de computadores é a metodologia top-down, em que as decisões começam de decisões mais genéricas e migram para questões mais específicas. No contexto de arquitetura de redes, primeiro são levantados os requisitos gerais para se desenvolver a rede e posteriormente a arquitetura e estrutura lógica e física da rede são projetadas. Para se desenvolver a etapa 1 esta foi a metodologia adotada.

Assim sendo, durante a etapa 1 foi utilizado um software de simulação de redes, o Cisco Packet Tracer, fornecido pela Cisco, para estruturar a rede de acordo com uma topologia em conjunto com planilhas para enumerar a quantidade, em média, de equipamentos por categoria, e os gastos de acordo com a conectividade entre os mesmos, como a velocidade de transferência entre um dispositivo e outro, por exemplo.

Para definir estes parâmetros, foram seguidas quatro etapas de acordo com a definição da metodologia top-down:

1. Identificação das necessidades e objetivos da rede: Uma empresa de agricultura com porte de pequeno a médio com duas filiais e uma sede, com o objetivo de montar uma rede bem otimizada com o menor desperdício de gastos possível e disponibilidade de alcance para regiões mais afastadas.
2. Projeto lógico: Escolha da topologia de barramento, em que todos os computadores estão conectados por um único cabo, para estruturar a relação entre os roteadores. Mapeamento das políticas de roteamento com otimização para se obter o melhor custo-benefício na ligação por barramento. Estruturação de um diagrama de cabeamento com direcionamento dos custos estimados. Mapeamento dos endereços IPv4 das máquinas e hierarquia de distribuição dos mesmos. Interconexão entre as redes locais para permitir o roteamento externo, utilizando cabos DTE/DCE.
3. Projeto físico: Organização do cabeamento no rack. Correlação e ordenação física das portas para conexão entre a matriz e as filiais. Distribuição entre os computadores por

unidade e os tipos de conexão seguindo a relação matriz-filial.

2 ETAPA 2- CONFIGURAÇÃO DOS SERVIÇOS DE REDE EM NUVEM E ON-PREMISE

Muitas empresas recentemente criadas enfrentam dificuldades para disponibilizar e gerenciar servidores físicos próprios em suas instalações. O custo para comprar computadores completos geralmente é muito elevado e o gerenciamento de toda a arquitetura e disponibilidade dos serviços do zero geralmente demanda um elevado conhecimento técnico e cautela, sendo assim a empresa responsável por gerenciar toda a infraestrutura. Portanto esse setup pode não ser a opção mais rentável para startups. Daí nasce o conceito de computação em nuvem, em que provedores terceirizados oferecem serviços virtuais hospedados em servidores remotos previamente configurados e gerenciados em troca de uma mensalidade em que o usuário paga geralmente conforme demanda de uso. Ou seja, o cliente pode escalar todo o setup e infraestrutura de seus serviços conforme necessário e já utiliza uma base de infraestrutura pré-pronta, recurso usualmente chamado de IaaS ou Infraestrutura como Serviço, sendo a AWS um ótimo exemplo.

Com isso, nessa etapa os serviços de rede exigidos anteriormente na etapa 1 foram configurados, cada um, ou em ambiente de nuvem utilizando a arquitetura da AWS, ou on-premise através do software VirtualBox. Para isso, inicialmente cada aluno conheceu os conceitos básicos de configuração de cada ambiente e posteriormente os serviços de rede foram divididos entre os alunos de acordo com as seções especificadas a seguir:

2.1 Ambiente Cloud

2.1.1 Protocolo HTTP

O protocolo de Transferência de Hipertexto(HTTP), é o alicerce da comunicação de dados na web. O HTTP é um protocolo de camada de aplicação, onde opera no modelo cliente-servidor, normalmente o cliente(que geralmente é um navegador) inicia a troca através de uma requisição HTTP e o servidor responde com uma Resposta HTTP. Os responsáveis pela comunicação entre o cliente e servidor são os protocolos de comunicação(como HTTP e o TCP/IP) que define as regras para a troca de dados e a própria infraestrutura de rede(roteadores, switches e cabos) que fisicamente conecta o cliente e o servidor. Seus principais métodos de requisições: GET, POST, PATCH, PUT e DELETE. Existem também os códigos de status, os mais comuns são: 200, 400, 403, 404 e 500. E existem os cabeçalhos, que são os metadados que são enviados entre cliente-servidor.

Neste passo a passo da configuração do HTTP, precisamos ter um servidor Ubuntu como um dos pré-requisitos. Nesse exemplo de configuração vamos instalar o Apache e atualizar o firewall. Vamos executar o primeiro comando, a atualização do gerenciador de pacotes do nosso sistema Ubuntu:

Figura 1 – Atualização do cache do gerenciador de pacotes

```
ubuntu@ip-172-31-20-184:~$ sudo apt-get update|
```

Após a atualização, vamos instalar o Apache com o seguinte comando:

Figura 2 – Instalação do Apache

```
ubuntu@ip-172-31-20-184:~$ sudo apt-get install apache2|
```

Confirme a instalação pressionando "Y", logo após "ENTER". Agora precisamos ajustar as configurações do firewall executando os seguintes comandos:

Figura 3 – Instalação do Apache(2)

```
ubuntu@ip-172-31-20-184:~$ sudo ufw app list sudo ufw allow in "Apache"|
```

Agora, logo após todos os comandos executados, deveremos visualizar com o IP do servidor(exemplo <http://12.345.678.91/>) no navegador de sua preferência, você deve encontrar a página padrão do Apache:

Figura 4 – Página do Apache



Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```

/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf

```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`

Agora, com tudo configurado, vamos entrar nas pastas “`var/www/html`” para poder criar uma nova pasta e um novo arquivo dentro da mesma, com isso devemos entrar na pasta mencionada.

Figura 5 – comando para entrar na pasta var

```
ubuntu@ip-172-31-20-184:/$ cd /var/www/html
```

Vamos criar uma pasta com o nome de sua escolha:

Figura 6 – comando para criar pasta

```
ubuntu@ip-172-31-20-184:/var/www/html$ sudo mkdir meuSite
```

Com a pasta criada agora vamos entrar na pasta com o comando “cd (nome da sua pasta criada) e vamos criar um arquivo .html para realizar a criação simples de um site e visualizá-lo na nosso ip do servidor:

Figura 7 – comando para criar um arquivo

```
ubuntu@ip-172-31-20-184:/var/www/html$ sudo touch index.html
```

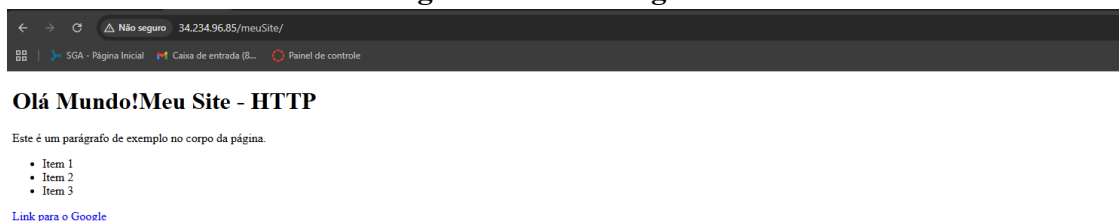
Agora com o arquivo criado vamos editá-lo, para conseguir visualizá-lo com o ip do servidor. Antes temos que realizar esse comando para entrar no arquivo .html e configurar com seu código de sua escolha.

Figura 8 – comando para entrar na pasta e comando para editar seu arquivo .html

```
ubuntu@ip-172-31-20-184:/var/www/html$ cd meuSite
ubuntu@ip-172-31-20-184:/var/www/html/meuSite$ sudo nano index.html
```

Com tudo configurado, acesse seu ip na seu navegador e acesse a pasta escolhida e você irá conseguir visualizar seu site.

Figura 9 –site configurado



2.1.2 Protocolo FTP

O protocolo FTP(File Transfer Protocol), traduzido para português como Protocolo de Transferência de Arquivo, permite a transferência de arquivos em um modelo cliente- servidor através de uma conexão padrão web padrão utilizando TCP nas portas 20 a 21. O princípio do funcionamento do FTP ocorre por dois canais de comunicação: o canal de comando e o canal de dados. O canal de comandos é responsável por lidar com as instruções de execução e o que retornam, enquanto o canal de dados se responsabiliza pelo transporte dos arquivos enviados e/ou recebidos. Geralmente a porta 21 é utilizada para o canal de comando e a 20 para dados.

Para iniciar a configuração do FTP, primeiramente é necessário instalar o pacote "vsftpd"no servidor remoto.

Figura 1 – Instalação do Pacote vsftpd no servidor remoto

```
tu@ip-172-31-48-141:~$ sudo apt install vsftpd
```

Após a instalação do pacote é necessário executar os comandos "sudo service vsftpd restart"e posteriormente "sudo service vsftpd status"para se certificar de que o serviço esteja rodando corretamente.

Figura 2 – Início e Checagem de status do serviço vsftpd

```

tu@ip-172-31-48-141: $ sudo service vsftpd restart
tu@ip-172-31-48-141: $ sudo service vsftpd status
ftpd.service - vsftpd FTP server
Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: enabled)
Active: active (running) since Sat 2025-10-18 00:33:50 UTC; 4s ago
Process: 1417 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty (code=exited, status=0/SUCCESS)
Main PID: 1420 (vsftpd)
Tasks: 1 (limit: 1121)
Memory: 804.0K (peak: 1.1M)
CPU: 7ms
CGroup: /system.slice/vsftpd.service
        └─1420 /usr/sbin/vsftpd /etc/vsftpd.conf

18 00:33:50 ip-172-31-48-141 systemd[1]: Starting vsftpd.service - vsftpd FTP server...
18 00:33:50 ip-172-31-48-141 systemd[1]: Started vsftpd.service - vsftpd FTP server.

```

Fonte:(Góes et al., 2005)

Uma conexão pelo protocolo FTP geralmente está associada à porta 20 ou 21. Entretanto esta conexão é insegura e não criptografada, ou seja, os dados de usuário e senha da máquina cliente são expostos em texto plano e as informações tornam-se visíveis. Por isso é importante optar pelo uso do protocolo SFTP(Secure File Transfer Protocol), que possui as mesmas funcionalidades do FTP mas com conexões seguras e criptografadas, assegurando-se de que esses dados sejam disponibilizados no formato de um código humanamente ilegível.

O próximo passo é configurar o firewall no terminal e no console da AWS. Primeiramente será efetuada a configuração no terminal: Para ter certeza de que o firewall está habilitado é necessário executar o comando "sudo ufw status".

Figura 3 – Checagem do status do firewall no terminal com o firewall inativo

```

ip-172-31-48-141:~$ sudo ufw status
inactive

```

Fonte:(Gropp, 2003)

Geralmente, o firewall pelo terminal vai estar desabilitado. Portanto, para habilitá-lo rodar o comando "sudo ufw enable".

Figura 4 – Ativação do firewall no terminal

```

ubuntu@ip-172-31-48-141:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y/n)? y
Firewall is active and enabled on system startup

```

Fonte:(Góes et al., 2005)

Uma vez habilitado, é possível rodar novamente "sudo ufw status" para verificar as configurações de tráfego adicionadas.

Figura 5 – Checagem do status do firewall no terminal

```
ubuntu@ip-172-31-48-141:~$ sudo ufw status
Status: active

To Action From
--
20:21/tcp ALLOW Anywhere
12000:12100/tcp ALLOW Anywhere
22/tcp ALLOW Anywhere
Apache ALLOW Anywhere
80/tcp ALLOW Anywhere
443 ALLOW Anywhere
990/tcp ALLOW Anywhere
30000:31000/tcp ALLOW Anywhere
30000:31000/tcp ALLOW Anywhere
5432/tcp ALLOW Anywhere
20:21/tcp (v6) ALLOW Anywhere (v6)
12000:12100/tcp (v6) ALLOW Anywhere (v6)
22/tcp (v6) ALLOW Anywhere (v6)
Apache (v6) ALLOW Anywhere (v6)
80/tcp (v6) ALLOW Anywhere (v6)
443 (v6) ALLOW Anywhere (v6)
990/tcp (v6) ALLOW Anywhere (v6)
30000:31000/tcp (v6) ALLOW Anywhere (v6)
5432/tcp (v6) ALLOW Anywhere (v6)
```

Fonte:(Góes et al., 2005)

Se não houver nenhuma configuração anterior de controle de tráfego, a lista geralmente estará vazia. Para o gerenciamento do controle de tráfego dos protocolos é possível utilizar o nome do protocolo ou a porta que utiliza conjuntamente ao(s) protocolo(s) de comunicação. Para permitir o tráfego FTP adicionar o intervalo de portas 20 a 21, portanto rodando o comando "sudo ufw allow 20:21/tcp", uma vez que para o caso apenas o protocolo de comunicação TCP basta.

Figura 6 – Permissão do tráfego para conexões FTP no terminal

```
ubuntu@ip-172-31-48-141:~$ sudo ufw allow 20:21/tcp
adding existing rule
adding existing rule (v6)
```

Fonte:(Góes et al., 2005)

Como no caso o tráfego já existe, o terminal retornou um output informando que a regra de tráfego já foi adicionada anteriormente.

Também será necessário adicionar o intervalo de portas 12000 a 12100 através do protocolo TCP. Esse intervalo serve para estabelecer as portas de conexão na parte da máquina cliente, visto que no modo ativo portas aleatórias serão escolhidas para a conexão da máquina cliente em uma porta local antes de se conectar à porta do servidor. Com isso, será rodado o comando "sudo ufw allow 12000:12100/tcp".

Figura 7 – Permissão do tráfego para conexões FTP no modo passivo no terminal

```
ubuntu@ip-172-31-48-141:~$ sudo ufw allow 12000:12100/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
ubuntu@ip-172-31-48-141:~$
```

Fonte:(Góes et al., 2005)

Uma vez com o firewall configurado no terminal, o mesmo procedimento precisará ser feito no grupo de segurança da instância EC2 pelo console da AWS.

Figura 8 – Permissão do tráfego para FTP na AWS

| Name | Security group rule... | IP version | Type | Protocol | Port range | Source | Description |
|-----------------------|------------------------|------------|------------|----------|---------------|-----------------|--------------|
| sg-01a8d01e69b49469 | SSH | IPv4 | SSH | TCP | 22 | 177.62.78.77/32 | SSH |
| sg-0708d8196d5866... | Custom TCP | IPv4 | Custom TCP | TCP | 20 - 21 | 0.0.0.0/0 | FTP |
| sg-0806b8a50a081a877 | PostgresQL | IPv4 | PostgresQL | TCP | 5432 | 0.0.0.0/0 | |
| sg-02433e088655556 | HTTP | IPv4 | HTTP | TCP | 80 | 0.0.0.0/0 | HTTP |
| sg-0c958d8c54e1405... | Custom TCP | IPv4 | Custom TCP | TCP | 12000 - 12100 | 0.0.0.0/0 | |
| sg-0e1474ef1cb0846be | Custom TCP | IPv4 | Custom TCP | TCP | 990 | 0.0.0.0/0 | TLS over FTP |
| sg-0f318942c5a41a082 | Custom TCP | IPv4 | Custom TCP | TCP | 30000 - 31000 | 0.0.0.0/0 | |
| sg-052d3e3c9c479369 | HTTPS | IPv4 | HTTPS | TCP | 443 | 0.0.0.0/0 | HTTPS |

Fonte:(Góes et al., 2005)

Após isso será necessário adicionar as informações do usuário que poderá efetuar as conexões FTP com o servidor. Para isso rodar "sudo useradd [nome]"e posteriormente "sudo passwd [nome]", substituindo o campo "[nome]"pelo nome escolhido. No caso o usuário escolhido será "random"então os comandos "sudo useradd random"e "sudo passwd random"serão rodados. Ao rodar "sudo passwd"para o respectivo usuário inserir uma senha de escolha e confirmar.

Figura 9 – Configuração do usuário que irá utilizar conexões FTP com o servidor

```
ubuntu@ip-172-31-48-141:~$ sudo useradd matheus
ubuntu@ip-172-31-48-141:~$ sudo passwd matheus
New password:
Retype new password:
passwd: password updated successfully
```

Fonte:(Góes et al., 2005)

Como o arquivo de configuração gerado pela instalação do vsftpd terá que ser editado, para evitar perdas é importante criar uma configuração de backup, com os dados iniciais. Para isso, será necessário rodar "sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.bkp", que copiará o arquivo original para uma réplica original.

Figura 10 – Cópia do arquivo de configuração vsftpd.conf

```
buntu@ip-172-31-48-141:~$ sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.bkp
buntu@ip-172-31-48-141:~$
```

Fonte:(Góes et al., 2005)

Para editar o arquivo vsftpd.conf será necessário utilizar um editor de, como vim ou nano. No caso será utilizado nano, portanto rodar "sudo nano /etc/vsftpd.conf" para abrir o editor de texto usando GNU nano.

Figura 11 – Comando para edição do arquivo de configuração vsftpd

```
ubuntu@ip-172-31-48-141:~$ sudo nano /etc/vsftpd.conf
```

Fonte:(Góes et al., 2005)

Figura 12 – Tela do arquivo de configuração vsftpd

```
Example config file /etc/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Run standalone? vsftpd can run either from an inetd or as a standalone
# daemon started from an init script.
listen=YES
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (:::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on vstps IPv6 and IPv4
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=NO
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Userlist deny=NO
userlist_deny=NO
userlist_file=/etc/vsftpd/user_list
#
# Path to config file
path=/etc/vsftpd
#
# Local root=/home/random/dados
local_root=/home/random/dados
#
# Local user=YES
local_user=YES
#
# Allow writeable chroot=YES
allow_writeable_chroot=YES
#
# Local enable=YES
local_enable=YES
#
# Write enable=YES
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftp'd's)
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
```

Fonte:(Góes et al., 2005)

No caso o arquivo já está com as configurações corretas, mas para adicioná-las precisará inserir o seguinte trecho no arquivo:

"userlist_deny=NO userlist_file=/etc/vsftpd/user_list tcp_wrappers=NO write_enable=YES local_root=/home/[nome]/dados chroot_local_user=YES allow_writeable_chroot=YES ". Substitua [nome] pelo nome do usuário escolhido, no caso "/home/random/dados".

Estas configurações permitem gerenciar corretamente os usuários adicionados à lista de usuários para acessar o vsftpd, fornecem permissão de escrita e estabelecem o caminho da pasta de dados para ser gerenciada pelo dado usuário, no caso dentro do diretório /home da

máquina virtual.

Será necessário executar o comando "sudo mkdir -p /etc/vsftpd" para criar a pasta e posteriormente "tp" para adicionar o arquivo de configuração dos

usuários permitidos e o inserir o nome do usuário escolhido, como especificado no caminho fornecido ao arquivo vsftpd.conf. A seguir executar "sudo mkdir -p /home/[nome]/dados" e "sudo chown [nome] /home/[nome]/dados" para que o usuário determinado em "[nome]" tenha acesso a escrita e leitura em /home/[nome]/dados, substituindo "[nome]" pelo nome do usuário. No caso, os comandos "sudo mkdir -p /home/random/dados" e "sudo chown random /home/random/dados".

Figura 13 – Comandos para criar o caminho para o arquivo de lista de usuários

```
31-48-141:~$ sudo mkdir -p /etc/vsftpd
31-48-141:~$ sudo nano /etc/vsftpd/user_list
31-48-141:~$
```

Fonte:(Góes et al., 2005)

Figura 14 – Edição do arquivo com a lista dos usuários

Fonte:(Góes et al., 2005)

Figura 15 – Criação do diretório de dados do usuário e modificação do dono do diretório

```
ubuntu@ip-172-31-48-141:~$ sudo mkdir -p /home/random2/dados
ubuntu@ip-172-31-48-141:~$ sudo chown random2 /home/random2/dados
```

Fonte:(Góes et al., 2005)

Com todas as configurações finalizadas, será necessário reiniciar o serviço "vsftpd". Portanto rodar "sudo service vsftpd restart" para se certificar de que todas as alterações sejam aplicadas.

Figura 16 – Comando para reinicializar o serviço vsftpd

```
ubuntu@ip-172-31-48-141:~$ sudo service vsftpd restart
ubuntu@ip-172-31-48-141:~$
```

Fonte:(Góes et al., 2005)

Agora será possível utilizar o comando "sudo ftp [nome]@localhost", onde [nome] substitui o nome do usuário escolhido, no caso ficaria "sudo ftp random@localhost", o que irá abrir

a conexão ftp via terminal. Também iremos executar "mkdir teste" para criar uma pasta chamada Teste pela CLI do vsftpd.

Figura 17 – Inicialização e uso do FTP pelo terminal

```

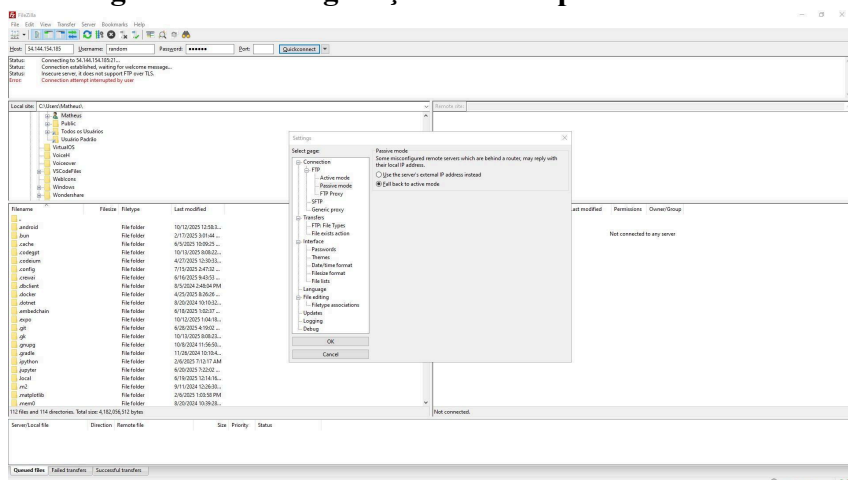
root@kali:~# ssh root@10.10.10.10
root@10.10.10.10:~# sudo ftp randomize@localhost
Connected to localhost.
220 (vsFTPd 3.0.5)
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> mkdir teste3
257 "/teste3" created
ftp>

```

Fonte:(Góes et al., 2005)

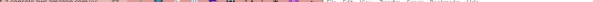
Outra maneira de efetuar a conexão FTP é utilizando o Filezilla. Basta instalar o Filezilla Client e executar na máquina. Após isso, será necessário configurá-lo para sempre optar por conexões no modo ativo por conta de configurações de firewall. Para isso vá em "Edit", depois em "Settings" e "Passive Mode" e então escolha "Fall back to active mode".

Figura 18 – Configuração Filezilla para modo ativo



Fonte:(Góes et al., 2005)

Após isso, inserir as informações do IP público do servidor da AWS e os dados do usuário para se conectar ao FTP. Aparecerá um aviso para que permita a conexão FTP insegura, e será necessário clicar em "OK" para abrir a conexão.

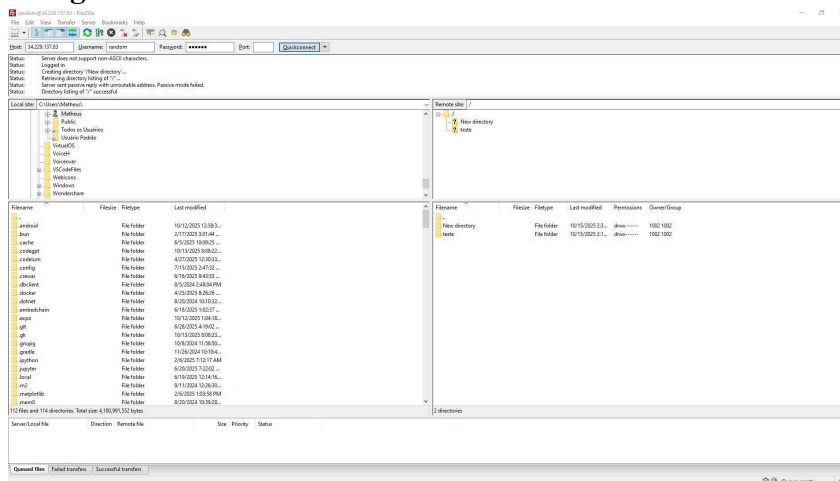


na pasta chamada "Test

note site: /



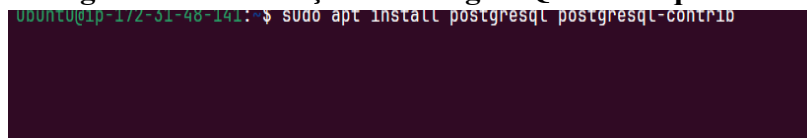
arquivo direto pelo Fil

Figura 21 – Conexão FTP Filezilla com o servidor AWS 3

Fonte:(Góes et al., 2005)

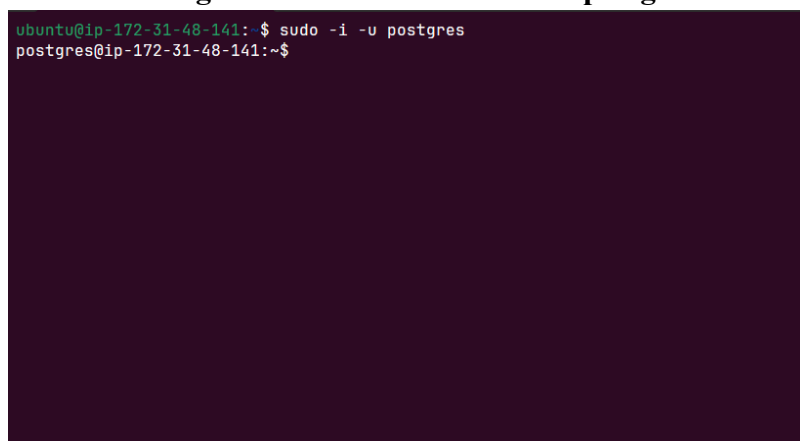
2.1.3 Banco de Dados Local PostgreSQL

A escolha para a configuração do banco de dados foi utilizar o PostgreSQL dentro de uma instância EC2, que é o modo de configuração local. Para inicializar a configuração, é necessário instalar o postgresql na máquina virtual. Para isso, no terminal rode "sudo apt install postgresql postgresql-contrib".

Figura 22 – Instalação do PostgreSQL na máquina EC2

Fonte:(Citelli, 2004)

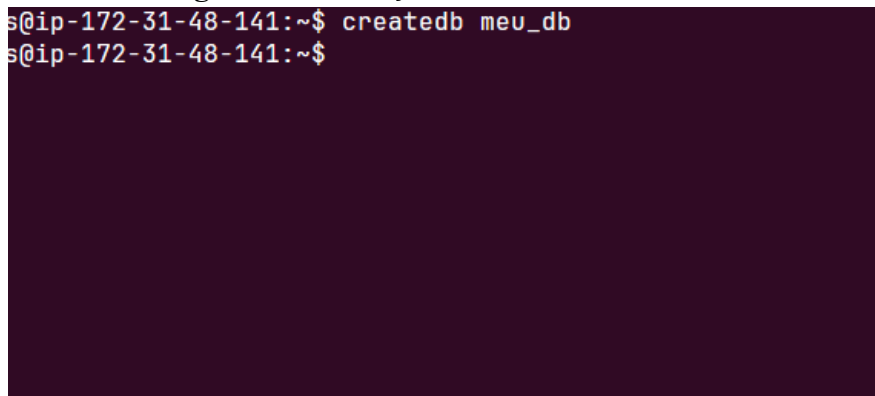
Com isso os arquivos e executáveis necessários serão instalados. Feito isso, um usuário "postgres" será instalado na instância. Com isso, ao executar o comando "sudo -i -u postgres" terá-se o acesso a este usuário.

Figura 23 – Acesso ao usuário postgres

Fonte:(Citelli, 2004)

Uma vez no terminal, será criado um banco de dados com o nome a sua escolha através do comando "createdb [nome]", onde "[nome]" será substituído pelo nome do banco de dados que deseja criar. No caso, como já existe um banco de dados chamado de teste no sistema, o terminal retornou um erro informando esta ocorrência e teve que ser criado um outro banco de dados chamado "teste2".

Figura 24 – Criação do banco de dados

A terminal window with a dark background and light-colored text. The prompt is 's@ip-172-31-48-141:~\$'. The command 'createdb meu_db' has been entered and executed, resulting in a new prompt 's@ip-172-31-48-141:~\$' on the next line.

```
s@ip-172-31-48-141:~$ createdb meu_db
s@ip-172-31-48-141:~$
```

Fonte:(Citelli, 2004)

Uma vez com o banco de dados criado, haverá a necessidade de se acessar o executável do banco de dados para gerar comandos SQL de acordo com o dialeto PostgreSQL. Para isso

Basta iniciar o cliente psql. É muito importante não se esquecer de inserir ";" ao final de cada query, caso contrário os resultados esperados não serão executados.

Figura 25 – Acesso aos comandos SQL com cliente psql

```
postgres@ip-172-31-48-141:~$ createdb meo_db
postgres@ip-172-31-48-141:~$ psql
psql (16.10 (Ubuntu 16.10-0ubuntu0.24.04.1))
Type "help" for help.

postgres=#
```

Fonte:(Citelli, 2004)

Será criado um usuário através do nome da ROLE escolhida e senha. Para isso, executar o comando SQL "CREATE USER [role] WITH ENCRYPTED PASSWORD '[pass]';", trocando "role" pelo nome do usuário e "pass" pela senha desejada ao ser requisitado o acesso ao banco por meio deste usuário.

Figura 26 – Criação de credenciais de acesso ao banco de dados com permissionamento

```
postgres=# CREATE USER qualquer WITH ENCRYPTED PASSWORD 'qualquer';
CREATE ROLE
postgres=#
```

Fonte:(Citelli, 2004)

Para fornecer as permissões para a ROLE criada, executar "GRANT ALL PRIVILEGES ON DATABASE [banco] TO [role]", onde "banco" é o nome da base de dados e a "role" corresponde ao nome da role usada para atribuir ao usuário.

Figura 27 – Permissionamento ao cargo criado

```
postgres=# GRANT ALL PRIVILEGES ON DATABASE meu_db to qualquer;  
GRANT  
postgres=#
```

Feito isso, pode-se fazer o logout do usuário postgres e voltar ao usuário ubuntu da instância EC2 para reinicializar o serviço do banco de dados e ter certeza de que todas as configurações foram devidamente aplicadas.

Figura 28 – Reinicialização do serviço postgresql

```
ubuntu@ip-172-31-48-141:~$ sudo systemctl restart postgresql  
ubuntu@ip-172-31-48-141:~$ sudo systemctl status postgresql  
● postgresql.service - PostgreSQL RDBMS  
   Loaded: loaded (/usr/lib/systemd/system/postgresql.service; enabled; preset: enabled)  
   Active: active (exited) since Sat 2025-10-18 01:27:07 UTC; 5s ago  
     Process: 2894 ExecStart=/bin/true (code=exited, status=0/SUCCESS)  
    Main PID: 2894 (code=exited, status=0/SUCCESS)  
      CPU: 3ms  
  
Oct 18 01:27:07 ip-172-31-48-141 systemd[1]: Starting postgresql.service - PostgreSQL RDBMS...  
Oct 18 01:27:07 ip-172-31-48-141 systemd[1]: Finished postgresql.service - PostgreSQL RDBMS.  
ubuntu@ip-172-31-48-141:~$
```

Fonte:(Citelli, 2004)

Com o serviço reinicializado, será necessário configurar os arquivos gerados pelo pacote postgresql. Para isso, precisaremos editar dois arquivos principais de configuração gerados no caminho "/etc/postgresql/[versao]/main", onde "[versao]" substitui a versão do postgres instalada na máquina.

Figura 29 – Caminho para editar os arquivos

```
ip-172-31-48-141:~$ cd /etc/postgresql/16/main  
ip-172-31-48-141:/etc/postgresql/16/main$ ls  
environment  pg_ctl.conf  pg_hba.conf  pg_ident.conf  postgresql.conf  start.conf  
ip-172-31-48-141:/etc/postgresql/16/main$
```

Fonte:(Citelli, 2004)

Execute o editor de texto Linux baseado em preferência. No caso será utilizado o GNU nano para modificar inicialmente o conteúdo do arquivo "postgresql.conf". Portanto será rodado "sudo nano postgresql.conf"

Figura 30 – Comando para editar o arquivo com GNU nano

```
ubuntu@ip-172-31-48-141:/etc/postgresql/16/main$ sudo nano postgresql.conf
```

Fonte:(Citelli, 2004)

Este arquivo é responsável por gerenciar os caminhos e para quais computadores/IPs o banco de dados ficará disposto à conexão. Por isso, será necessário localizar o campo "listen_addresses" e o valor mapeado, trocando para "*" como forma de aceitar conexões com qualquer outro host. Certificar-se inclusive de que o caminho do diretório para "hba_file" consta como o correto. A porta fica a critério de escolha, mas geralmente se utiliza o padrão(5432).

Figura 31 – Configuração do arquivo postgresql.conf

```
data_directory = '/var/lib/postgresql/16/main'      # use data in another directory
                                                    # (change requires restart)
hba_file = '/etc/postgresql/16/main/pg_hba.conf'    # host-based authentication file
                                                    # (change requires restart)
ident_file = '/etc/postgresql/16/main/pg_ident.conf' # ident configuration file
                                                    # (change requires restart)

# If external_pid_file is not explicitly set, no extra PID file is written.
external_pid_file = '/var/run/postgresql/16-main.pid' # write an extra PID file
                                                    # (change requires restart)

#-----
# CONNECTIONS AND AUTHENTICATION
#-----

# - Connection Settings -

listen_addresses = '*' # what IP address(es) to listen on;
                        # comma-separated list of addresses;
                        # defaults to 'localhost'; use '*' for all
                        # (change requires restart)
port = 5432            # (change requires restart)
max_connections = 100  # (change requires restart)
#reserved_connections = 0 # (change requires restart)
#superuser_reserved_connections = 3 # (change requires restart)
```

Em seguida, será necessário editar o arquivo "pg_hba.conf", referenciado anteriormente no arquivo "postgresql.conf". O "pg_hba.conf" é responsável por autenticar as conexões que são efetuadas através dos IPs das máquinas que requerem o acesso ao banco de dados. Rodar "sudo nano pg_hba.conf".

Figura 32 – Comando para editar o arquivo com GNU nano

```
ubuntu@ip-172-31-48-141:/etc/postgresql/16/main$ sudo nano pg_hba.conf
```

Fonte:(Citelli, 2004)

Ao ser exibido, o arquivo demonstrará várias configurações, entretanto é no final do arquivo que se encontram as configurações de autenticação de hosts.

Dentre essas configurações será necessário adicionar uma linha setada como "host", utilizando "all" no IP "0.0.0/0", ou seja, qualquer IP, e no modo "md5".

Para aplicar as alterações, reinicialize novamente o serviço postgresql.

Figura 33 – Comando para editar o arquivo com GNU nano

```
# DO NOT DISABLE!
# If you change this first entry you will need to make sure that the
# database superuser can access the database using some other method.
# Noninteractive access to all databases is required during automatic
# maintenance (custom daily cronjobs, replication, and similar tasks).
#
# Database administrative login by Unix domain socket
local  all                postgres                                peer

# TYPE  DATABASE        USER            ADDRESS              METHOD

# "local" is for Unix domain socket connections only
local  all                all                                peer
# IPv4 local connections:
host   all          all              127.0.0.1/32         scram-sha-256
# IPv6 local connections:
host   all          all              ::1/128              scram-sha-256
# Allow replication connections from localhost, by a user with the
# replication privilege.
local  replication    all                                peer
host   replication    all              127.0.0.1/32         scram-sha-256
host   replication    all              ::1/128              scram-sha-256
host   all            all              0.0.0.0/0            md5
```

Figura 34 – Reinicialização do serviço postgresql

```
1: $ sudo systemctl restart postgresql
1: $ sudo systemctl status postgresql
PostgreSQL RDBMS
usr/lib/systemd/system/postgresql.service; enabled; preset: enabled)
xited) since Sat 2025-10-18 01:33:54 UTC; 4s ago
Start=/bin/true (code=exited, status=0/SUCCESS)
e=exited, status=0/SUCCESS

-31-48-141 systemd[1]: Starting postgresql.service - PostgreSQL RDBMS...
-31-48-141 systemd[1]: Finished postgresql.service - PostgreSQL RDBMS.
1: $
```

Fonte:(Citelli, 2004)

Com isso, o servidor que hospeda o banco de dados já está configurado. Entretanto, para que as máquinas cliente possam se conectar ao banco de dados é preciso expor a porta em que o serviço postgresql irá rodar, especificada anteriormente no arquivo de configuração "postgresql.conf". No caso, como especificado anteriormente, a porta padrão "5432" está sendo utilizada para que o servidor aceite as conexões mapeadas para esta porta. Portanto, será necessário configurar isso para o firewall do servidor no console AWS e no terminal. Rodar "sudo ufw allow [porta]", em que [porta] será a porta mapeada como dito, no caso "5432".

Figura 35 – Permissão de tráfego para a porta 5432 pelo terminal


```
Oct 18 01:55:34 ip-172-31-48-141 systemd[1]: Finished postgresql.service - PostgreSQL Kubuntu.  
ubuntu@ip-172-31-48-141:~$ sudo ufw allow 5432/tcp  
Skipping adding existing rule  
Skipping adding existing rule (v6)  
ubuntu@ip-172-31-48-141:~$
```

Como a regra já havia sido adicionada, houve um aviso. Para a AWS, basta ir no grupo de segurança da instância e adicionar a mesma regra.

Figura 36 – Permissão de tráfego para a porta 5432 no Console AWS

sgp-0806b8e50ed81a8f7 5432 TCP 0.0.0.0/0 launch-wizard-1

Fonte:(Citelli, 2004)

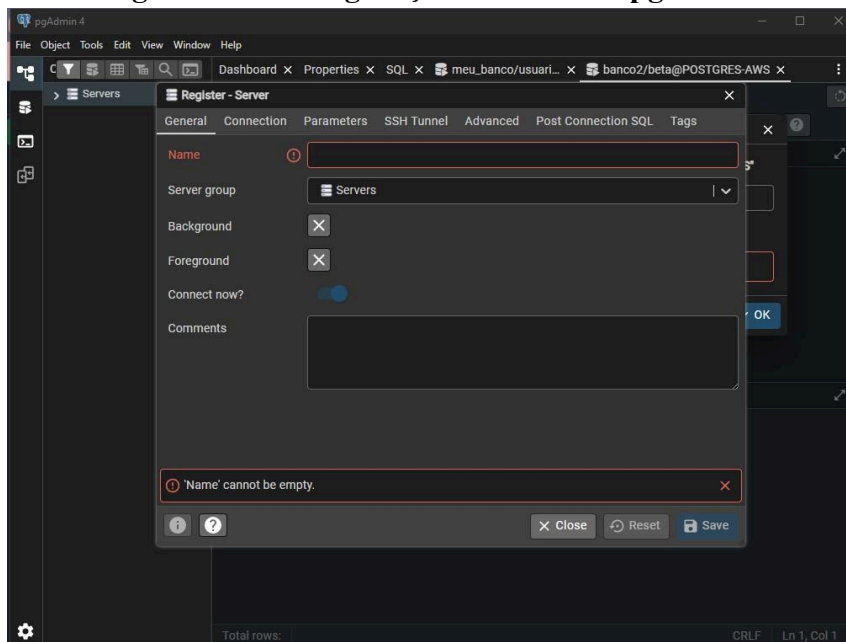
O teste será efetuado de duas maneiras: pelo terminal e pgAdmin. Para fazer o teste via terminal utilize uma outra instância EC2 na AWS no mesmo grupo de segurança ou em um grupo que tenha permissão para tráfego na porta 5432 e, no terminal, execute "sudo psql -h "[ip]-U [role] -d [db]". Os campos "[ip]", "[role]" e "[db]" se referem respectivamente ao IP do servidor: no caso de estar na mesma rede virtual privada criada pela AWS pode utilizar o IP privado do servidor, do contrário utilizar o IP público, ao usuário que deseja utilizar para estabelecer a conexão e por fim em qual banco de dados estabelecer tal conexão.

Figura 37 – Conexão com o banco de dados pelo terminal

```
ubuntu@ip-172-31-51-29:~$ sudo psql -h "172.31.48.141" -U qualquer -d meu_db  
Password for user qualquer:  
psql (16.10 (Ubuntu 16.10-0ubuntu0.24.04.1))  
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off)  
Type "help" for help.  
  
meu_db=>
```

Já pelo pgAdmin, o usuário terá que acessar o menu "Object", "register" e "Server", o que irá abrir a janela de configurações.

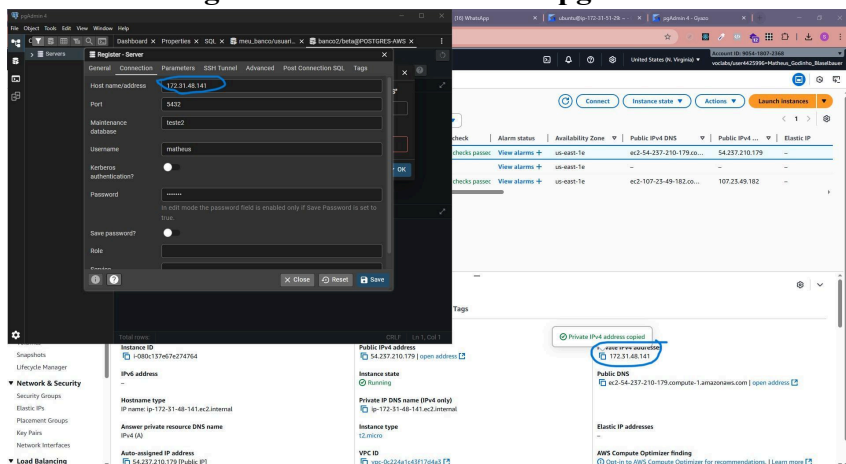
Figura 38 – Configuração conexão via pgAdmin



Fonte:(Citelli, 2004)

Insira um nome qualquer a sua escolha para o grupo de servidores, pois não pode estar vazio. Depois vá à aba "Connection" ou "Conexões", preencha as informações corretamente e clique em "Save" ou "Salvar". Pronto, o cliente pgAdmin já está conectado ao banco de dados.

Figura 39 – Credenciais via pgAdmin



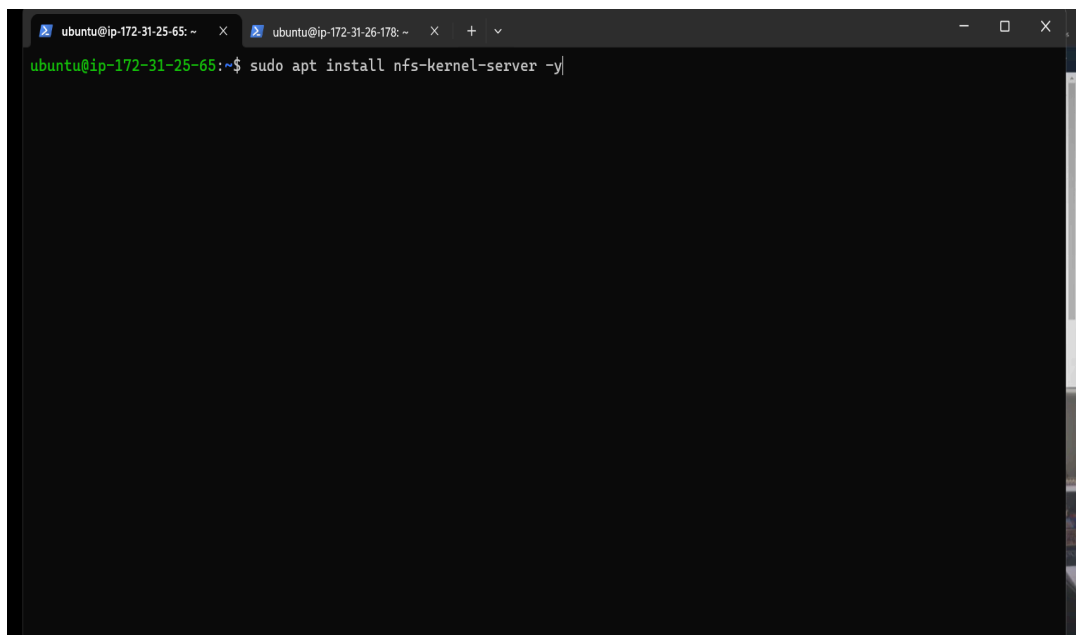
Fonte:(Citelli, 2004)

2.1.4 Network File System

O protocolo NFS (Network File System), traduzido como Sistema de Arquivos em Rede, permite que um computador cliente acesse diretórios e arquivos em um servidor pela rede como se eles estivessem em seu próprio disco local. Ele opera em uma arquitetura cliente-servidor e utiliza o protocolo RPC (Remote Procedure Call) para a comunicação. O funcionamento baseia-se em dois conceitos principais: a exportação no lado do servidor, onde se define qual diretório será compartilhado, e a montagem (mount) no lado do cliente, que anexa esse diretório remoto a uma pasta local.

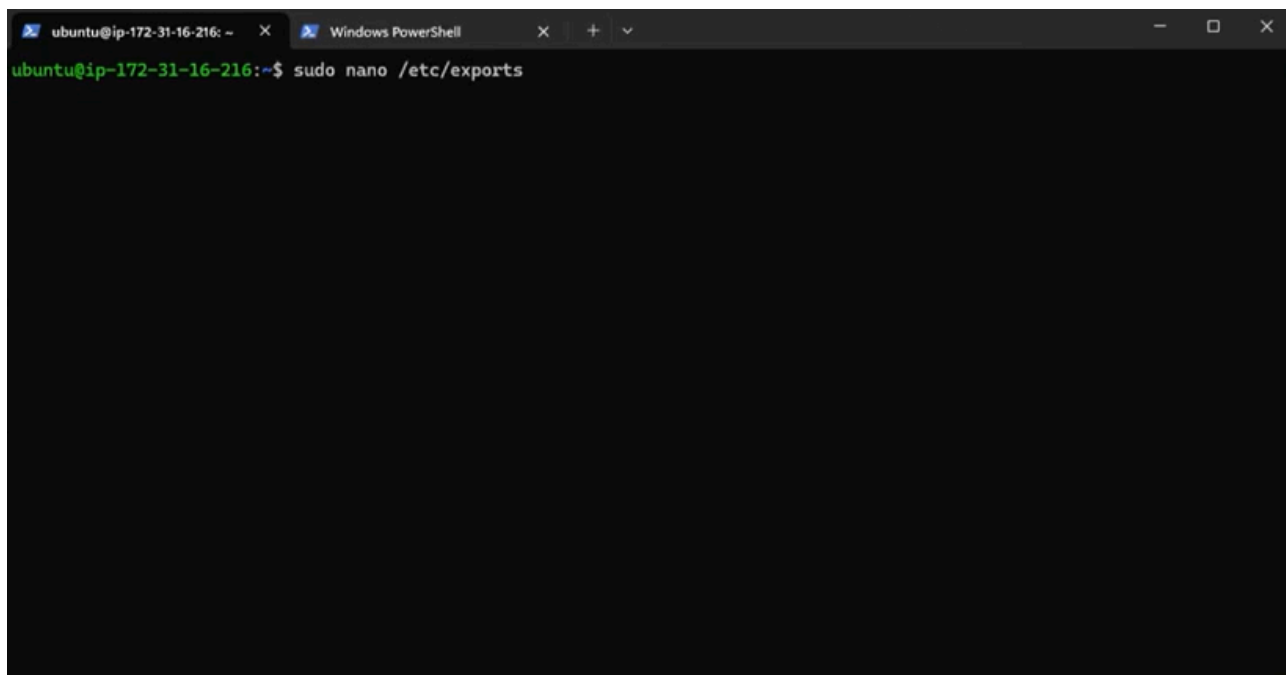
Para configurar um servidor NFS em um sistema Linux, primeiramente é necessário instalar o pacote `nfs-kernel-server` em uma máquina que vamos usar como servidor..

Figura 40 – Instalação do nfs-server



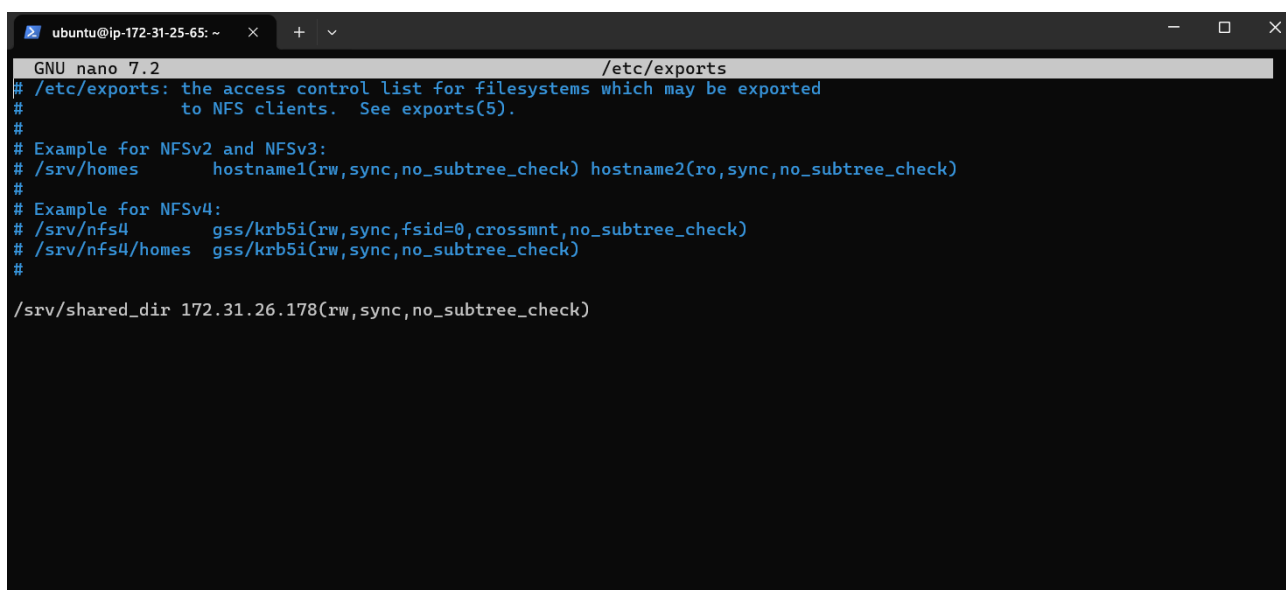
Com isso vamos ter acesso a parte do arquivo de configuração dentro do `/etc/exports`, então basta acessa-lo usando por exemplo o editor NANO como foi feito na imagem abaixo:

Figura 41 – acessando as configurações



Já nas configurações a gente vai usar o comando **/srv/shared_dir 172.31.26.178(rw, sync, no_subtree_check)**, o **/srv** é para deixar dentro da pasta srv mesmo, já o **/shared_dir** é a pasta compartilhada que vamos criar, em seguida vem a faixa, é uma boa pratica adicionar os IPs de quem vai acessar esse serviço, e então esse IP vai ter essas propriedades que estão entre parênteses(escrita, sincronização e o **no_subtree_check** que não deixa ficar com o desempenho muito lento)

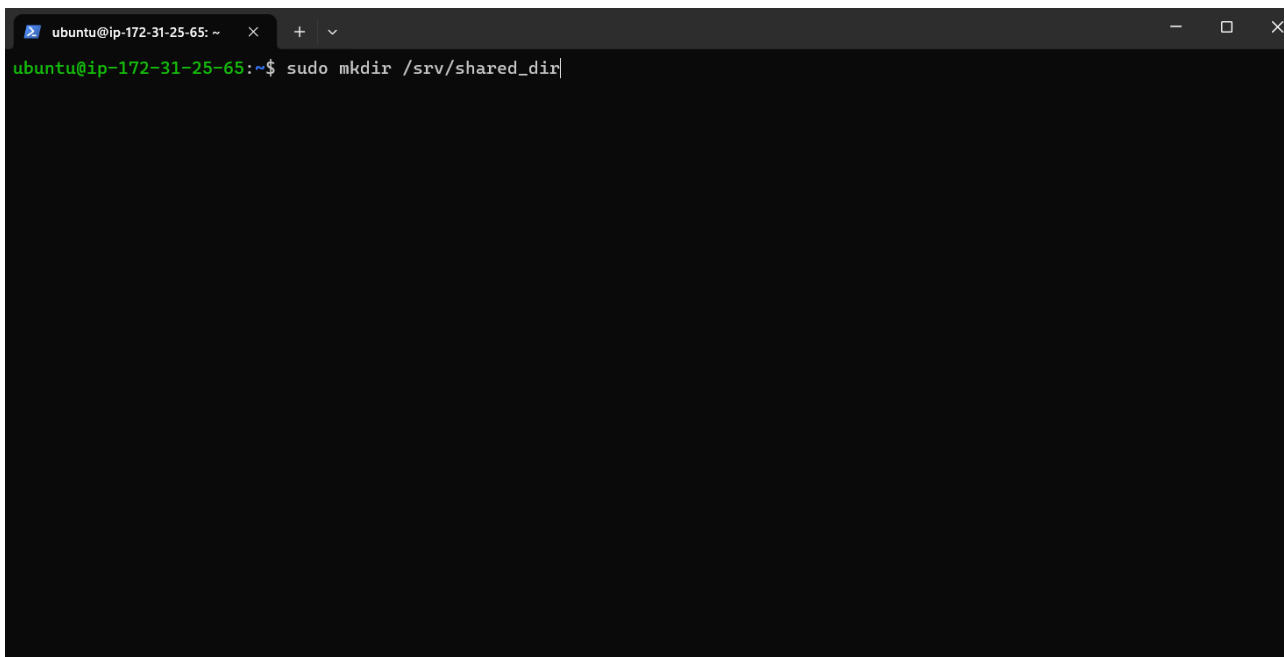
Figura 42 – adicionando as configurações necessarias



Logo em seguida, salvamos essa configuração e então criamos a pasta **shared_dir** com o comando

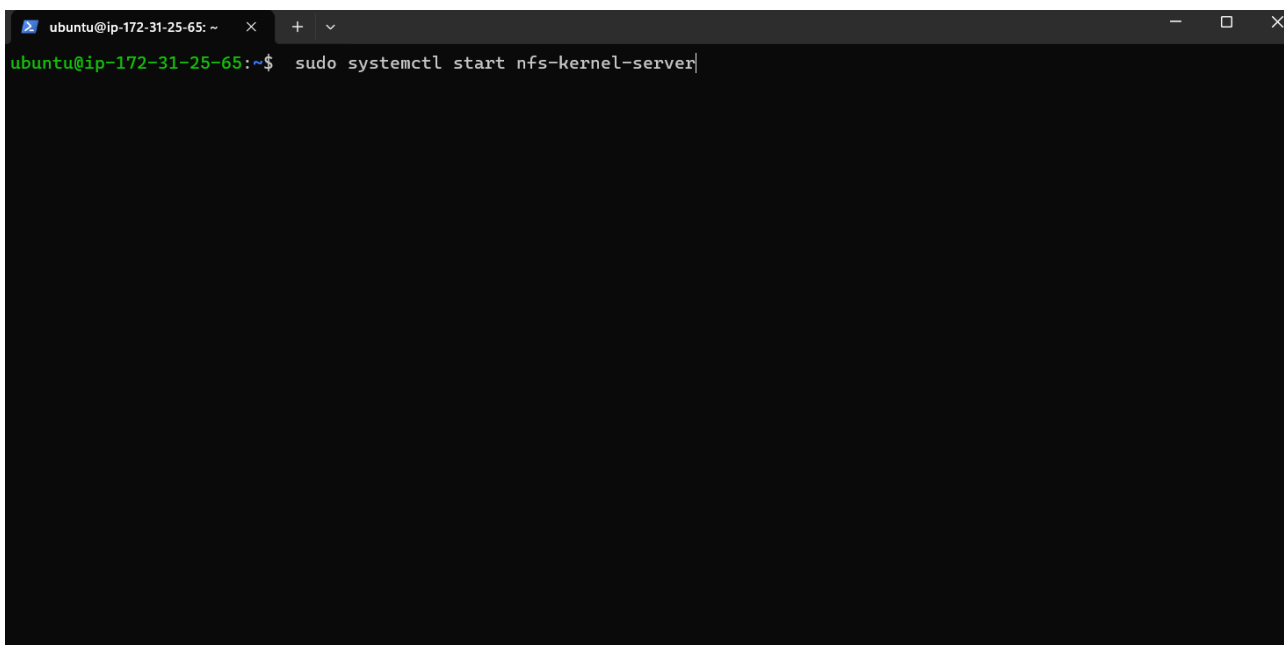
sudo mkdir /srv/shared_dir

Figura 43 – Criando a pasta compartilhada



Depois disso vamos iniciar o serviço com o comando **sudo systemctl start nfs-kernel-server**

Figura 44 – Iniciando o serviço



Para confirmar que foi iniciado com sucesso você pode usar o comando **sudo systemctl status nfs-kernel-server**

Figura 45 – Testando se o serviço foi iniciado com sucesso

```
ubuntu@ip-172-31-25-65: ~  
ubuntu@ip-172-31-25-65:~$ sudo systemctl status nfs-kernel-server  
● nfs-server.service - NFS server and services  
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; preset: enabled)  
   Drop-In: /run/systemd/generator/nfs-server.service.d  
            └─order-with-mounts.conf  
   Active: active (exited) since Thu 2025-10-16 22:47:05 UTC; 44min ago  
   Process: 648 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)  
   Process: 661 ExecStart=/usr/sbin/rpc.nfsd (code=exited, status=0/SUCCESS)  
   Main PID: 661 (code=exited, status=0/SUCCESS)  
      CPU: 17ms  
  
Oct 16 22:47:04 ip-172-31-25-65 systemd[1]: Starting nfs-server.service - NFS server and services...  
Oct 16 22:47:05 ip-172-31-25-65 systemd[1]: Finished nfs-server.service - NFS server and services.  
ubuntu@ip-172-31-25-65:~$ |
```

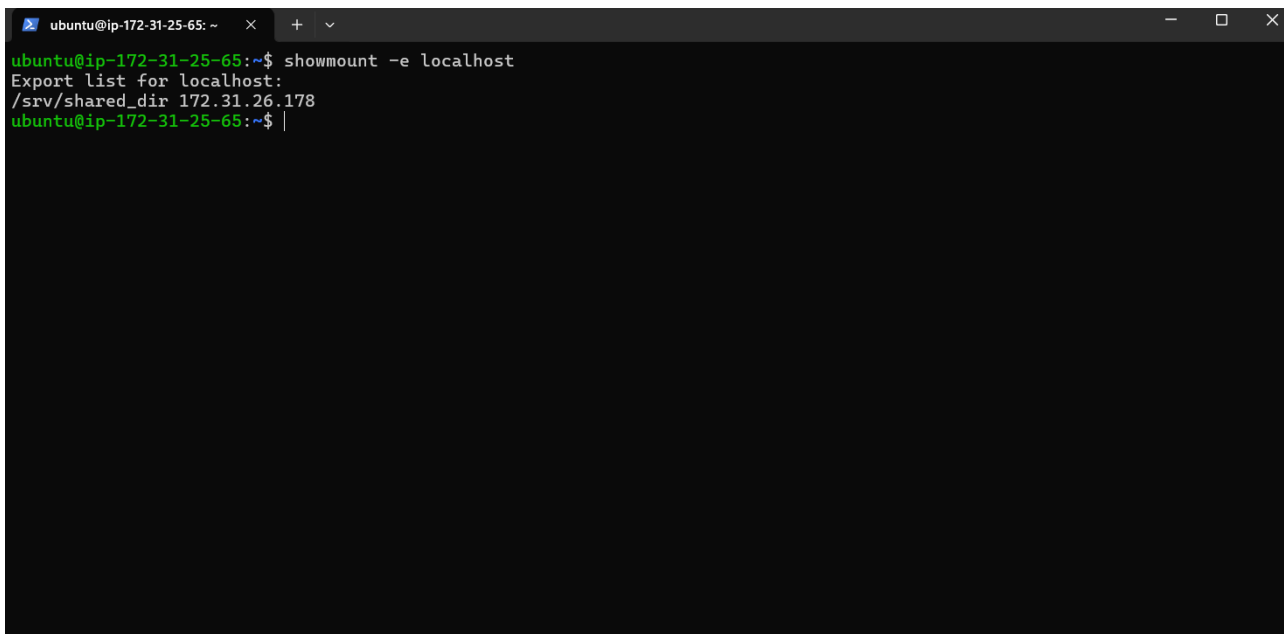
Nota-se que precisa estar tudo verde para dizer que deu certo. Em seguida vamos usar o comando **sudo exportfs -ra** para aplicar as configurações que foram feitas anteriormente

Figura 46 – Aplicando as configurações feitas

```
ubuntu@ip-172-31-25-65: ~  
ubuntu@ip-172-31-25-65:~$ sudo systemctl status nfs-kernel-server  
● nfs-server.service - NFS server and services  
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; preset: enabled)  
   Drop-In: /run/systemd/generator/nfs-server.service.d  
            └─order-with-mounts.conf  
   Active: active (exited) since Thu 2025-10-16 22:47:05 UTC; 44min ago  
   Process: 648 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)  
   Process: 661 ExecStart=/usr/sbin/rpc.nfsd (code=exited, status=0/SUCCESS)  
   Main PID: 661 (code=exited, status=0/SUCCESS)  
      CPU: 17ms  
  
Oct 16 22:47:04 ip-172-31-25-65 systemd[1]: Starting nfs-server.service - NFS server and services...  
Oct 16 22:47:05 ip-172-31-25-65 systemd[1]: Finished nfs-server.service - NFS server and services.  
ubuntu@ip-172-31-25-65:~$ sudo exportfs -ra|
```

Um teste que pode ser feito para ver o que está sendo oferecido pelo servidor é o do comando **showmount -e localhost** que vai mostrar todos os diretórios que estão sendo exportados nesse servidor, como mostra a imagem abaixo:

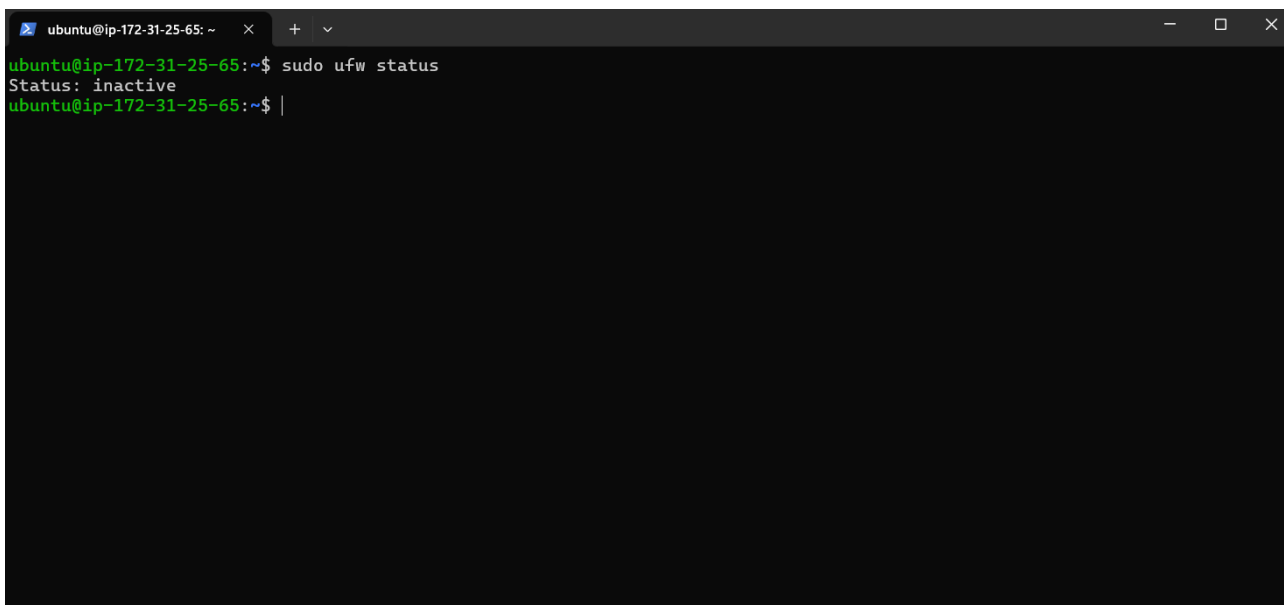
Figura 47 – Teste para ver diretórios exportados



```
ubuntu@ip-172-31-25-65: ~  
ubuntu@ip-172-31-25-65:~$ showmount -e localhost  
Export list for localhost:  
/srv/shared_dir 172.31.26.178  
ubuntu@ip-172-31-25-65:~$
```

Agora precisamos liberar as portas do serviço para saber se está tudo ok, a gente consegue olhar usando o comando **sudo ufw status**

Figura 48 – Testar os estados das portas



```
ubuntu@ip-172-31-25-65: ~  
ubuntu@ip-172-31-25-65:~$ sudo ufw status  
Status: inactive  
ubuntu@ip-172-31-25-65:~$
```

Para liberar as portas nós vamos no AWS Management Console, em seguida vamos na instância servidor que no meu caso foi o SrvUbunto1.

Figura 49 – Acessando a instancia servidor

Resumo da instância para i-0a5fd07de8cfc078d (SrvUbuntu1)

Informações

Atualizado há less than a minute

ID da instância

i-0a5fd07de8cfc078d

Endereço IPv6

-

Tipo de nome do host

Nome do IP: ip-172-31-25-65.ec2.internal

Nome do DNS do recurso privado de resposta

IPv4 (A)

Endereço IP atribuído automaticamente

3.85.44.235 [IP público]

Função do IAM

-

ARMADA

Endereço IPv4 público

3.85.44.235 | endereço aberto

Estado da instância

Executando

Nome do DNS de IP privado (somente IPv4)

ip-172-31-25-65.ec2.internal

Tipo de instância

t3.micro

ID da VPC

vpc-0de67ca73ac91e2e5

ID da sub-rede

subnet-065ca60ba1de40006

ARN da instância

Endereços IPv4 privados

172.31.25.65

DNS pública

ec2-3-85-44-235.compute-1.amazonaws.com | endereço aberto

Endereços IP elásticos

-

Descoberta do AWS Compute Optimizer

Opte por participar do AWS Compute Optimizer para obter recomendações. Saiba mais

Nome do Grupo do Auto Scaling

-

Capacidade

Conectar

Estado da instância

Ações

Depois vamos em segurança-> grupo de segurança-> editar regras de entrada e adicionar duas regra, uma de TCP personalizada com a porta 111 e outra TCP personalizada 2049

Figura 50 – Adicionando as portas 2049 e 111 nas regras de entrada

Editar regras de entrada

Informações

As regras de entrada controlam o tráfego de entrada que tem permissão para acessar a instância.

Regras de entrada

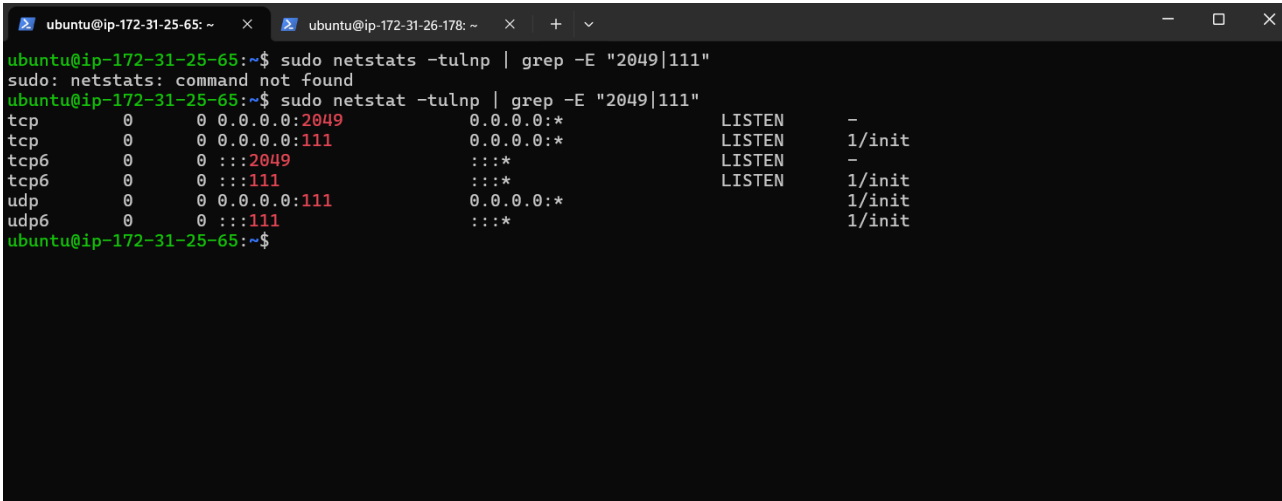
Informações

| ID da regra do grupo de segurança | Tipo | Protocolo | Intervalo de portas | Origem | Descrição - opcional |
|-----------------------------------|-----------------------|-------------|---------------------|-------------|----------------------|
| | Informações | Informações | Informações | Informações | Informações |
| sgr-0d182b7f3c31b125f | Todos os ICMPs - IPv4 | ICMP | Tudo | Perso... | Ping ICMP |
| sgr-0c8e94728bca9ff5b | SSH | TCP | 22 | Perso... | Acesso remoto |
| sgr-004d45e7f3aafb079 | TCP personalizado | TCP | 111 | Perso... | |
| sgr-036216d0cc90a1928 | NFS | TCP | 2049 | Perso... | |

Adicionar regra

Para verificar se as portas que foram adicionadas estão funcionando é so usar o comando `sudo netstats -tulnp | grep -E "2049|111"`

Figura 51 – Verificando se as portas foram adicionadas e se estão funcionando

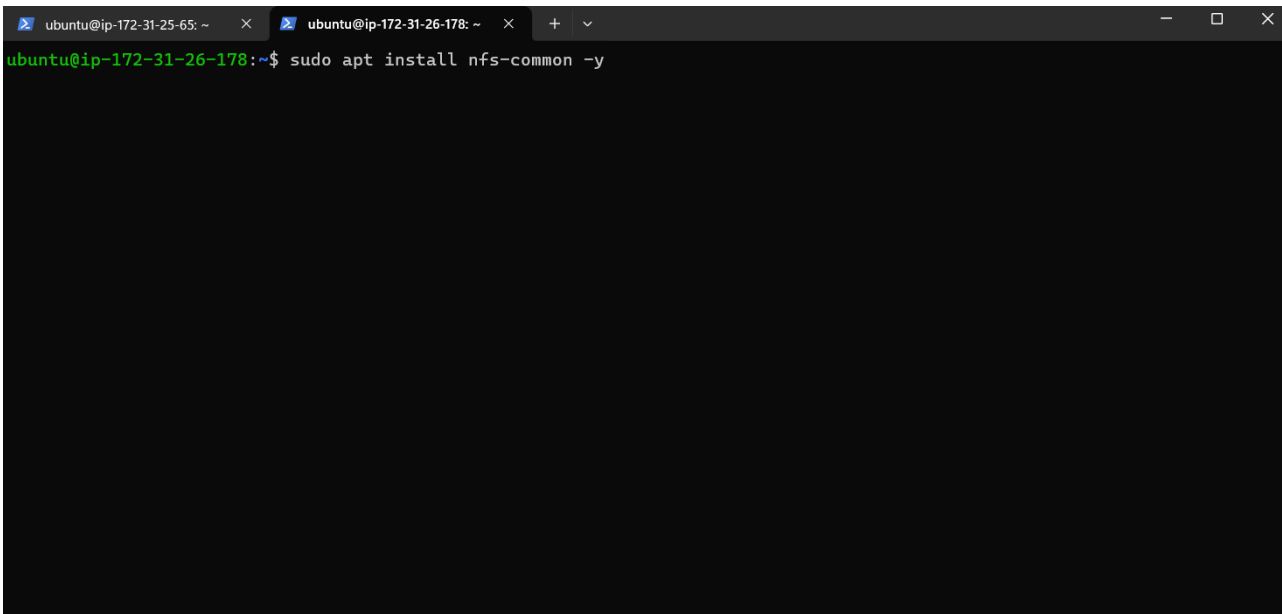


```

ubuntu@ip-172-31-25-65: ~$ sudo netstats -tulnp | grep -E "2049|111"
sudo: netstats: command not found
ubuntu@ip-172-31-25-65:~$ sudo netstat -tulnp | grep -E "2049|111"
tcp        0      0 0.0.0.0:2049          0.0.0.0:*        LISTEN     -
tcp        0      0 0.0.0.0:111          0.0.0.0:*        LISTEN     1/init
tcp6       0      0 :::2049              :::*              LISTEN     -
tcp6       0      0 :::111               :::*              LISTEN     1/init
udp        0      0 0.0.0.0:111          0.0.0.0:*        1/init
udp6       0      0 :::111               :::*              1/init
ubuntu@ip-172-31-25-65:~$
  
```

Agora vamos configurar a máquina do outro lado, no caso a do cliente. Começamos baixando o **nfs-common** na máquina cliente:

Figura 52 – Instalando o nfs-common

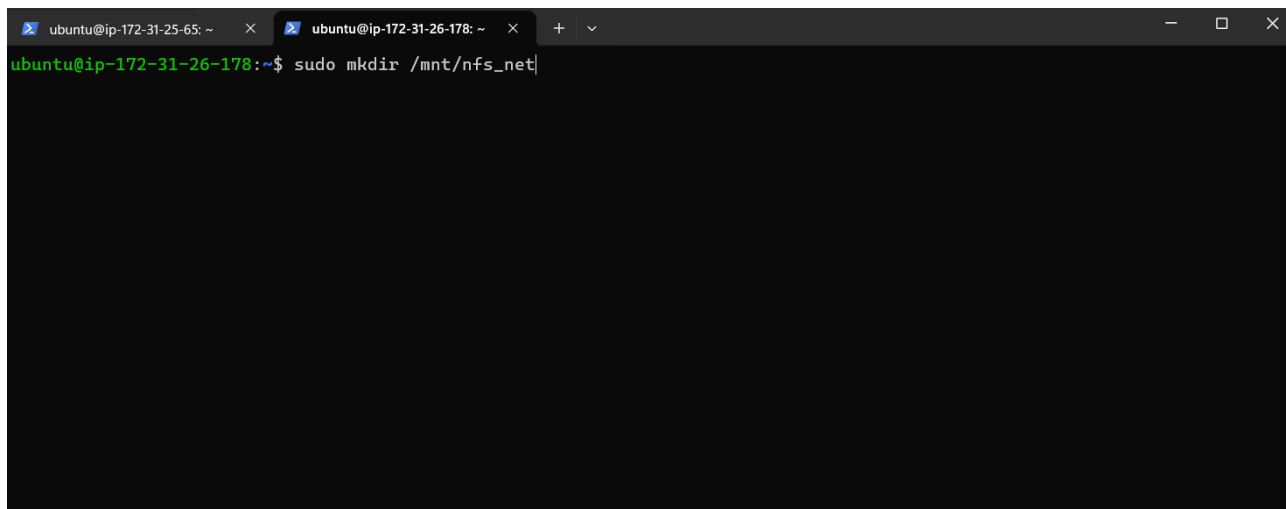


```

ubuntu@ip-172-31-26-178:~$ sudo apt install nfs-common -y
  
```

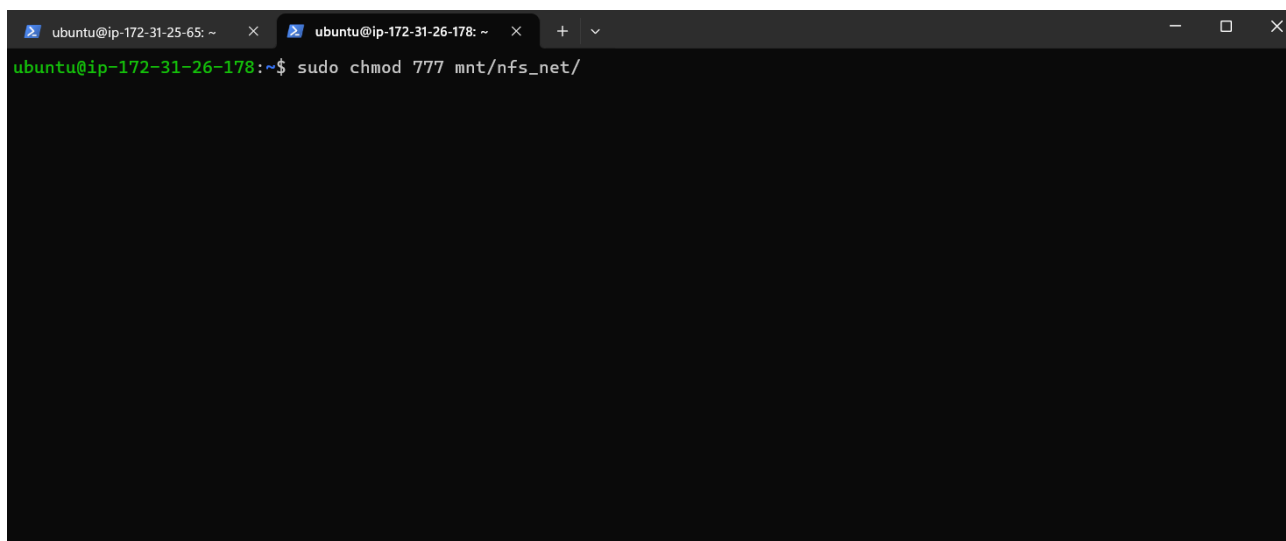
Depois de baixado criamos um diretório dentro do repositório **/mnt** com o comando **sudo mkdir /mnt/nfs_net**

Figura 53 – Criando repositório nfs_net



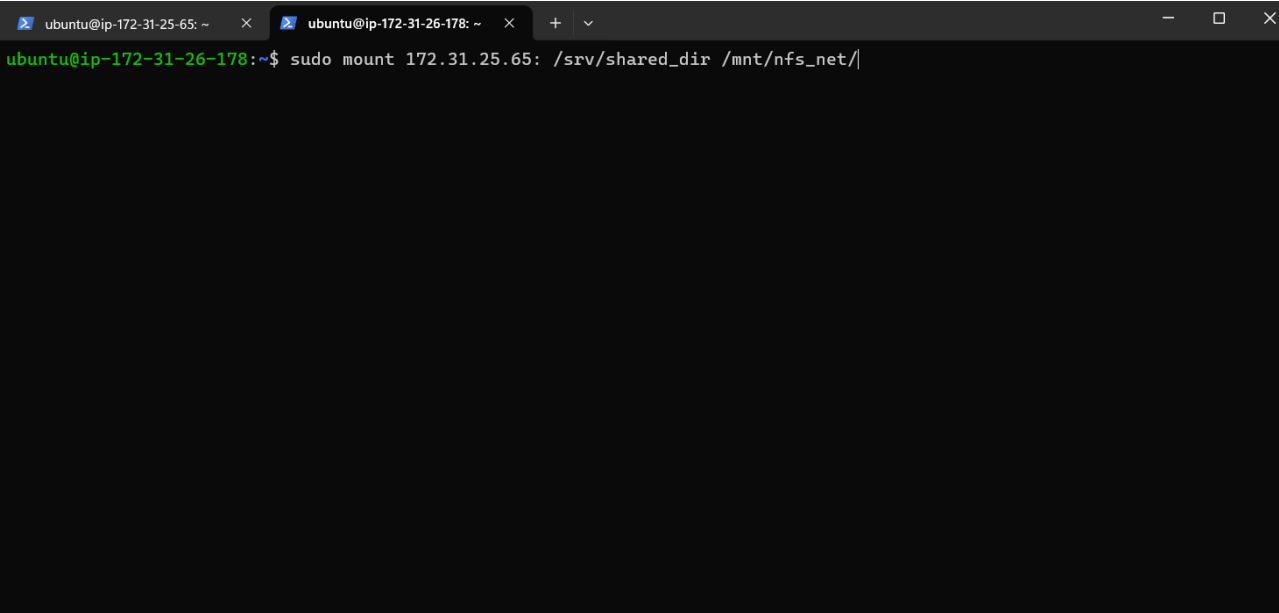
Em seguida damos permissão a esse diretório com o comando **sudo chmod 777 mnt/nfs_net/**

Figura 54 – Dando permissão total pro diretório



E agora vamos montar um ponto dentro desse diretório que foi criado, usando o IP da máquina servidor usando o comando **sudo mount 172.31.25.65:/srv/shared_dir mnt/nfs_net/**

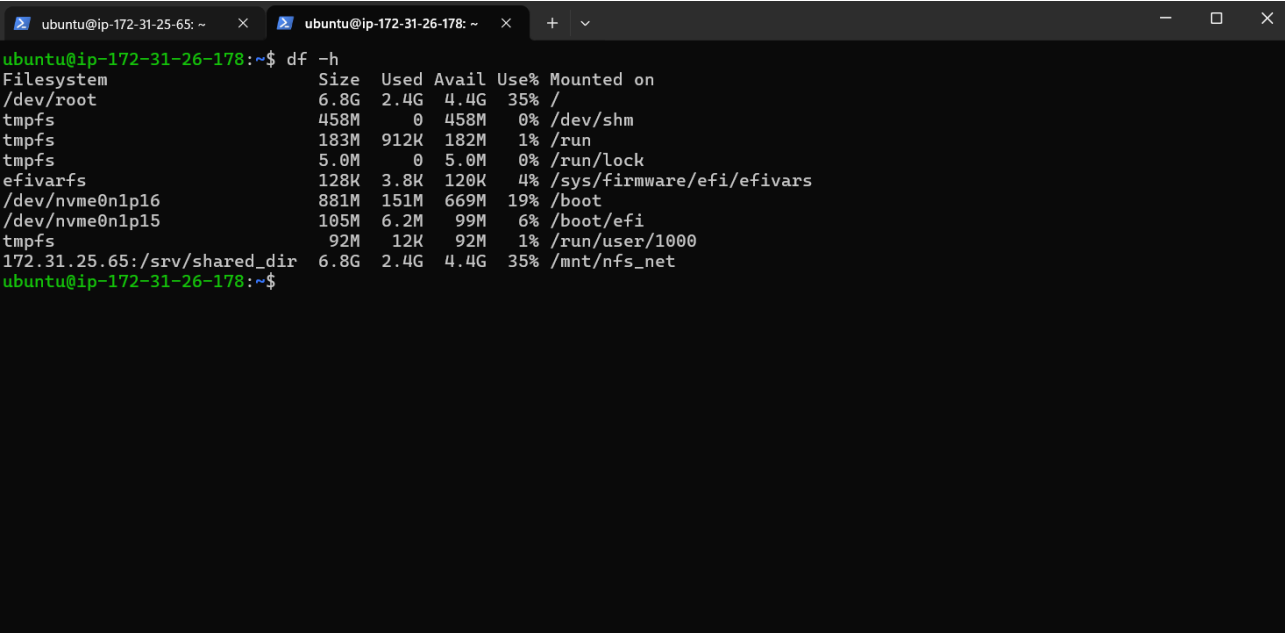
Figura 55 – Criando um ponto no diretório



```
ubuntu@ip-172-31-25-65: ~  
ubuntu@ip-172-31-26-178: ~  
ubuntu@ip-172-31-26-178:~$ sudo mount 172.31.25.65: /srv/shared_dir /mnt/nfs_net/
```

Para testar se nao deu nenhum erro é so usar o comando **df -h**

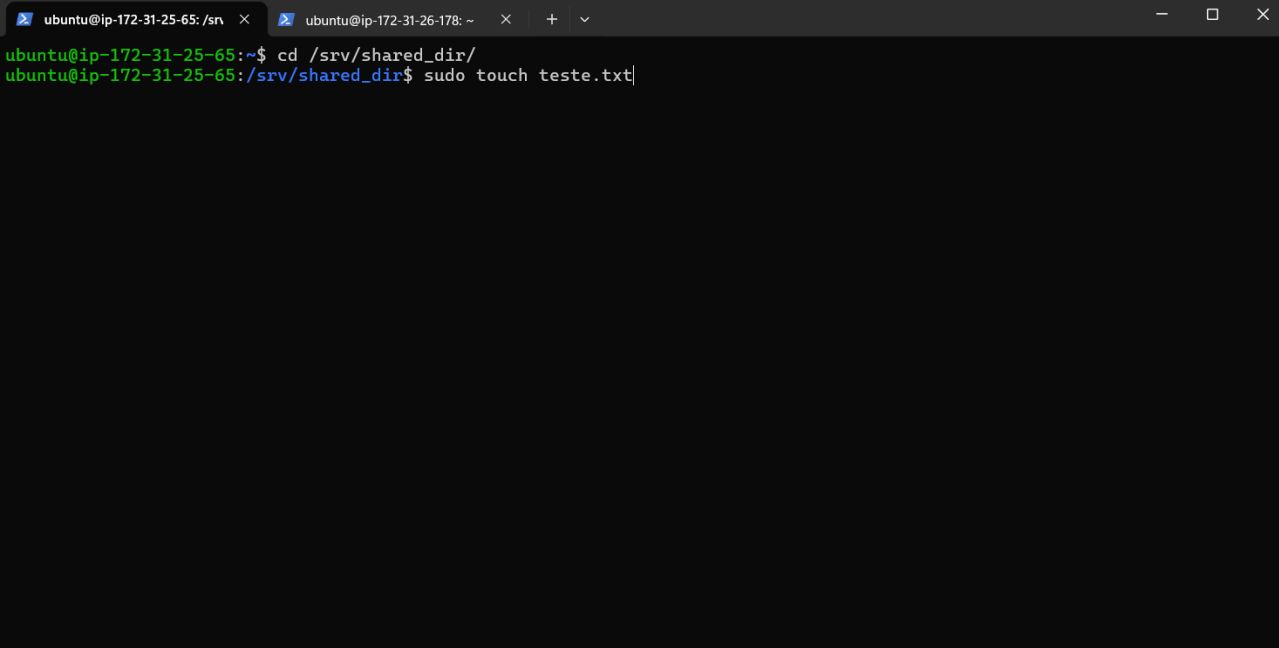
Figura 56 – Testando para ver se nao deu erro



```
ubuntu@ip-172-31-25-65: ~  
ubuntu@ip-172-31-26-178: ~  
ubuntu@ip-172-31-26-178:~$ df -h  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/root        6.8G  2.4G  4.4G  35% /  
tmpfs            458M   0  458M   0% /dev/shm  
tmpfs            183M  912K  182M   1% /run  
tmpfs            5.0M   0   5.0M   0% /run/lock  
efivarfs         128K  3.8K  120K   4% /sys/firmware/efi/efivars  
/dev/nvme0n1p16  881M  151M  669M  19% /boot  
/dev/nvme0n1p15  105M   6.2M   99M   6% /boot/efi  
tmpfs            92M   12K   92M   1% /run/user/1000  
172.31.25.65:/srv/shared_dir 6.8G  2.4G  4.4G  35% /mnt/nfs_net  
ubuntu@ip-172-31-26-178:~$
```

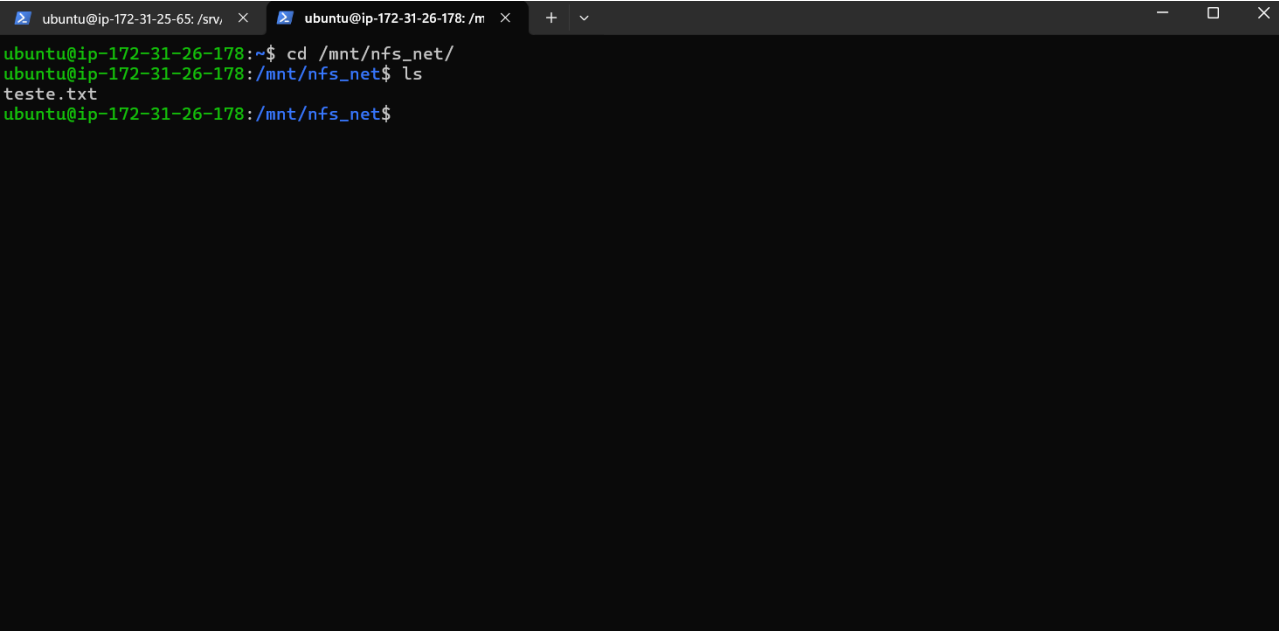
Agora vamos testar na prática entrando nesse diretório com o comando **cd /mnt/nfs_net** na máquina cliente e na máquina servidor usamos o **cd /srv/shared_dir/** e então usamos o **sudo touch teste.txt**

Figura 57 – Criando um arquivo teste no servidor



```
ubuntu@ip-172-31-25-65: /srv/ × ubuntu@ip-172-31-26-178: ~ × + v
ubuntu@ip-172-31-25-65:~$ cd /srv/shared_dir/
ubuntu@ip-172-31-25-65:/srv/shared_dir$ sudo touch teste.txt|
```

Figura 58 – Conferindo se o arquivo chegou no cliente



```
ubuntu@ip-172-31-25-65: /srv/ × ubuntu@ip-172-31-26-178: /m × + v
ubuntu@ip-172-31-26-178:~$ cd /mnt/nfs_net/
ubuntu@ip-172-31-26-178:/mnt/nfs_net$ ls
teste.txt
ubuntu@ip-172-31-26-178:/mnt/nfs_net$
```

Pronto o **Network File System** está completo.

2.1.5 Domain Name System(DNS)

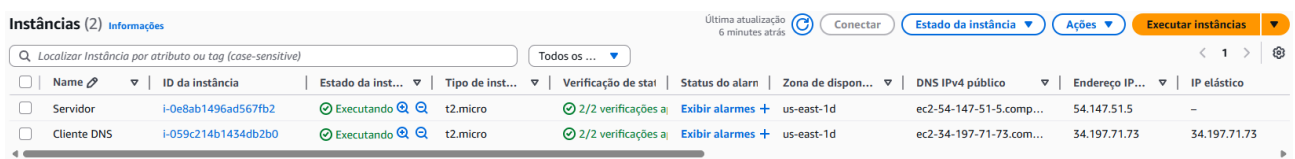
O DNS, sigla para Domain Name System (Sistema de Nomes de Domínio), é um dos pilares fundamentais da internet. Sua principal função é atuar como uma "lista telefônica da web", traduzindo os nomes de domínio que são fáceis de ler e memorizar para humanos (como `www.google.com`) nos endereços de IP numéricos (como `111.111.111.11`) que as máquinas usam para se identificar e se comunicar na rede.

Quando um usuário digita o endereço de um site em seu navegador, uma consulta é enviada a um servidor DNS. Este servidor localiza o endereço IP correspondente àquele domínio e o devolve ao navegador, permitindo que a conexão com o servidor do site seja estabelecida. Sem o DNS, teríamos que memorizar sequências complexas de números para cada site que quiséssemos visitar, o que tornaria a navegação na internet inviável. Portanto, ele é um sistema essencial que organiza e facilita o acesso a recursos online de forma transparente e eficiente.

Foi criado na AWS, duas instâncias:

Instância do servidor DNS;

Instância do cliente;



| Nome | ID da instância | Estado da instância | Tipo de instância | Verificação de status | Status do alarm | Zona de disponibilidade | DNS IPv4 público | Endereço IP público | IP elástico |
|-------------|---------------------|---------------------|-------------------|--------------------------------|------------------|-------------------------|-------------------------|---------------------|--------------|
| Servidor | i-0e8ab1496ad567fb2 | Executando | t2.micro | 2/2 verificações bem-sucedidas | Exibir alarmes + | us-east-1d | ec2-54-147-51-5.com... | 54.147.51.5 | - |
| Cliente DNS | i-059c214b1434db2b0 | Executando | t2.micro | 2/2 verificações bem-sucedidas | Exibir alarmes + | us-east-1d | ec2-34-197-71-73.com... | 34.197.71.73 | 34.197.71.73 |

Servidor Bind9 (172.31.30.87)

Este é o servidor DNS principal. Ele foi configurado para resolver nomes dentro do domínio privado `soloforteagro.teste` e também fazer a resolução reversa (IP → nome).

Figura 59 - Verificação se o serviço Bind9 está rodando:

```

ubuntu@ip-172-31-30-87: ~$ sudo systemctl status named
● named.service - BIND Domain Name Server
   Loaded: loaded (/usr/lib/systemd/system/named.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-10-17 21:27:30 UTC; 19min ago
     Docs: man:named(8)
  Main PID: 530 (named)
    Status: "running"
     Tasks: 5 (limit: 1121)
  Memory: 31.0M (peak: 31.2M)
     CPU: 67ms
  CGroup: /system.slice/named.service
          └─530 /usr/sbin/named -f -u bind

Oct 17 21:27:30 ip-172-31-30-87 named[530]: network unreachable resolving './NS/IN': 2001:7fe::53#53
Oct 17 21:27:30 ip-172-31-30-87 named[530]: network unreachable resolving './NS/IN': 2001:503:ba3e::2>
Oct 17 21:27:30 ip-172-31-30-87 named[530]: network unreachable resolving './NS/IN': 2001:503:c27::2>
Oct 17 21:27:30 ip-172-31-30-87 named[530]: network unreachable resolving './NS/IN': 2001:500:a8::e#53
Oct 17 21:27:30 ip-172-31-30-87 named[530]: network unreachable resolving './NS/IN': 2001:500:2f::f#53
Oct 17 21:27:30 ip-172-31-30-87 named[530]: network unreachable resolving './NS/IN': 2001:dc3::35#53
Oct 17 21:27:30 ip-172-31-30-87 named[530]: network unreachable resolving './NS/IN': 2001:500:2::c#53
Oct 17 21:27:30 ip-172-31-30-87 named[530]: network unreachable resolving './NS/IN': 2001:7fd::1#53
Oct 17 21:27:30 ip-172-31-30-87 named[530]: managed-keys-zone: Key 20326 for zone . is now trusted (a>
Oct 17 21:27:30 ip-172-31-30-87 named[530]: managed-keys-zone: Key 38696 for zone . is now trusted (a>
lines 1-22/22 (END)

```

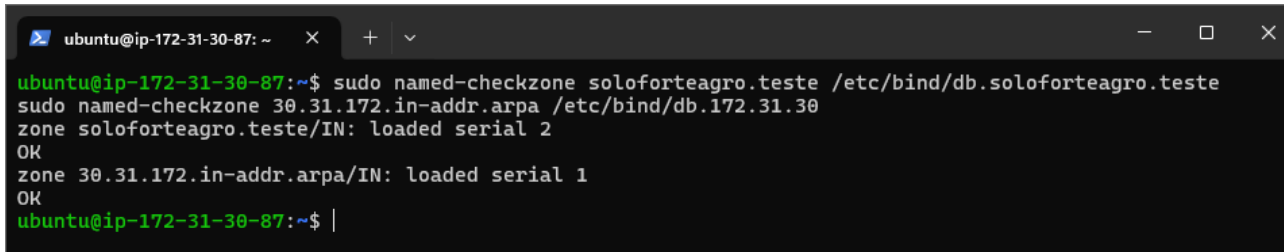
Este comando verifica o status do serviço `named` (processo principal do Bind9).

O objetivo é confirmar que o serviço foi iniciado corretamente e está ativado, sem erros.

Conclusão: o Bind9 está em execução e pronto para responder às consultas DNS.

O comando `named-checkzone` valida os arquivos de zona direta (`db.soloforteagro.teste`) e reversa.

Figura 60 - Verificação se as zonas foram carregadas corretamente:



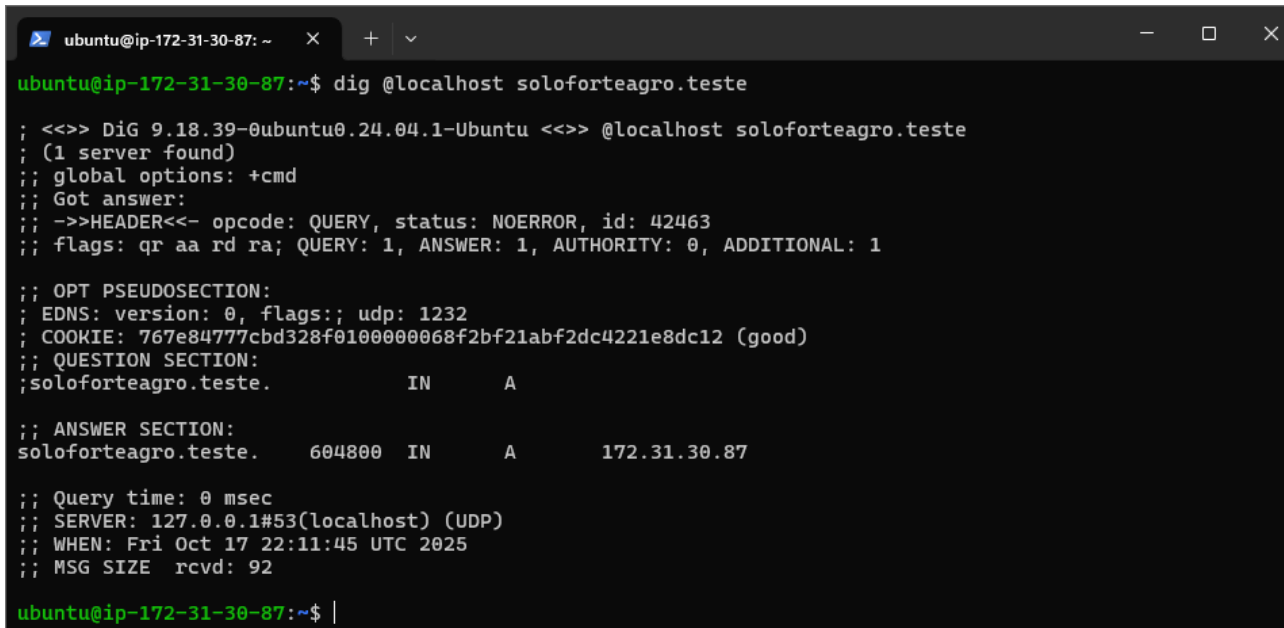
```
ubuntu@ip-172-31-30-87: ~  
ubuntu@ip-172-31-30-87:~$ sudo named-checkzone soloforteagro.teste /etc/bind/db.soloforteagro.teste  
sudo named-checkzone 30.31.172.in-addr.arpa /etc/bind/db.172.31.30  
zone soloforteagro.teste/IN: loaded serial 2  
OK  
zone 30.31.172.in-addr.arpa/IN: loaded serial 1  
OK  
ubuntu@ip-172-31-30-87:~$ |
```

Conclusão: as zonas foram carregadas sem erros e estão ativas no servidor DNS.

@localhost indica que o teste está sendo feito diretamente contra o servidor Bind9 local.

O primeiro comando testa o nome principal (`soloforteagro.teste`).

Figura 61 - Testando resolução local no próprio servidor:



```
ubuntu@ip-172-31-30-87: ~  
ubuntu@ip-172-31-30-87:~$ dig @localhost soloforteagro.teste  
  
; <<>> Dig 9.18.39-0ubuntu0.24.04.1-Ubuntu <<>> @localhost soloforteagro.teste  
; (1 server found)  
;; global options: +cmd  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42463  
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1  
  
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags:; udp: 1232  
; COOKIE: 767e84777cbd328f0100000068f2bf21abf2dc4221e8dc12 (good)  
;; QUESTION SECTION:  
;soloforteagro.teste.          IN      A  
  
;; ANSWER SECTION:  
soloforteagro.teste.        604800  IN      A          172.31.30.87  
  
;; Query time: 0 msec  
;; SERVER: 127.0.0.1#53(localhost) (UDP)  
;; WHEN: Fri Oct 17 22:11:45 UTC 2025  
;; MSG SIZE  rcvd: 92  
  
ubuntu@ip-172-31-30-87:~$ |
```

Figura 62 - O segundo comando testa o subdomínio www:

```
ubuntu@ip-172-31-30-87: ~  
ubuntu@ip-172-31-30-87:~$ dig @localhost www.soloforteagro.teste  
  
; <<>> DiG 9.18.39-0ubuntu0.24.04.1-Ubuntu <<>> @localhost www.soloforteagro.teste  
; (1 server found)  
;; global options: +cmd  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34027  
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1  
  
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags:; udp: 1232  
; COOKIE: 2c82bc5dae2e93b70100000068f2bf49d96eba6e3e6b4f91 (good)  
;; QUESTION SECTION:  
;www.soloforteagro.teste.      IN      A  
  
;; ANSWER SECTION:  
www.soloforteagro.teste. 604800 IN      A      172.31.30.87  
  
;; Query time: 0 msec  
;; SERVER: 127.0.0.1#53(localhost) (UDP)  
;; WHEN: Fri Oct 17 22:12:25 UTC 2025  
;; MSG SIZE rcvd: 96  
ubuntu@ip-172-31-30-87:~$ |
```

Figura 63 - O terceiro faz a resolução reversa (PTR), convertendo o IP de volta em nome de domínio:

```
ubuntu@ip-172-31-30-87: ~  
ubuntu@ip-172-31-30-87:~$ dig -x 172.31.30.87  
  
; <<>> DiG 9.18.39-0ubuntu0.24.04.1-Ubuntu <<>> -x 172.31.30.87  
;; global options: +cmd  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47651  
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1  
  
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags:; udp: 1232  
; COOKIE: 94a9db094d3bb92d0100000068f2bf6b6d3442f45555f11a (good)  
;; QUESTION SECTION:  
;87.30.31.172.in-addr.arpa.    IN      PTR  
  
;; ANSWER SECTION:  
87.30.31.172.in-addr.arpa. 604800 IN      PTR      soloforteagro.teste.  
  
;; Query time: 0 msec  
;; SERVER: 127.0.0.1#53(127.0.0.1) (UDP)  
;; WHEN: Fri Oct 17 22:12:59 UTC 2025  
;; MSG SIZE rcvd: 115  
ubuntu@ip-172-31-30-87:~$ |
```

Conclusão: o servidor DNS responde corretamente a consultas diretas e reversas internamente.

Agora teste com o Cliente DNS (172.31.26.100)

O objetivo dessa etapa é demonstrar que o servidor DNS configurado na instância anterior funciona corretamente para outras máquinas da rede.

Esse comando verifica se a instância cliente consegue alcançar o servidor DNS via rede interna (VPC).

Figura 64 - Testar conectividade entre cliente e servidor:

```
ubuntu@ip-172-31-26-100: ~  
ubuntu@ip-172-31-26-100:~$ ping 172.31.30.87  
PING 172.31.30.87 (172.31.30.87) 56(84) bytes of data.  
|
```

Conclusão esperada: as instâncias estão se comunicando dentro da mesma rede privada da AWS.

Testar resolução do domínio e subdomínio

O que faz: envia uma consulta direta ao servidor DNS (172.31.30.87);

Testa se o domínio e o subdomínio estão sendo resolvidos corretamente;

Conclusão esperada:

O cliente consegue resolver nomes definidos no Bind9 do servidor, confirmando que o DNS funciona externamente.

Figura 65 - Testar resolução do domínio e subdomínio

```
ubuntu@ip-172-31-26-100: ~  
ubuntu@ip-172-31-26-100:~$ dig @172.31.30.87 soloforteagro.teste  
  
; <<>> DiG 9.18.30-0ubuntu0.24.04.2-Ubuntu <<>> @172.31.30.87 soloforteagro.teste  
; (1 server found)  
;; global options: +cmd  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 16140  
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1  
  
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags:; udp: 1232  
; COOKIE: 134ca5b60a1d23540100000068f2c0fc4af9925a4980078b (good)  
;; QUESTION SECTION:  
;soloforteagro.teste. IN A  
  
;; ANSWER SECTION:  
soloforteagro.teste. 604800 IN A 172.31.30.87  
  
;; Query time: 0 msec  
;; SERVER: 172.31.30.87#53(172.31.30.87) (UDP)  
;; WHEN: Fri Oct 17 22:19:40 UTC 2025  
;; MSG SIZE rcvd: 92
```

Testar resolução reversa

Realiza uma consulta reversa (IP → nome), usando a zona in-addr.arpa configurada no servidor.

Conclusão esperada:

O Bind9 responde corretamente com o nome associado ao IP, provando que a **resolução reversa** está funcional.

Figura 66 - Testar resolução reversa:


```
ubuntu@ip-172-31-26-100: ~  
ubuntu@ip-172-31-26-100:~$ dig @172.31.30.87 -x 172.31.30.87  
  
; <<>> DiG 9.18.30-0ubuntu0.24.04.2-Ubuntu <<>> @172.31.30.87 -x 172.31.30.87  
; (1 server found)  
;; global options: +cmd  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18553  
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1  
  
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags:; udp: 1232  
; COOKIE: 488b86e3439c27630100000068f2c652af29b5b49fa17bdd (good)  
;; QUESTION SECTION:  
;87.30.31.172.in-addr.arpa.      IN      PTR  
  
;; ANSWER SECTION:  
87.30.31.172.in-addr.arpa. 604800 IN      PTR      soloforteagro.teste.  
  
;; Query time: 1 msec  
;; SERVER: 172.31.30.87#53(172.31.30.87) (UDP)  
;; WHEN: Fri Oct 17 22:42:26 UTC 2025  
;; MSG SIZE  rcvd: 115
```

2.2 Ambiente On-Premise(Virtualbox)

Nesta seção, são detalhados os serviços configurados em um ambiente local (on-premise), utilizando máquinas virtuais gerenciadas pelo Oracle VM VirtualBox. Este ambiente simula uma infraestrutura interna da empresa, permitindo a configuração e o teste de serviços essenciais de rede de forma isolada e controlada.

2.2.1 Serviço de DHCP

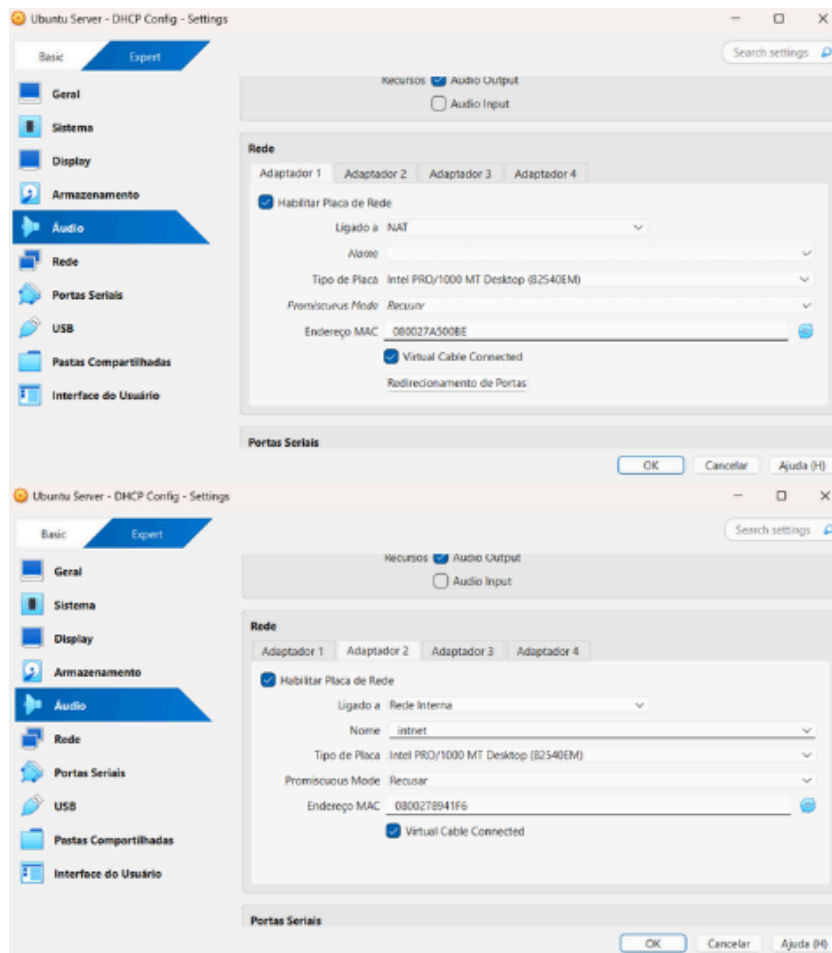
O protocolo DHCP (Dynamic Host Configuration Protocol, ou Protocolo de Configuração Dinâmica de Host) é um serviço fundamental em uma rede de computadores. Sua principal função é automatizar a atribuição de endereços IP e outras configurações de rede necessárias para que os dispositivos (clientes) possam se comunicar. Em vez de configurar manualmente cada máquina com um IP, máscara de sub-rede, gateway e servidores DNS, o servidor DHCP centraliza e gerencia essa distribuição, entregando as informações de forma automática a cada cliente que se conecta à rede.

Para este projeto, o serviço DHCP foi configurado em um servidor Ubuntu Server 22.04 e testado por um cliente Windows Server, ambos virtualizados no VirtualBox. O passo a passo da implementação é descrito a seguir.

Configuração das Interfaces de Rede (Servidor)

O primeiro passo consiste em garantir que a máquina virtual do servidor possua duas interfaces de rede com funções distintas. A primeira (Adaptador 1) foi configurada em modo NAT, permitindo que a máquina virtual acesse a internet para baixar pacotes e atualizações. A segunda (Adaptador 2) foi configurada em modo Rede Interna (intnet), criando uma rede local isolada onde o servidor DHCP atuará, conectando-se futuramente à máquina cliente.

Figura 67 - Configuração de rede da máquina servidora no VirtualBox



Configuração do Endereço IP Fixo (Servidor)

Para que o servidor DHCP possa operar, sua interface de rede interna precisa ter um endereço IP estático. A configuração foi realizada editando o arquivo de plano de rede (/etc/netplan/00-installer-config.yaml). Na interface de rede interna (enp0s8), a opção “dhcp4” foi definida como “false” e um endereço estático (192.168.99.1/24) foi atribuído.

Figura 68 - Arquivo de configuração Netplan com IP estático

```
network:
  version: 2
  ethernet:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      addresses: [192.168.99.1/24]
      gateway4: 10.0.2.15
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
guigs@dhcp-config:~$ _
```

Após salvar a configuração, o comando `sudo netplan apply` foi executado para aplicar as alterações.

Instalação do Servidor DHCP

Com a rede configurada, o pacote do servidor DHCP, `isc-dhcp-server`, foi instalado no Ubuntu Server utilizando o gerenciador de pacotes APT.

Figura 69 - Comando de instalação do pacote `isc-dhcp-server`

```
guigs@dhcp-config:~$ sudo apt install isc-dhcp-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
isc-dhcp-server is already the newest version (4.4.3-P1-4ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
guigs@dhcp-config:~$ _
```

Definição do Escopo DHCP

O arquivo principal de configuração do DHCP, `/etc/dhcp/dhcpd.conf`, foi editado para definir o escopo (range) de endereços que seriam distribuídos aos clientes. Foi definida uma sub-rede (subnet) 192.168.1.0 com máscara 255.255.255.0. Dentro dela, foram especificados:

range: O intervalo de IPs a serem distribuídos (de 192.168.1.51 a 192.168.1.100).

option routers: O gateway padrão para os clientes (o próprio servidor, 192.168.1.1).

option domain-name-servers: Os servidores DNS a serem utilizados pelos clientes (8.8.8.8 e 1.1.1.1).

Figura 70 - Configuração da sub-rede e escopo no arquivo `dhcpd.conf`

```
#shared-network 224-29 {
#   subnet 10.17.224.0 netmask 255.255.255.0 {
#       option routers rtr-224.example.org;
#   }
#   subnet 10.0.29.0 netmask 255.255.255.0 {
#       option routers rtr-29.example.org;
#   }
#   pool {
#       allow members of "foo";
#       range 10.17.224.10 10.17.224.250;
#   }
#   pool {
#       deny members of "foo";
#       range 10.0.29.10 10.0.29.230;
#   }
#}
subnet 192.168.99.0 netmask 255.255.255.0 {
    range 192.168.99.51 192.168.99.100;
    option routers 192.168.99.1;
    option domain-name-servers 8.8.8.8, 1.1.1.1;
    option domain-name "exemplo.org";
}
guigs@dhcp-config:~$ _
```

Definição da Interface de Atuação

Para garantir que o servidor DHCP operasse apenas na rede correta, o arquivo `/etc/default/isc-dhcp-server` foi editado, especificando no parâmetro `INTERFACESv4` o nome da interface de rede interna (`enp0s8`).

Figura 71 - Definição da interface de atuação do serviço DHCP

```
guigs@dhcp-config:~$ sudo cat /etc/default/isc-dhcp-server
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

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

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

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="enp0s8"
INTERFACESv6=""
guigs@dhcp-config:~$
```

Reinicialização e Verificação do Serviço

Após todas as configurações, o serviço foi reiniciado com o comando `sudo service isc-dhcp-server restart`. Em seguida, o status do serviço foi verificado com `sudo service isc-dhcp-server status` para confirmar que ele estava ativo e rodando sem erros.

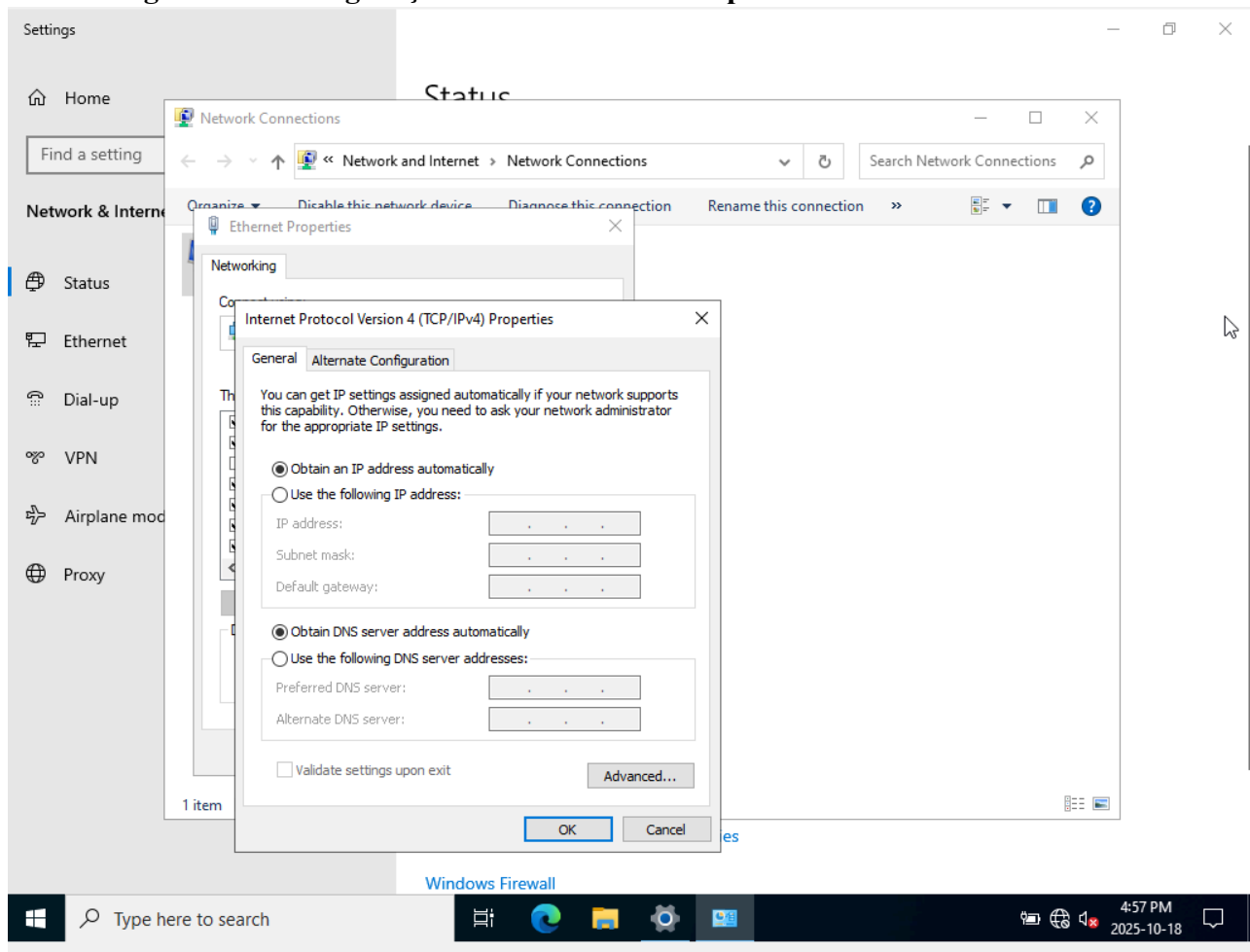
Figura 72 - Verificação do status do serviço DHCP como ativo (running)

```
guigs@dhcp-config:~$ sudo service isc-dhcp-server status
• isc-dhcp-server.service - ISC DHCP IPv4 server
   Loaded: loaded (/usr/lib/systemd/system/isc-dhcp-server.service; enabled; preset: enabled)
   Active: active (running) since Sat 2025-10-18 19:09:50 UTC; 43min ago
     Docs: man:dhcpd(8)
  Main PID: 868 (dhcpd)
    Tasks: 1 (limit: 2268)
   Memory: 3.7M (peak: 4.0M)
      CPU: 16ms
  CGroup: /system.slice/isc-dhcp-server.service
          └─868 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/dhcpd.pid -cf /etc/dhcp/dhcpd.conf enp0s8

Oct 18 19:09:50 dhcp-config sh[868]: PID file: /run/dhcp-server/dhcpd.pid
Oct 18 19:09:50 dhcp-config dhcpd[868]: Wrote 0 leases to leases file.
Oct 18 19:09:50 dhcp-config sh[868]: Wrote 0 leases to leases file.
Oct 18 19:09:50 dhcp-config dhcpd[868]: Listening on LPF/enp0s8/08:00:27:89:41:f6/192.168.99.0/24
Oct 18 19:09:50 dhcp-config sh[868]: Listening on LPF/enp0s8/08:00:27:89:41:f6/192.168.99.0/24
Oct 18 19:09:50 dhcp-config dhcpd[868]: Sending on LPF/enp0s8/08:00:27:89:41:f6/192.168.99.0/24
Oct 18 19:09:50 dhcp-config sh[868]: Sending on LPF/enp0s8/08:00:27:89:41:f6/192.168.99.0/24
Oct 18 19:09:50 dhcp-config dhcpd[868]: Sending on Socket/fallback/fallback-net
Oct 18 19:09:50 dhcp-config sh[868]: Sending on Socket/fallback/fallback-net
Oct 18 19:09:50 dhcp-config dhcpd[868]: Server starting service.
guigs@dhcp-config:~$ _
```

Teste com a Máquina Cliente

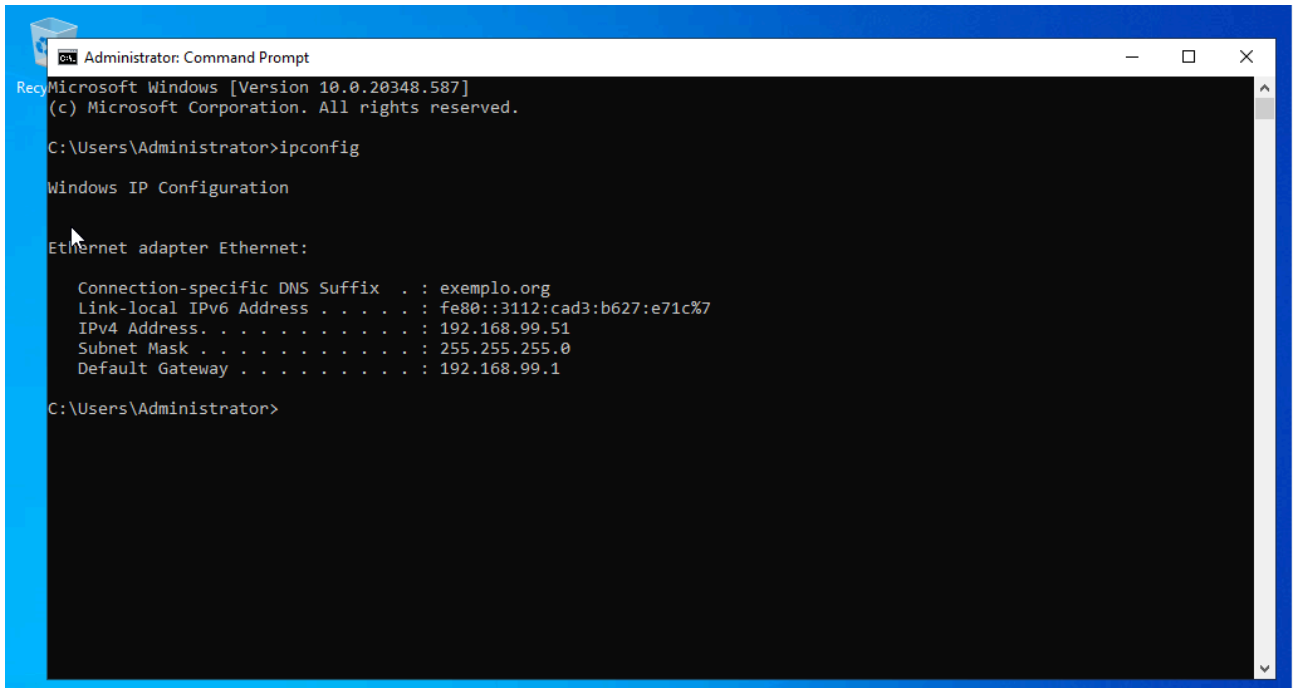
Para validar o funcionamento do servidor, uma máquina virtual com Windows Server foi configurada para atuar como cliente. Sua interface de rede foi definida para o mesmo modo Rede Interna (intnet) do servidor. Nas configurações do adaptador de rede IPv4, a máquina foi ajustada para obter um endereço IP e endereços de servidor DNS automaticamente.

Figura 73 - Configuração do cliente Windows para obter IP automaticamente

Validação Final

No prompt de comando do cliente Windows, o comando `ipconfig /all` foi executado. A saída confirmou o sucesso da operação: o cliente recebeu o endereço IP 192.168.1.51, o primeiro disponível no escopo definido, além do gateway e dos servidores DNS corretos, validando que o servidor DHCP está funcionando como esperado.

Figura 74 - Saída do comando `ipconfig /all` no cliente, mostrando o IP recebido

A screenshot of a Windows Command Prompt window titled "Administrator: Command Prompt". The window shows the output of the 'ipconfig' command. The text displayed is: "Microsoft Windows [Version 10.0.20348.587] (c) Microsoft Corporation. All rights reserved. C:\Users\Administrator>ipconfig Windows IP Configuration Ethernet adapter Ethernet: Connection-specific DNS Suffix . : exemplo.org Link-local IPv6 Address : fe80::3112:cad3:b627:e71c%7 IPv4 Address. : 192.168.99.51 Subnet Mask : 255.255.255.0 Default Gateway : 192.168.99.1 C:\Users\Administrator>".

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.20348.587]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : exemplo.org
    Link-local IPv6 Address . . . . . : fe80::3112:cad3:b627:e71c%7
    IPv4 Address. . . . . : 192.168.99.51
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.99.1

C:\Users\Administrator>
```

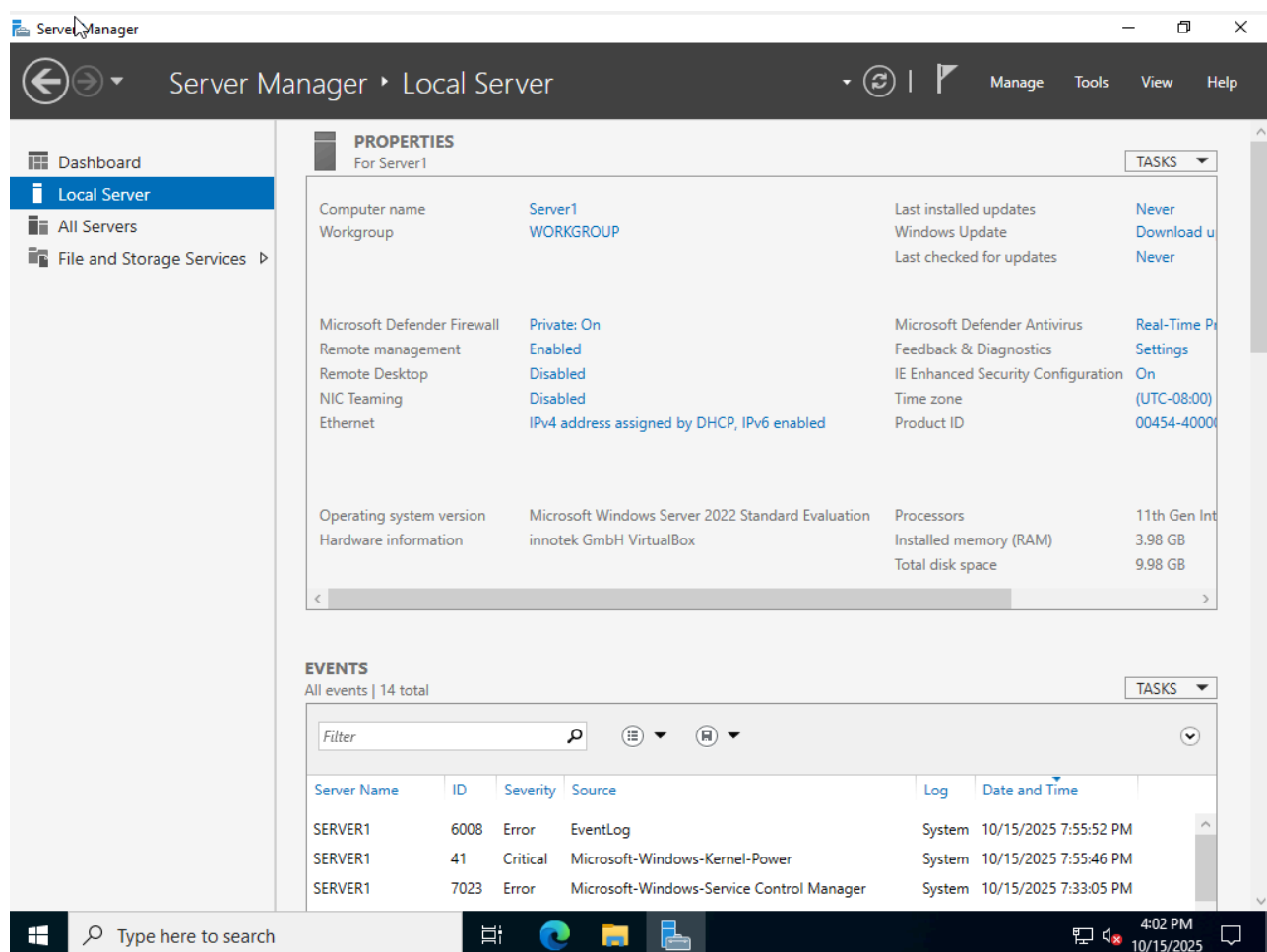
2.2.2 Configuração de AD e GPO

As configurações AD (Active Directory) e GPO (Group Policy Object) são ferramentas disponíveis para o gerenciamento da segurança em uma rede. Elas trabalham em conjunto para centralizar a configuração e segurança de usuários e máquinas. O AD organiza a rede em domínios e unidades organizacionais, enquanto os GPOs contêm configurações que podem ser aplicadas aos objetos contidos no AD através do Console de Gerenciamento de Políticas de Grupo (GPMC).

Para este projeto, o AD e o GPO foram configurados em um servidor Windows virtualizado no VirtualBox. O passo a passo da implementação é descrito a seguir.

Promovendo o servidor a controlador de domínio

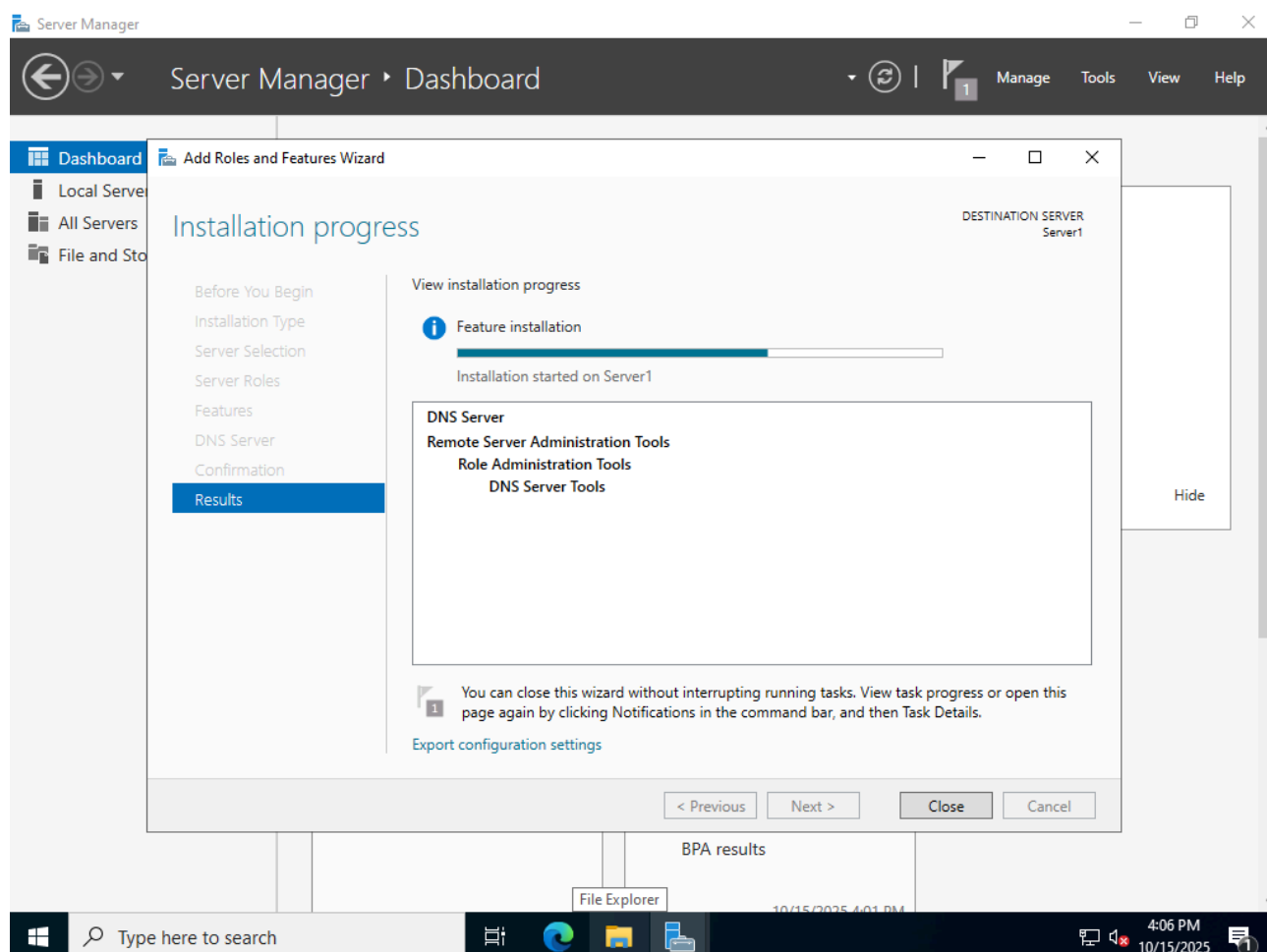
Para realizar esta promoção é necessário renomear o servidor que será promovido, abrindo o gerenciador de servidor e trocando o nome da máquina, junto a troca de nome do computador deve-se permitir as conexões remotas na aba “Remoto” nas propriedades do sistema onde também o nome foi modificado.

Figura 75 - Configuração do Server Local para controlador de domínio

Instalando o DNS e o AD

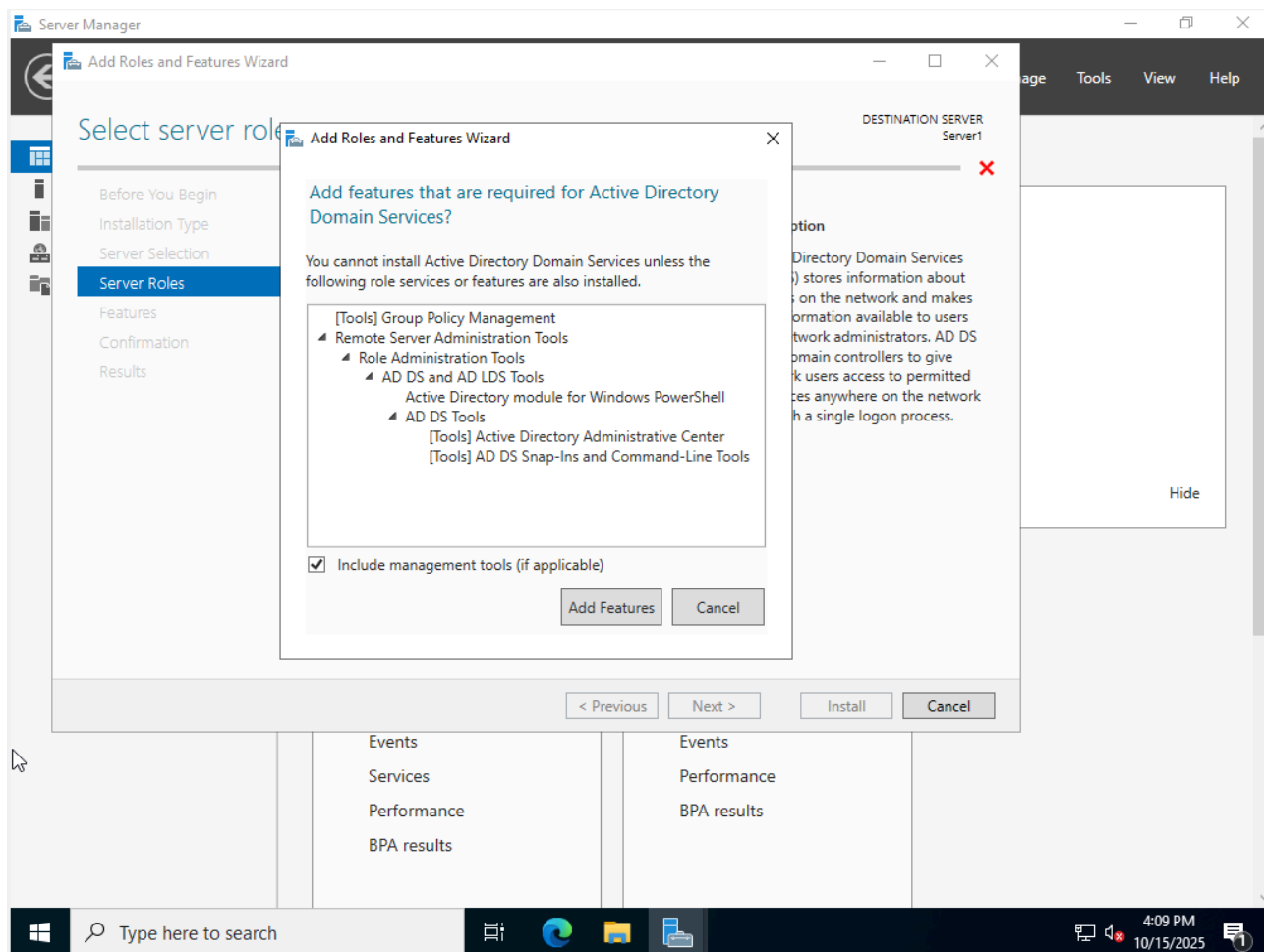
Após renomear o servidor inicia-se o processo de instalação do DNS e do AD. A aba Painel foi selecionada e por ela foi iniciado o processo de adição dos recursos AD e DNS clicando em adicionar funções e recursos, escolhendo a opção “instalação baseada em função ou recurso”, selecionando o servidor escolhido e marcando a opção “Servidor DNS”. Após isso inicia-se a instalação.

Figura 76 - instalando o DNS



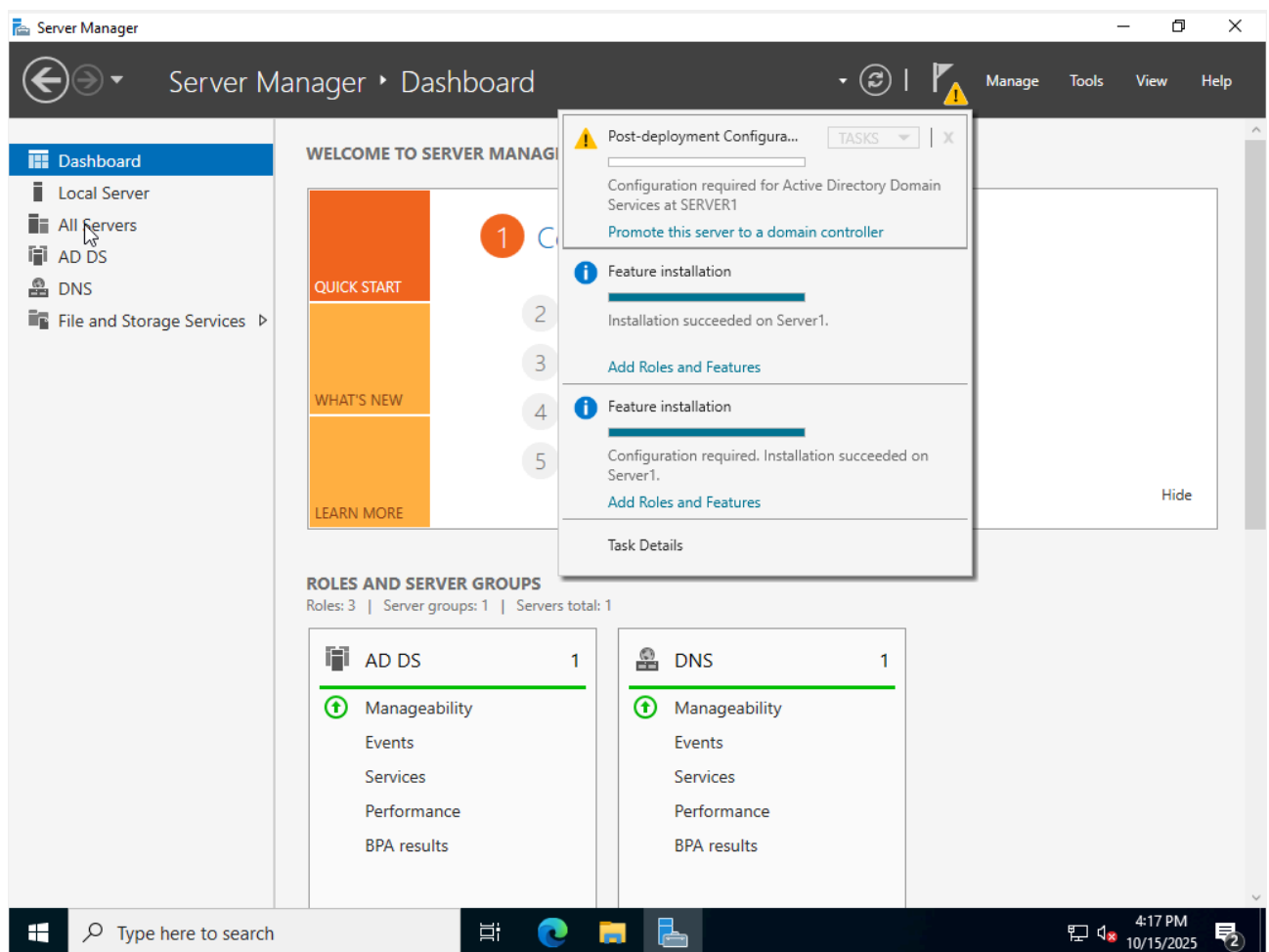
Agora é necessário repetir o processo porém nas funções do servidor deve-se selecionar a opção “Serviços de domínio Active Directory”

Figura 77 - instalando o AD



Promovendo o servidor para Controlador de domínio

Com as configurações acima instaladas deve-se voltar ao painel e selecionar o símbolo de aviso ao lado da bandeira para iniciar a promoção do servidor a controlador de domínio.

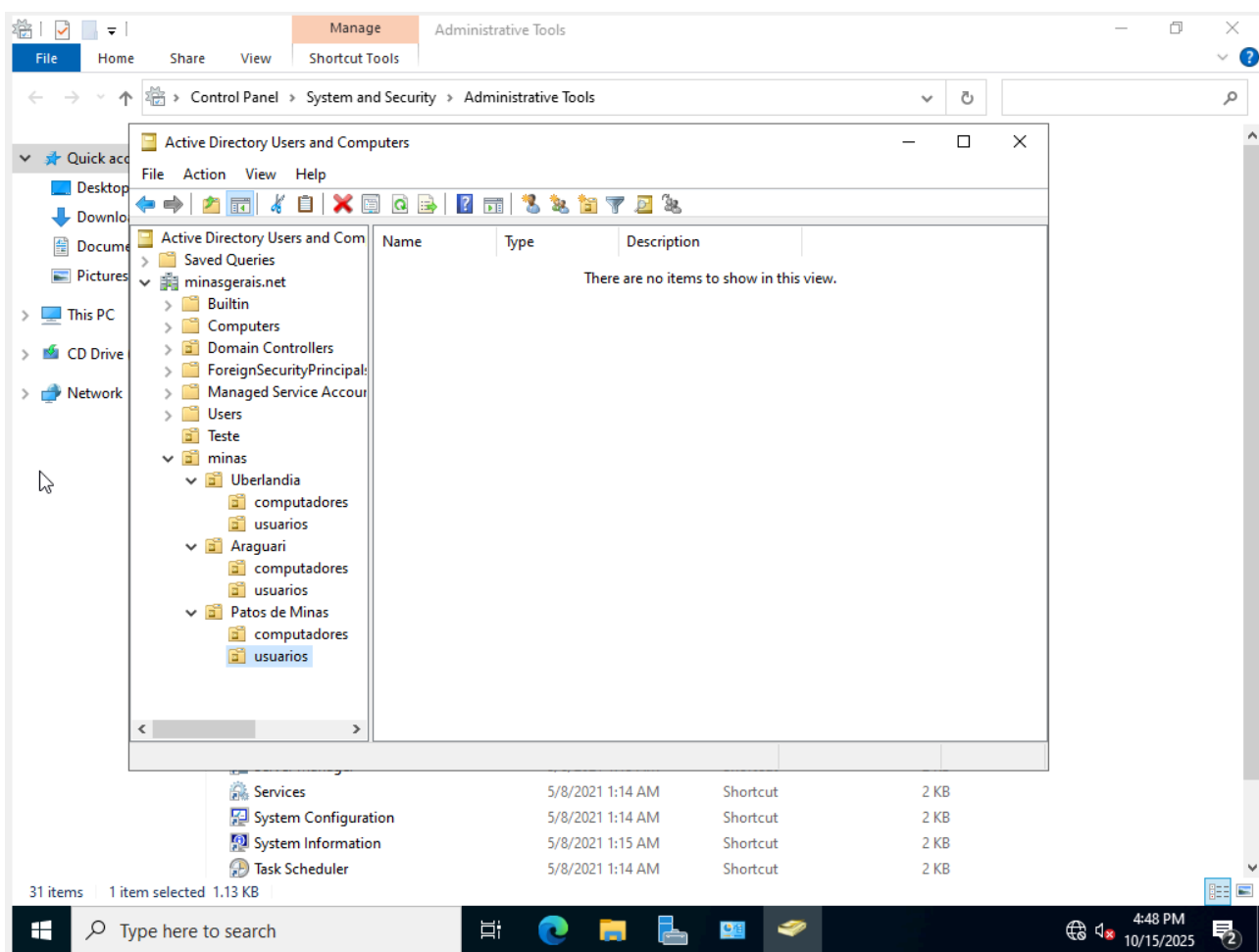
Figura 78- Promovendo servidor

Criando a raiz da árvore

A Microsoft projetou o AD para usar o DNS como sistema de nomeação, e DNS é hierárquico por natureza. O primeiro domínio do AD é chamado de raiz da árvore. Abaixo será mostrado como iniciar a configuração da raiz para o início da floresta. Na adição de uma nova floresta criaremos o domínio raiz nomeado de “minasgerais.net”. Prosseguimos com o passo a passo até verificar os requisitos necessários para o funcionamento, instalá-los e reiniciar o servidor.

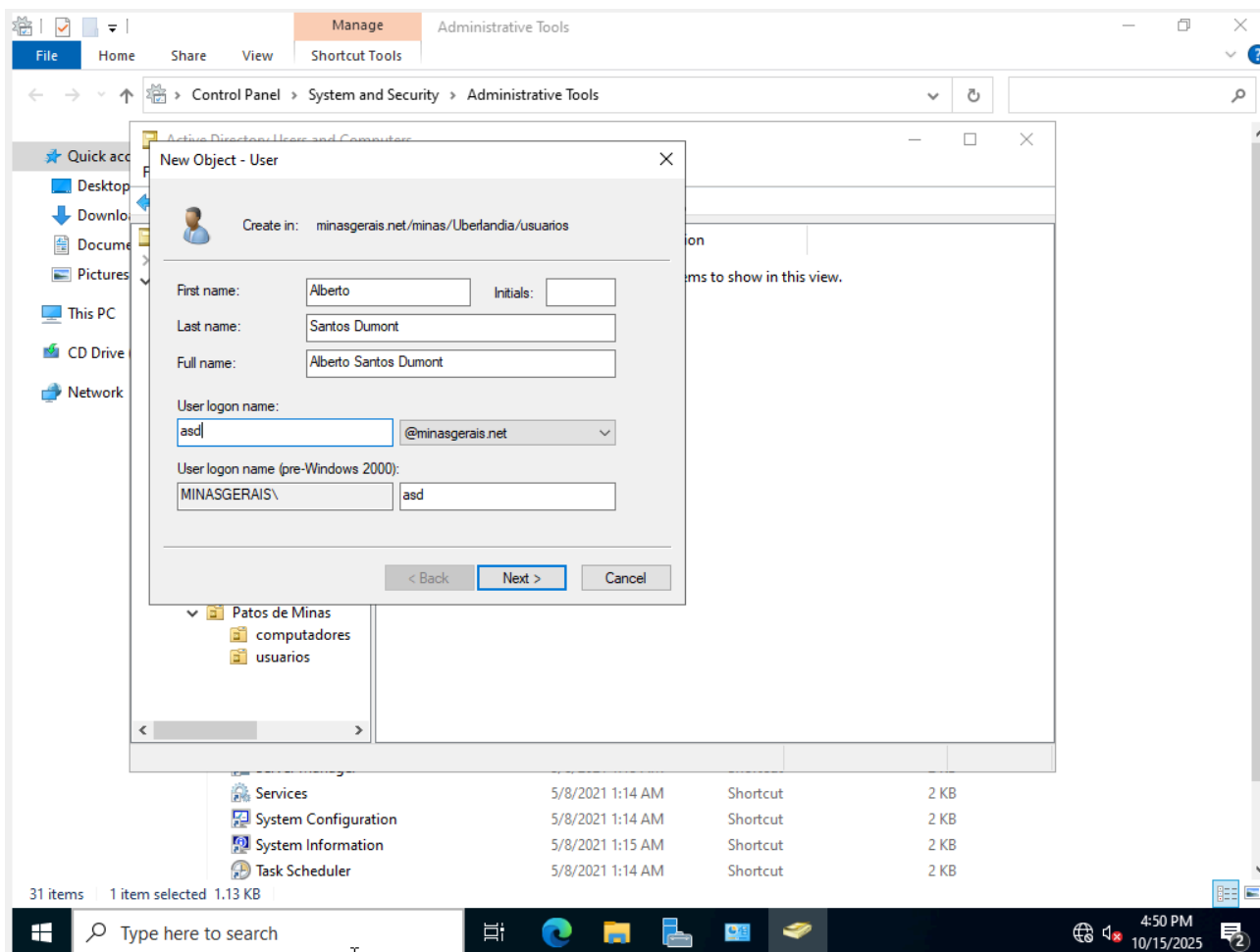
Após essa configuração inicial deve-se abrir as ferramentas administrativas, e selecionar “usuários e computadores do Active Directory” onde deve-se selecionar o domínio “minasgerais.net” e criar unidades organizacionais para cada filial onde serão colocados os computadores e os usuários.

Figura 79- Criando as unidades de organização



Criando o novo usuário

Dentro de uma unidade organizacional foi criado o primeiro usuário com o nome fictício de Alberto Santos Dumont com a senha “puc@1958”.

Figura 80 - Criando o usuário

Criando as políticas de grupo

Após a criação de usuários é necessário criar as políticas de grupo que restringem os acessos de cada usuário. Para isso é necessário voltar às ferramentas administrativas, e acessar as opções de Gerenciamento de Políticas de Grupo. A GPO foi nomeada de pgub. Nesta política foram configuradas as opções de filtragem, depois foram habilitadas as políticas recomendadas e por fim o relatório da política é exibido ao clicar nas opções.

Figura 81 - Configuração das opções de filtro

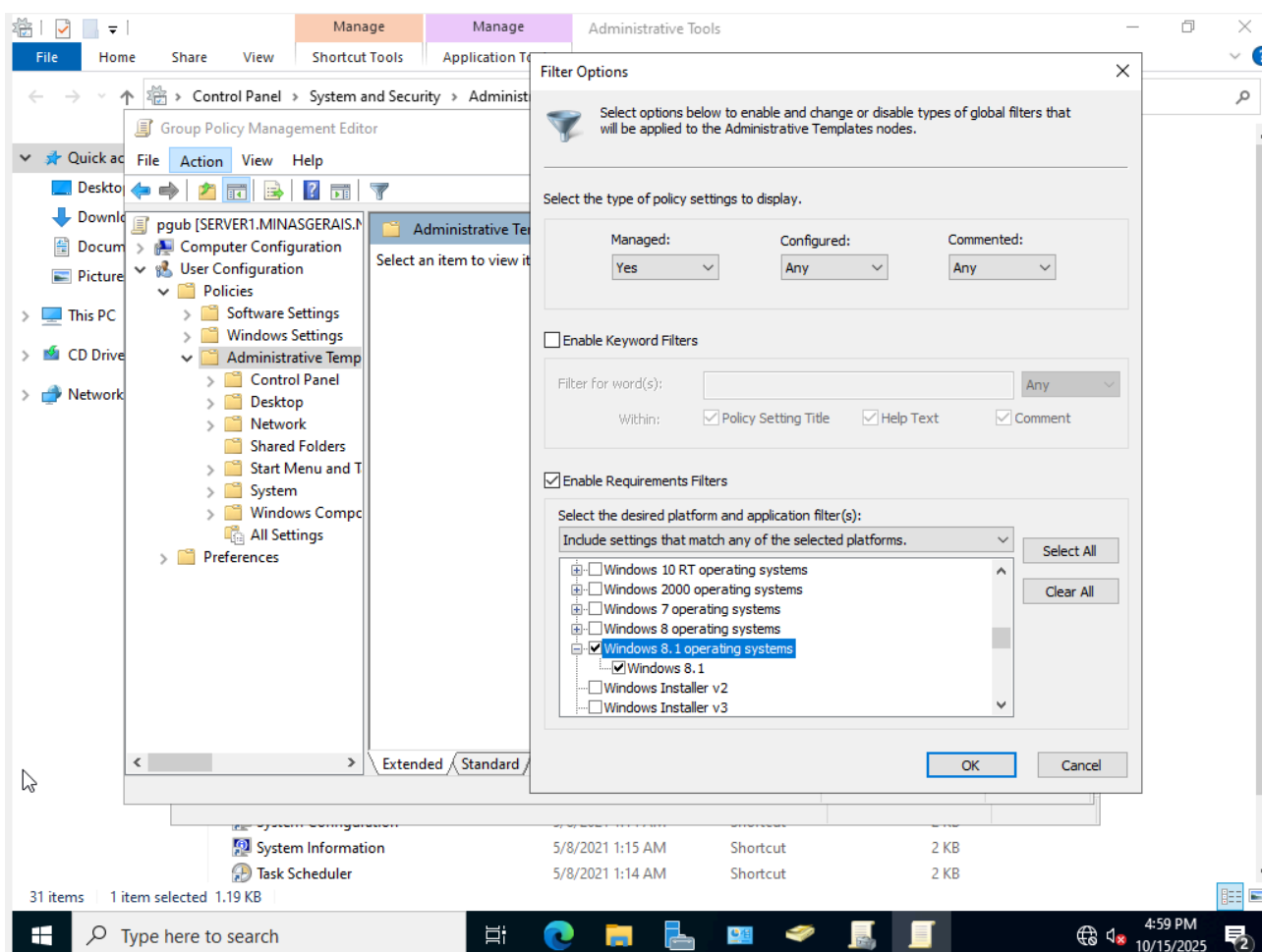


Figura 82 - Políticas recomendadas

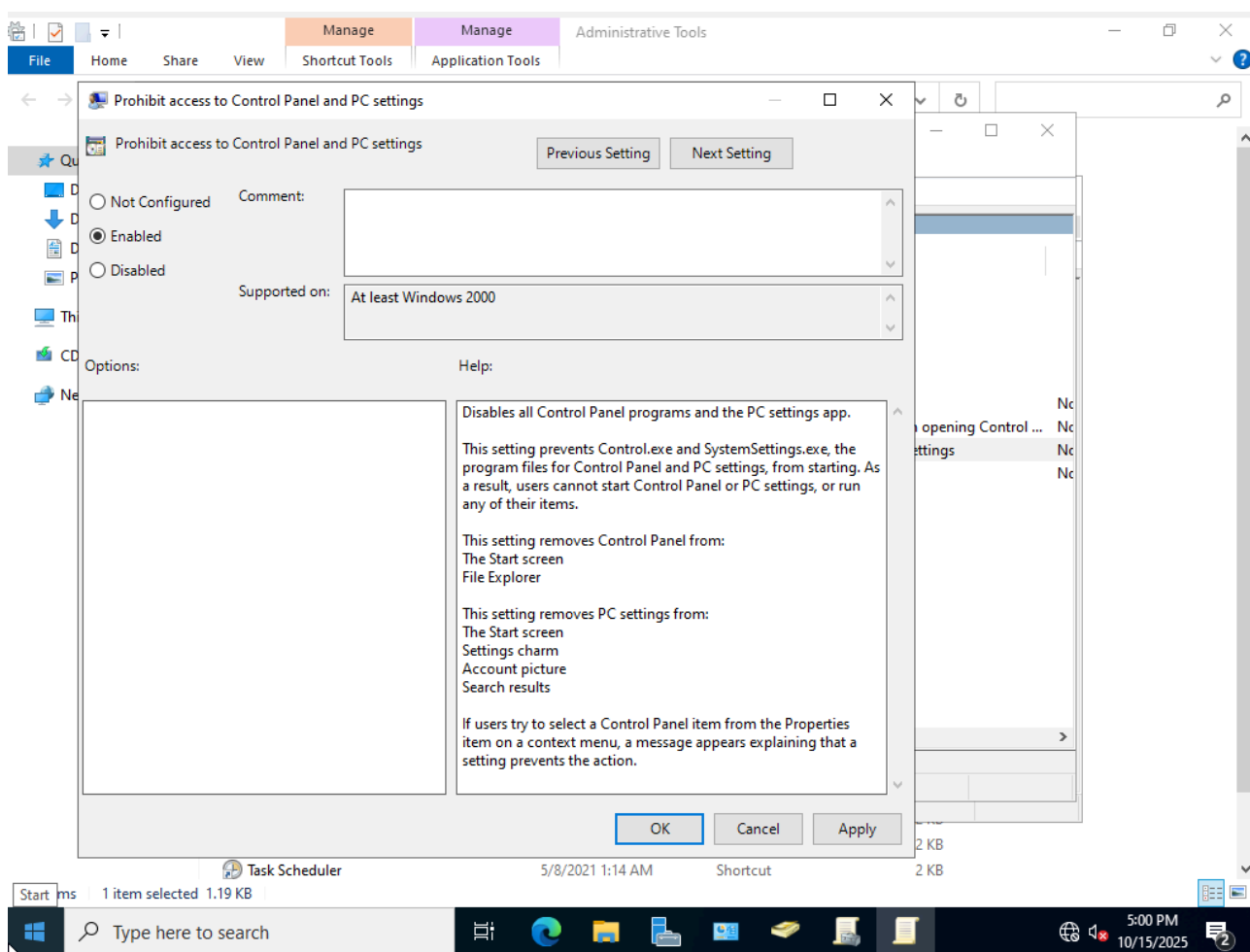
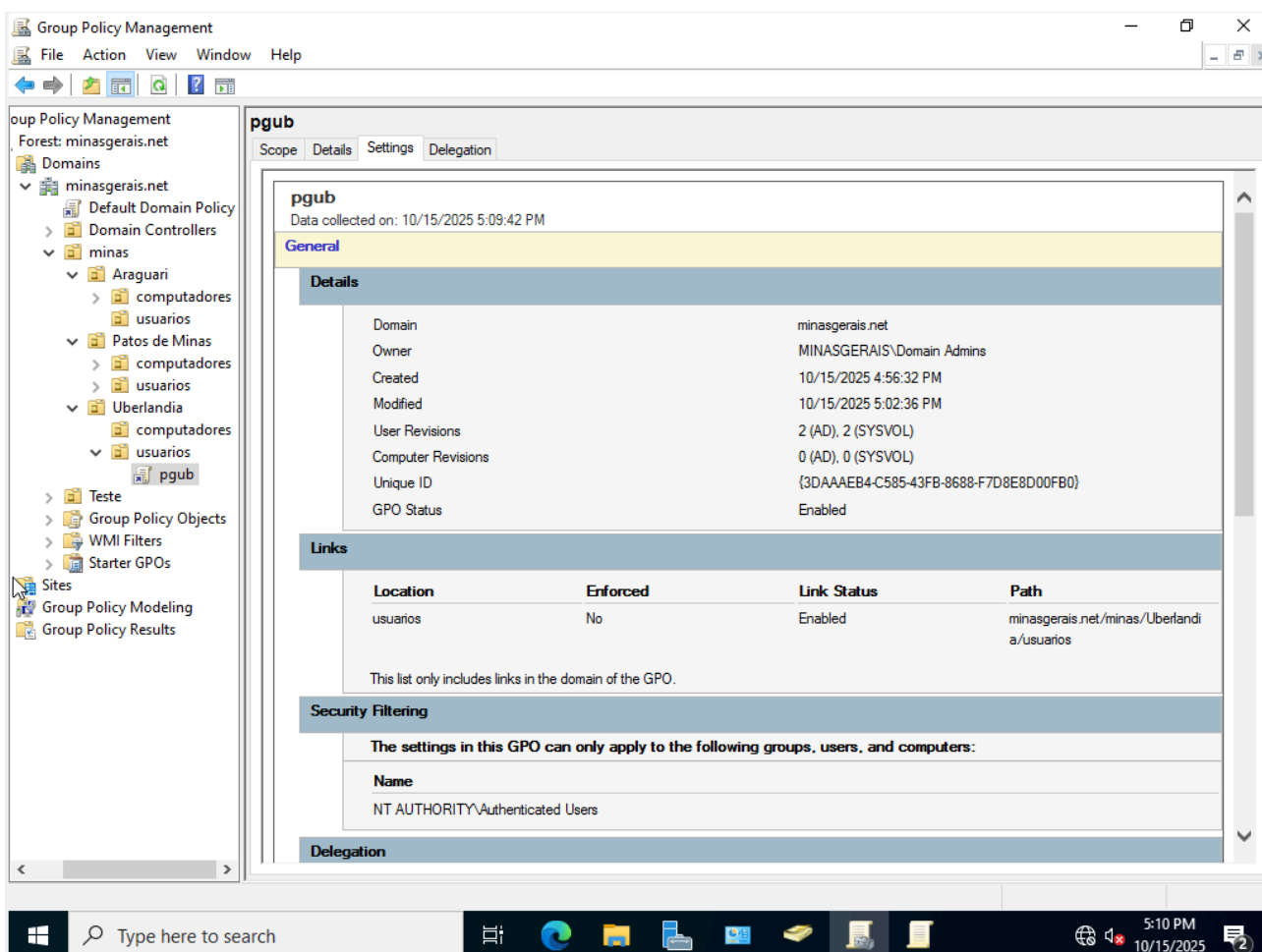


Figura 83 - Relatório da Política



2.3 Tópico intermediário (Ex.: Inclusão informacional)

O mesmo procedimento descrito na Seção 2.1 deve ser aplicado a cada parágrafo do capítulo do referencial teórico.

2.4 Tópico específico (Ex.: Inclusão social)

O mesmo procedimento descrito na Seção 2.1 deve ser aplicado a cada parágrafo do capítulo do referencial teórico.

2.5 Elementos flutuantes

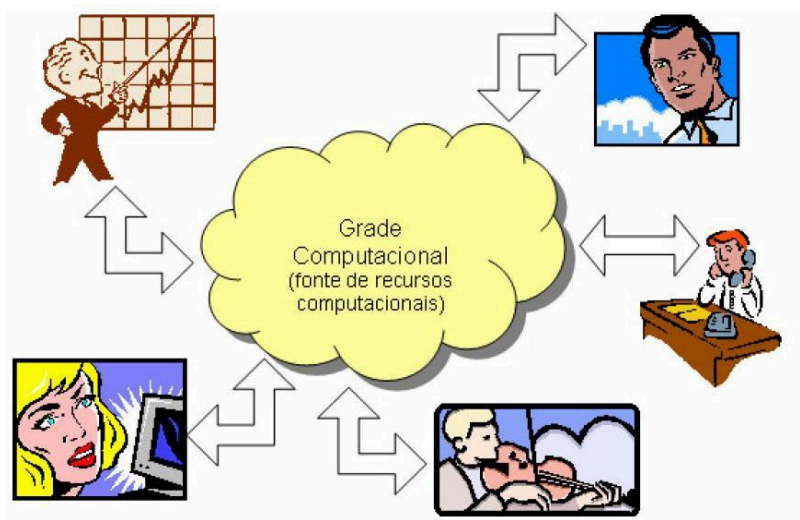
São elementos inseridos no texto como imagens, tabelas, algoritmos, etc. De acordo com as normas ABNT, há a necessidade de se observar que todos os elementos flutuantes

inseridos devem ter a formatação básica:

- a) título centralizado localizado na parte superior;
- b) fonte em tamanho 10 na parte inferior;
- c) se o elemento for de autoria própria, siga o exemplo do Quadro 1; caso contrário, faça a citação da referência conforme o exemplo da Figura 40. Para elementos criados a partir de outros autores, usar a expressão “Adaptado de ...”
- d) devem ser inseridos o mais próximos do texto que os referenciam.

2.5.1 Inserções de ilustrações

As ilustrações devem ser inseridas seguindo o exemplo da Figura 40. Nos casos de telas de software, estas também devem ser inseridas como figuras, e referenciadas no texto. Além disso, é necessário que seja citada no texto a empresa desenvolvedora, quando aplicável.

Figura 40 – Uma Grade Computacional como fonte transparente

Fonte: (Góes et al., 2005)

2.5.2 Tabelas

As tabelas devem ser abertas nas laterais, com espaços verticais separando as colunas e sem espaços horizontais, exceto na separação do cabeçalho. Um exemplo é a Tabela 1:

Tabela 1 – Exemplo de uma tabela

| Imagem | transferência | tempo |
|---------------|----------------------|--------------|
| estação 1 | 7,72 MB/s | 1:22:18 |
| estação 2 | 7,72 MB/s | 1:22:17 |
| estação 3 | 7,59 MB/s | 1:24:25 |
| estação 4 | 7,53 MB/s | 1:43:27 |
| estação 5 | 6,14 MB/s | 1:24:41 |
| estação 6 | 7,50 MB/s | 1:23:53 |
| estação 7 | 7,58 MB/s | 1:24:02 |
| estação 8 | 7,8 MB/s | 1:29:06 |
| estação 9 | 7,9 MB/s | 1:30:05 |
| estação 10 | 8,0 MB/s | 1:32:03 |

Fonte: Cordeiro (2010)

2.5.3 Quadros

Os quadros diferem das tabelas por apresentarem dados textuais. Esses dados podem ser esquemáticos, comparativos ou descritivos.

Quadro 1 – Bandas/Artistas de Rock e outros

| Bandas ou Artistas de Rock e outros | | | |
|--|--------------|--------------------|-----------------------|
| Progressivo | Pink Floyd | Jethro Tull | Yesterday |
| Metal | Metallica | Iron Maidam | Black Sabath |
| Arena Rock | Led Zeppelin | The Rolling Stones | Beatles |
| Punk | Ramones | Black Flag | NOFX |
| Nacional | Ira | Engenheiros | Vinil |
| S.J.E. | Apolo XI | Invasão 7 | Por do Sol |
| Grunge | Nirvana | Pear Jam | Alice in Chains |
| Rock Folk | Bod Dylan | The Byrds | The Mamas & the Papas |
| Blues | B.B. King | Albert Colins | Mady Wathers |
| New Wave | The Police | The Pretenders, | Duran Duran |
| Rock Folk | Bod Dylan | The Byrds | The Mamas & the Papas |
| Rock alternativo | R.E.M. | Hüsker Dü | Big Black |

Fonte: Elaborado pelo autor

2.5.4 Inserção de algoritmos

Para inserir um algoritmo, utilizar o exemplo do Algoritmo 1. Todos os algoritmos devem ser inseridos assim como é feito para figuras, tabelas, ou seja, devem ser indicados por nome e fonte.

Algoritmo 1 - CAC RD Neural

Algoritmo 1: CAC-RD Neural

```
1: Entrada: Requisição da chamada
2: Saída: Aceitação ou bloqueio da solicitação
3: Preenche o vetor de attributes.size + 1 atributos com os valores dos atributos, sendo a primeira
   posição do vetor preenchida com o valor 1
4: hidden_layer_size = attributes.size * 2 + 1;
5: for i = 1 to attributes.size + 1 do
6:   normalizar(Entrada,)
7: end for
8: double[]net = newdouble[hidden_layer_size];
9: net = hidden_layer_weights * attributes;
10: for i = 0 to hidden_layer_size do
11:   net[i] = 1.0/(1.0 + exp((-1.0) * net[i]));
12: end for
13: double[]ipV ector = newdouble[hidden_layer_size + 1];
14: ipV ector[0] = 1.0;
15: for i = 1 to hidden_layer_size + 1 do
16:   ipV ector[i] = net[i - 1];
17: end for
18: output = output_layer_weights * ipV ector;
19: output = desnormalizar(Saída)
20: net_update (requisition);
21: Retorna output; FIM
```

Fonte: Ribeiro (2010).

3 TRABALHOS RELACIONADOS (1 PÁGINA)

A seção de Trabalhos Relacionados deve apresentar um pequeno resumo (um parágrafo) de cada artigo, monografia ou demais trabalhos que tenham feito algo parecido com o seu trabalho. Em seguida, devem ser destacadas as diferenças e semelhanças entre o seu trabalho e o relacionado. Devem ser apresentados no mínimo 3 trabalhos relacionados. Os trabalhos relacionados não devem ter sido usados como referências no capítulo anterior.

Lembre-se de usar corretamente a formatação para citações. As citações podem ser classificadas como livres, diretas ou citação de citação, esta última não abordada neste documento.

3.1 Citação livre ou indireta

Quando se reproduzir ideias, sem transcrever as palavras do autor, a indicação da página é opcional. Exemplos desse tipo de citação:

a) citação com um autor (Knuth, 1968).

b) citação de artigos em revistas com dois autores (Prenner; Robbes, 2022).³

³Quando citar um artigo, inclua o DOI (Identificador de Objeto Digital) sempre que estiver disponível.

- c) trabalho em congresso com três autores (Vasconcelos; Cardoso; Fernandes, 2017).
- d) trabalhos com mais de três autores (Góes et al., 2005).
- e) citação de dois autores de uma vez em duas obras distintas (Gil, 2022; Gropp, 2003).

3.2 Citação direta ou textual

Transcrição literal de textos de outros autores. Nesse caso, deverão ser especificadas as páginas consultadas.

3.2.1 Textual curtas

Quando curtas (até 3 linhas) serão inseridas na sequência normal do texto, entre aspas duplas com a mesma formatação.

3.2.2 Textual longas

Citações longas (mais de 3 linhas) deverão constituir um parágrafo independente, recuado a 4 cm da margem esquerda, com letra tamanho 10 e digitado em espaço simples, sem aspas.

Hegel chama trabalho à forma específica da satisfação das necessidades, que distingue da natureza o espírito existente. Assim como a linguagem infringe a imposição da intuição e ordena o caos das múltiplas sensações em coisas identificáveis, assim o trabalho infringe a imposição do desejo imediato e suspende, por assim dizer, o processo de satisfação das necessidades. (Haber- mas, 1997, 25).

3.2.3 Textual de outros idiomas (tradução)

Quando a citação estiver em outro idioma e for traduzida, indique após a chamada da citação a expressão tradução nossa ou tradução própria, entre parênteses. Obs: Em nota de rodapé informe, se desejar, a citação direta no idioma consultado.

Um cluster é um computador paralelo construído de componentes e processos de software (tal como sistema de software). Um cluster é formado de nós, cada um contendo um ou mais processadores, memória que é compartilhada por todos os processadores do nodo (somente eles), e dispositivos periféricos adicionais (tais como discos), conectados pela rede e que permitem tráfego de dados entre os nós... (Gropp, 2003, p. 10, tradução

nossa)⁴.

⁴... a cluster is a parallel computer that is constructed of commodity components and runs (as its system software)

3.3 Exemplos de citações

Seguem alguns exemplos de citações mais utilizadas e/ou que geram algumas dúvidas. É válido observar que não serão citadas aqui todas as possibilidades de citações. Sendo assim é de extrema relevância que se consulte o documento no site da Biblioteca (Pontifícia Universidade Católica de Minas Gerais (2023)) para maiores esclarecimentos acerca de citações.

3.3.1 Citação de monografia, dissertação e tese

Um exemplo de citação de monografia de curso de graduação ou especialização pode ser visto em Cordeiro (2010). Já para dissertação de mestrado veja Ribeiro (2010). E para o doutorado a citação é feita da seguinte forma: Góes (2012).

3.3.2 Livros e partes de livros

Exemplo de capítulo de livro fica conforme esse exemplo: (Góes et al., 2005).

Para livros citados no corpo do texto e com duas citações juntas, ver os exemplos: Knuth (1968), Gropp (2003).

commodity software. A cluster is made of nodes, each containing one or more processors, memory that is shared by all of the processors in (and only on) the node, and additional peripheral devices (such as disks), connected by network that allows data to move between the nodes.

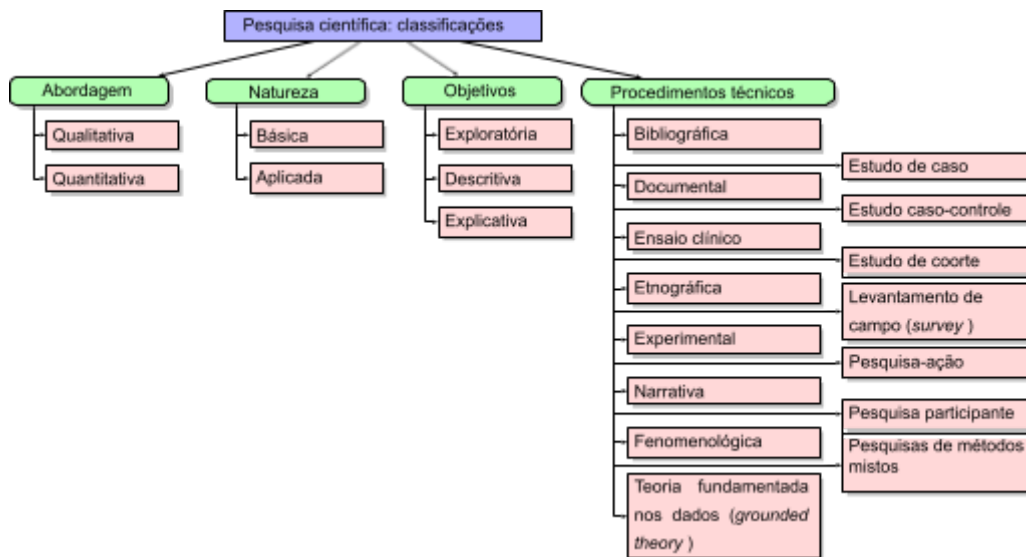
4 METODOLOGIA (1 - 2 PÁGINAS)

A Metodologia descreve o tipo e as etapas da pesquisa, estabelecendo um plano detalhado para a coleta e análise de dados. Além disso, uma metodologia bem definida permite a replicação do estudo, contribui para a transparência dos procedimentos e facilita a interpretação e aplicação dos achados.

4.1 Classificação da pesquisa

As pesquisas podem ser classificadas de diversas formas, como pode ser visualizado na Figura 41. Para mais informações sobre tais classificações, consulte materiais como Gil (2022), Wazlawick (2021).

Figura 41 – Classificação das pesquisas científicas Gil (2022)



Fonte: Elaborada pelo autor

É importante esclarecer em quais tipos de pesquisa o seu estudo se enquadra. Ex: Este trabalho apresenta uma pesquisa de natureza qualitativa, que, a partir de dados coletados por meio de entrevistas e questionários, os analisa, classifica e interpreta.

4.2 Etapas da pesquisa

Nesta subseção, deve-se descrever as etapas para a realização da pesquisa. Como foi identificada e selecionada a amostra? Como foi feita a elaboração e aplicação dos questionários da pesquisa? Cada etapa deve ser detalhada.

Ex: Esta pesquisa é dividida nas seguintes etapas:

- a) levantamento bibliográfico;
- b) elaboração de questionários;
- c) estruturação e definição das entrevistas;
- d) aplicação dos questionários e entrevistas;
- e) apresentação, interpretação e análise dos resultados obtidos.

Na elaboração dos questionários, foram levantados os principais fatores que podem influenciar na formação dos adolescentes e as ferramentas possivelmente utilizadas. Para cada um desses fatores, foram elaboradas questões de múltipla escolha e abertas. O questionário foi respondido por 30 pessoas, sendo 10 pessoas de cada classe social (baixa, média e alta). A determinação de classe social foi feita utilizando-se o Critério Brasil de Classificação Social.

5 RESULTADOS (3 - 5 PÁGINAS)

Nos Resultados você deve apresentar os gráficos com os resultados obtidos e também analisá-los. Note que apresentar é apenas descrever o que pode ser visto no gráfico, enquanto que analisar significa explicar o porquê dos resultados apresentados (essa explicação deve ser sustentada por dados da pesquisa. Caso não seja, deve-se deixar claro que se trata de uma possibilidade não comprovada).

Ex: O Quadro 2 apresenta a quantidade de adolescentes que utilizam computador nas escolas. Pode-se observar que todos os alunos da classe alta utilizam computadores na escola, comparado com apenas 30% dos alunos da classe baixa. (apresentação)

Quadro 2 – Quantidade de adolescentes que utilizam computador nas escolas

| Classe | Você utiliza computador na escola? | |
|--------|------------------------------------|-----|
| | Sim | Não |
| Baixa | 3 | 7 |
| Média | 8 | 2 |
| Alta | 10 | 0 |

Fonte: Elaborado pelo autor

Ex: Na classe média, 20% dos alunos não utilizam computadores nas escolas. Este resultado se deve ao fato de que estes dois adolescentes estudam em escolas públicas. Apesar disso, todos os alunos da classe baixa estudam em escola pública, mas 30% delas possuem aulas de informática. (análise)

6 CONCLUSÃO (1 PÁGINA)

É o capítulo de encerramento do trabalho, que contém a discussão dos resultados obtidos na pesquisa. É onde se colocam as observações do autor. A conclusão deve estar de acordo com os objetivos do trabalho. Ela não deve apresentar citações ou interpretações de outros autores.

Sendo assim, uma conclusão é composta de: Conclusão Geral. Síntese do que foi realizado. Os objetivos foram alcançados totalmente? Os resultados foram compatíveis com as expectativas? Ex: Este trabalho apresentou um estudo de caso da inclusão digital de adolescentes, realizado por meio da análise de questionários, que foram aplicados para identificação da influência da informática no processo de formação destes adolescentes. Os resultados mostram que os adolescentes de classe baixa com acesso a informática ainda são uma minoria.

Discussão dos Resultados. Extrapolação dos resultados obtidos (e se...); destaque das limitações, vantagens e desvantagens da pesquisa, entre outras observações. Ex: Nesta pesquisa não foi analisado o impacto da informática na formação de crianças e jovens. Apesar disso, os resultados poderiam ser estendidos para estas diferentes faixas etárias.

Contribuições da Pesquisa. Metas alcançadas, ou seja, os objetos importantes produzidos. Ex: A principal contribuição desta pesquisa foi identificar que adolescentes são excluídos do mundo digital principalmente pela falta de aulas de informática nas escolas públicas, independente da classe econômica.

Trabalhos Futuros. Possíveis evoluções da pesquisa, sugestões de pesquisas relacionadas etc. Ex: Como trabalhos futuros, pretende-se realizar um estudo de caso do impacto da informática na formação de crianças e jovens. Além disso, pretende-se fazer um estudo sobre a inclusão digital mais restrita aos adolescentes estudantes de escolas públicas.

REFERÊNCIAS

- CITELLI, A. **Comunicação e educação: a linguagem em movimento**. 3. ed. São Paulo: Editora Senac, 2004.
- CORDEIRO, F. L. R. **Estudo comparativo entre plataforma monoprocessada e clustercom-puting sobre as métricas de desempenho**. 2010. 46 f. Monografia (Graduação em Sistemas de Informação) — Pontifícia Universidade Católica de Minas Gerais, Guanhães, 2010.
- GIL, A. C. **Como elaborar projetos de pesquisa**. 7. ed. Barueri: Atlas, 2022.
- GÓES, L. F. W. **Automatic Skeleton-Driven performance optimizations for transactional memory**. 2012. 108 f. Tese (Doutorado em Informatics) — The University of Edinburgh, Edinburgh, 2012.
- GÓES, L. F. W. et al. Computação em grade: Conceitos, tecnologias, aplicações e tendências. In: __. **Escola Regional de Informática de Minas Gerais**. Belo Horizonte: ERI MG, 2005. cap. 11, p. 40.
- GROPP, W. **Beowulf cluster computing with linux**. 2. ed. Cambridge: MIT Press, 2003. 618 p.
- HABERMAS, J. Trabalho e interação. In: __. **Técnica e ciência como “ideologia”**. Lisboa: Edições 70, 1997. cap. 2, p. 11–28.
- KNUTH, D. E. **The art of computer programming**. 16. ed. Boston: Addison-Wesley, 1968. Fundamental Algorithms.
- MARTINS, J. H. de O. **Apropriação da Informática: Estudo de Caso com Adolescentes do Programa Socioassistencial Espaço Dignidade e Cidadania**. 2012. 48 f. Monografia (Graduação em Sistemas de Informação) — Pontifícia Universidade Católica de Minas Gerais, Contagem, 2012.
- PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS. **Orientações para elaboração de projetos de pesquisa, trabalhos acadêmicos, relatórios técnicos e/ou científicos e artigos científicos**: conforme a associação Brasileira de Normas Técnicas (ABNT). 5. ed. Belo Horizonte: PUC Minas, 2023. Disponível em: <<http://www.pucminas.br/biblioteca/>>. Acesso em: 29 de jul. 2024.
- PRENNER, J. A.; ROBBES, R. Making the most of small software engineering datasets with modern machine learning. **IEEE Transactions on Software Engineering**, v. 48, n. 12, p. 5050–5067, 2022. Disponível em: <<https://doi.org/10.1109/TSE.2021.3135465>>.
- RIBEIRO, A. I. J. T. **Representações neural e fuzzy de controle de admissão de chamadas para redes E-UMTS**. 2010. 103 f. Dissertação (Mestrado em Informática) — Pontifícia Universidade Católica de Minas Gerais, Programa de Pós-graduação em Informática, Belo Horizonte, 2010.
- VASCONCELOS, A. S. V.; CARDOSO, R. T. N.; FERNANDES, J. L. A. Um modelo de otimização multiobjetivo com influência da pluviosidade no controle do mosquito da dengue. In: XXXVI CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL - CNMAC 2016, 2016, Gramado. **Anais**. 2017. v. 5, n. 1, p. 1–7. Disponível em: <<https://doi.org/10.5540/03.2017.005.01.0480>>.

WAZLAWICK, R. S. **Metodologia de pesquisa para ciência da computação**. 3. ed. Rio de Janeiro: LTC, 2021.