



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Instituto de Ciências Exatas e de Informática

Ar Puro MG

Abel Leony Macedo da Paixão
Caio Ignatz Martins
Gabriel Víctor Guimarães Xavier
Iago Oliveira de Almeida
Katryn Ribeiro de Jesus Oliveira
Phelipe Octavio Antunes Silva

Resumo

A ONG Ar Puro MG realiza medições e monitoramento contínuos da qualidade do ar na região metropolitana de Belo Horizonte em resposta ao aumento da poluição atmosférica. Com o objetivo de fornecer dados confiáveis tanto para o poder público quanto para a sociedade civil, a ONG contribui para a formulação de políticas públicas e práticas sustentáveis. A API REST pública, principal produto da organização, permite o acesso em tempo real aos dados de qualidade do ar, bem como o armazenamento e a análise histórica dessas informações. A infraestrutura tecnológica envolve uma rede resiliente de sensores distribuídos, utilizando tecnologias como LPWAN e NB-IoT, para garantir a integridade e segurança dos dados, assegurando a continuidade do serviço. Através de relatórios detalhados e uma gestão eficiente de dados, a ONG busca promover a conscientização ambiental, impactando diretamente a saúde pública e o meio ambiente. Além disso, a Ar Puro MG oferece suporte remoto aos colaboradores e serviços integrados de e-mail, login, e VPN, para garantir a continuidade do trabalho e a comunicação segura entre os sensores e a matriz.

Palavras-chave: Monitoramento do ar; Sensores; LPWAN; NB-IoT; Saúde pública.

Abstract

The NGO Ar Puro MG conducts continuous air quality measurement and monitoring in the metropolitan region of Belo Horizonte in response to rising atmospheric pollution. With the objective of providing reliable data to both public authorities and civil society, the organization contributes to the development of public policies and sustainable practices. The organization's main product, a public REST API,

enables real-time access to air quality data, as well as the storage and historical analysis of this information. The technological infrastructure involves a resilient network of distributed sensors using technologies such as LPWAN and NB-IoT to ensure data integrity and security, guaranteeing service continuity. Through detailed reports and efficient data management, the NGO aims to raise environmental awareness, directly impacting public health and the environment. Additionally, Ar Puro MG provides remote support to collaborators and integrated services for email, login, and VPN to ensure continuous work and secure communication between sensors and the main office.

Keywords: Monitoring; Sensors; LPWAN; NB-IoT; public health.

1. INTRODUÇÃO

A *Ar Puro MG* é uma organização não governamental (ONG) dedicada à medição e monitoramento da qualidade do ar na região metropolitana de Belo Horizonte. Em resposta ao aumento da poluição atmosférica e à necessidade crescente de informações precisas sobre a qualidade do ar, nossa organização se compromete a fornecer dados confiáveis que possam ser utilizados tanto pelo poder público quanto pela sociedade civil para o desenvolvimento de políticas públicas e práticas sustentáveis.

Para garantir o direito à informação sobre a qualidade do ar, a missão da *Ar Puro MG* é promover a saúde pública e o bem-estar da população. Isso é realizado através do monitoramento contínuo e da divulgação transparente dos níveis de poluição atmosférica para órgãos governamentais e para a sociedade civil.

Com o objetivo de ser referência nacional na medição da qualidade do ar, a visão da organização é contribuir para a construção de cidades mais saudáveis e ambientalmente responsáveis, gerando um impacto positivo na vida das pessoas e no meio ambiente.

Além disso, um dos pilares da *Ar Puro MG* é agir com clareza e honestidade na divulgação dos dados e na comunicação com o público, reafirmando o compromisso de proteger e melhorar a saúde da população. As suas atividades possuem práticas que promovem o desenvolvimento sustentável e a preservação do meio ambiente, enquanto busca constantemente por novas tecnologias e metodologias para aprimorar os serviços prestados.

A equipe formada por membros responsáveis pelo desenvolvimento, manutenção, comunicação externa e interna, financeiro e gerência mantém a ONG em funcionamento com o modelo de trabalho presencial na matriz e nas unidades, onde estão os equipamentos que permitem a realização dos trabalhos.

A *Ar Puro MG* opera nos modelos B2B (Business to Business), B2G (Business to Government) e B2C (Business to Consumer), fornecendo serviços para empresas privadas, organizações não governamentais e órgãos públicos, além de atender ao público geral. A empresa oferece dados precisos sobre a qualidade do ar, auxiliando na tomada de decisões e na formulação de políticas ambientais.

As atividades da empresa incluem a instalação e manutenção de dispositivos de medição estrategicamente posicionados na região metropolitana de Belo Horizonte, a coleta e análise de dados em tempo real, e a divulgação dessas informações por meio de uma Interface de Programação de Aplicações (API) aberta.

Com sua infraestrutura de rede e sistemas de computação, a *Ar Puro MG* oferece como principal produto uma API REST aberta ao público, permitindo que desenvolvedores, pesquisadores e organizações acessem e utilizem os dados coletados pelos sensores com segurança e integridade em tempo real.

A API REST pública facilita a integração com aplicações externas e possibilita a criação de soluções inovadoras utilizando os dados fornecidos. Para garantir a eficácia dessa integração, a empresa conta com um sistema robusto de gestão de dados, que abrange a coleta, armazenamento e análise das informações obtidas pelos sensores, assegurando a sua disponibilidade e acessibilidade por meio da API. A segurança dos dados é garantida por

mecanismos de autenticação e autorização rigorosos, assegurando um uso responsável e protegido das informações. Além disso, a infraestrutura está equipada para suportar o trabalho remoto, oferecendo acesso seguro e protegido aos sistemas e servidores da organização.

1.1 SERVIÇOS DE INFRAESTRUTURA

O serviço de monitoramento da qualidade do ar oferecido pela empresa Ar Puro MG utiliza uma infraestrutura de rede moderna e eficiente, baseada em dispositivos Raspberry Pi equipados com sensores de dióxido de carbono e conectados à rede da unidade através de módulos USB WiFi. A topologia da rede segue o modelo estrela (*star*), onde todas as unidades se conectam a um ponto central, neste caso, a sede, para comunicação com um serviço de backend hospedado no Google Cloud. Essa configuração garante que os dados coletados pelos sensores sejam transmitidos de forma rápida e segura para a nuvem, onde são processados e armazenados.

Cada unidade é uma ponta da topologia estrela. As unidades coletam os dados dos sensores e os enviam à sede, que é o centro da topologia, que por sua vez, os envia para o backend via HTTPS. A segurança da comunicação é garantida não apenas pelo uso do protocolo HTTPS, mas também pela implementação de um serviço de VPN que cria túneis criptografados entre as unidades e o backend. Isso assegura que os dados transmitidos estejam protegidos contra interceptação e ataques, mantendo a integridade e a confidencialidade das informações.

O serviço de backend, hospedado no Google Cloud, é responsável por receber, processar e armazenar os dados enviados pelas unidades próximas aos dispositivos Raspberry Pi. A utilização da infraestrutura do Google Cloud permite escalabilidade, alta disponibilidade e facilidade na gestão dos dados, que são críticos para o funcionamento do sistema de monitoramento da qualidade do ar. Além disso, o uso de APIs RESTful facilita a integração com outros serviços e a criação de dashboards e relatórios personalizados para análise dos dados.

Também hospedado no Google Cloud vai estar a aplicação web que possibilita visualizar dados de monitoramentos de maneira consolidada em dashboards utilizando BI (Business Intelligence) e também possibilita ao usuário realizar seu cadastro para obter acesso à API Rest de monitoramento.

Além da rede de dispositivos Raspberry Pi mencionados anteriormente, a infraestrutura de rede da Ar Puro MG inclui um escritório central de desenvolvimento com 7 máquinas conectadas à internet via banda larga compartilhada tanto por Wi-Fi quanto por cabo e filiais com quantidades de máquinas variáveis, cada uma delas possui uma rede local, que é responsável pelos seus sensores e dispositivos, sendo essencial para o desenvolvimento e manutenção do software que gerencia o serviço de monitoramento. A conexão via banda larga permite que a equipe de desenvolvimento tenha acesso à internet para realizar atualizações, acessar ferramentas de desenvolvimento em nuvem e se comunicar com a infraestrutura do Google Cloud. Na rede do escritório também possui a gestão de usuários, grupos e permissões utilizando o serviço de

Active Directory (AD), fornecido pela Microsoft.

A segurança das redes e da comunicação entre elas é feita por firewalls que analisam os dados que trafegam na rede, bloqueando aqueles não desejados, e pela VPN, que isola o tráfego interno.

O modelo de topologia em estrela adotado facilita a escalabilidade da rede, permitindo a integração simples de novas filiais e dispositivos Raspberry Pi. Quando for necessário adicionar mais filiais para expandir o monitoramento, os novos dispositivos podem ser facilmente conectados à rede interna da filial e configurados para enviar dados ao mesmo serviço de backend. Essa flexibilidade é crucial para a Ar Puro MG, pois possibilita a escalabilidade da cobertura de monitoramento conforme a demanda dos clientes aumenta, sem exigir reconfigurações extensivas da infraestrutura de rede.

1.2 INFRAESTRUTURA DA REDE

Para garantir segurança, estabilidade e confiabilidade, o rack da matriz inclui dois servidores dedicados a gestão de DHCP, Active Directory e FTP, um switch central que conecta o switch de cada setor que por sua vez é responsável por interligar os dispositivos cabeados, e um firewall dedicado para controlar o acesso externo e gerenciar a VPN para a conexão entre as unidades com segurança. Adicionalmente, o roteador principal, em conjunto com dois repetidores de sinal, assegura a cobertura WiFi na empresa.

Ao todo, a empresa possui um total de 80 máquinas, 50 máquinas na sede e 30 distribuídas igualmente entre as filiais, incluindo notebooks e computadores. Na sede, as máquinas são distribuídas entre os departamentos e conectadas tanto à rede WiFi quanto aos switches de cada setor. Os dispositivos estão organizados da seguinte forma:

- Diretoria Executiva:
 - 3 Notebook
 - 2 Computadores
 - 1 Smartphone corporativo
 - 1 Impressora wireless
- TI
 - 5 Computadores
 - 5 Notebook's
 - 1 Roteador Wireless WRT300N
 - 1 Switch 2960
- RH / DP
 - 5 Computadores
 - 5 Notebook
 - 1 Impressora wireless
 - 1 Telefone utilizando VoIP
- Laboratório
 - 20 Computadores

- 1 Impressora Wireless
- 1 Switch 2960
- Financeiro
 - 5 Computadores
 - 1 Impressora conectada via cabo Ethernet
 - 1 Switch 2960
- Sala de Equipamentos
 - 2 Servidores para DHCP, AD e FTP
 - 1 Roteador para acesso a rede externa (Gateway)
 - 1 Roteador wireless WRT300N para gerenciar rede para visitantes
 - 1 Firewall

Os serviços relacionados a DNS, VPN e aplicação web são disponibilizados utilizando servidores em nuvem com o intuito de promover uma alta disponibilidade e possibilitar uma escalabilidade que não afete o espaço físico.

Sobre as filiais, para garantir um suporte eficiente e a continuidade das operações, a estrutura de rede inclui uma sala de equipamentos dedicada em cada uma das três filiais. Cada sala de equipamentos está equipada com servidores dedicados para gerenciar as funções críticas da rede e garantir a disponibilidade dos serviços essenciais. As filiais possuem 10 máquinas cada, incluindo computadores e notebooks, conectadas a switches centrais que interligam todos os dispositivos de rede e 1 dispositivo Raspberry Pi por filial. A equipe de desenvolvimento, que é responsável pela manutenção e atualização contínua da infraestrutura de TI, tem acesso a equipamentos e recursos adequados para suportar suas atividades. Os roteadores nas filiais são configurados para fornecer acesso seguro à internet e manter a comunicação entre as unidades, enquanto os switches garantem que todos os dispositivos estejam conectados de forma estável e eficiente. A integração com os serviços em nuvem assegura a escalabilidade e a alta disponibilidade dos recursos, promovendo uma operação contínua e a capacidade de responder rapidamente às necessidades da empresa.

Departamento de Suporte:

- 3 Máquinas: Essas máquinas serão usadas para realizar o suporte técnico diário, gerenciamento de tickets e manutenção de sistemas, assegurando que qualquer problema de TI seja resolvido de forma rápida e eficiente.

Sala de Equipamentos:

- 2 Máquinas: Essas máquinas estarão dedicadas ao gerenciamento e monitoramento da infraestrutura de rede, além de servir como estações para a configuração e manutenção dos equipamentos da rede.

Equipe de Desenvolvimento:

- 5 Máquinas: Esses computadores serão utilizados pelos desenvolvedores para codificação, testes de software e desenvolvimento de aplicações. Ter uma quantidade maior de máquinas aqui permitirá que a equipe trabalhe simultaneamente em projetos diferentes, aumentando a produtividade e eficiência no desenvolvimento.

As comunicações entre os sensores conectados via 5G e a API em nuvem são criptografadas, garantindo a privacidade e a segurança dos dados. A rede é monitorada continuamente, com a coleta de logs para detectar e responder rapidamente a qualquer atividade suspeita. Esta abordagem proporciona uma rede bem estruturada, capaz de suportar o monitoramento eficaz da qualidade do ar e garantir uma operação contínua e confiável em diversas localidades da região metropolitana de Minas Gerais.

Além disso, o roteador será configurado com NAT (Network Address Translation) para permitir que os dispositivos da rede local acessem a internet. O roteador também atuará como servidor DHCP, distribuindo endereços IP para os dispositivos, exceto para o Raspberry Pi, que manterá um IP fixo. Para a segurança da rede, será implementada uma senha de acesso ao console do roteador e à configuração via SSH.

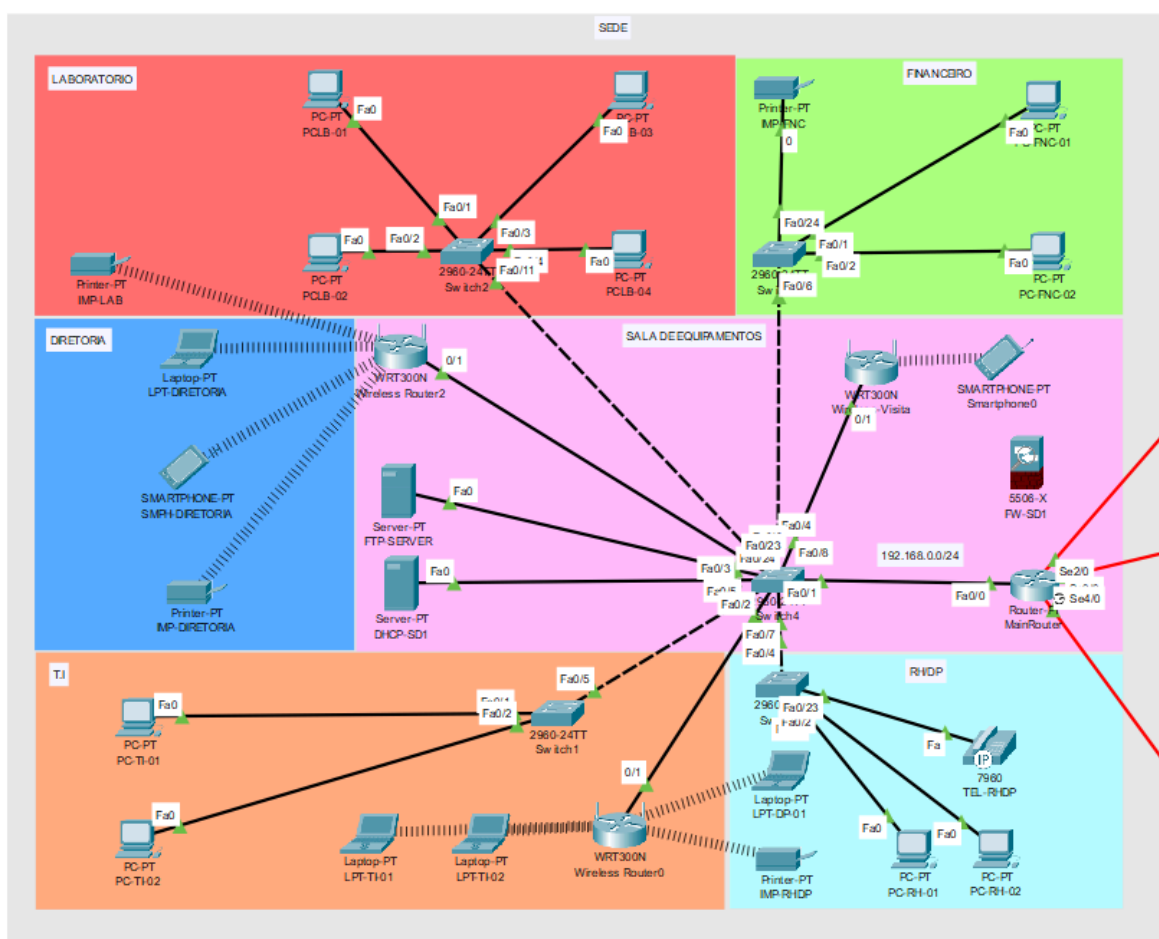


Figura 1 -Proposta de Infraestrutura de Rede para a Sede da Ar Puro MG

2. CLOUD E VIRTUALIZAÇÃO

2.1. CLOUD

IP	Nome	Usuário de acesso
54.84.175.69	ip-172.31.37.158.ec2.internal	phe-ubuntu
13.59.36.208	ip-172-31-33-21.us-east-2.compute.internal	arpuromg

A tabela pode rapidamente se tornar desatualizada devido à natureza dinâmica dos endereços IP e à possibilidade de mudanças nas configurações dos serviços em nuvem.

2.1.1. LINK DO VÍDEO

- <https://www.youtube.com/watch?v=RS-oDfOPGzo>

2.1.1. FTP

O processo de criação de um servidor FTP na AWS envolve várias etapas, desde a configuração inicial de uma instância até o teste final de conectividade e acesso. Primeiramente, foi configurada uma instância EC2 com IP público, essencial para que a máquina esteja acessível via internet. Em seguida, definimos regras de segurança específicas no grupo de segurança da instância para liberar as portas 20 e 21 (utilizadas para o protocolo FTP) e as portas 12000 a 12100 (para o modo passivo do FTP, essencial em firewalls que restringem conexões de entrada e saída). Essas configurações garantem que o servidor FTP esteja pronto para receber conexões externas e que possa operar sem restrições de portas durante a transferência de arquivos.

Após a configuração inicial da rede, realizamos um teste básico de conectividade, utilizando o comando ping a partir de uma máquina local, para confirmar que a instância estava ativa e acessível. Com a confirmação de conectividade, o próximo passo foi a criação de um usuário na máquina Linux da AWS, o qual servirá como credencial de acesso para o FTP. Esse usuário foi configurado com as permissões necessárias para o diretório de FTP, permitindo um controle seguro e restrito dos arquivos que poderão ser acessados e transferidos.


```
Prompt de Comando
Microsoft Windows [versão 10.0.19045.5011]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Phelipe>ping 52.87.188.21

Disparando 52.87.188.21 com 32 bytes de dados:
Resposta de 52.87.188.21: bytes=32 tempo=147ms TTL=54
Resposta de 52.87.188.21: bytes=32 tempo=140ms TTL=54
Resposta de 52.87.188.21: bytes=32 tempo=141ms TTL=54
Resposta de 52.87.188.21: bytes=32 tempo=144ms TTL=54

Estatísticas do Ping para 52.87.188.21:
    Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de
    perda),
Aproximar um número redondo de vezes em milissegundos:
    Mínimo = 140ms, Máximo = 147ms, Média = 143ms

C:\Users\Phelipe>
```

Figura 7 - Ping de conexão local

Resumo da instância para i-071d1f2629fc7e6d6 (phe-ubuntu) <small>informações</small>				Conectar	Estado da instância ▾	Ações ▾
<small>Atualizado há about 2 hours</small>						
ID da instância i-071d1f2629fc7e6d6 (phe-ubuntu)	Endereço IPv4 público 52.87.188.21 endereço aberto	Endereços IPv4 privados 172.31.37.158				
Endereço IPv6 -	Estado da instância Executando	DNS IPv4 público ec2-52-87-188-21.compute-1.amazonaws.com endereço aberto				
Tipo de nome do host Nome do IP: ip-172-31-37-158.ec2.internal	Nome do DNS de IP privado (somente IPv4) ip-172-31-37-158.ec2.internal	Endereços IP elásticos -				
Nome do DNS do recurso privado de resposta IPv4 (A)	Tipo de instância t2.micro	Descoberta do AWS Compute Optimizer Opte por participar do AWS Compute Optimizer para obter recomend ações. Saiba mais				
Endereço IP atribuído automaticamente 52.87.188.21 [IP público]	ID da VPC vpc-0d83f1c37b224ba21	Nome do Grupo do Auto Scaling -				
Função do IAM -	ID da sub-rede subnet-0e465fa683f8e39e7					
IMDSv2 Required	ARN da instância arn:aws:ec2:us-east-1:090895162896:instance/i-071d1f2629fc7e6d6					

Figura 8 - Instância criada no servidor aws

Para garantir que o servidor FTP estivesse acessível e seguro, configuramos regras específicas no grupo de segurança da instância na AWS. Foram criadas permissões de entrada para as portas 20 e 21, utilizadas pelo protocolo FTP para estabelecer conexões e transferir dados, e para o intervalo de portas 12000 a 12100, essencial para o modo passivo do FTP, que facilita a comunicação através de firewalls e melhora a estabilidade das conexões. Essas regras foram definidas para permitir apenas o tráfego necessário, minimizando os riscos de acesso não autorizado. O print a seguir demonstra a configuração das regras de entrada no grupo de segurança, exibindo as portas liberadas e os protocolos permitidos, garantindo a segurança e a funcionalidade do servidor.

Regras de entrada [Informações](#)

ID da regra do grupo de segurança	Tipo Informações	Protocolo Informações	Intervalo de portas Informações	Origem Informações	Descrição - opcional Informações	
sgr-00b595cae263f9fb5	TCP personalizado	TCP	20 - 21	Qualquer L... 0.0.0.0/0 X	porta FTP	Excluir
sgr-081783fd61235db34	Todos os ICMPs - IPv4	ICMP	Tudo	Qualquer L... 0.0.0.0/0 X	acesso ping	Excluir
sgr-03f624006f15dcee8	TCP personalizado	TCP	12000 - 12100	Qualquer L... 0.0.0.0/0 X	porta alta FTP	Excluir
sgr-009a2abdd7c8e4cd8	SSH	TCP	22	Qualquer L... 0.0.0.0/0 X		Excluir

[Adicionar regra](#)

⚠ As regras com a origem 0.0.0.0/0 ou ::/0 permitem que todos os endereços IP acessem a instância. Recomendamos configurar as regras de grupo de segurança para permitir o acesso apenas de endereços IP conhecidos.

[Cancelar](#) [Visualizar alterações](#) [Salvar regras](#)

Figura 9 - Regras aplicadas

```
Last login: Sat Oct 19 20:35:20 2024 from 201.172.172.171
ubuntu@ip-172-31-37-158:~$ sudo ufw allow 20:21/tcp
Rules updated
Rules updated (v6)
ubuntu@ip-172-31-37-158:~$ sudo ufw allow 12000:12100/tcp
Rules updated
Rules updated (v6)
ubuntu@ip-172-31-37-158:~$ sudo ufw allow 22/tcp
Rules updated
Rules updated (v6)
ubuntu@ip-172-31-37-158:~$ sudo ufw allow ssh
Skipping adding existing rule
Skipping adding existing rule (v6)
ubuntu@ip-172-31-37-158:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
ubuntu@ip-172-31-37-158:~$
```

Figura 10 - Regras aplicadas na maquina linux

```
C:\> ubuntu@ip-172-31-37-158: ~

ubuntu@ip-172-31-37-158:~$ sudo ufw status
Status: active

To Action From
--
20:21/tcp ALLOW Anywhere
12000:12100/tcp ALLOW Anywhere
22/tcp ALLOW Anywhere
20:21/tcp (v6) ALLOW Anywhere (v6)
12000:12100/tcp (v6) ALLOW Anywhere (v6)
22/tcp (v6) ALLOW Anywhere (v6)

ubuntu@ip-172-31-37-158:~$
```

Figura 11 - Permissão das regras aplicadas na máquina linux

Por fim, a configuração do cliente FTP foi realizada no FileZilla, onde inserimos o IP público da instância, o nome de usuário criado, e as portas configuradas para a conexão. Com

isso, foi possível acessar a máquina remotamente e iniciar as transferências de arquivos de forma segura e eficiente.

```
ubuntu@ip-172-31-37-158:/etc/vsftpd$ sudo service vsftpd status
● vsftpd.service - vsftpd FTP server
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-10-19 21:43:38 UTC; 37s ago
     Process: 2578 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty (code=exited, status=0/SUCCESS)
    Main PID: 2579 (vsftpd)
      Tasks: 1 (limit: 1130)
     Memory: 708.0K (peak: 1.1M)
        CPU: 6ms
      CGroup: /system.slice/vsftpd.service
              └─2579 /usr/sbin/vsftpd /etc/vsftpd.conf

Oct 19 21:43:38 ip-172-31-37-158 systemd[1]: Starting vsftpd.service - vsftpd FTP server...
Oct 19 21:43:38 ip-172-31-37-158 systemd[1]: Started vsftpd.service - vsftpd FTP server.
ubuntu@ip-172-31-37-158:/etc/vsftpd$
```

Figura 12 - Estrutura de pastas

```
Oct 19 21:43:38 ip-172-31-37-158 systemd[1]: Starting vsftpd.service - vsftpd FTP server...
Oct 19 21:43:38 ip-172-31-37-158 systemd[1]: Started vsftpd.service - vsftpd FTP server.
ubuntu@ip-172-31-37-158:/etc/vsftpd$ ftp Phelipe@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>
```

Figura 13 - Criação de usuário para conexão FTP

Após a configuração completa, realizamos testes de envio e recebimento de arquivos entre o cliente FileZilla e o servidor FTP na instância AWS. Inicialmente, testamos o envio de arquivos do computador local para o servidor, selecionando arquivos de tamanhos variados no FileZilla para verificar a estabilidade e a integridade da conexão. Observamos que os arquivos foram transferidos corretamente para o diretório FTP configurado, confirmando que as permissões de escrita e leitura estavam devidamente configuradas para o usuário. Em seguida, realizamos o teste inverso, baixando arquivos do servidor para a máquina local, o que garantiu que o fluxo de dados bidirecional funcionasse sem problemas. Ambos os testes foram bem-sucedidos, demonstrando que o servidor FTP estava devidamente configurado para operações de upload e download, atendendo aos requisitos de segurança e conectividade definidos no projeto.

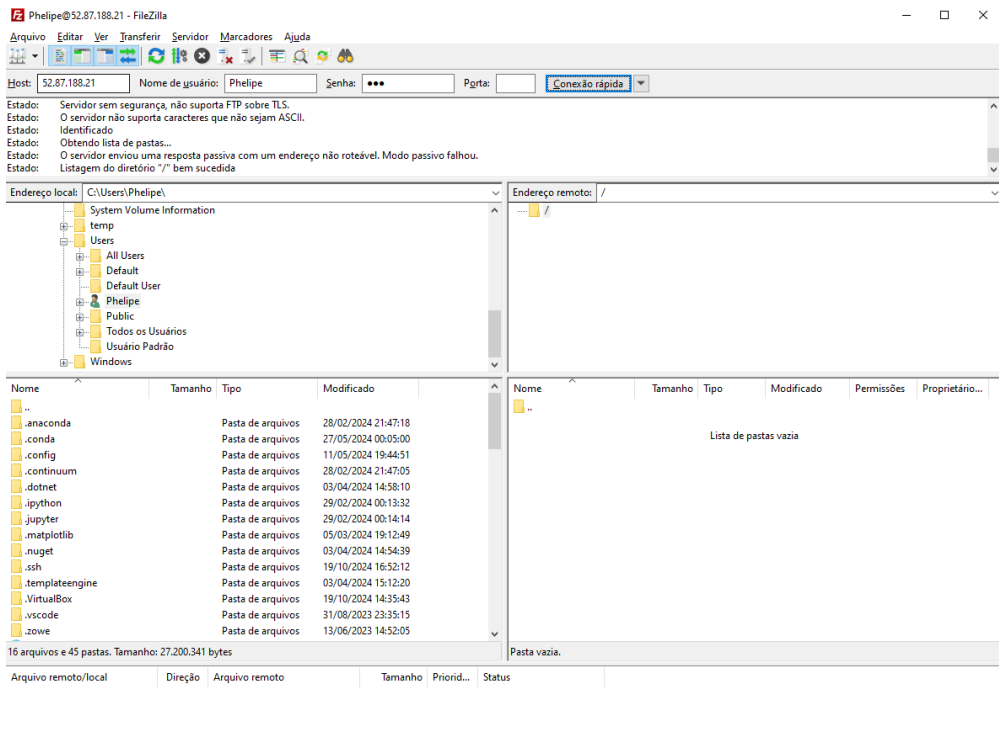


Figura 14 - Conexão FileZilla

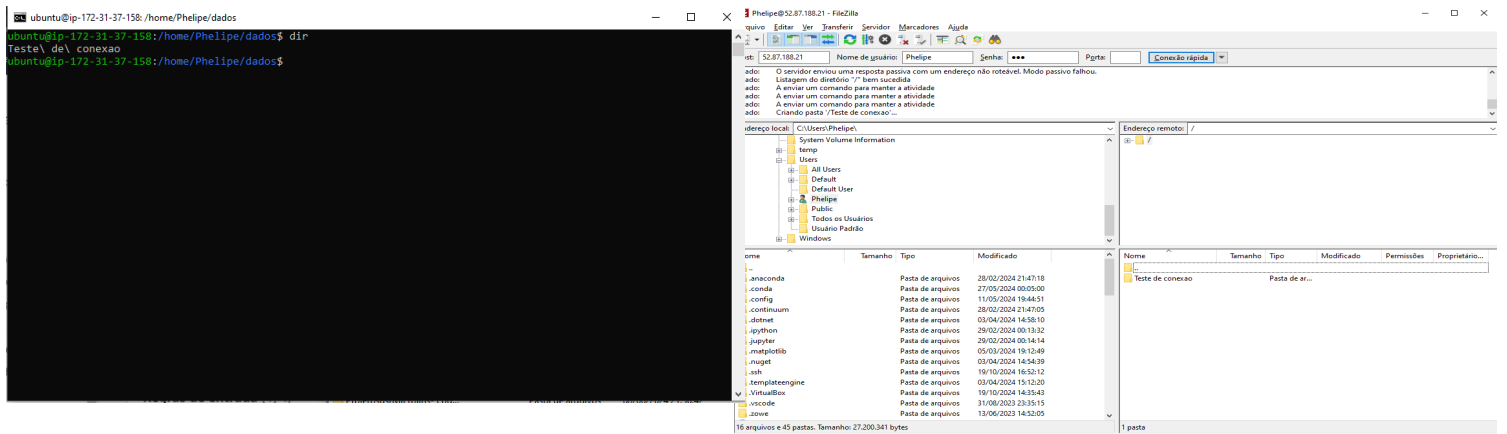


Figura 15 - Criação de pasta para testar conexão através do lado do FileZilla

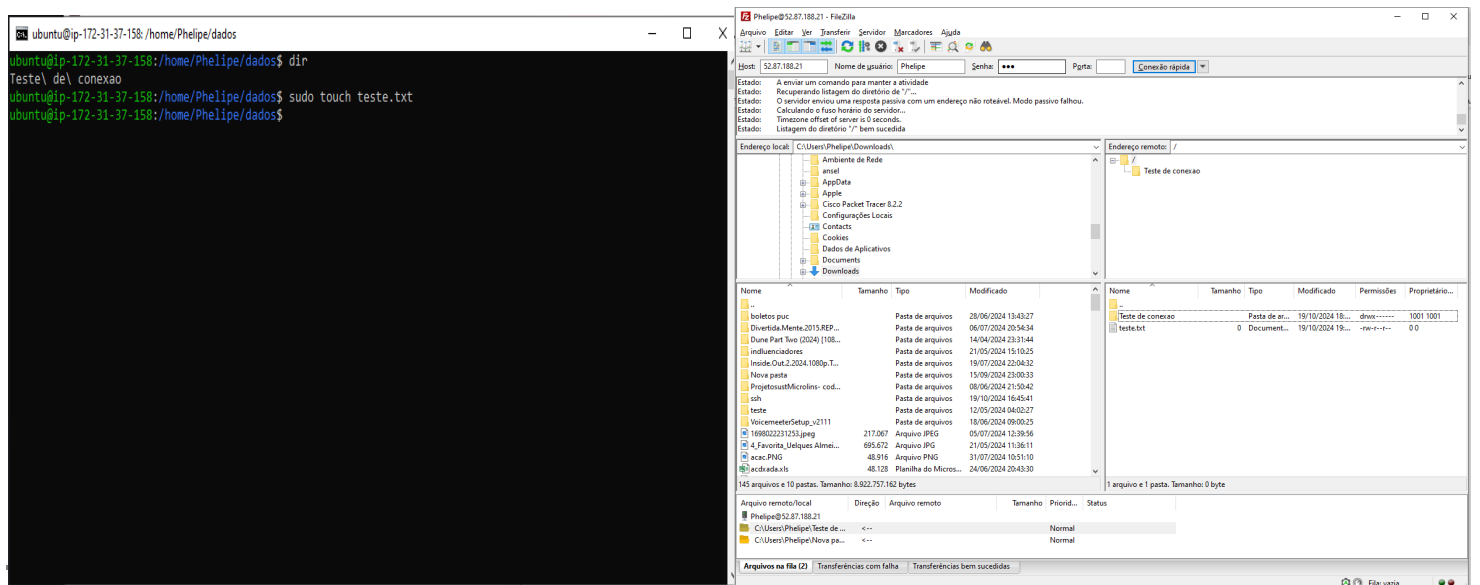


Figura 16 - Criação de arquivo pela máquina virtual.

Com a implementação e os testes bem-sucedidos do servidor FTP na instância AWS, concluiu-se a configuração garantindo segurança, conectividade e funcionalidade para transferências de arquivos. A definição cuidadosa das regras de segurança e o uso de um cliente FTP como o FileZilla asseguraram que o acesso fosse seguro e restrito ao usuário configurado, evitando acessos indesejados. Além disso, o teste de conectividade e a verificação do fluxo de dados bidirecional mostraram que o servidor está operando conforme o esperado, possibilitando o uso eficiente para futuras operações de upload e download. Essa configuração serve como uma base sólida para integração e para o uso contínuo do FTP de forma segura e eficiente na AWS.

2.1.2. API

A API REST foi desenvolvida utilizando a linguagem Python 3.11 em conjunto com o framework FastAPI. FastAPI é uma escolha robusta para a criação de APIs devido à sua performance e suporte à tipagem, facilitando o desenvolvimento de aplicações escaláveis e com segurança em foco. Durante o desenvolvimento, os endpoints da API foram expostos pela aplicação localmente na porta 8000, o que é uma prática comum para facilitar testes e validações.

Após a conclusão do desenvolvimento, a aplicação foi encapsulada em uma imagem Docker. O uso de Docker para empacotar a aplicação oferece benefícios de portabilidade e consistência entre ambientes, garantindo que a API REST funcione da mesma forma em qualquer sistema que suporte Docker. Essa imagem foi então implantada em uma máquina AWS EC2 com sistema operacional Debian, uma distribuição Linux confiável para aplicações de servidor, que oferece estabilidade e segurança.

Para disponibilizar a aplicação de forma segura e acessível, o container Docker foi configurado para ser executado de forma contínua, mapeando a porta 8000 do container para a porta 80 do host. Esse mapeamento de portas é essencial para tornar a aplicação acessível via HTTP na porta padrão (80), simplificando o acesso para os clientes e eliminando a necessidade de especificar a porta na URL. A escolha da porta e a configuração de mapeamento também foram orientadas pelo security group configurado na máquina EC2. Esse security group é configurado para permitir apenas requisições HTTP provenientes de fora da rede em que a máquina está inserida, garantindo que apenas o tráfego necessário chegue à aplicação e bloqueando acessos indesejados.

Para manutenção e ajustes contínuos na aplicação, o desenvolvedor autorizado pode acessar a máquina AWS EC2 por meio de login via SSH, utilizando um certificado .pem para autenticação segura. Esse método de acesso remoto via chave pública garante que apenas indivíduos com as credenciais corretas possam realizar alterações na API, fortalecendo a segurança e a integridade do sistema.

Esta configuração atende a diversos requisitos fundamentais de Infraestrutura de Rede, como segurança, controle de acesso e facilidade de manutenção. Além disso, a utilização de containers Docker e a configuração de uma máquina virtual em nuvem são práticas alinhadas com o conceito de infraestrutura como serviço (IaaS), facilitando a escalabilidade e o gerenciamento dos recursos de rede.

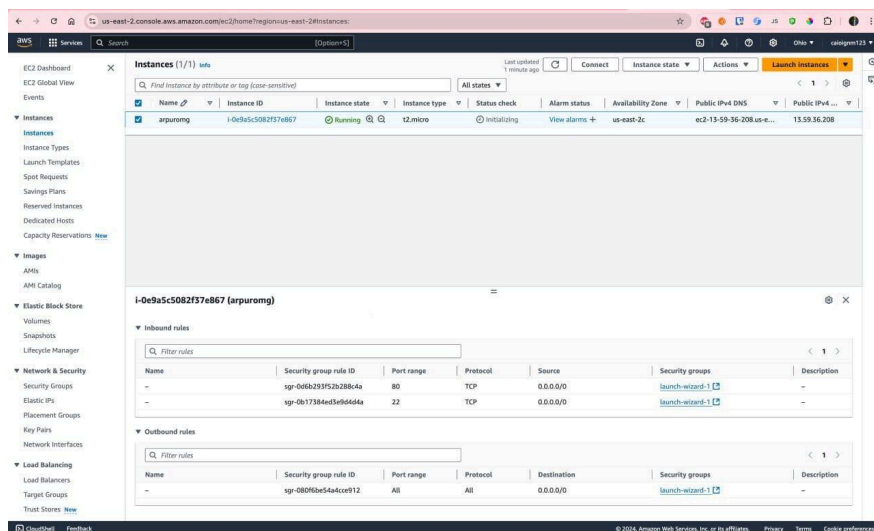


Figura 17 - Visão geral da máquina AWS EC2.

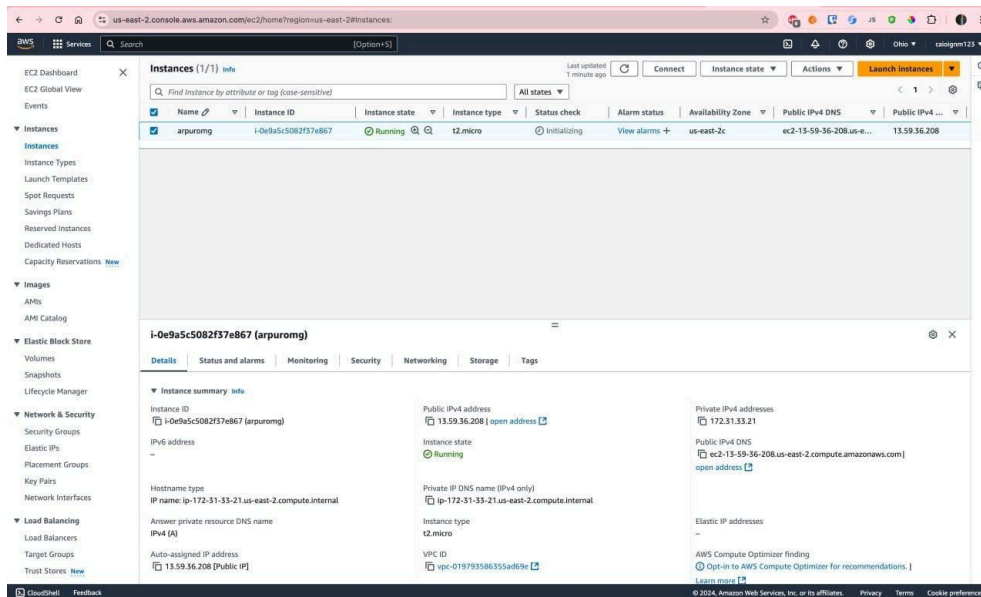


Figura 18 - Detalhes da máquina AWS EC2.

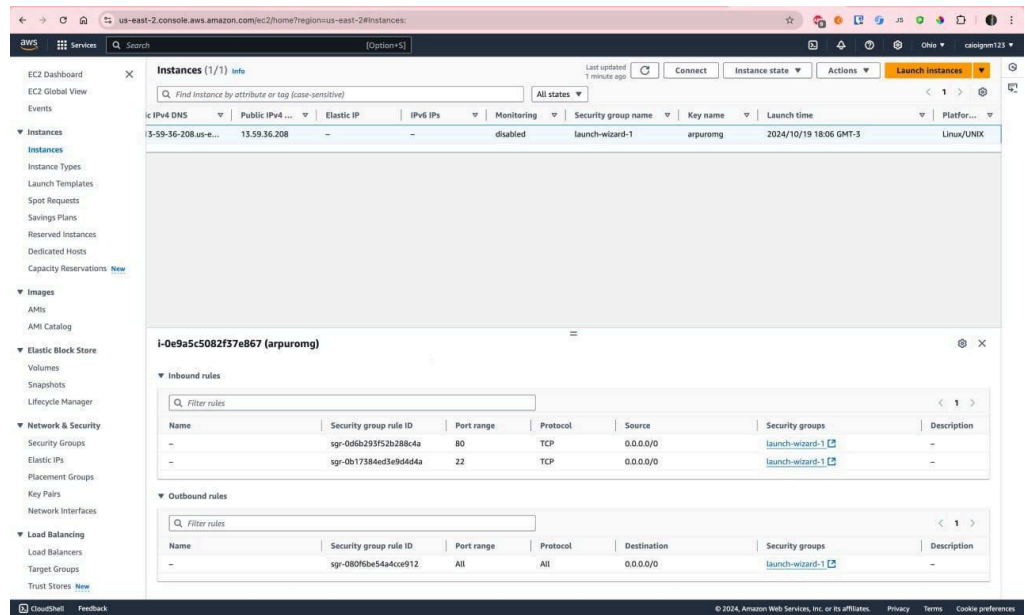


Figura 19 - Regras de rede da máquina AWS EC2.

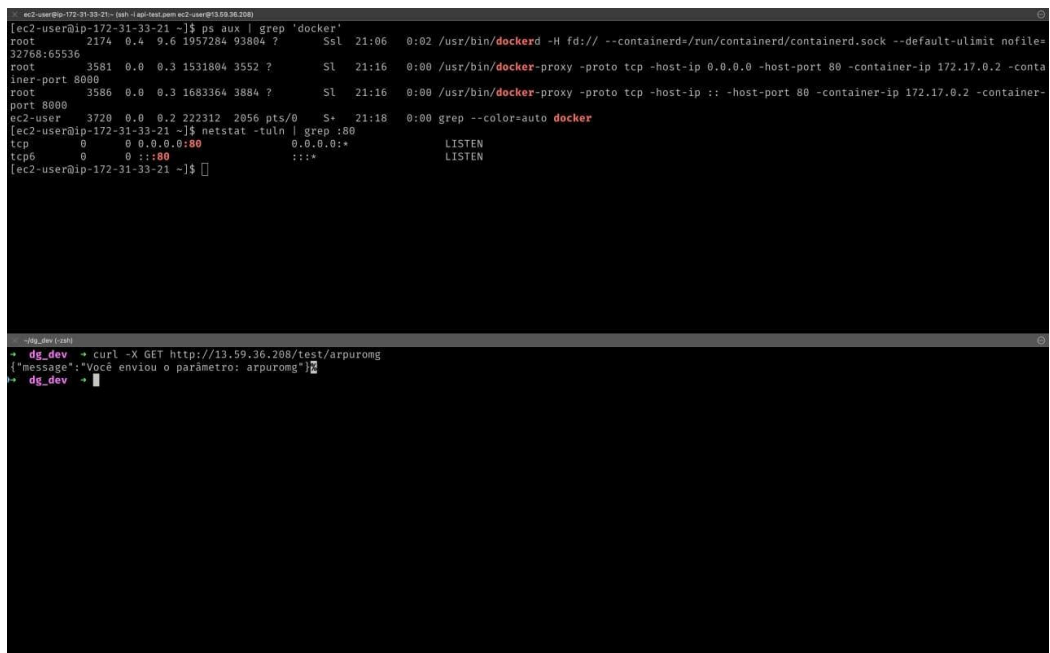


Figura 20 - Processo contínuo do Docker na máquina AWS EC2.

2.2. VIRTUALIZAÇÃO

2.2.1. DHCP

O DHCP (Dynamic Host Configuration Protocol) é um protocolo de rede que automatiza a atribuição de configurações de rede para dispositivos conectados. Ele elimina a necessidade de configurar manualmente cada dispositivo na rede.

Abaixo na **Figura 1** é exibido a interface de rede enp0s3 da máquina virtual que simula um cliente que já possui ip obtido via dhcp (192.168.2.20)

```
root@client:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:59:b8:99 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.20/24 brd 192.168.2.255 scope global dynamic enp0s3
        valid_lft 520sec preferred_lft 520sec
    inet6 fe80::a00:27ff:fe59:b899/64 scope link
        valid_lft forever preferred_lft forever
root@client:~#
```

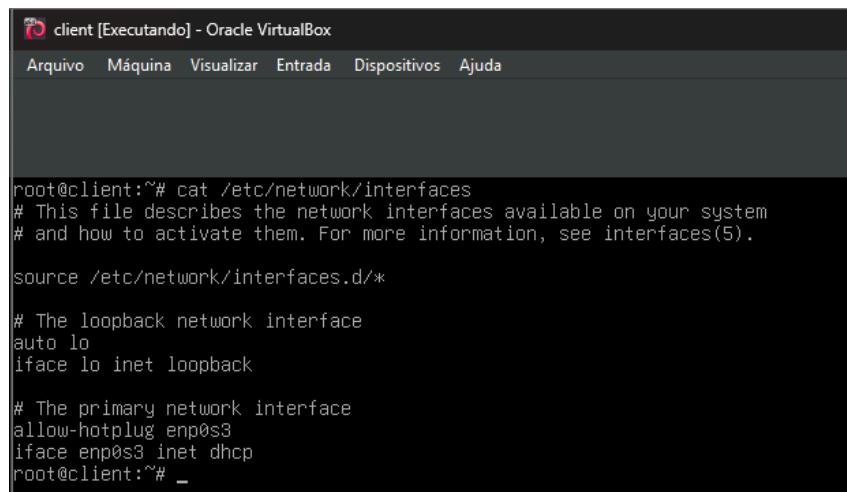
Figura 1 - Interfaces de Rede (Cliente)

Na **Figura 2** são exibidas as interfaces eth0 e eth1 na máquina virtual que simula o servidor dhcp. A interface eth0 funciona como uma placa de rede em modo bridge para acesso a rede externa (rede da máquina host). A interface eth1 aponta para a rede interna para distribuir endereços ip (dhcp do servidor) constando ip fixo 192.168.2.1 para o proprio servidor dhcp

```
root@server:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:fe:73:d4 brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 192.168.0.103/24 brd 192.168.0.255 scope global dynamic eth0
        valid_lft 7121sec preferred_lft 7121sec
    inet6 fe80::a00:27ff:fe73:d4/64 scope link
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:6a:fd:df brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 192.168.2.1/24 brd 192.168.2.255 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe6a:fdff/64 scope link
        valid_lft forever preferred_lft forever
root@server:~#
```

Figura 2 - Interfaces de Rede (Servidor)

Os ip's exibidos nas figuras anteriores para as interfaces de cliente e servidor são listadas abaixo pelo mapeamento no arquivo **/etc/network/interfaces**. A **Figura 3** exhibe a interface enp0s3 do cliente operando em modo **dhcp**.



```
client [Executando] - Oracle VirtualBox
Arquivo  Máquina  Visualizar  Entrada  Dispositivos  Ajuda

root@client:~# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

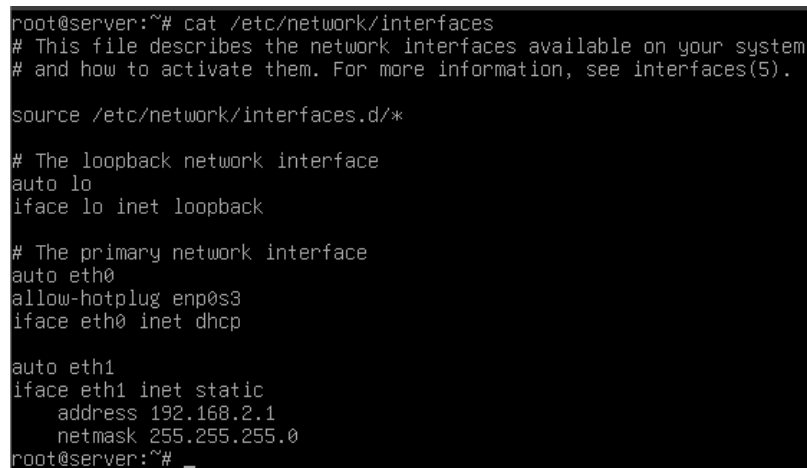
source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet dhcp
root@client:~# _
```

Figura 3 - Mapeamento DHCP da interface cliente

Na **Figura 4** abaixo é exibida a interface eth1 do servidor com ip estático **192.168.2.1** e a interface eth0 (bridge) obtendo ip via dhcp (rede host)



```
root@server:~# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

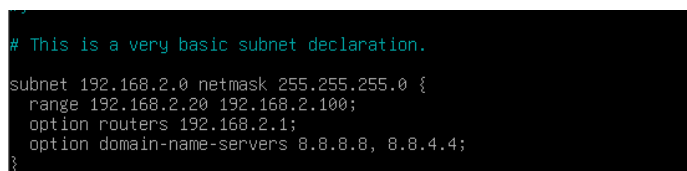
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
allow-hotplug enp0s3
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
    address 192.168.2.1
    netmask 255.255.255.0
root@server:~# _
```

Figura 4 - Mapeamento de ip estático para interface de rede interna do servidor

Na **Figura 5** é exibido o intervalo de ip onde o servidor irá operar para atribuir ip aos dispositivos da rede, iniciando em **192.168.2.20** indo até **192.168.2.100**. A definição deste intervalo é feita no arquivo **/etc/dhcp/dhcpd.conf**.



```
# This is a very basic subnet declaration.

subnet 192.168.2.0 netmask 255.255.255.0 {
    range 192.168.2.20 192.168.2.100;
    option routers 192.168.2.1;
    option domain-name-servers 8.8.8.8, 8.8.4.4;
}
```

Figura 5 - Intervalo de ip a ser atribuído pelo servidor dhcp

2.2.2. VOIP

O VoIP (Voice over Internet Protocol) é uma tecnologia que permite fazer chamadas de voz através da internet, convertendo a voz em dados digitais que são transmitidos pela rede IP.

Exibido na **Figura 1** abaixo, se encontra as configurações do plano de discagem (dialplan) no Asterisk, onde 1000/2000 são os números de ramal, 1 é o parâmetro de prioridade, Dial o comando para discar e 10 o tempo em segundos para a chamada tocar até ser considerada não atendida.

```
[internal]
exten => 1000, 1, Dial(SIP/1000, 10)
exten => 2000, 1, Dial(SIP/2000, 10)
root@server:~# cat /etc/asterisk/extensions.conf
```

Figura 1 - Configuração do arquivo /etc/asterisk/extensions.conf

Na **Figura 2** são exibidas as portas que foram liberadas via ufw para não haver bloqueio por parte do firewall no tráfego do VoIP.

```
root@server:~# ufw status
Status: active

To Action From
--
5060/udp ALLOW Anywhere
5060/tcp ALLOW Anywhere
4569/udp ALLOW Anywhere
10000:20000/udp ALLOW Anywhere
5060/udp (v6) ALLOW Anywhere (v6)
5060/tcp (v6) ALLOW Anywhere (v6)
4569/udp (v6) ALLOW Anywhere (v6)
10000:20000/udp (v6) ALLOW Anywhere (v6)
```

Figura 2 - Portas liberadas no firewall via UFW

Por fim, na **Figura 3** é exibida a configuração no arquivo /etc/asterisk/sip.conf dos usuários registrados, senhas, porta de comunicação do VoIP entre outras parametrizações.

```
[general]
context=internal
allowguest=no
allowoverlap=no
port=5060
bindport=5060
bindaddr=0.0.0.0
srvlookup=yes
;tos=0x18

[1000]
type=friend
username=1000
secret=senha1000
host=dynamic
context=internal

[2000]
type=friend
username=2000
secret=senha2000
host=dynamic
context=internal
root@server:~# cat /etc/asterisk/sip.conf
```

Figura 3 - Definição dos usuários do VoIP

2.2.3. AD/DNS

Os serviços de AD/DNS (Active Directory e Domain Name System) são responsáveis principalmente por determinar permissões de usuários e máquinas e simplificar o endereçamento das mesmas para as pessoas, respectivamente.

As **Figuras 1 e 2** abaixo mostram as configurações utilizadas no Virtualizador, com uma máquina Windows Server 2022 e ao menos uma máquina Windows 10 para conferências e testes, com o foco caindo sobre os adaptadores de rede, com o servidor com uma para rede interna e outra para rede externa em modo bridge, já na máquina cliente, somente apontando para a rede interna do virtualizador.

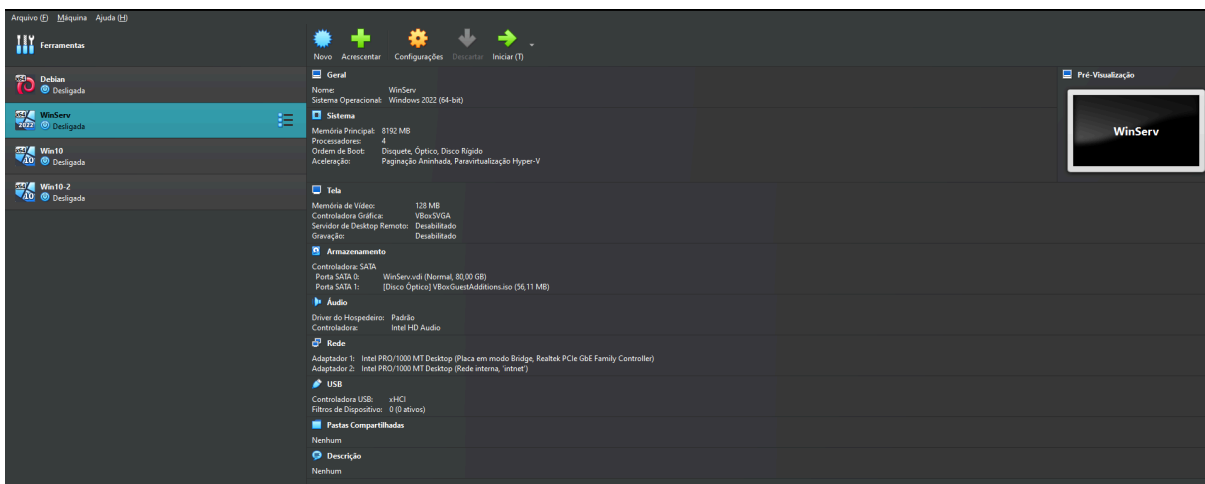


Figura 1 - Configuração Windows Server

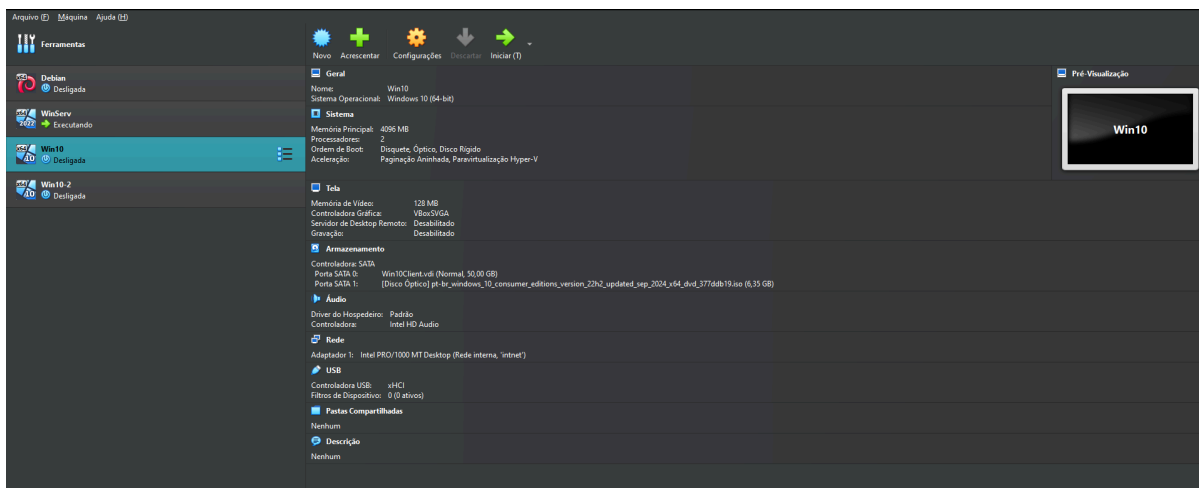


Figura 2 - Configuração Windows Client

Já na **Figura 3**, temos os serviços que estão configurados e ativos no servidor.

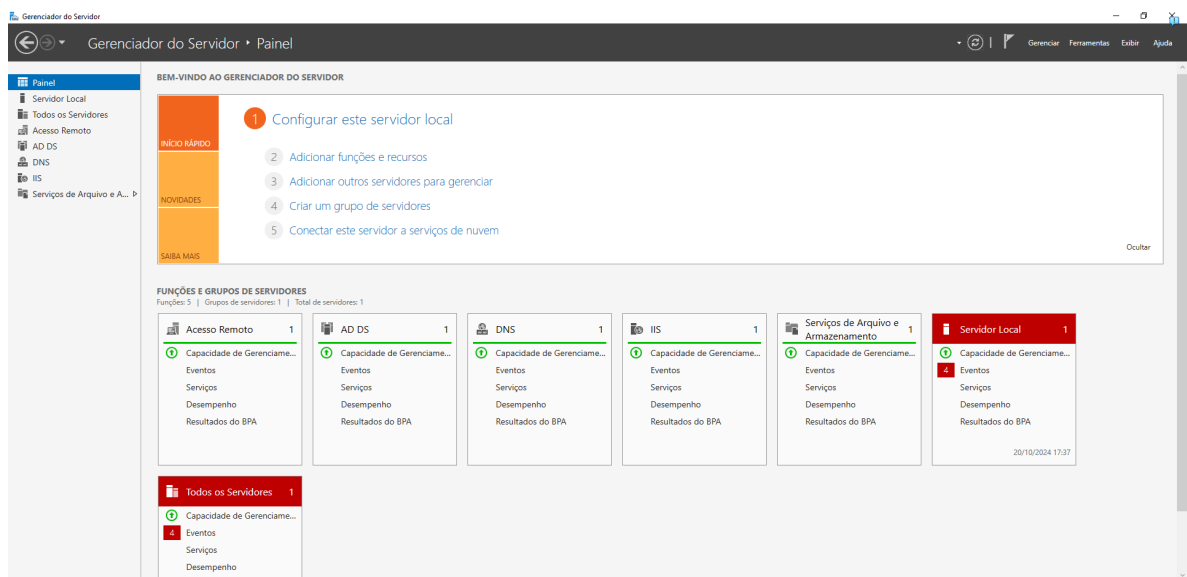


Figura 3 - Serviços ativos

Temos na **Figura 4**, os IPs configurados manualmente no servidor, com um ip para a rede externa e outro para a rede interna, nesse último, o IP 172.16.0.254 é o endereço do servidor na rede interna, sendo também o endereço do Servidor DNS para as duas máquinas e para o gateway padrão da máquina cliente, que tem o IP 172.16.0.1.

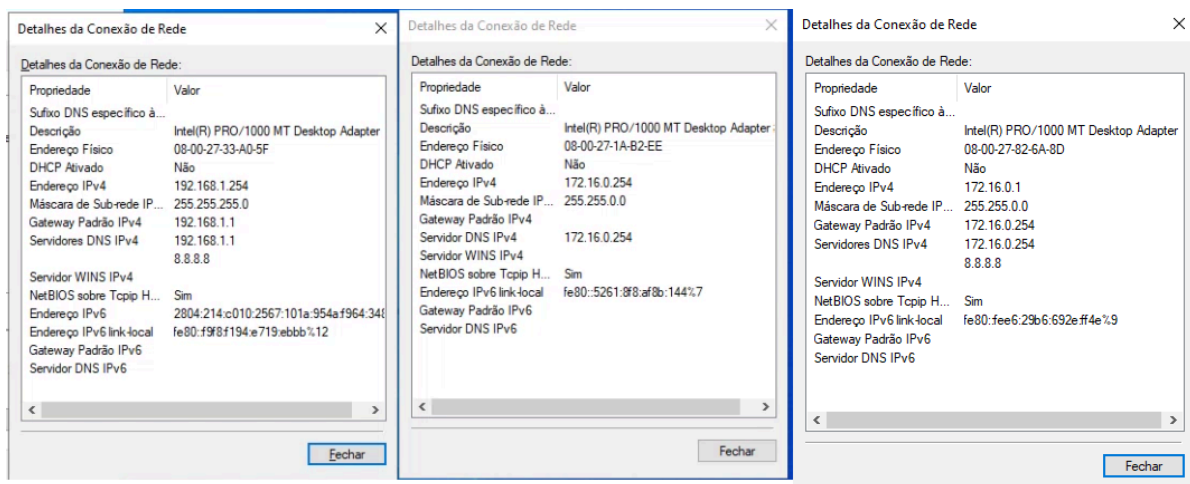


Figura 4 - IPs Servidor e IP Cliente

Já a **Figura 5** mostra a estrutura da política de grupo, essa que define quais são as permissões dos usuários e características das máquinas dentro daquele grupo criado, seja ele de usuários, máquinas ou até mais abrangente, como uma localização.

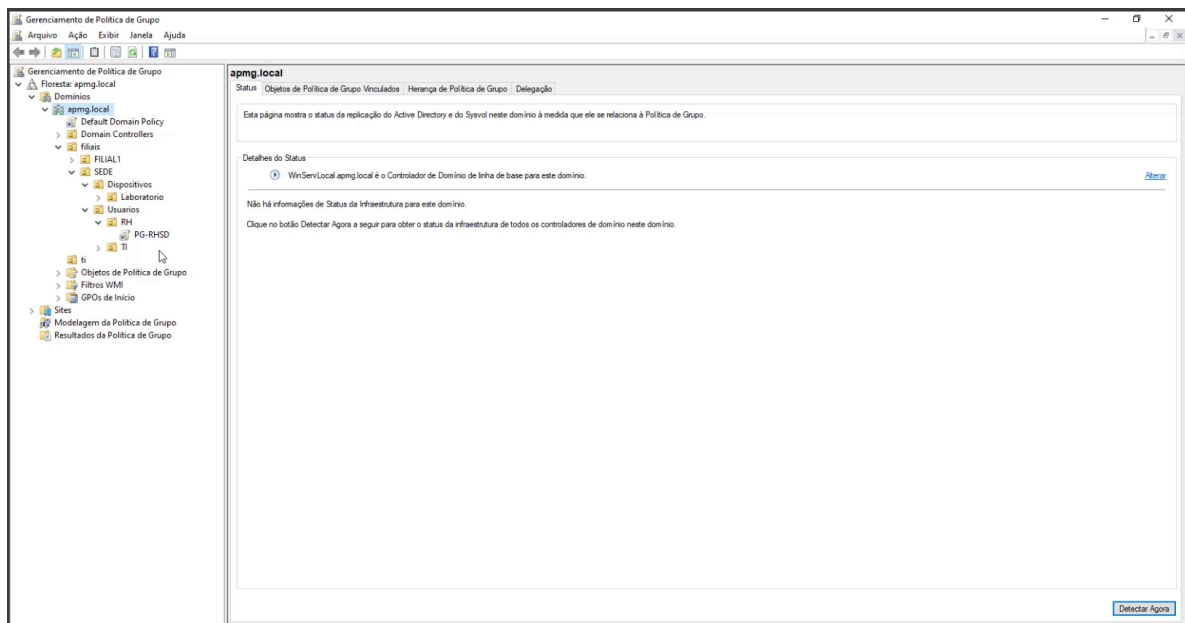


Figura 5 - Estrutura GPO

E na **Figura 6** um exemplo de proibição definida para um grupo, e a demonstração da mensagem retornada pela falta de permissão de um usuário dentro daquele grupo ao tentar acessar o painel de controle **Figuras 7 e 8**.

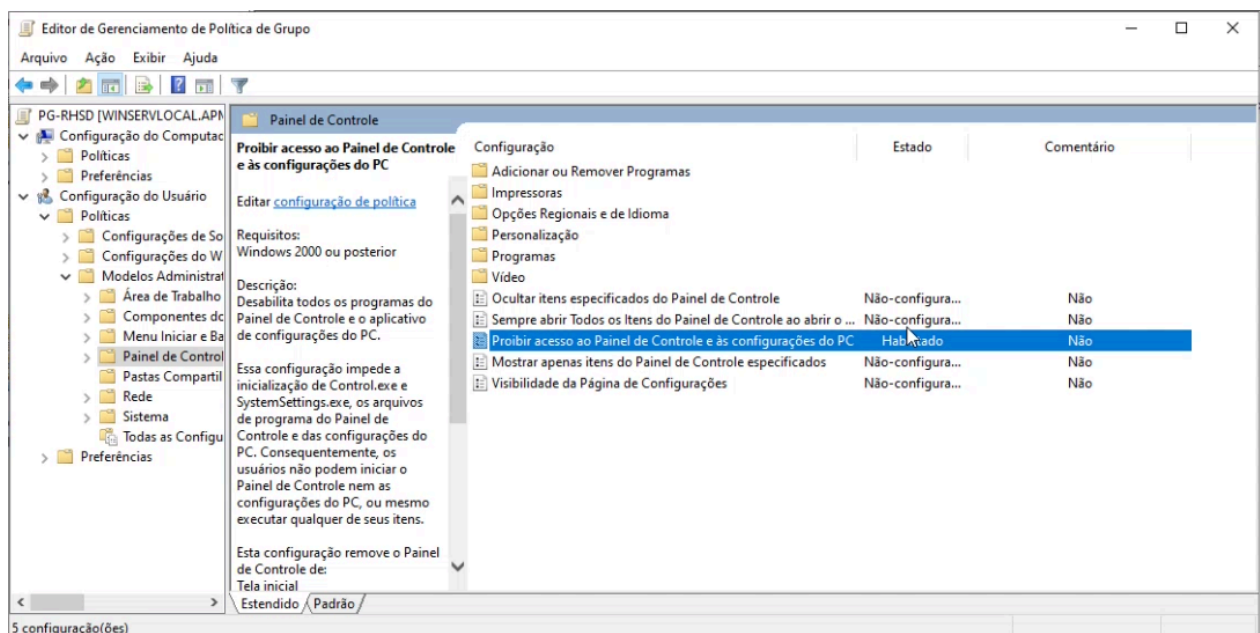


Figura 6 - Exemplo de permissão

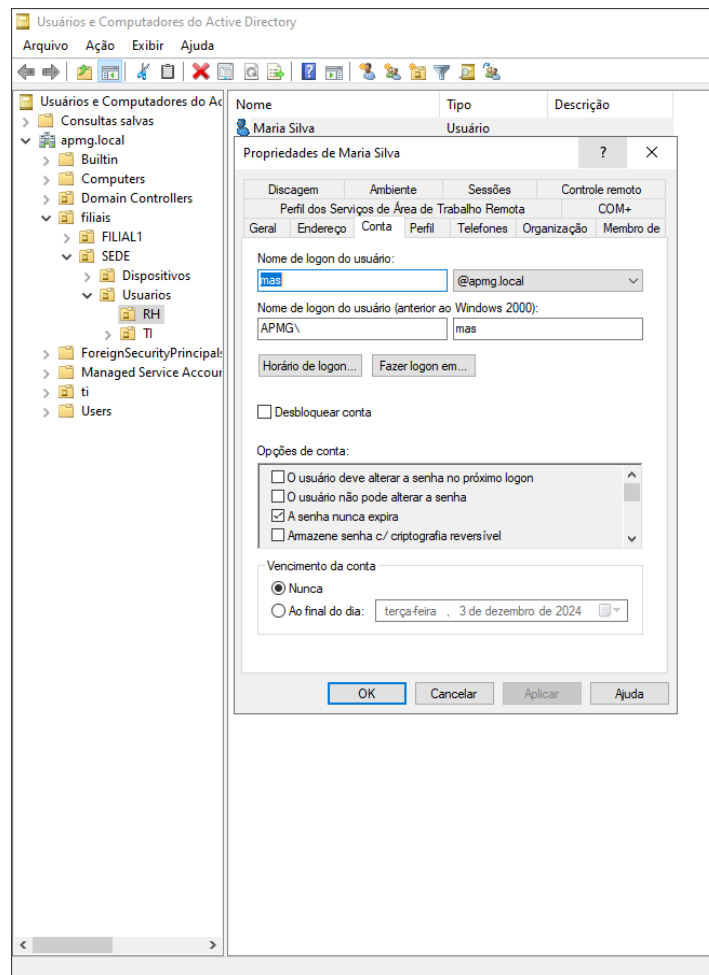


Figura 7 - Exemplo de usuário

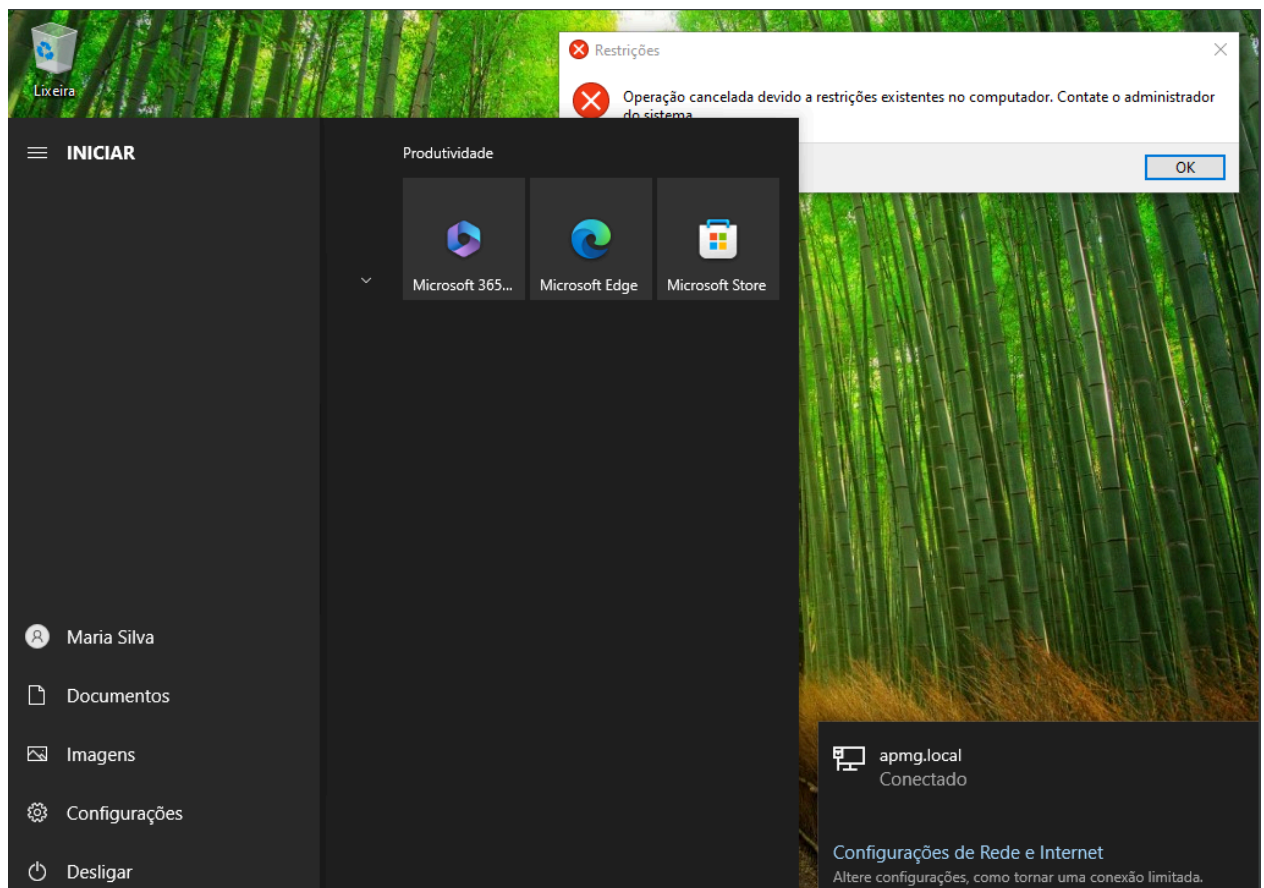


Figura 8 - Falta de permissão