IFPR - Câmpus Pinhais

Bacharelado em Ciência da Computação

Configuração de Serviços

Aléxia de Anhaia Zanon, Guilherme Kazuya Mizutani, Heloisa de Sales Mariano, Matheus Carsten de Oliveira

Professor: Gabriel V. C. Candido

Pinhais, 2025

SUMÁRIO

1 INTRODUÇÃO

A crescente demanda por redes locais flexíveis, seguras e de baixo custo tem incentivado a adoção de soluções alternativas aos roteadores tradicionais. Nesse contexto, computadores comuns podem ser configurados para desempenhar papéis avançados de infraestrutura de rede, oferecendo maior controle, personalização e capacidade de processamento. Sistemas operacionais baseados em Linux, como o Ubuntu, destacam-se nesse tipo de aplicação devido à sua robustez, estabilidade e ampla disponibilidade de ferramentas de rede.

O objetivo deste projeto foi transformar um computador com Ubuntu em um roteador completo e funcional, capaz de criar sua própria rede Wi-Fi, gerenciar a distribuição de endereços IP, controlar o tráfego entre interfaces e assegurar proteção por meio de regras de firewall. Para isso, foram implementados serviços essenciais, como hostapd para criação do ponto de acesso sem fio, dnsmasq para fornecimento de DHCP e DNS, iptables para firewall e NAT, além de ajustes persistentes em arquivos de configuração do sistema para garantir funcionamento contínuo mesmo após reinicializações.

A execução deste trabalho permitiu compreender em profundidade os mecanismos internos de uma infraestrutura de rede, incluindo roteamento entre interfaces, encapsulamento de pacotes, atribuição dinâmica de endereços, tradução de endereços de rede e políticas de filtragem de tráfego. Ao longo do relatório, são descritas as etapas realizadas, as decisões técnicas adotadas, os testes efetuados e as soluções aplicadas para assegurar que o sistema operasse de forma estável, segura e eficiente.

2 CONFIGURAÇÃO DA INFRAESTRUTURA DE REDE

A implementação iniciou-se com o reconhecimento e organização das interfaces de rede presentes no sistema Ubuntu. Esse passo é essencial para definir corretamente quais dispositivos serão responsáveis pela recepção da internet e pela criação da rede local sem fio.

2.0.1 Identificação das Interfaces

Utilizando comandos como `ip a` e `lsusb`, foram identificadas as seguintes interfaces disponíveis no equipamento:

- **wlp63s0**: Interface Wi-Fi interna do notebook, utilizada como interface WAN, responsável por receber acesso à internet.
- **wlxe0d362c98306**: Interface correta do adaptador USB TP-Link usada como ponto de acesso.
- Interface antiga (wlx40ed00f7f304): removida, pois não existia mais.

A partir disso, definiu-se:

- **WAN**: wlp63s0
- **LAN/AP**: wlxe0d362c98306

2.0.2 Configuração do Hostapd

O hostapd foi utilizado para transformar o adaptador TP-Link em um ponto de acesso Wi-Fi. A configuração incluiu:

- Definição do SSID da rede
- Definição da senha WPA2-PSK
- Seleção do canal de operação
- Modo de operação (802.11n/g)
- Associação da interface correta (wlxe0d362c98306)

Comandos utilizados:

```
sudo systemctl enable hostapd
sudo systemctl start hostapd
```

Com isso, o Ubuntu passou a emitir o sinal Wi-Fi através do adaptador USB.

2.0.3 Servidor DHCP (isc-dhcp-server)

Para a distribuição automática de endereços IP na rede local, foi configurado o serviço ISC-DHCP-Server. A sub-rede escolhida para o ambiente interno foi:

```
subnet 192.168.10.0 netmask 255.255.255.0 {
  range 192.168.10.10 192.168.10.50;
  option routers 192.168.10.1;
  option domain-name-servers 8.8.8.8, 1.1.1.1;
}
```

O DHCP foi vinculado à interface do ponto de acesso:

```
INTERFACESv4="wlxe0d362c98306"
```

Após as configurações, o serviço foi reiniciado:

```
sudo systemctl restart isc-dhcp-server
```

Esse passo garantiu que qualquer dispositivo conectado ao AP receba automaticamente IP, gateway e DNS.

2.0.4 Ativação do Roteamento IPv4

Para permitir que o Ubuntu encaminhe pacotes entre WAN e LAN, foi ativado o encaminhamento IPv4:

```
sudo sysctl -w net.ipv4.ip_forward=1
```

E a configuração foi tornada permanente no arquivo `/etc/sysctl.conf`, assegurando que a funcionalidade continue ativa após reinicializações.

2.0.5 NAT e iptables

A rede foi configurada utilizando iptables como firewall principal, sendo responsável tanto pelo roteamento quanto pelo controle de acesso. O UFW não é utilizado para filtragem; seu papel ficou limitado apenas ao controle básico da política de NAT no início do projeto, mas posteriormente toda a lógica de firewall foi migrada para um script manual. O NAT foi configurado manualmente através do iptables, utilizando a tabela nat e a regra de masquerading, permitindo que todos os dispositivos da rede interna (192.168.10.0/24) acessem a Internet através da interface WAN (wlp63s0):

```
sudo iptables -t nat -A POSTROUTING -s 192.168.10.0/24 -o wlp63s0 -j MASQUERADE
```

Essa regra garante que todos os dispositivos conectados ao ponto de acesso criado pelo hostapd saiam para a Internet usando o endereço público/externo do próprio computador Linux.

2.0.6 Configuração do Firewall

O firewall utilizado no projeto foi implementado através de um script próprio chamado ativar-firewall.sh, responsável por aplicar todas as políticas de segurança. Esse script define regras de filtragem, controle de DNS e bloqueio seletivo de sites (Google/YouTube) para determinados hosts.

2.0.6.1 Limpeza e política padrão

O script inicia limpando regras antigas e definindo a política padrão para DROP, garantindo que somente tráfego explicitamente permitido possa passar:

```
iptables -F FORWARD
iptables -P FORWARD DROP
```

2.0.6.2 Bloqueio global de DNS e liberação apenas para IPs autorizados

Primeiro, todo o tráfego DNS é bloqueado:

```
iptables -A FORWARD -p udp --dport 53 -j DROP
iptables -A FORWARD -p tcp --dport 53 -j DROP
```

Depois, apenas dois hosts recebem permissão de usar DNS:

- 192.168.10.10
- 192.168.10.13

```
iptables -A FORWARD -s 192.168.10.10 -p udp --dport 53 -j ACCEPT
iptables -A FORWARD -s 192.168.10.10 -p tcp --dport 53 -j ACCEPT
iptables -A FORWARD -s 192.168.10.13 -p udp --dport 53 -j ACCEPT
iptables -A FORWARD -s 192.168.10.13 -p tcp --dport 53 -j ACCEPT
```

Isso impede que os demais dispositivos utilizem a Internet normalmente.

2.0.6.3 Controle seletivo de acesso ao Google e YouTube

Os IPs 192.168.10.11 e 192.168.10.12 foram configurados para ter acesso somente à infraestrutura do Google/YouTube.

Para isso, foram adicionadas regras liberando apenas os principais blocos de IP pertencentes ao Google:

```
8.8.8.0/24
8.8.4.0/24
142.250.0.0/15
172.217.0.0/16
216.58.0.0/16
```

O script aplica:

```
iptables -A FORWARD -s <IP_BLOQUEADO> -d <REDE_GOOGLE> -j ACCEPT
```

Assim, esses dispositivos conseguem abrir YouTube e serviços do Google, mas não acessam nenhum outro site.

Após isso, todo o resto é bloqueado:

```
iptables -A FORWARD -s 192.168.10.11 -j DROP
iptables -A FORWARD -s 192.168.10.12 -j DROP
```

Função	**Implementação**
NAT	Masquerade via `iptables` na tabela `POSTROUTING`.
DNS restrito	Liberação de DNS apenas para IPs específicos (192.168.10.10 e 192.168.10.13).
Bloqueio seletivo	IPs 192.168.10.11 e 192.168.10.12 podem acessar somente Google/YouTube.
Política padrão	`DROP` (modelo de segurança por padrão fechado).
Script próprio	Firewall aplicado via script `ativar-firewall.sh`.

Tabela 1 – Resumo das funcionalidades implementadas no firewall

- O firewall implementado utiliza exclusivamente iptables, aplicando NAT, controle fino de DNS, bloqueio seletivo de domínios e políticas restritivas de acesso. O uso de um script próprio torna as regras reprodutíveis e facilmente editáveis, além de permitir controle preciso sobre cada dispositivo conectado à rede.

3 IMPLEMENTAÇÃO DO SERVIÇO WEB

Para este trabalho, foi escolhida a implementação de um serviço web acessível aos usuários da rede, com o objetivo de facilitar a visualização do monitoramento do servidor. O serviço principal é a página web (index.html), que integra o Netdata como serviço complementar, permitindo que qualquer dispositivo conectado à rede visualize em tempo real informações de CPU, memória, tráfego de rede e processos, sem necessidade de utilizar o terminal.

3.0.1 Instalação e verificação do Apache2

Para realizarmos nossa aplicação foi utilizado o Apache2 um software de servidor web que recebe requisições HTTP/HTTPS, para realizar a instalação dele foi utilizado o seguinte comando:

```
sudo apt install apache2
```

Após finalizar a instalação, verificamos o status:

```
systemctl status apache2
```

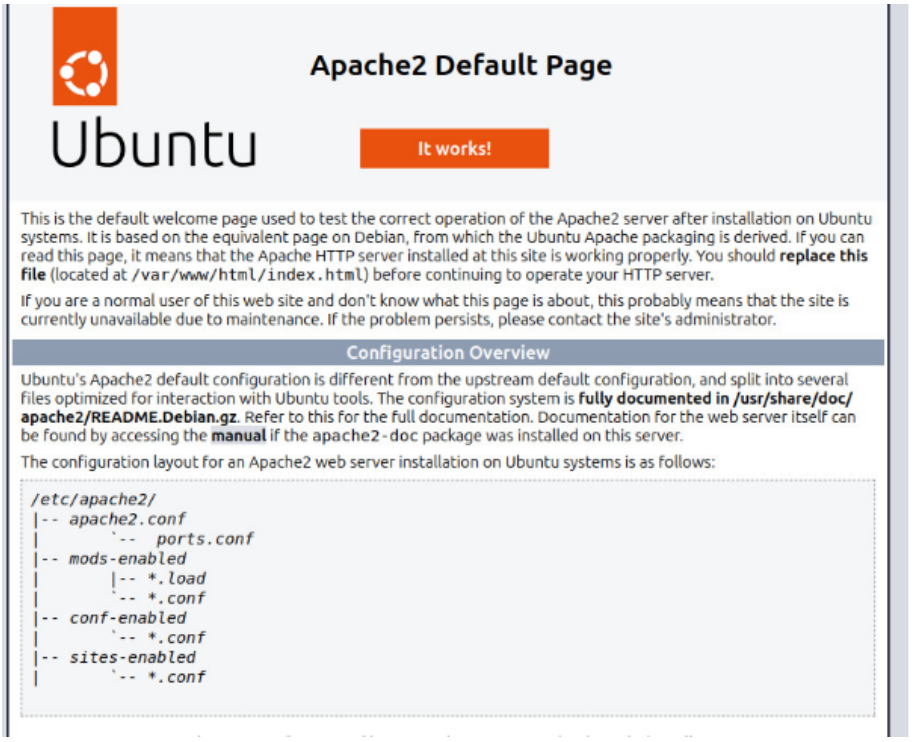
Mostrando o resultado “Ativo”, ou seja nosso servidor web já está funcionando e pronto para ser modificado. Podemos visualizar o servidor inserindo em qualquer navegador:

```
http://IP_DO_SERVIDOR
```

Exemplo:

```
http://192.168.105.1/
```

Ao acessar nosso servidor web pelo IP, podemos ver a tela inicial do apache2



Nossa tela inicialmente é assim pois ainda precisamos programar o front-end da nossa página como desejamos.

3.0.2 Editação do ServidorWeb

Para podermos editar como desejamos precisamos abrir nosso index.html no diretório do Apache:

```
/var/www/html
```

Todos os arquivos HTML colocados nesse diretório podem ser acessados via navegador. Para listar o conteúdo:

```
ls -l /var/www/html
```

Esse comando irá mostrar tudo que está dentro do diretório do apache, assim que visualizamos o arquivo (index.html) podemos editar a partir do seguinte comando

```
sudo nano /var/www/html/index.html
```

No terminal irá aparecer uma página de edição onde podemos editar visualmente nosso site com linguagem de marcação html, como título, cor, tamanho e entre outros Foram alterados:

3.0.3 Criação de fotos e vídeos

Para conseguirmos adicionar fotos e vídeos em nosso site foi feito o seguinte procedimento.

- Baixar arquivos de vídeo e foto desejados.
- Entramos na pasta onde baixamos no terminal , utilizando

  ```
  cd ~/Pasta
  ```

- Copiamos as imagens para o diretório do apache

  ```
  sudo cp nome-da-foto.jpg /var/www/html/
  ```

- Para vídeos é o mesmo processo só mudando de jpg/png para mp4
- Podemos verificar se foi copiado pelo comando

  ```
  ls -l /var/www/html
  ```

- Após isso é em nosso html, inserimos o nome do arquivo e o formato. Exemplo:

  ```
  <img src="nome-da-foto.jpg">
  ```

3.0.4 Criação de páginas

Agora que temos nossa página principal feita como queremos, podemos adicionar outras páginas que são abertas por botões ou atalhos, utilizamos o seguinte comando:

```
sudo nano /var/www/html/mais.html
```

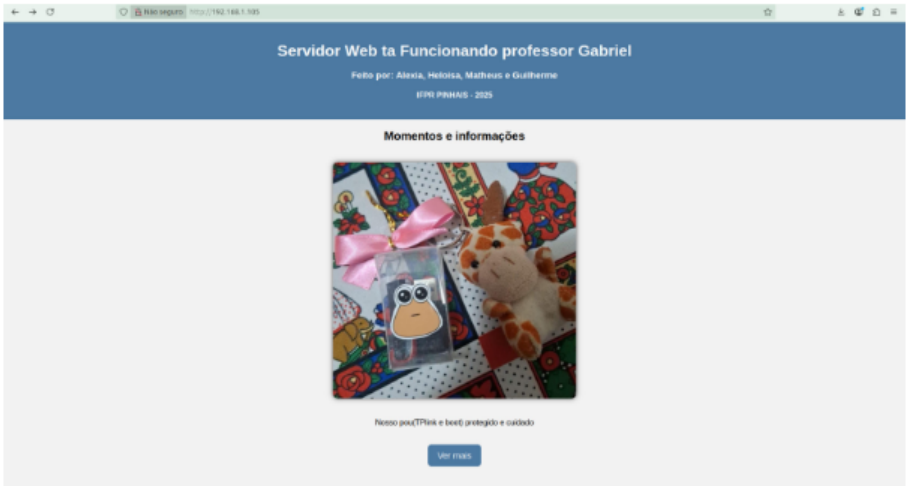
Cria outro arquivo html que podemos referenciar no nosso arquivo principal, ele será usado para andarmos entre as páginas do site.

Após criar editamos o front-end como desejarmos.

Para referenciamos a página que queremos seguir a partir de alguma ação como apertar botão. Utilizamos a seguinte linha de código em nosso arquivo html. Exemplo:

```
<a href="Arquivo.html" class="botao">Ver mais</a>
```

Após todas as configurações e edições, o servidor web exibe a página principal customizada, com links para outras páginas, imagens, vídeos.



4 CONFIGURAÇÃO DO NETDATA E MONITORAMENTO DA REDE

No projeto de monitoramento utilizamos a ferramenta Netdata, escolhida por ser uma solução leve, rápida e altamente intuitiva. Ela foi escolhida, pois como estamos executando o sistema por meio de um boot via pendrive, temos uma limitação de desempenho e armazenamento. Assim, optamos utilizar ela ao invés de outras ferramentas de monitoramento mais pesadas, o Netdata oferece visualização em tempo real sem sobrecarregar o sistema, garantindo eficiência mesmo em hardware mais limitado.

4.0.1 Atualização e Instalação do Netdata

Antes de instalarmos o netdata atualizamos o Ubuntu para garantir as versões mais recentes do pacote evitando erros de dependência. Utilizamos os seguintes comandos para atualizar:

```
sudo apt update
sudo apt install netdata -y
```

Após a atualização ser concluída com sucesso foi realizada a instalação do NetData a partir dos repositórios oficiais. Utilizamos o seguinte comando:

```
sudo apt install netdata -y sudo apt install netdata -y
```

Para vermos se o status do Netdata e garantir que estava funcionando utilizamos:

```
systemctl status netdata
```

Aparecendo “active(running)” significa que está funcionando com sucesso

4.0.2 Configurações do NetData

Agora que concluímos a instalação e tudo está funcionando em perfeito estado, precisamos realizar algumas configurações para acessarmos nosso monitoramento em todos os dispositivos desejados (ex: Celular, Desktop entre outros...)

Utilizamos o comando :

```
sudo nano /etc/netdata/netdata.conf
```

Para acessarmos o painel de configuração do NetData, após isso alteramos a seleção “bind socket ” zerando nosso IP.

- Antes: bind socket to IP = 127.0.0.1
- Depois: bind socket to IP = 0.0.0.0

Essa configuração foi feita para que o Netdata não fique restrito ao localhost (127.0.0.1). Ao usar 0.0.0.0, ele passa a aceitar conexões de qualquer dispositivo dentro da mesma rede (LAN/Wi-fi).

Após isso reiniciamos o serviço para ser iniciado com as novas configurações com sucesso, utilizamos:

```
sudo systemctl restart netdata
```

Para confirmarmos que essa configuração realmente deu certo e que nosso servidor está ‘escutando’ em 0.0.0.0, utilizamos os seguintes comandos:

```
sudo netstat -tulpn | grep 19999
```

Assim retornando:

```
0.0.0.0:19999   LISTEN
```

Significando que a mudança foi aplicada com sucesso.

Somente as configurações do localhost não foram o suficiente para ser liberado o acesso pois havia um bloqueio do FireWall. Para desbloquearmos foi feito os seguintes procedimentos:

4.0.3 Configuração Firewall para DataNet

Após isso foi visualizado que a porta do Netdata(19999) não estava liberada e como estávamos com a política padrão de entrada como "dany" , então mesmo com as configurações anteriores os acessos externos iam ser bloqueados. Para liberarmos utilizamos os comandos:

```
sudo ufw allow 19999/tcp
```

Ativa a porta

```
sudo ufw reload
```

recarregamos o firewall

```
 sudo ufw status verbose
```

listamos novamente para ver se foi ativado com sucesso

Agora o monitoramento via Netdata foi instalado, configurado e liberado com sucesso para acesso em qualquer dispositivo dentro da rede. Todas as etapas foram executadas corretamente, garantindo visibilidade completa do servidor.

4.0.4 Implementação do monitoramento na nossa aplicação Web

Para facilitar a visualização do nosso monitoramento sem a necessidade de ficar executando códigos no terminal, foi realizado uma implementação no nosso arquivo index.html (nosso site), assim quando o usuário acessar nosso site irá aparecer um botão "Abrir painel completo" direcionando ao Netdata.

O primeiro passo é realizar a criação do nosso arquivo html onde ficará o monitoramento, para isso utilizamos o mesmo comando das demais criações html:

```
sudo nano /var/www/html/monitoramento.html
```

pós isso realizamos algumas edições visuais como tamanho,título e entre outros, mas a parte mais importante é o "Iframe", que incorpora os dados do Netdata:

```
<iframe src="http://localhost:19999"></iframe>
```

Utilizamos o Iframe pois o NetData não roda dentro do Apache, pois ele possui um servidor web próprio, assim basta rodarmos na nossa porta (19999)

Agora precisamos inserir nosso monitoramento em nossa página web principal, para isso acessamos nosso “index.html” pelo comando:

```
sudo nano /var/www/html/index.html
```

Para isso basta referenciar o nosso arquivo html criado para o monitoramento (monitoramento.html)

```
<a href="monitoramento.html"
(aqui estamos referenciando qual arquivo html deve ser iniciado)
style="padding:10px 20px;
background:#333; color:white; border-radius:5px; text-decoration:none;">
(configurações visuais)
Monitoramento do Servidor (Título)
</a>
```

Para garantir o funcionamento correto do nosso Netdata e da nossa aplicação Web foi realizado com sucesso , abrimos as configurações do Netdata pelo comando:

```
 sudo nano /etc/netdata/netdata.conf
```

Após isso, localizamos a aba [Web] (localizada abaixo de onde configuramos o bind socket. e fazemos nossa configuração.

```
[web]
    mode = static-threaded
    (Modo padrão, adequado para o boot com pendrive)
    allow connections from = *
    (permite qualquer dispositivo na rede)
    allow dashboard from = *
    (permite que o nosso painel seja carregado dentro do iframe, evitando
    que o netdata bloqueie carregamentos externos por segurança)
    clickjacking protection = allow
    (Liberação dos iframes manualmente pois o Netdata bloqueia)
```

Em resumo o iframe foi utilizado para incorporar o painel do Netdata dentro da nossa página web principal. Ele funciona como uma janela que exibe outro site (no caso, o Netdata rodando na porta 19999) dentro do nosso HTML(index.html). Tivemos que editar o iframe para ajustar o endereço IP, o tamanho e permitir que o painel fosse carregado externamente através da rede, garantindo compatibilidade com celulares e outros computadores.

4.0.5 Implementação do Nmap

Para complementar o monitoramento do servidor, utilizamos o Nmap, que foi utilizado para identificar os dispositivos ativos na rede local, pois o Netdata monitora apenas o servidor e não fornece informações sobre outros dispositivos conectados. Com ele, foi possível mapear todos os IPs ativos sem alterar nenhuma configuração da rede. Para isso realizamos a instalação do Nmap com os seguintes comandos:

```
sudo apt update
sudo apt install nmap -y
```

Após realizar a instalação podemos utilizar o comando:

```
nmap -sn (REDE)
```

Exemplo: nmap -sn 192.168.1.0/24

A opção -sn realiza um "ping scan", mostrando quais dispositivos estão online sem precisar abrir as portas. Exemplo de saída na rede 192.168.1.0/24:

```
 Nmap scan report for _gateway (192.168.1.1)
Host is up
Nmap scan report for 192.168.1.100
Host is up
Nmap scan report for ubuntu (192.168.1.105)
Host is up
Nmap scan report for 192.168.1.109
Host is up
```

Podemos utilizar também o comando:

```
nmap -sn 192.168.1.0/24 | grep "report for"
```

Ele também serve para mostrar os dispositivos online , sua diferença é que ele imprime as informações de uma maneira filtrada e limpa, mostrando apenas as linhas que contêm o "report for" ficando mais fácil visualizar os IPs e Hostnames

Exemplo de saida na rede 192.168.1.0/24:

```
Nmap scan report for _gateway (192.168.1.1)
Nmap scan report for 192.168.1.100
Nmap scan report for ubuntu (192.168.1.105)
Nmap scan report for 192.168.1.109
```

5 OBSERVAÇÃO FINAL

Algumas configurações realizadas neste trabalho, como o endereço IP da rede, as portas do firewall e a interface de rede utilizada, podem variar conforme a máquina em que o Ubuntu com boot persistente esteja sendo executado. Por isso, ajustes específicos podem ser necessários para que os serviços web, o Netdata e a varredura de dispositivos funcionem corretamente em outros equipamentos ou redes.

5.0.1 Máquinas utilizadas no projeto para melhor entendimento do relatório.

- monitoramento e aplicações foi realizada na máquina: 192.168.105.1 ; 192.168.1.0/24
- Os demais serviços foram realizados na máquina:

5.1 DIAGNÓSTICO E CORREÇÕES

Durante os testes, alguns problemas foram identificados:

- O serviço DHCP não estava ativo, impedindo distribuição de IP.
- O firewall continha regras ligadas à interface antiga, causando bloqueio de tráfego.
- A configuração de NAT estava incompleta, impossibilitando acesso à internet.

Cada problema foi corrigido individualmente. Após os ajustes, todos os serviços passaram a operar normalmente e de maneira persistente.

5.2 RESULTADO FINAL

Com todas as configurações implementadas, o sistema apresentou o seguinte conjunto de funcionalidades:

- Ubuntu funcionando como roteador completo;
- Wi-Fi transmitido pelo adaptador TP-Link via hostapd;
- Servidor DHCP distribuindo endereços corretamente na rede interna;
- NAT configurado, permitindo acesso à internet pelos clientes;
- Firewall UFW ajustado com regras coerentes e seguras;

- Serviço web funcional hospedado no Apache2, com página principal e página adicional de conteúdo;
- Integração do site com a página de monitoramento do Netdata;
- Netdata configurado para permitir acesso externo ao painel de monitoramento;
- Uso do Nmap para identificação de dispositivos ativos na rede, auxiliando no diagnóstico e monitoramento;
- Sistema testado tanto em computadores quanto em dispositivos móveis, comprovando o funcionamento prático de todos os serviços.

5.3 REFERÊNCIAS

- TANENBAUM, Andrew S.; WETHERALL, David J. **Redes de Computadores**. 5. ed. São Paulo: Pearson, 2011.
- NEMETH, E.; SNYDER, G.; HEIN, T.; WHALEY, B. **UNIX and Linux System Administration Handbook**. 4. ed. Addison-Wesley, 2010.
- DOCUMENTAÇÃO OFICIAL DO UBUNTU. Disponível em: <https://ubuntu.com>. Acesso em: 27 nov. 2025.
- APACHE HTTP SERVER PROJECT. Disponível em: <https://httpd.apache.org>. Acesso em: 27 nov. 2025.
- NETDATA MONITORING. Disponível em: <https://www.netdata.cloud>. Acesso em: 27 nov. 2025.