UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS REITOR JOÃO DAVID FERREIRA LIMA
CENTRO TECNOLÓGICO
SISTEMAS DE INFORMAÇÃO

Bruno Petri

Desenvolvimento do RFSight: um protótipo de sistema de gerenciamento de dispositivos para provedores de Internet via radiofrequência

Florianópolis, Santa Catarina – Brasil
2025

Bruno Petri

Desenvolvimento do RFSight: um protótipo de sistema de gerenciamento de dispositivos para provedores de Internet via radiofrequência

Trabalho de Conclusão de Curso submetido ao curso de Sistemas de Informação do Departamento de Informática e Estatística, CTC - Centro Tecnológico da Universidade Federal de Santa Catarina como requisito parcial para a obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Carlos Becker Westphall, Dr.
Coorientador: Prof. Wesley dos Reis Bezerra, Dr.

Florianópolis, Santa Catarina – Brasil
2025

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC. Dados inseridos pelo próprio autor.

Petri, Bruno
Desenvolvimento do RFSight: um protótipo de sistema de gerenciamento de dispositivos para provedores de Internet via radiofrequência / Bruno Petri ; orientador, Carlos Becker Westphall, coorientador, Wesley dos Reis Bezerra, 2025.
135 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Sistemas de Informação, Florianópolis, 2025.

Inclui referências.

1. Sistemas de Informação. 2. Redes. 3. Radiofrequência. 4. Provedor. 5. Gerenciamento. I. Westphall, Carlos Becker. II. Bezerra, Wesley dos Reis. III. Universidade Federal de Santa Catarina. Graduação em Sistemas de Informação. IV. Título.

Bruno Petri

Desenvolvimento do RFSight: um protótipo de sistema de gerenciamento de dispositivos para provedores de Internet via radiofrequência

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de Bacharel em Sistemas de Informação e aprovado em sua forma final pelo Curso de Sistemas de Informação – Departamento de Informática e Estatística, CTC - Centro Tecnológico da Universidade Federal de Santa Catarina.

Florianópolis, Santa Catarina – Brasil, 10 de dezembro de 2025.

_____

Coordenação do Curso

Banca examinadora

_____

Prof. Carlos Becker Westphall, Dr.

Orientador

_____

Prof. Wesley dos Reis Bezerra, Dr.

Coorientador

Instituto Federal Catarinense – IFC

_____

Profa. Carla Merkle Westphall, Dra.

Universidade Federal de Santa Catarina – UFSC

Dedico este trabalho à minha família, que sempre foi meu porto seguro e a base de todas as minhas conquistas. Em especial, dedico à memória do meu avô, cujos ensinamentos, amor e exemplo de vida permanecem vivos em meu coração, guiando meus passos e inspirando minha jornada.

AGRADECIMENTOS

Agradeço primeiramente à minha família, pelo amor incondicional, pelo incentivo constante e por sempre acreditarem no meu potencial. O apoio de vocês foi fundamental para que eu pudesse superar os desafios e chegar até aqui.

Aos meus amigos, agradeço pelo companheirismo, pela parceria nos momentos de estudo e pelas risadas que tornaram a caminhada acadêmica mais leve e memorável. A presença de vocês foi essencial para manter o equilíbrio e a motivação ao longo do curso.

Por fim, ao meu orientador, Prof. Dr. Carlos Becker Westphall, e ao meu coorientador, Prof. Dr. Wesley dos Reis Bezerra, expresso minha profunda gratidão pela orientação e pelos valiosos ensinamentos técnicos e acadêmicos durante o processo de desenvolvimento deste trabalho.

RESUMO

O avanço da conectividade rural é fundamental para o desenvolvimento do agronegócio e de serviços essenciais em regiões remotas, sendo as redes via radiofrequência uma alternativa viável e econômica para esse cenário. No entanto, provedores locais que operam esse tipo de infraestrutura enfrentam limitações operacionais relevantes, sobretudo devido à ausência de ferramentas acessíveis e customizáveis para o gerenciamento técnico da rede. Este trabalho tem como objetivo o desenvolvimento de um protótipo de sistema para gerenciamento e monitoramento de dispositivos em redes de radiofrequência, com foco em provedores regionais. A metodologia adotada é de natureza exploratória e abordagem qualitativa, estruturada a partir de revisão sistemática de literatura, análise de ferramentas existentes e aplicação de formulário semi-estruturado a profissionais das áreas de desenvolvimento e negócios em telecomunicações. A partir das respostas obtidas, foram consolidados os requisitos funcionais e não funcionais que guiaram o desenvolvimento da solução. Por fim, o estudo apresenta o processo de concepção, testes e validação inicial do protótipo em ambiente simulado, destacando sua aplicabilidade e potencial de expansão para atender às demandas operacionais e de escalabilidade dos provedores de Internet em áreas rurais.

Palavras-chave: Redes; Radiofrequência; Provedor; Gerenciamento; Sistema.

ABSTRACT

Advancing rural connectivity is essential for the development of agribusiness and the provision of essential services in remote regions. In this context, radio frequency (RF) networks emerge as a viable and cost-effective alternative. However, local providers operating such infrastructure face significant operational challenges, primarily due to the lack of accessible and customizable tools for technical network management. This study aims to develop a prototype system for managing and monitoring devices in RF networks, with a focus on regional Internet service providers (ISPs). The adopted methodology is exploratory in nature, with a qualitative approach, structured around a systematic literature review, analysis of existing tools, and the application of a semi-structured questionnaire to professionals in telecommunications development and business strategy. Based on the collected responses, the functional and non-functional requirements guiding the system's development were consolidated. Finally, this study presents the design process, testing, and initial validation of the prototype in a simulated environment, highlighting its applicability and potential for scalability in addressing the operational needs of rural ISPs.

Keywords: Networks; Radiofrequency; Provider; Management; System.

LISTA DE FIGURAS

Figura 1 – Topologia ponto a ponto via radiofrequência 16
Figura 2 – Enlaces P2P e P2MP 22
Figura 3 – Enlace P2P 25
Figura 4 – Enlace P2MP 26
Figura 5 – Padrão de radiação: Antena Direcional 27
Figura 6 – Padrão de radiação: Antena Omnidirecional 28
Figura 7 – Padrão de radiação: Antena Setorial 29
Figura 8 – Arquitetura SNMP 31
Figura 9 – Arquitetura REST 32
Figura 10 – UNMS/UISP 34
Figura 11 – The Dude 35
Figura 12 – Zabbix para monitoramento de redes 36
Figura 13 – Respostas: Ferramentas de Gerenciamento/Monitoramento 41
Figura 14 – WebSocket vs. HTTP 48
Figura 15 – Arquitetura proposta do RFSight 50
Figura 16 – Geral: Protótipos de telas 52
Figura 17 – Modelagem BD 53
Figura 18 – Entidade *Profile* 55
Figura 19 – Componentes Docker RFSight (simplificado) 56
Figura 20 – Configuração Nginx 57
Figura 21 – Configuração Duck DNS 57
Figura 22 – Certificado RFSight 58
Figura 23 – Tela de Autenticação do Sistema 60
Figura 24 – Inserir *e-mail* para recuperação 61
Figura 25 – *E-mail* de recuperação 61
Figura 26 – Redefinição de senha 62
Figura 27 – Usuários 63
Figura 28 – Criação de usuários 63
Figura 29 – Edição de usuário 64
Figura 30 – Remoção de usuário 65
Figura 31 – *Profiles* 66
Figura 32 – Formulário de *Profile* 67
Figura 33 – Formulário Ação 1 69
Figura 34 – Formulário Ação 2 70
Figura 35 – Mapeamento de *Headers* 71
Figura 36 – Mapeamento de *Query Parameters* e Resposta JSON 72
Figura 37 – Ação de Gerenciamento SSH 73
Figura 38 – Fluxo *DeviceDriver* 75
Figura 39 – Tela de Organizações 76

Figura 40 – Tela de Redes 77
Figura 41 – Formulário de Rede 78
Figura 42 – Tela de Dispositivos 79
Figura 43 – Tela de Adoção 79
Figura 44 – Fluxo *AdoptionController* 81
Figura 45 – Tela de gerenciamento de dispositivos 82
Figura 46 – Fluxo *ConfigController* 83
Figura 47 – *Dashboard* da rede 85
Figura 48 – Monitoramento lista de dispositivos 86
Figura 49 – Rede Principal 89
Figura 50 – Dados de adoção 90
Figura 51 – Validação monitoramento individual 91
Figura 52 – Validação monitoramento rede 91
Figura 53 – Execução da ação de *reboot* 92
Figura 54 – Dispositivo *offline* após *reboot* 93
Figura 55 – Dispositivo *online* novamente 93
Figura 56 – Execução da ação de *change-config* 94
Figura 57 – Configuração atualizada 94
Figura 58 – Confirmar remoção de dispositivo 95
Figura 59 – Dispositivo removido 95

LISTA DE TABELAS

Tabela 1 – Comparativo de Antenas 29
Tabela 2 – Classificação de Funcionalidades por Grau de Importância 42
Tabela 3 – Comparativo com Soluções de Mercado 98
Tabela 4 – Lacunas Identificadas e Contribuição 99

LISTA DE ABREVIATURAS E SIGLAS

ABIMAQ	Associação Brasileira da Indústria de Máquinas e Equipamentos
API	Application Programming Interface
ARP	Address Resolution Protocol
CIDR	Classless Inter-Domain Routing
CNA	Confederação da Agricultura e Pecuária do Brasil
CPE	Customer Premises Equipment
DB	Banco de Dados
DNS	Domain Name System
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
ICR	Indicador de Conectividade Rural
IoT	Internet of Things
ISP	Internet Service Provider
JSON	JavaScript Object Notation
JWT	JSON Web Token
LLDP	Link Layer Discovery Protocol
MAC	Media Access Control
MIB	Management Information Base
P2MP	Ponto-Multiponto
P2P	Ponto a Ponto
PIB	Produto Interno Bruto
PPP	Prestador de Pequeno Porte
RF	Radiofrequência
RX	Receptor
SNMP	Simple Network Management Protocol
SSH	Secure Shell
TLS	Transport Layer Security
TX	Transmissor
UFV	Universidade Federal de Viçosa
UI	Interface de Usuário
VPB	Valor Bruto da Produção Agropecuária

XML	Extensible Markup Language

SUMÁRIO

1. INTRODUÇÃO **16**
2. FUNDAMENTAÇÃO TEÓRICA **20**
2.1. Conectividade Rural no Brasil 20
2.2. Provedores Regionais de Internet (ISPs regionais) 21
2.3. Tecnologias de Radiofrequência 23
2.4. Gerenciamento de Redes e Protocolos 30
3. ESTADO DA ARTE **34**
3.1. Soluções de Mercado - Soluções Correlatas 34
3.2. Trabalhos correlatos 36
4. PROPOSTA **39**
4.1. Levantamento de requisitos 39
4.2. Análise e Definição de Escopo 40
4.3. Definição formal de requisitos e escopo 44
4.4. Arquitetura e tecnologias da solução 46
5. DESENVOLVIMENTO **51**
5.1. Protótipos de telas 51
5.2. Entidades e banco de dados 52
5.3. Arquitetura dos componentes 56
5.4. Características técnicas 59
5.4.1. Gerência e autenticação de usuários 59
5.4.2. Camada de Abstração de Dispositivo 66
5.4.2.1. *Profiles* e ações 66
5.4.2.2. *DeviceDriver* 74
5.4.3. Gerenciamento 76
5.4.4. Monitoramento 84
6. TESTES E VALIDAÇÕES **87**
6.1. Cenário 87
6.2. Testes realizados 88
6.3. Resultados 96
7. ANÁLISE COMPARATIVA **98**
7.1. Comparativo com Soluções de Mercado 98
7.2. Posicionamento Frente aos Trabalhos Correlatos 99
8. CONCLUSÃO **101**

1. INTRODUÇÃO

As redes de radiofrequência têm se estabelecido como uma tecnologia essencial para a expansão do acesso à Internet, particularmente em áreas remotas ou com infraestrutura cabeada limitada. Utilizadas por provedores de pequeno e médio porte, essas redes emergem como uma solução prática e relativamente econômica, porém enfrentam desafios significativos em termos de gerenciamento, monitoramento de dispositivos e otimização do desempenho. Tais dificuldades decorrem de características inerentes à comunicação sem fio, como interferência, variação de sinal e a propagação multipercurso (STALLINGS, 2007; TANENBAUM & WETHERALL, 2011).

Entre tantos casos de uso, um relativamente simples e rentável de implementação para o contexto desses provedores, se trata de uma conexão ponto-a-ponto (Figura 1), que se dá por dois nós com antenas direcionais, apontando um para o outro, utilizando o protocolo IEEE 802.11. O baixo custo comercial do *hardware* necessário, a utilização de antenas direcionais e uma maior potência de transmissão permitem que o protocolo seja utilizado como um meio barato de prover conexões sem fio ponto a ponto por dezenas de quilômetros. (KUROSE & ROSS, 2013).

Figura 1 – Topologia ponto a ponto via radiofrequência

Fonte: KHAN, 2016

De acordo com Stallings (2007), a gestão de redes sem fio requer ferramentas específicas de gerenciamento, capazes de configurar o sistema com um todo, monitorá-lo, identificar falhas, possibilitando reações às mesmas e por fim tornar possível o planejamento de escalabilidade para futuras melhorias. No contexto dos provedores de Internet via radiofrequência, a ausência de tais ferramentas personalizadas cria uma lacuna tecnológica que impacta diretamente a experiência dos usuários e a sustentabilidade dos serviços.

Alinhado a esse contexto, a possibilidade de uma operação ainda mais especializada e técnica dos provedores, incentiva ainda mais o atendimento de áreas remotas e rurais, possibilitando que o cenário de Internet via radiofrequência seja considerado no projeto de ampliação de conectividade rural que busca soluções tecnológicas de baixo custo, proposto, no evento "Soluções para conectividade rural" (2023)[1], pela Confederação da Agricultura e Pecuária do Brasil (CNA). Estima-se que uma maior conectividade pode aumentar o Valor Bruto da Produção Agropecuária (VPB) em R$100 bilhões ao ano, gerando assim um impacto econômico significativo. (CNA & ABIMAQ, 2023).

Diante desse cenário, evidencia-se um problema central: a ausência de sistemas de gerenciamento e monitoramento adaptáveis à realidade de provedores regionais que operam redes via radiofrequência. Esses profissionais, embora atuem em frentes críticas para a inclusão digital, lidam com ferramentas que muitas vezes não atendem à especificidade de suas operações, seja por dependência de fabricantes, limitações de personalização, interfaces pouco amigáveis ou altos custos de licenciamento. A lacuna por soluções tecnológicas mais flexíveis, acessíveis e modulares prejudica a escalabilidade, a manutenção da qualidade de serviço e a sustentabilidade de suas operações.

Nesse contexto, a justificativa para o desenvolvimento deste trabalho reside na necessidade de um sistema de gerenciamento e monitoramento com foco na modularidade, personalização e compatibilidade com múltiplos dispositivos,

[1] Disponível em: <https://www.cnabrasil.org.br/noticias/cna-propoe-grande-projeto-de-estado-para-ampliar-conectividade>. Acesso em: 16 nov. 2024.

especialmente voltado aos provedores regionais, especificamente para redes ponto a ponto (P2P) e ponto multiponto (P2MP). A ausência de soluções com essas características representa uma lacuna no ecossistema de conectividade rural, o que reforça a relevância do desenvolvimento proposto.

1.1. Objetivo geral e específicos

Com isso, o **objetivo Geral** deste trabalho consiste em: desenvolver um protótipo de sistema de gerenciamento e monitoramento de redes radiofrequência ponto a ponto (P2P) e ponto multiponto (P2MP), com foco em ISPs regionais, oferecendo um sistema modular, personalizável, de baixo custo e independente de fabricantes.

Para atingir esse propósito, os **objetivos específicos** estabelecidos são:

- Identificar o estado da arte da área de gerenciamento de redes de radiofrequência;
- Projetar uma solução conceitual para o gerenciamento e monitoramento dinâmico de dispositivos de radiofrequência;
- Estudar e aplicar protocolos de gerenciamento (SNMP, LLDP, SSH, WebSocket, REST, HTTP/HTTPS);
- Desenvolver *Backend* em Python com FastAPI, utilizando autenticação JWT e banco de dados MongoDB;
- Construir *Frontend* em React, com foco em responsividade e usabilidade;
- Implementar abstração de comunicação por API modular, adaptável por dispositivo;
- Realizar testes e validações em ambiente local simulado.

A metodologia adotada assume caráter exploratório, com abordagem qualitativa. Foi realizada uma revisão sistemática de literatura sobre redes de radiofrequência, protocolos de gerenciamento e ferramentas existentes no mercado. Em seguida, aplicou-se um formulário semi-estruturado a profissionais das áreas de gestão de negócios e desenvolvimento de produtos em telecomunicações, com o objetivo de identificar funcionalidades prioritárias em cenários reais de operação. A

análise dessas respostas possibilitou a consolidação do conjunto de requisitos funcionais e não funcionais para orientar o desenvolvimento do sistema.

Ademais, o escopo do trabalho contempla a definição teórica e técnica do projeto, a estruturação dos requisitos, a fundamentação das decisões de arquitetura, etapas de preparação (prototipação de telas e modelagens), o desenvolvimento do sistema, e por fim os testes e validações.

1.2. Organização de capítulos

Este trabalho está organizado da seguinte forma: o Capítulo 2 apresenta a fundamentação teórica, com os principais conceitos técnicos relacionados a redes de rádio frequência, protocolos, dispositivos e tecnologias utilizadas. O Capítulo 3 traz o estado da arte, abordando ferramentas comerciais e soluções acadêmicas correlatas. O Capítulo 4 detalha a proposta de solução, incluindo sua estrutura técnica, levantamento de requisitos e escopo de implementação definido. Os Capítulos 5 e 6 tratam respectivamente do desenvolvimento e testes do protótipo. O Capítulo 7 traz uma análise comparativa do protótipo desenvolvido com o estado da arte. Por fim, o Capítulo 8 apresenta a conclusão juntamente dos trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Para compreender os desafios enfrentados por provedores de Internet via rádio frequência e a necessidade de ferramentas específicas de gerenciamento, é essencial analisar o contexto da conectividade rural no Brasil, as características dos ISPs regionais e, os fundamentos das redes radiofrequência e as tecnologias envolvidas.

2.1. Conectividade Rural no Brasil

Atualmente, a tecnologia é primordial para o avanço dos diversos setores que compõem a economia brasileira e mundial. Ela está presente no dia a dia de milhares de pessoas, e ainda mais com o avanço da conectividade da população, mais e mais serviços se tornam “inteligentes”, a ponto de terem melhor rastreabilidade, monitoramento e precisão, por exemplo. Uma aplicação prática disso é vista no contexto do agronegócio, pilar da economia que responde por uma fatia substancial do Produto Interno Bruto (PIB) do Brasil (AGRIHUB, 2024), onde a conectividade alinhada ao uso de tecnologias IoT permite que a produção se torne mais eficiente e precisa, refletindo assim diretamente no VPB, como apontam a CNA e a ABIMAQ (2023).

No entanto, a conectividade no meio rural brasileiro é um desafio persistente, impactando diretamente a produtividade agrícola, o acesso à educação, saúde e outros serviços essenciais. Segundo dados do Ministério da Agricultura e Pecuária (2022), mais de 70% das propriedades rurais no país ainda permanecem *offline*, sem qualquer conexão com a Internet. Ademais, de acordo com o Indicador de Conectividade Rural (ICR)[2], desenvolvido pela Associação ConectarAGRO e pela Universidade Federal de Viçosa (UFV), somente 33,9% da área disponível para uso agrícola no Brasil tem cobertura 4G ou 5G e também, apenas 48,1% dos imóveis rurais brasileiros têm cobertura 4G ou 5G em toda área de uso agropecuário.

[2] Disponível em: <https://www.conectaragro.com.br/indicadordeconectividaderural>. Acesso em: 17 mai. 2025.

Para viabilizar a expansão da conectividade rural, especialmente com foco no agronegócio, é ideal que sejam utilizadas tecnologias de conexão de longo alcance, adaptadas às condições geográficas específicas de cada região rural. Isso pode incluir a utilização de diferentes tipos de conexão, como rádio, satélite, ou tecnologias móveis como 3G/4G (CONECTARAGRO, 2024), e mais atualmente inclusive o 5G. Cada uma dessas soluções apresenta diferentes custos, requisitos técnicos e prazos de implementação.

Nesse contexto, as redes via rádio tornam-se especialmente vantajosas por permitirem a rápida instalação de enlaces ponto a ponto, a partir do uso de equipamentos comercialmente acessíveis, que não necessitam de infraestrutura cabeada quilométrica. Ainda, o fato de não depender diretamente de políticas públicas e grandes operadoras, como no caso da tecnologia celular (3G/4G/5G), permite que pequenos e médios provedores de Internet tornem-se pontos chaves, atuando com maior autonomia e agilidade na expansão da conectividade rural, contribuindo diretamente para o desenvolvimento do agronegócio. Dessa forma, segundo Bertollo (2021, p. 7),

> "[...] os provedores regionais têm a possibilidade de conectar os produtores rurais e suas propriedades à Internet, para que progressivamente sejam utilizadas tecnologias informacionais para realizar a detecção, controle e monitoramento da produção, bem como avaliar o processamento de alimentos, prever variáveis meteorológicas, controlar pragas, etc."

2.2. Provedores Regionais de Internet (ISPs regionais)

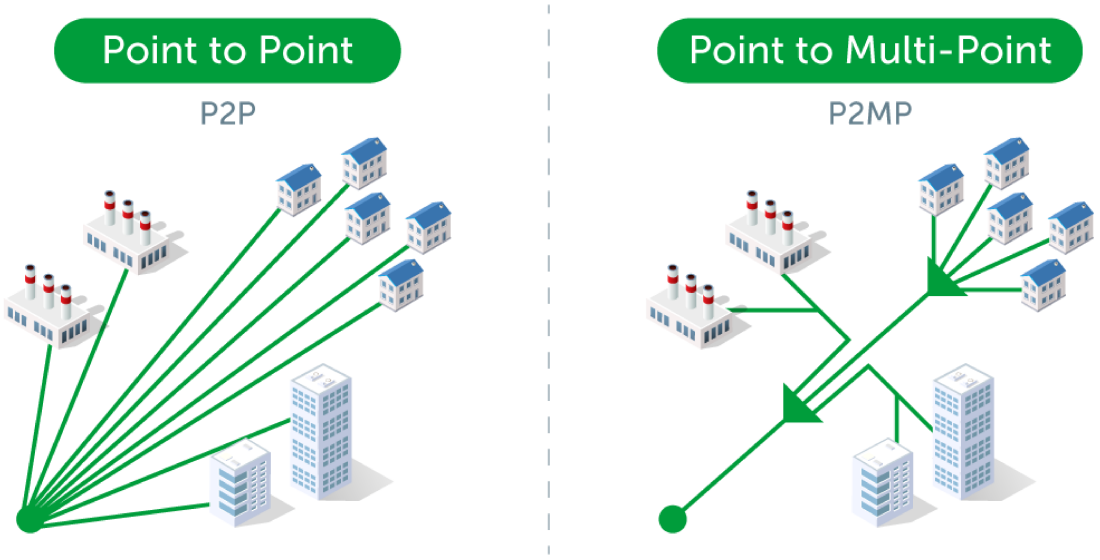
Os provedores de Internet, também conhecidos como *Internet Service Providers* (ISPs), têm papel fundamental na distribuição de conexão em áreas não atendidas pelas grandes operadoras, e também para promover a concorrência, algo valioso para o consumidor. De acordo com dados dos paineis de outorga e licenciamento da ANATEL, por volta de 12 mil ISPs estão propriamente licenciados no Brasil.

A infraestrutura de um provedor exige conectividade constante com a Internet. Para isso, o *backbone* é composto por um ou mais *links* dedicados, servidores responsáveis por serviços essenciais de autenticação, controle de tráfego,

segurança, banco de dados de clientes, sistema financeiro e gerenciamento de conexões (POSSEBON, 2014).

No contexto das áreas rurais, esses provedores costumam utilizar tecnologias de rádio (*wireless*), pelo custo reduzido e alcance considerável, como enlaces P2P e ponto multiponto P2MP, baseados no padrão IEEE 802.11. A Figura 2 demonstra ambos os enlaces.

Figura 2 – Enlaces P2P e P2MP



Fonte: Intelbras, 2021

Conforme apresentado na Figura 2, o enlace P2MP envolve a conexão de diversos dispositivos à uma *basestation*, por isso o nome ponto (*basestation*) multiponto ("n" dispositivos), esses que geralmente são chamados de CPEs (*Customer Premises Equipment*) ou clientes. Com essa topologia, um provedor pode distribuir conectividade de uma *basestation* para diversos clientes a quilômetros de distância, desde que exista visada. O enlace P2P é utilizado quando envolve a conexão entre dois dispositivos, para levar a conectividade de um ponto a outro específico. Ambas as topologias são vantajosas para os ISPs, sendo a primeira considerada a melhor, dado que uma *basestation* pode conectar diversas áreas dentro de seu alcance.

No entanto, os *ISPs* enfrentam limitações técnicas significativas, como falta de ferramentas adequadas de gestão, dependência de configuração manual e pouca visibilidade em tempo real sobre a saúde da rede. Essas limitações podem estar diretamente ligadas não somente ao orçamento desses provedores, que é consideravelmente menor quando comparado ao das grandes operadoras, mas também à disponibilidade de ferramentas específicas no mercado.

Ademais, destaca Humberto Calza, coordenador de processos da Anatel, durante o Abrint Global Congress 2025 (TELESÍNTESE, 2025),

> "[...] o Brasil é o maior país de PPPs (prestadores de pequeno porte) do mundo. Segundo o BNDES, 90% dos provedores se enquadram como micro ou pequenas empresas, com faturamento inferior a R$ 5 milhões ao ano."

Com operações enxutas e equipes técnicas reduzidas, é comum que o suporte aos clientes e o monitoramento da infraestrutura sejam feitos de forma mais próxima. Isso possibilita um tempo de atendimento menor, porém pode degradar qualidade do serviço contínuo e a escalabilidade do negócio. Portanto, é essencial o uso de soluções/ferramentas acessíveis e customizadas, que permitam uma melhor gerência de serviços para o provedor.

2.3. Tecnologias de Radiofrequência

As tecnologias de radiofrequência são fundamentais para a viabilização da comunicação de dados em regiões de difícil acesso, sendo amplamente utilizadas por ISPs em áreas rurais. Essa seção descreve os principais fundamentos físicos e técnicos que sustentam a operação de redes sem fio baseadas em RF.

2.3.1. Comunicação de dados sem fio

O uso da comunicação de dados pelo meio *wireless* se dá principalmente em locais onde a infraestrutura cabeada se torna inviável. Dessa forma a conectividade pode chegar a locais distantes de maneira mais eficiente e acessível. Uma das principais, se não a mais importante, formas de propagação para redes RF é conhecida como *Line-of-Sight Propagation* (Propagação por Linha de Visada), que basicamente explica que toda forma de comunicação com frequências acima de 30 MHz, deve ser realizada por meio da existência de uma linha de visada efetiva entre

as antenas transmissoras (TX) e as receptoras (RX). O termo “efetiva” é usado pois as micro-ondas de sinal são dobradas ou sofrem refração pelo efeito da atmosfera e curvatura da Terra (STALLINGS, 2014).

A visada direta entre os dispositivos é essencial, principalmente em frequências mais altas, como 5 GHz, que oferecem maior largura de banda, mas exigem trajetos livres de obstáculos (visada efetiva limpa), pois sofrem com maior *path loss*, motivo pelo qual o alcance é consideravelmente reduzido. Já a faixa de 2,4 GHz possui maior penetração, sendo mais tolerante a barreiras, embora esteja mais sujeita à interferência por ser amplamente utilizada (a maioria dos dispositivos opera em 2.4GHz, causando uma poluição de sinal no meio).

No entanto, a estabilidade do enlace não depende apenas da visada direta e da frequência utilizada. Diferentes fenômenos físicos interferem diretamente na propagação do sinal e devem ser considerados no planejamento e operação de redes de rádio frequência, especialmente em cenários de longa distância.

Entre esses efeitos, a refração se destaca por modificar a trajetória da onda ao atravessar meios com diferentes densidades, como camadas atmosféricas de temperaturas distintas. Esse fenômeno altera a velocidade e a direção da onda, podendo estender o alcance do sinal em certos contextos ou causar perdas significativas quando o desvio é excessivo.

Além disso, é comum que o sinal encontre superfícies reflexivas ao longo do caminho, como estruturas metálicas, telhados ou corpos d’água. Nesses casos, a reflexão redireciona parte da energia da onda de volta à origem ou para trajetórias alternativas, gerando ecos ou interferências por múltiplos caminhos (*multipath*), que afetam a integridade dos dados transmitidos.

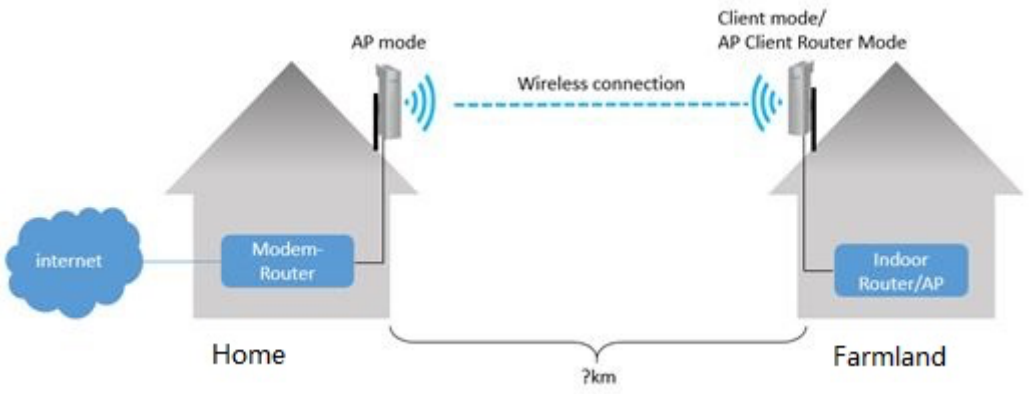
Já a difração permite que o sinal contorne obstáculos ou se propague por aberturas estreitas. Embora contribua para manter a conectividade mesmo sem visada direta perfeita, esse efeito reduz a potência útil do sinal recebido e pode gerar distorções, sendo mais relevante em faixas de frequência mais baixas, como 900 MHz ou 2,4 GHz.

Portanto, compreender e considerar a influência desses fenômenos (refração, reflexão e difração), é fundamental para dimensionar corretamente os enlaces de rádio frequência, orientar o posicionamento de antenas e escolher a faixa espectral mais adequada para cada cenário.

2.3.2. Topologias

As redes *wireless* operadas por provedores regionais geralmente utilizam uma combinação das topologias P2P e P2MP, de acordo com a área a ser coberta e o tipo de serviço fornecido. No caso da topologia P2P, o enlace é estabelecido entre dois dispositivos com antenas direcionais, geralmente parabólicas ou painéis de alto ganho, configurados para estabelecer uma conexão direta e dedicada. Essa estrutura é ideal para aplicações que exigem desempenho elevado e confiável, como a extensão de *backbone* até localidades remotas. Em regiões onde a infraestrutura cabeada é inviável, esse modelo se torna fundamental para garantir conectividade.

Figura 3 – Enlace P2P



Fonte: HAPPYNET, 2025

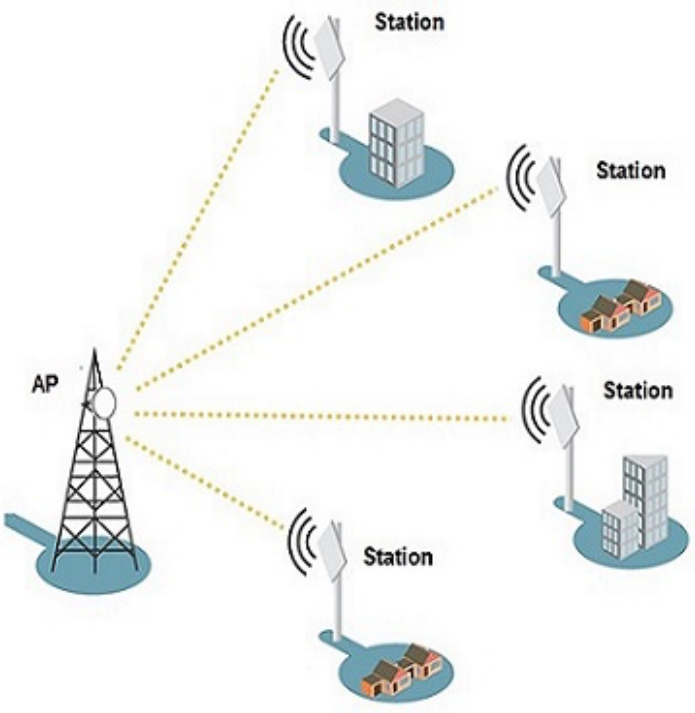
A Figura 3 ilustra um exemplo típico da aplicação da topologia P2P, destacando a simetria e o foco do enlace, bem como a importância da visada direta e consequentemente da configuração adequada das antenas para assegurar a integridade do sinal. Com o uso dessa topologia, é possível levar conectividade de

um ponto a outro sem a necessidade de cabeamento no caminho e com o *link* de Internet apenas no nó com dispositivo em modo AP.

Por sua vez, a topologia P2MP é amplamente empregada na distribuição de sinal a partir de um ponto central para múltiplos clientes espalhados por uma área geográfica. Nessa configuração, uma torre equipada com uma ou mais *basestations*, geralmente com antenas setoriais com abertura angular de 60°, 90° ou 120°, transmite o sinal para diversos dispositivos cliente (CPE). Cada CPE é instalado nas instalações dos assinantes finais e se comunica com a estação base, desde que haja visada efetiva.

A principal vantagem da topologia P2MP está na sua eficiência de infraestrutura: com uma única estação base, é possível atender dezenas de clientes simultaneamente, otimizando os custos com equipamentos e instalações, conforme aponta a Figura 4.

Figura 4 – Enlace P2MP



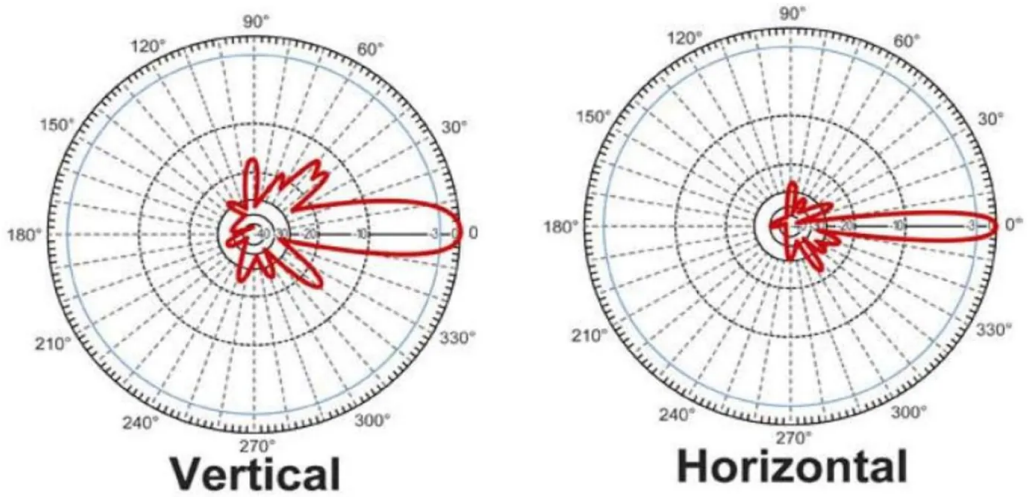
Fonte: HAPPYNET, 2025

A Figura 4 representa visualmente uma aplicação comum da topologia P2MP em um cenário de provedor regional. É possível observar como uma única torre pode cobrir uma região extensa, conectando vários pontos espalhados e ilustrando a escalabilidade dessa abordagem, considerando claro o planejamento adequado de canal, potência e posicionamento de antenas que garantem a visada efetiva entre *basestation* e CPEs.

2.3.3. Antenas

As antenas utilizadas em redes de RF influenciam diretamente o desempenho do enlace. Antenas direcionais são utilizadas em enlaces ponto a ponto por oferecerem maior alcance e menor interferência lateral. As setoriais cobrem ângulos específicos (geralmente entre 60° e 120°) e são indicadas para o modelo P2MP, permitindo a cobertura de setores distintos a partir de uma única torre. Já as omnidirecionais irradiam o sinal em 360° no plano horizontal e são usadas em ambientes urbanos ou de curta distância.

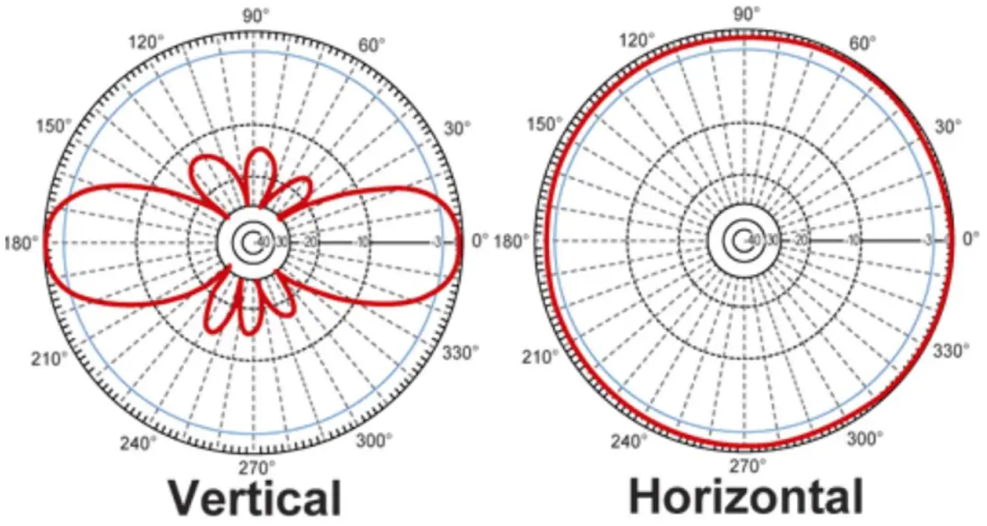
Figura 5 – Padrão de radiação: Antena Direcional



Fonte: NGUYEN, 2021

Conforme apresentado na Figura 5, o feixe e ângulo de abertura de radiação de uma antena direcional são bem pequenos, por isso essas antenas tendem a ser utilizadas para longas distâncias, considerando a concentração de seu ganho em praticamente uma direção.

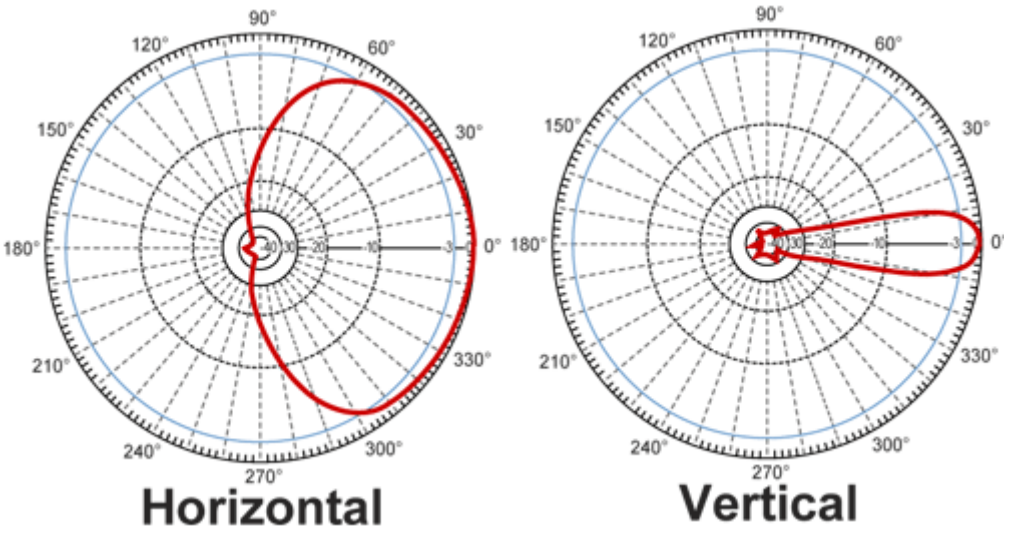
Figura 6 – Padrão de radiação: Antena Omnidirecional



Fonte: NGUYEN, 2021

A Figura 6 demonstra o feixe de uma antena omnidirecional, o qual ocupa 360° do plano horizontal, isto é, irradia horizontalmente para todas as direções. Contudo, devido à dispersão da energia em todas as direções, o alcance tende a ser mais limitado em comparação com antenas direcionais. Dessa forma, seu principal caso de uso é para curtas distâncias. Em cenários P2MP, por exemplo, antenas omnidirecionais podem ser úteis em torres menores com raio de cobertura reduzido, principalmente em locais com poucos obstáculos e clientes próximos ao ponto de acesso. Vale citar que esse tipo de antena é comumente utilizado para roteadores e pontos de acesso *indoor*.

Figura 7 – Padrão de radiação: Antena Setorial



Fonte: WIFI BOOSTER, 2025

Conforme demonstrado na Figura 7, o feixe de uma antena setorial não é tão fechado como a de uma direcional, nem tão aberto como a omnidirecional, cobrindo realmente um setor do plano horizontal, garantindo um alcance e cobertura de mediano à bom. A escolha do tipo de antena deve considerar o padrão de radiação desejado, o ambiente e o objetivo da transmissão. A Tabela 1 a seguir resume as principais características das antenas mais comuns em redes RF:

Tabela 1 – Comparativo de Antenas

Tipo	Direcionalidade	Principal Aplicação	Ganho médio (dBi)
Direcional	Feixe estreito (10°–30°)	P2P	15 a 30
Setorial	Setor (60°–120°)	P2MP	10 a 18
Omnidirecional	360° (plano horizontal)	Curtas distâncias	5 a 9

Fonte: Autor

A Tabela 1 permite visualizar claramente as diferenças entre direcionalidade, aplicação e ganho oferecidos por cada tipo de antena. Essa compreensão é essencial para dimensionar corretamente uma rede sem fio eficiente e funcional, considerando a topologia desejada e as restrições operacionais do ambiente, dado

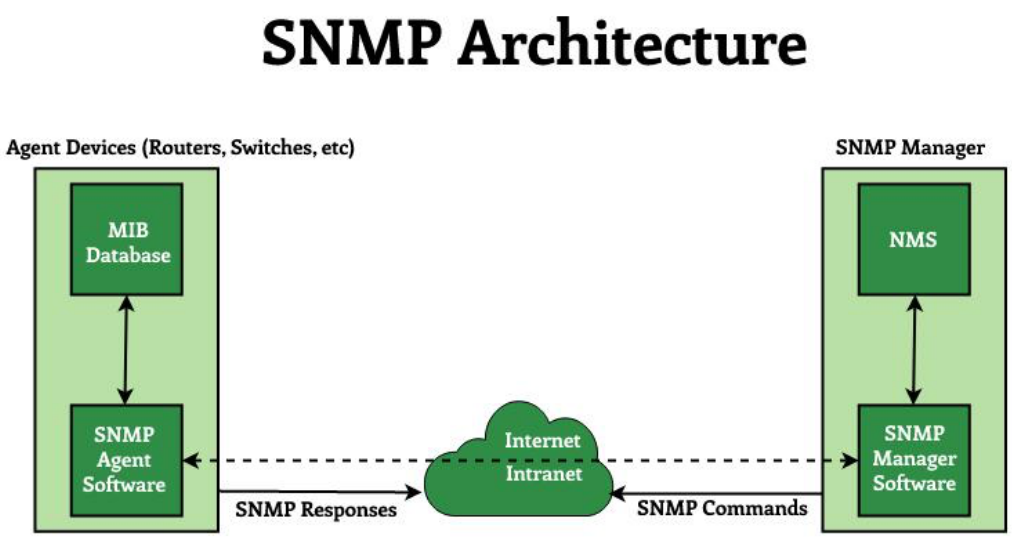
que as antenas direcionais e setoriais podem ter um alcance de quilômetros, em comparação às omnidirecionais que geralmente cobrem uma área inferior a 1 km.

2.4. Gerenciamento de Redes e Protocolos

O gerenciamento eficiente de dispositivos em redes de rádio frequência exige o uso de protocolos padronizados que permitam monitoramento remoto, coleta de dados, automação de tarefas e diagnósticos. Neste contexto, destacam-se os protocolos SNMP (*Simple Network Management Protocol*), LLDP (*Link Layer Discovery Protocol*), SSH (*Secure Shell*), REST APIs, ARP (*Address Resolution Protocol*), ICMP (*Internet Control Message Protocol*) e HTTP/HTTPS. Essa seção compreende os protocolos acima listados, necessários para lidar com a gerência robusta de uma topologia de rede radiofrequência, no contexto deste trabalho.

Primeiramente, o SNMP (*Simple Network Management Protocol*) é um dos principais protocolos utilizados para gerência de rede. Sua arquitetura baseia-se no modelo *manager-agent*, em que um gerenciador central coleta informações de dispositivos gerenciados, denominados agentes. Essas informações são organizadas em MIBs (*Management Information Bases*), que estruturam os dados disponíveis em cada equipamento. Segundo a RFC 1157, o SNMP permite o monitoramento de variáveis como sinal de rádio, tráfego, tempo de atividade (*uptime*) e uso de CPU, sendo amplamente utilizado em ferramentas como Zabbix, por exemplo. A Figura 8 demonstra a arquitetura base do SNMP.

Figura 8 – Arquitetura SNMP



Fonte: GEEKSFORGEEKS, 2024

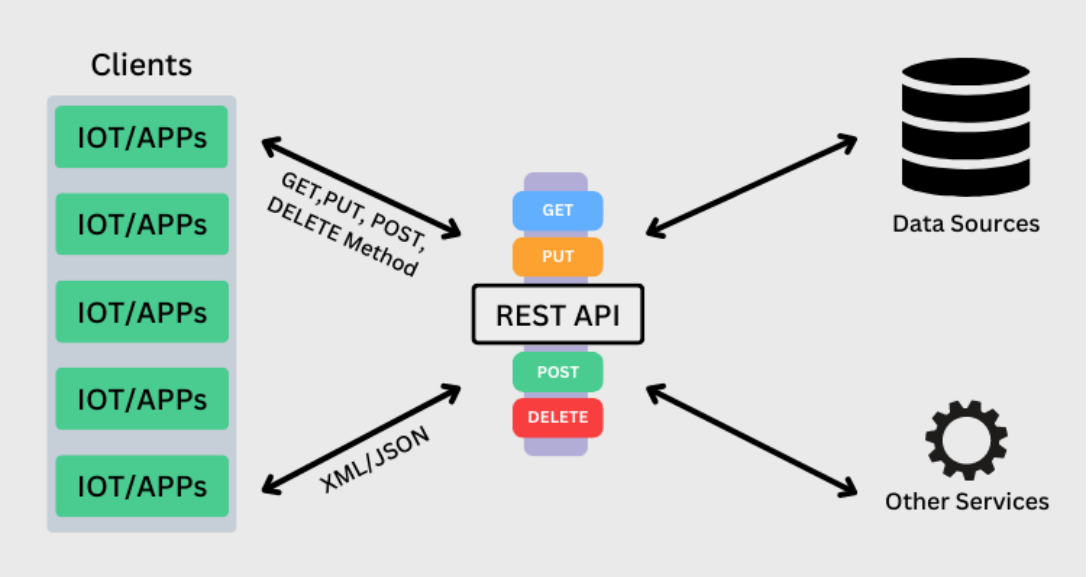
Outro protocolo relevante para a estruturação de redes é o LLDP (*Link Layer Discovery Protocol*), padronizado pela norma IEEE 802.1AB. Ele opera na camada de enlace e permite que dispositivos disponibilizem sua identidade, capacidades e conectividade para os vizinhos na rede. Por meio da troca periódica de pacotes, *switches*, rádios e roteadores podem ser automaticamente descobertos e mapeados, o que facilita o diagnóstico de problemas topológicos e a atualização da base de dados de dispositivos ativos.

Para acesso remoto seguro aos dispositivos, execução de comandos e busca de informações, geralmente utiliza-se o SSH (*Secure Shell*), definido pela RFC 4251. O SSH substitui protocolos inseguros como *Telnet* (onde os dados trafegam em texto plano), proporcionando confidencialidade, autenticação forte e integridade dos dados trafegados. Em ambientes de rede, o SSH permite a execução de comandos em terminais remotos, alteração de configurações e coleta de dados dos dispositivos. É uma ferramenta essencial tanto para administradores quanto para *scripts* automatizados de gerência.

O uso de REST APIs tem crescido significativamente em soluções de gerenciamento, permitindo a comunicação entre sistemas por meio do protocolo

HTTP e dados estruturados em JSON ou XML. Segundo Fielding (2000), REST é definido por um conjunto de restrições arquiteturais que proporcionam escalabilidade, independência de cliente e servidor e *interface* uniforme. Muitos dispositivos e sistemas modernos expõem APIs REST para integração com sistemas de monitoramento personalizados, permitindo a leitura e escrita de dados operacionais diretamente pela aplicação.

Figura 9 – Arquitetura REST



Fonte: DREAMFACTORY, 2024

Ademais, o ARP (*Address Resolution Protocol*), padronizado pela norma RFC 826, é um protocolo essencial de camada de enlace utilizado para mapear endereços IP para endereços físicos (MAC) em redes locais. Em redes de rádio frequência, especialmente em topologias P2P e P2MP, o ARP assegura que os pacotes destinados a um endereço lógico sejam corretamente roteados até o dispositivo físico correspondente. Sem ele, dispositivos não conseguiriam localizar uns aos outros em uma sub-rede, o que inviabilizaria a comunicação mesmo em redes de curto alcance, inclusive as cabeadas.

O ICMP (*Internet Control Message Protocol*), definido pela RFC 792, é utilizado para o envio de mensagens de erro, bem como testes e diagnósticos de conectividade, se tornando indispensável para identificar falhas de comunicação entre rádios e detectar instabilidades nos enlaces por meio de ferramentas como *ping* e *traceroute*, e se torna essencial para verificar o *status* de conectividade dos

dispositivos. A simplicidade e baixa sobrecarga do protocolo o tornam adequado para verificação constante do estado da rede em sistemas de gerenciamento automatizados.

Ainda, os protocolos HTTP (*Hypertext Transfer Protocol*) e HTTPS (versão segura, com criptografia TLS) , amplamente utilizados para interação com a *interface* gráfica dos dispositivos, também servem como base para o funcionamento das APIs REST e sistemas embarcados com servidores web. Em soluções embarcadas, como rádios e roteadores, é extremamente comum encontrar servidores web internos que expõem informações e controles via páginas HTML autenticadas e acessadas por HTTP/HTTPS. A adoção do HTTPS é especialmente crítica para segurança, pois garante integridade e confidencialidade dos dados trafegados durante operações sensíveis como autenticação, configuração e atualização de firmware.

A combinação desses protocolos forma a base técnica de gerenciamento de redes para o desenvolvimento do RFSight, ao integrar a capacidade de leitura granular do SNMP, a execução segura de comandos via SSH, a interoperabilidade das APIs REST e a descoberta de nível de enlace via ARP/ICMP, o sistema transcende as limitações de uma abordagem monolítica. Essa fusão permite consolidar múltiplas metodologias de coleta e controle em um único protótipo modular, capaz de adaptar-se dinamicamente às especificidades de cada dispositivo sem sacrificar a padronização necessária para uma gestão centralizada e eficiente.

3. ESTADO DA ARTE

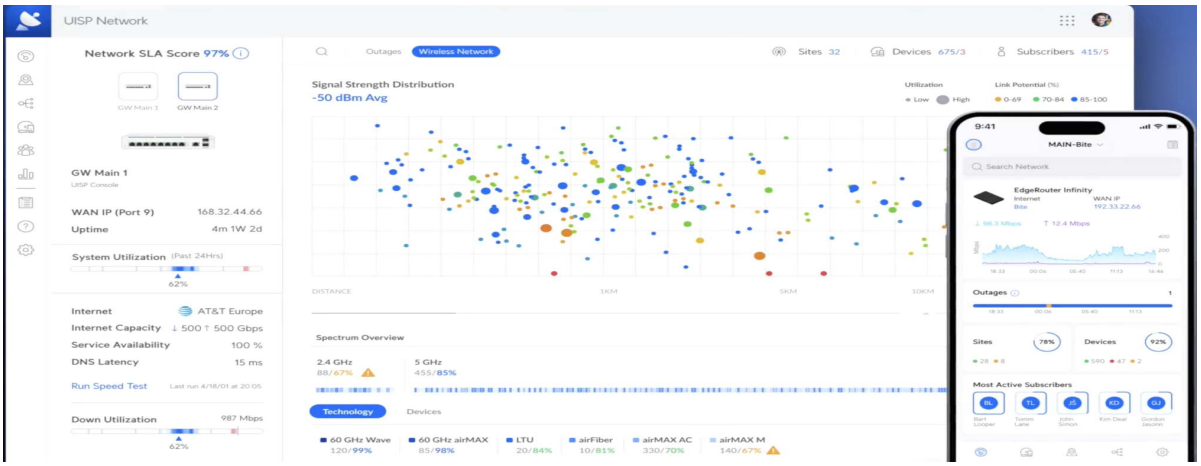
Esta seção analisa as principais soluções de mercado relacionadas com gerência de dispositivos de rede e examina trabalhos acadêmicos correlatos, destacando as lacunas que justificam o desenvolvimento do RFSight, bem como prepara o terreno para a análise comparativa realizada no capítulo 7.

3.1. Soluções de Mercado - Soluções Correlatas

Existem inúmeras ferramentas e soluções que se propõem ao gerenciamento e monitoramento de redes no mercado, porém existem algumas consideravelmente mais bem consolidadas. Nesta seção iremos tratar do The Dude (Mikrotik), UNMS/UISP (Ubiquiti) e Zabbix.

O UNMS/UISP, da Ubiquiti, é uma solução voltada para redes de dispositivos da própria marca (exclusivamente). A ferramenta oferece visualização topológica automatizada, monitoramento em tempo real, controle de clientes e geração de alertas. Embora seja gratuita e possua versão local e em nuvem, sua limitação principal reside na dependência da linha de produtos Ubiquiti, tornando-a pouco versátil para redes mistas, e não tão atrativa pelo custo elevado dos dispositivos da marca, que pode ser inviável para os ISPs em questão. A Figura 10 demonstra a página principal do sistema, bem como o acesso *mobile*.

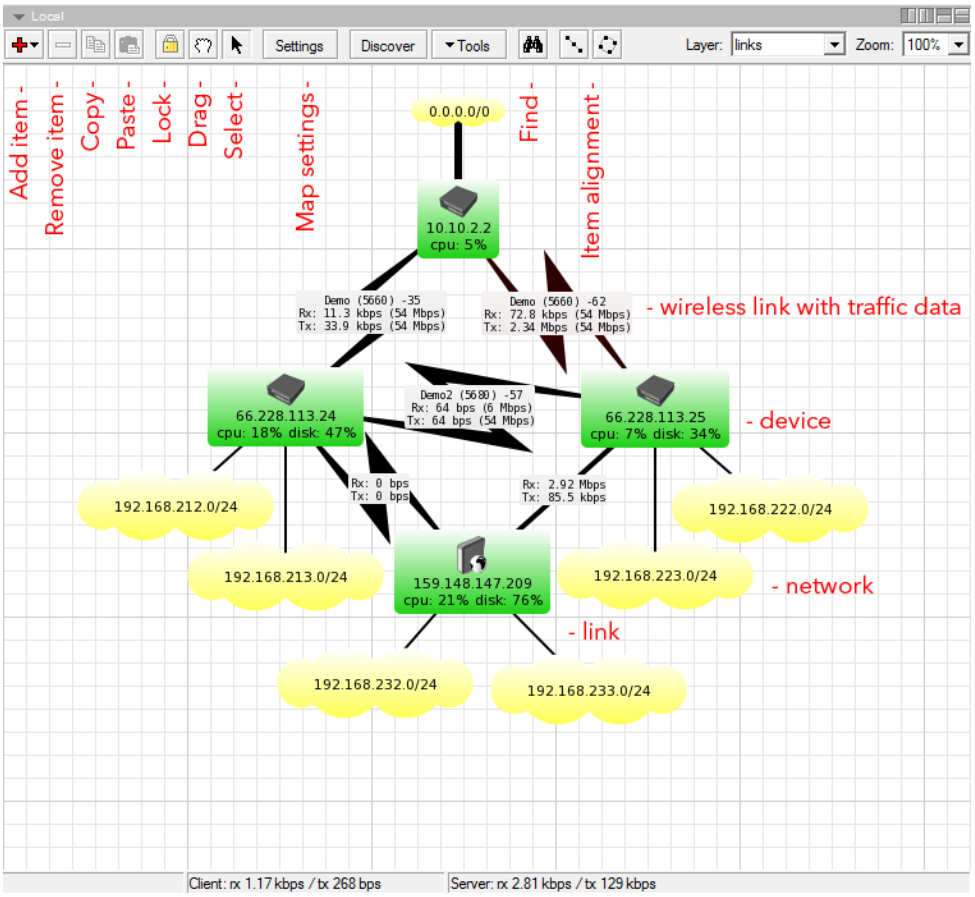
Figura 10 – UNMS/UISP



Fonte: UBIQUITI

Já o The Dude, desenvolvido pela Mikrotik, é uma aplicação leve e funcional, que realiza varreduras da rede, mapeamento gráfico de topologia e coleta de métricas de desempenho. Possui suporte nativo a SNMP, ICMP, DNS (*Domain Name System*) e TCP. Sua instalação é simples, com operação local, mas apresenta limitações na escalabilidade, integração com APIs externas e suporte a gerenciamento de dispositivos de outros fabricantes (gerencia por completo apenas dispositivos RouterOS).

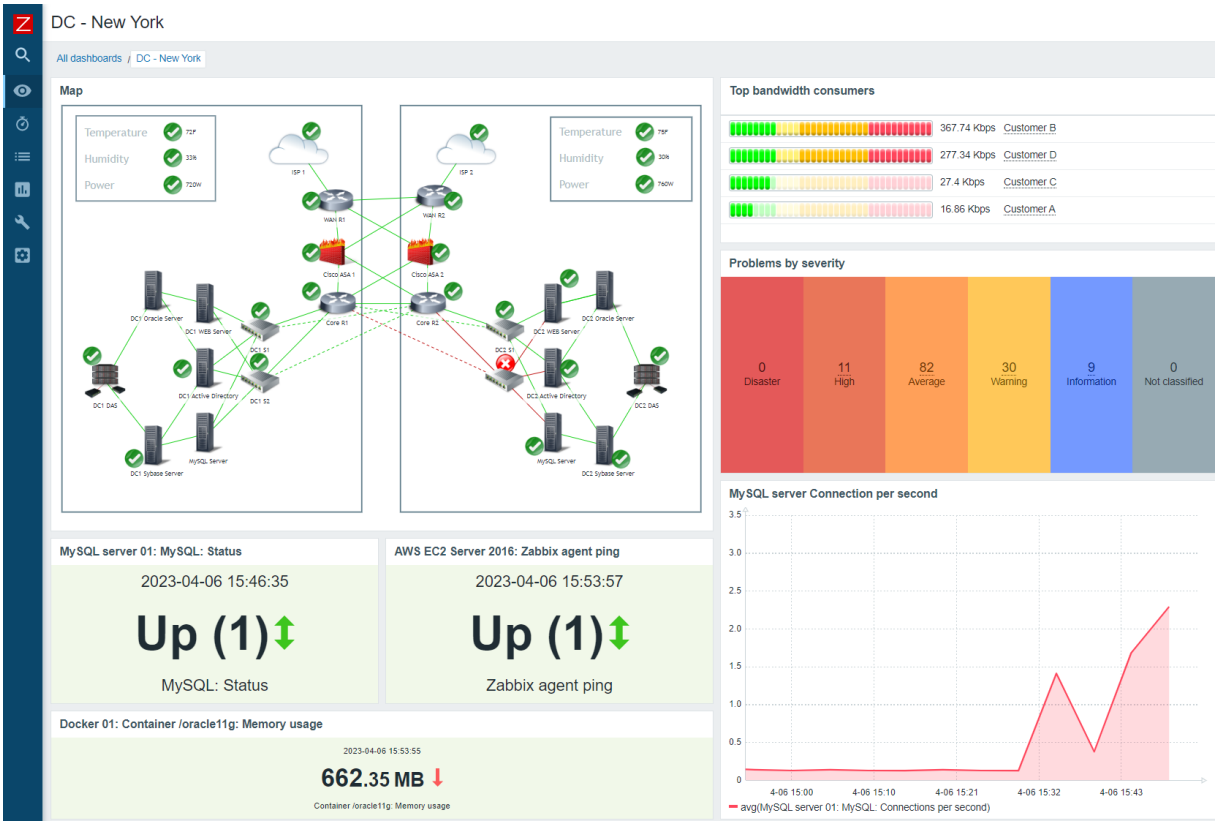
Figura 11 – The Dude



Fonte: MIKROTIK

O Zabbix é uma plataforma robusta e altamente configurável, com suporte nativo ao SNMP. Sua força reside na flexibilidade de personalização, geração de gráficos, escalabilidade e integração com terceiros. No entanto, seu uso exige configuração técnica mais aprofundada, o que pode representar uma barreira para provedores com equipes técnicas reduzidas. Uma das principais aplicações do Zabbix é a geração de *dashboards* para acompanhamento da saúde da rede.

Figura 12 – Zabbix para monitoramento de redes



Fonte: ZABBIX

Essas ferramentas atendem setores específicos do gerenciamento e monitoramento de redes, porém compartilham uma lacuna: adaptabilidade, sendo a falta dela no caso das duas primeiras, e o seu alto custo técnico no caso da terceira.

3.2. Trabalhos correlatos

Esta seção apresenta trabalhos acadêmicos e propostas técnicas que se aproximam, em abordagem ou escopo, da solução proposta pelo sistema RFSight. São considerados aqui projetos que envolvem o gerenciamento de redes sem fio, conectividade em áreas remotas, modularidade de sistemas e protocolos de gerenciamento.

Primeiramente, Campos (2018) desenvolve um estudo de caso focado na aplicação de ferramentas de gerenciamento de redes. A autora explora soluções como Zabbix, Nagios e SpiceWorks (sendo este o implementado) no ambiente

corporativo da Fundação de Estudos e Pesquisas Socioeconômicos (FEPESE), detalhando o processo de implementação e os benefícios gerados pelo monitoramento proativo de ativos de rede. O estudo é de grande relevância por demonstrar empiricamente o valor de plataformas de gerência, que centralizam a visualização do *status* da rede e automatizam a detecção de falhas. Contudo, a análise se restringe a um cenário de infraestrutura cabeada específico e a ferramentas de propósito geral, que, conforme discutido anteriormente, podem apresentar uma curva de aprendizado elevada para provedores com equipes reduzidas.

Ampliando o escopo para novos domínios de aplicação, o trabalho de Bertollo (2021) investiga o papel da Internet das Coisas (IoT) no campo brasileiro e a atuação dos provedores regionais como agentes viabilizadores dessa transformação digital. O autor argumenta que a expansão da conectividade para áreas rurais, impulsionada por esses provedores, permite a adoção de tecnologias para monitoramento da produção, controle de pragas e otimização de processos agrícolas. Dessa forma, o estudo evidencia uma demanda crescente por uma infraestrutura de rede confiável e bem gerenciada em cenários de alta complexidade e criticidade, como o agronegócio, reforçando a necessidade de soluções de gerência, como o RFSight, que sejam acessíveis e adaptadas às operações desses ISPs.

Olhando para o futuro da conectividade rural, o trabalho de Santana *et al*. (2023) analisa o potencial da tecnologia 5G para o agronegócio. O estudo exalta as possibilidades da nova geração para criar fazendas inteligentes por meio de altíssima velocidade e baixíssima latência, permitindo a interconexão de múltiplas tecnologias. Contudo, a pesquisa também revela uma lacuna fundamental: uma análise de patentes demonstrou uma quase inexistente participação do Brasil no desenvolvimento de tecnologias que unem 5G e agronegócio.

Corroborando essa visão, Farias (2021) detalha os imensos desafios para a implantação do 5G no Brasil. O autor foca nos entraves financeiros e regulatórios, destacando o altíssimo custo de investimento para as operadoras, que precisam realizar uma densificação de antenas, expandir massivamente a fibra óptica e

modernizar toda a arquitetura de rede. O estudo aponta que, historicamente, novas tecnologias móveis enfrentam dificuldades de investimento no país , e a situação é agravada pela necessidade de atender regiões de baixo retorno financeiro.

Em suma, os trabalhos analisados tratam da crescente demanda por conectividade rural voltada ao aumento de produtividade no agronegócio, suprida majoritariamente por ISPs regionais, alinhada à expansão de tecnologias de conectividade como o 5G, ressaltando suas barreiras financeiras e estruturais que postergam a massificação. Dessa forma, soluções alternativas, como a proposta desse trabalho, tendem a ganhar espaço em meio aos provedores regionais, conforme demonstra a análise comparativa do protótipo completo no Capítulo 7.

4. PROPOSTA

O presente trabalho tem como foco o desenvolvimento de um protótipo funcional de sistema de gerenciamento modular e adaptável, nomeado RFSight, voltado ao monitoramento de redes de radiofrequência utilizadas por provedores regionais, especialmente em áreas rurais com baixa infraestrutura.

A solução nasce da constatação, fundamentada ao longo deste trabalho, de que os provedores locais enfrentam carência de ferramentas específicas, acessíveis e independentes de fabricantes para gerenciar enlaces ponto a ponto (P2P) e ponto multiponto (P2MP). Como demonstrado na análise comparativa, a maioria das ferramentas comerciais é limitada por ecossistemas fechados, enquanto as propostas acadêmicas carecem de implementação prática ou foco em redes RF.

O RFSight se propõe a preencher essa lacuna com uma arquitetura modular, construída com tecnologias de código aberto, compatível com diversos fabricantes por meio do uso de uma camada de abstração da API para comunicação com os dispositivos, e projetado para operação local em ambientes com restrições de conectividade.

4.1. Levantamento de requisitos

Com o objetivo de identificar e validar as funcionalidades mais relevantes a serem contempladas no sistema proposto, foi aplicado um formulário semi-estruturado (Apêndice – A) voltado a profissionais das áreas de produtos e negócios, desenvolvimento e gestão de redes de telecomunicações. A intenção foi levantar, de forma sistemática, as necessidades operacionais relacionadas ao gerenciamento de redes de rádio frequência em contextos reais de atuação.

A elaboração do instrumento foi fundamentada em três pilares principais: (i) os conceitos apresentados nas seções de fundamentação teórica, especialmente no que se refere aos protocolos e estruturas de redes sem fio; (ii) as funcionalidades oferecidas por soluções existentes e discutidas na seção de estado da arte; e (iii) a estrutura de monitoramento comumente requerida em cenários de conectividade

rural com restrições operacionais, focando na base essencial. Dessa forma, o questionário contemplou temas como funcionalidades e métricas de desempenho indispensáveis e dados necessários dos dispositivos. Além de disponibilizar uma seção adicional de texto livre para sugestões e/ou comentários pertinentes ao contexto.

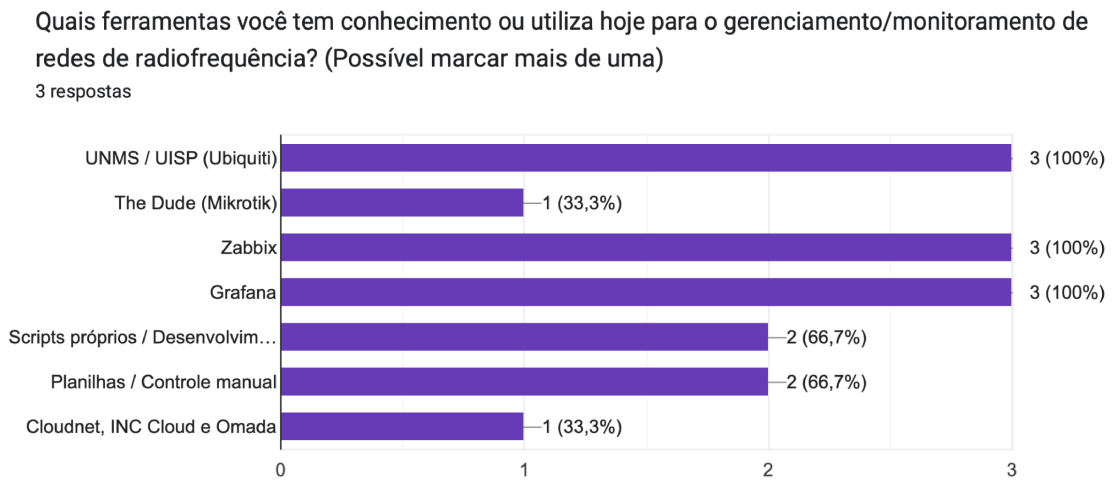
Assim, o formulário foi disponibilizado de forma digital e teve adesão de três analistas, que atuam em ambientes compatíveis com o escopo técnico deste trabalho, possibilitando a coleta de dados coerentes com o contexto proposto. As respostas obtidas serviram de base para a análise qualitativa e definição do escopo da solução, conforme descrito na subseção a seguir.

4.2. Análise e Definição de Escopo

Primeiramente, obteve-se as respostas de três profissionais com experiência entre 5 e 10 anos nas áreas de “Desenvolvimento de Produtos para Telecom” e “Gestão de Negócios / Estratégia em ISP”. Foram considerados como Entrevistado 1, Entrevistado 2 e Entrevistado 3, cujas respostas se encontram no Anexo – A, Anexo – B e Anexo – C, respectivamente.

De forma geral, observou-se convergência entre os respondentes quanto ao conhecimento de ferramentas existentes, apontando a presença do UNMS / UISP (Ubiquiti) e do Zabbix no dia a dia dos profissionais de forma consolidada, como descrito na seção 3. A Figura 13, apresenta de forma gráfica as respostas dos entrevistados quanto ao conhecimento e/ou uso das ferramentas.

Figura 13 – Respostas: Ferramentas de Gerenciamento/Monitoramento



Fonte: Autor

Conforme apresenta a Figura 13, as ferramentas UNMS / UISP, Zabbix e Grafana (*software* para criação de *dashboards* de monitoramento avançados, comumente integrado com o Zabbix) foram apontadas por todos os entrevistados, demonstrando sua consolidação como soluções de mercado no contexto de gerenciamento e monitoramento de redes. Além disso, o uso de *scripts* próprios e controles manuais também foi mencionado, indicando que, embora ferramentas robustas sejam utilizadas, há lacunas que exigem soluções personalizadas e intervenções operacionais diretas.

Também, foram apontadas lacunas referentes às ferramentas selecionadas pelos entrevistados 1 e 2, sendo elas consolidadas na falta de alertas para problemas na rede, dificuldade no gerenciamento customizado (por dispositivo) e centralizado (contendo gráficos em tempo real), bem como a escassez da possibilidade de aplicar configurações de forma temporária nos dispositivos, com opção de *rollback* (desfazer alterações).

Outrossim, a importância de funcionalidades básicas de gestão, como adoção e remoção de dispositivos, monitoramento do *status* da rede e dispositivos individuais teve alto grau apontado, além da visualização de indicadores como *uptime* (tempo de atividade), endereços IP/MAC, versão de *firmware*, latência e

throughput (taxa de transferência de dados). Funcionalidades essas, que estão presentes nas ferramentas existentes e são consideradas praticamente a base para qualquer uma. A análise das respostas deu-se principalmente pela média do grau de importância apontado pelos três entrevistados para cada item. Para isso, os itens do formulário foram divididos em dois tipos primários: Monitoramento e Gerenciamento, e classificados seguindo a mesma escala do formulário, a fim de facilitar a visualização na Tabela 2.

Tabela 2 – Classificação de Funcionalidades por Grau de Importância

Funcionalidade	**Grau de Importância (média)**	**Classificação**	**Tipo**
Monitorar *status* individual dos dispositivos	4.33	Importante	Monitoramento
Monitorar *status* agregado da rede	4.33	Importante	Monitoramento
Status (*Online*/*Offline*)	4.33	Importante	Monitoramento
Uptime (Tempo de atividade)	4.33	Importante	Monitoramento
Endereços IP e MAC	4.33	Importante	Monitoramento
Quantidade total de dispositivos *Online* / *Offline*	4.33	Importante	Monitoramento
Throughput (*Download*/*Upload*) agregado de toda a rede	4.33	Importante	Monitoramento
Hostname	4.00	Importante	Monitoramento
Latência	4.00	Importante	Monitoramento
Throughput por enlace individual	4.00	Importante	Monitoramento

Visualizar gráficos de desempenho da rede	3.67	Importante	Monitoramento
Topologia da rede	3.67	Importante	Monitoramento
Localização	3.33	Útil	Monitoramento
Um "Gráfico de Saúde da Rede" (tipo Gauge).	3.00	Útil	Monitoramento
Adotar dispositivos	4.33	Importante	Gerenciamento
Remover dispositivos adotados	4.33	Importante	Gerenciamento
Configurar dispositivos em massa	4.33	Importante	Gerenciamento
Atualizar dispositivos em massa	4.33	Importante	Gerenciamento
Versão de *firmware*	4.33	Importante	Gerenciamento
Gerência de usuários do sistema (cadastro, edição e exclusão)	3.67	Importante	Gerenciamento
Acesso com *login* por nome de usuário/senha	3.67	Importante	Gerenciamento
Configurar individualmente cada dispositivo adotado	3.33	Útil	Gerenciamento
Atualizar dispositivos de forma individual	3.00	Útil	Gerenciamento

Fonte: Autor

A Tabela 2 apresenta a média das avaliações atribuídas a cada funcionalidade, evidenciando que os recursos considerados essenciais à operação, como monitoramento de *status*, adoção e remoção de dispositivos, e análise de

indicadores, foram classificados como “Importante”. As funcionalidades classificadas como “Útil”, especialmente aquelas de tipo “Monitoramento”, podem ser consideradas complementares ao escopo principal. Já as ações de gerenciamento individual, como configurar e atualizar dispositivos, embora não tenham atingido o grau máximo de prioridade, representam a base operacional necessária para evoluções futuras como a gestão em massa, especialmente em redes de maior escala.

Além disso, o Entrevistado 2 apontou como sugestão que o sistema possua uma forma direta de acesso remoto aos dispositivos via *browser*, funcionalidade que pode ser considerada como diferencial e alinhada à proposta modular do sistema como um adicional.

A análise das respostas dos profissionais da área foi crucial para o levantamento de requisitos e definição de uma arquitetura para o protótipo, a fim de comportar as principais funcionalidades e definir um escopo viável.

4.3. Definição formal de requisitos e escopo

A partir desta análise, juntamente do contexto do protótipo (considerando a viabilidade dentro da janela de tempo de desenvolvimento deste trabalho), formou-se o conjunto de requisitos funcionais priorizados para o desenvolvimento do protótipo, definido por:

- RF01 – O sistema deve permitir a adoção de dispositivos de rede.
- RF02 – O sistema deve permitir a remoção de dispositivos adotados.
- RF03 – O sistema deve permitir a configuração individual dos dispositivos.
- RF05 – O sistema deve permitir o monitoramento de *status* individual dos dispositivos, incluindo:
    - *Online*/*Offline*;
    - *Uptime*;
    - *Hostname*;
    - Endereço IP;
    - Endereço MAC;

    - Versão de *firmware*;
    - Latência.
- RF06 – O sistema deve apresentar um painel com gráficos em tempo real sobre o estado da rede, contemplando:
    - Topologia da rede;
    - Quantidade de dispositivos *online* e *offline*;
    - *Throughput* agregado por rede;
    - *Throughput* por enlace.
- RF07 – O sistema deve organizar os dispositivos de forma hierárquica, agrupados por organização e rede.
- RF08 – O sistema deve permitir autenticação de usuários por nome de usuário e senha.
- RF09 – O sistema deve ser acessado diretamente por navegador *web*, sem necessidade de cliente externo.

As funcionalidades de configuração e atualização em massa, embora classificadas como importantes, foram preteridas em favor da gestão individualizada, a fim de priorizar a consolidação da arquitetura modular e da camada de abstração. Dentro deste escopo individual, a funcionalidade de configuração foi plenamente contemplada. Contudo, a atualização de *firmware* não integrou o conjunto de requisitos funcionais validados, devido à inviabilidade de testes práticos com os dispositivos disponíveis (ausência de arquivos de *firmware* compatíveis). Apesar disso, ressalta-se que a arquitetura do protótipo foi desenvolvida já suportando o envio de *payloads* do tipo “arquivo”, garantindo a extensibilidade futura do sistema para este fim.

Além dos requisitos funcionais, foram definidos os seguintes requisitos não funcionais com base nas diretrizes técnicas do projeto e sua arquitetura, a fim de garantir desempenho, segurança e portabilidade da solução:

- RNF01 – O sistema deve operar em ambiente local, diretamente conectado à rede alvo.
- RNF02 – O sistema deve ser compatível com sistemas operacionais baseados em distribuições Linux.

- RNF03 – O sistema deve ser implementado utilizando arquitetura conteinerizada, Docker (DOCKER, INC., 2025).
- RNF04 – O sistema deve ser tolerante a falhas, garantindo continuidade da operação em caso de interrupções.
- RNF05 – O sistema deve apresentar uma interface de gerenciamento *web* desenvolvida com React (META PLATFORMS, 2025).
- RNF06 – O *Backend* do sistema deve ser desenvolvido em Python (PYTHON SOFTWARE FOUNDATION, 2025).
- RNF07 – A API do sistema deve seguir o padrão REST e ser implementada utilizando o *framework* FastAPI (RAMÍREZ, 2025).
- RNF08 – O sistema deve ser escalável, suportando operação eficiente mesmo com o aumento do número de dispositivos monitorados.
- RNF09 – Os dados de autenticação de usuários devem ser armazenados de forma criptografada.
- RNF10 – O sistema deve utilizar MongoDB (MONGODB, INC., 2025) como banco de dados para persistência das informações.

Dessa forma, o escopo de implementação do sistema, se dá pelo conjunto de requisitos funcionais e não funcionais aqui definidos, representando a síntese entre as prioridades indicadas pelos profissionais consultados e as condições técnicas viáveis e de maior alinhamento ao objetivo modular do sistema RFSight. Essa estrutura permitirá uma implementação escalável e adequada às necessidades reais de monitoramento e gestão de redes via radiofrequência.

4.4. Arquitetura e tecnologias da solução

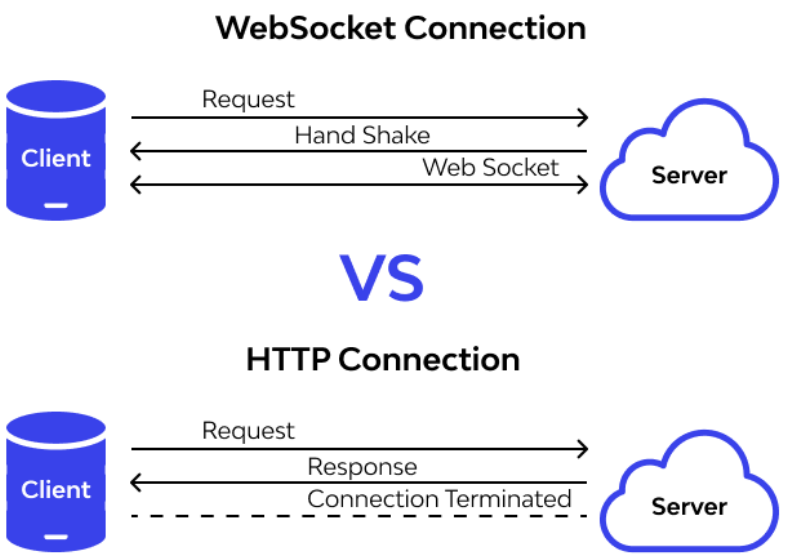
Conforme os requisitos definidos, o protótipo RFSight adotou uma arquitetura modular, baseada em tecnologias modernas, de código aberto e amplamente utilizadas em ambientes de produção (*softwares* finais). A escolha dessas tecnologias ateve-se à portabilidade, escalabilidade, segurança, facilidade de integração e baixo custo de implantação, todos aspectos essenciais para aplicações voltadas a ISPs.

Para o *Backend* (servidor) da aplicação, foi definido o uso da linguagem de programação Python, reconhecida por sua sintaxe simplificada, amplo suporte à integração com redes e grande variedade de bibliotecas. Dentre elas, destaca-se o *FastAPI*, um framework moderno que permite a criação de APIs REST de forma rápida, segura e com excelente desempenho. Além disso, foi mapeado a necessidade de uso de algumas bibliotecas principais: *requests*, para comunicação com APIs externas ao sistema (dispositivo); *paramiko*, para automação de conexões via SSH; e por fim *scapy* para operações de análise de rede.

A comunicação entre o *Frontend* (*interface* gráfica) e o *Backend* do protótipo é estruturada por meio de *endpoints* de APIs REST, que permitem a troca de informações entre cliente e servidor utilizando o protocolo HTTP e o formato JSON. Essa arquitetura facilita a separação de responsabilidades e a manutenção do sistema, permitindo alterações e correções mais atomizadas.

Já para exibição de dados da rede em tempo real, foi adotado o uso de *WebSockets*, pois permitem uma conexão persistente e bidirecional entre *Frontend* e *Backend*. De acordo com a especificação do protocolo (RFC 6455), essa abordagem reduz a sobrecarga de comunicação e melhora significativamente a experiência do usuário ao eliminar a necessidade de consultas periódicas (*polling*). A Figura 14 demonstra a diferença entre o fluxo de requisições HTTP e *WebSocket*.

Figura 14 – *WebSocket* vs. HTTP



Fonte: BESCHOKOV, 2025

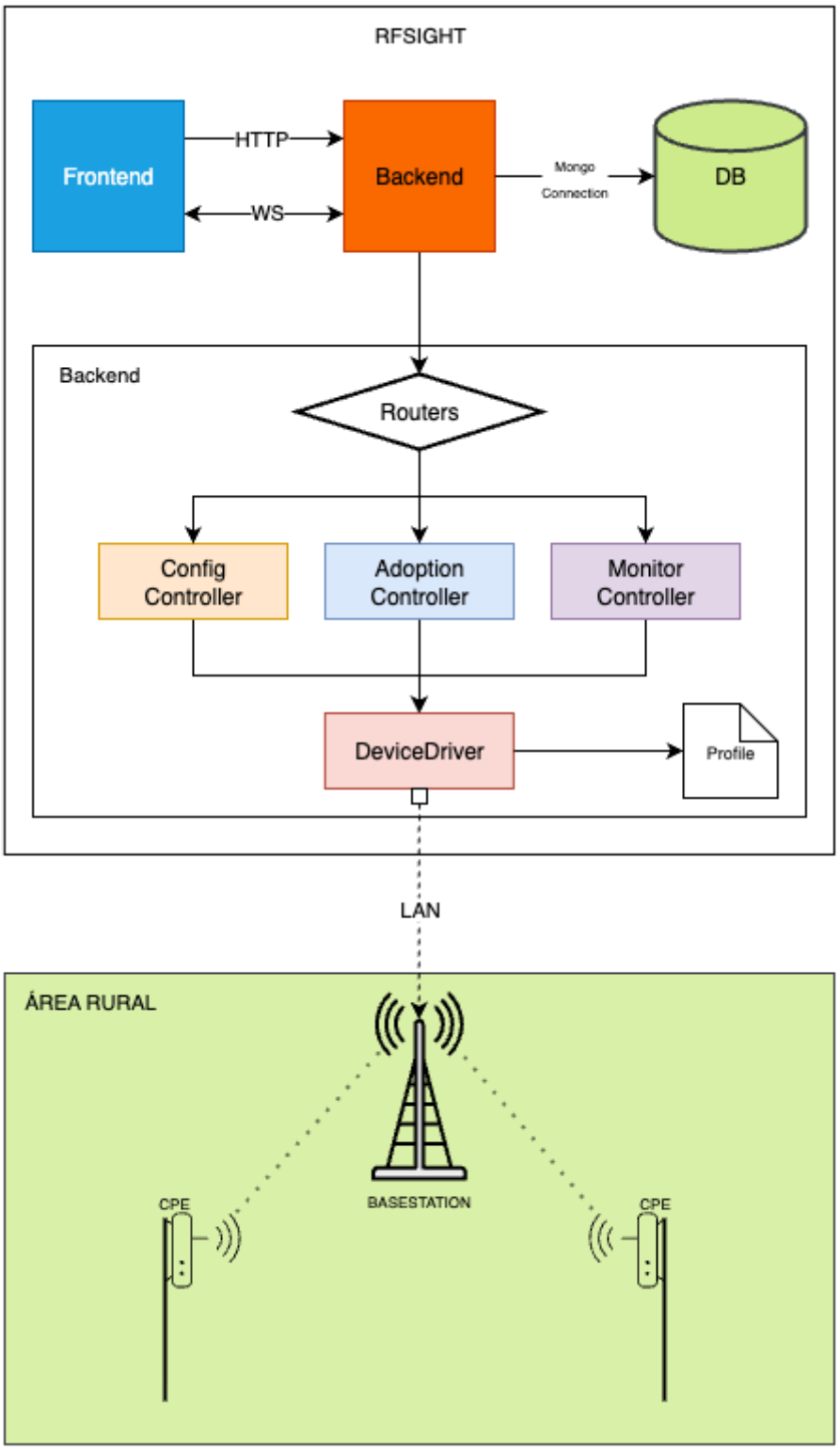
A segurança das APIs foi garantida por meio do uso de JWT (JSON *Web Tokens*), conforme definido pela RFC 7519. O JWT é um padrão aberto que permite a troca segura de informações entre partes por meio de *tokens* compactos e autoexplicativos. Ele é amplamente utilizado para autenticação e autorização, evitando o uso contínuo de credenciais sensíveis e permitindo controle de sessões com baixo custo computacional.

Decidiu-se para o *Frontend* da solução, o uso da biblioteca React e por meio da linguagem TypeScript, tecnologias que oferecem robustez, tipagem estática e melhor organização do código. O React permite a construção de *interfaces* responsivas a partir do uso de componentes reutilizáveis, enquanto o TypeScript contribui para a detecção precoce de erros, melhor qualidade de código e manutenibilidade. Para gerência de estados de componentes e requisições para *Backend*, optou-se pela biblioteca Redux Toolkit (REDUX TOOLKIT, 2025), pela sua versatilidade e robustez para criação de *stores* e *slices* (desempenhando função análoga à de um banco de dados para a camada do *Frontend*). A escolha dessas ferramentas permite a criação de páginas dinâmicas para visualização de métricas de rede em tempo real.

Para persistência dos dados, optou-se pelo MongoDB, um banco de dados NoSQL (não relacional) orientado a documentos, que armazena informações em formato JSON. Sua estrutura flexível permite lidar com dados não estruturados e de diferentes tipos, além de simplificar a escalabilidade da aplicação.

Para a organização de componentes e arquitetura do sistema, as tecnologias de conteinerização e orquestração de serviços Docker e Docker Compose foram escolhidas, facilitando a inicialização do sistema, com foco na escalabilidade de recursos por módulo. A aplicação é dividida em múltiplos serviços atômicos, como API, *interface web* (UI) e banco de dados (DB), permitindo controle granular e manutenção independente de cada módulo. A Figura 15 apresenta a arquitetura proposta para o protótipo.

Figura 15 – Arquitetura proposta do RFSight



Fonte: Autor

Dessa forma, a arquitetura proposta para o RFSight, conforme ilustrado na Figura 15, adota um design modular e orientado a serviços para garantir flexibilidade

e escalabilidade. A separação entre a *interface* (*Frontend*), a lógica de negócio (*Backend*) e os controladores especializados (*Config*, *Adoption* e *Monitor*) permite um desenvolvimento e manutenção independentes de cada componente. O elemento central da proposta, contudo, é a Camada de Abstração de Dispositivo, que desacopla o sistema dos detalhes de *hardware* de cada fabricante por meio da abstração da forma de gerência de cada dispositivo (*Device*). Ela é composta pelo *DeviceDriver* (responsável por interagir proativamente com os dispositivos) juntamente do uso da entidade *Profile* (que armazena informações cruciais para comunicação com os dispositivos, como: ações disponíveis, tipos de payloads, rotas de API e OID SNMP). Essa abordagem assegura a extensibilidade da solução, permitindo que novos dispositivos sejam integrados futuramente com menor esforço de desenvolvimento.

Ademais, a Figura 15 também apresenta o cenário de uso do sistema, sendo: a conexão via LAN diretamente à rede alvo em uma área rural, adotando e gerenciando principalmente os dispositivos caracterizados como *Basestations*, a fim de capturar informações essenciais do monitoramento, relacionadas aos enlaces e estações conectadas (CPEs).

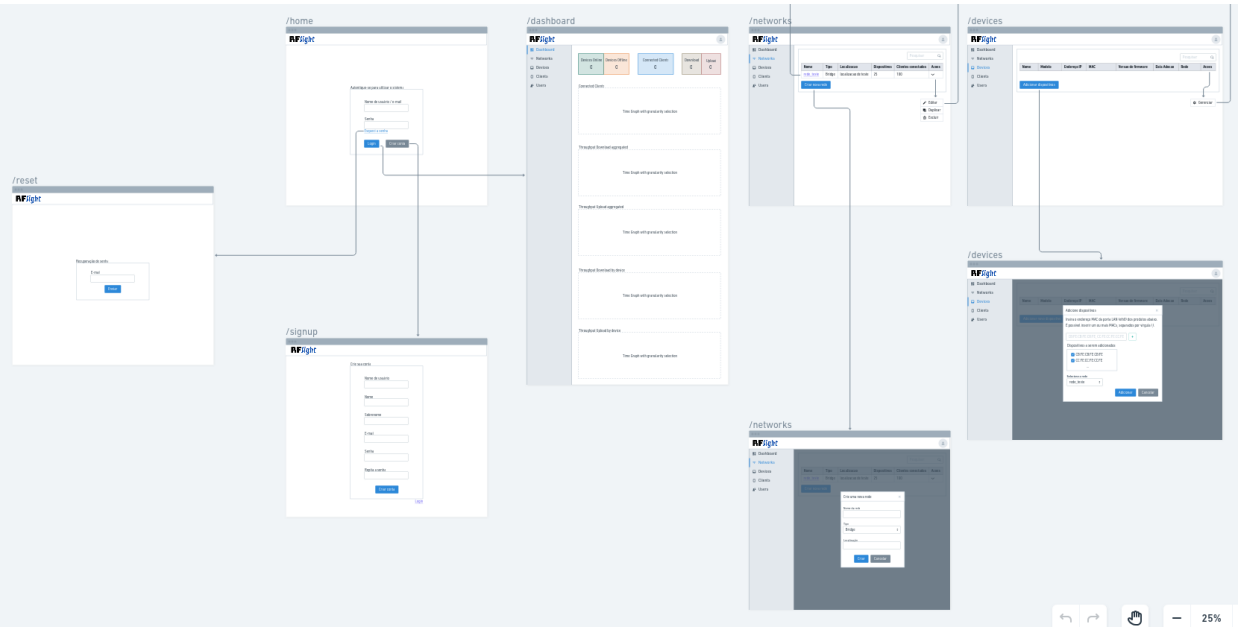
5. DESENVOLVIMENTO

O desenvolvimento do protótipo RFSight foi guiado pelos requisitos funcionais e não funcionais definidos no Capítulo 4. Esta seção detalha as etapas de implementação, desde a modelagem dos dados e arquitetura dos componentes até a implementação das funcionalidades centrais de gerenciamento e monitoramento.

5.1. Protótipos de telas

Durante a prototipação de telas, buscou-se definir as principais telas do sistema, bem como os componentes que pudessem ser reutilizados em mais de uma página. Assim, o sistema seguiu uma *interface* padronizada, simples e amigável, com suas funções em acesso rápido. A Figura 16 apresenta uma visão ampliada do que foi a prototipação das telas do RFSight.

Figura 16 – Geral: Protótipos de telas



Fonte: Autor

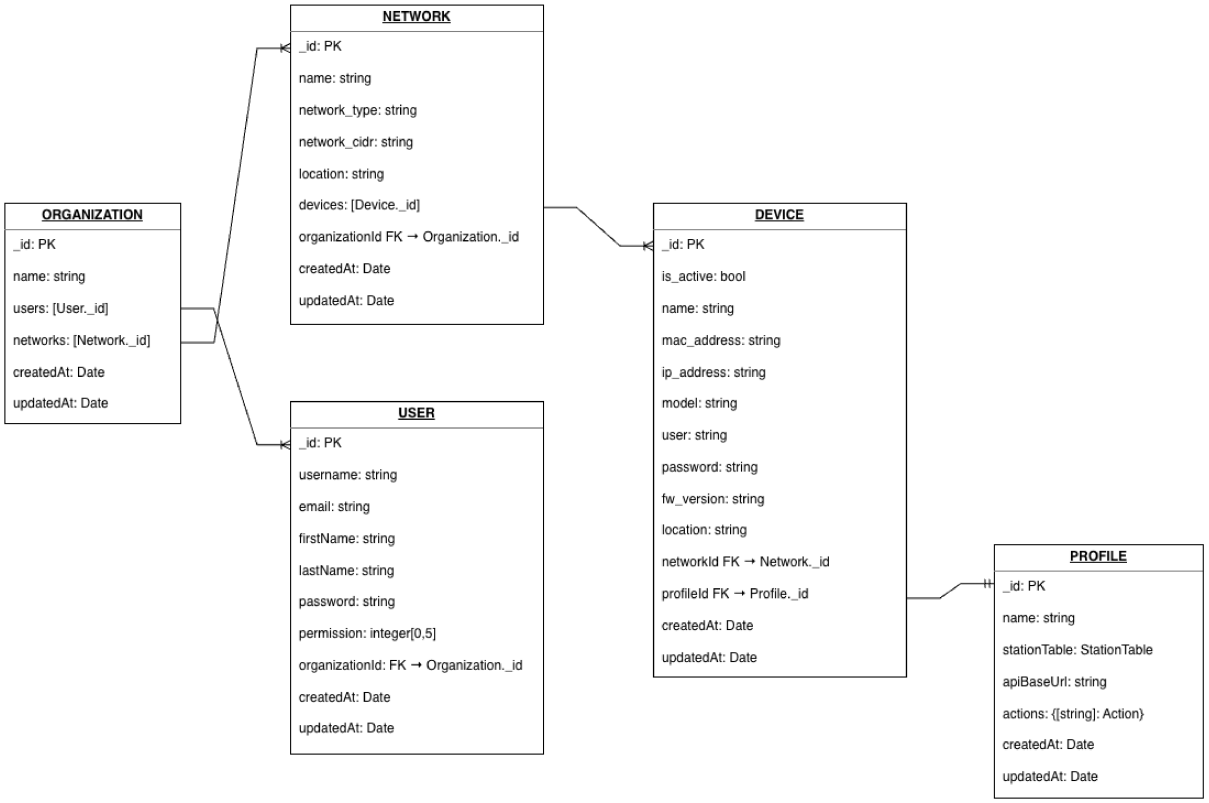
O desenvolvimento dos protótipos, conforme a Figura 16, se deu por meio da ferramenta web Whimsical. Já a padronização visual dos componentes se baseou no guia de estilos e identidade visual propostos pelo Material Design (GOOGLE, 2025), o que facilitou a exatidão e a concordância da versão final do *Frontend*, a partir do uso da biblioteca de componentes React MUI (MUI, 2025).

Ainda que uma etapa consideravelmente simples, a prototipação de telas foi essencial para definir o ponto de partida de como cada funcionalidade se comportaria, do ponto de vista do usuário. Assim, os fluxos do *Backend* foram desenvolvidos em torno da jornada do usuário no sistema.

5.2. Entidades e banco de dados

A persistência dos dados do sistema RFSight ocorreu (em acordância ao RNF10) utilizando o MongoDB como banco de dados NoSQL. Para isso no *Backend* foram utilizados modelos do Pydantic (biblioteca comumente utilizada para a finalidade de definir entidades de banco de dados e validação de tipos). A Figura 17 apresenta o diagrama de modelagem do banco de dados.

Figura 17 – Modelagem BD



Fonte: Autor

Conforme a Figura 17, o modelo lógico do banco de dados utilizado no protótipo, está organizado em cinco entidades principais: *Organization*, *User*, *Network*, *Device* e *Profile*. As relações entre os objetos seguem uma estrutura hierárquica, na qual uma organização pode conter múltiplos usuários e redes, enquanto cada rede pode agregar diversos dispositivos. Os dispositivos, por sua vez, associam-se a um único perfil, responsável por definir seu comportamento operacional. Por fim, cada usuário pode estar relacionado apenas com uma organização.

Para o controle de usuários, é possível verificar na modelagem que a entidade *User* é permissionada, ou seja, o acesso à certas funcionalidades específicas do sistema é atrelada à permissão ativa do usuário, que varia de 0 a 5, sendo: 0 = *Guest* (apenas acessa o sistema e pode visualizar a página inicial); 1 = *Guest Monitor* (mesmas permissões do *Monitor*, porém um nível abaixo na hierarquia); 2 = *Monitor* (pode apenas monitorar as redes); 3 = *Guest Admin* (mesmas permissões de um admin, porém um nível abaixo na hierarquia); 4 = *Admin*

(tem acesso geral no sistema, pode gerenciar usuários, redes, organizações e dispositivos); 5 = *Master* (possui acesso completo ao sistema, pode gerenciar os *Admins*). Importante ressaltar que os usuários *Master* podem ser gerenciados apenas por outros usuários de mesma permissão.

Em relação às entidades *Network* e *Device*, é possível verificar que persistem informações mais específicas para o funcionamento do *core* do sistema. No caso das redes, por exemplo, é persistido o CIDR (Classless Inter-Domain Routing) da rede, essencial para a execução dos serviços de discovery e monitoramento. Já para os dispositivos, são salvas informações base do dispositivo, como seu endereço IP e MAC, versão de firmware, hostname e também credenciais de acesso, que são utilizadas pelo *DeviceDriver* ao executar uma ação relacionada ao perfil do dispositivo.

Ademais, o pilar da arquitetura de modularidade do RFSight é a entidade *Profile* junto da classe *DeviceDriver*, toda a camada de abstração de dispositivo se baseia na premissa do uso de perfis. Este documento define como o sistema deve interagir com uma classe específica de dispositivos, abstraindo os comandos e métodos de comunicação. A Figura 18 apresenta as estruturas de dados Pydantic, utilizadas no *Backend* para validar e modelar os perfis.

Figura 18 – Entidade *Profile*

```python
class HttpDetails(BaseModel):
    port: int = 80
    method: Literal['GET', 'PATCH', 'PUT', 'POST', 'DELETE']
    successStatusCode: int = 200
    path: str
    queryParameters: dict | None = None
    payloadType: Literal['file', 'text/plain', 'text/json'] | None = None
    payloadTemplate: any | None = None
    responseType: Literal['text/plain', 'text/json', 'boolean', 'blank']
    responseMapping: dict | None = None
    responseHeaderMapping: dict | None = None

class SshDetails(BaseModel):
    port: int = 22
    command: str

class Action(BaseModel):
    actionType: Literal['monitor', 'manage', 'auth']
    protocol: Literal['http', 'ssh']
    sshDetails: SshDetails | None = None
    httpDetails: HttpDetails | None = None

class Profile(BaseModel):
    id: str | None = None
    name: str
    stationTable: any
    apiBaseUrl: str
    actions: dict[str, Action]
    createdAt: datetime | None = None
    updatedAt: datetime | None = None
```

Fonte: Autor

A Figura 18 demonstra a estrutura central do Profile. Cada perfil armazena um *apiBaseUrl*, que serve como base para a comunicação, e um dicionário de *Actions*. Cada *Action* define o tipo (*monitor*, *manage*, *auth*), o protocolo (*http*, *ssh*) e os detalhes específicos (como *HttpDetails* ou *SshDetails*). Notavelmente, *HttpDetails* inclui *responseMapping*, um dicionário que mapeia chaves de resposta da API do dispositivo para chaves de dados padronizadas no RFSight (ex: "*device.hostname*" para "*name*"), o que é fundamental para a camada de abstração. A chave stationTable é fundamental ao monitoramento, pois é nela que se encontram os dados de mapeamento SNMP para buscar estações (CPEs) conectadas ao dispositivo em questão.

Portanto, a organização do sistema se dá de forma hierárquica: organização > rede > dispositivo > perfil. Uma organização pode ter várias redes, cada rede pode ter vários dispositivos, cada dispositivo tem apenas um perfil relacionado. Isso faz

com que seja possível separar topologias que não se interligam, para uma melhor análise e gerência.

5.3. Arquitetura dos componentes

A arquitetura do sistema segue uma abordagem de microsserviços conteinerizada, conforme RNF03, orquestrada com Docker Compose. Essa abordagem garante a portabilidade e a escalabilidade dos componentes. A Figura 19 detalha os serviços definidos no arquivo, os quais compõem o funcionamento da aplicação.

Figura 19 – Componentes Docker RFSight (simplificado)

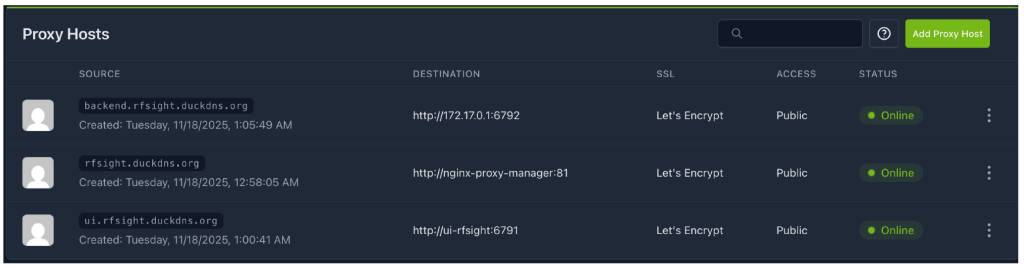
```yaml
services:
  nginx-proxy-manager:
    image: jc21/nginx-proxy-manager:latest
    ports:
      - "80:80"
      - "81:81"
      - "443:443"
    volumes:
      - ./nginx-proxy/data:/data
      - ./nginx-proxy/letsencrypt:/etc/letsencrypt
    networks: [frontend]

  api-rfsight:
    build: ./backend
    volumes:
      - ./backend:/app
    depends_on: [mongo]
    ports:
      - "6792:6792"
    network_mode: host

  ui-rfsight:
    build: ./frontend
    ports:
      - "6791:6791"
      - "4173:4173"
    volumes:
      - ./frontend:/app
    networks: [frontend]

  mongo:
    image: mongo:latest
    volumes:
      - mongodb-data-rfsight:/data/db
    networks: [db_network]
    ports:
      - "27017:27017"
```

Fonte: Autor

Conforme a Figura 19, o sistema executa a partir de quatro componentes principais: *nginx-proxy-manager* (*proxy* para redirecionamento de requisições e servir a aplicação); *api-rfsight* (o *Backend* do sistema, contendo toda a lógica de gerenciamento e monitoramento); *ui-rfsight* (*interface* de usuário do sistema, desenvolvida em React); e por fim *mongo* (banco de dados não relacional utilizado para persistência).

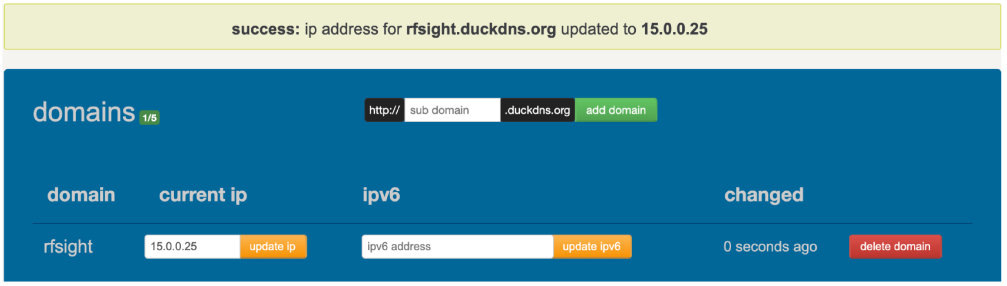
O componente *nginx-proxy-manager* (CURNOW, 2025) opera com foco principal em simplificar o acesso aos serviços do RFSight, centralizando e redirecionando as requisições, bem como mantendo a segurança do acesso da aplicação por meio do uso de um certificado digital, adquirido gratuitamente por meio da autoridade certificadora Let's Encrypt (INTERNET SECURITY RESEARCH GROUP, 2025), alinhado à utilização de um domínio Duck DNS (DUCK DNS, 2025) apontado para um IP local na rede. Essa configuração permite que o RFSight opere em HTTPS, bem como utilize conexões WSS (*Secure Websocket*) para o monitoramento em tempo real. As Figuras 20 e 21 apresentam a configuração do *nginx-proxy-manager* e do domínio Duck DNS, respectivamente.

Figura 20 – Configuração Nginx



Fonte: Autor, 2025

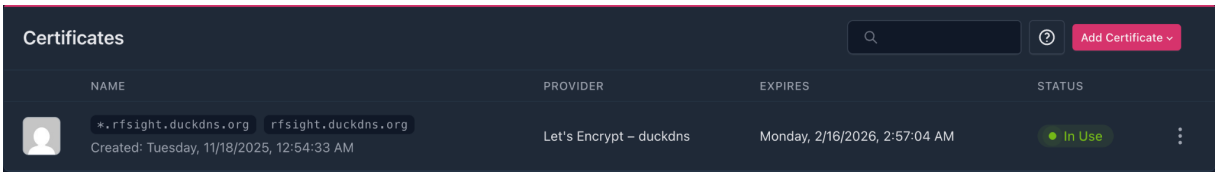
Figura 21 – Configuração Duck DNS



Fonte: Autor, 2025

Conforme apresentam as Figuras 20 e 21, estão configurados três *hosts* principais da aplicação, sendo: “*backend.rfsight.duckdns.org”* (apontado para o componente *api-rfsight*, utilizando o endereço IP da rede Docker que aponta para o *localhost*, dado que o componente em si opera em modo de rede *host*); “*ui.rfsight.duckdns.org*” (apontado para o componente *ui-rfsight* utilizando o *hostname* do serviço na rede Docker); e por fim “*rfsight.duckdns.org*” (apontado para o próprio serviço de *proxy*). Em relação ao domínio Duck DNS, sua configuração foi realizada para apontar para um endereço IPv4 da rede local, mais especificamente o endereço da máquina servidor executando o RFSight (15.0.0.25). Por fim, a conexão segura dos *hosts* se dá pelo uso do certificado apresentado na Figura 22.

Figura 22 – Certificado RFSight



Fonte: Autor

Ademais, dentre todos os componentes, o mais complexo é definitivamente o que compõe o *Backend*, pois nele estão localizados todos os serviços de comunicação com os dispositivos, orquestração e sincronia de tarefas, bem como seu processamento. Internamente, o *Backend* é modularizado em controladores e serviços, gerenciados por roteadores assíncronos de requisições HTTP do FastAPI.

Já o serviço do *Frontend*, integra toda a *interface* visual do RFSight, isto é, todas as páginas e componentes que tornam a gerência e o monitoramento facilitados possíveis. Neste componente, acontece a validação de dados inseridos pelo usuário (cadastro e atualização), juntamente do controle de acesso por meio de autenticação e permissionamento integrado com o *Backend* utilizando JWT permissionado.

Para alcançar um sistema tolerável a falhas, todo o *Backend* foi desenvolvido em modelo assíncrono, assim as execuções de suas funções operam primordialmente usando *threads* e subprocessos, com controles de concorrência.

Nesse contexto, o *Frontend* aplica o uso de *toasts* (notificações) de *status* para manter a experiência do usuário fluída e informativa, com mensagens e *feedbacks* específicos a respeito do sucesso ou falha das ações requisitadas.

5.4. Características técnicas

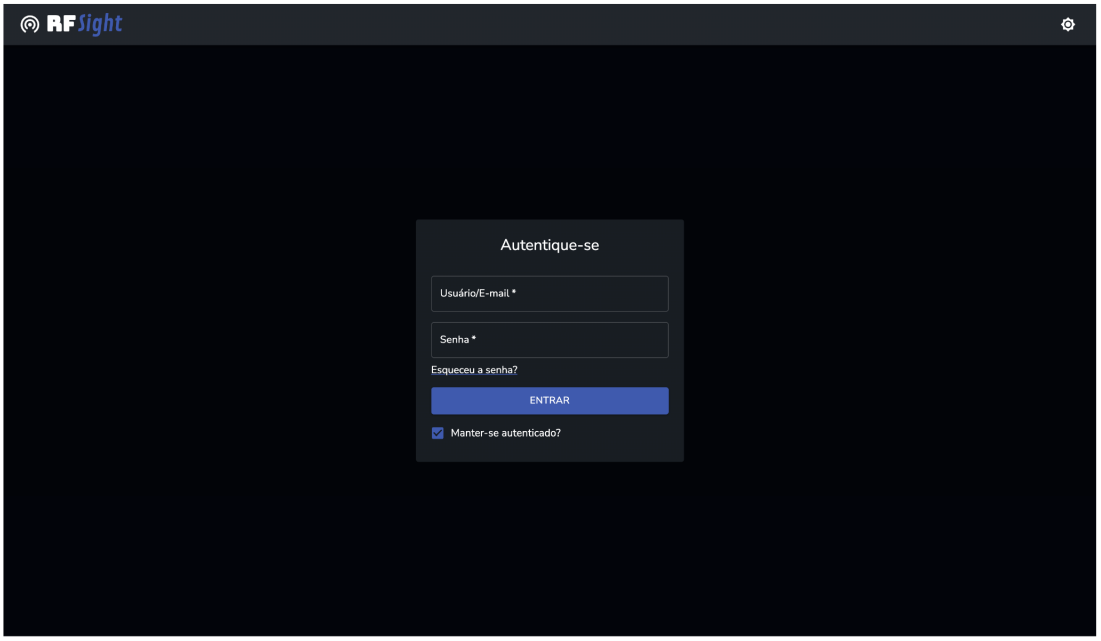
A seguir são descritas as principais características dos fluxos do sistema, apresentando a lógica por trás das principais funcionalidades do RFSight, que envolvem a autenticação, gerência de usuários, definição da hierarquia de organizações e redes, bem como a adoção, configuração e monitoramento dos dispositivos de radiofrequência.

5.4.1. Gerência e autenticação de usuários

O controle de acesso ao sistema é baseado no padrão JWT, garantindo a segurança das rotas da API e a gestão de sessões no *Frontend*. Para a segurança das credenciais, foi utilizada a biblioteca *bcrypt* para o *hashing* das senhas antes da persistência no banco de dados MongoDB e também para sua posterior validação.

O fluxo de autenticação valida as credenciais e emite dois tokens: um *access token* de curta duração (15 minutos) para as requisições imediatas e um *refresh token* de maior validade (1 dia), armazenado em *cookies* seguros (tipo *httpOnly*, e atributo *secure* ativo), permitindo a renovação da sessão sem a necessidade de novos *logins* constantes. A Figura 23 apresenta a tela de autenticação do RFSight.

Figura 23 – Tela de Autenticação do Sistema

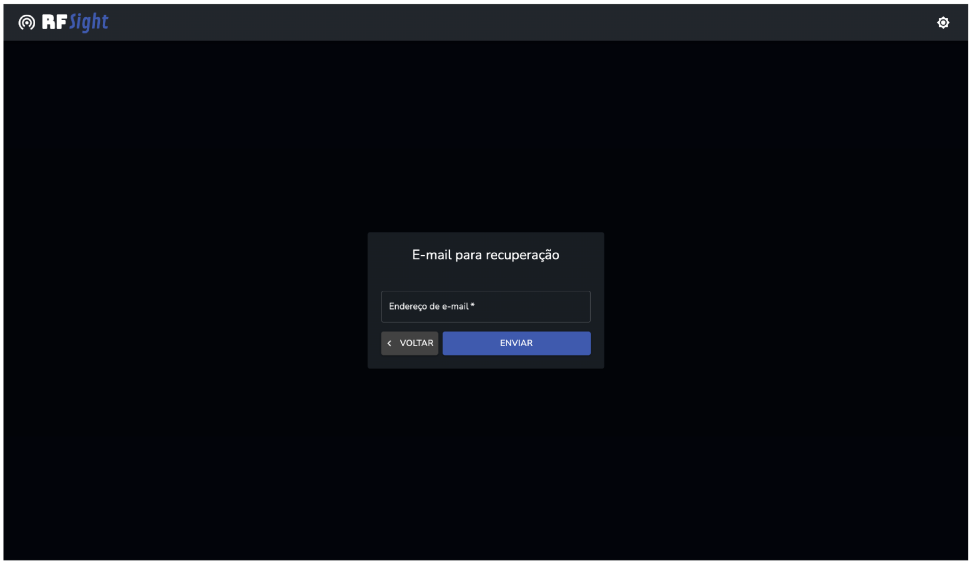


Fonte: Autor

A Figura 23 apresenta a *interface* de *login*, ponto de entrada da aplicação, onde o usuário insere suas credenciais. Além do fluxo padrão, o sistema contempla um mecanismo de recuperação de conta. Caso o usuário esqueça sua senha, a rota "*/forgot-password"* (acessada pelo usuário ao clicar em "Esqueceu a senha?") possibilita que o usuário insira seu *e-mail* para recuperação.

O funcionamento da funcionalidade de recuperação reside no envio de um *e-mail* contendo um *token* temporário e um *link* para redefinição (válido por 10 minutos), garantindo que o acesso possa ser recuperado de forma autônoma e segura. A configuração do servidor de *e-mail* utilizado para esse serviço pode ser alterada por de variáveis de ambiente (.env) do serviço *Backend*. Para fins de desenvolvimento e testes, o protótipo utilizou o serviço *smtp4dev* (WOOD, 2025) que simula um servidor de *e-mail* em ambiente local. As Figuras 24 a 26 demonstram o fluxo de recuperação de senha.

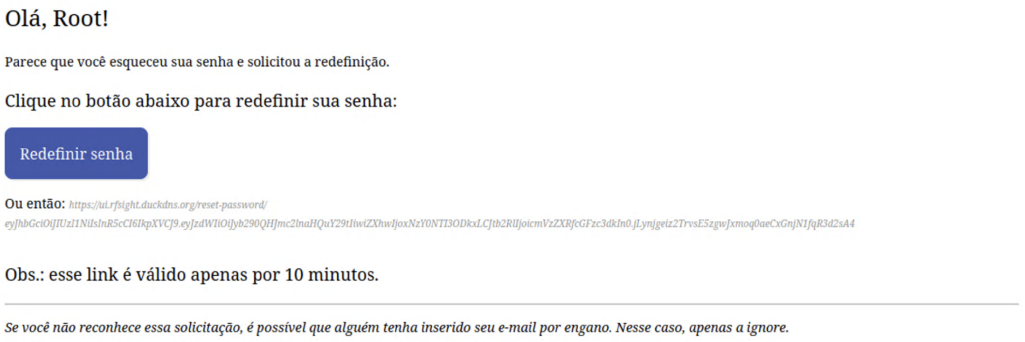
Figura 24 – Inserir *e-mail* para recuperação



Fonte: Autor

Conforme a Figura 24, é solicitado o endereço de *e-mail* do usuário para prosseguir com o processo de recuperação de senha. Caso o sistema identifique que não existe um usuário cadastrado, uma notificação de “usuário inexistente” é apresentada na tela, porém se o usuário for encontrado, o sistema prossegue com o procedimento de recuperação e o usuário deve receber uma mensagem semelhante ao que é apresentado pela Figura 25 em sua caixa de *e-mail*.

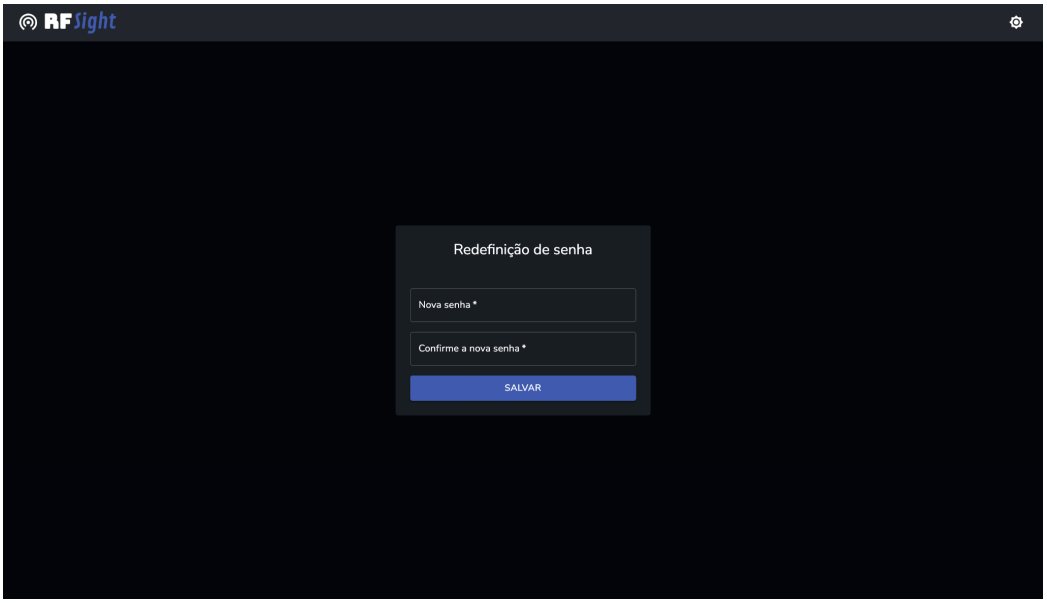
Figura 25 – *E-mail* de recuperação



Fonte: Autor

Dessa forma, ao acessar o *link* de recuperação (clicando em “Redefinir senha” ou acessando diretamente) o usuário tem acesso à tela apresentada na Figura 26, onde pode criar e confirmar a nova senha para posteriormente autenticar-se.

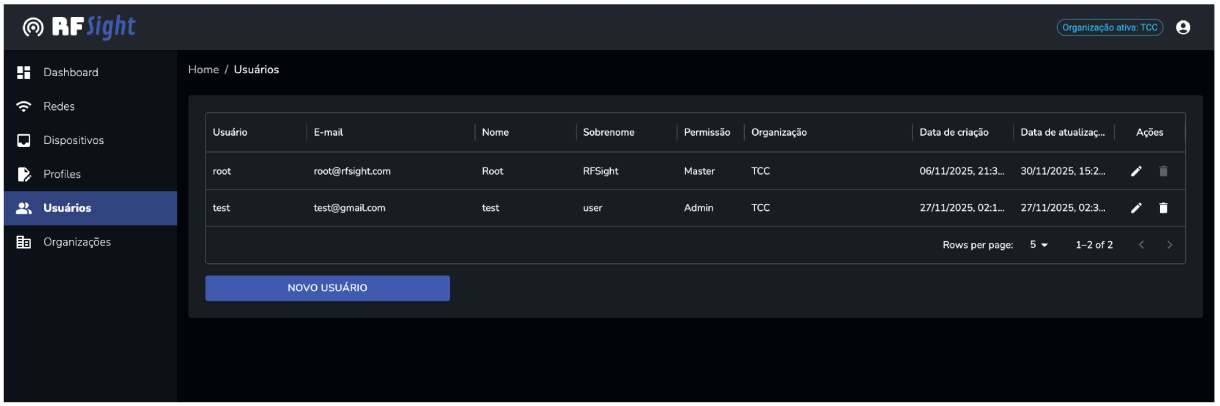
Figura 26 – Redefinição de senha



Fonte: Autor

Ademais, outra funcionalidade crucial relacionada aos usuários é a sua gestão. Ela pode ser realizada a partir da aba “Usuários” que pode ser acessada pelo menu lateral, desde que o usuário ativo tenha as devidas permissões (*Admin, Guest Admin* e *Master*). A tela de usuários apresenta a listagem completa de usuários e possibilita ações rápidas, como: criação, edição e deleção. A Figura 27 apresenta a página de usuários.

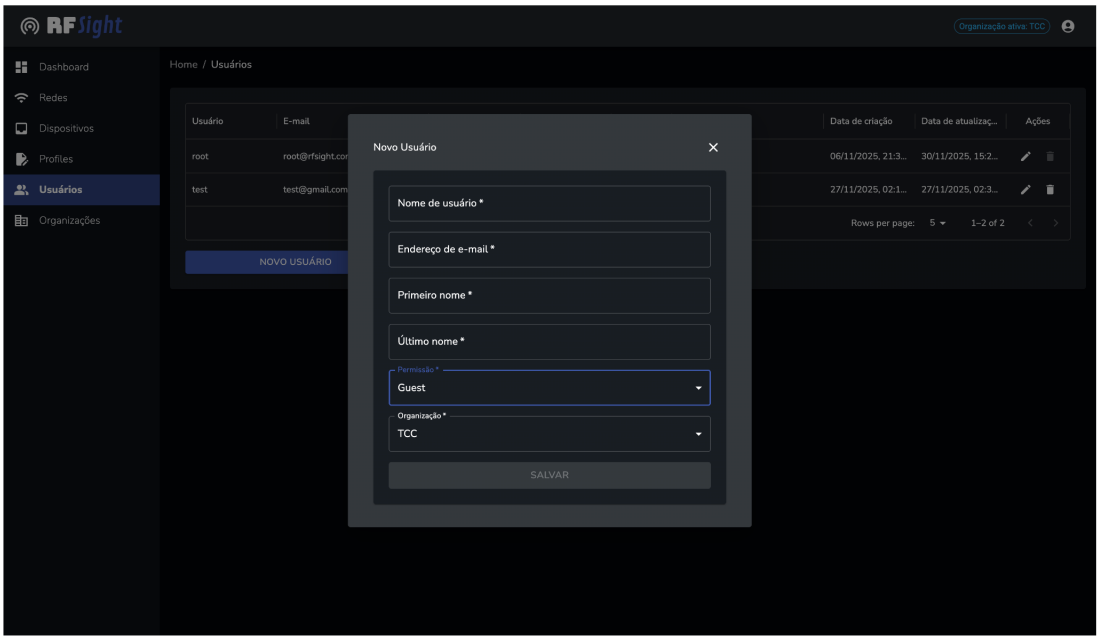
Figura 27 – Usuários



Fonte: Autor

As ações presentes na página de usuários, conforme a Figura 27, são condicionadas à permissão do usuário ativo e são definidas por: criar, editar e remover usuários. Para criar ou editar usuários, há uma série de validações de campos realizadas pelo *Frontend* para garantir a qualidade dos registros e o bom funcionamento do sistema, como validar o formato de *e-mail*, comprimento mínimo de nome de usuário, entre outras. Quanto à ação de remoção, é necessário confirmação direta antes de prosseguir. A Figura 28 contempla o formulário de criação de usuários.

Figura 28 – Criação de usuários

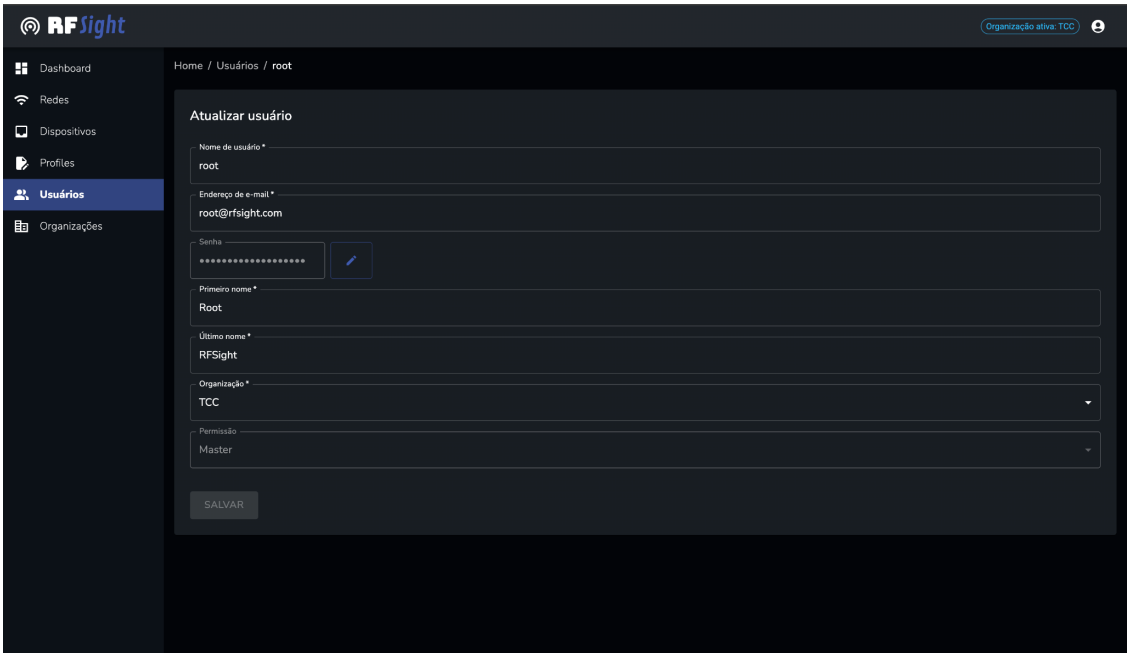


Fonte: Autor

O fluxo de criação de usuários iniciado no formulário da Figura 28, utiliza um mecanismo de geração de senhas, para que o administrador que esteja realizando o cadastro possa compartilhar de forma rápida as credenciais com o usuário que deve receber o acesso. As senhas do sistema seguem um requisito de segurança (controlado por *Frontend* e *Backend)*, descrito por: um mínimo de oito caracteres, contendo obrigatoriamente uma letra maiúscula, uma letra minúscula, um número e um caractere especial (#, ?, !, @, $, %, ^, *, -).

Para realizar a edição de um usuário existente, é possível clicar no ícone de lápis na linha do usuário desejado, que o sistema irá redirecionar para a página de gerência do usuário específico (podendo ser o próprio usuário ativo). Nessa tela, é possível realizar a alteração de dados, como *username*, *e-mail* e permissão, e também realizar um *reset* de senha do usuário, fornecendo primeiramente a senha atual e depois criando e confirmando a nova senha. A Figura 29 se refere à página de edição do usuário.

Figura 29 – Edição de usuário

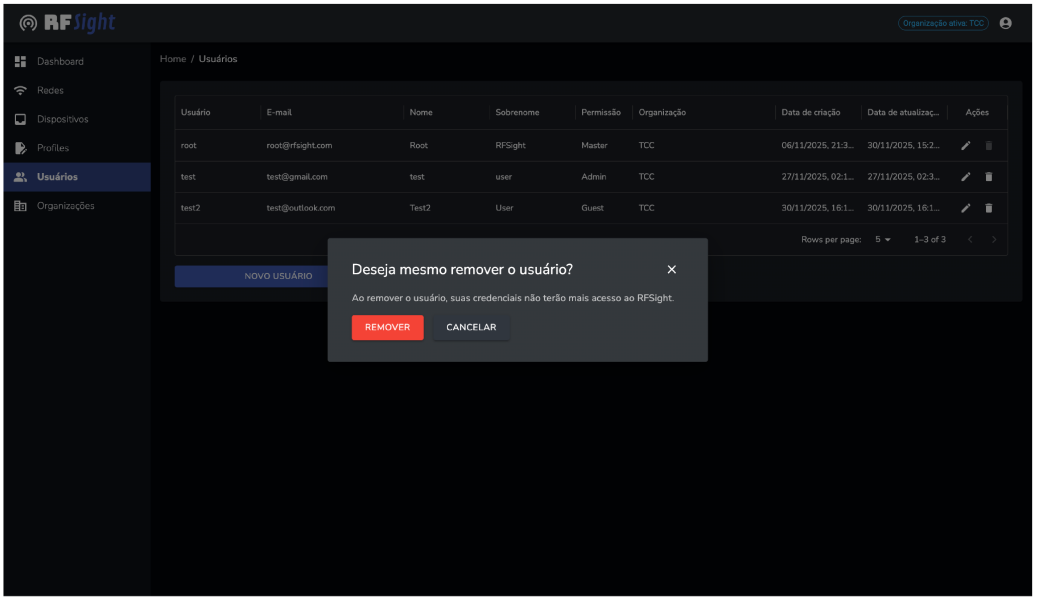


Fonte: Autor

Conforme apresenta a Figura 29, é possível alterar uma série de informações referentes ao usuário, porém alguns desses campos afetam consideravelmente a interação do usuário com o sistema, como por exemplo: alterar a organização ativa afeta quais redes e dispositivos o usuário terá acesso; mudar a permissão do usuário implica diretamente em que ações ele pode executar ao navegar pelo sistema.

Por fim, caso seja necessário remover algum usuário do sistema, é possível fazer isso a partir da lista de usuários (Figura 27), clicando no ícone de lixeira da linha do usuário desejado. Após isso , o sistema irá requisitar a confirmação da ação antes de concluir a ação destrutiva, conforme a Figura 30.

Figura 30 – Remoção de usuário



Fonte: Autor

Portanto, essa seção consolida as funcionalidades de autenticação e gestão de usuários do RFSight, requisitos cruciais para o bom funcionamento do sistema. As demais seções adentram o comportamento e questões mais específicas do gerenciamento e monitoramento de dispositivos, seguindo a arquitetura elaborada com base nos requisitos previamente definidos.

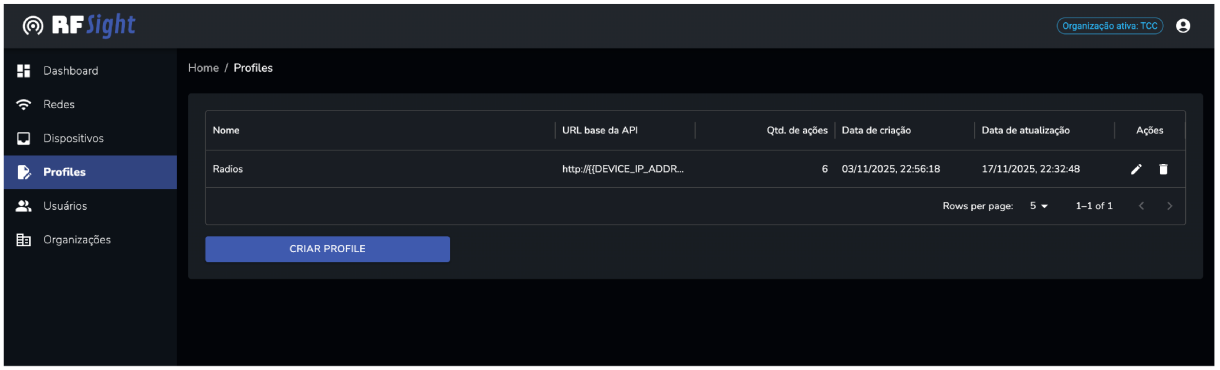
5.4.2. Camada de Abstração de Dispositivo

A estrutura central da proposta de valor do RFSight se encontra na Camada de Abstração de Dispositivo, responsável por tornar possível a adoção, gerenciamento e monitoramento de dispositivos independente de fabricantes. Esta seção trata de apresentar os detalhes técnicos da gestão de perfis (entidade *Profile*), assim como o funcionamento do *DeviceDriver*.

5.4.2.1. *Profiles* e ações

Conforme apresentado anteriormente, a entidade *Profile* desempenha um papel crucial na proposta central do RFSight, materializando a camada de abstração de comunicação com os dispositivos. Esta seção detalha os fluxos que envolvem a gestão desses perfis e a definição de suas respectivas ações dentro do sistema. A gestão de perfis de dispositivos está disponível a partir da tela de *Profiles*, acessível a partir do menu lateral, demonstrada na Figura 31.

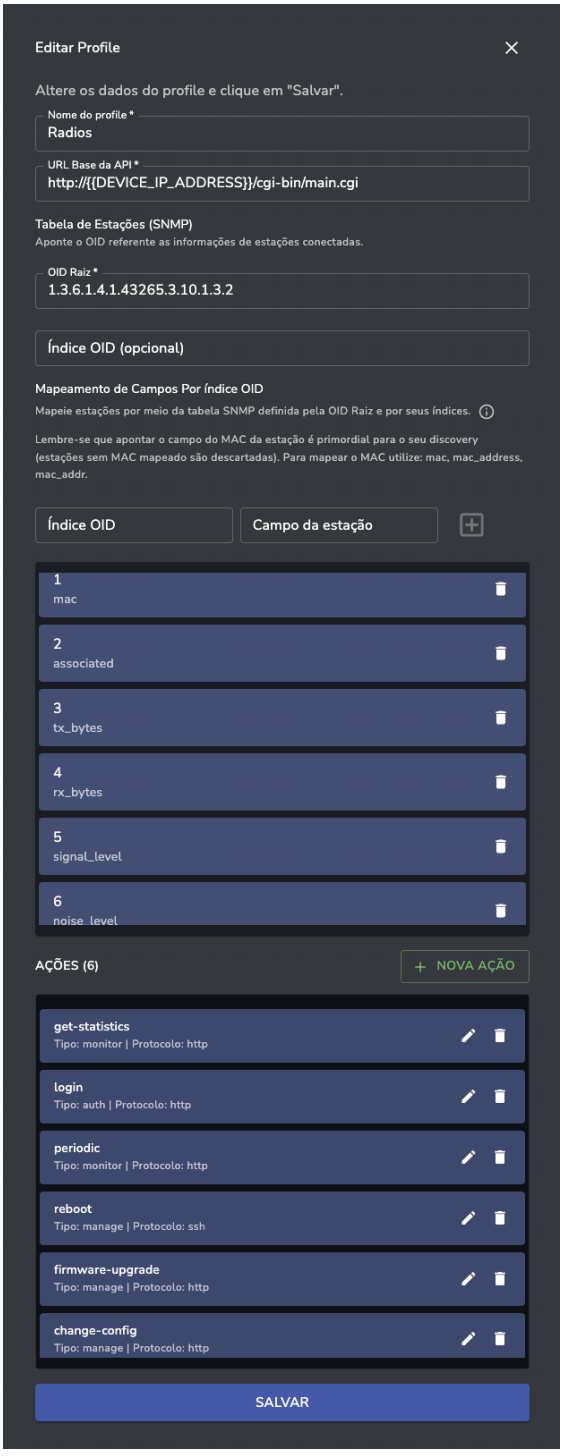
Figura 31 – *Profiles*



Fonte: Autor

A página de *Profiles* possibilita a criação, edição e remoção de perfis de dispositivos do sistema. A criação e edição de perfis requer uma série de informações específicas, dado que os perfis podem ser comparados a um manual de acesso aos dispositivos. Desse modo, a Figura 32 apresenta o formulário compartilhado de criação e edição de perfis.

Figura 32 – Formulário de *Profile*



Fonte: Autor

Conforme observado na Figura 32, cada Profile exige um nome único e parâmetros de comunicação para viabilizar a interação proativa do sistema após a adoção do dispositivo. Entre as informações mandatórias constam: a URL Base da API, que define o *endpoint* raiz para as requisições; e os dados da Tabela de

Estações (SNMP), incluindo o OID raiz e o mapeamento de campos por índices. O mapeamento SNMP é essencial para que o sistema identifique as estações clientes (CPEs) conectadas à *basestation* e colete métricas vitais.

Para garantir que a URL Base da API seja dinâmica e adaptável a diferentes dispositivos que compartilham o mesmo perfil, o sistema utiliza campos de merge (delimitados por "{{" e "}}"). Estes *placeholders* são substituídos em tempo de execução pelos valores reais do dispositivo alvo. Os campos disponíveis são:

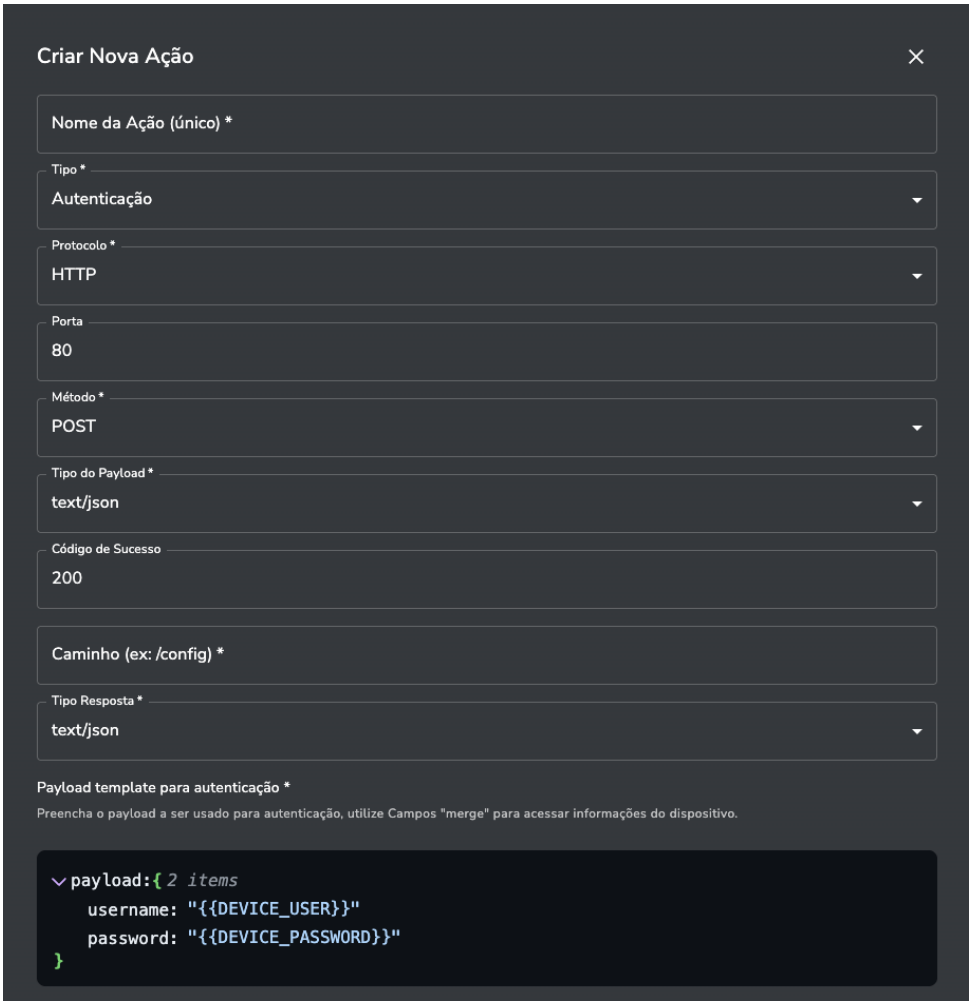
1. *{{DEVICE_IP_ADDRESS}}*
2. *{{DEVICE_MAC_ADDRESS}}*
3. *{{DEVICE_USER}}*
4. *{{DEVICE_PASSWORD}}*
5. *{{DEVICE_MODEL}}*

Dessa forma, no exemplo da Figura 32, caso um dispositivo adotado com este perfil possuir o endereço IP 15.0.0.10, o sistema interpretará a URL Base da API como "*http://15.0.0.10/cgi-bin/main.cgi*".

No que tange ao monitoramento das CPEs, é necessário indicar o OID da tabela SNMP correspondente (comumente referida como *stations* ou *peers*). O RFSight percorre essa tabela e associa os valores retornados aos campos internos do sistema. Além disso, o sistema busca por alguns campos específicos das estações para facilitar a exibição dos dados e um mapeamento mais completo, como por exemplo: endereço MAC; *tx_bytes* (total de *bytes* transmitidos) e *rx_bytes* (total de *bytes* recebidos). Estes contadores acumulativos são processados diferencialmente pelo algoritmo de topologia do sistema para calcular o *throughput* em tempo real.

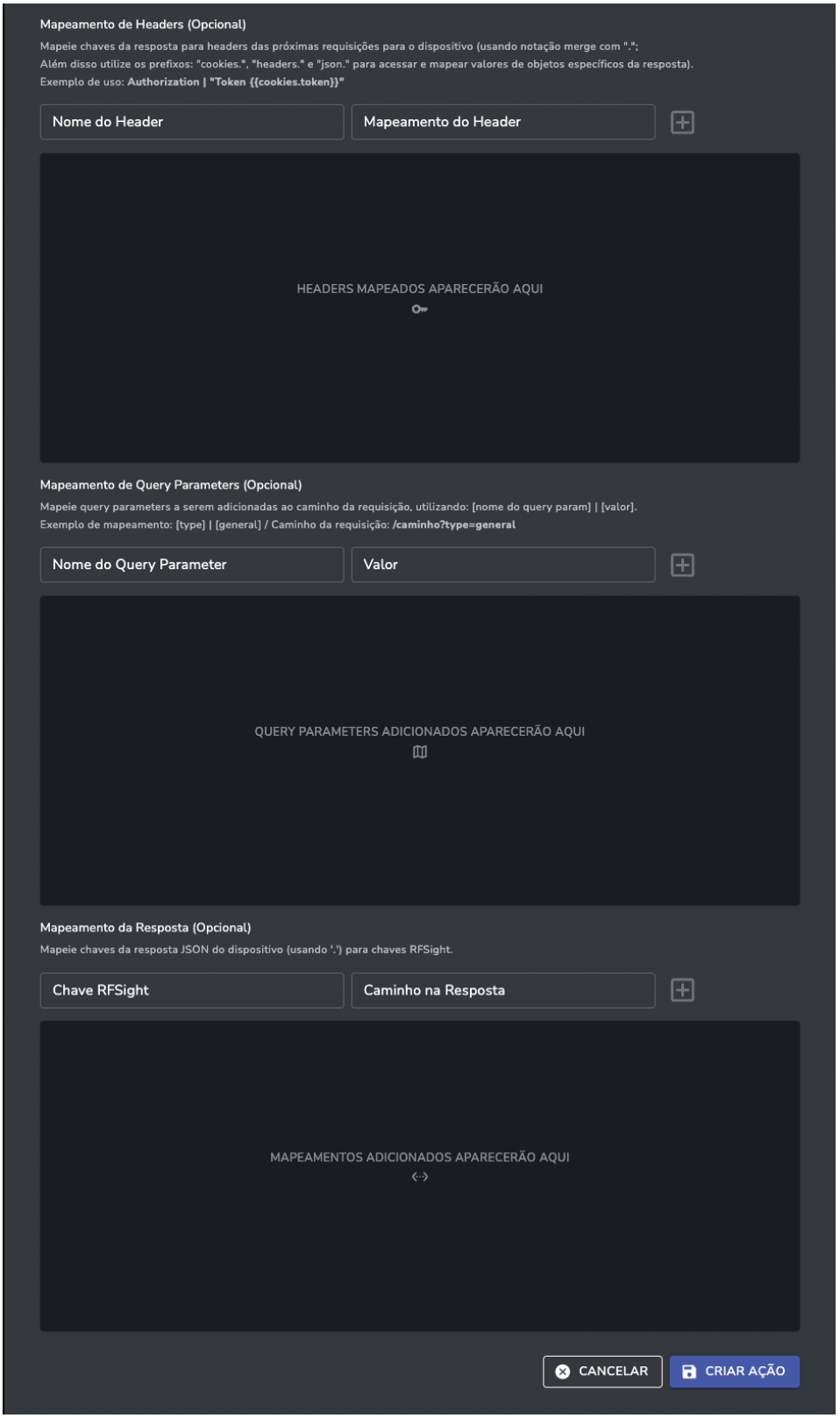
As ações do perfil, por sua vez, definem os comandos específicos que o sistema pode executar. Elas são configuradas através dos formulários apresentados nas Figuras 33 e 34, onde define-se o nome, tipo, protocolo e os parâmetros específicos de comunicação.

Figura 33 – Formulário Ação 1



Fonte: Autor

Figura 34 – Formulário Ação 2



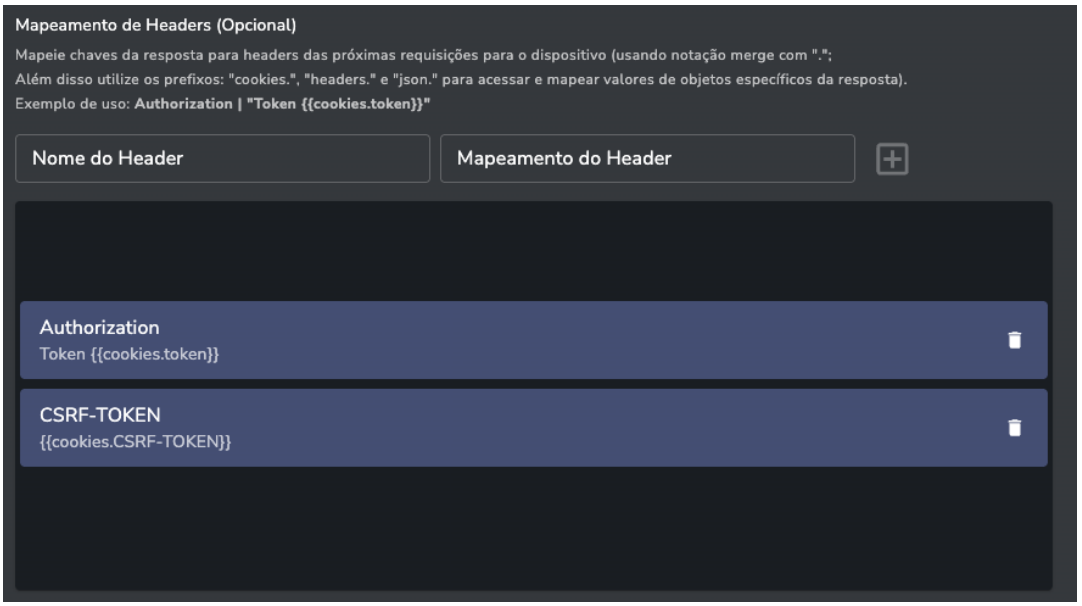
Fonte: Autor

As ações são categorizadas em três tipos funcionais: Autenticação (essencial para o funcionamento de qualquer outra ação HTTP), Monitoramento (coleta de dados do dispositivo) e Gerenciamento (configuração do dispositivo). Cada ação define o protocolo a ser utilizado (HTTP ou SSH) e os detalhes da execução.

Um aspecto importante é que apenas uma ação do tipo Autenticação pode ser definida por *Profile* e também, esse tipo de ação é o único que permite definir um *payload template* para realizar a autenticação (Figura 33). Isso se dá pelo fato de o RFSight automaticamente executar as ações do tipo Autenticação antes de prosseguir com a execução dos demais tipos. Assim, é necessário que a ação já tenha seu *payload* preparado sem que seja necessária qualquer interação do usuário, algo essencial para que as rotinas de monitoramento funcionem automaticamente.

No caso de requisições HTTP é possível realizar mapeamentos específicos de *headers*, *query parameters* e dados da resposta JSON. O processo de mapear *headers* é extremamente interessante para ações do tipo Autenticação, dado que *os* valores mapeados são usados posteriormente por outras ações, como apresenta a Figura 35.

Figura 35 – Mapeamento de *Headers*

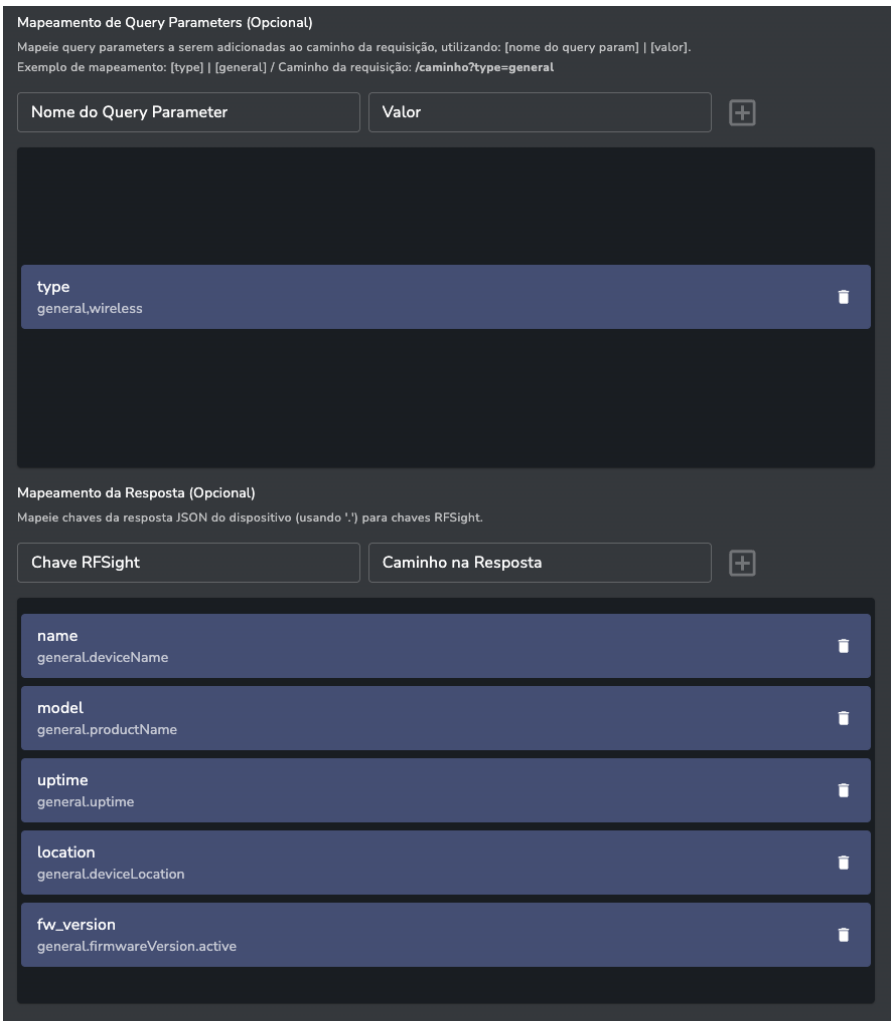


Fonte: Autor

A Figura 35 demonstra como os campos de *merge* podem ser aplicados para extrair valores dinâmicos da resposta, utilizando prefixos como “*headers*.”, “*cookies*.” ou “*json*.” para acessar as respectivas estruturas de dados da resposta HTTP.

As outras duas possibilidades de mapeamento para uma ação HTTP, mapeamento de *query parameters* e de dados da resposta (JSON), estão dispostas na Figura 36.

Figura 36 – Mapeamento de *Query Parameters* e Resposta JSON



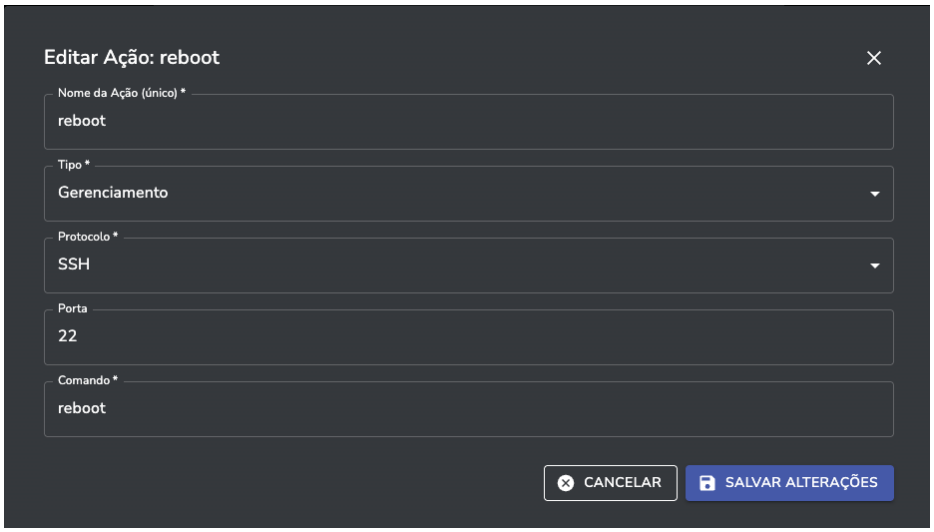
Fonte: Autor

Os *query parameters* permitem a parametrização das requisições GET, como em filtros de API (ex: “?type=general”). O usuário define o par chave-valor conforme a necessidade do *endpoint*. Já no mapeamento da resposta, utiliza-se a notação de ponto para navegar na estrutura do JSON retornado. Por exemplo, para extrair o

tempo de atividade de um objeto JSON aninhado como *{ "system": { "uptime": 12300 }}*, o mapeamento deve ser configurado como *system.uptime*, associando-o ao campo interno correspondente do RFSight.

Por fim, em relação às ações do tipo SSH, a configuração disponibilizada resume-se à definição da porta de serviço e do comando de terminal a ser executado no dispositivo. A Figura 37 exemplifica uma ação de gerenciamento do tipo SSH, configurada para enviar o comando "*reboot*" (reinicializar).

Figura 37 – Ação de Gerenciamento SSH



Fonte: Autor

Portanto, a gestão de perfis e ações constitui o núcleo da flexibilidade do RFSight. Ao permitir a definição granular de requisições HTTP e comandos SSH, aliada ao mapeamento dinâmico de dados e cálculos de métricas em tempo real, o sistema viabiliza a interoperabilidade com dispositivos de múltiplos fabricantes sem a necessidade de alteração no código-fonte, concretizando a base da camada de abstração proposta neste trabalho e apresentada na seção seguinte.

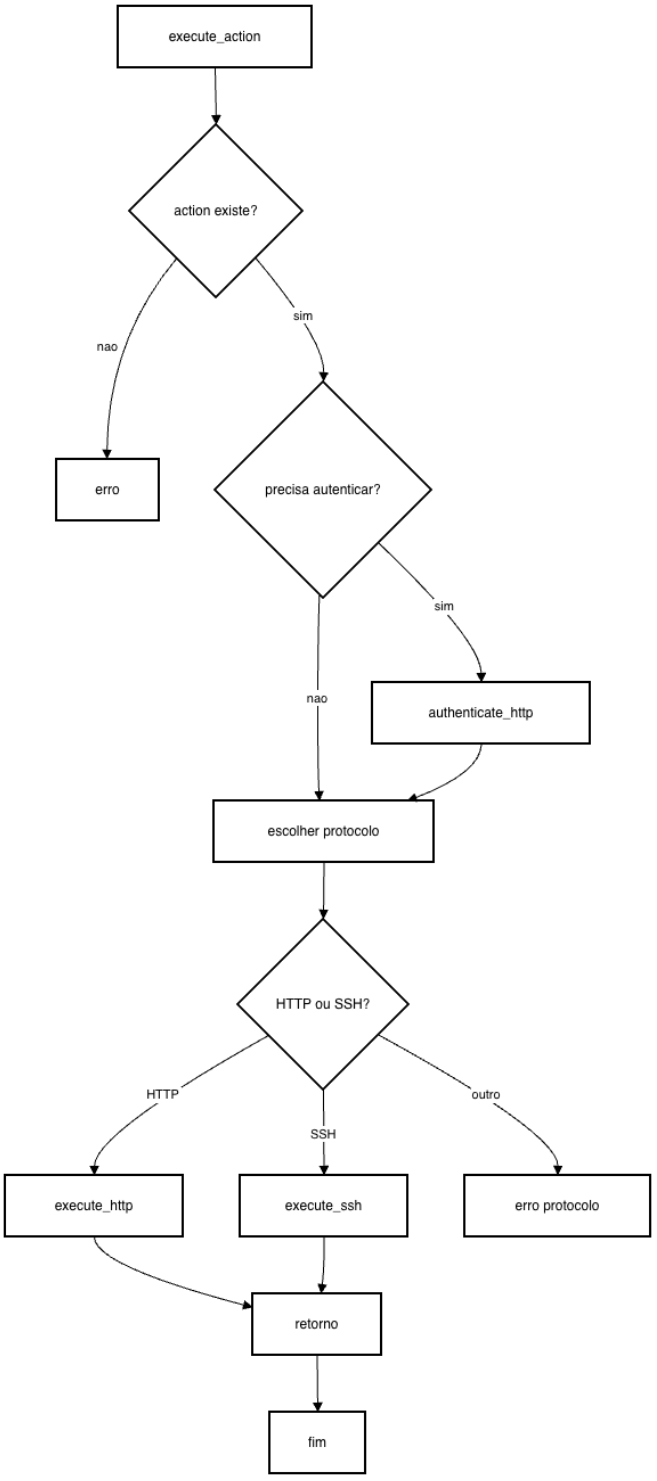
5.4.2.2. DeviceDriver

O componente central da camada de abstração responsável por executar os três tipos de ações disponíveis no *Profile* é o *DeviceDriver* (conforme Figura 15). Este serviço atua análogo a um intérprete universal: ele consome as definições do *Profile*, hidrata variáveis dinâmicas (substituindo placeholders como {{*DEVICE_IP*}} pelos dados reais dos dispositivos) e traduz ações genéricas em comandos específicos (HTTP ou SSH) para o dispositivo alvo.

O *DeviceDriver* é projetado para ser consumido pelos demais serviços que compõem o funcionamento do *Backend*, como os controladores de Adoção, Configuração e Monitoramento. Estes controladores implementam todos os fluxos e rotinas de execução até o ponto de realizar a comunicação com o dispositivo, momento em que a responsabilidade recai sobre o *driver*. Para que a performance do sistema se mantenha estável, os controladores utilizam estratégias assíncronas para chamadas que necessitem a interação com os dispositivos, fazendo com que a responsabilidade do *DeviceDriver* seja exclusivamente a comunicação.

Além da execução, o driver é responsável por mapear as respostas brutas para o formato padronizado do sistema e garantir a resiliência da comunicação, implementando estratégias automáticas de retentativa (*retries*) HTTP baseadas em *status codes* para mitigar possíveis falhas na rede. A Figura 38 ilustra o fluxo de execução deste componente.

Figura 38 – Fluxo *DeviceDriver*



Fonte: Autor

Conforme a Figura 38, o processo de execução de ações no dispositivo a partir de seu *Profile* configurado inicia com a validação da existência da ação solicitada e, caso necessário, realiza automaticamente o procedimento de

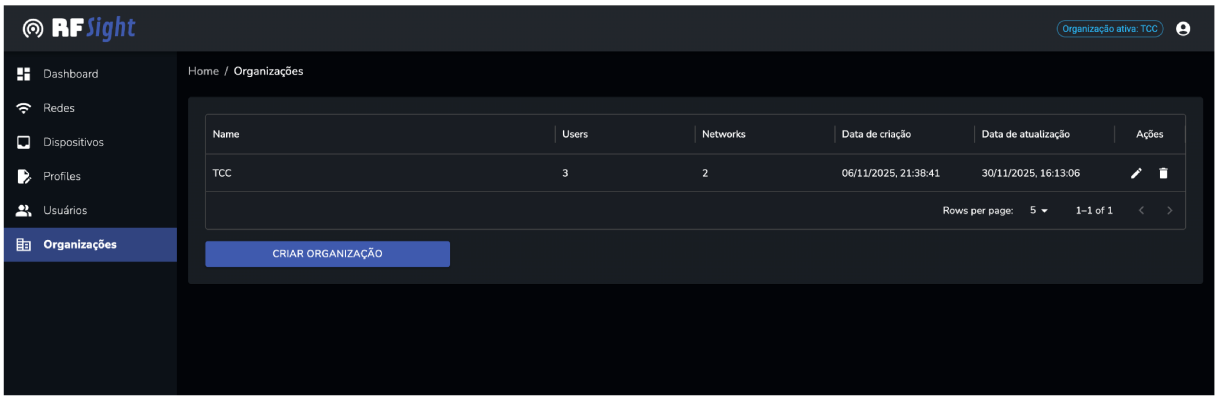
autenticação HTTP antes de prosseguir. Em seguida, a lógica encaminha a requisição para o protocolo adequado, distinguindo entre operações HTTP e SSH, de acordo com a definição contida no *Profile* do dispositivo. Cada protocolo é então tratado por seu respectivo método especializado, responsável por executar a ação e retornar o resultado final consolidado com o mapeamento de *headers,* dados a partir de chaves da resposta (JSON), se existentes na configuração da ação.

Assim, a combinação do *DeviceDriver* com a entidade *Profile* compõe a Camada de Abstração de Dispositivo. Essa arquitetura permite que o sistema gerencie equipamentos de múltiplos fabricantes de forma transparente e escalável, desacoplando a lógica de negócio das especificidades de *hardware* e assegurando que a integração de novos modelos possa ser realizada apenas via configuração, sem alterações no código-fonte.

5.4.3. Gerenciamento

Primeiramente, o gerenciamento de dispositivos no RFSight inicia-se pela definição da hierarquia de entidades do sistema, sendo definida por Organizações > Redes > Dispositivos. Para adotar um dispositivo e gerenciá-lo, é necessário que exista ao menos uma Organização com uma Rede disponível. Para criar e alterar organizações, é necessário acessar a tela de organizações pelo menu lateral. Conforme demonstra a Figura 39.

Figura 39 – Tela de Organizações

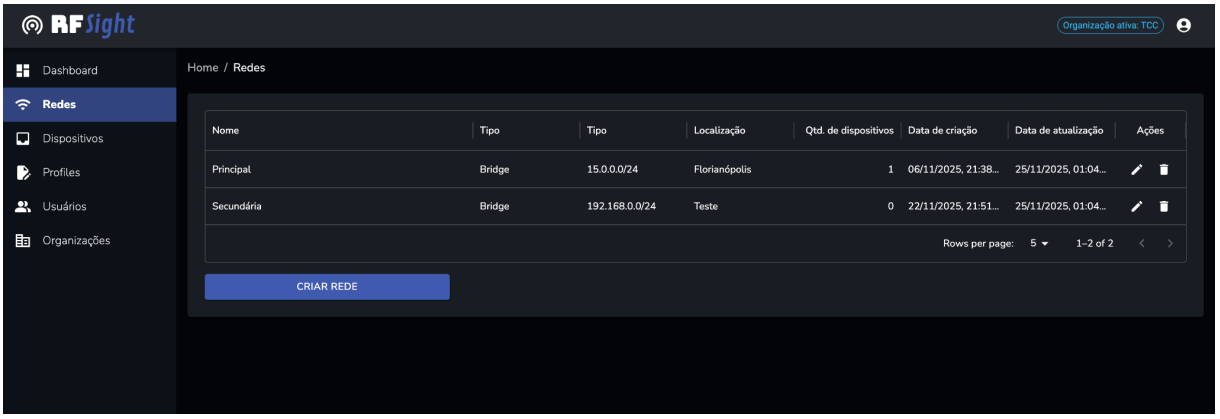


Fonte: Autor

A Figura 39 apresenta a tela de gestão de organizações do RFSight, o nível mais alto da hierarquia de entidades do sistema. Nessa tela é possível criar, editar e remover organizações (mediante confirmação). Entre alguns detalhes importantes estão: o único campo necessário para o cadastro e disponível para a edição de uma organização é o seu Nome; cada usuário pode ter apenas uma organização ativa por vez; Ao remover uma organização, todas as redes e seus dispositivos relacionados serão perdidos.

Com uma organização criada, é possível cadastrar redes relacionadas, onde posteriormente serão adotados dispositivos. A gestão de redes pode ser feita pela tela de redes, disponível pelo menu lateral, assim como apresenta a Figura 40.

Figura 40 – Tela de Redes

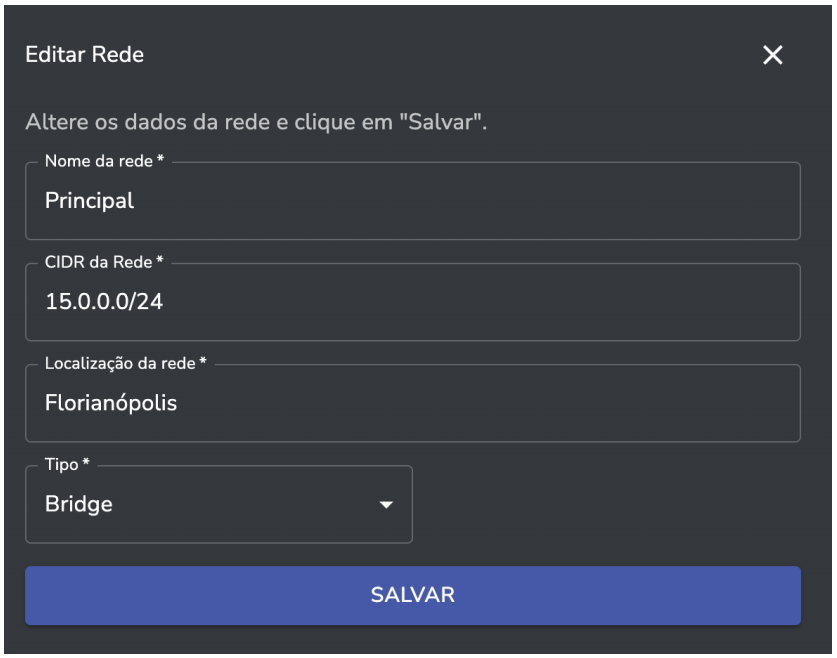


Fonte: Autor

Entre as ações disponíveis para a gestão de redes, estão criar, editar e remover redes. Sendo que o requisito dos campos e dados para criação e edição é a mesma, assim como a remoção de uma rede requer confirmação expressa do usuário, dado que todos os dispositivos relacionados serão perdidos no processo.

Para criar ou editar uma rede é necessário fornecer um nome (único), apontar o CIDR da rede (utilizado para as tarefas de descoberta de dispositivos), a localização e o tipo da rede (que para o protótipo está fixo em “Bridge”). Esse formulário pode ser visualizado na Figura 41.

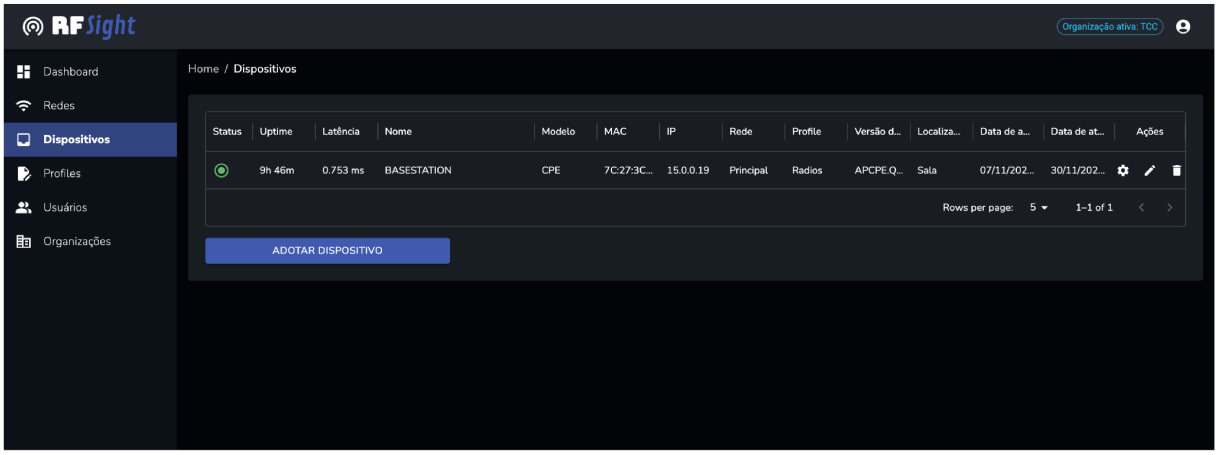
Figura 41 – Formulário de Rede



Fonte: Autor

Ao concluir a criação da organização e da rede, é possível prosseguir para a adoção de dispositivos no RFSight. O processo de adoção pode ser considerado o ponto de partida e também um momento crucial do gerenciamento e monitoramento correto do dispositivo alvo, pois é onde será verificado a conectividade do dispositivo (por meio de testes de conexão TCP), além da busca de informações iniciais a partir da execução de ações de monitoramento durante a rotina de adoção. Uma falha no momento da adoção pode significar algum problema de conectividade com o dispositivo ou até mesmo uma seleção ou configuração incorreta de perfil. O gerenciamento de dispositivos ocorre a partir da tela de dispositivos, apresentada na Figura 42.

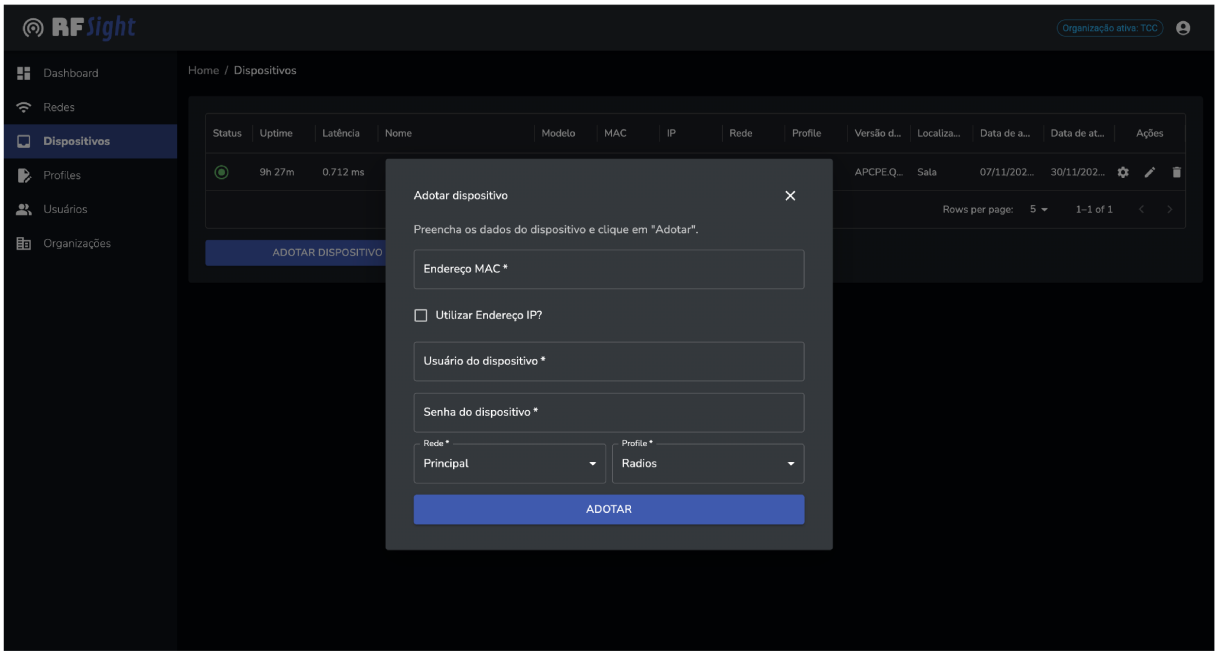
Figura 42 – Tela de Dispositivos



Fonte: Autor

A adoção de dispositivos pode ser realizada por meio da descoberta utilizando um endereço MAC, IP ou ambos. O usuário deve fornecer as credenciais do dispositivo a serem usadas para autenticação das ações cadastradas no perfil selecionado, e selecionar a rede de que o dispositivo deve fazer parte. A Figura 43 apresenta a tela de adoção, acessível a partir da tela de Dispositivos.

Figura 43 – Tela de Adoção

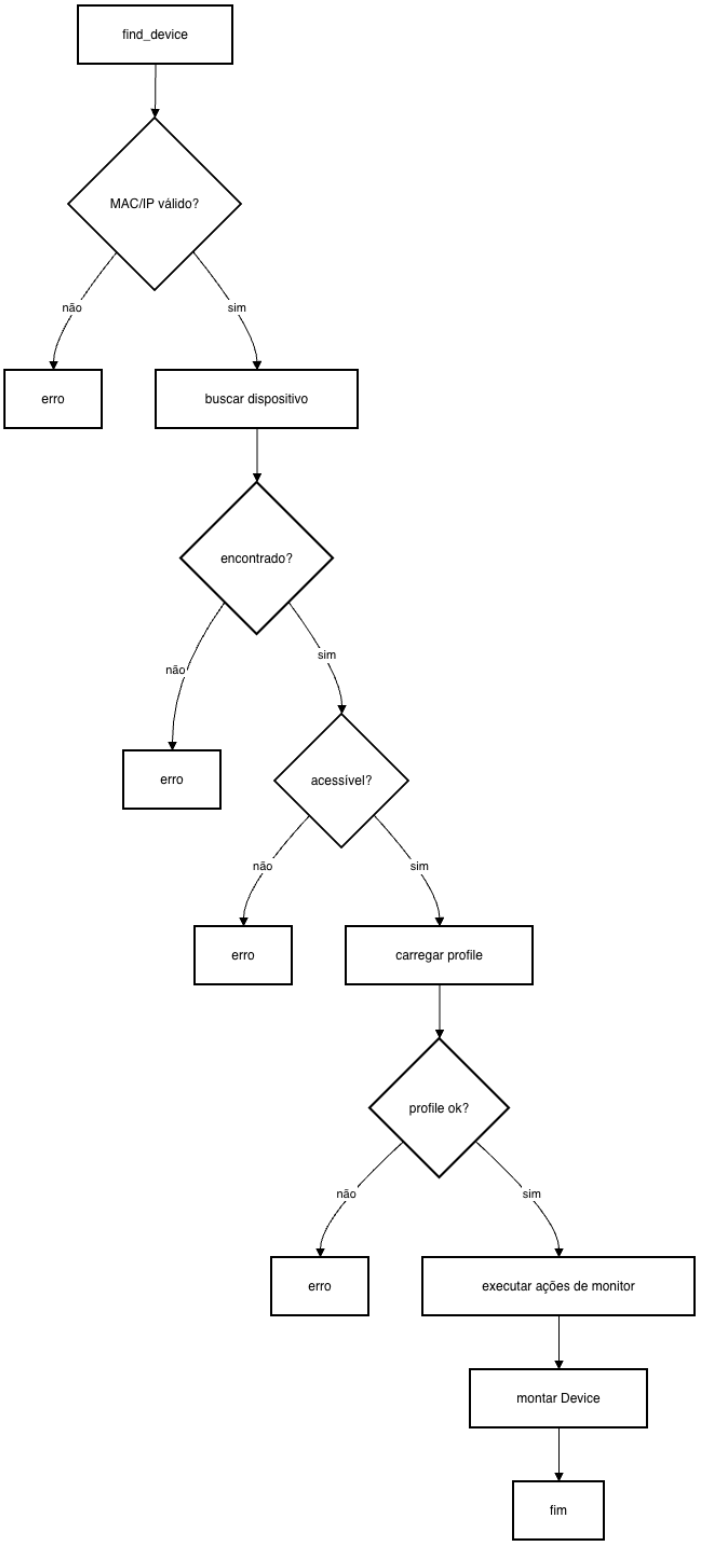


Fonte: Autor

Com os dados fornecidos, o fluxo de adoção inicia no *Backend*, orquestrado pelo *AdoptionController*, que executa em segundo plano a varreduras contínuas de dispositivos da rede por meio da ferramenta *arp-scan*. A cada execução o controlador atualiza uma tabela em memória (*available_devices*) que é consultada pela requisição de adoção de dispositivos quando enviada pelo usuário.

A partir disso, o sistema então verifica se o dispositivo é “conhecido” e caso encontrado, parte para iniciar a adoção: programa a execução de ações do tipo Monitoramento do perfil selecionado para coletar metadados essenciais do dispositivo, como nome do sistema, modelo e versão de *firmware*. Ao fim, caso as ações tenham sucesso, o dispositivo é registrado no banco de dados e passa a aparecer na lista da rede em que foi adotado. A Figura 44 apresenta o fluxo simplificado do método principal do controlador.

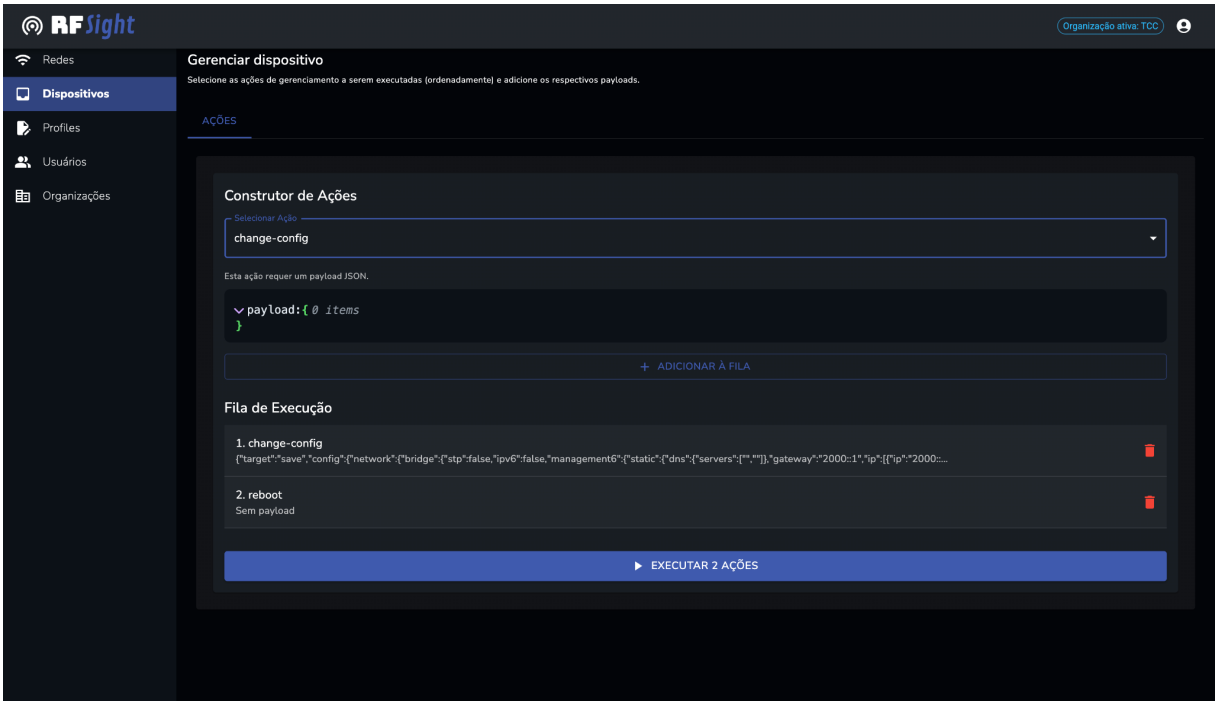
Figura 44 – Fluxo *AdoptionController*



Fonte: Autor

Após a adoção completa, o RFSight passa a monitorar automaticamente o estado do dispositivo a partir das rotinas de monitoramento e, desde que o dispositivo esteja marcado como "*Online*", o usuário pode clicar no ícone de engrenagem na lista da rede (Figura 42) para executar ações do tipo *Gerenciamento*. Além disso, o usuário pode alterar os dados de adoção do dispositivo a qualquer momento por meio da ação de "editar" na lista de dispositivos, e também remover dispositivos do sistema (ícones de lápis e lixeira, respectivamente, conforme Figura 42). É possível analisar a tela de gerenciamento de dispositivos a partir da Figura 45.

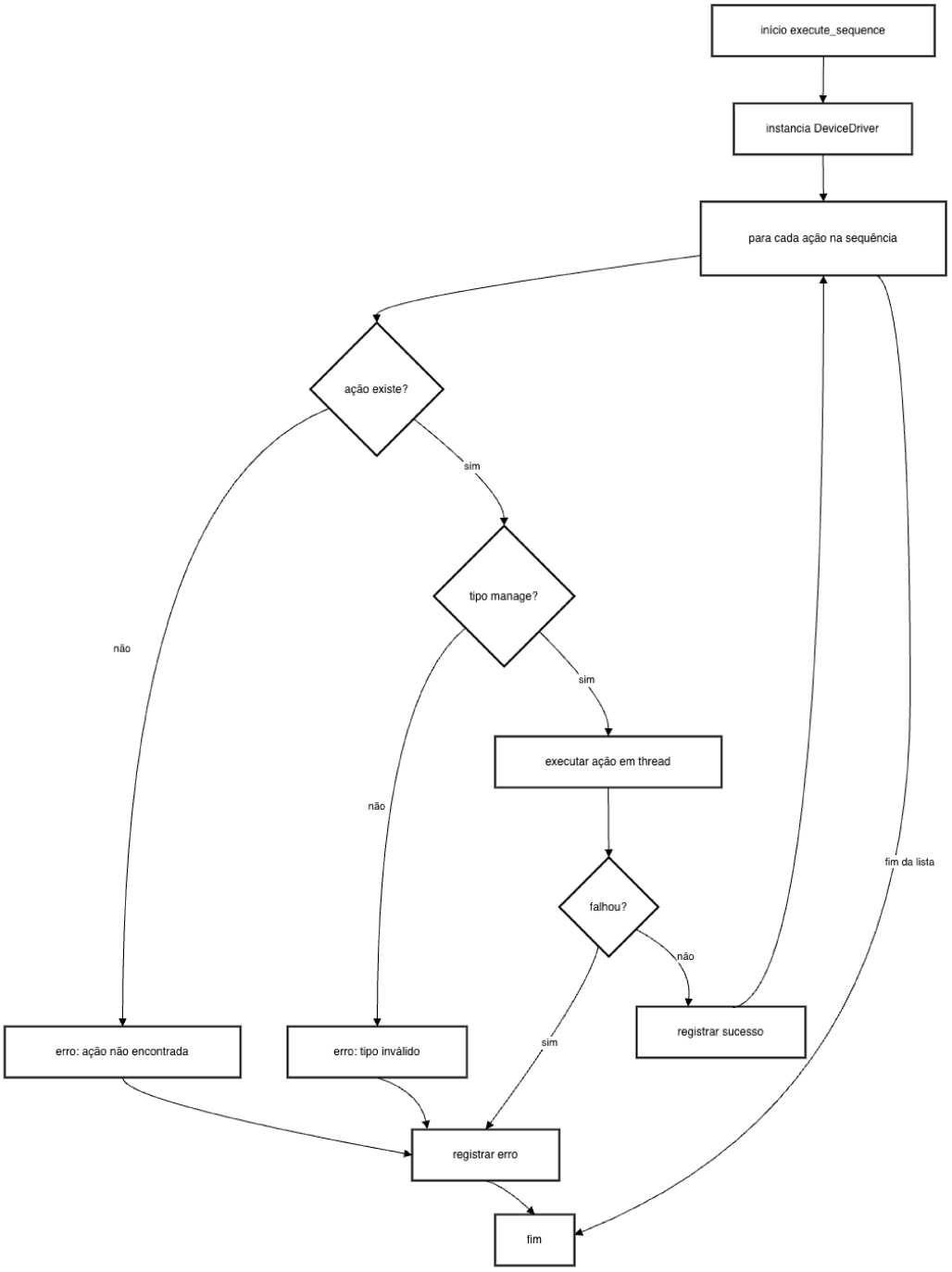
Figura 45 – Tela de gerenciamento de dispositivos



Fonte: Autor

A Figura 45 destaca o componente "Construtor de Ações". Nele, o usuário seleciona ações do tipo Gerenciamento disponíveis no perfil e configura os *payloads* necessários, suportando texto plano, JSON ou arquivos (convertidos automaticamente para *base64*). Ao iniciar a execução, a requisição é enviada ao *ConfigController*, cujo fluxo é detalhado na Figura 46.

Figura 46 – Fluxo *ConfigController*



Fonte: Autor

Conforme observado na Figura 46, o *ConfigController* atua como um orquestrador que garante a execução sequencial e atômica das ações, delegando a complexidade da comunicação (protocolo e *parsing*) para a camada de abstração do *DeviceDriver*, tendo cada execução sendo realizada em uma *thread*, a fim de não bloquear o fluxo principal do *Backend*. Essa abordagem assegura que operações

complexas, compostas por múltiplos passos, possam ser realizadas de forma controlada e auditável.

5.4.4. Monitoramento

O monitoramento em tempo real (requisitos RF05 e RF06) constitui o núcleo operacional do RFSight, sendo responsável por fornecer visibilidade instantânea sobre as métricas da rede. Esta funcionalidade é suportada por uma arquitetura de comunicação assíncrona e bidirecional entre o *Backend* e o *Frontend*, orquestrada pelo protocolo WebSocket.

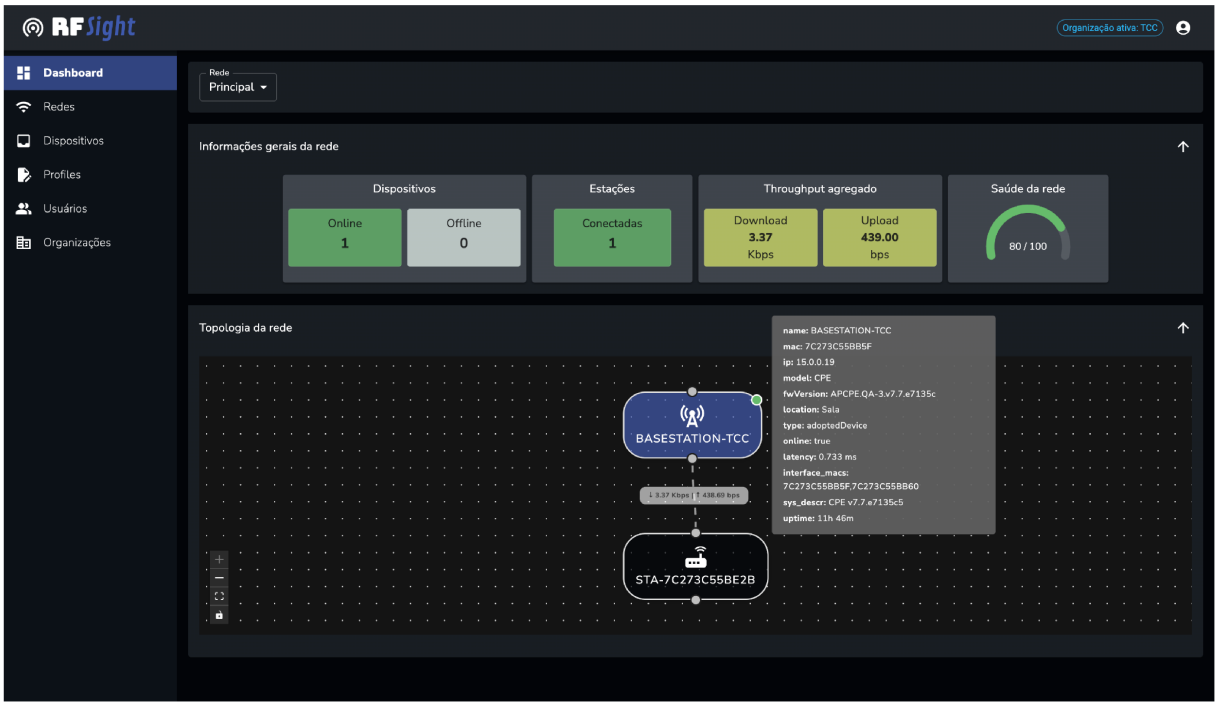
No servidor, a inicialização da aplicação *Backend* dispara duas tarefas principais em segundo plano, gerenciadas pelo *MonitorController*: o *device_monitor_loop* e o *topology_loop*. A primeira tarefa, executada a cada 10 segundos, verifica a conectividade ICMP (*ping*) de todos os dispositivos adotados para determinar o *status* (*Online*/*Offline*) e a latência. Para os dispositivos ativos, o controlador instancia dinamicamente os *DeviceDrivers* correspondentes para executar ações de coleta de métricas definidas no perfil, como por exemplo uptime e versão de *firmware*.

A segunda tarefa, *topology_loop*, executada a cada 30 segundos, é responsável pelo mapeamento físico e lógico da rede. Além de varreduras ARP, este processo utiliza o protocolo SNMP para interrogar os dispositivos adotados (*basestations)* sobre as estações clientes (CPEs) conectadas. O sistema percorre a tabela de estações (cujo OID raiz é definido no *Profile*) e extrai dados críticos baseados no mapeamento de índices, como endereço MAC e os contadores de tráfego (*tx_bytes* e *rx_bytes*). Para estações que possuem a informação de endereço MAC, são realizados cruzamentos de dados com as varreduras ARP a fim de mapear seus endereços IPs.

Os dados brutos coletados são transmitidos via *broadcast* para todos os clientes conectados através do *websocket_connection_manager*. No *Frontend*, um *middleware* de conexão WebSocket personalizado intercepta essas mensagens e

atualiza o estado da aplicação, permitindo que a *interface* calcule métricas derivadas em tempo real. Essas informações convergem na tela de *dashboard* da rede, acessível pelo menu lateral e demonstrada na Figura 47.

Figura 47 – *Dashboard* da rede



Fonte: Autor

A Figura 47 destaca os gráficos agregados da rede, juntamente do visualizador de topologia, que renderiza automaticamente os nós (dispositivos e estações) e suas conexões físicas, desenvolvido com a biblioteca React Flow (XYFLOW, 2025). Cada nó do gráfico de topologia apresenta um consolidado de informações de monitoramento em um *tooltip* (ao passar o ponteiro do *mouse* pelo nó). O painel também exibe a métrica de Saúde da Rede, calculada pelo algoritmo do RFSight para oferecer uma visão rápida da qualidade do serviço. O cálculo desta métrica pondera três fatores principais:

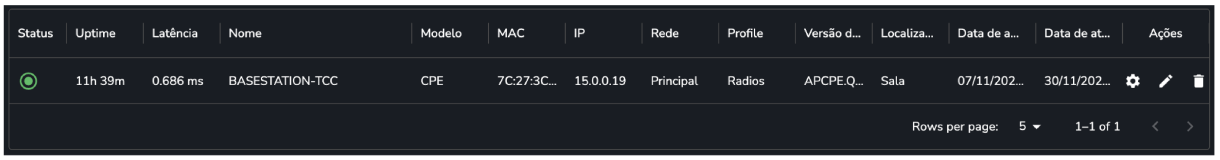
1. Disponibilidade (50%): Relação entre dispositivos *online* e o total de dispositivos.
2. Latência (30%): Penalização progressiva baseada na latência média da rede (com base de referência de 200ms).

3. Throughput (20%): Comparação entre o tráfego agregado atual e a capacidade estimada da rede (baseada no número de estações ativas, com limiar de 5 *mbps*).

Outras métricas essenciais disponíveis no *dashboard* são as de *throughput*: agregado (apresentado pelo gráfico “Throughput agregado” e por enlace (apresentado em cada conexão entre nós, em que um dos nós seja do tipo *wifiStation*). O cálculo do *throughput ocorre a partir* da diferença entre a leitura atual e a anterior do total de *bytes* transmitidos e recebidos, dividindo pelo intervalo de tempo decorrido para obter a velocidade instantânea em *bits* por segundo (*bps*).

Para otimizar a experiência do usuário, o Frontend utiliza funções auxiliares de formatação, convertendo automaticamente as grandezas para escalas legíveis (ex: bps para Mbps ou Gbps, segundos para dias/horas), garantindo uma leitura fluida dos dados técnicos. Além da visão macroscópica do *dashboard*, o sistema permite o acompanhamento granular em tempo real através da lista de dispositivos, conforme ilustrado na Figura 48.

Figura 48 – Monitoramento lista de dispositivos



Fonte: Autor

A Figura 48 destaca a capacidade de monitoramento individualizado diretamente na listagem da rede. A *interface* fornece *feedback* visual imediato sobre o *status* (*Online*/*Offline*) e apresenta métricas vitais como tempo de atividade (*Uptime*), latência, identificação (*Hostname*/IP/MAC) e versão de *firmware*.

Em suma, o módulo de monitoramento consolida a arquitetura reativa do RFSight, integrando a coleta robusta de dados heterogêneos (via ICMP, SNMP e ARP) no *Backend* com a visualização dinâmica no *Frontend*. A utilização de WebSockets para o transporte de telemetria em tempo real, aliada ao

processamento de métricas derivadas diretamente na camada de aplicação, assegura que a *interface* permaneça responsiva e atualizada sem sobrecarregar a rede. Esta funcionalidade atende integralmente aos requisitos de observabilidade do sistema, instrumentalizando o provedor regional com dados acionáveis para uma gestão proativa da infraestrutura.

Com o desenvolvimento das funcionalidades de gerenciamento e monitoramento concluído, o próximo capítulo dedica-se à validação destas implementações através de testes em ambiente simulado.

6. TESTES E VALIDAÇÕES

Para validar a arquitetura proposta e o protótipo desenvolvido, foi realizado um conjunto de testes em um ambiente controlado, simulando a infraestrutura de um provedor de Internet regional. Esta seção descreve o cenário, os testes executados e os resultados obtidos.

6.1. Cenário

O ambiente de teste foi configurado para validar os requisitos funcionais centrais, especialmente a adoção (RF01), configuração (RF03) e monitoramento (RF05). O cenário consistiu em:

1. **Sistema RFSight:** Executado em um container Docker em uma máquina *host* na rede local.
2. **Dispositivos Alvo:** Foi utilizada uma topologia P2MP simplificada, composta por uma *basestation* e uma CPE, ambas de marca *white label* (não-proprietária), cujas *interfaces* de gerenciamento são acessíveis via API HTTP REST e SSH.
3. **Rede:** A *basestation* foi conectada à mesma sub-rede local (LAN) que a máquina *host* do RFSight, permitindo a descoberta via ARP (*Layer 2*) e a comunicação via HTTP. A CPE foi conectada sem fio à *basestation* em modo “cliente”.

6.2. Testes realizados

Durante a preparação para os testes, foi realizada a descoberta das rotas de API e dos comandos aceitos pelo dispositivo a ser adotado, a partir de uma análise exploratória das requisições realizadas pela *interface* de gerenciamento dos próprios dispositivos (aba “Network” do console de desenvolvedor do navegador).

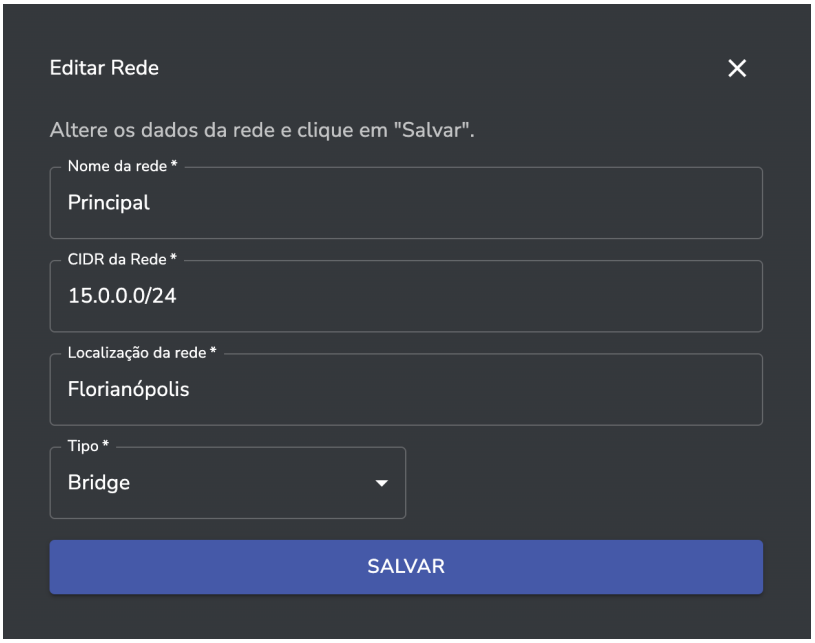
Ademais, o mapeamento da tabela SNMP das estações conectadas aconteceu também por meio da análise dos OIDs SNMP expostos pela *basestation*, efetuando comparações em paralelo com as informações da própria *interface* de gerência do dispositivo para identificar seus índices.

Com isso, foi possível mapear cinco ações (sendo uma para autenticação, duas para gerenciamento e duas para monitoramento) e seis índices OID das estações (*mac*, *associated*, *tx_bytes*, *rx_bytes*, *signal_level* e *noise_level*), que possibilitaram a execução da sequência de validações do RFSight.

O fluxo de validação seguiu as etapas lógicas da operação do sistema:

1. Criação da Rede: Primeiramente, foi criada a rede “Principal” no sistema, provendo o CIDR da rede em que os dispositivos se encontram conectados (15.0.0.0/24), assim como destaca a Figura 49.

Figura 49 – Rede Principal



Fonte: Autor

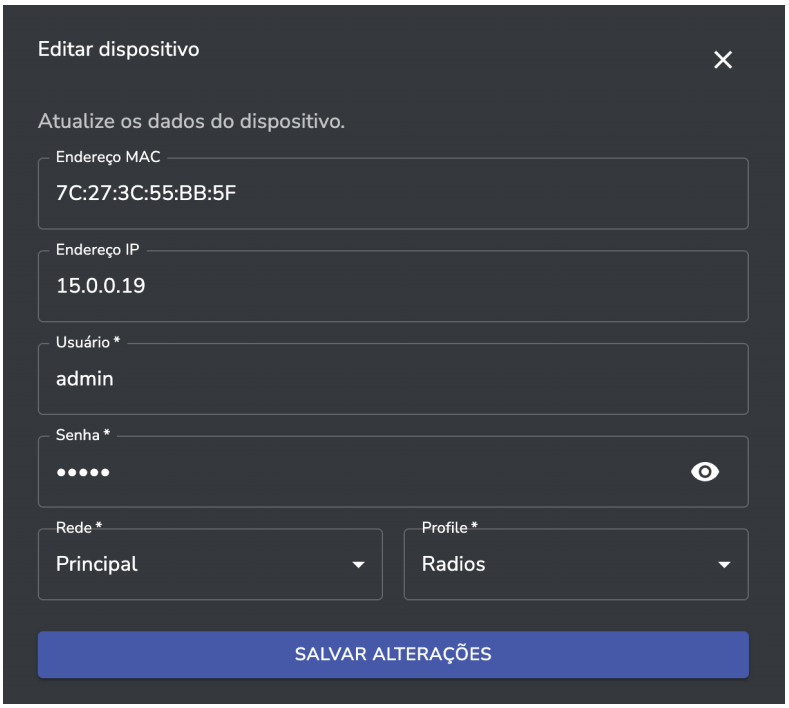
2. Criação do Perfil (*Profile*): Segundamente, foi criado um *Profile* no RFSight específico para os dispositivos do cenário, nomeado “Radios” (Figura 32). Este perfil foi configurado com a *URL Base da API* correta (*http://{{DEVICE_IP_ADDRESS}}/cgi-bin/main.cgi*) e um conjunto de *Actions*:

   a. ***auth* (Autenticação, HTTP POST):** Realizar a autenticação na API do dispositivo, enviando credenciais em *payload JSON* e mapeando os *tokens* de sessão recebidos para *headers* de futuras ações.
   b. ***get_statistics* (Monitoramento, HTTP GET):** Buscar métricas e dados do dispositivo utilizando o mapeamento de respostas JSON.
   c. ***periodic* (Monitoramento, HTTP GET):** Buscar métricas e dados do dispositivo utilizando o mapeamento de respostas JSON.
   d. ***change-config* (Gerenciamento, HTTP POST):** Ação de gerenciamento para configuração geral do dispositivo utilizando *payload* de tipo JSON.
   e. ***reboot* (Gerenciamento, SSH):** Ação de gerenciamento para reinicializar o dispositivo.

3. **Adoção do Dispositivo:** A basestation foi ligada na rede. O *AdoptionController* a detectou via ARP. Pela *interface* do RFSight, a basestation foi selecionada

para adoção, fornecendo-se o endereço IP (15.0.0.19), credenciais de acesso e o *Profile* “Radios”. O sistema executou o fluxo de adoção, autenticou o dispositivo, executou as ações de monitoramento e salvou a *basestation* como um dispositivo gerenciado. A Figura 50 apresenta os dados atualizados de adoção da *basestation*.

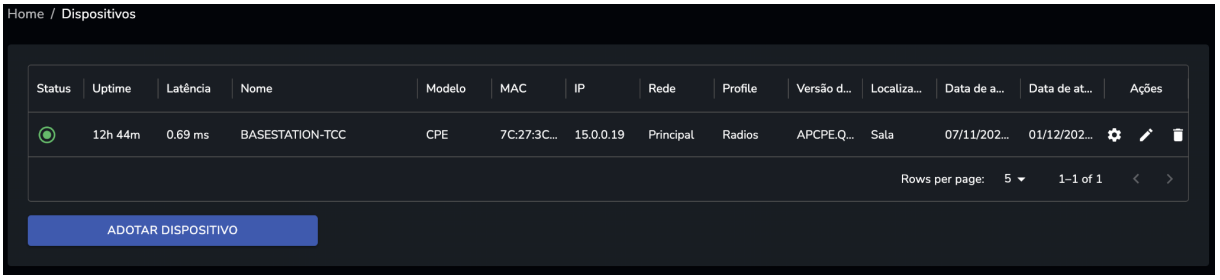
Figura 50 – Dados de adoção



Fonte: Autor

4. **Monitoramento Contínuo:** Após a adoção, o *MonitorController* iniciou o monitoramento ativo e periódico da *basestation*. Utilizando o *DeviceDriver* e o *Profile*. O *Backend* executou as ações de monitoramento em loops, coletando dados de modelo, nome, *uptime* e versão de firmware. Bem como disponibilizando seu *status* e latência, conforme a Figura 51.

Figura 51 – Validação monitoramento individual



Fonte: Autor

Além disso, foi possível verificar a topologia da rede Principal em tempo real, com o mapeamento correto da CPE conectada, bem como as métricas gerais da rede, destacado pela Figura 52.

Figura 52 – Validação monitoramento rede

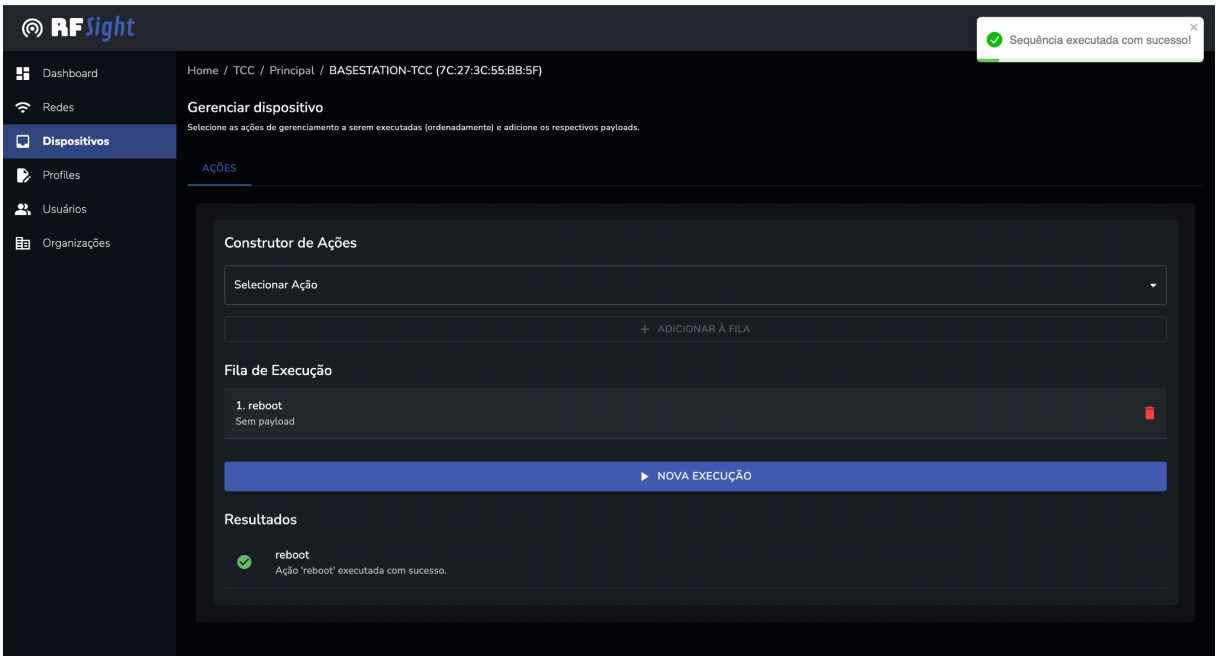
Fonte: Autor

A Figura 52 apresenta a topologia do cenário de validação do RFSight, com seu throughput agregado e por enlaces, contagem de dispositivos adotados *online/offline* e estações conectadas, tendo essas métricas todas afetando diretamente o cálculo da saúde da rede. O principal fator redutor da saúde da rede

no momento das validações foi o baixo *throughput* agregado da rede, devido ao baixo tráfego no enlace (apenas conectado).

5. **Gerenciamento Ativo:** Por fim, foram testadas as execuções de duas ações de gerenciamento (RF03) de forma separada, devido ao comportamento do dispositivo de ficar indisponível automaticamente durante um certo período após aplicar uma configuração. Primeiramente, através da *interface*, foi disparada a ação *reboot* para a *basestation*. O *DeviceDriver* executou a requisição HTTP POST correspondente, e o dispositivo foi reiniciado com sucesso, conforme é possível verificar pela sequência de Figuras de 53 a 55.

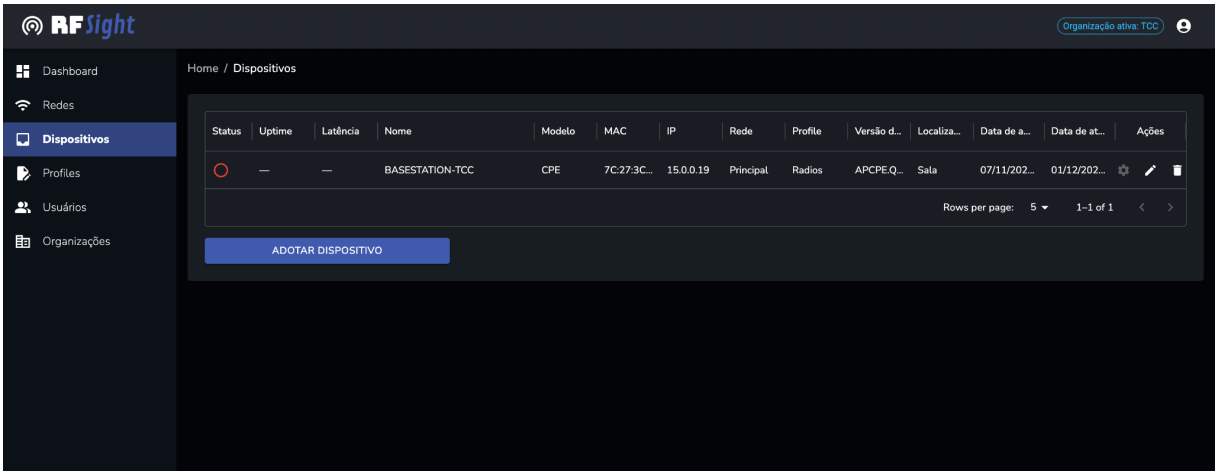
Figura 53 – Execução da ação de *reboot*



Fonte: Autor

A Figura 53 demonstra claramente a execução da ação de *reboot* no dispositivo alvo, bem como seu *status* de sucesso. Para validar realmente o seu funcionamento, é necessário analisar o *status* do dispositivo na página de dispositivos ou *dashboard*. A Figura 50 destaca o *status* do dispositivo logo após a execução do *reboot* com sucesso (dispositivo devidamente *offline*).

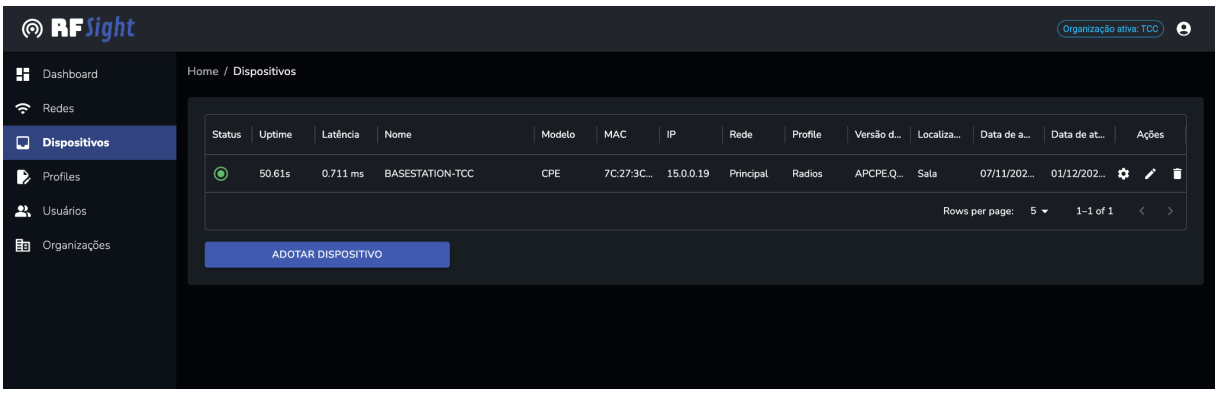
Figura 54 – Dispositivo *offline* após *reboot*



Fonte: Autor

Como o dispositivo recebeu um comando de reinicialização, o comportamento esperado é que após ficar *offline*, ao aguardar por volta de 1 minuto, o dispositivo deve aparecer *online* novamente no sistema, e é isso que demonstra a Figura 55.

Figura 55 – Dispositivo *online* novamente

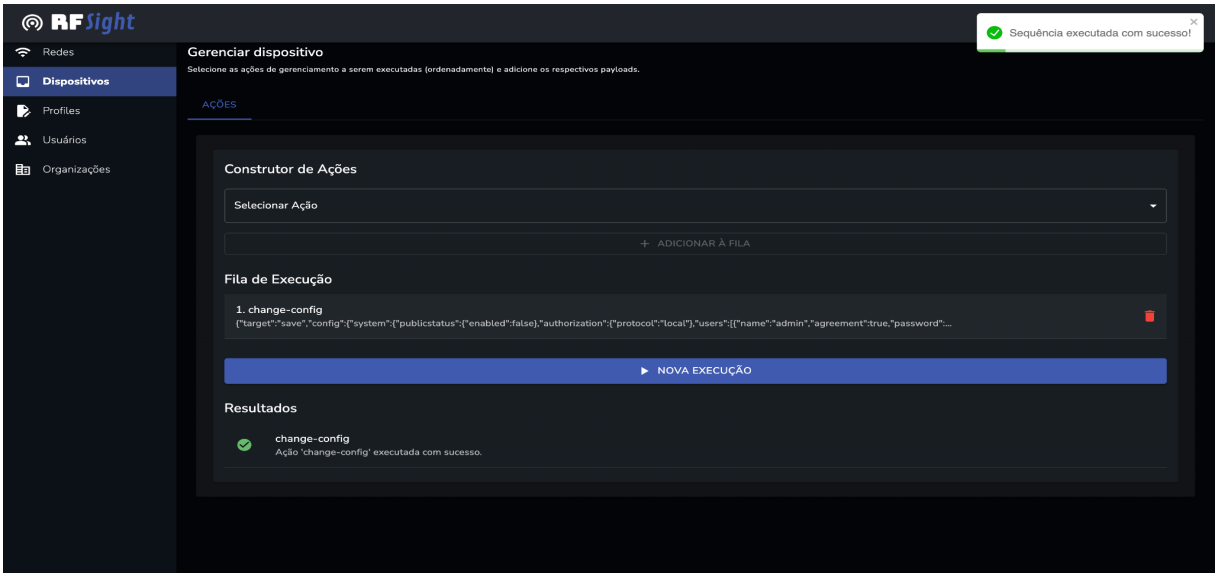


Fonte: Autor

Ao analisarmos a Figura 55, é possível perceber que o dispositivo realmente foi reinicializado, pois seu *Uptime* está marcando em torno de 50 segundos.

A segunda ação realizada foi a *change-config*, com a finalidade de alterar o hostname do dispositivo de “BASESTATION-TCC” para “BASESTATION-VALID”, por meio do envio do *payload* JSON requerido por esta ação. As Figuras 56 e 57 demonstram a execução da configuração.

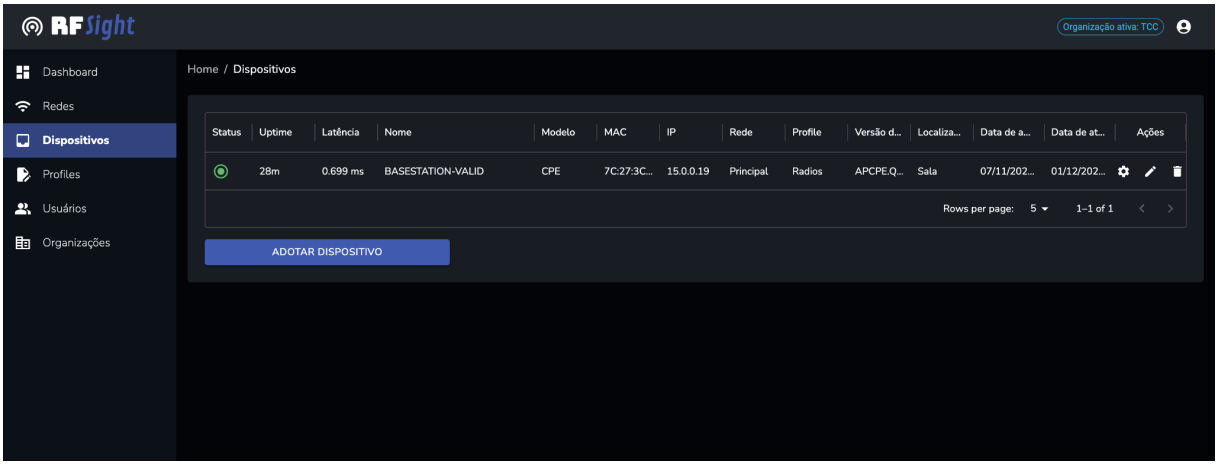
Figura 56 – Execução da ação de *change-config*



Fonte: Autor

A Figura 52 apresenta a correta execução da ação de *change-config* com o envio do payload JSON para a alteração do *hostname*. Alteração posteriormente confirmada pelas informações destacadas pela Figura 57.

Figura 57 – Configuração atualizada

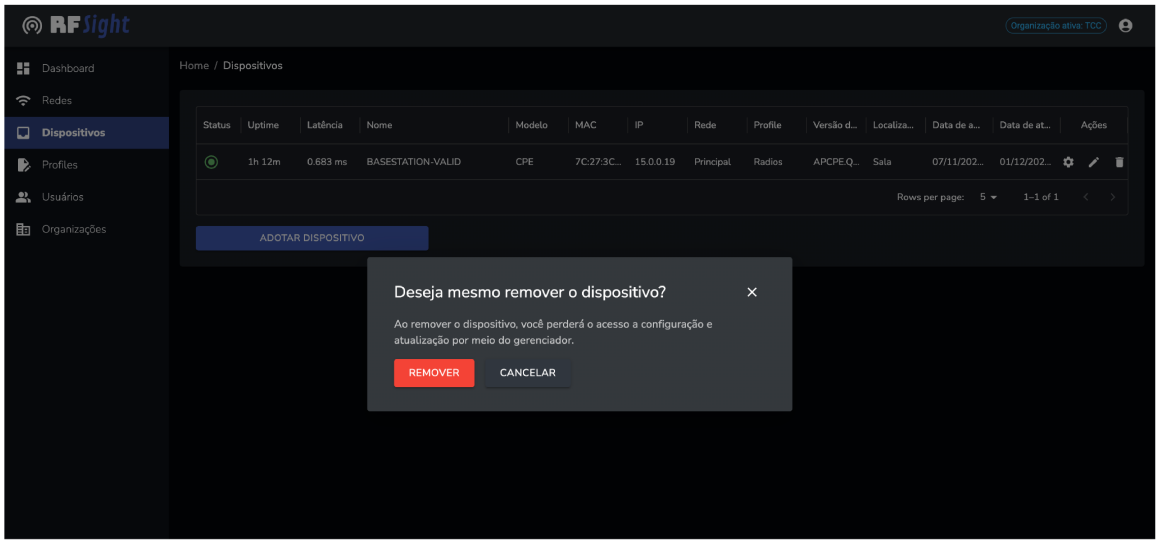


Fonte: Autor

Portanto, conforme apresentado pela Figura 57, o dispositivo teve seu Nome (*hostname*) devidamente alterado para o esperado durante a execução da ação de gerenciamento.

6. **Remoção de dispositivo:** Ao final dos testes, foi executada a ação de remoção do dispositivo adotado, a fim de validar a integridade do sistema durante essa rotina. As Figuras 58 e 59 apresentam a execução da remoção do dispositivo a partir da tela de dispositivos.

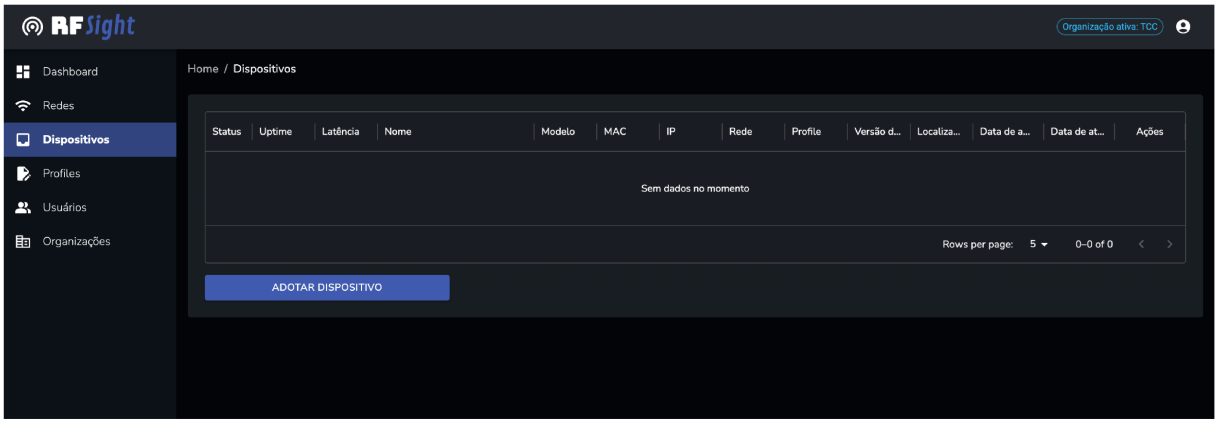
Figura 58 – Confirmar remoção de dispositivo



Fonte: Autor

A Figura 58 destaca a necessidade de confirmar a ação destrutiva de remover o dispositivo, onde o sistema adverte das consequências da ação selecionada. Ao prosseguir, temos a completa remoção do dispositivo do sistema, não fazendo mais parte de qualquer execução de monitoramento ou gerenciamento, passa a ser um dispositivo desconhecido, conforme apresentado pela Figura 59.

Figura 59 – Dispositivo removido



Fonte: Autor

Por fim, como é possível analisar na Figura 59, após a remoção do dispositivo, não há mais nenhum tipo de monitoramento ou gerenciamento disponível para o mesmo, nem sequer consta nas listas do RFSight.

6.3. Resultados

Os testes realizados no ambiente controlado permitiram a validação integral dos requisitos funcionais estabelecidos para o RFSight. A execução dos cenários de teste demonstrou que o protótipo não apenas atende às necessidades de gerenciamento e monitoramento, mas também valida as decisões de arquitetura relacionadas à segurança e usabilidade.

A validação inicial confirmou o atendimento aos requisitos de acesso e estrutura (RF08, RF09 e RF07). O sistema mostrou-se acessível via navegador *web* (RF09), com o fluxo de autenticação (RF08) operando corretamente através da emissão e validação de *tokens* JWT, garantindo que apenas usuários autorizados tivessem acesso às funcionalidades. A criação de Organizações e Redes validou a estrutura hierárquica proposta (RF07), servindo como contêiner lógico obrigatório para o agrupamento dos equipamentos.

No que tange ao ciclo de vida dos dispositivos, o processo de Adoção (RF01) provou ser robusto. O *AdoptionController* identificou corretamente os dispositivos na rede local via varredura ARP e, mediante a aplicação do perfil "Radios", instanciou a comunicação com o *hardware white label*. O sucesso na extração automática de metadados (modelo e *firmware*) confirmou a eficácia da camada de abstração. Além disso, o teste de Remoção (RF02) foi validado com êxito, garantindo que, ao excluir um dispositivo, todos os seus vínculos de monitoramento e persistência fossem encerrados corretamente, mantendo a integridade do banco de dados.

Quanto ao Monitoramento em Tempo Real (RF05 e RF06), a arquitetura baseada em WebSockets garantiu a fluidez na atualização dos dados na *interface* gráfica. A validação evidenciou que o algoritmo híbrido de descoberta de topologia foi capaz de mapear corretamente a hierarquia da rede P2MP simulada (RF06), identificando a *basestation* e a CPE conectada. As métricas de desempenho individuais (RF05), incluindo latência, *uptime* e *throughput*, foram coletadas e processadas corretamente, refletindo o estado real da rede e impactando coerentemente o cálculo do índice de "Saúde da Rede".

Por fim, os testes de Gerenciamento Ativo (RF03) validaram a flexibilidade do sistema em orquestrar comandos complexos. A execução bem-sucedida das ações de *reboot* (via protocolo SSH) e de alteração de *hostname* (via requisição HTTP POST) comprovou que o *ConfigController* gerencia adequadamente a tradução de intenções genéricas do usuário para instruções proprietárias do dispositivo. A detecção automática da indisponibilidade do dispositivo durante a reinicialização e o subsequente retorno ao estado *online* com as novas configurações aplicadas fecham o ciclo de validação.

Em suma, os resultados obtidos demonstram que o RFSight é uma solução funcional e coesa, capaz de cobrir todo o espectro de operações essenciais para um provedor regional, desde a segurança no acesso até o controle granular dos ativos de rede.

7. ANÁLISE COMPARATIVA

Com o protótipo desenvolvido, é possível ampliar a análise das ferramentas existentes e dos trabalhos acadêmicos correlatos, a fim de localizar a solução desenvolvida em meio ao estado da arte. A análise será conduzida em duas frentes: primeiramente, comparando o protótipo com as soluções de mercado para evidenciar seus diferenciais práticos e de negócio; em seguida, contrastando o RFSight com os desafios e contextos apontados pela literatura, para demonstrar sua relevância estratégica.

7.1. Comparativo com Soluções de Mercado

As soluções de mercado analisadas na seção 3.1, embora consolidadas, apresentam barreiras significativas para os provedores regionais. Ferramentas como o UNMS/UISP (Ubiquiti) e o The Dude (Mikrotik) promovem um forte aprisionamento tecnológico (*vendor lock-in*), limitando a flexibilidade do provedor. Em contrapartida, plataformas como o Zabbix, embora abertas e poderosas, impõem uma alta complexidade de implantação e manutenção. A Tabela 3 apresenta o comparativo do RFSight com essas duas abordagens.

Tabela 3 – Comparativo com Soluções de Mercado

Característica	UNMS/UISP & The Dude	Zabbix	RFSight
Modelo de Negócio	Ecossistema fechado, atrelado à venda de *hardware* específico.	*Open-source*, com custos indiretos de consultoria, treinamento e manutenção.	*Open-source*, focado em baixo custo de customização, implantação e operação.
Flexibilidade (Multi-vendor)	Inexistente ou muito limitada.	Alta, porém requer customização complexa via desenvolvimento de *templates*.	Alta por concepção, utilizando uma camada de abstração para facilitar a integração.

Característica	UNMS/UISP & The Dude	Zabbix	RFSight
Público-alvo	Usuários assíduos da marca.	Corporações com equipes de TI especializadas e de grande porte.	Provedores de Internet de pequeno e médio porte (PPPs) com foco em radiofrequência.
Curva de Aprendizado	Baixa a moderada.	Alta, exigindo conhecimento técnico aprofundado em redes e sistemas.	Baixa, com foco em usabilidade e funcionalidades direcionadas ao problema.

Fonte: Autor

A Tabela 3 demonstra que o RFSight se posiciona em um nicho sem foco específico. Ele busca combinar a flexibilidade de uma plataforma aberta com a simplicidade de uma ferramenta especializada, evitando tanto a rigidez do *vendor lock-in* quanto a barreira de complexidade das soluções de propósito geral.

7.2. Posicionamento Frente aos Trabalhos Correlatos

A segunda análise posiciona o RFSight em relação aos problemas e contextos identificados na literatura. A Tabela 4 mostra como a solução desenvolvida neste trabalho responde diretamente às lacunas e necessidades apontadas pelos estudos da seção 3.2.

Tabela 4 – Lacunas Identificadas e Contribuição

Tópico (Trabalhos Correlatos)	Lacuna ou Contexto Identificado	Contribuição do RFSight
Gerência de Redes (Campos, 2018)	Ferramentas genéricas são eficientes, mas consideravelmente complexas e não especializadas para o cenário de redes de RF.	Oferece uma solução especializada, com foco em métricas e gerência específicas para o cenário de radiofrequência, e com menor curva de aprendizado.

Conectividade Rural (Bertollo, 2021)	Crescente demanda por conectividade confiável no agronegócio, suprida principalmente por ISPs regionais.	Age como uma ferramenta de apoio para os ISPs, permitindo-lhes melhorar a qualidade e a confiabilidade do serviço que viabiliza a inovação no campo.
Tecnologias Futuras (Santana et al., 2023; Farias, 2020)	O 5G enfrenta barreiras substanciais de custo e infraestrutura que postergam sua massificação em áreas rurais.	Otimiza a gestão da tecnologia de radiofrequência, que é uma solução de conectividade economicamente viável no presente e médio prazo para essas regiões.

Fonte: Autor

Esta segunda análise aprofunda a relevância do trabalho. Fica evidente que o RFSight não é apenas uma alternativa a ferramentas existentes, mas uma solução pragmática a um contexto socioeconômico e tecnológico bem definido. Por um lado, Campos (2018) apresenta a eficiência e os resultados positivos significativos da aplicação do gerenciamento de redes por meio de ferramentas centralizadoras. Enquanto os estudos de Santana et al. (2023) e Farias (2021) demonstram que a conectividade via 5G não será uma realidade ubíqua no campo a curto prazo, o trabalho de Bertollo (2021) mostra que a necessidade por essa conectividade é imediata.

Portanto, a dupla análise comparativa consolida a importância do protótipo desenvolvido. O RFSight se destaca por sua proposta de valor técnico frente às soluções de mercado e por sua pertinência estratégica diante do cenário de conectividade brasileiro. O sistema foi projetado para capacitar os provedores que são, na prática, os responsáveis pela inclusão digital em vastas porções do território, oferecendo uma ferramenta alinhada às suas capacidades técnicas e financeiras para otimizar a infraestrutura que já operam.

8. CONCLUSÃO

Este trabalho de conclusão de curso teve como objetivo o desenvolvimento de um protótipo de sistema de gerenciamento e monitoramento de dispositivos para provedores de Internet via radiofrequência, nomeado RFSight. A principal motivação foi a lacuna identificada no mercado: a carência de ferramentas que sejam simultaneamente flexíveis (independentes de fabricante), acessíveis (baixo custo e baixa curva de aprendizado) e especializadas para as necessidades de ISPs regionais com foco em radiofrequência.

O protótipo RFSight foi projetado para preencher essa lacuna através de uma arquitetura modular, cuja peça central é a Camada de Abstração de Dispositivo, implementada através de um sistema de "Perfis" (*Profile*). Cada perfil define, de forma declarativa, como o RFSight deve se comunicar com um dispositivo específico, seja via API HTTP ou SSH.

Avaliando os objetivos propostos:

O **Objetivo Geral** (desenvolver um protótipo de sistema de gerenciamento e monitoramento de redes radiofrequência ponto a ponto (P2P) e ponto multiponto (P2MP), com foco em ISPs regionais, oferecendo um sistema modular, personalizável, de baixo custo e independente de fabricantes) foi alcançado. O *Backend* do protótipo implementa com sucesso a arquitetura modular e a independência de fabricantes através dos *Profiles* e do *DeviceDriver*. Os testes (Capítulo 6) validaram que o sistema pode, de fato, adotar, monitorar em tempo real e gerenciar dispositivos *white label* baseando-se apenas em perfis configurados.

Os **Objetivos Específicos** foram integralmente atingidos:

1. O estado da arte foi identificado e posteriormente utilizado para fins comparativos com a solução completa (Capítulos 3 e 7).
2. A solução conceitual foi projetada e consolidada em uma definição de arquitetura sólida (Capítulo 4, Figura 15).

3. Os protocolos de gerenciamento foram estudados e aplicados na solução (Capítulo 2.4).
4. O *Backend* em Python/FastAPI e a camada de abstração foram desenvolvidos com sucesso (Capítulo 5).
5. O *Frontend* em React foi construído, totalmente responsivo e integrado ao *Backend* com o foco na usabilidade e jornada do usuário desde a prototipação de telas (Capítulo 5).
6. A Camada de Abstração de Dispositivo foi desenvolvida e se provou robusta no funcionamento do protótipo (Capítulo 5).
7. Testes e validações em ambiente simulado foram realizados e provaram a funcionalidade do RFSight (Capítulo 6).

O RFSight apresenta-se, portanto, como uma ferramenta estratégica para a profissionalização de pequenos e médios provedores. Ao reduzir a complexidade técnica da gestão multimarca e oferecer uma *interface* unificada, o sistema contribui diretamente para a eficiência operacional desses provedores, essenciais para a expansão da conectividade em áreas rurais do Brasil. Como trabalhos futuros, sugere-se a expansão para protocolos como TR-069 e NETCONF, bem como operação do sistema em nuvem, alinhado à expansão das possibilidades de configurações de ações e *payloads* dentro do contexto da Camada de Abstração de Dispositivo, como a execução em massa, por exemplo. Além das expansões propostas, sugere-se a validação prática da funcionalidade de atualização de *firmware* individual. Embora a arquitetura do RFSight já suporte o envio de *payloads* de arquivo (tratados em *base64)*, testes em campo com dispositivos que possuam imagens de *firmware* disponíveis são necessários para confirmar a integridade e a segurança deste fluxo crítico de gerenciamento.

REFERÊNCIAS

STALLINGS, William. **Data and Computer Communications**. 10. ed. Harlow, Essex: Pearson Education Limited, 2014. ISBN 978-1-29-201438-8. Documento eletrônico. Disponível em: <https://archive.org/details/data-and-computer-communication-stallings-10e>. Acesso: 16 nov. 2024.

TANENBAUM, Andrew S.; WETHERALL, David J. **Computer Networks**. 5. ed. Upper Saddle River, NJ: Prentice Hall, 2011. ISBN 978-0-13-212695-3. Documento eletrônico. Disponível em: <https://www.inf.ufsc.br/~bosco.sobral/ensino/ine5645/Computer-Networks---A-Tanenbaum---5th-edition.pdf>. Acesso em: 16 nov. 2024.

KUROSE, James F.; ROSS, Keith W. **Computer Networking: A Top-Down Approach**. 7. ed. Hoboken, NJ: Pearson, 2017. ISBN 978-0-13-359414-0. Documento eletrônico. Disponível em: <https://archive.org/details/kurose-redes-de-computadores-e-a-Internet-uma-abordagem-top-down-6a>. Acesso em: 16 nov. 2024.

KHAN, Rida et al. **Long Range Wireless Point to Point Link Network on 5 GHz Frequency Band with VoIP**. ResearchGate. Disponível em: <https://www.researchgate.net/publication/311855298_Long_Range_Wireless_Point_to_Point_Link_Network_on_5_GHz_Frequency_Band_with_VoIP>. Acesso em: 16 nov. 2024.

CNA, Confederação da Agricultura e Pecuária do Brasil; ABIMAQ, Associação Brasileira da Indústria de Máquinas e Equipamentos. **Soluções para conectividade rural**, 2023, Brasília. Disponível em: <https://www.cnabrasil.org.br/noticias/cna-propoe-grande-projeto-de-estado-para-ampliar-conectividade>. Acesso em: 16 nov. 2024.

AGRIHUB. **Conectividade no Campo: Desafios e oportunidades para o agronegócio brasileiro**, 2024. Disponível em: <https://agrihub.com.br/conectividade-no-campo-desafios-e-oportunidades-para-o-agronegocio-brasileiro>. Acesso em: 17 mai. 2025.

CONECTARAGRO. **Indicador de Conectividade Rural – ICR**. Disponível em: <https://www.conectaragro.com.br/indicadordeconectividaderural>. Acesso em: 17 mai. 2025.

BRASIL, Ministério da Agricultura e Pecuária. **Internet no campo**. Disponível em: <https://www.gov.br/agricultura/pt-br/assuntos/inovacao/agrohub-brasil/produtores-rurais/Internet-no-campo-1/Internet-no-campo>. Acesso em: 17 mai. 2025.

CONECTARAGRO. **4 benefícios da Internet rural para sua propriedade**, 2024. Disponível em: <https://blog.conectaragro.com.br/4-beneficios-da-Internet-rural-para-sua-propriedade>. Acesso em: 17 mai. 2025.

BERTOLLO, Mait. **A Internet das Coisas (IoT) no campo brasileiro: as redes informacionais, as novas dinâmicas da produção agrícola e os provedores regionais**. XIV Encontro Nacional de Pós Graduação e Pesquisa em Geografia, 2021. Disponível em: <https://www.editorarealize.com.br/editora/anais/enanpege/2021/TRABALHO_COMPLETO_EV154_MD1_SA138_ID147212112021132134.pdf>. Acesso em: 17 mai. 2025.

AGÊNCIA NACIONAL DE TELECOMUNICAÇÕES (ANATEL). **Paineis de outorga e licenciamento**. Disponível em: <https://informacoes.anatel.gov.br/paineis/outorga-e-licenciamento>. Acesso em: 17 mai. 2025.

POSSEBON, Tainã Vieira. **Provedor de Internet Via Radiofrequência**. Trabalho de Conclusão de Curso (Graduação em Redes de Computadores) – Universidade Federal de Santa Maria, Santa Maria, 2014. Disponível em: <https://www.ufsm.br/app/uploads/sites/495/2018/12/TCC_Taina.pdf>. Acesso em: 17 mai. 2025.

INTELBRAS. **Glossário para provedores de Internet: o que significam os principais termos usados por ISPs**. 2021. Disponível em: <https://blog.intelbras.com.br/glossario-para-provedores/>. Acesso em: 17 mai. 2025.

TELESÍNTESE. **ISPs concentram investimentos em banda larga no Brasil, mas não a receita**. Disponível em: <https://www.telesintese.com.br/isps-concentram-investimentos-em-banda-larga-no-brasil-mas-nao-a-receita>. Acesso em: 19 mai. 2025.

HAPPYNET. **Point-to-Multipoint Wireless**. Disponível em: <http://happynet.in/p2mp.html>. Acesso em: 20 mai. 2025.

TP-LINK. **What is the difference between wireless standards 802.11a/b/g/n/ac/ax? (WI-FI 4, WI-FI 5, WI-FI 6)**, 2024. Disponível em: <https://www.tp-link.com/in/support/faq/1226/>. Acesso em: 20 mai. 2025.

NGUYEN, Augustine. **Comprehensive Guide to External Antennas**. Disponível em: <https://www.mouser.mx/blog/comprehensive-guide-external-antennas>. Acesso em: 21 maio 2025.

WIFI BOOSTER. **Horizontal Sector Antenna Calculator for Wi-Fi Signal Coverage**. Disponível em: <https://blog.wifibooster.eu/horizontal-sector-antenna-calculator-for-wifi-signal-coverage/>. Acesso em: 21 maio 2025.

CASE, J. et al. **Simple Network Management Protocol (SNMP). RFC 1157**, maio 1990. Disponível em: <https://datatracker.ietf.org/doc/html/rfc1157>. Acesso em: 17 mai. 2025.

GEEKSFORGEEKS. **Simple Network Management Protocol (SNMP)**, 2024. Disponível em: <https://www.geeksforgeeks.org/simple-network-management-protocol-snmp/>. Acesso em: 20 mai. 2025.

IEEE. **802.1AB-2005 - IEEE Standard for Local and Metropolitan Area Networks - Station and Media Access Control Connectivity Discovery**. 2005. Disponível em: <https://standards.ieee.org/standard/802_1AB-2005.html>. Acesso em: 17 mai. 2025.

YLONEN, T.; LONVICK, C. **The Secure Shell (SSH) Protocol Architecture. RFC 4251**, jan. 2006. Disponível em: <https://datatracker.ietf.org/doc/html/rfc4251>. Acesso em: 17 mai. 2025.

FIELDING, Roy Thomas. **Architectural Styles and the Design of Network-based Software Architectures**. University of California, Irvine, 2000. Disponível em: <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>. Acesso em: 17 mai. 2025.

DREAMFACTORY. **REST vs GraphQL: Which API Design Style is Right For Your Organization**. Disponível em: <https://blog.dreamfactory.com/rest-vs-graphql-which-api-design-style-is-right-for-your-organization>. Acesso em: 21 mai. 2025.

POSTEL, J. **Internet Control Message Protocol. RFC 792**, set. 1981. Disponível em: <https://datatracker.ietf.org/doc/html/rfc792>. Acesso em: 17 mai. 2025.

PLUMMER, D. **Address Resolution Protocol. RFC 826**, nov. 1982. Disponível em: <https://datatracker.ietf.org/doc/html/rfc826>. Acesso em: 17 mai. 2025.

FAHEY, Ian; FETTERMAN, Gabriel. ***The WebSocket Protocol*. RFC 6455, Internet Engineering Task Force**, dez. 2011. Disponível em: <https://datatracker.ietf.org/doc/html/rfc6455>. Acesso em: 21 mai. 2025.

BESCHOKOV, Mukhadin. **WebSocket vs HTTP: How Are These 2 Different?**, 2025. Disponível em: <https://www.wallarm.com/what/websocket-vs-http-how-are-these-2-different>. Acesso em: 21 mai 2025.

UBIQUITI. **UISP Overview**. Disponível em: <https://uisp.com/uisp-overview>. Acesso em: 21 mai. 2025.

MIKROTIK. **The Dude**. Disponível em: <https://mikrotik.com/thedude>. Acesso em: 21 mai. 2025.

ZABBIX. **Network Monitoring**. Disponível em: <https://www.zabbix.com/network_monitoring>. Acesso em: 21 mai. 2025.

CAMPOS, Elaine Aparecida. **Estudo de caso de ferramentas para gerenciamento de redes de computadores**, 2018.Trabalho de Conclusão de Curso (Graduação em Tecnologia em Redes de Computadores) - Instituto Federal de Santa Catarina, São José, 2018. Disponível em: <https://repositorio.ifsc.edu.br/bitstream/handle/123456789/345/TCC_ElaineCampos.pdf?sequence=1&isAllowed=y>. Acesso em: 01 jul. 2025.

SANTANA, Daniele de et al. **Tecnologia 5G e suas potencialidades de utilização no agronegócio**, 2023. Revista Observatorio de la Economía Latinoamericana, Curitiba, v. 21, n. 10, p. 14522-14541, 2023. Disponível em: <https://ojs.observatoriolatinoamericano.com/ojs/index.php/olel/article/view/1670/1065>. Acesso em: 01 jul. 2025.

FARIAS, Rafael Brito de. **Análise da necessidade de mudança regulatória para a implantação do 5G no Brasil**. 2021. 128 f. Dissertação (Mestrado em Gestão Empresarial) - Escola Brasileira de Administração Pública e de Empresas, Fundação Getulio Vargas, Rio de Janeiro, 2021. Disponível em: <https://repositorio.fgv.br/server/api/core/bitstreams/4b9121bc-147d-4313-a1fc-34d2c01424b8/content>. Acesso em: 01 jul. 2025.

PYTHON SOFTWARE FOUNDATION. **Python**. [S.l.], 2025. Disponível em: <https://www.python.org/>. Acesso em: 01 dez. 2025.

DOCKER, INC. **Docker**: accelerate how you build, share, and run applications. [S.l.], 2025. Disponível em: <https://www.docker.com/>. Acesso em: 01 dez. 2025.

MONGODB, INC. **MongoDB**: the developer data platform. [S.l.], 2025. Disponível em: <https://www.mongodb.com/>. Acesso em: 01 dez. 2025.

META PLATFORMS, INC. **React**: the library for web and native user interfaces. [S.l.], 2025. Disponível em: <https://react.dev/>. Acesso em: 01 dez. 2025.

REDUX TOOLKIT. **Redux Toolkit**: the official, opinionated, batteries-included toolset for efficient Redux development. [S.l.], 2025. Disponível em: <https://redux-toolkit.js.org/>. Acesso em: 01 dez. 2025.

GOOGLE. **Material Design 3**: Style Guide. [S.l.], 2025. Disponível em: <https://m3.material.io/foundations/content-design/style-guide/ux-writing-best-practices>. Acesso em: 01 dez. 2025.

MUI. **MUI**: the React component library you always wanted. [S.l.], 2025. Disponível em: <https://mui.com/>. Acesso em: 01 dez. 2025.

XYFLOW. **React Flow**: node-based UIs in React. [S.l.], 2025. Disponível em: https://reactflow.dev/. Acesso em: 01 dez. 2025.

WOOD, Robert. **smtp4dev**. [S.l.]: GitHub, 2025. Disponível em: <https://github.com/rnwood/smtp4dev>. Acesso em: 01 dez. 2025.

CURNOW, Jamie. **Nginx Proxy Manager**. [S.l.], 2025. Disponível em: <https://nginxproxymanager.com/>. Acesso em: 01 dez. 2025.

INTERNET SECURITY RESEARCH GROUP (ISRG). **Let's Encrypt**: uma autoridade de certificação gratuita, automatizada e aberta. [S.l.], 2025. Disponível em: <https://letsencrypt.org/pt-br/>. Acesso em: 01 dez. 2025.

DUCK DNS. **Duck DNS**: free dynamic DNS. [S.l.], 2025. Disponível em: <https://www.duckdns.org/>. Acesso em: 01 dez. 2025.

APÊNDICE A – Formulário de Levantamento de Requisitos

Formulário de Levantamento de Requisitos para Sistema de Gerenciamento de Redes Radiofrequência (RFSight)

Prezado(a) profissional,

Este formulário faz parte de uma pesquisa acadêmica para o desenvolvimento de um Trabalho de Conclusão de Curso em Sistemas de Informação na Universidade Federal de Santa Catarina (UFSC).

O objetivo deste trabalho é desenvolver um **protótipo de sistema de gerenciamento para redes de radiofrequência**, nomeado **RFSight**, voltado especialmente para as necessidades de Provedores de Internet (ISPs).

Sua experiência e conhecimento prático são fundamentais para validarmos as funcionalidades mais importantes e garantirmos que a solução proposta seja relevante para os desafios reais do setor. As respostas são confidenciais e serão usadas exclusivamente para fins acadêmicos.

Tempo estimado para preenchimento: 10-15 minutos.

Agradeço imensamente sua colaboração!

* Indica uma pergunta obrigatória

Perfil e Contexto Profissional

1. Qual sua principal área de atuação? *

   Marcar apenas uma oval.

   - ◯ Gestão de Redes / Infraestrutura
   - ◯ Desenvolvimento de Produtos para Telecom
   - ◯ Gestão de Negócios / Estratégia em ISP
   - ◯ Engenharia de Redes
   - ◯ Outro: _____

2. Quantos anos tem de experiência na área? *

   _____

3. Quais ferramentas você tem conhecimento ou utiliza hoje para o gerenciamento/monitoramento de redes de radiofrequência? (Possível marcar mais de uma)

Marque todas que se aplicam.

- [ ] UNMS / UISP (Ubiquiti)
- [ ] The Dude (Mikrotik)
- [ ] Zabbix
- [ ] Grafana
- [ ] Scripts próprios / Desenvolvimento interno
- [ ] Planilhas / Controle manual
- [ ] Outro: _____

4. [OPCIONAL] Em sua opinião, qual é a maior dificuldade ou funcionalidade que falta nas ferramentas selecionadas?

_____

_____

_____

_____

_____

Validação de Funcionalidades e Métricas

5. Quanto às funcionalidades essenciais do protótipo: *

Para cada funcionalidade listada a seguir, indique o grau de importância para sua operação, considerando a seguinte escala:

1 - Não é necessária
2 - Pouco relevante
3 - Útil
4 - Importante
5 - Essencial

Marcar apenas uma oval por linha.

	1	2	3	4	5
Adotar dispositivos	◯	◯	◯	◯	◯
Remover dispositivos adotados	◯	◯	◯	◯	◯
Configurar individualmente cada dispositivo adotado	◯	◯	◯	◯	◯
Configurar dispositivos em massa	◯	◯	◯	◯	◯
Atualizar dispositivos de forma individual	◯	◯	◯	◯	◯
Atualizar dispositivos em massa	◯	◯	◯	◯	◯
Monitorar status individual dos dispositivos	◯	◯	◯	◯	◯
Monitorar status agregado da rede	◯	◯	◯	◯	◯
Visualizar gráficos de desempenho da rede	◯	◯	◯	◯	◯
Gerência de usuários do sistema (cadastro, edição e exclusão)	◯	◯	◯	◯	◯
Acesso com login por nome de usuário/senha	◯	◯	◯	◯	◯

6. Quanto aos dados de monitoramento dos dispositivos: *

Para cada item listado a seguir, indique o grau de importância para sua operação, considerando a seguinte escala:

1 - Não é necessário
2 - Pouco relevante
3 - Útil
4 - Importante
5 - Essencial

Marcar apenas uma oval por linha.

	1	2	3	4	5
Status (Online/Offline)	◯	◯	◯	◯	◯
Uptime (Tempo de atividade)	◯	◯	◯	◯	◯
Hostname	◯	◯	◯	◯	◯
Endereços IP e MAC	◯	◯	◯	◯	◯
Versão de firmware	◯	◯	◯	◯	◯
Latência	◯	◯	◯	◯	◯
Localização	◯	◯	◯	◯	◯

7. Quanto às métricas e organização de um dashboard de monitoramento: *

Para cada item listado a seguir, indique o grau de importância para sua operação, considerando a seguinte escala:

1 - Não é necessário
2 - Pouco relevante
3 - Útil
4 - Importante
5 - Essencial

Marcar apenas uma oval por linha.

	1	2	3	4	5
Quantidade total de dispositivos Online / Offline	◯	◯	◯	◯	◯
Throughput (Download/Upload) agregado de toda a rede	◯	◯	◯	◯	◯
Throughput por enlace individual	◯	◯	◯	◯	◯
Um "Gráfico de Saúde da Rede" (tipo Gauge).	◯	◯	◯	◯	◯
Topologia da rede	◯	◯	◯	◯	◯

Comentários Finais

8. [OPCIONAL] Você tem alguma sugestão, crítica ou comentário adicional que possa contrib para o desenvolvimento do protótipo RFSight?

Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários

APÊNDICE B – Artigo sobre o TCC

RFSight: um protótipo de sistema de gerenciamento de dispositivos para provedores de Internet via radiofrequência

Bruno Petri[1], Carlos Becker Westphall[1], Wesley dos Reis Bezerra[2]

[1]Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC) Florianópolis – SC – Brasil

[2]Instituto Federal Catarinense (IFC) – Rio do Sul – SC – Brasil

{brunoopetri, carlosbwestphall, wesleybez}@gmail.com

Abstract. Rural connectivity is vital for agribusiness, yet regional ISPs struggle with heterogeneous networks and vendor lock-in. Existing tools are either proprietary or complex to manage. This paper presents RFSight, a prototype management system for RF networks built on microservices. Its core innovation is a hardware abstraction layer using declarative profiles, enabling multi-vendor device management without code changes. Simulated tests validated successful device adoption, real-time monitoring via WebSockets, and active management, proving RFSight as a viable, low-cost solution for overcoming lock-in.

Resumo. A conectividade rural é crucial para o agronegócio, mas Provedores Regionais (ISPs) enfrentam desafios com redes heterogêneas e aprisionamento tecnológico (*vendor lock-in*). As ferramentas atuais ou são proprietárias limitadas ou genéricas complexas. Este artigo apresenta o RFSight, um protótipo de gerência para redes de radiofrequência baseado em microsserviços. Sua inovação reside na camada de abstração de *hardware* via perfis declarativos, permitindo gerenciar dispositivos multimarcas sem refatoração. Testes simulados validaram a adoção, monitoramento em tempo real via *WebSockets* e gerenciamento ativo, confirmando o RFSight como solução viável e de baixo custo.

1. Introdução

A conectividade no meio rural brasileiro é um vetor estratégico para o crescimento econômico, impactando diretamente a produtividade do agronegócio e a inclusão digital de comunidades remotas. Estima-se que a ampliação da conectividade possa incrementar significativamente o Valor Bruto da Produção (VBP) agropecuária [CNA e ABIMAQ 2023]. Nesse cenário, os Provedores Regionais de Internet (ISPs) desempenham um papel protagonista, suprindo a demanda em áreas onde grandes operadoras não possuem viabilidade econômica para atuar com infraestrutura cabeada (fibra óptica).

Para atender a essas regiões, a tecnologia de radiofrequência (RF) apresenta-se como a solução mais viável, permitindo a implementação de enlaces Ponto a Ponto (P2P) e Ponto-Multiponto (P2MP) com custos reduzidos e rápida implantação [Stallings 2014]. No entanto, a operação dessas redes impõe desafios técnicos severos. ISPs regionais frequentemente operam infraestruturas heterogêneas, compostas por equipamentos de múltiplos fabricantes, visando otimização de custos.

Essa diversidade de *hardware* resulta em fragmentação da gestão e no problema de *vendor lock-in* (aprisionamento tecnológico). Ferramentas proprietárias, como o UNMS/UISP (Ubiquiti) ou The Dude (Mikrotik), são eficientes, mas gerenciam apenas dispositivos de suas respectivas marcas. Por outro lado, soluções genéricas de mercado, como o Zabbix, embora agnósticas, exigem alto nível de especialização técnica para configuração e manutenção, o que é muitas vezes inviável para as equipes enxutas dos pequenos provedores [Campos 2018].

Este trabalho propõe o RFSight, um protótipo de sistema de gerenciamento e monitoramento de dispositivos de redes radiofrequência, projetado para ser modular, de baixo custo e independente de fabricantes. A principal contribuição reside no desenvolvimento de uma Camada de Abstração de Dispositivo, que permite a interação com equipamentos de diferentes marcas através de perfis de configuração, sem necessidade de alteração no código-fonte do sistema.

2. Trabalhos Relacionados

A literatura e o mercado apresentam abordagens distintas para o gerenciamento de redes. Ferramentas comerciais proprietárias oferecem alta usabilidade e integração vertical, mas perpetuam a dependência do fornecedor. Em contraste, soluções *open-source* genéricas focam na flexibilidade.

Campos (2018) demonstra a eficiência de ferramentas de gerenciamento centralizado na detecção proativa de falhas, mas ressalta a complexidade de implementação de soluções como Zabbix em cenários com recursos humanos limitados. Bertollo (2021) destaca o papel dos provedores regionais como viabilizadores da Internet das Coisas (*IoT*) no campo, reforçando a necessidade de ferramentas que garantam a confiabilidade da rede sem elevar os custos operacionais.

Diferente das abordagens citadas, o RFSight posiciona-se na interseção entre a flexibilidade das ferramentas abertas e a especialização das ferramentas proprietárias, focando especificamente nas primitivas de redes de rádio (como tabelas de associação sem fio e topologia P2MP) através de uma arquitetura agnóstica.

3. RFSight: Arquitetura e Proposta

O RFSight foi desenvolvido sob uma arquitetura de microsserviços conteinerizada (*Docker*), garantindo portabilidade e escalabilidade. A solução é composta por três camadas principais: Interface de Usuário (*Frontend*), Lógica de Negócio (*Backend*) e Persistência (*Banco de Dados*).

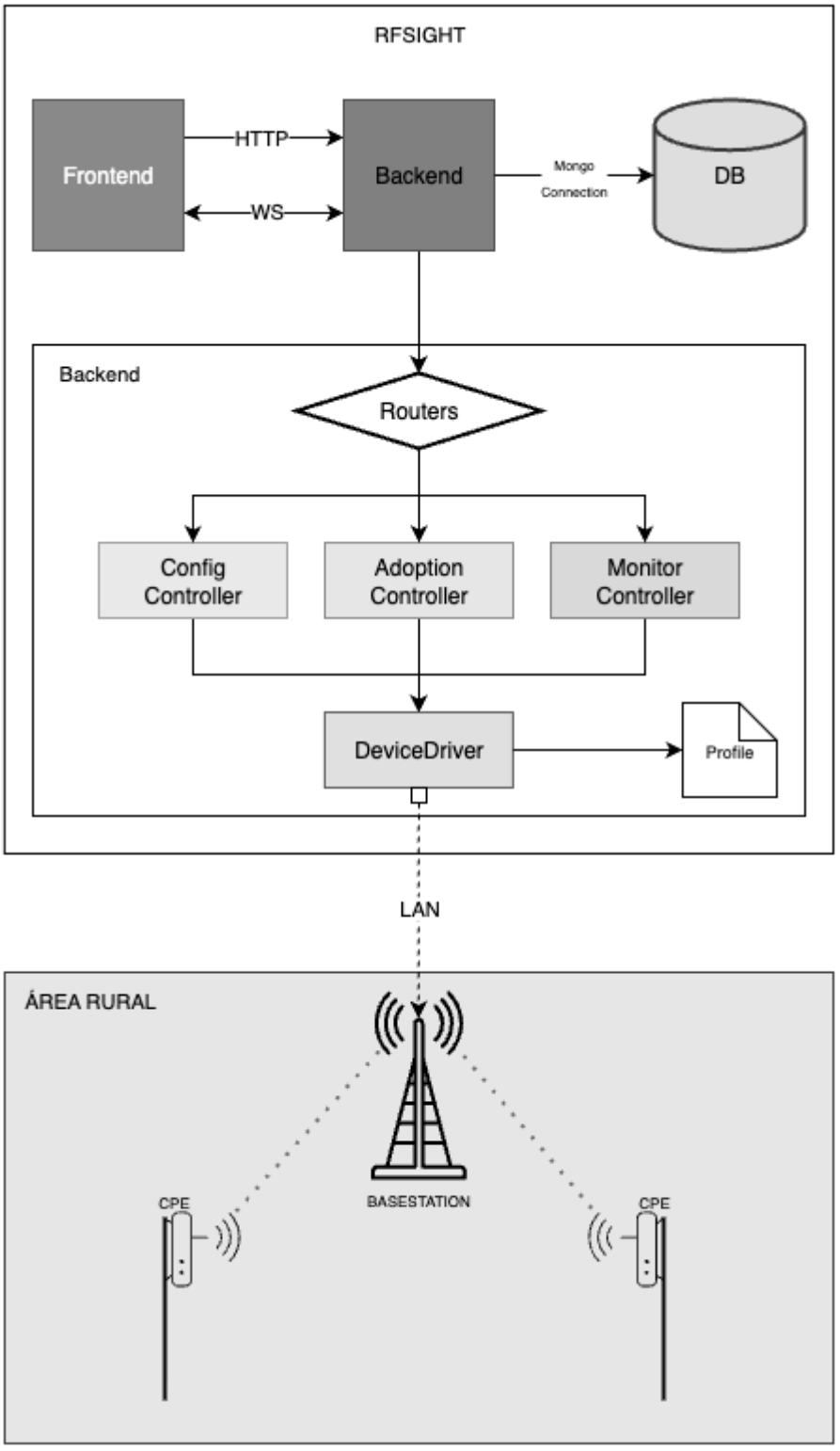


Figura 1. Arquitetura RFSight

3.1. Tecnologias e Comunicação

O *Backend* foi implementado em Python utilizando o *framework* FastAPI, escolhido pelo suporte nativo a programação assíncrona (*asyncio*), essencial para lidar com operações de I/O intensivas (como *ping* e requisições HTTP em massa) sem bloqueio. O *Frontend* utiliza React com Redux Toolkit, comunicando-se com o servidor via APIs REST para gerenciamento e WebSockets para telemetria em tempo real. A persistência de dados é realizada no MongoDB, um banco *NoSQL* orientado a documentos, adequado para a flexibilidade dos esquemas de dados de diferentes dispositivos.

3.2. Camada de Abstração de Dispositivo

O núcleo da inovação do RFSight é a sua capacidade de gerenciar dispositivos heterogêneos de forma transparente. Isso é alcançado através de dois componentes:

1. **Profile (Perfil):** Um modelo de dados (JSON) que atua como um "manual de instruções" para o sistema. Ele define a URL base da API do dispositivo, os OIDs SNMP para leitura de estações conectadas e mapeia as ações disponíveis (ex: autenticação, coleta de estatísticas, *reboot*). O perfil utiliza *placeholders* dinâmicos (como *{{DEVICE_IP}}* e *{{DEVICE_USER}}*) que são substituídos em tempo de execução.
2. **Device Driver:** Um serviço que atua como intérprete universal. Ele consome o *Profile*, hidrata as variáveis com os dados do dispositivo alvo e traduz intenções genéricas do sistema (ex: "Obter uso de CPU") em comandos específicos (requisições HTTP ou comandos SSH), normalizando a resposta para o formato interno do RFSight.

3.3. Monitoramento e Topologia Híbrida

O sistema implementa um algoritmo híbrido para descoberta automática da topologia de rede, combinando duas técnicas:

- **Camada 2 (Enlace):** Varreduras ARP para identificar a presença de dispositivos (IP/MAC) na rede local.
- **Camada 3 (Aplicação/Rede):** Consultas SNMP às *basestations* para recuperar a tabela de estações sem fio (*wireless stations*) conectadas.

O cruzamento desses dados permite ao RFSight desenhar o grafo da rede automaticamente, identificando relações de dependência (quem está conectado a quem) e calculando métricas de *throughput* e saúde da rede em tempo real.

4. Avaliação e Resultados

A validação do protótipo foi realizada em um ambiente controlado simulando a infraestrutura de um ISP, composto por dispositivos de rede *white label* configurados em topologia Ponto-Multiponto.

4.1. Cenário de Testes

Foram definidos requisitos funcionais críticos para validação: Adoção de Dispositivos (RF01), Gerenciamento Ativo (RF03) e Monitoramento em Tempo Real (RF05/RF06). Foi criado um *Profile* específico ("Radios") mapeando a API proprietária e a tabela SNMP dos dispositivos de teste.

4.2. Resultados Obtidos

Os testes demonstraram a eficácia da arquitetura proposta:

- **Adoção e Abstração:** O sistema foi capaz de descobrir o dispositivo na rede e, ao aplicar o perfil configurado, autenticou-se e extraiu metadados (versão de *firmware*, modelo) automaticamente, validando a camada de abstração sem necessidade de alteração de código.
- **Monitoramento em Tempo Real:** A integração via *WebSockets* permitiu a visualização instantânea de métricas como latência, *uptime* e tráfego. O algoritmo de topologia renderizou corretamente o grafo, exibindo a *basestation* e a CPE conectada.
- **Gerenciamento Ativo:** O sistema executou com sucesso sequências de comandos complexos, incluindo alteração de configurações (via HTTP JSON) e reinicialização remota (*reboot* via SSH). O controlador de configuração (*ConfigController*) gerenciou a execução assíncrona, mantendo a responsividade da interface.

A única limitação observada foi a validação da atualização de *firmware* em massa, devido à indisponibilidade de arquivos de imagem compatíveis no ambiente de laboratório, embora a arquitetura para envio de *payloads* de arquivo tenha sido implementada.

5. Conclusão

O RFSight atingiu seu objetivo de prover uma ferramenta de gerenciamento modular e agnóstica a fabricantes, mitigando o problema do *vendor lock-in* enfrentado por ISPs regionais. A implementação da camada de abstração via *Profiles* e *Drivers* provou-se uma estratégia de engenharia de *software* robusta para lidar com a heterogeneidade de *hardware*.

Os resultados validam a viabilidade técnica de utilizar tecnologias *open-source* modernas para criar soluções de baixo custo e alta performance. Como trabalhos futuros, sugere-se a expansão para protocolos como TR-069 e NETCONF, bem como operação do sistema em nuvem, alinhado à expansão das possibilidades de configurações de ações e payloads dentro do contexto da Camada de Abstração de Dispositivo, como a execução em massa, por exemplo.

Referências

Bertollo, M. (2021). "A Internet das Coisas (IoT) no campo brasileiro: as redes informacionais, as novas dinâmicas da produção agrícola e os provedores regionais". *XIV Encontro Nacional de Pós Graduação e Pesquisa em Geografia*. https://www.editorarealize.com.br/editora/anais/enanpege/2021/TRABALHO_COMPLETO_EV154_MD1_SA138_ID147212112021132134.pdf

Campos, E. A. (2018). "Estudo de caso de ferramentas para gerenciamento de redes de computadores". Trabalho de Conclusão de Curso, Instituto Federal de Santa Catarina. https://repositorio.ifsc.edu.br/bitstream/handle/123456789/345/TCC_ElaineCampos.pdf?sequence=1&isAllowed=y

CNA e ABIMAQ (2023). "Soluções para conectividade rural". https://www.cnabrasil.org.br.

Stallings, W. (2014). *Data and Computer Communications*. 10. ed. Pearson Education Limited. https://archive.org/details/data-and-computer-communication-stallings-10e

APÊNDICE C – Código-fonte

O código-fonte do sistema desenvolvido pode ser encontrado no repositório de códigos da Universidade Federal de Santa Catarina (UFSC), através do seguinte *link*: https://codigos.ufsc.br/bruno.petri/rfsight. As orientações de instalação e uso do sistema se encontram no arquivo *README.md* na raiz do repositório.

ANEXO A – Respostas Entrevistado 1

Formulário de Levantamento de Requisitos para Sistema de Gerenciamento de Redes Radiofrequência (RFSight)

Prezado(a) profissional,

Este formulário faz parte de uma pesquisa acadêmica para o desenvolvimento de um Trabalho de Conclusão de Curso em Sistemas de Informação na Universidade Federal de Santa Catarina (UFSC).

O objetivo deste trabalho é desenvolver um **protótipo de sistema de gerenciamento para redes de radiofrequência**, nomeado **RFSight**, voltado especialmente para as necessidades de Provedores de Internet (ISPs).

Sua experiência e conhecimento prático são fundamentais para validarmos as funcionalidades mais importantes e garantirmos que a solução proposta seja relevante para os desafios reais do setor. As respostas são confidenciais e serão usadas exclusivamente para fins acadêmicos.

Tempo estimado para preenchimento: 10-15 minutos.

Agradeço imensamente sua colaboração!

Perfil e Contexto Profissional

Qual sua principal área de atuação? *

- ○ Gestão de Redes / Infraestrutura
- ◉ Desenvolvimento de Produtos para Telecom
- ○ Gestão de Negócios / Estratégia em ISP
- ○ Engenharia de Redes
- ○ Outro: _____

Quantos anos tem de experiência na área? *

7

Quais ferramentas você tem conhecimento ou utiliza hoje para o gerenciamento/monitoramento de redes de radiofrequência? (Possível marcar mais de uma) *

- [x] UNMS / UISP (Ubiquiti)
- [ ] The Dude (Mikrotik)
- [x] Zabbix
- [x] Grafana
- [ ] Scripts próprios / Desenvolvimento interno
- [ ] Planilhas / Controle manual
- [x] Outro: Cloudnet, INC Cloud e Omada

[OPCIONAL] Em sua opinião, qual é a maior dificuldade ou funcionalidade que falta nas ferramentas selecionadas?

Informação ou triggers de erros de rede, link ou APs com falhas. Alarmes de tentativas de invasão. E controle customizado por AP ou switch.

Validação de Funcionalidades e Métricas

Quanto às funcionalidades essenciais do protótipo: *

Para cada funcionalidade listada a seguir, indique o grau de importância para sua operação, considerando a seguinte escala:

1 - Não é necessária
2 - Pouco relevante
3 - Útil
4 - Importante
5 - Essencial

	1	2	3	4	5
Adotar dispositivos	○	○	○	○	◉
Remover dispositivos adotados	○	○	○	○	◉
Configurar individualmente cada dispositivo adotado	○	○	◉	○	○
Configurar dispositivos em massa	○	○	○	○	◉
Atualizar dispositivos de forma individual	○	○	◉	○	○
Atualizar dispositivos em massa	○	○	○	○	◉
Monitorar status individual dos dispositivos	○	○	○	○	◉
Monitorar status agregado da rede	○	○	○	○	◉
Visualizar gráficos de desempenho da rede	○	○	○	○	◉
Gerência de usuários do	○	○	○	◉	○

sistema (cadastro, edição e exclusão)					
Acesso com login por nome de usuário/senha	○	○	○	○	◉

Quanto aos dados de monitoramento dos dispositivos: *

Para cada item listado a seguir, indique o grau de importância para sua operação, considerando a seguinte escala:

1 - Não é necessário
2 - Pouco relevante
3 - Útil
4 - Importante
5 - Essencial

	1	2	3	4	5
Status (Online/Offline)	○	○	○	○	◉
Uptime (Tempo de atividade)	○	○	○	○	◉
Hostname	○	○	○	○	◉
Endereços IP e MAC	○	○	○	○	◉
Versão de firmware	○	○	○	○	◉
Latência	○	○	○	◉	○
Localização	○	○	◉	○	○

Quanto às métricas e organização de um dashboard de monitoramento: *

Para cada item listado a seguir, indique o grau de importância para sua operação, considerando a seguinte escala:

1 - Não é necessário
2 - Pouco relevante
3 - Útil
4 - Importante
5 - Essencial

	1	2	3	4	5
Quantidade total de dispositivos Online / Offline	○	○	○	○	◉
Throughput (Download/Upload) agregado de toda a rede	○	○	○	○	◉
Throughput por enlace individual	○	○	○	◉	○
Um "Gráfico de Saúde da Rede" (tipo Gauge).	○	○	◉	○	○
Topologia da rede	○	○	○	◉	○

Comentários Finais

[OPCIONAL] Você tem alguma sugestão, crítica ou comentário adicional que possa contribuir para o desenvolvimento do protótipo RFSight?

Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários

ANEXO B – Respostas Entrevistado 2

Formulário de Levantamento de Requisitos para Sistema de Gerenciamento de Redes Radiofrequência (RFSight)

Prezado(a) profissional,

Este formulário faz parte de uma pesquisa acadêmica para o desenvolvimento de um Trabalho de Conclusão de Curso em Sistemas de Informação na Universidade Federal de Santa Catarina (UFSC).

O objetivo deste trabalho é desenvolver um **protótipo de sistema de gerenciamento para redes de radiofrequência**, nomeado **RFSight**, voltado especialmente para as necessidades de Provedores de Internet (ISPs).

Sua experiência e conhecimento prático são fundamentais para validarmos as funcionalidades mais importantes e garantirmos que a solução proposta seja relevante para os desafios reais do setor. As respostas são confidenciais e serão usadas exclusivamente para fins acadêmicos.

Tempo estimado para preenchimento: 10-15 minutos.

Agradeço imensamente sua colaboração!

Perfil e Contexto Profissional

Qual sua principal área de atuação? *

- ○ Gestão de Redes / Infraestrutura
- ◉ Desenvolvimento de Produtos para Telecom
- ○ Gestão de Negócios / Estratégia em ISP
- ○ Engenharia de Redes
- ○ Outro: _____

Quantos anos tem de experiência na área? *

10

Quais ferramentas você tem conhecimento ou utiliza hoje para o gerenciamento/monitoramento de redes de radiofrequência? (Possível marcar mais de uma) *

- [x] UNMS / UISP (Ubiquiti)
- [x] The Dude (Mikrotik)
- [x] Zabbix
- [x] Grafana
- [x] Scripts próprios / Desenvolvimento interno
- [x] Planilhas / Controle manual
- [ ] Outro: _____

[OPCIONAL] Em sua opinião, qual é a maior dificuldade ou funcionalidade que falta nas ferramentas selecionadas?

Possibilidade de acesso aos produtos gerenciados de forma centralizada, gráficos explicando o estado atual da rede em tempo real e uma forma de aplicar configurações de forma temporária e caso algo de errado a configuração anterior funcional volte a funcionar.

Validação de Funcionalidades e Métricas

Quanto às funcionalidades essenciais do protótipo: *

Para cada funcionalidade listada a seguir, indique o grau de importância para sua operação, considerando a seguinte escala:

1 - Não é necessária
2 - Pouco relevante
3 - Útil
4 - Importante
5 - Essencial

	1	2	3	4	5
Adotar dispositivos	○	○	◉	○	○
Remover dispositivos adotados	○	○	◉	○	○
Configurar individualmente cada dispositivo adotado	○	◉	○	○	○
Configurar dispositivos em massa	○	○	◉	○	○
Atualizar dispositivos de forma individual	○	◉	○	○	○
Atualizar dispositivos em massa	○	○	◉	○	○
Monitorar status individual dos dispositivos	○	○	◉	○	○
Monitorar status agregado da rede	○	○	◉	○	○
Visualizar gráficos de desempenho da rede	○	○	◉	○	○
Gerência de usuários do	○	○	◉	○	○

sistema (cadastro, edição e exclusão)					
Acesso com login por nome de usuário/senha	○	○	◉	○	○

Quanto aos dados de monitoramento dos dispositivos: *

Para cada item listado a seguir, indique o grau de importância para sua operação, considerando a seguinte escala:

1 - Não é necessário
2 - Pouco relevante
3 - Útil
4 - Importante
5 - Essencial

	1	2	3	4	5
Status (Online/Offline)	○	○	◉	○	○
Uptime (Tempo de atividade)	○	○	◉	○	○
Hostname	○	◉	○	○	○
Endereços IP e MAC	○	○	◉	○	○
Versão de firmware	○	○	◉	○	○
Latência	○	○	◉	○	○
Localização	○	○	◉	○	○

Quanto às métricas e organização de um dashboard de monitoramento: *

Para cada item listado a seguir, indique o grau de importância para sua operação, considerando a seguinte escala:

1 - Não é necessário
2 - Pouco relevante
3 - Útil
4 - Importante
5 - Essencial

	1	2	3	4	5
Quantidade total de dispositivos Online / Offline	○	○	◉	○	○
Throughput (Download/Upload) agregado de toda a rede	○	○	◉	○	○
Throughput por enlace individual	○	○	◉	○	○
Um "Gráfico de Saúde da Rede" (tipo Gauge).	○	○	◉	○	○
Topologia da rede	○	○	◉	○	○

Comentários Finais

[OPCIONAL] Você tem alguma sugestão, crítica ou comentário adicional que possa contribuir para o desenvolvimento do protótipo RFSight?

Proporcionar uma forma de acessar o device via cli ou web é uma boa forma.

Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários

ANEXO C – Respostas Entrevistado 3

Formulário de Levantamento de Requisitos para Sistema de Gerenciamento de Redes Radiofrequência (RFSight)

Prezado(a) profissional,

Este formulário faz parte de uma pesquisa acadêmica para o desenvolvimento de um Trabalho de Conclusão de Curso em Sistemas de Informação na Universidade Federal de Santa Catarina (UFSC).

O objetivo deste trabalho é desenvolver um **protótipo de sistema de gerenciamento para redes de radiofrequência**, nomeado **RFSight**, voltado especialmente para as necessidades de Provedores de Internet (ISPs).

Sua experiência e conhecimento prático são fundamentais para validarmos as funcionalidades mais importantes e garantirmos que a solução proposta seja relevante para os desafios reais do setor. As respostas são confidenciais e serão usadas exclusivamente para fins acadêmicos.

Tempo estimado para preenchimento: 10-15 minutos.

Agradeço imensamente sua colaboração!

Perfil e Contexto Profissional

Qual sua principal área de atuação? *

- ○ Gestão de Redes / Infraestrutura
- ○ Desenvolvimento de Produtos para Telecom
- ◉ Gestão de Negócios / Estratégia em ISP
- ○ Engenharia de Redes
- ○ Outro: _____

Quantos anos tem de experiência na área? *

5

Quais ferramentas você tem conhecimento ou utiliza hoje para o gerenciamento/monitoramento de redes de radiofrequência? (Possível marcar mais de uma) *

- [x] UNMS / UISP (Ubiquiti)
- [ ] The Dude (Mikrotik)
- [x] Zabbix
- [x] Grafana
- [x] Scripts próprios / Desenvolvimento interno
- [x] Planilhas / Controle manual
- [ ] Outro: _____

[OPCIONAL] Em sua opinião, qual é a maior dificuldade ou funcionalidade que falta nas ferramentas selecionadas?

_____

Validação de Funcionalidades e Métricas

Quanto às funcionalidades essenciais do protótipo: *

Para cada funcionalidade listada a seguir, indique o grau de importância para sua operação, considerando a seguinte escala:

1 - Não é necessária
2 - Pouco relevante
3 - Útil
4 - Importante
5 - Essencial

	1	2	3	4	5
Adotar dispositivos	○	○	○	○	◉
Remover dispositivos adotados	○	○	○	○	◉
Configurar individualmente cada dispositivo adotado	○	○	○	○	◉
Configurar dispositivos em massa	○	○	○	○	◉
Atualizar dispositivos de forma individual	○	○	○	◉	○
Atualizar dispositivos em massa	○	○	○	○	◉
Monitorar status individual dos dispositivos	○	○	○	○	◉
Monitorar status agregado da rede	○	○	○	○	◉
Visualizar gráficos de desempenho da rede	○	○	◉	○	○
Gerência de usuários do	○	○	○	◉	○

sistema (cadastro, edição e exclusão)					
Acesso com login por nome de usuário/senha	○	○	◉	○	○

Quanto aos dados de monitoramento dos dispositivos: *

Para cada item listado a seguir, indique o grau de importância para sua operação, considerando a seguinte escala:

1 - Não é necessário
2 - Pouco relevante
3 - Útil
4 - Importante
5 - Essencial

	1	2	3	4	5
Status (Online/Offline)	○	○	○	○	◉
Uptime (Tempo de atividade)	○	○	○	○	◉
Hostname	○	○	○	○	◉
Endereços IP e MAC	○	○	○	○	◉
Versão de firmware	○	○	○	○	◉
Latência	○	○	○	○	◉
Localização	○	○	○	◉	○

Quanto às métricas e organização de um dashboard de monitoramento: *

Para cada item listado a seguir, indique o grau de importância para sua operação, considerando a seguinte escala:

1 - Não é necessário
2 - Pouco relevante
3 - Útil
4 - Importante
5 - Essencial

	1	2	3	4	5
Quantidade total de dispositivos Online / Offline	○	○	○	○	◉
Throughput (Download/Upload) agregado de toda a rede	○	○	○	○	◉
Throughput por enlace individual	○	○	○	○	◉
Um "Gráfico de Saúde da Rede" (tipo Gauge).	○	○	◉	○	○
Topologia da rede	○	○	○	◉	○

Comentários Finais

[OPCIONAL] Você tem alguma sugestão, crítica ou comentário adicional que possa contribuir para o desenvolvimento do protótipo RFSight?

Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários