Campus Vilhena
Coordenação do Curso Superior em Tecnologia em Análise e Desenvolvimento de Sistemas

GEOVANNA ROCHA DA SILVA

MicroNET: Aplicação Web para Gerenciamento de Usuários e Planos Integrados ao MikroTik

VILHENA - RO
2026

Geovanna Rocha Da Silva

MicroNET: Aplicação Web para Gerenciamento de Usuários e Planos Integrados ao MikroTik

Artigo entregue como Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO), Campus Vilhena, como requisito parcial para obtenção do grau de Tecnólogo, junto ao Curso Superior em Tecnologia em Análise e Desenvolvimento de Sistemas, sob a orientação do professor José Lucas Brandão Montes

VILHENA - RO
2026

Ficha catalográfica elaborada pelo Sistema Gerador de Ficha Catalográfica do IFRO.

Silva, Geovanna Rocha da.
Micronet: aplicação web para gerenciamento de usuários e planos integrados ao MikroTik / Geovanna Rocha da Silva. - Vilhena, 2026.
34 f. : il.

Orientador(a): Prof. Me. Jose Lucas Brandao Montes.

Trabalho de Conclusão de Curso (Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO, Vilhena, 2026.

1. Mikro Tik. 2. PPPoE. 3. Provedores de internet. 4. API REST. 5. Gerenciamento de contratos. I. Montes, Jose Lucas Brandao (orient.). II. Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO. III. Título.

Bibliotecário(a) Responsável: Rosilene Maria do Couto Marques, CRB-11/321

Geovanna Rocha Da Silva

MicroNET: Aplicação Web para Gerenciamento de Usuários e Planos Integrados ao MikroTik

Artigo entregue como Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO), Campus Vilhena, como requisito parcial para obtenção do grau de Tecnólogo, junto ao Curso Superior em Tecnologia em Análise e Desenvolvimento de Sistemas, sob a orientação do professor José Lucas Brandão Montes

Aprovado em 16/11/2026 pela banca examinadora.

_____

Prof. Mestre Roberto Simplício Guimarães

Examinador interno

_____

Prof. Mestre Gilberto Pereira da Silva

Examinador interno

_____

Prof. Mestre José Lucas Brandão Montes

Orientador

VILHENA - RO
2026

Resumo

Este trabalho apresenta o MicroNET, uma aplicação web integrada ao MikroTik RouterOS para o gerenciamento de clientes, planos, contratos, cobranças e acessos à Internet por PPPoE. O objetivo foi aproximar a gestão administrativa das operações de rede, centralizando informações e buscando reduzir procedimentos manuais. A solução utiliza React, Python com FastAPI, MongoDB e autenticação JWT, e se comunica com o roteador pela RouterOS REST API sobre HTTPS. O desenvolvimento foi incremental, organizado por Kanban, e validado por testes automatizados e testes práticos em laboratório virtual. O sistema gerencia cadastros, contratos e cobranças, provisiona PPP Profiles e PPP Secrets, consulta conexões ativas e executa bloqueio, desbloqueio e desconexão como operações distintas, com registro do resultado da sincronização. Também oferece controle de acesso por perfil, processos automáticos, auditoria e publicação em nuvem, com a comunicação HTTPS com o roteador encaminhada por túnel SSH reverso e certificado TLS validado. Os resultados incluem 1.103 testes aprovados na API, 421 testes unitários no frontend e 274 testes de ponta a ponta, além da validação prática da autenticação PPPoE e das operações no RouterOS. A avaliação foi realizada em laboratório virtual, o acesso ao roteador depende da estação do laboratório, a sincronização atua somente sobre Secrets criados pelo MicroNET e cada contrato contempla um único login PPPoE.

Palavras-chave: MikroTik; PPPoE; provedores de internet; API REST; gerenciamento de contratos.

Abstract

This work presents MicroNET, a web application integrated with MikroTik RouterOS for managing customers, plans, contracts, billing, and Internet access through PPPoE. Its objective was to connect administrative management with network operations by centralizing information and seeking to reduce manual procedures. The solution uses React, Python with FastAPI, MongoDB, and JWT authentication, communicating with the router through the RouterOS REST API over HTTPS. Development was incremental, organized with Kanban, and validated through automated tests and practical tests in a virtual laboratory. The system supports registration and management of customers, contracts, and billing; PPP Profile and PPP Secret provisioning; active connection queries; and distinct blocking, unblocking, and disconnection operations, with synchronization results recorded. It also offers role-based access control, automated processes, auditing, and cloud deployment, with the HTTPS communication with the router forwarded through a reverse SSH tunnel and a validated TLS certificate. Results include 1,103 passing API tests, 421 passing frontend unit tests, and 274 passing end-to-end tests, in addition to practical validation of PPPoE authentication and RouterOS operations. Evaluation was carried out in the virtual laboratory, router access depends on the laboratory workstation, synchronization affects only PPP Secrets created by MicroNET, and each contract supports a single PPPoE login.

Keywords: MikroTik; PPPoE; internet service providers; REST API; contract management.

1 INTRODUÇÃO

O crescimento dos serviços de acesso à Internet amplia a necessidade de organizar informações de clientes, contratos, planos, cobranças e recursos de rede. Em provedores, esses dados e as configurações dos equipamentos podem permanecer distribuídos em ferramentas distintas, exigindo procedimentos manuais e repetição de alterações (Maria, 2016; Lessi; Oliani Junior, 2023). Nesse cenário, o MikroTik RouterOS oferece recursos de autenticação PPPoE e gerenciamento de sessões (MikroTik, 2026a), enquanto sua API REST permite integrar aplicações externas às operações do equipamento (Biz, 2023; MikroTik, 2026b).

Diante desse contexto, a pergunta que orienta a pesquisa é: que operações de provisionamento e bloqueio de acesso PPPoE, normalmente executadas manualmente no MikroTik, podem ser automatizadas por uma aplicação web que utiliza o contrato como elo entre o cadastro, o financeiro e as credenciais de acesso? Para respondê-la, este trabalho tem como objetivo desenvolver o MicroNET, uma aplicação web integrada ao MikroTik RouterOS para o gerenciamento de clientes, planos, contratos, cobranças e acessos à Internet por PPPoE. A proposta consiste em aproximar a gestão administrativa das ações realizadas na rede, utilizando o contrato como vínculo entre cliente, plano e credenciais de acesso. Para isso, foram definidos como objetivos específicos: levantar os requisitos e as regras de negócio; desenvolver uma API com FastAPI, autenticação JWT e persistência em MongoDB; implementar a integração com o RouterOS por HTTPS, incluindo o provisionamento de PPP Profiles e PPP Secrets; construir a interface web em React; e implementar e validar as operações de bloqueio, desbloqueio e desconexão, com registro do resultado da sincronização.

A solução foi desenvolvida de forma incremental e validada por testes automatizados e testes práticos em laboratório virtual. O escopo deste trabalho contempla o gerenciamento de acessos por PPPoE. O trabalho está organizado em seis seções: a Seção 2 apresenta a fundamentação teórica; a Seção 3 descreve a metodologia; a Seção 4 detalha o desenvolvimento e a implementação; a Seção 5 apresenta e discute os resultados; e a Seção 6 reúne as considerações finais.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Provedores de Internet e Gerenciamento de Clientes

A operação de um provedor de Internet envolve o gerenciamento conjunto de clientes, serviços, contratos e recursos de rede, o que motiva a adoção de sistemas capazes de organizar essas informações. Soluções voltadas a provedores buscam centralizar dados e reduzir atividades manuais, favorecendo maior controle administrativo (Maria, 2016; Lessi; Oliani Junior, 2023). Além do cadastro de clientes, sistemas desse tipo podem reunir funções financeiras, acompanhamento de contratos e informações operacionais da rede. Essa integração reduz a fragmentação dos dados e contribui para que diferentes setores utilizem uma mesma base de informações (Maria, 2016; Lessi; Oliani Junior, 2023).

2.2 MikroTik RouterOS e PPPoE

Segundo Mamakos et al. (1999), o PPPoE permite estabelecer sessões PPP sobre redes Ethernet, combinando autenticação e identificação individual dos acessos. No MikroTik RouterOS, esse mecanismo é disponibilizado juntamente com recursos de roteamento, firewall e gerenciamento de usuários, permitindo associar credenciais a perfis de serviço (MikroTik, 2026a). Estudos aplicados com MikroTik RouterBoard também demonstram o uso do equipamento para controle de acesso e autenticação em ambientes de rede (Silva; Machado; Souza, 2023). No subsistema PPP do RouterOS, os perfis permitem definir parâmetros de acesso e as credenciais dos usuários são mantidas em PPP Secrets, enquanto as sessões efetivamente conectadas podem ser consultadas em /ppp active. Esses recursos são compatíveis com cenários em que o provedor necessita controlar individualmente os acessos de seus clientes (Matozinhos; Ramires, 2015; MikroTik, 2026a). Em provedores de maior porte, a autenticação costuma ser centralizada em um servidor RADIUS, que implementa autenticação, autorização e contabilização (AAA) e permite gerenciar múltiplos concentradores a partir de uma base única (MikroTik, 2026a). Neste trabalho, optou-se pelos PPP Secrets locais do próprio RouterOS, adequados ao escopo de um único concentrador e suficientes para demonstrar a integração, deixando a adoção de RADIUS como trabalho futuro para cenários com vários roteadores.

2.3 Gerenciamento de Redes

O gerenciamento de redes envolve o acompanhamento dos recursos e serviços da infraestrutura, permitindo centralizar informações e apoiar a execução de tarefas administrativas. Em provedores de internet, a integração entre sistemas de gestão e recursos de rede contribui para reduzir atividades manuais e melhorar o acompanhamento da operação (Lessi; Oliani Junior, 2023). Trabalhos recentes também destacam o monitoramento de ativos e da qualidade dos enlaces como apoio à disponibilidade e ao acompanhamento da infraestrutura (Rodrigues et al., 2025; Souza et al., 2024). A utilização de APIs em equipamentos de rede permite que aplicações externas consultem informações e executem operações de gerenciamento. Em ambientes RouterOS, essa abordagem possibilita integrar o software administrativo ao equipamento de rede e automatizar parte das rotinas relacionadas aos usuários e serviços (Biz, 2023).

2.4 Sistemas Web e APIs REST

Segundo Fielding (2000), o estilo arquitetural REST favorece interfaces uniformes e a separação entre componentes em aplicações distribuídas. No MicroNET, a interface foi construída em React e separada da API desenvolvida com FastAPI, mantendo independentes a apresentação, as regras de negócio e a integração com serviços externos. A documentação oficial dessas tecnologias descreve o uso de componentes no frontend e a construção de APIs baseadas em padrões abertos no backend (React, 2026; FastAPI, 2026). No contexto de equipamentos RouterOS, o uso de uma API possibilita que uma aplicação web consulte e altere recursos do roteador sem depender de configuração manual em cada operação. Essa característica é relevante para sistemas que precisam manter sincronizadas informações administrativas e recursos da rede (Biz, 2023).

2.5 Trabalhos Relacionados

Os trabalhos relacionados permitem situar o MicroNET entre soluções de gestão de provedores e pesquisas voltadas à integração entre software e infraestrutura de rede. A comparação evidencia diferentes abordagens para gestão administrativa, monitoramento por API e gerenciamento de dispositivos, servindo de base para identificar a lacuna explorada neste trabalho. O trabalho de Maria (2016) apresenta um sistema gerenciador de clientes, equipamentos e circuitos para provedores de

internet, voltado à organização desses registros em uma aplicação própria. Lessi e Oliani Junior (2023) abordam o papel de sistemas ERP, como o IXC Provedor, na gestão de serviços e no mapeamento de processos de empresas de telecomunicações. Biz (2023), por sua vez, aborda o monitoramento de redes por API em sistemas RouterOS, aproximando o software de gerenciamento da infraestrutura de rede.

Rodrigues et al. (2025) abordaram o monitoramento de ativos de rede com ferramentas de código aberto, e Souza et al. (2024) apresentaram um serviço de monitoramento da qualidade de link de Internet. Em conjunto, esses trabalhos tratam de partes específicas do problema, como gestão administrativa, monitoramento e acompanhamento da infraestrutura. Diferentemente dessas propostas, o MicroNET integra em um único fluxo Cliente → Contrato → Plano, o provisionamento PPPoE no MikroTik e o controle financeiro. Assim, o contrato funciona como elo entre a gestão administrativa e o acesso à Internet, permitindo executar pela aplicação operações como ativação, bloqueio, desbloqueio e desconexão.

No mercado brasileiro, sistemas consolidados de gestão para provedores — como IXC, SGP e Hubsoft, comerciais, e o MK-Auth, de distribuição gratuita — já integram cadastro, financeiro com emissão de boletos e notas fiscais, provisionamento de acesso (em geral por servidor RADIUS) e bloqueio automático por inadimplência, além de central do assinante e ordens de serviço. O MicroNET não compete em abrangência com essas plataformas: posiciona-se como um protótipo acadêmico, de código aberto, que demonstra e documenta a construção e a validação da integração entre a gestão de contratos e o provisionamento PPPoE diretamente no RouterOS, com PPP Secrets locais em vez de RADIUS, voltado a provedores de pequeno porte com um único concentrador. O Quadro 1 sintetiza essa comparação por critérios.

Quadro 1 – Comparação por critérios entre soluções de mercado e o MicroNET

Critério	IXC	SGP	Hubsoft	MK-Auth	MicroNET
Provisionamento PPPoE automático	**Sim**	**Sim**	**Sim**	**Sim**	**Sim**

Bloqueio automático por inadimplência	**Sim**	**Sim**	**Sim**	**Sim**	**Sim**
Cobrança integrada (boleto/Pix/NF-e)	**Sim**	**Sim**	**Sim**	**Parcial**	**Não (baixa manual)**
Autenticação via RADIUS / múltiplos concentradores	**Sim**	**Sim**	**Sim**	**Sim**	**Não (Secret local, 1 roteador)**
Central do assinante	**Sim**	**Sim**	**Sim**	**Sim**	**Não**
Auditoria / logs de operação	**Sim**	**Sim**	**Sim**	**Parcial**	**Sim**
Código aberto	**Não**	**Não**	**Não**	**Parcial**	**Sim**
Maturidade	**Produção**	**Produção**	**Produção**	**Produção**	**Protótipo (laboratório)**

Fonte: elaborada pela autora (2026).

2.6 Autenticação com JWT e Controle de Acesso

A autenticação baseada em tokens permite que o servidor identifique o usuário a cada requisição sem manter o estado da sessão, em coerência com o princípio de que cada requisição REST deve conter as informações necessárias ao seu processamento (Fielding, 2000). O JSON Web Token (JWT) é um formato compacto para representar declarações entre as partes, entre as quais está o identificador único do token (jti), que viabiliza a sua revogação individual (Jones; Bradley; Sakimura, 2015). No MicroNET, o token identifica o usuário e o servidor verifica o perfil, Administrador ou Operador, antes de liberar cada operação protegida, conforme a Seção 4.1.

2.7 Bancos de Dados Orientados a Documentos

O MongoDB armazena os dados em documentos agrupados em coleções, sem exigir um esquema único e rígido para todos os registros, e permite relacionar documentos por meio de identificadores guardados em campos de referência (MongoDB, 2026). Essa característica atende ao MicroNET, em que o contrato reúne dados administrativos, financeiros e de integração com o equipamento, e as relações entre cliente, contrato e plano são mantidas por identificadores. O MongoDB também oferece índices com expiração automática (TTL), recurso empregado para descartar registros temporários, como os tokens revogados. Como as relações entre cliente, contrato e plano são bem definidas, um banco relacional também seria adequado; optou-se pelo MongoDB pela aderência ao formato JSON retornado pela API do RouterOS e pela flexibilidade de esquema durante o desenvolvimento. A ausência de transações entre documentos e de integridade referencial imposta pelo banco é tratada na aplicação, por meio de validações nas regras de negócio e de índices de unicidade (MongoDB, 2026).

3 METODOLOGIA

3.1 Caracterização da Pesquisa

O trabalho caracteriza-se como pesquisa aplicada e exploratória, pois busca desenvolver e validar uma solução para um problema prático de gerenciamento de provedores. A abordagem foi predominantemente qualitativa, com a observação do comportamento da aplicação e de sua integração com a rede, apoiada em evidências quantitativas dos testes automatizados. O desenvolvimento foi incremental, organizado em um quadro Kanban com as colunas backlog, desenvolvimento, testes e concluído: cada funcionalidade era definida, implementada, testada e integrada às demais. O Git foi usado para o controle de versão, com *branches* para o desenvolvimento isolado antes da integração ao código principal.

3.2 Levantamento e Definição dos Requisitos

O levantamento contemplou requisitos funcionais, não funcionais e regras de negócio. Os requisitos funcionais estão no Quadro 2 e os não funcionais, no Quadro 3.

3.2.1 Requisitos Funcionais

Quadro 2 – Requisitos funcionais

Categoria	Códigos	Requisitos
Autenticação e acesso	RF01–RF07	Cadastrar usuários por administradores; realizar login e logout; controlar os perfis Administrador e Operador; restringir ações por nível de acesso; editar e desativar usuários.
Gestão de clientes	RF08–RF12	Cadastrar, editar, excluir, listar e buscar clientes por nome ou CPF.
Gestão de planos	RF13–RF16	Cadastrar, editar, excluir e listar os planos de Internet disponíveis.
Gestão de contratos	RF17–RF22	Cadastrar contratos vinculados a clientes; permitir múltiplos contratos por cliente; vincular o contrato a um plano; armazenar data de início, status e valor; ativar, bloquear e cancelar; listar contratos por cliente.
Gestão financeira	RF23–RF29	Gerar cobranças automáticas; registrar pagamentos; listar cobranças; indicar situação paga, pendente ou vencida; manter histórico; permitir baixa manual; calcular valores recebidos, pendentes e vencidos.
Integração e controle de acesso	RF30–RF38	Configurar a conexão com o MikroTik; criar e atualizar a credencial PPPoE; remover ou desativar ao cancelar; sincronizar os sistemas; exibir conexões ativas; bloquear contratos inadimplentes; desbloquear automática ou manualmente.
Dashboard	RF39–RF43	Exibir total de clientes, contratos ativos e bloqueados, planos disponíveis, gráficos de crescimento e indicadores financeiros.
Logs	RF44–RF45	Registrar as ações realizadas, com data, usuário e tipo de ação.

Fonte: elaborada pela autora (2026).

A principal regra de negócio é que o cliente não se liga diretamente a um plano: o vínculo ocorre pelo Contrato, que concentra o plano, o valor mensal, a situação do serviço e as credenciais PPPoE. A estrutura do domínio é Cliente → Contrato → Plano → PPPoE → MikroTik. Os perfis Administrador e Operador compartilham as funções operacionais; o gerenciamento de usuários do sistema é exclusivo do Administrador, que ainda pode forçar o reenvio das senhas PPPoE na sincronização. As demais regras constam no Quadro 4, na Seção 3.2.3.

3.2.2 Requisitos Não Funcionais

Quadro 3 – Requisitos não funcionais

Categoria	Códigos	Requisitos
Segurança	RNF 01–RNF 03	Armazenar senhas de forma criptografada; utilizar autenticação baseada em token; controlar o acesso conforme o perfil.
Desempenho	RNF 04–RNF 05	Responder em até 2 segundos e oferecer suporte a múltiplos usuários simultâneos.
Integração	RNF 06–RNF 07	Comunicar-se com o MikroTik por API e tratar falhas de comunicação.
Usabilidade	RNF 08–RNF 09	Disponibilizar interface responsiva e alternância entre os temas claro e escuro.
Escalabilidade	RNF 10–RNF 11	Suportar o crescimento do número de usuários sem perda de desempenho e permitir futuras expansões.

Fonte: elaborada pela autora (2026).

3.2.3 Regras de Negócio

Quadro 4 – Regras de negócio

Código	Regra de negócio
RN 01	Um cliente pode possuir múltiplos contratos.
RN 02	Cada contrato deve estar vinculado a apenas um plano.
RN 03	O acesso à Internet pertence ao contrato, e não diretamente ao cliente.
RN 04	O status do contrato depende da situação financeira.
RN 05	Contratos inadimplentes devem ser bloqueados.
RN 06	O desbloqueio deve ocorrer após a confirmação do pagamento.
RN 07	Um plano não pode ser excluído enquanto estiver vinculado a contratos ativos.
RN 08	Apenas administradores podem cadastrar usuários.
RN 09	Operadores não podem alterar permissões.
RN 10	O sistema deve permitir intervenção manual no bloqueio e no desbloqueio.
RN 11	Cada contrato deve gerar cobranças periódicas automaticamente.

Fonte: elaborada pela autora (2026).

3.3 Organização do Desenvolvimento

Para organizar o desenvolvimento foi adotado o Kanban, método que permite visualizar o fluxo de trabalho e acompanhar o andamento das tarefas. No MicroNET, o quadro foi dividido em backlog, desenvolvimento, testes e concluído (Quadro 5).

Quadro 5 – Organização das atividades no Kanban

Etapa	Descrição
Backlog	Funcionalidades planejadas
Em desenvolvimento	Funcionalidades em implementação
Testes	Funcionalidades em processo de validação
Concluído	Funcionalidades implementadas e validadas

Fonte: elaborada pela autora (2026).

O Git foi utilizado para controle de versão, com branches destinados ao desenvolvimento isolado antes da integração ao código principal.

3.4 Tecnologias e Validação

O Quadro 6 resume as tecnologias empregadas. A validação seguiu duas frentes: testes automatizados, que verificam autenticação, permissões, persistência, regras de negócio e rotas, e testes práticos de integração, que verificam o acesso à RouterOS REST API por HTTPS, o provisionamento de PPP Profiles e Secrets, a identificação de sessões em /ppp active e as operações de bloqueio, desbloqueio e desconexão.

Quadro 6 – Tecnologias e ferramentas utilizadas

Categoria	Tecnologias e utilização
Interface	React - desenvolvimento da interface web
Backend	Python e FastAPI - implementação da API REST e das regras de negócio
Dados e autenticação	MongoDB, MinIO e JWT - dados, arquivos e autenticação

Infraestrutura de rede	MikroTik RouterOS e RouterOS REST API - PPPoE e integração com o equipamento
Apoio e testes	Docker, VirtualBox, Ubuntu, Git, WinBox e OpenSSH (serviços, laboratório, cliente PPPoE, versionamento, administração e túnel).

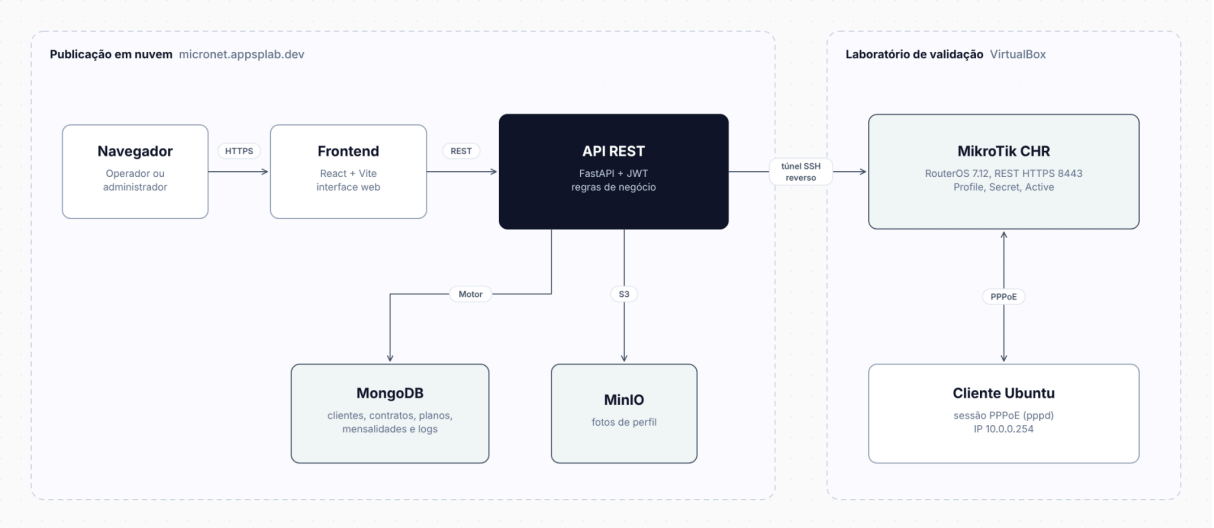
Fonte: elaborada pela autora (2026).

4 DESENVOLVIMENTO E IMPLEMENTAÇÃO

4.1 Arquitetura, banco de dados e autenticação

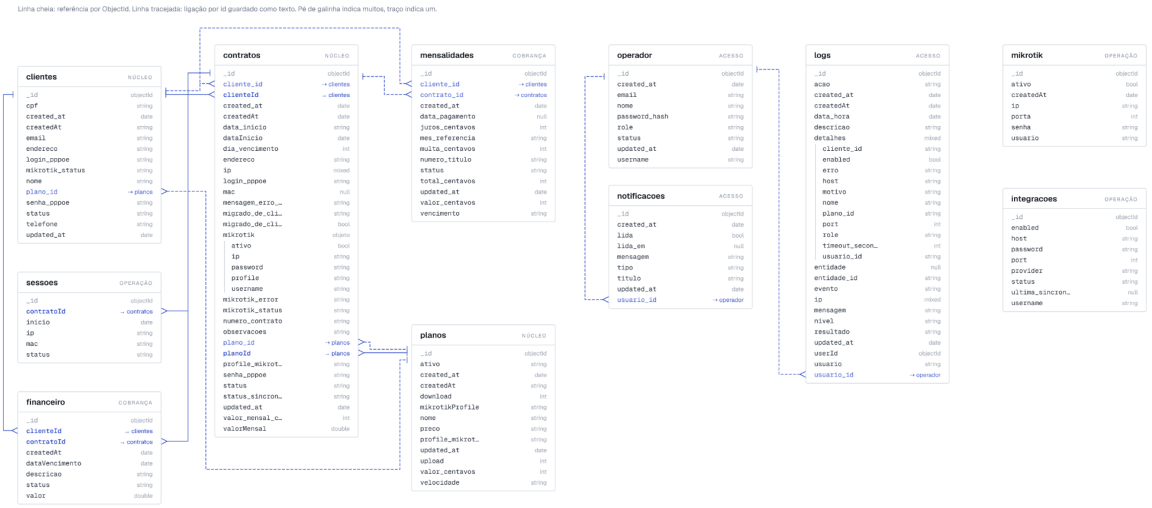
O MicroNET tem frontend em React e backend em Python com FastAPI. A API concentra as regras de negócio, o acesso ao MongoDB e ao MinIO e a comunicação com o MikroTik pela RouterOS REST API sobre HTTPS; a interface não se comunica diretamente com o equipamento (Figura 1). O backend é organizado em rotas, controladores, serviços e repositórios: a rota declara o endpoint e verifica o perfil, o serviço concentra regras de negócio, auditoria e comunicação com o RouterOS, e o repositório isola o MongoDB (FastAPI, 2026). As rotinas automáticas executam no próprio processo da API (Seção 4.3.3).

Figura 1 – Arquitetura do MicroNET e ambientes de execução



Fonte: elaborada pela autora (2026).

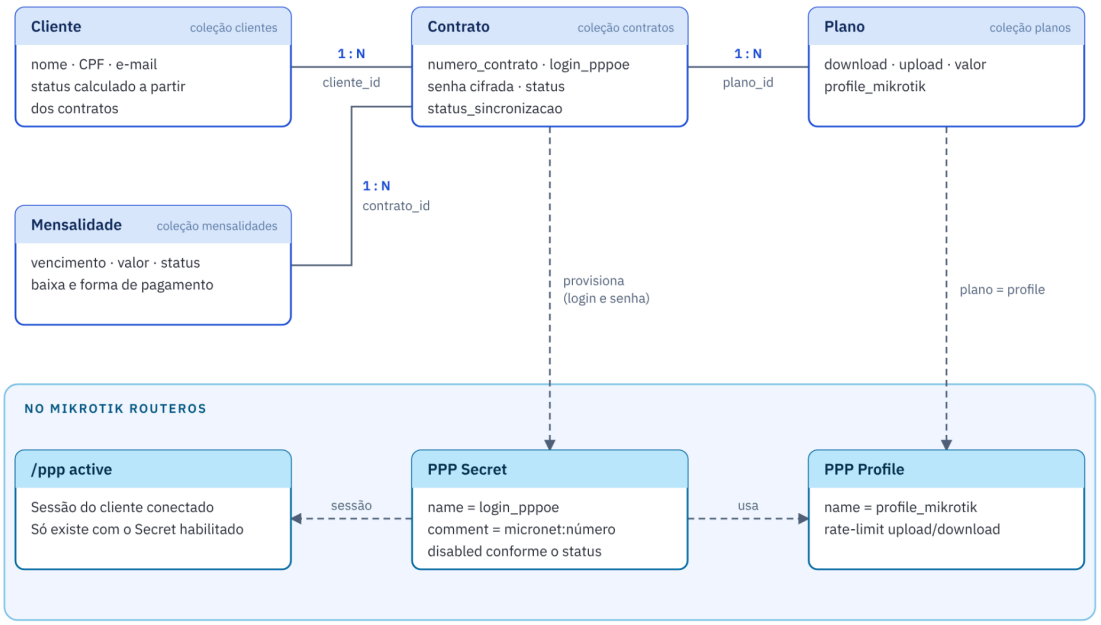
Figura 2 – Coleções do MongoDB e relacionamentos entre elas



Fonte: elaborada pela autora (2026).

No MongoDB, a persistência abrange coleções destinadas a usuários, cadastros de clientes, planos comercializados, vínculos contratuais, mensalidades geradas, parâmetros de integração e registros de auditoria, estabelecendo a associação entre entidades por identificadores (MongoDB, 2026). A Figura 2 apresenta as coleções e seus relacionamentos. A Figura 3 ilustra a relação entre o cliente, o contrato e o plano com os elementos do subsistema PPP, representados pelo PPP Secret e pelo PPP Profile. O contrato concentra igualmente o IP e o MAC da conexão PPPoE, dispensando uma coleção específica para sessões ativas, as quais são obtidas diretamente do caminho /ppp active no equipamento. As quantias financeiras são mantidas em centavos, a credencial de acesso ao roteador é armazenada de forma cifrada para viabilizar sua retransmissão ao RouterOS, e a integridade de dados como CPF, nome e profile é assegurada mediante índices de unicidade.

Figura 3 – Vínculo entre cliente, contrato, plano e os recursos PPP do MikroTik



Fonte: elaborada pela autora (2026).

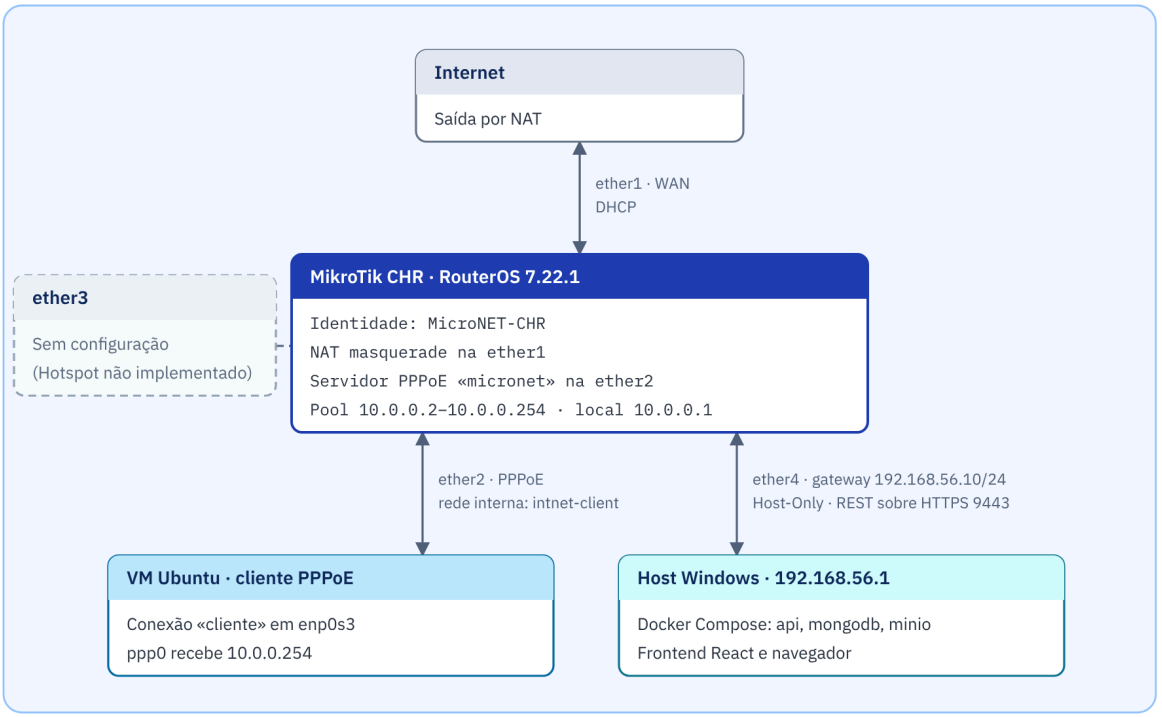
O login valida usuário e senha (hash bcrypt) e emite um JWT com expiração configurável, entregue ao navegador em cookie HttpOnly; as requisições que alteram dados exigem cabeçalho próprio do frontend como defesa contra requisições forjadas, e clientes de API podem usar o cabeçalho *Authorization*. A cada requisição protegida, a API valida o token, confere se o jti não consta da coleção de tokens revogados e verifica o perfil exigido, respondendo 403 quando não permitido (Jones; Bradley; Sakimura, 2015). No logout, o jti é revogado até a expiração do token, e o índice TTL remove o registro depois disso. Após tentativas inválidas sucessivas, o login responde 429 e registra o bloqueio temporário.

4.2 Ambiente de laboratório com MikroTik

O desenvolvimento e a validação da integração foram realizados em um laboratório virtual com um MikroTik CHR (RouterOS 7.22.1) e uma máquina Ubuntu como cliente PPPoE, ambos no VirtualBox (Figuras 4 e 5). No CHR, a ether1 é a WAN por DHCP, a ether2 atende o servidor PPPoE, a ether3 não é utilizada, pois o Hotspot

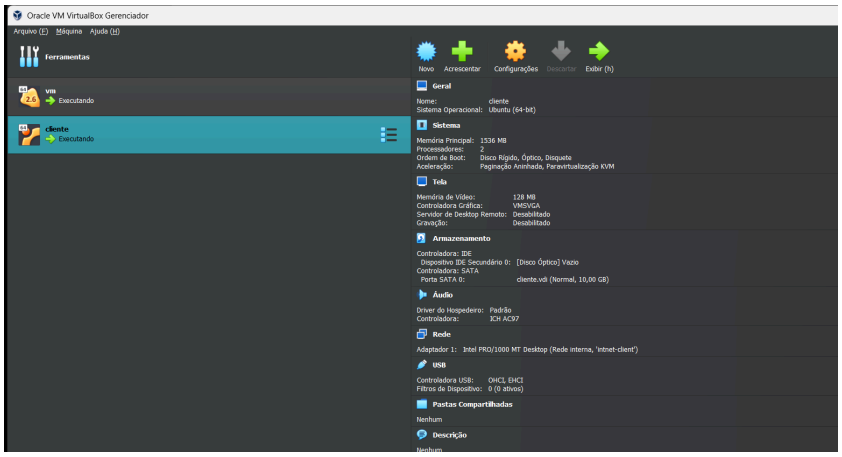
está fora do escopo deste trabalho, e a ether4 é a rede de gerenciamento (192.168.56.10/24). A comunicação administrativa do MicroNET usa apenas a rede Host-Only, e o tráfego do cliente fica em rede interna separada. O servidor PPPoE usa o pool 10.0.0.2 -- 10.0.0.254, com endereço local 10.0.0.1. Cada Plano é associado a um PPP Profile e cada Contrato concentra as credenciais usadas para provisionar o PPP Secret; as sessões conectadas são verificadas em /ppp active, distinguindo usuários provisionados de conexões ativas (MikroTik, 2026a). A Figura 6 mostra, no WinBox, as interfaces do CHR, com a interface dinâmica da sessão PPPoE, e o servidor PPPoE na ether2.

Figura 4 – Topologia do ambiente virtual de validação



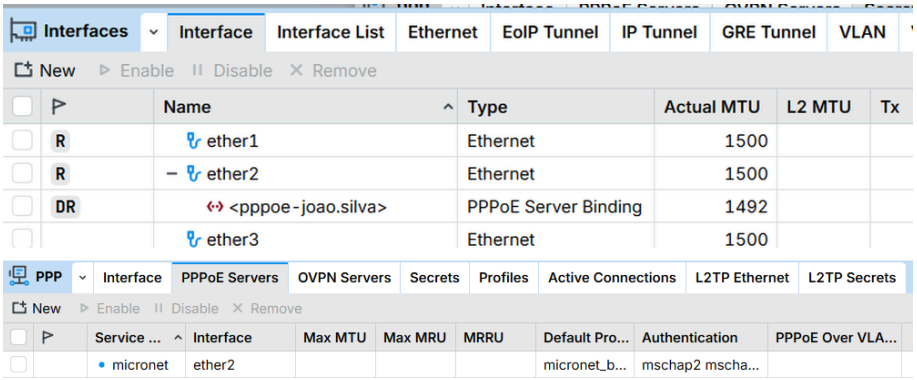
Fonte: elaborada pela autora (2026).

Figura 5 – Máquinas virtuais do laboratório no VirtualBox: MikroTik CHR (vm) e cliente Ubuntu



Fonte: elaborada pela autora (2026).

Figura 6 – Interfaces do CHR com a interface dinâmica da sessão PPPoE e servidor PPPoE na ether2, vistos no WinBox



Fonte: elaborada pela autora (2026).

4.3 Integração entre a aplicação e o MikroTik

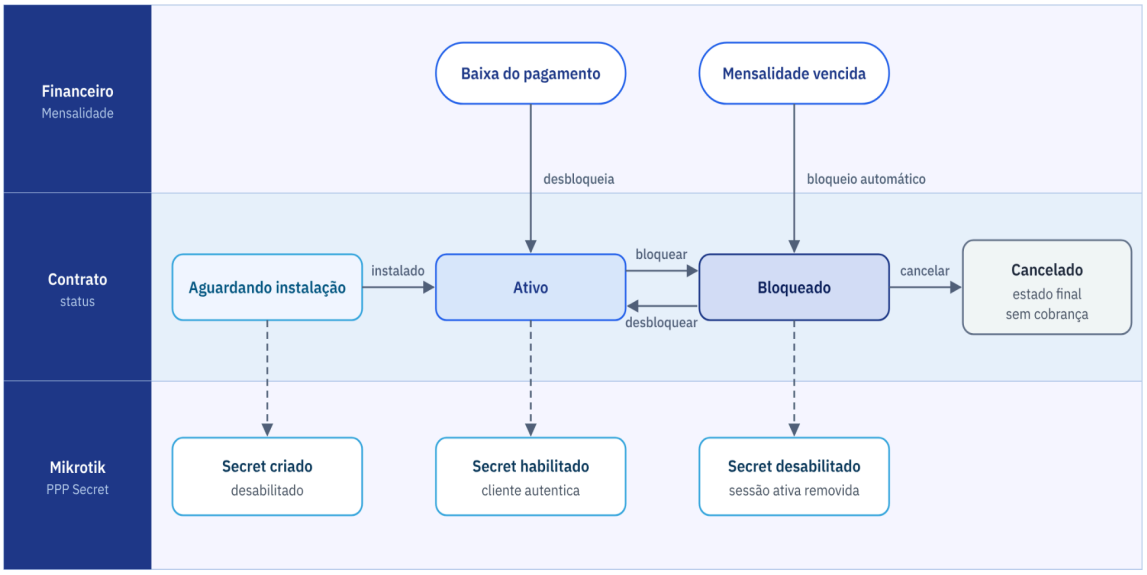
A comunicação com o RouterOS usa a REST API sobre HTTPS, que expõe leitura, criação, alteração e remoção de recursos por métodos HTTP (MikroTik, 2026b). A interface React aciona a FastAPI, que aplica as regras e envia as operações ao RouterOS; o WinBox serve apenas para conferência visual. A API usa um cliente dedicado, com tempos limite e tentativas configuráveis.

4.3.1 Ciclo de vida do contrato e provisionamento

O contrato tem cinco estados: aguardando instalação, ativo, bloqueado, cancelado e ausente (Figura 7). Um contrato novo nasce aguardando instalação: seu PPP Secret é criado desabilitado e nenhuma mensalidade é gerada até que a ação ativar

registre uma data de instalação não futura; a cobrança começa no primeiro vencimento igual ou posterior a ela, sem títulos retroativos. O bloqueio pode ser manual ou automático por inadimplência, com a origem registrada, e o desbloqueio automático por pagamento só ocorre quando o bloqueio decorreu de inadimplência. O estado ausente é a remoção lógica, que preserva o histórico, desabilita o Secret e encerra a sessão. No cadastro, a API verifica cliente e plano, grava o contrato como pendente, cifra a senha PPPoE, registra log de notificação e garante no RouterOS o PPP Profile do plano, com o limite de velocidade, e o PPP Secret do contrato, identificado pelo comentário *micronet:número*.

Figura 7 – Estados do contrato e transições

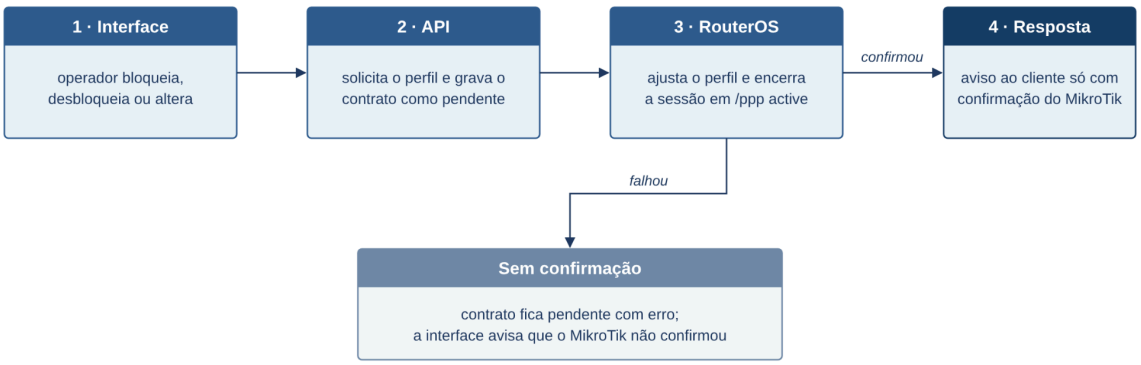


Fonte: elaborada pela autora (2026).

4.3.2 Bloqueio, desbloqueio, desconexão e sincronização

Bloquear desabilita o PPP Secret e encerra a sessão em /ppp active, impedindo nova autenticação; desbloquear habilita o Secret; desconectar encerra a sessão ativa, sem alterar o contrato nem o Secret. Nas ações que alteram o contrato, a mudança é gravada antes da comunicação com o equipamento e o resultado fica em *status sincronização*: toda ação marca o contrato como pendente; a confirmação do RouterOS o leva a sincronizado; requisições inválidas, como a integração desabilitada, mantêm pendente; as demais falhas resultam em erro, com o motivo registrado. Na interface, a ação só é apresentada como concluída quando o banco confirma o estado esperado e o campo é sincronizado; do contrário, um aviso informa que o MikroTik não confirmou (Figura 8).

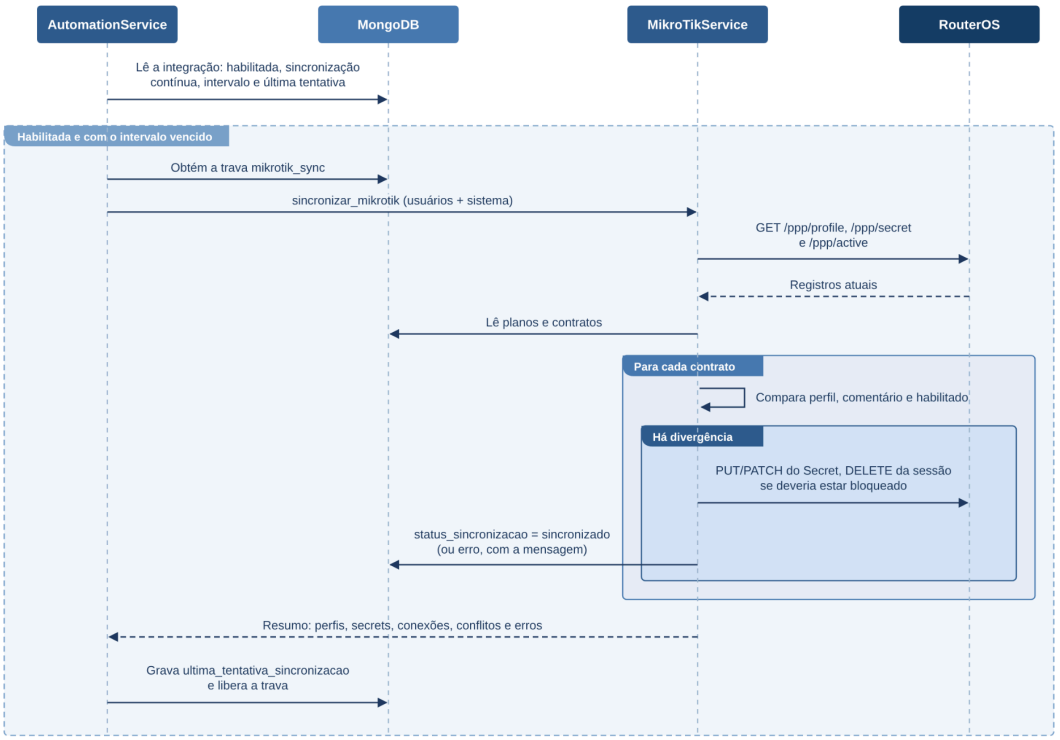
Figura 8 – Da ação do operador à confirmação do MikroTik



Fonte: elaborada pela autora (2026).

Além das ações, a API compara periodicamente os contratos com os PPP Secrets e as sessões do equipamento (Figura 9): cria perfis e secrets ausentes, ajusta perfil, comentário e estado habilitado ou desabilitado e encerra as sessões de contratos que deveriam estar bloqueados. Secrets não criados pelo MicroNET, identificados pelo comentário, não são alterados e contam como conflitos. A rotina também pode rodar em modo de simulação (*dry run*), que apenas calcula as divergências.

Figura 9 – Sincronização periódica entre o MicroNET e o MikroTik

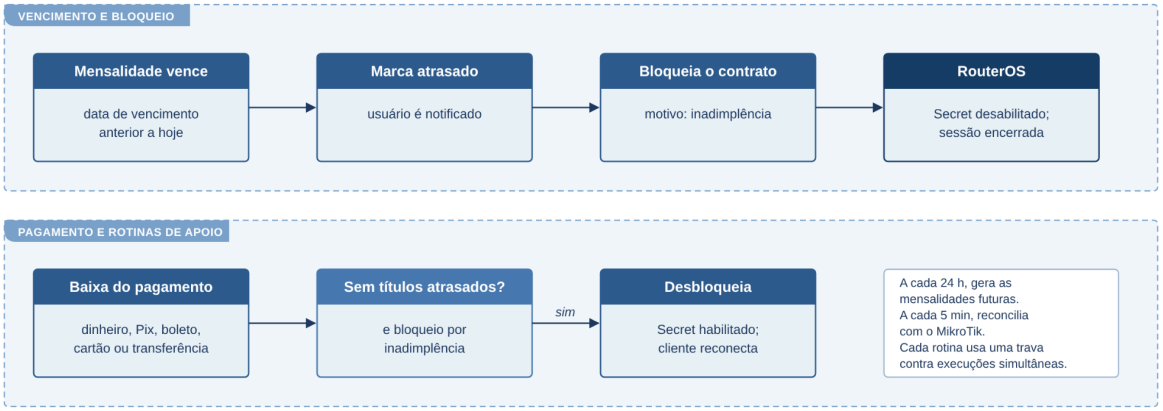


Fonte: elaborada pela autora (2026).

4.3.3 Processos automáticos, financeiro e rastreabilidade

A mensalidade assume os estados pendentes, atrasado, pago e cancelado, e a baixa registra a forma de pagamento. Um serviço de automação, iniciado com a API, verifica a cada 15 segundos três rotinas (Figura 10): a geração de mensalidades futuras, no máximo a cada 24 horas; o processamento de vencimentos, no máximo a cada 15 minutos; e a sincronização contínua com o MikroTik, em intervalo configurável (padrão de 5 minutos). Cada rotina obtém uma trava antes de executar, o que evita execuções simultâneas. No processamento de vencimentos, as mensalidades vencidas passam a ser atrasadas, o usuário é notificado e o contrato é bloqueado por inadimplência, exceto se estiver aguardando instalação ou cancelado; quando a baixa deixa o contrato sem títulos atrasados, ele é desbloqueado, se o bloqueio foi por inadimplência. As operações relevantes geram registros de auditoria, com nível, evento, usuário, IP e detalhes, que expiram automaticamente por índice TTL e são consultados na tela Registros.

Figura 10 – Rotinas automáticas e fluxo financeiro de bloqueio e desbloqueio



Fonte: elaborada pela autora (2026).

4.4 Publicação em nuvem e acesso ao roteador

A aplicação foi publicada na plataforma Light Cloud, em uma máquina virtual Ubuntu com Docker Compose e quatro serviços: a API, o servidor nginx (que serve a interface e encaminha as rotas /api e /minio), o MongoDB e o MinIO, acessíveis por HTTPS em micronet.app.fslab.dev (Docker, 2026; MinIO, 2025). Como o roteador do laboratório está em rede privada, a API na nuvem não o alcançava e o painel o exibia como offline. A solução foi um túnel SSH reverso, iniciado pela estação do laboratório, que expõe na máquina da nuvem o endereço 192.168.56.10, porta 8443,

encaminhada ao CHR. Assim, a API usa o mesmo endereço configurado e mantém a verificação TLS habilitada, validando o certificado do roteador por uma autoridade certificadora fixada, sem desativar a verificação. O túnel usa apenas conexões de saída (portas 22 e 443 liberadas na rede da instituição) e é iniciado automaticamente na inicialização da estação, de modo que, ao chegar à rede, o equipamento passa a constar como online.

4.5 Uso de inteligência artificial

O desenvolvimento contou com o apoio de ferramentas de inteligência artificial generativa, com uso e revisão da autora. O ChatGPT e o Codex foram empregados na revisão textual, no esclarecimento de conceitos e na análise e depuração de código; o GitHub Copilot, pontualmente, na implementação; e o Figma AI, na organização das interfaces. O Claude Code, da Anthropic, foi utilizado de forma mais ampla: na ampliação dos testes automatizados, na automação do túnel de acesso ao laboratório, no apoio à implementação e à correção de regras de negócio (como o estado aguardando instalação, o início da cobrança após a instalação e a derivação do status do cliente a partir dos contratos), em correções da integração com o MikroTik, do painel e do módulo financeiro, no redesenho visual da interface e na elaboração de figuras e de trechos de texto submetidos à revisão da autora. Todo o versionamento é de autoria própria, e a maior parte do sistema e dos testes unitários é anterior ao uso mais intenso dessas ferramentas. As sugestões foram analisadas, adaptadas e validadas pela autora, responsável pelas decisões metodológicas, pela implementação, pelos testes e pela versão final. As figuras elaboradas com apoio de ferramentas de IA estão identificadas como tal em sua fonte.

4.6 Tratamento de dados pessoais e LGPD

O MicroNET trata dados pessoais de assinantes — nome, CPF, endereço e informações contratuais e financeiras —, o que o enquadra na Lei nº 13.709/2018 (Lei Geral de Proteção de Dados Pessoais — LGPD). A base legal para esse tratamento é a execução de contrato e o cumprimento das obrigações a ele associadas (art. 7º, V), pois os dados são necessários à prestação e à cobrança do serviço de acesso à Internet.

O sistema aplica medidas técnicas alinhadas ao princípio da segurança (art. 46): as senhas dos operadores são armazenadas como hash (bcrypt); a credencial PPPoE do assinante, que precisa ser retransmitida ao roteador, é mantida de forma cifrada, com chave em variável de ambiente; a comunicação entre a aplicação, o navegador e o roteador ocorre sobre HTTPS com validação de certificado TLS; o acesso às funções é controlado por perfil; e as operações relevantes são registradas em auditoria. O princípio da minimização é observado ao coletar apenas os dados necessários ao cadastro, ao contrato e à cobrança.

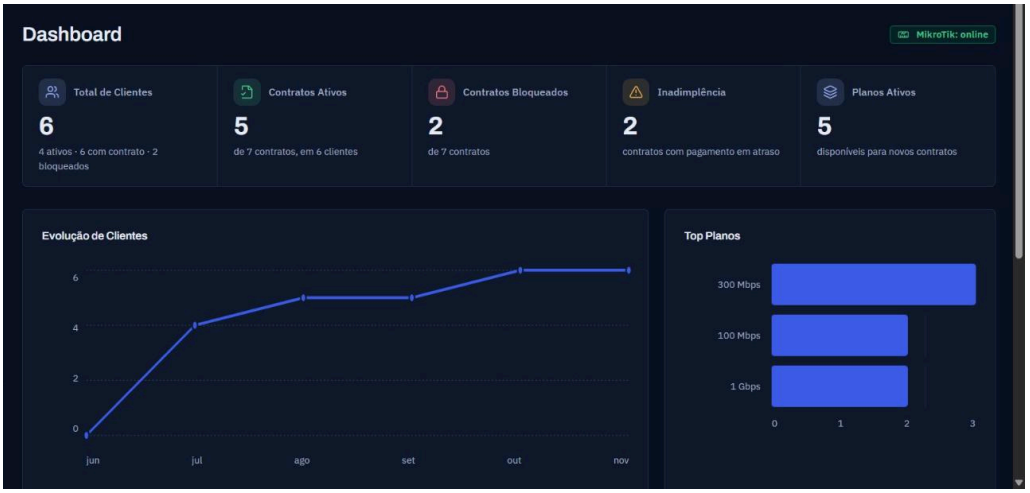
Como protótipo acadêmico, o sistema ainda não implementa alguns recursos previstos como trabalhos futuros de adequação à LGPD e de relacionamento com o assinante: a notificação ativa do cliente por e-mail — e, futuramente, por SMS ou aplicativos de mensagem — sobre faturas, vencimentos e avisos de bloqueio (atualmente as notificações do sistema são internas, destinadas ao operador); a anonimização de dados em relatórios; uma política formal de retenção e descarte; o atendimento aos direitos do titular (acesso, correção e eliminação); e a criptografia do banco de dados em repouso. A auditoria utiliza expiração automática por índice TTL (180 dias no código), prazo que deve ser revisto à luz da obrigação de manter histórico.

5 RESULTADOS E DISCUSSÃO

5.1 Sistema MicroNET e funcionalidades

O resultado é um sistema web que reúne gestão administrativa, financeira e controle de rede. Foram implementados autenticação e controle de acesso, gerenciamento de usuários, clientes, planos e contratos, controle financeiro, painel administrativo, configuração do MikroTik, gerenciamento de credenciais PPPoE, conexões ativas e as operações de bloqueio, desbloqueio e desconexão. A Figura 11 mostra o painel, com indicadores de clientes, contratos, inadimplência e o estado do MikroTik, e a Figura 12 apresenta clientes e contratos, com o vínculo entre cliente, plano, contrato e credencial PPPoE.

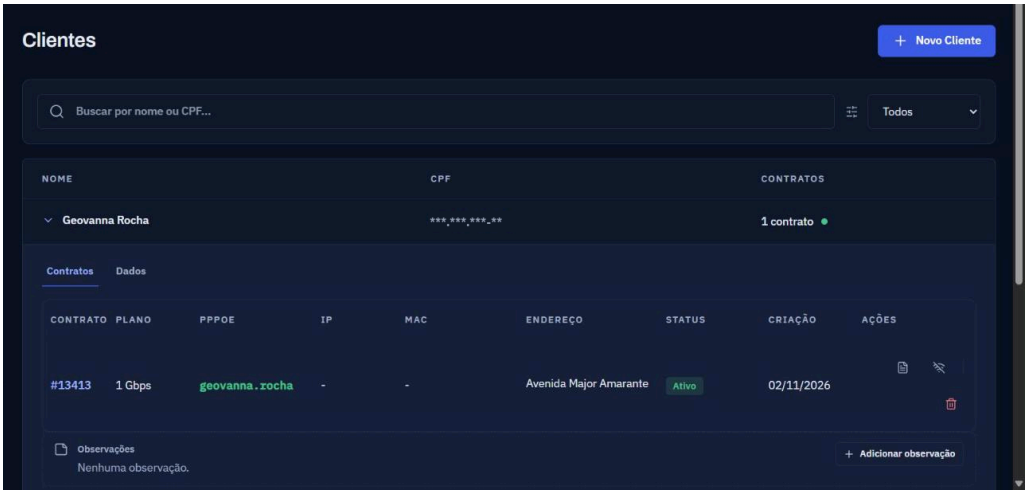
Figura 11 – Painel principal do MicroNET, com o MikroTik online



Fonte: elaborada pela autora (2026).

Após autenticar-se, o usuário visualiza o painel principal. A consulta começa pelos indicadores de clientes e contratos e pelos valores relacionados à inadimplência; em seguida, verifica o estado de comunicação do MikroTik. Esse percurso oferece uma visão inicial da situação administrativa e da disponibilidade da integração antes de acessar os módulos específicos.

Figura 12 – Gerenciamento de clientes e contratos (dados fictícios; CPFs ocultados)

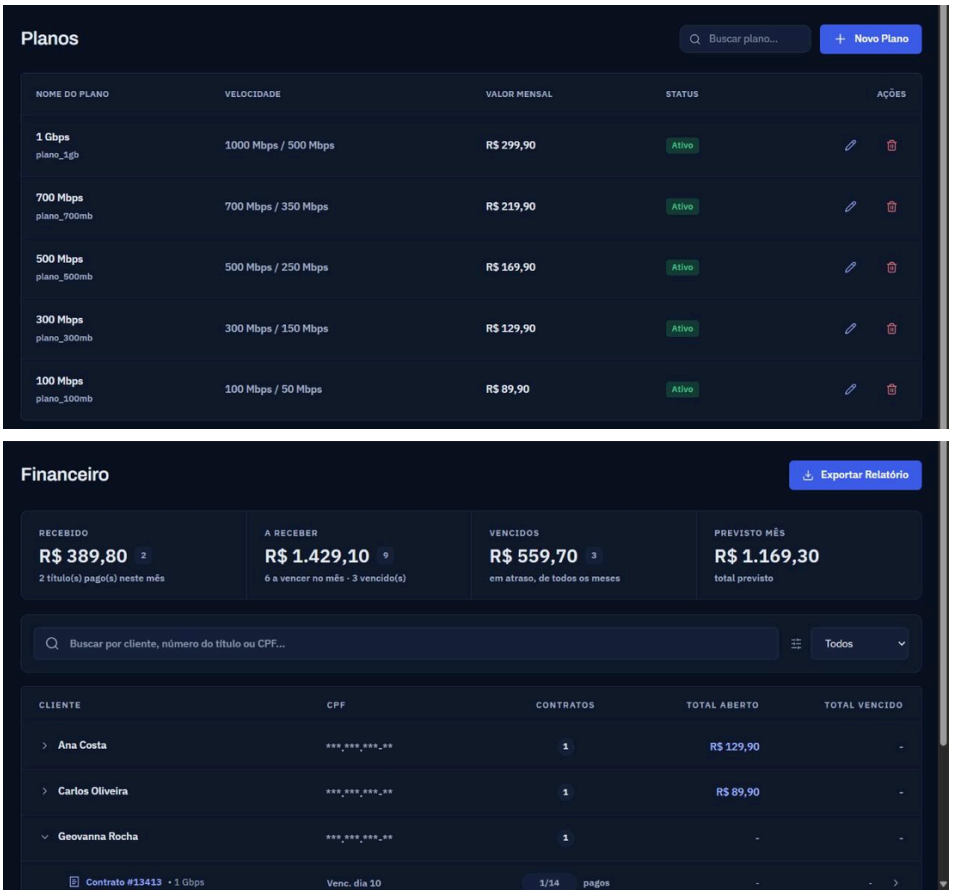


Fonte: elaborada pela autora (2026).

O gerenciamento começa pela consulta ou inclusão do cliente na área de Clientes. Depois, o usuário acessa os contratos desse cadastro, seleciona o plano de Internet e confere as informações do serviço e da credencial PPPoE vinculada. A tela reúne esses dados para que o relacionamento entre cliente, contrato e plano seja acompanhado em um único lugar. Os dados exibidos na figura são fictícios e os

CPFs foram ocultados. A Figura 13 apresenta os planos cadastrados (100 Mbps, 300 Mbps, 500 Mbps, 700 Mbps e 1 Gbps), cada um associado a um PPP Profile, e a tela financeira, com valores recebidos, a receber e vencidos e a lista de clientes com seus títulos.

Figura 13 – Planos de Internet e controle financeiro (dados fictícios; CPFs ocultados)



Fonte: elaborada pela autora (2026).

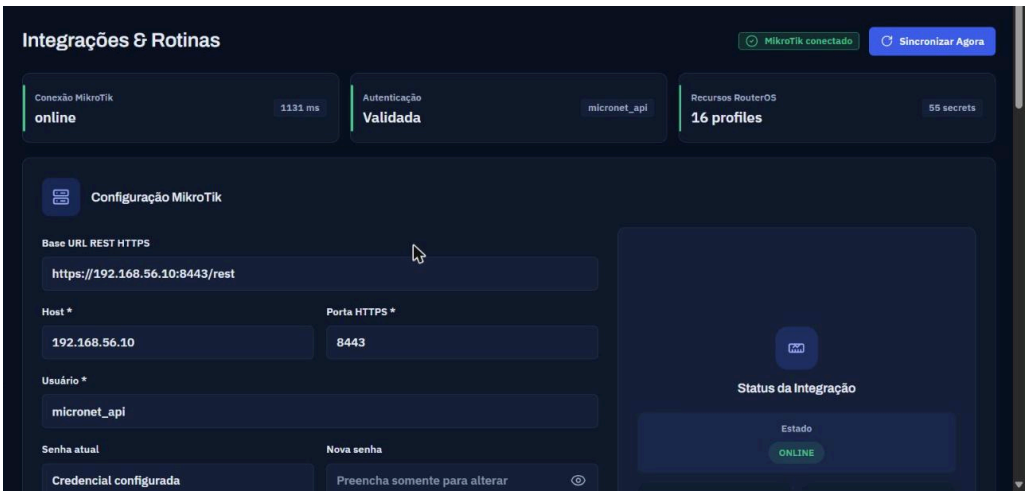
Na área de planos, o usuário consulta os planos disponíveis e identifica a velocidade configurada para cada serviço, associada ao respectivo PPP Profile no MikroTik. Na área financeira, acompanha os valores recebidos, a receber e vencidos e consulta os títulos vinculados aos clientes. Assim, a sequência de uso consiste em selecionar o plano para o serviço e, depois, acompanhar as cobranças geradas para o contrato correspondente.

5.2 Integração com o MikroTik

A integração foi validada pela RouterOS REST API sobre HTTPS: a aplicação autenticou com um usuário de serviço, consultou PPP Profiles e Secrets e executou as operações de escrita do provisionamento. A Figura 14 mostra o estado online da

integração no MicroNET publicado, com o roteador do laboratório alcançado pelo túnel. Nos testes com o cliente PPPoE, o contrato 13402, com o login joao.silva, foi provisionado no RouterOS e autenticado com o endereço 10.0.0.254. Ao bloquear, o Secret foi desabilitado e a sessão removida, impedindo nova autenticação; após o desbloqueio, a reconexão voltou a ser permitida; na desconexão, apenas a sessão em /ppp active foi removida. Duas execuções consecutivas da sincronização não criaram nem atualizaram perfis ou secrets, o que indica comportamento idempotente. Com a sessão ativa, o cliente obteve saída para a Internet pela tradução de endereços (NAT) do CHR na interface de WAN (ether1), o que corresponde ao acesso de ida e retorno sintetizado no Quadro 7.

Figura 14 – Estado online da integração entre o MicroNET e o MikroTik

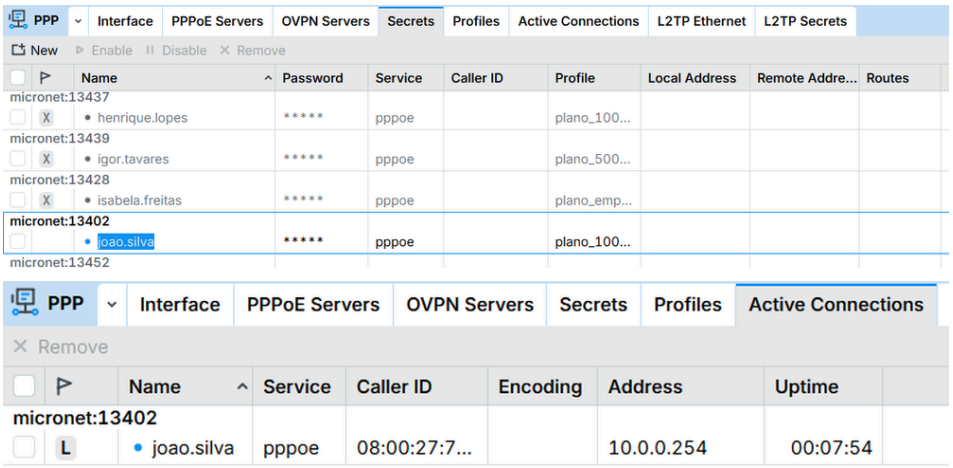


Fonte: elaborada pela autora (2026).

Para validar a integração, verifica-se inicialmente se o MikroTik aparece como online no painel do MicroNET. Esse estado indica que a API conseguiu alcançar o equipamento pelo túnel SSH reverso e estabelecer a comunicação HTTPS com validação do certificado TLS. Com o roteador acessível, a aplicação pode consultar os recursos PPP e executar as operações previstas para os contratos.

A Figura 15 apresenta, no WinBox, o PPP Secret do contrato 13402 habilitado e os demais desabilitados (marcados com X), e a sessão ativa do mesmo contrato em PPP > Active Connections. A Figura 16 mostra a mesma sessão no MicroNET, com contrato, login, plano, tempo de atividade, IP e MAC, e o cliente Ubuntu com a interface ppp0 em 10.0.0.254. A Figura 17 reúne os registros de auditoria, com os logins e a sincronização com o MikroTik executada pelo sistema a cada cinco minutos. O Quadro 7 sintetiza os testes práticos.

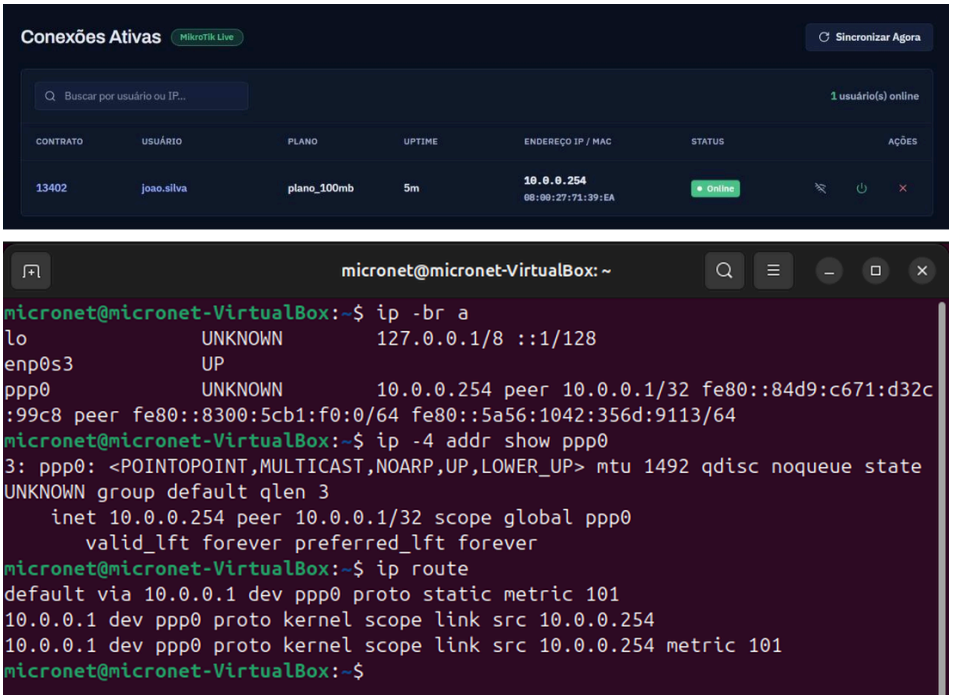
Figura 15 – PPP Secrets e sessão PPPoE ativa do contrato 13402 no WinBox



Fonte: elaborada pela autora (2026).

A conferência no roteador é feita pelo WinBox. Primeiro, acessa-se PPP > Secrets e localiza-se o usuário vinculado ao contrato 13402, verificando se o Secret está habilitado. Depois, em PPP > Active Connections, confirma-se a existência da sessão correspondente. Essa verificação permite observar separadamente o cadastro da credencial e a sessão efetivamente conectada.

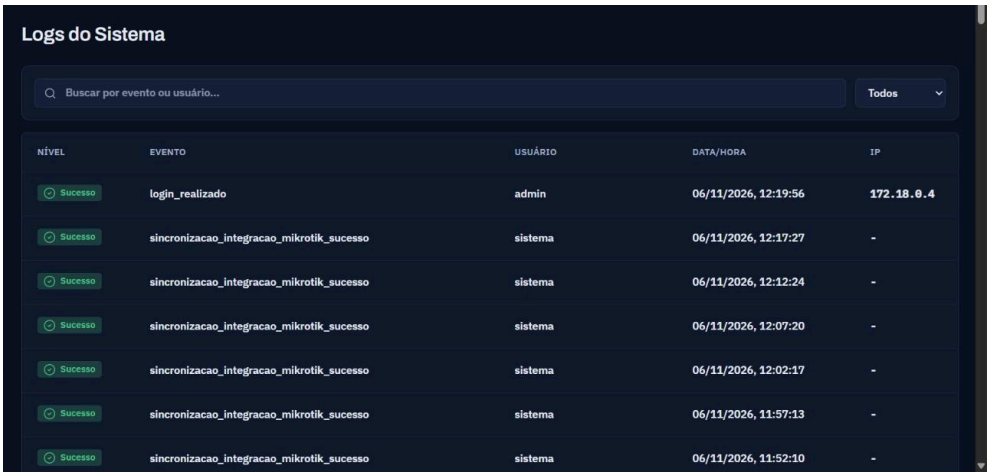
Figura 16 – Conexão ativa no MicroNET e cliente Ubuntu com a interface ppp0



Fonte: elaborada pela autora (2026).

Para conferir uma conexão em funcionamento, primeiro localiza-se o contrato na lista de conexões ativas do MicroNET e verificam-se o login PPPoE, o plano, o tempo de atividade, o endereço IP e o endereço MAC. Em seguida, no cliente Ubuntu, consulta-se a interface ppp0 para confirmar que recebeu o endereço 10.0.0.254. A comparação das duas telas permite verificar a correspondência entre a sessão exibida pela aplicação e a conexão estabelecida no cliente.

Figura 17 – Registros de auditoria das operações e da sincronização com o MikroTik



Fonte: elaborada pela autora (2026).

Para consultar o histórico de atividades, o operador acessa a área de registros de auditoria e verifica os eventos apresentados pelo sistema. Nessa tela, é possível acompanhar os registros de login e as execuções de sincronização com o MikroTik, realizadas automaticamente no intervalo configurado. Essa consulta permite conferir se as rotinas estão sendo executadas e manter o histórico das operações relevantes.

Quadro 7 – Síntese dos resultados dos testes práticos

Teste realizado	Resultado obtido
FastAPI ↔ RouterOS	OK – REST sobre HTTPS, com certificado validado
Site publicado ↔ roteador do laboratório	OK – túnel SSH reverso; status online
Autenticação PPPoE	OK – sessão autenticada (10.0.0.254)
Acesso à Internet	OK – ida e retorno validados
Bloqueio PPPoE	OK – Secret desabilitado; sessão encerrada
Desbloqueio PPPoE	OK – Secret habilitado; reconexão permitida
Desconexão	OK – sessão removida; Secret habilitado

Fonte: elaborada pela autora (2026).

5.3 Testes automatizados

Figura 18 – Execução dos testes automatizados no terminal: API (pytest, acima) e interface (Vitest, abaixo), com cobertura

```
PS C:\Users\kelly\api-micronet> .venv\Scripts\python -m pytest --cov=app -q


Name    Stmts   Miss Branch BrPart  Cover   Missing
-----------------------------------------------------
TOTAL    7001      0   1566      0   100%

Required test coverage of 100.0% reached. Total coverage: 100.00%

1103 passed, 3 skipped in 170.72s (0:02:50)


PS C:\Users\kelly\front-micronet> npx vitest run --coverage

 Test Files  27 passed (27)
      Tests  421 passed (421)

-------------------|---------|----------|---------|---------|-------------------
File               | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-------------------|---------|----------|---------|---------|-------------------
All files          |   99.31 |    93.36 |   96.61 |   99.31 |
```

Fonte: elaborada pela autora (2026).

Os testes da API (pytest) somam 1.103 casos aprovados e 3 ignorados, com cobertura de 100% das instruções e dos ramos, e os da interface (Vitest) somam 421 casos aprovados em 27 arquivos, com 99,31% de cobertura das instruções (Figura 18). Os testes de ponta a ponta (Cypress) percorrem as telas sobre a API em execução; nas ações que dependem do roteador — bloquear, desbloquear e desconectar —, as respostas do RouterOS são simuladas, de modo que validam o fluxo da interface e da API, e não a comunicação real com o equipamento. A execução abrangeu 17 arquivos e 275 testes, dos quais 274 foram aprovados e um permaneceu pendente por depender de uma conexão PPPoE ativa (Cypress, 2026). Durante essa etapa, também foram identificados e corrigidos problemas relacionados ao vencimento de mensalidades de teste e ao cálculo de datas na UTC. Cabe destacar que a cobertura de 100% das instruções e dos ramos mede o código da aplicação, e não a integração com o roteador: os três testes ignorados são justamente os que exigem o RouterOS real, simulado nos demais, de modo que a validação da comunicação efetiva com o equipamento se apoia nos testes práticos da Seção 5.2 e no Quadro 7.

5.4 Consistência dos dados e discussão

Durante o uso do sistema em laboratório, surgiram divergências entre os indicadores do painel, do financeiro e os cadastros: o status do cliente, mantido manualmente, diferia dos contratos; mensalidades de contratos removidos continuavam somadas; e contratos sem valor geravam cobranças de R$0,00. A correção adotou o contrato como fonte da verdade: o status do cliente é recalculado a cada mudança de contrato, contratos removidos saem das listas e rotinas, o total vencido considera atrasos de todos os meses e o login e o número de contratos removidos são liberados para reutilização. Os dados existentes foram regularizados por um procedimento executado antes em modo de simulação, com backup, e os valores passaram a coincidir.

Os resultados mostram que o MicroNET integra funções administrativas, financeiras e de controle de rede, com operações de bloqueio, desbloqueio e desconexão distintas e coerentes com as regras. Duas fontes de inconsistência do ambiente foram identificadas: duas bases sincronizando com o mesmo MikroTik e duas discagens PPPoE simultâneas no cliente; a estabilização exigiu uma única instância de sincronização e uma única discagem. Como limitações, a validação ocorreu no laboratório virtual e o acesso da nuvem ao roteador depende de a estação do laboratório estar ligada e conectada; cada contrato corresponde a um único login PPPoE; e a sincronização atua só sobre os Secrets criados pelo MicroNET.

Em relação aos trabalhos relacionados, enquanto Maria (2016) concentra-se no cadastro e Biz (2023), Rodrigues et al. (2025) e Souza et al. (2024) tratam de monitoramento, o MicroNET distingue-se por executar, a partir do contrato, operações de provisionamento e bloqueio no roteador, e não apenas observar a rede. Quanto aos requisitos, os funcionais foram atendidos e verificados pelos testes automatizados e práticos, assim como os não funcionais de segurança, integração e usabilidade; já os de desempenho e escalabilidade (RNF04, RNF05 e RNF10) não foram avaliados por ensaio específico e são tratados como limitação e trabalho futuro. A medição da taxa de transferência por plano ficou limitada pelo ambiente virtualizado e pelos recursos da máquina de testes, que não reproduzem as condições de um enlace de produção; assim, embora o limite de velocidade seja configurado no PPP Profile de cada plano, sua aferição em escala real permanece como trabalho futuro. Respondendo criticamente à pergunta de pesquisa,

confirma-se não só a viabilidade da integração, mas a automação, a partir do contrato, de operações de provisionamento, bloqueio e desbloqueio antes executadas manualmente no roteador; a principal limitação é que a validação se restringiu a um único concentrador em laboratório. Delimita-se, assim, o MicroNET como um protótipo funcional validado em laboratório, aplicável a provedores de pequeno porte com um único concentrador MikroTik. O sistema não emite documentos fiscais nem atende a obrigações regulatórias de informação; a baixa de pagamentos é manual, sem integração a meio de pagamento; o bloqueio por inadimplência ocorre sem carência configurável e sem aviso prévio ao assinante, e a desativação do Secret deixa o cliente sem página de aviso de débito; e não há central do assinante, ordens de serviço nem gestão de equipamentos. Do ponto de vista técnico, as rotinas automáticas executam no próprio processo da API, o que impede a execução de múltiplas instâncias simultâneas; o túnel SSH reverso expõe a interface de gerência do roteador na máquina em nuvem, o que exige restrição de escuta e privilégio mínimo do usuário de serviço no RouterOS; e a expiração automática dos registros de auditoria por índice TTL deve ser conciliada com a necessidade de preservar histórico.

6 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo desenvolver o MicroNET, uma aplicação web integrada ao MikroTik RouterOS para o gerenciamento de clientes, planos, contratos, cobranças e acessos à Internet por PPPoE. A questão que orientou a pesquisa foi como aproximar as informações administrativas das operações realizadas no roteador. Os resultados demonstram que essa integração não apenas é possível, mas automatiza, por meio do contrato — que reúne o cliente, o plano e as credenciais PPPoE —, operações de provisionamento, bloqueio e desbloqueio antes executadas manualmente no roteador, funcionando como referência para as ações administrativas e de rede. Dessa forma, o sistema centraliza operações que, sem essa integração, tenderiam a exigir consultas e alterações manuais em ferramentas distintas.

Os objetivos específicos foram alcançados com a implementação da API em FastAPI, da interface web em React, da persistência em MongoDB e da autenticação JWT, além da comunicação com o RouterOS pela REST API sobre HTTPS. Foram desenvolvidos o gerenciamento de clientes, planos, contratos e

financeiro, o provisionamento de perfis e credenciais PPPoE, a consulta de conexões ativas e os registros de auditoria. Também foram implementadas as operações de bloqueio, desbloqueio e desconexão com comportamentos distintos: o bloqueio desabilita o PPP Secret e encerra a sessão, o desbloqueio habilita novamente o Secret e a desconexão remove apenas a sessão ativa.

A validação reuniu 1.103 testes aprovados na API, 421 testes unitários aprovados no frontend e 274 testes de ponta a ponta aprovados, além dos testes práticos de autenticação e das operações no MikroTik. A publicação em nuvem e o túnel SSH reverso demonstraram a possibilidade de a aplicação alcançar o roteador do laboratório mantendo a validação TLS habilitada. Durante o uso, também foram identificadas inconsistências entre contratos, cadastros e indicadores financeiros; a adoção do contrato como referência e a regularização dos dados corrigiram essas divergências. Apesar dos resultados, a avaliação foi realizada em laboratório virtual, o acesso ao roteador depende de a estação do laboratório permanecer ligada e conectada, cada contrato contempla um único login PPPoE e a sincronização atua somente sobre Secrets criados pelo MicroNET. Como trabalhos futuros, propõem-se: integrar um meio de pagamento com baixa automática, como boleto ou Pix; implementar carência e aviso ao assinante antes do bloqueio, além de notificações por e-mail, SMS ou aplicativos de mensagem; adotar RADIUS e suporte a múltiplos concentradores; desenvolver uma central do assinante e funções de ordem de serviço; realizar testes de carga e de desempenho e uma avaliação com operadores; e validar o sistema em equipamento físico, estabelecendo uma conexão permanente entre a nuvem e a rede do roteador.

REFERÊNCIAS

BIZ, Júlio César Martins. Monitoramento de redes de computadores via API em sistemas operacionais RouterOS. Araranguá: Universidade Federal de Santa Catarina, 2023. Disponível em: https://repositorio.ufsc.br/handle/123456789/253694?show=full. Acesso em: 17 nov. 2026.

CYPRESS. Cypress Documentation. 2026. Disponível em: https://docs.cypress.io/. Acesso em: 17 nov. 2026.

DOCKER. What is Docker? Docker Docs, 2026. Disponível em: https://docs.docker.com/get-started/docker-overview/. Acesso em: 17 nov. 2026.

FASTAPI. FastAPI. 2026. Disponível em: https://fastapi.tiangolo.com. Acesso em: 17 nov. 2026.

FIELDING, Roy Thomas. Architectural styles and the design of network-based software architectures. Tese (Doctor of Philosophy in Information and Computer Science) – University of California, Irvine, 2000. Disponível em: https://ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf. Acesso em: 17 nov. 2026.

JONES, Michael; BRADLEY, John; SAKIMURA, Nat. JSON Web Token (JWT). RFC 7519. Internet Engineering Task Force, 2015. DOI: 10.17487/RFC7519. Disponível em: https://www.rfc-editor.org/info/rfc7519. Acesso em: 17 nov. 2026.

LESSI, Luis Felipe; OLIANI JUNIOR, Santo. Avaliação da usabilidade do ERP IXC: uma abordagem centrada no usuário. Revista Interface Tecnológica, Taquaritinga, v. 20, n. 1, p. 76-86, 2023. DOI: 10.31510/infla.201.1634. Disponível em: https://revista.fatectq.edu.br/interfacetecnologica/pt_BR/article/view/1634/879. Acesso em: 17 nov. 2026.

MAMAKOS, L. et al. A Method for Transmitting PPP Over Ethernet (PPPoE). RFC 2516. Internet Engineering Task Force, 1999. DOI: 10.17487/RFC2516. Disponível em: https://www.rfc-editor.org/info/rfc2516. Acesso em: 17 nov. 2026.

MARIA, Lucas dos Santos. Sistema gerenciador de clientes, equipamentos e circuitos para provedores de internet. 2016. Trabalho de Conclusão de Curso (Sistemas de Informação) – Faculdades Integradas de Taquara (FACCAT), Taquara, 2016. Disponível em: https://ti.faccat.br/wp-content/uploads/2022/08/Lucas-dos-Santos-Maria.pdf. Acesso em: 17 nov. 2026.

MATOZINHOS, Anderson Marin; RAMIRES, Guilherme. Gerenciando usuários com MikroTik User Manager. MUM Brasil, 2015. Disponível em: https://mum.mikrotik.com/presentations/BR15/presentation_2527_1447344805.pdf. Acesso em: 17 nov. 2026.

MIKROTIK. PPP AAA. RouterOS Documentation, 2026a. Disponível em: https://help.mikrotik.com/docs/spaces/ROS/pages/132350049/PPP+AAA. Acesso em: 17 nov. 2026.

MIKROTIK. REST API. RouterOS Manual, 2026b. Disponível em: https://manual.mikrotik.com/docs/developer-guides/rest-api/. Acesso em: 17 nov. 2026.

MINIO. MinIO Object Storage for Linux. 2025. Disponível em: https://min.io/docs/minio/linux/index.html. Acesso em: 17 nov. 2026.

MONGODB. What is MongoDB? An Overview of the Database. Database Manual, 2026. Disponível em: https://www.mongodb.com/docs/manual/. Acesso em: 17 nov. 2026.

REACT. Quick Start. React Documentation, 2026. Disponível em: https://react.dev/learn. Acesso em: 17 nov. 2026.

RODRIGUES, Diego Guimarães Viana; GONÇALVES, Lucas de Souza; OLIVEIRA, Lucas Moreno de; BARRETO, Plínio Rodrigues Rosa. Monitoramento de ativos de rede utilizando ferramenta de software de código aberto. Revista Destaques Acadêmicos, v. 17, n. 4, 2025. DOI: 10.22410/issn.2176-3070.17i4a2025.4291. Disponível em: https://www.univates.br/revistas/index.php/destaques/article/view/4291. Acesso em: 17 nov. 2026.

SILVA, Guilherme Barbosa Vieira da; MACHADO, João Mário da Silva; SOUZA, Kleber Ferreira de. Estudo de caso no ambiente corporativo utilizando MikroTik RouterBoard. 2023. Trabalho de Conclusão de Curso (Tecnologia em Redes de Computadores) – Centro Universitário Brasileiro (UNIBRA), Recife, 2023. Disponível em: https://www.grupounibra.com/repositorio/REDES/2023/estudo-de-caso-no-ambiente-corporativo-utilizando-mikrotik-routerboard.pdf. Acesso em: 17 nov. 2026.

SOUZA, João Paulo Cuba de; NUNES, Wallison Alves; SILVA, Samir Patrice Batista da; PINHEIRO, Sérgio Clayton Viana. Serviço para monitoramento de qualidade de link de internet. Revista Interdisciplinar da Meta, v. 3, n. 3, p. 1-53, 2024. Disponível em: https://ojs.meta.edu.br/index.php/rifm/article/view/33/27. Acesso em: 17 nov. 2026.

https://gitlab.fslab.dev/tcc-micronet/api-micronet

https://gitlab.fslab.dev/tcc-micronet/front-micronet