

TRUNKFLOW

VoIP Management System

DOCUMENTACAO TECNICA COMPLETA

Versao: 1.0.0

Data: 11/01/2026

Autor: TrunkFlow Team

Licenca: Proprietaria

SUMARIO

1. Visao Geral do Projeto
2. Arquitetura do Sistema
3. Stack Tecnologico
4. Estrutura do Banco de Dados
5. Backend - API FastAPI
6. Frontend - React
7. Integracao com Asterisk
8. Arquivos de Configuracao
9. Deploy e Infraestrutura
10. Endpoints da API
11. Fluxo de Chamadas
12. Seguranca
13. Manutencao e Troubleshooting

1. VISAO GERAL DO PROJETO

O TrunkFlow é um sistema completo de gerenciamento VoIP desenvolvido para administrar centrais telefônicas baseadas em Asterisk. O sistema permite gerenciar clientes, DIDs, provedores, gateways, rotas, ramais e tarifas através de uma interface web moderna e intuitiva.

1.1 Principais Funcionalidades

- * Dashboard com métricas em tempo real
- * Gerenciamento de Clientes (Trunk e Ramal)
- * Inventário de DIDs com alocação
- * Cadastro de Provedores de telefonia
- * Configuração de Gateways SIP
- * Rotas de saída com prioridade
- * Ramais para clientes tipo extensão
- * Tarifação com cálculo de margem
- * Relatórios de CDR e DIDs
- * Conferência entre números
- * Sincronização automática com Asterisk

2. ARQUITETURA DO SISTEMA

O sistema segue uma arquitetura de três camadas: Frontend (React), Backend (FastAPI) e Banco de Dados (PostgreSQL), com integração direta ao Asterisk via arquivos de configuração.

2.1 Componentes

Componente	Tecnologia	Porta	Função
Frontend	React + Vite	80/443	Interface do usuário
Backend	FastAPI + Uvicorn	8000	API REST
Banco de Dados	PostgreSQL 15+	5432	Persistência
Web Server	Nginx	80/443	Proxy reverso
PBX	Asterisk 20	5060	Telefonia SIP

3. STACK TECNOLOGICO

3.1 Backend

Tecnologia	Versao	Uso
Python	3.11+	Linguagem principal
FastAPI	0.104+	Framework web async
SQLAlchemy	2.0+	ORM async
Pydantic	2.0+	Validacao de dados
Uvicorn	0.24+	Servidor ASGI
python-jose	3.3+	JWT tokens
bcrypt	4.0+	Hash de senhas
asyncpg	0.29+	Driver PostgreSQL async

3.2 Frontend

Tecnologia	Versao	Uso
React	18+	Framework UI
Vite	5+	Build tool
TailwindCSS	3+	Estilizacao
React Query	5+	Gerenciamento de estado
Axios	1.6+	Cliente HTTP
Lucide React	0.29+	Icones

4. ESTRUTURA DO BANCO DE DADOS

O banco de dados PostgreSQL armazena todas as informações do sistema. Utiliza UUIDs como chaves primárias para melhor distribuição e segurança.

4.1 Tabelas Principais

Tabela	Descrição	Campos Principais
users	Usuários do sistema	id, username, password_hash, role
customers	Clientes (trunk/ramal)	id, code, name, type, trunk_ip
providers	Provedores de telefonia	id, name, type, cost_per_minute
gateways	Gateways SIP	id, name, ip_address, port, codecs
dids	Números DID	id, number, provider_id, gateway_id
customer_dids	Alocação DID-Cliente	customer_id, did_id, destination
routes	Rotas de saída	id, name, pattern, gateway_id, priority
extensions	Ramais	id, extension, secret, customer_id
tariffs	Tarifas	id, pattern, cost, price_per_minute
cdr	Registros de chamadas	id, src, dst, duration, billsec, price

5. BACKEND - API FASTAPI

5.1 Estrutura de Diretorios

```
backend/
| -- app/
| | -- __init__.py
| | -- main.py # Aplicacao FastAPI
| | -- api/ # Routers/Endpoints
| | | -- auth.py # Autenticacao
| | | -- customers.py # CRUD clientes
| | | -- dids.py # CRUD DIDs
| | | -- providers.py # CRUD provedores
| | | -- gateways.py # CRUD gateways
| | | -- routes.py # CRUD rotas
| | | -- extensions.py # CRUD ramais
| | | -- tariffs.py # CRUD tarifas
| | | -- reports.py # Relatorios
| | | -- dashboard.py # Estatisticas
| | -- core/
| | | -- config.py # Configuracoes
| | | -- database.py # Conexao DB
| | | -- security.py # JWT/Auth
| | -- models/ # Modelos SQLAlchemy
| | -- schemas/ # Schemas Pydantic
| | -- services/
| | -- asterisk.py # Integracao Asterisk
| -- requirements.txt
| -- .env
```

5.2 Autenticacao

O sistema utiliza JWT (JSON Web Tokens) para autenticacao. O token e gerado no login e deve ser enviado no header Authorization de todas as requisicoes protegidas. Expiracao: 24 horas | Algoritmo: HS256

7. INTEGRACAO COM ASTERISK

O TrunkFlow sincroniza automaticamente as configuracoes com o Asterisk atraves de arquivos de configuracao PJSIP e Dialplan.

7.1 Arquivos Gerados

Arquivo	Modulo	Conteudo
pjsip_gateways.conf	Gateways	Endpoints SIP dos gateways
pjsip_customer_trunks.conf	Clientes Trunk	Endpoints dos clientes
pjsip_extensions.conf	Ramais	Endpoints dos ramais
extensions_routes.conf	Rotas	Dialplan de saida
extensions_dids.conf	DIDs	Dialplan de entrada

7.2 Processo de Sincronizacao

1. Usuario cria/edita/exclui registro no painel
2. API salva no banco de dados
3. Servico AsteriskService gera arquivo de configuracao
4. Faz backup do arquivo anterior
5. Escreve novo arquivo em /etc/asterisk/
6. Executa reload no Asterisk (pjsip reload ou dialplan reload)

9. DEPLOY E INFRAESTRUTURA

9.1 Requisitos Minimos

Recurso	Mínimo	Recomendado
CPU	2 vCPUs	4+ vCPUs
RAM	2 GB	4+ GB
Disco	20 GB SSD	50+ GB SSD
SO	Debian 12 / Ubuntu 24	Debian 12

9.2 Capacidade Estimada

Servidor	Canais Simultaneos	CPS
2 vCPU / 2GB	60-80	10-15
4 vCPU / 8GB	150-180	25-30
8 vCPU / 16GB	300-400	50-60
32 vCPU / 64GB	1500-2000	150-200

* Baseado em codec G711 (alaw/ulaw) sem gravacao

10. ENDPOINTS DA API

Metodo	Endpoint	Descricao
POST	/api/v1/auth/login	Autenticacao
GET	/api/v1/dashboard/stats	Estatisticas
GET/POST	/api/v1/customers/	Listar/Criar clientes
PUT/DELETE	/api/v1/customers/{id}	Editar/Excluir cliente
GET/POST	/api/v1/dids/	Listar/Criar DIDs
POST	/api/v1/dids/{id}/allocate	Alocar DID
GET/POST	/api/v1/providers/	Listar/Criar provedores
GET/POST	/api/v1/gateways/	Listar/Criar gateways
GET/POST	/api/v1/routes/	Listar/Criar rotas
GET/POST	/api/v1/extensions/	Listar/Criar ramais
GET/POST	/api/v1/tariffs/	Listar/Criar tarifas
GET	/api/v1/reports/cdr	Relatorio CDR

12. SEGURANCA

12.1 Medidas Implementadas

- * Senhas armazenadas com hash bcrypt
- * Autenticacao via JWT com expiracao
- * Validacao de entrada com Pydantic
- * Queries parametrizadas (SQLAlchemy)
- * CORS configurado
- * Proxy reverso com Nginx

12.2 Recomendacoes Adicionais

- * Configurar HTTPS com certificado SSL
- * Usar firewall (iptables/ufw)
- * Limitar acesso SSH por IP
- * Habilitar fail2ban para Asterisk
- * Backup regular do banco de dados

13. MANUTENCAO E TROUBLESHOOTING

13.1 Comandos Uteis

```
# Ver status do servico  
systemctl status trunkflow  
  
# Ver logs do backend  
journalctl -u trunkflow -f  
  
# Ver endpoints PJSIP  
asterisk -rx "pjsip show endpoints"  
  
# Recarregar PJSIP  
asterisk -rx "pjsip reload"  
  
# Ver canais ativos  
asterisk -rx "core show channels"
```

13.2 Problemas Comuns

Problema	Solucao
500 Internal Server Error	journalctl -u trunkflow -n 50
Tela branca apos login	Verificar console do browser (F12)
Gateway nao aparece	Verificar pjsip_gateways.conf
Ramal nao registra	Verificar senha e pjsip reload

FIM DA DOCUMENTACAO

Documento gerado em 11/01/2026 13:33

TrunkFlow v1.0.0 - VoIP Management