# ALETHEIA

A verdade sobre o que você come

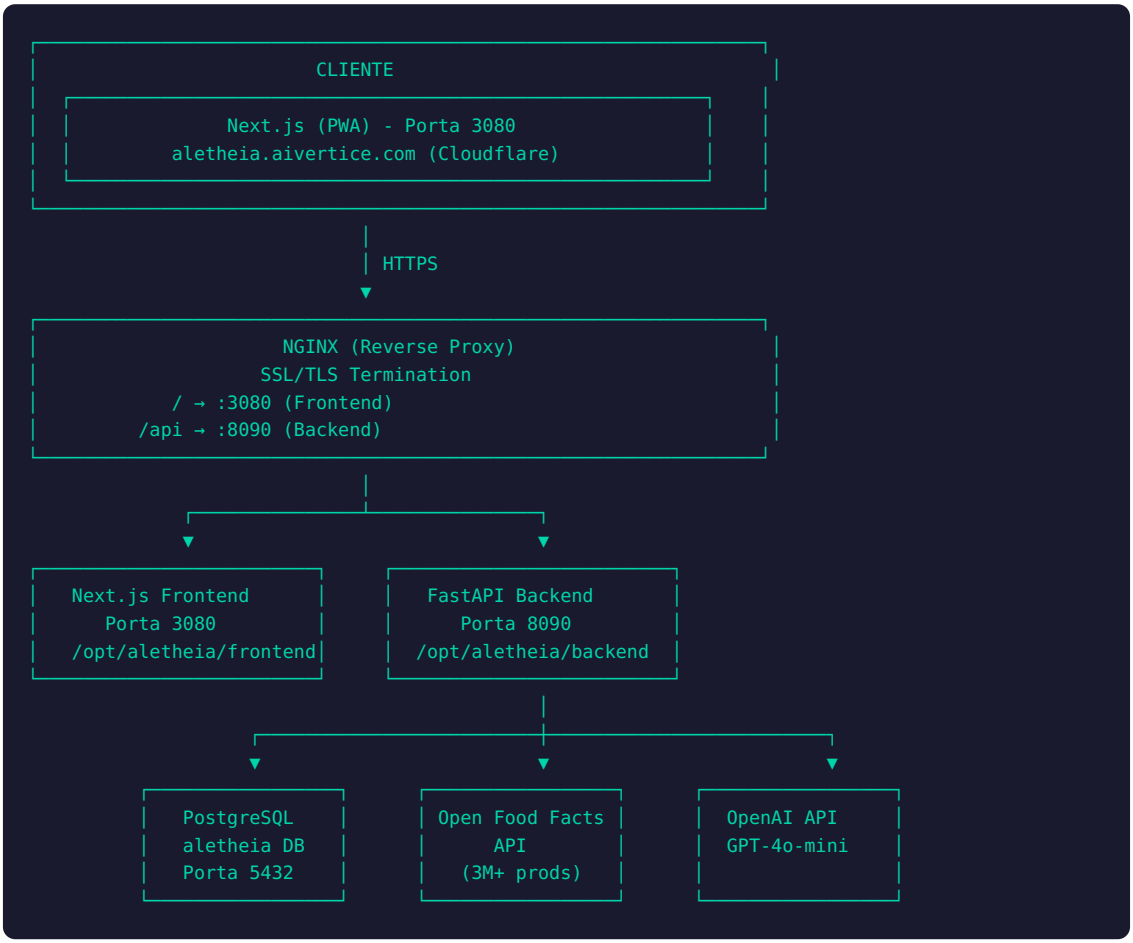
# Manual Técnico de Infraestrutura

Versão 1.0 — Fevereiro 2026

Kislanski Industries | AI Vertice

# ALETHEIA - Manual Técnico

## Arquitetura Geral




### Stack Tecnológico

| Camada | Tecnologia | Versão |
|---|---|---|
| Frontend | Next.js (React) | 14.x |
| Backend | FastAPI (Python) | 0.100+ |
| Banco de Dados | PostgreSQL | 15+ |
| IA | OpenAI GPT-4o-mini | - |
| Process Manager | PM2 | 5.x |
| Proxy/SSL | Nginx | 1.24+ |
| CDN/DNS | Cloudflare | - |

## Deploy e Infraestrutura

### Estrutura de Diretórios

```
/opt/aletheia/
├── backend/
│   ├── main.py
│   ├── requirements.txt
│   ├── .env
│   └── ...
├── frontend/
│   ├── package.json
│   ├── .env.local
│   ├── public/
│   │   └── sw.js (Service Worker)
│   └── ...
└── docs/
```

### PM2 - Gerenciamento de Processos

**Processos ativos:** - `aletheia-backend` - FastAPI (porta 8090) - `aletheia-frontend` - Next.js (porta 3080)

**Comandos úteis:**

```
# Status dos processos
pm2 status

# Logs em tempo real
pm2 logs aletheia-backend
pm2 logs aletheia-frontend

# Reiniciar processos
pm2 restart aletheia-backend
pm2 restart aletheia-frontend

# Reiniciar tudo
pm2 restart all

# Salvar configuração
pm2 save
```

### Nginx - Configuração

Arquivo: `/etc/nginx/sites-available/aletheia`

```
server {
    listen 443 ssl http2;
    server_name aletheia.aivertice.com;

    ssl_certificate /etc/letsencrypt/live/aletheia.aivertice.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/aletheia.aivertice.com/privkey.pem;

    # Frontend
    location / {
        proxy_pass http://127.0.0.1:3080;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection CONTENT_PLACEHOLDER#39;upgradeCONTENT_PLACEHOLDER#39;;
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }

    # Backend API
```

```
    location /api {
        proxy_pass http://127.0.0.1:8090;
        proxy_http_version 1.1;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}

server {
    listen 80;
    server_name aletheia.aivertice.com;
    return 301 https://$server_name$request_uri;
}
```

### SSL/TLS

- **Provedor**: Let's Encrypt (Certbot)
- **Renovação**: Automática via cron
- **CDN**: Cloudflare (SSL Full Strict)

```
# Renovar certificado manualmente
sudo certbot renew

# Verificar certificado
sudo certbot certificates
```

---

## 🗄 Banco de Dados

### Conexão

```
Host: localhost
Porta: 5432
Database: aletheia
User: aletheia
Password: Aletheia2026!
```

**Connection String:**

```
postgresql://aletheia:Aletheia2026!@localhost:5432/aletheia
```

### Tabelas Principais

**users**

```
CREATE TABLE users (
    id SERIAL PRIMARY KEY,
    email VARCHAR(255) UNIQUE NOT NULL,
    password_hash VARCHAR(255) NOT NULL,
    name VARCHAR(255),
    plan VARCHAR(50) DEFAULT CONTENT_PLACEHOLDER#39;freeCONTENT_PLACEHOLDER#39;,  -- CONTENT_PLACEH
    scans_today INTEGER DEFAULT 0,
    last_scan_date DATE,
    created_at TIMESTAMP DEFAULT NOW(),
    updated_at TIMESTAMP DEFAULT NOW()
);
```

**products**

```
CREATE TABLE products (
    id SERIAL PRIMARY KEY,
    barcode VARCHAR(50) UNIQUE NOT NULL,
    name VARCHAR(255),
    brand VARCHAR(255),
    categories TEXT,
    ingredients TEXT,
    nutrition_data JSONB,
    image_url TEXT,
    source VARCHAR(50),  -- CONTENT_PLACEHOLDER#39;openfoodfactsCONTENT_PLACEHOLDER#39;, CONTENT_PLA
    created_at TIMESTAMP DEFAULT NOW(),
    updated_at TIMESTAMP DEFAULT NOW()
);
```

**scans**

```
CREATE TABLE scans (
    id SERIAL PRIMARY KEY,
    user_id INTEGER REFERENCES users(id),
    product_id INTEGER REFERENCES products(id),
    barcode VARCHAR(50),
    score INTEGER,  -- 0-100
    analysis JSONB,  -- Análise completa da IA
    recipe TEXT,     -- Receita sugerida
    scanned_at TIMESTAMP DEFAULT NOW()
);
```

### Comandos Úteis

```
# Conectar ao banco
psql -U aletheia -d aletheia -h localhost

# Backup
pg_dump -U aletheia -d aletheia CONTENT_PLACEHOLDERgt; backup_$(date +%Y%m%d).sql

# Restore
psql -U aletheia -d aletheia CONTENT_PLACEHOLDERlt; backup_20260210.sql
```

---

## APIs e Endpoints

### Autenticação

**POST /api/auth/register**

Registra novo usuário.

**Request:**

```
{
    CONTENT_PLACEHOLDERquot;emailCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;usuario@email.co
    CONTENT_PLACEHOLDERquot;passwordCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;senha123CONTE
    CONTENT_PLACEHOLDERquot;nameCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;Nome do UsuárioCO
}
```

**Response (201):**

```
{
    CONTENT_PLACEHOLDERquot;idCONTENT_PLACEHOLDERquot;: 1,
    CONTENT_PLACEHOLDERquot;emailCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;usuario@email.co
    CONTENT_PLACEHOLDERquot;nameCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;Nome do UsuárioCO
    CONTENT_PLACEHOLDERquot;planCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;freeCONTENT_PLACE
    CONTENT_PLACEHOLDERquot;tokenCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;eyJhbGciOiJIUzI1
}
```

**POST /api/auth/login**

Autentica usuário existente.

**Request:**

```
{
    CONTENT_PLACEHOLDERquot;emailCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;usuario@email.co
    CONTENT_PLACEHOLDERquot;passwordCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;senha123CONTE
}
```

**Response (200):**

```
{
    CONTENT_PLACEHOLDERquot;tokenCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;eyJhbGciOiJIUzI1
    CONTENT_PLACEHOLDERquot;userCONTENT_PLACEHOLDERquot;: {
        CONTENT_PLACEHOLDERquot;idCONTENT_PLACEHOLDERquot;: 1,
        CONTENT_PLACEHOLDERquot;emailCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;usuario@emai
        CONTENT_PLACEHOLDERquot;nameCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;Nome do Usuár
        CONTENT_PLACEHOLDERquot;planCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;freeCONTENT_P
        CONTENT_PLACEHOLDERquot;scans_todayCONTENT_PLACEHOLDERquot;: 2
    }
}
```

## Scans

**POST /api/scan**

Analisa um produto pelo código de barras.

**Headers:**

```
Authorization: Bearer CONTENT_PLACEHOLDERlt;tokenCONTENT_PLACEHOLDERgt;
```

**Request:**

```
{
    CONTENT_PLACEHOLDERquot;barcodeCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;7891000100103C
}
```

**Response (200):**

```
{
    CONTENT_PLACEHOLDERquot;idCONTENT_PLACEHOLDERquot;: 123,
    CONTENT_PLACEHOLDERquot;productCONTENT_PLACEHOLDERquot;: {
        CONTENT_PLACEHOLDERquot;barcodeCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;789100010
        CONTENT_PLACEHOLDERquot;nameCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;Leite Condens
        CONTENT_PLACEHOLDERquot;brandCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;MoçaCONTENT_
        CONTENT_PLACEHOLDERquot;image_urlCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;https://
```

```
    },
    CONTENT_PLACEHOLDERquot;scoreCONTENT_PLACEHOLDERquot;: 25,
    CONTENT_PLACEHOLDERquot;classificationCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;Péssimo
    CONTENT_PLACEHOLDERquot;analysisCONTENT_PLACEHOLDERquot;: {
        CONTENT_PLACEHOLDERquot;summaryCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;Produto co
        CONTENT_PLACEHOLDERquot;positivesCONTENT_PLACEHOLDERquot;: [CONTENT_PLACEHOLDERquot;Fonte d
        CONTENT_PLACEHOLDERquot;negativesCONTENT_PLACEHOLDERquot;: [CONTENT_PLACEHOLDERquot;Alto te
        CONTENT_PLACEHOLDERquot;additivesCONTENT_PLACEHOLDERquot;: [],
        CONTENT_PLACEHOLDERquot;nutritionCONTENT_PLACEHOLDERquot;: {
            CONTENT_PLACEHOLDERquot;caloriesCONTENT_PLACEHOLDERquot;: 321,
            CONTENT_PLACEHOLDERquot;sugarCONTENT_PLACEHOLDERquot;: 55,
            CONTENT_PLACEHOLDERquot;sodiumCONTENT_PLACEHOLDERquot;: 128
        }
    },
    CONTENT_PLACEHOLDERquot;recipeCONTENT_PLACEHOLDERquot;: {
        CONTENT_PLACEHOLDERquot;titleCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;Leite conden
        CONTENT_PLACEHOLDERquot;ingredientsCONTENT_PLACEHOLDERquot;: [CONTENT_PLACEHOLDERquot;1 lit
        CONTENT_PLACEHOLDERquot;instructionsCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;...CO
    }
}
```

**Erros:** - `403` : Limite de scans atingido (plano free) - `404` : Produto não encontrado - `500` : Erro na análise

## Histórico

**GET /api/history**

Lista histórico de scans do usuário.

**Headers:**

```
Authorization: Bearer CONTENT_PLACEHOLDERlt;tokenCONTENT_PLACEHOLDERgt;
```

**Query Params:** - `limit` (opcional): Número de resultados (default: 20) - `offset` (opcional): Paginação

**Response (200):**

```
{
    CONTENT_PLACEHOLDERquot;totalCONTENT_PLACEHOLDERquot;: 45,
    CONTENT_PLACEHOLDERquot;scansCONTENT_PLACEHOLDERquot;: [
        {
            CONTENT_PLACEHOLDERquot;idCONTENT_PLACEHOLDERquot;: 123,
            CONTENT_PLACEHOLDERquot;barcodeCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;789100
            CONTENT_PLACEHOLDERquot;product_nameCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;L
            CONTENT_PLACEHOLDERquot;scoreCONTENT_PLACEHOLDERquot;: 25,
            CONTENT_PLACEHOLDERquot;scanned_atCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;202
        },
        ...
    ]
}
```

**GET /api/history/{id}**

Retorna detalhes de um scan específico.

**Response (200):**

```
{
    CONTENT_PLACEHOLDERquot;idCONTENT_PLACEHOLDERquot;: 123,
    CONTENT_PLACEHOLDERquot;productCONTENT_PLACEHOLDERquot;: { ... },
    CONTENT_PLACEHOLDERquot;scoreCONTENT_PLACEHOLDERquot;: 25,
    CONTENT_PLACEHOLDERquot;analysisCONTENT_PLACEHOLDERquot;: { ... },
    CONTENT_PLACEHOLDERquot;recipeCONTENT_PLACEHOLDERquot;: { ... },
    CONTENT_PLACEHOLDERquot;scanned_atCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;2026-02-10T
```

```
}
```

## Integrações

### Open Food Facts API

Base de dados aberta com mais de 3 milhões de produtos.

**Endpoint:**

```
https://world.openfoodfacts.org/api/v2/product/{barcode}.json
```

**Exemplo:**

```
import requests

def get_product(barcode: str):
    url = fCONTENT_PLACEHOLDERquot;https://world.openfoodfacts.org/api/v2/product/{barcode}.jsonCONT
    response = requests.get(url)
    if response.status_code == 200:
        data = response.json()
        if data.get(CONTENT_PLACEHOLDERquot;statusCONTENT_PLACEHOLDERquot;) == 1:
            return data[CONTENT_PLACEHOLDERquot;productCONTENT_PLACEHOLDERquot;]
    return None
```

### OpenAI GPT-4o-mini

Usamos o modelo GPT-4o-mini para análise inteligente dos ingredientes.

**Uso:**

```
from openai import OpenAI

client = OpenAI(api_key=os.getenv(CONTENT_PLACEHOLDERquot;OPENAI_API_KEYCONTENT_PLACEHOLDERquot;))

def analyze_product(product_data: dict) -CONTENT_PLACEHOLDERgt; dict:
    prompt = fCONTENT_PLACEHOLDERquot;CONTENT_PLACEHOLDERquot;CONTENT_PLACEHOLDERquot;
    Analise este produto alimentício e forneça:
    1. Score de saúde (0-100)
    2. Pontos positivos
    3. Pontos negativos
    4. Análise dos aditivos
    5. Sugestão de receita saudável alternativa

    Produto: {product_data[CONTENT_PLACEHOLDER#39;nameCONTENT_PLACEHOLDER#39;]}
    Ingredientes: {product_data[CONTENT_PLACEHOLDER#39;ingredientsCONTENT_PLACEHOLDER#39;]}
    Valores nutricionais: {product_data[CONTENT_PLACEHOLDER#39;nutritionCONTENT_PLACEHOLDER#39;]}
    CONTENT_PLACEHOLDERquot;CONTENT_PLACEHOLDERquot;CONTENT_PLACEHOLDERquot;

    response = client.chat.completions.create(
        model=CONTENT_PLACEHOLDERquot;gpt-4o-miniCONTENT_PLACEHOLDERquot;,
        messages=[{CONTENT_PLACEHOLDERquot;roleCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;us
        response_format={CONTENT_PLACEHOLDERquot;typeCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERqu
    )

    return json.loads(response.choices[0].message.content)
```

## PWA / Service Worker

O ALETHEIA é uma Progressive Web App (PWA), permitindo: - Instalação na home screen - Funcionamento parcial offline - Notificações push (futuro)

### Service Worker

Arquivo: `/opt/aletheia/frontend/public/sw.js`

```
const CACHE_NAME = CONTENT_PLACEHOLDER#39;aletheia-v1CONTENT_PLACEHOLDER#39;;
const urlsToCache = [
    CONTENT_PLACEHOLDER#39;/CONTENT_PLACEHOLDER#39;,
    CONTENT_PLACEHOLDER#39;/scanCONTENT_PLACEHOLDER#39;,
    CONTENT_PLACEHOLDER#39;/historyCONTENT_PLACEHOLDER#39;,
    CONTENT_PLACEHOLDER#39;/offline.htmlCONTENT_PLACEHOLDER#39;
];

self.addEventListener(CONTENT_PLACEHOLDER#39;installCONTENT_PLACEHOLDER#39;, (event) =CONTENT_PLACE
    event.waitUntil(
        caches.open(CACHE_NAME)
            .then((cache) =CONTENT_PLACEHOLDERgt; cache.addAll(urlsToCache))
    );
});

self.addEventListener(CONTENT_PLACEHOLDER#39;fetchCONTENT_PLACEHOLDER#39;, (event) =CONTENT_PLACEHO
    event.respondWith(
        caches.match(event.request)
            .then((response) =CONTENT_PLACEHOLDERgt; response || fetch(event.request))
    );
});
```

### Manifest

Arquivo: `/opt/aletheia/frontend/public/manifest.json`

```
{
    CONTENT_PLACEHOLDERquot;nameCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;ALETHEIACONTENT_P
    CONTENT_PLACEHOLDERquot;short_nameCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;ALETHEIACONT
    CONTENT_PLACEHOLDERquot;descriptionCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;Scanner de
    CONTENT_PLACEHOLDERquot;start_urlCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;/CONTENT_PLA
    CONTENT_PLACEHOLDERquot;displayCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;standaloneCONT
    CONTENT_PLACEHOLDERquot;background_colorCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;#ffff
    CONTENT_PLACEHOLDERquot;theme_colorCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;#22c55eCONT
    CONTENT_PLACEHOLDERquot;iconsCONTENT_PLACEHOLDERquot;: [
        {
            CONTENT_PLACEHOLDERquot;srcCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;/icon-192.
            CONTENT_PLACEHOLDERquot;sizesCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;192x192C
            CONTENT_PLACEHOLDERquot;typeCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;image/pngC
        },
        {
            CONTENT_PLACEHOLDERquot;srcCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;/icon-512.
            CONTENT_PLACEHOLDERquot;sizesCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;512x512C
            CONTENT_PLACEHOLDERquot;typeCONTENT_PLACEHOLDERquot;: CONTENT_PLACEHOLDERquot;image/pngC
        }
    ]
}
```

---

## ⚙ Variáveis de Ambiente

### Backend (.env)

```
# Database
DATABASE_URL=postgresql://aletheia:Aletheia2026!@localhost:5432/aletheia

# OpenAI
OPENAI_API_KEY=sk-...

# JWT
JWT_SECRET=sua-chave-secreta-muito-longa
JWT_ALGORITHM=HS256
JWT_EXPIRATION_HOURS=24

# App
APP_ENV=production
DEBUG=false
CORS_ORIGINS=https://aletheia.aivertice.com
```

### Frontend (.env.local)

```
NEXT_PUBLIC_API_URL=https://aletheia.aivertice.com/api
NEXT_PUBLIC_APP_NAME=ALETHEIA
```

---

## 📊 Monitoramento

### PM2

```
# Dashboard em tempo real
pm2 monit

# Status detalhado
pm2 show aletheia-backend

# Métricas
pm2 info aletheia-backend
```

### Logs

```
# Logs do backend
tail -f ~/.pm2/logs/aletheia-backend-out.log
tail -f ~/.pm2/logs/aletheia-backend-error.log

# Logs do frontend
tail -f ~/.pm2/logs/aletheia-frontend-out.log

# Logs do Nginx
tail -f /var/log/nginx/access.log
tail -f /var/log/nginx/error.log

# Logs do PostgreSQL
tail -f /var/log/postgresql/postgresql-15-main.log
```

---

## Backup e Manutenção

### Backup Automático (Cron)

```
# Editar crontab
crontab -e

# Adicionar backup diário às 3h
0 3 * * * pg_dump -U aletheia -d aletheia CONTENT_PLACEHOLDERgt; /opt/aletheia/backups/backup_$(dat

# Limpar backups antigos (manter últimos 30 dias)
0 4 * * * find /opt/aletheia/backups -name CONTENT_PLACEHOLDERquot;*.sqlCONTENT_PLACEHOLDERquot; -m
```

### Manutenção do Banco

```
-- Vacuum e análise (rodar semanalmente)
VACUUM ANALYZE;

-- Verificar tamanho das tabelas
SELECT
    relname as table,
    pg_size_pretty(pg_total_relation_size(relid)) as size
FROM pg_catalog.pg_statio_user_tables
ORDER BY pg_total_relation_size(relid) DESC;
```

### Atualizações

```
# Backend
cd /opt/aletheia/backend
git pull
pip install -r requirements.txt
pm2 restart aletheia-backend

# Frontend
cd /opt/aletheia/frontend
git pull
npm install
npm run build
pm2 restart aletheia-frontend
```

---

## Troubleshooting

### Problema: Backend não inicia

```
# Verificar logs
pm2 logs aletheia-backend --lines 50

# Verificar porta em uso
lsof -i :8090

# Testar manualmente
cd /opt/aletheia/backend
python -m uvicorn main:app --host 0.0.0.0 --port 8090
```

### Problema: Erro de conexão com banco

```
# Verificar se PostgreSQL está rodando
```

```
sudo systemctl status postgresql

# Testar conexão
psql -U aletheia -d aletheia -h localhost -c CONTENT_PLACEHOLDERquot;SELECT 1CONTENT_PLACEHOLDERquo
```

### Problema: Certificado SSL expirado

```
# Renovar
sudo certbot renew

# Reiniciar Nginx
sudo systemctl restart nginx
```

### Problema: Open Food Facts não responde

```
# Testar API diretamente
curl CONTENT_PLACEHOLDERquot;https://world.openfoodfacts.org/api/v2/product/7891000100103.jsonCONTE

# Verificar rate limiting (máx 100 req/min)
```

## Contatos Técnicos

- **Infraestrutura**: infra@aivertice.com
- **Desenvolvimento**: dev@aivertice.com

---

Última atualização: Fevereiro 2026