Projeto: Integração Snipe-IT ⇄ Active Directory (LDAPS)

Visão Geral

Este projeto implementa uma **integração bidirecional automatizada** entre o **Snipe-IT** e o **Active Directory**, permitindo sincronização de usuários em ambas as direções através de um **serviço Python (FastAPI)**.

O serviço comunica-se via **LDAPS (porta 636)** com o **Controlador de Domínio**, possibilitando:

- Criação, atualização e desativação de usuários
- Sincronização periódica automática
- Sincronização manual via webhooks
- Rate limiting para proteger as APIs

Todo o fluxo ocorre **dentro da rede local (LAN)**, com opção de integração com **Microsoft Entra ID** (Microsoft 365) através do **Cloud Sync**.

Funcionalidades

Sincronização Bidirecional

- **LDAP → Snipe-IT**: Replica usuários do Active Directory para o Snipe-IT
- **Snipe-IT → LDAP**: Replica usuários do Snipe-IT para o Active Directory

Operações Suportadas

- **Criação de usuários**: Cria usuários automaticamente em ambos os sistemas
- **Atualização de dados**: Sincroniza alterações de nome, email, telefone, departamento, cargo
- **Desativação**: Desativa usuários que não existem mais no sistema de origem
- **Habilitação**: Reativa usuários desativados quando necessário

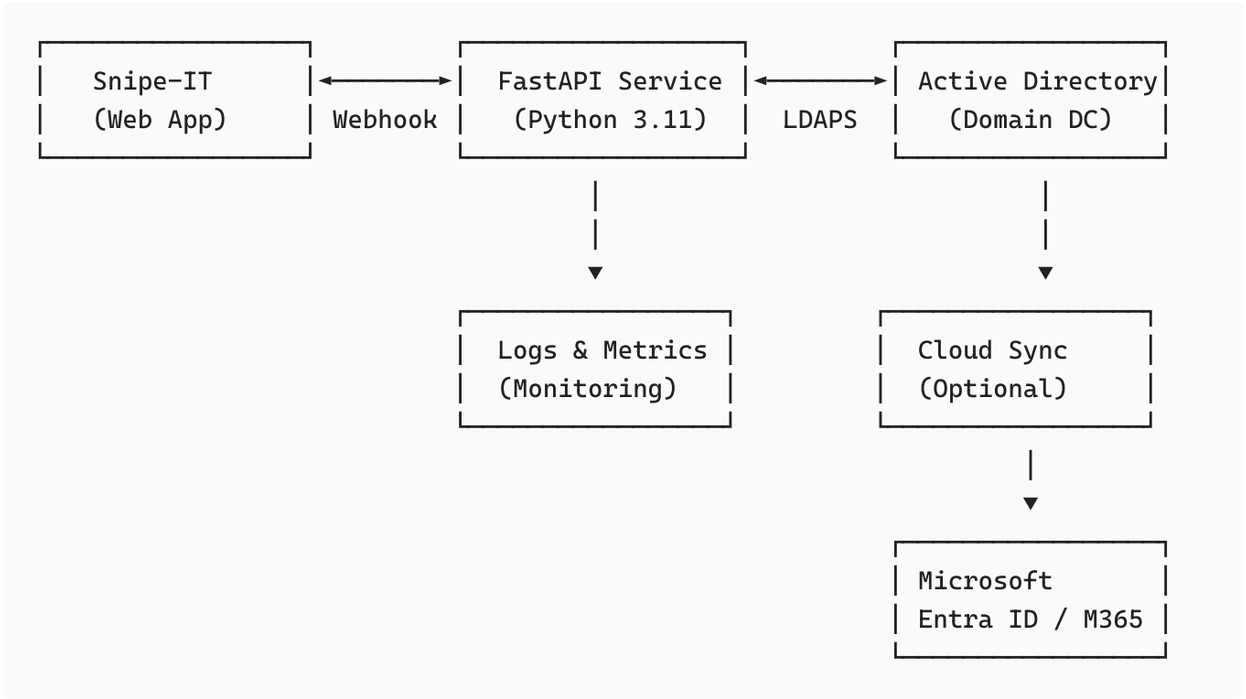
Sincronização Automática

- Sincronização periódica configurável (padrão: 15 minutos)
- Execução na inicialização da aplicação
- Logs detalhados de cada operação

Segurança

- Autenticação via webhook secret
- Suporte a SSL/TLS para LDAPS
- Rate limiting para proteção das APIs
- Validação de dados com Pydantic

Arquitetura



Fluxo de Sincronização

LDAP → Snipe-IT

1. LDAPService busca todos os usuários do AD
2. SnipeITService busca todos os usuários do Snipe-IT
3. SyncService compara os dados e identifica:
   - Usuários novos (existem no LDAP, não no Snipe-IT)
   - Usuários para atualizar (dados diferentes)
   - Usuários para desativar (não existem mais no LDAP)
4. Executa as operações necessárias

Snipe-IT → LDAP

1. SnipeITService busca todos os usuários do Snipe-IT
2. LDAPService busca todos os usuários do AD
3. SyncService compara os dados e identifica:
   - Usuários novos (existem no Snipe-IT, não no LDAP)
   - Usuários para atualizar (dados diferentes)
   - Usuários para desativar (não existem mais no Snipe-IT)
4. Executa as operações necessárias no AD

Estrutura do Projeto

```
WebhookSnipeIT_AD/
├── app/
│   ├── __init__.py
│   ├── main.py                 # Aplicação FastAPI principal
│   ├── config.py               # Configurações e variáveis de ambiente
│   │
│   ├── models/
│   │   ├── __init__.py
│   │   └── user.py             # Modelos Pydantic (LDAPUser, SnipeITUser,
SyncResult)
│   │
```

```
│   ├── routes/
│   │   ├── __init__.py
│   │   └── routes.py           # Endpoints da API (webhooks e health check)
│   │
│   ├── services/
│   │   ├── __init__.py
│   │   ├── ldap_service.py     # Serviço de integração com LDAP/AD
│   │   ├── snipeit_service.py  # Serviço de integração com Snipe-IT
│   │   └── sync_service.py     # Lógica de sincronização bidirecional
│   │
│   └── utils/
│       ├── __init__.py
│       └── rate_limiter.py     # Rate limiter para proteção de APIs
│
├── logs/                       # Diretório de logs (gerado automaticamente)
├── docker-compose.yml          # Configuração Docker Compose
├── Dockerfile                  # Imagem Docker da aplicação
├── requirements.txt            # Dependências Python
├── .env                        # Variáveis de ambiente (não versionado)
├── README.md                   # Documentação principal
└── DOCUMENTATION.md            # Este arquivo (documentação detalhada)
```

Tecnologias Utilizadas

Backend

- **Python 3.11**: Linguagem de programação
- **FastAPI**: Framework web moderno e rápido
- **Uvicorn**: Servidor ASGI de alta performance
- **Pydantic**: Validação de dados e configurações

Integração

- **ldap3**: Cliente LDAP/AD para Python
- **requests**: Cliente HTTP para API do Snipe-IT
- **backoff**: Retry automático com exponential backoff

Agendamento & Monitoramento

- **APScheduler**: Agendamento de tarefas periódicas
- **logging**: Sistema de logs do Python

Rate Limiting

- **Custom RateLimiter**: Limitador de taxa customizado
- **pyrate-limiter**: Biblioteca auxiliar de rate limiting

Container

- **Docker**: Containerização da aplicação
- **Docker Compose**: Orquestração de containers

Configuração

1. Variáveis de Ambiente

Crie um arquivo `.env` na raiz do projeto com as seguintes variáveis:

```
# LDAP/Active Directory Configuration
LDAP_SERVER=dc01.dominio.local
LDAP_PORT=636
LDAP_USE_SSL=true
LDAP_BIND_DN=CN=ServiceAccount,OU=ServiceAccounts,DC=dominio,DC=local
LDAP_BIND_PASSWORD=SenhaSegura123!
LDAP_BASE_DN=DC=dominio,DC=local
LDAP_USER_FILTER=(&(objectClass=user)(objectCategory=person))

# LDAP User Creation (Opcional)
LDAP_CREATE_USER_OU=OU=Colaboradores,DC=dominio,DC=local
LDAP_DEFAULT_PASSWORD=Senha@Padrao2024

# Snipe-IT Configuration
SNIPEIT_URL=https://snipeit.dominio.local
SNIPEIT_API_KEY=eyJ0eXAiOiJKV1QiLCJhbGc...
SNIPEIT_COMPANY_ID=1
SNIPEIT_VERIFY_SSL=true
SNIPEIT_DEACTIVATED_STATUS_ID=2

# Rate Limiting
CALLS_PER_MINUTE=55
ONE_MINUTE=60

# Webhook Configuration
WEBHOOK_SECRET=seu-secret-super-seguro-aqui
SYNC_INTERVAL_MINUTES=15
DRY_RUN=false

# Application Settings
APP_NAME=Snipe-IT LDAP Sync
APP_VERSION=1.0.0
LOG_LEVEL=INFO
```

2. Permissões no Active Directory

A conta de serviço configurada em `LDAP_BIND_DN` precisa das seguintes permissões:

Para Leitura (mínimo)

- Permissão de **Ler** no container base (BASE_DN)
- Permissão de **Listar conteúdo** no container base

Para Criação/Atualização de Usuários

- Permissão de **Criar objetos de usuário** na OU especificada
- Permissão de **Modificar** propriedades dos usuários
- Permissão de **Redefinir senha** (para criar usuários com senha)

Para Desativação de Usuários

- Permissão de **Modificar** o atributo `userAccountControl`

3. Configuração no Snipe-IT

1. Acesse **Settings** → **API**
2. Clique em **Create New Token**
3. Dê um nome descritivo (ex: "LDAP Sync Service")
4. Copie o token gerado e configure em `SNIPEIT_API_KEY`

Instalação e Execução

Opção 1: Docker (Recomendado)

```
# 1. Clone o repositório
git clone https://github.com/MoonAmon/WebhookSnipeIT_AD.git
cd WebhookSnipeIT_AD

# 2. Configure o arquivo .env
cp .env.example .env
nano .env

# 3. Inicie o container
docker-compose up -d

# 4. Verifique os logs
docker-compose logs -f
```

Opção 2: Ambiente Local

```
# 1. Clone o repositório
git clone https://github.com/MoonAmon/WebhookSnipeIT_AD.git
cd WebhookSnipeIT_AD

# 2. Crie um ambiente virtual
python -m venv venv
source venv/bin/activate  # Linux/Mac
# ou
.\venv\Scripts\activate  # Windows

# 3. Instale as dependências
pip install -r requirements.txt

# 4. Configure o .env
cp .env.example .env
nano .env

# 5. Execute a aplicação
uvicorn app.main:app --host 0.0.0.0 --port 8000 --reload
```

Endpoints da API

Root

```
GET /
```

Retorna informações básicas da aplicação e links para outros endpoints.

Resposta:

```
{
  "app": "Snipe-IT LDAP Sync",
  "version": "1.0.0",
  "status": "running",
  "timestamp": "2024-11-15T10:30:00.000Z",
  "endpoints": {
    "health": "/health",
    "sync_manual": "/webhook/sync",
    "sync_report": "/webhook/sync/report",
    "last_sync": "/status"
  }
}
```

Health Check

```
GET /webhook/health
```

Verifica conectividade com o LDAP e status da aplicação.

Resposta de sucesso:

```
{
  "status": "healthy",
  "ldap": "connected",
  "timestamp": "2024-11-15T10:30:00.000Z"
}
```

Resposta de erro:

```
{
  "detail": "Service unhealthy: Connection refused"
}
```

Status

```
GET /status
```

Retorna status detalhado da aplicação, incluindo informações sobre a última sincronização e próxima execução agendada.

Resposta:

```
{
  "app_status": "running",
  "ldap_server": "dc01.dominio.local:636",
  "snipeit_server": "https://snipeit.dominio.local",
  "sync_interval_minutes": 15,
  "scheduler_running": true,
  "next_scheduled_sync": "2024-11-15T10:45:00.000Z",
  "last_sync": {
    "timestamp": "2024-11-15T10:30:00.000Z",
    "status": "success",
    "result": {
```

```
      "total_ldap_users": 150,
      "users_created": 5,
      "users_updated": 12,
      "users_deactivated": 2,
      "errors": []
    }
  },
  "current_timestamp": "2024-11-15T10:32:00.000Z"
}
```

Info

```
GET /info
```

Retorna informações de configuração da aplicação (dados não sensíveis).

Resposta:

```
{
  "application": {
    "name": "Snipe-IT LDAP Sync",
    "version": "1.0.0",
    "log_level": "INFO"
  },
  "ldap": {
    "server": "dc01.dominio.local",
    "port": 636,
    "use_ssl": true,
    "bind_dn": "CN=ServiceAccount,OU=ServiceAccounts,DC=dominio,DC=local"
  },
  "snipeit": {
    "url": "https://snipeit.dominio.local",
    "company_id": 1
  },
  "sync": {
    "interval_minutes": 15,
    "scheduler_running": true
  }
}
```

🔁 Sincronização Manual: LDAP → Snipe-IT

```
POST /webhook/to/snipeit?dry_run=false
Headers:
  x-webhook-secret: seu-secret-aqui
```

Sincroniza usuários do Active Directory para o Snipe-IT.

Parâmetros:

- `dry_run` (bool, opcional): Se `true`, apenas simula sem fazer alterações

Resposta:

```
{
  "total_ldap_users": 150,
  "user_created": 5,
```

```
  "user_updated": 12,
  "user_skipped": 0,
  "user_deactivated": 2,
  "users_created_in_ad": 0,
  "errors": [],
  "timestamp": "2024-11-15T10:30:00.000Z"
}
```

Sincronização Manual: Snipe-IT → LDAP

```
POST /webhook/to/ldap?dry_run=false
Headers:
  x-webhook-secret: seu-secret-aqui
```

Sincroniza usuários do Snipe-IT para o Active Directory.

Parâmetros:

- `dry_run` (bool, opcional): Se `true`, apenas simula sem fazer alterações

Resposta:

```
{
  "status": "success",
  "direction": "Snipe-IT -> LDAP",
  "dry_run": false,
  "total_snipeit_users": 145,
  "total_ldap_users": 150,
  "users_created": 3,
  "users_updated": 8,
  "users_deactivated": 1,
  "errors": []
}
```

Relatório de Sincronização

```
GET /webhook/sync/report
Headers:
  x-webhook-secret: seu-secret-aqui
```

Gera um relatório detalhado de pré-sincronização sem fazer alterações.

Resposta:

```
{
  "ldap_users": [
    {
      "username": "john.doe",
      "email": "john.doe@dominio.local",
      "name": "John Doe",
      "enabled": true
    }
  ],
  "snipeit_users": [
    {
      "username": "john.doe",
      "email": "john.doe@dominio.local",
```

```
      "name": "John Doe",
      "activated": true
    }
  ],
  "matches": [
    {
      "ldap_username": "john.doe",
      "snipeit_username": "john.doe",
      "match_type": "username"
    }
  ],
  "new_users": [
    {
      "username": "jane.smith",
      "email": "jane.smith@dominio.local",
      "name": "Jane Smith"
    }
  ],
  "to_deactivate": [],
  "conflicts": []
}
```

Modelos de Dados

LDAPUser

Representa um usuário do Active Directory.

```
class LDAPUser(BaseModel):
    username: str                          # sAMAccountName
    first_name: Optional[str] = None       # givenName
    last_name: Optional[str] = None        # sn
    display_name: Optional[str] = None     # displayName
    email: Optional[EmailStr] = None       # mail
    department: Optional[str] = None       # department
    title: Optional[str] = None            # title
    phone_number: Optional[str] = None     # telephoneNumber
    phone: Optional[str] = None            # telephoneNumber (alias)
    last_modified: Optional[datetime] = None
    enabled: bool = True                   # userAccountControl
```

SnipeITUser

Representa um usuário do Snipe-IT.

```
class SnipeITUser(BaseModel):
    username: str                          # Nome de usuário único
    first_name: Optional[str] = None       # Primeiro nome
    last_name: Optional[str] = None        # Sobrenome
    email: EmailStr = None                 # Email (obrigatório)
    department: Optional[str] = None       # Departamento
    company_id: Optional[int] = None       # ID da empresa
    job_title: Optional[str] = None        # Cargo/função
    phone: Optional[str] = None            # Telefone
    enabled: bool = True                   # Status ativo/inativo
```

SyncResult

Representa o resultado de uma operação de sincronização.

```
class SyncResult(BaseModel):
    total_ldap_users: int                    # Total de usuários no LDAP
    user_created: int                        # Usuários criados
    user_updated: int                        # Usuários atualizados
    user_skipped: int                        # Usuários ignorados
    user_deactivated: int                    # Usuários desativados
    users_created_in_ad: int = 0            # Usuários criados especificamente
no AD
    errors: list[str] = []                   # Lista de erros ocorridos
    timestamp: datetime                      # Timestamp da operação
```

🔍 Lógica de Correspondência de Usuários

O sistema usa uma estratégia em cascata para encontrar usuários correspondentes:

1. Por Username (Prioridade Alta)

Compara `sAMAccountName` (LDAP) com `username` (Snipe-IT), ignorando maiúsculas/minúsculas.

```
if ldap_user.username.lower() == snipeit_user['username'].lower():
    return snipeit_user  # Match encontrado
```

2. Por Email (Prioridade Média)

Se username não corresponder, compara emails.

```
if ldap_user.email.lower() == snipeit_user['email'].lower():
    return snipeit_user  # Match encontrado
```

3. Por Nome Completo (Prioridade Baixa)

Se email não corresponder, compara nome completo.

```
ldap_fullname = f"{ldap_user.first_name} {ldap_user.last_name}".lower()
snipeit_fullname = f"{snipeit_user['first_name']}
{snipeit_user['last_name']}".lower()

if ldap_fullname == snipeit_fullname:
    return snipeit_user  # Match encontrado
```

Esta abordagem garante máxima correspondência mesmo quando os dados não estão perfeitamente alinhados.

Sistema de Logs

Os logs são armazenados em arquivos diários no diretório `logs/` :

Características

- **Formato de arquivo**: `YYYY-MM-DD.log`
- **Níveis**: DEBUG, INFO, WARNING, ERROR
- **Rotação**: Automática por dia
- **Saída**: Console + Arquivo

Formato de Log

```
%(asctime)s - %(name)s - %(levelname)s - %(message)s
```

Exemplo de Log

```
2024-11-15 10:30:00 - app.services.sync_service - INFO -
============================================================
2024-11-15 10:30:00 - app.services.sync_service - INFO - SCHEDULED SYNC IN
PROGRESS
2024-11-15 10:30:00 - app.services.sync_service - INFO - Last sync timestamp:
2024-11-15T10:15:00
2024-11-15 10:30:00 - app.services.sync_service - INFO -
============================================================
2024-11-15 10:30:01 - app.services.ldap_service - INFO - Connected to LDAP
server.
2024-11-15 10:30:02 - app.services.ldap_service - INFO - LDAP Search - Base
DN: DC=dominio,DC=local
2024-11-15 10:30:02 - app.services.ldap_service - INFO - LDAP Search - Filter:
(&(objectClass=user)(objectCategory=person))
2024-11-15 10:30:03 - app.services.ldap_service - INFO - Found 150 users in
LDAP.
2024-11-15 10:30:04 - app.services.snipeit_service - INFO - Total 145 users
retrieved from SnipeIT.
2024-11-15 10:30:15 - app.services.sync_service - INFO - LDAP user john.doe
created in SnipeIT.
2024-11-15 10:30:16 - app.services.sync_service - INFO - LDAP user jane.smith
updated in SnipeIT.
2024-11-15 10:30:20 - app.services.sync_service - INFO -
============================================================
2024-11-15 10:30:20 - app.services.sync_service - INFO - SYNC INFOS
2024-11-15 10:30:20 - app.services.sync_service - INFO - LDAP Total: 150
2024-11-15 10:30:20 - app.services.sync_service - INFO - Users Created in
SnipeIT: 5
2024-11-15 10:30:20 - app.services.sync_service - INFO - Users Created in AD:
0
2024-11-15 10:30:20 - app.services.sync_service - INFO - Users Updated: 12
2024-11-15 10:30:20 - app.services.sync_service - INFO - Users Skipped: 0
2024-11-15 10:30:20 - app.services.sync_service - INFO - Users Deactivated: 2
2024-11-15 10:30:20 - app.services.sync_service - INFO -
============================================================
```

Níveis de Log Recomendados

Ambiente	Nível	Uso
Desenvolvimento	DEBUG	Ver todos os detalhes e debug de código
Homologação	INFO	Ver operações normais e importantes
Produção	WARNING	Ver apenas avisos e erros

Rate Limiting

O sistema implementa rate limiting para proteger as APIs e evitar sobrecarga.

Snipe-IT API

Configuração

```
CALLS_PER_MINUTE=55
ONE_MINUTE=60
```

Comportamento

- **Limite**: 55 chamadas por minuto (configurável)
- **Janela**: 60 segundos deslizantes
- **Ação**: Aguarda automaticamente quando o limite é atingido
- **Retry**: Exponential backoff em caso de erro 429

Implementação

```
class RateLimiter:
    def __init__(self, max_calls: int, period: int):
        self.max_calls = max_calls
        self.period = period
        self.calls = deque()
        self.lock = Lock()
```

LDAP

- Sem limite rígido (geralmente LDAP suporta muitas conexões simultâneas)
- Conexões são abertas e fechadas adequadamente para evitar esgotamento de recursos
- Cada operação abre uma nova conexão e a fecha ao finalizar

Retry com Backoff

Para requisições ao Snipe-IT, o sistema usa retry automático:

```
@backoff.on_exception(
    backoff.expo,
    HTTPError,
    max_tries=5,
    giveup=lambda e: e.response is not None and
                     e.response.status_code not in [429, 500, 502, 503, 504]
)
def _make_request(self, method, endpoint, **kwargs):
    # ... código da requisição
```

- **Estratégia**: Exponential backoff
- **Tentativas máximas**: 5
- **Códigos para retry**: 429, 500, 502, 503, 504

Sincronização Periódica

A aplicação executa sincronização automática usando APScheduler.

Quando Sincroniza

1. **Na inicialização**: Executa uma sincronização imediatamente ao iniciar
2. **Periódica**: A cada `SYNC_INTERVAL_MINUTES` (padrão: 15 minutos)
3. **Manual**: Via endpoints da API

Configuração

```
scheduler = BackgroundScheduler(timezone="America/Sao_Paulo")

scheduler.add_job(
    scheduled_sync,
    "interval",
    id="sync_snipeit_to_ldap",
    name="Auto sync SnipeIT -> LDAP",
    minutes=settings.SYNC_INTERVAL_MINUTES,
    replace_existing=True
)
```

Informações da Última Sincronização

O sistema mantém registro da última sincronização:

```
last_sync_info = {
    "timestamp": "2024-11-15T10:30:00",
    "status": "success",  # ou "error", "pending"
    "result": {
        "total_ldap_users": 150,
        "users_created": 5,
        "users_updated": 12,
        "users_deactivated": 2,
        "errors": []
    }
}
```

Verificar Status

```
# Via API
curl http://localhost:8000/status

# Via logs
tail -f logs/$(date +%Y-%m-%d).log
```

Modo Dry-Run

O modo dry-run permite testar a sincronização sem fazer alterações reais nos sistemas.

Quando Usar

- **Testes iniciais**: Verificar o que seria feito antes de executar
- **Validação**: Confirmar que a lógica está correta
- **Auditoria**: Ver quais usuários seriam afetados

- **Troubleshooting**: Diagnosticar problemas sem risco

Como Usar

Via API

```
# LDAP → Snipe-IT (dry-run)
curl -X POST "http://localhost:8000/webhook/to/snipeit?dry_run=true" \
  -H "x-webhook-secret: seu-secret-aqui"

# Snipe-IT → LDAP (dry-run)
curl -X POST "http://localhost:8000/webhook/to/ldap?dry_run=true" \
  -H "x-webhook-secret: seu-secret-aqui"
```

Via Configuração Global

```
DRY_RUN=true
```

Isso fará com que TODAS as sincronizações (incluindo periódicas) executem em modo dry-run.

O Que Acontece

- Busca usuários normalmente
- Compara dados e identifica diferenças
- Registra nos logs o que seria feito
- NÃO cria usuários
- NÃO atualiza dados
- NÃO desativa contas

Exemplo de Log

```
2024-11-15 10:30:15 - app.services.sync_service - INFO - [DRY-RUN] LDAP user
john.doe would be created in SnipeIT.
2024-11-15 10:30:16 - app.services.sync_service - INFO - [DRY-RUN] LDAP user
jane.smith would be updated in SnipeIT.
2024-11-15 10:30:17 - app.services.sync_service - INFO - [DRY-RUN] SnipeIT
user bob.jones would be deactivated.
```

Monitoramento

Health Check para Sistemas de Monitoramento

1. Health Check HTTP

```
# Deve retornar status 200
curl -f http://localhost:8000/webhook/health || exit 1
```

Script de monitoramento:

```
#!/bin/bash
# check_webhook_health.sh
```

```
WEBHOOK_URL="http://localhost:8000/webhook/health"
RESPONSE=$(curl -s -o /dev/null -w "%{http_code}" $WEBHOOK_URL)

if [ $RESPONSE -eq 200 ]; then
    echo "OK - Webhook is healthy"
    exit 0
else
    echo "CRITICAL - Webhook is unhealthy (HTTP $RESPONSE)"
    exit 2
fi
```

2. Status da Última Sincronização

```
# Verificar se última sincronização foi bem-sucedida
curl -s http://localhost:8000/status | jq -r '.last_sync.status'
```

Script de monitoramento:

```
#!/bin/bash
# check_last_sync.sh

STATUS_URL="http://localhost:8000/status"
LAST_SYNC_STATUS=$(curl -s $STATUS_URL | jq -r '.last_sync.status')

case $LAST_SYNC_STATUS in
    "success")
        echo "OK - Last sync was successful"
        exit 0
        ;;
    "error")
        echo "CRITICAL - Last sync failed"
        exit 2
        ;;
    "pending")
        echo "WARNING - Sync is pending"
        exit 1
        ;;
    *)
        echo "UNKNOWN - Unable to determine sync status"
        exit 3
        ;;
esac
```

Documentação de Referência

- FastAPI Documentation
- ldap3 Documentation
- Snipe-IT API Documentation
- Active Directory LDAP