

# **DOCUMENTAÇÃO DA INSTALAÇÃO DO SISTEMA SNIPE-IT**

## **1. Introdução**

Este documento tem como objetivo descrever detalhadamente o processo de **instalação, configuração e integração do sistema Snipe-IT**, utilizado para o gerenciamento de ativos de Tecnologia da Informação (TI) na instituição.

A documentação foi elaborada para garantir **rastreabilidade, reproduzibilidade e manutenção futura** do ambiente implantado, além de servir como **evidência técnica** da implementação prática do TCC.

---

## **2. Ambiente de Instalação**

Componente	Especificação
Sistema Operacional	Ubuntu Server 22.04 LTS
Virtualização	Máquina Virtual (VM) local
Gerenciador de containers	Docker e Docker Compose
Banco de Dados	MariaDB
Acesso remoto	SSH
Autenticação	Microsoft 365 (Entra ID)
Serviço de e-mail (SMTP)	Outlook 365
Certificado SSL	Let's Encrypt / Autoassinatura
Backup	Diário automático (volumes Docker)

---

## **3. Etapas da Instalação**

### **3.1 Instalação via Docker**

#### **1. Atualização do sistema e instalação de dependências**

```
sudo apt update && sudo apt install docker.io docker-compose -y  
Criação do diretório do projeto'''
```

#### **2. Criação do diretório do projeto**

```
mkdir /opt/snipeit && cd /opt/snipeit
```

### 3. Download do docker-compose.yml

- Obtenção do arquivo oficial do Snipe-IT via GitHub.
- Configuração dos volumes persistentes e rede Docker.

### 4. Configuração do docker-compose.yml

```
volumes:  
  db_data:  
    storage:  
  
services:  
  app:  
    image: snipe/snipe-it:${APP_VERSION:-v7.0.11}  
    restart: unless-stopped  
    volumes:  
      - storage:/var/lib/snipeit  
    ports:  
      - "${APP_PORT:-8000}:80"  
    depends_on:  
      db:  
        condition: service_healthy  
        restart: true  
    env_file:  
      - .env  
  
  db:  
    image: mariadb:11.5.2  
    restart: unless-stopped  
    volumes:  
      - db_data:/var/lib/mysql  
    environment:  
      MYSQL_DATABASE: snipeitdb  
      MYSQL_USER: admin  
      MYSQL_PASSWORD: Adm1n!  
      MYSQL_ROOT_PASSWORD: Adm1n!  
    healthcheck:  
      # https://mariadb.com/kb/en/using-healthcheck-sh/#compose-file-example  
      test: ["CMD", "healthcheck.sh", "--connect", "--innodb_initialized"]  
      interval: 5s  
      timeout: 1s  
      retries: 5
```

### 5. Inicialização dos containers

```
sudo docker-compose up -d
```

## 3.2 Configuração do Arquivo .env

O arquivo .env define as variáveis de ambiente e parâmetros do sistema.

Principais parâmetros configurados:

```
# -----
# REQUIRED: DOCKER SPECIFIC SETTINGS
# -----
APP_VERSION=v6.4.1
APP_PORT=8000

# -----
# REQUIRED: BASIC APP SETTINGS
# -----
APP_ENV=production
APP_DEBUG=false
# Please regenerate the APP_KEY value by calling `docker compose run --rm snipeit php artisan key:generate --show`. Copy paste the value here
APP_KEY=dkckh3%6kd8!@ahg9
APP_URL=https://ativos.escola.local
# https://en.wikipedia.org/wiki/List_of_tz_database_time_zones - TZ identifier
APP_TIMEZONE='America/Sao_Paulo'
APP_LOCALE=pt-BR
MAX_RESULTS=500

# -----
# REQUIRED: UPLOADED FILE STORAGE SETTINGS
# -----
PRIVATE_FILESYSTEM_DISK=local
PUBLIC_FILESYSTEM_DISK=local_public

# -----
# REQUIRED: DATABASE SETTINGS
# -----
DB_CONNECTION=mysql
DB_HOST=db
DB_PORT='3306'
DB_DATABASE=snipeitdb
DB_USERNAME=admin!
DB_PASSWORD=Adm1n!pass3o7d!
MYSQL_ROOT_PASSWORD=Adm1n!pass3o7d!
DB_PREFIX=null
DB_DUMP_PATH='/usr/bin'
DB_CHARSET=utf8mb4
DB_COLLATION=utf8mb4_unicode_ci

# -----
```

```
# OPTIONAL: SSL DATABASE SETTINGS
# -----
DB_SSL=false
DB_SSL_IS_PAAS=false
DB_SSL_KEY_PATH=null
DB_SSL_CERT_PATH=null
DB_SSL_CA_PATH=null
DB_SSL_CIPHER=null
DB_SSL_VERIFY_SERVER=null

# -----
# REQUIRED: OUTGOING MAIL SERVER SETTINGS
# -----
MAIL_MAILER=smtp
MAIL_HOST=smtp.server.com
MAIL_PORT=587
MAIL_USERNAME=ativos@escola.com.br
MAIL_PASSWORD=smtp#password!
MAIL_ENCRYPTION=tls
MAIL_TLS_VERIFY_PEER=false
MAIL_FROM_ADDR=ativos@escola.com.br
MAIL_FROM_NAME='Ativos Escola'
MAIL_REPLYTO_ADDR=ativos@escola.com.br
MAIL_REPLYTO_NAME='Escola Altaneiro'
MAIL_AUTO_EMBED_METHOD='attachment'

# -----
# REQUIRED: DATA PROTECTION
# -----
ALLOW_BACKUP_DELETE=false
ALLOW_DATA_PURGE=false

# -----
# REQUIRED: IMAGE LIBRARY
# This should be gd or imagick
# -----
IMAGE_LIB=gd

# -----
# OPTIONAL: BACKUP SETTINGS
# -----
MAIL_BACKUP_NOTIFICATION_DRIVER=null
MAIL_BACKUP_NOTIFICATION_ADDRESS=null
BACKUP_ENV=true

# -----
# OPTIONAL: SESSION SETTINGS
# -----
SESSION_LIFETIME=12000
EXPIRE_ON_CLOSE=false
```

```
ENCRYPT=false
COOKIE_NAME=snipeit_session
COOKIE_DOMAIN=null
SECURE_COOKIES=false
API_TOKEN_EXPIRATION_YEARS=40

# -----
# OPTIONAL: SECURITY HEADER SETTINGS
# -----
APP_TRUSTED_PROXIES=192.168.1.1,10.0.0.1,172.16.0.0/12
ALLOW_IFRAMING=false
REFERRER_POLICY=same-origin
ENABLE_CSP=false
CORS_ALLOWED_ORIGINS=null
ENABLE_HSTS=false

# -----
# OPTIONAL: CACHE SETTINGS
# -----
CACHE_DRIVER=file
SESSION_DRIVER=file
QUEUE_DRIVER=sync
CACHE_PREFIX=snipeit

# -----
# OPTIONAL: REDIS SETTINGS
# -----
REDIS_HOST=null
REDIS_PASSWORD=null
REDIS_PORT=6379

# -----
# OPTIONAL: MEMCACHED SETTINGS
# -----
MEMCACHED_HOST=null
MEMCACHED_PORT=null

# -----
# OPTIONAL: PUBLIC S3 Settings
# -----
PUBLIC_AWS_SECRET_ACCESS_KEY=null
PUBLIC_AWS_ACCESS_KEY_ID=null
PUBLIC_AWS_DEFAULT_REGION=null
PUBLIC_AWS_BUCKET=null
PUBLIC_AWS_URL=null
PUBLIC_AWS_BUCKET_ROOT=null

# -----
# OPTIONAL: PRIVATE S3 Settings
# -----
```

```

PRIVATE_AWS_ACCESS_KEY_ID=null
PRIVATE_AWS_SECRET_ACCESS_KEY=null
PRIVATE_AWS_DEFAULT_REGION=null
PRIVATE_AWS_BUCKET=null
PRIVATE_AWS_URL=null
PRIVATE_AWS_BUCKET_ROOT=null

# -----
# OPTIONAL: AWS Settings
# -----
AWS_ACCESS_KEY_ID=null
AWS_SECRET_ACCESS_KEY=null
AWS_DEFAULT_REGION=null

# -----
# OPTIONAL: LOGIN THROTTLING
# -----
LOGIN_MAX_ATTEMPTS=5
LOGIN_LOCKOUT_DURATION=60
RESET_PASSWORD_LINK_EXPIRES=900

# -----
# OPTIONAL: MISC
# -----
LOG_CHANNEL=stderr
LOG_MAX_DAYS=10
APP_LOCKED=false
APP_CIPHER=AES-256-CBC
APP_FORCE_TLS=false
GOOGLE_MAPS_API=
LDAP_MEM_LIM=500M
LDAP_TIME_LIM=600

```

### 3.3 Configuração do HTTPS

- Instalação do Nginx como proxy reverso.
- Criação de arquivo de configuração:

```
sudo vim /etc/nginx/sites-available/snipeit
```

- Configuração do nginx :

```

server {
    listen 80;
    server_name ativos.escola.local;
    return 301 https://$host$request_uri;
}

```

```

}

server {
    listen 443 ssl;
    server_name ativos.escola.local;

    ssl_certificate /etc/ssl/certs/snipeit.crt;
    ssl_certificate_key /etc/ssl/private/snipeit.key;

    location / {
        proxy_pass http://127.0.0.1:8000; # Porta do Snipe-IT no Docker
        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;
    }
}

```

- Geração de certificado SSL:

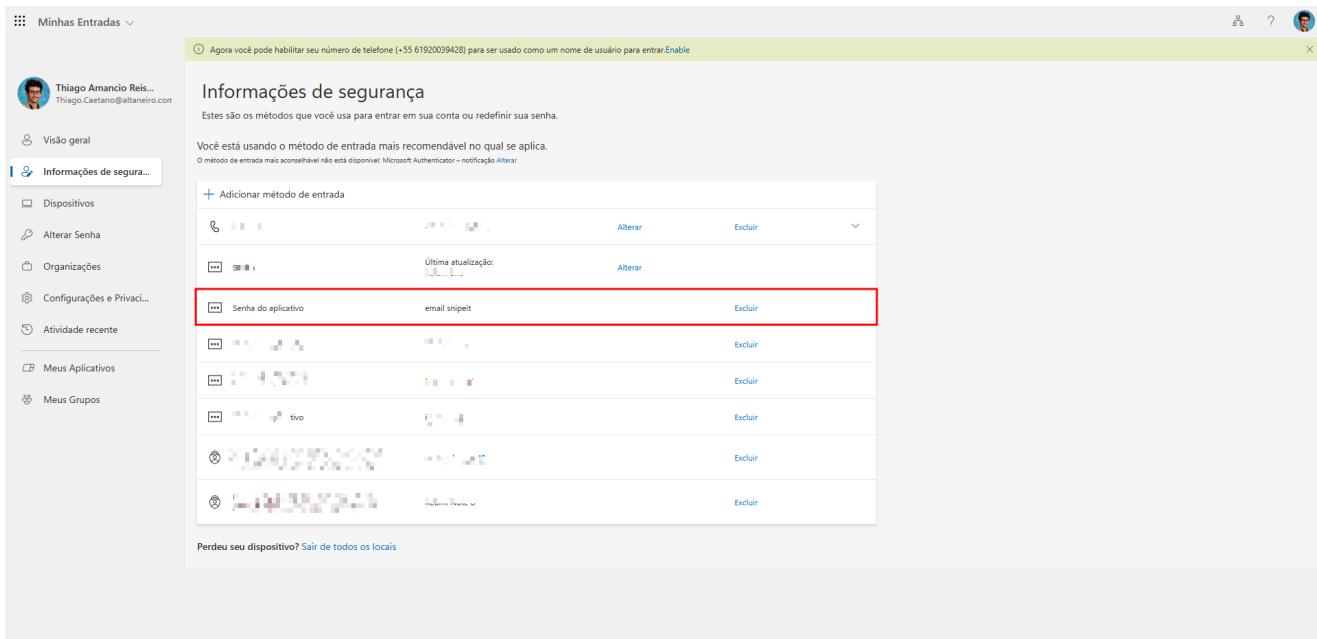
```
sudo certbot --nginx -d ativos.escola.local
```

- Redirecionamento HTTP → HTTPS.

## 3.4 Configuração do SMTP

Configuração com Microsoft Exchange para envio de notificações e recuperação de senha.

- Geração de senha de aplicativo para configuração no .env :



- Configuração do server SMTP no .env :

```

# -----
# REQUIRED: OUTGOING MAIL SERVER SETTINGS
# -----
MAIL_MAILER=smtp
MAIL_HOST=smtp.server.com
MAIL_PORT=587
MAIL_USERNAME=ativos@escola.com.br
MAIL_PASSWORD=smtp#password!
MAIL_ENCRYPTION=tls
MAIL_TLS_VERIFY_PEER=false
MAIL_FROM_ADDR=ativos@escola.com.br
MAIL_FROM_NAME='Ativos Escola'
MAIL_REPLYTO_ADDR=ativos@escola.com.br
MAIL_REPLYTO_NAME='Ativos Escola'
MAIL_AUTO_EMBED_METHOD='attachment'

```

## 3.5 Integração com Microsoft 365 (Entra ID)

- Registro da aplicação no Azure Portal.

- Configuração dos parâmetros no Snipe-IT.

Teste de autenticação federada com contas institucionais.

## 3.6 Exportação de Dados

### 3.6.1 Ativos

- Exportação inicial de dados em formato CSV/Excel

### 3.6.2 Usuários

- Exportação via Microsoft Admin Center (CSV).

## **3.7 Importação no Snipe-IT**

- Acesso ao módulo Importação > CSV Upload.
- Mapeamento das colunas CSV com os campos internos do Snipe-IT.
- Verificação de duplicidades e inconsistência dos dados.