



Guia de Instalação - Sistema SaaS de Agendamento Online



Visão Geral

Este guia fornece instruções detalhadas para instalar e configurar o sistema SaaS de agendamento online em um servidor VPS Ubuntu. O sistema permite que prestadores de serviços criem suas próprias páginas de agendamento e gerenciem seus clientes de forma profissional.



Arquitetura do Sistema

- **Backend:** Flask (Python) com SQLAlchemy
- **Frontend:** React com TailwindCSS e shadcn/ui
- **Banco de Dados:** PostgreSQL
- **Cache:** Redis
- **Proxy Reverso:** NGINX
- **Containerização:** Docker + Docker Compose
- **SSL:** Let's Encrypt (Certbot)



Pré-requisitos

Servidor

- VPS Ubuntu 20.04 LTS ou superior
- Mínimo 2GB RAM (recomendado 4GB)
- Mínimo 20GB de armazenamento SSD
- Acesso root via SSH

- Domínio apontando para o servidor (para SSL)

Conhecimentos Necessários

- Comandos básicos do Linux
- Conceitos de Docker
- Configuração de DNS



Instalação Rápida

Passo 1: Preparar o Servidor

```
# Conectar ao servidor
ssh root@seu-servidor.com

# Atualizar sistema
apt update && apt upgrade -y

# Instalar dependências básicas
apt install -y curl wget git unzip
```

Passo 2: Baixar o Projeto

```
# Navegar para o diretório home
cd /home/ubuntu

# Clonar ou baixar o projeto
# (substitua pelo método de sua preferência)
wget https://github.com/seu-usuario/saas-agendamentos/archive/main.zip
unzip main.zip
mv saas-agendamentos-main saas-agendamentos
cd saas-agendamentos







# Ou se usando git:
# git clone https://github.com/seu-usuario/saas-agendamentos.git
# cd saas-agendamentos
```

Passo 3: Executar Deploy Automático

```
# Tornar script executável
chmod +x scripts/deploy.sh

# Deploy em produção com SSL
sudo ./scripts/deploy.sh --domain seudominio.com --email admin@seudominio.com -
-ssl

# Ou deploy em desenvolvimento (sem SSL)
sudo ./scripts/deploy.sh --dev
```

O script de deploy irá: -  Instalar todas as dependências -  Configurar firewall - 
Configurar NGINX -  Configurar SSL (se solicitado) -  Criar containers Docker - 
Configurar backups automáticos

Instalação Manual Detalhada

Se preferir fazer a instalação passo a passo:

Passo 1: Instalar Dependências

```
# Atualizar repositórios
apt update && apt upgrade -y

# Instalar Docker
apt install -y docker.io docker-compose

# Instalar NGINX
apt install -y nginx

# Instalar Certbot (para SSL)
apt install -y certbot python3-certbot-nginx

# Instalar UFW (firewall)
apt install -y ufw

# Instalar utilitários
apt install -y curl wget git unzip htop
```

Passo 2: Configurar Firewall

```
# Configurar UFW
ufw default deny incoming
ufw default allow outgoing
ufw allow OpenSSH
ufw allow 'Nginx Full'
ufw --force enable

# Verificar status
ufw status
```

Passo 3: Configurar Ambiente

```
# Navegar para o projeto
cd /home/ubuntu/saas-agendamentos

# Copiar arquivo de ambiente
cp .env.example .env

# Editar configurações (IMPORTANTE!)
nano .env
```

Configurações importantes no .env:

```
# Domínio
DOMAIN=seudominio.com
FRONTEND_URL=https://seudominio.com
NEXT_PUBLIC_SITE_URL=https://seudominio.com

# Banco de dados (gere senhas seguras!)
DB_PASSWORD=sua_senha_super_segura_aqui
REDIS_PASSWORD=sua_senha_redis_aqui

# JWT (gere tokens seguros!)
JWT_SECRET=seu_jwt_secret_de_32_caracteres
JWT_REFRESH_SECRET=seu_refresh_secret_de_32_caracteres

# Email SMTP
EMAIL_HOST=smtp.gmail.com
EMAIL_USER=seu-email@gmail.com
EMAIL_PASS=sua-senha-de-app

# SSL
SSL_EMAIL=admin@seudominio.com
```

Passo 4: Configurar NGINX

```
# Parar NGINX
systemctl stop nginx

# Backup da configuração atual
cp /etc/nginx/sites-available/default /etc/nginx/sites-available/default.backup

# Copiar nova configuração
cp nginx/default.conf /etc/nginx/sites-available/default

# Substituir domínio na configuração
sed -i "s/DOMAIN/seudominio.com/g" /etc/nginx/sites-available/default

# Testar configuração
nginx -t

# Se OK, continuar...
```

Passo 5: Iniciar Containers

```
# Criar diretórios necessários
mkdir -p backend/uploads backend/logs backups

# Ajustar permissões
chown -R 1000:1000 backend/uploads backend/logs
chmod -R 755 backend/uploads backend/logs

# Iniciar containers
docker-compose up -d --build

# Verificar status
docker-compose ps

# Ver logs se necessário
docker-compose logs -f
```

Passo 6: Configurar SSL

```
# Iniciar NGINX
systemctl start nginx
systemctl enable nginx

# Obter certificado SSL
certbot --nginx -dseudominio.com --email admin@seudominio.com --agree-tos --non-interactive

# Configurar renovação automática
echo "0 12 * * * /usr/bin/certbot renew --quiet" | crontab -
```

Passo 7: Configurar Backups

```
# Tornar script executável
chmod +x scripts/backup.sh scripts/restore.sh

# Configurar backup automático (diário às 2h)
echo "0 2 * * * cd /home/ubuntu/saas-agendamentos && ./scripts/backup.sh >>
/var/log/saas-backup.log 2>&1" | crontab -

# Testar backup manual
./scripts/backup.sh
```

✓ Verificação da Instalação

Verificar Containers

```
cd /home/ubuntu/saas-agendamentos
docker-compose ps
```

Todos os containers devem estar com status "Up": - saas_postgres - saas_redis
- saas_backend - saas_frontend - saas_nginx

Verificar Aplicação

```
# Testar localmente
curl http://localhost/health

# Testar via domínio
curl https://seudominio.com/health

# Testar API
curl https://seudominio.com/api
```

Verificar Logs

```
# Logs dos containers
docker-compose logs

# Logs específicos
docker-compose logs backend
docker-compose logs frontend

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

Configurações Adicionais

Configurar Email SMTP

Para envio de notificações, configure um provedor SMTP no arquivo `.env` :

Gmail:

```
EMAIL_HOST=smtp.gmail.com
EMAIL_PORT=587
EMAIL_USER=seu-email@gmail.com
EMAIL_PASS=sua-senha-de-app # Gere em:
https://myaccount.google.com/apppasswords
```

SendGrid:

```
EMAIL_HOST=smtp.sendgrid.net
EMAIL_PORT=587
EMAIL_USER=apikey
EMAIL_PASS=sua-api-key-sendgrid
```

Configurar Backup em Nuvem (Opcional)

Para backup automático no AWS S3, adicione no `.env` :

```
AWS_ACCESS_KEY_ID=sua-access-key
AWS_SECRET_ACCESS_KEY=sua-secret-key
AWS_REGION=us-east-1
BACKUP_S3_BUCKET=seu-bucket-backup
```

Configurar Monitoramento (Opcional)

Para monitoramento com Sentry, adicione no `.env` :

```
SENTRY_DSN=https://sua-dsn@sentry.io/projeto
```



Solução de Problemas

Container não inicia

```
# Ver logs detalhados
docker-compose logs nome-do-container

# Reconstruir container
docker-compose down
docker-compose up -d --build nome-do-container
```

Erro de permissão

```
# Ajustar permissões
sudo chown -R 1000:1000 backend/uploads backend/logs
sudo chmod -R 755 backend/uploads backend/logs
```

Erro de SSL

```
# Verificar se domínio aponta para servidor
dig +short seudominio.com

# Renovar certificado manualmente
certbot renew --dry-run
```

Banco de dados não conecta

```
# Verificar se PostgreSQL está rodando
docker-compose ps postgres

# Verificar logs do banco
docker-compose logs postgres

# Reiniciar banco
docker-compose restart postgres
```


Aplicação lenta

```
# Verificar recursos do servidor
htop
df -h

# Verificar logs de erro
docker-compose logs | grep -i error

# Reiniciar containers
docker-compose restart
```



Monitoramento

Comandos Úteis

```
# Status dos containers
docker-compose ps

# Uso de recursos
docker stats

# Logs em tempo real
docker-compose logs -f

# Espaço em disco
df -h

# Processos do sistema
htop
```

Arquivos de Log

- **Aplicação:** `backend/logs/`
- **NGINX:** `/var/log/nginx/`
- **Backup:** `/var/log/saas-backup.log`
- **Sistema:** `/var/log/syslog`



Atualizações

Atualizar Sistema

```
cd /home/ubuntu/saas-agendamentos

# Fazer backup antes
./scripts/backup.sh

# Baixar nova versão
git pull origin main
# ou baixar novo arquivo zip

# Atualizar containers
sudo ./scripts/deploy.sh --update
```

Rollback

```
# Listar backups disponíveis
./scripts/restore.sh --list

# Restaurar backup específico
./scripts/restore.sh db_backup_20240101_120000.sql.gz
```



Segurança

Recomendações

1. **Senhas Fortes:** Use senhas complexas no `.env`
2. **Firewall:** Mantenha UFW ativo
3. **SSL:** Sempre use HTTPS em produção
4. **Backups:** Configure backups automáticos
5. **Atualizações:** Mantenha sistema atualizado
6. **Monitoramento:** Monitore logs regularmente

Hardening Adicional

```
# Desabilitar login root via SSH
sed -i 's/PermitRootLogin yes/PermitRootLogin no/' /etc/ssh/sshd_config
systemctl restart ssh

# Configurar fail2ban
apt install -y fail2ban
systemctl enable fail2ban
systemctl start fail2ban

# Configurar logrotate para logs da aplicação
cat > /etc/logrotate.d/saas-agendamentos << EOF
/home/ubuntu/saas-agendamentos/backend/logs/*.log {
    daily
    missingok
    rotate 30
    compress
    delaycompress
    notifempty
    copytruncate
}
EOF
```

Suporte

Recursos de Ajuda

- **Documentação:** README.md do projeto
- **Logs:** Sempre verifique os logs primeiro
- **Comunidade:** GitHub Issues
- **Email:** suporte@seudominio.com

Informações do Sistema

```
# Versão do sistema
cat /etc/os-release

# Recursos do servidor
free -h
df -h
lscpu

# Versões instaladas
docker --version
docker-compose --version
nginx -v
```

✓ **Instalação concluída com sucesso!**

Acesse seu sistema em: <https://seudominio.com>

Lembre-se de: 1. Revisar todas as configurações no arquivo `.env` 2. Configurar email SMTP para notificações 3. Fazer backup regularmente 4. Monitorar logs e recursos do servidor