

Guia de Instalação e Configuração

Requisitos do Sistema

Requisitos Mínimos

Servidor

- **Sistema Operacional:** Linux Ubuntu 20.04+ ou Windows Server 2019+
- **Processador:** 2 cores, 2.0 GHz
- **Memória RAM:** 4 GB
- **Armazenamento:** 20 GB de espaço livre
- **Conexão:** Internet banda larga

Software

- **Python:** 3.9 ou superior
- **Node.js:** 18.0 ou superior
- **Banco de Dados:** SQLite (desenvolvimento) ou PostgreSQL 12+ (produção)
- **Servidor Web:** Nginx (recomendado)

Requisitos Recomendados

Servidor

- **Processador:** 4 cores, 3.0 GHz
- **Memória RAM:** 8 GB
- **Armazenamento:** SSD 50 GB
- **Backup:** Sistema de backup automatizado

Instalação do Backend (API)

1. Preparação do Ambiente

```
# Atualizar sistema (Ubuntu)
sudo apt update && sudo apt upgrade -y

# Instalar Python e dependências
sudo apt install python3 python3-pip python3-venv git -y
```

```
# Instalar PostgreSQL (opcional, para produção)
sudo apt install postgresql postgresql-contrib -y
```

2. Download e Configuração

```
# Criar diretório do projeto
mkdir /opt/salon-booking
cd /opt/salon-booking

# Clonar ou extrair os arquivos do sistema
# (substitua pelo método de distribuição escolhido)
unzip salon-booking-system.zip
cd salon-booking-api

# Criar ambiente virtual
python3 -m venv venv
source venv/bin/activate

# Instalar dependências
pip install -r requirements.txt
```

3. Configuração do Banco de Dados

SQLite (Desenvolvimento)

```
# Criar banco de dados
python src/main.py db init
python src/main.py db migrate
python src/main.py db upgrade
```

PostgreSQL (Produção)

```
# Criar usuário e banco
sudo -u postgres psql
CREATE USER salon_user WITH PASSWORD 'senha_segura';
CREATE DATABASE salon_booking OWNER salon_user;
GRANT ALL PRIVILEGES ON DATABASE salon_booking TO salon_user;
\q

# Configurar variável de ambiente
export DATABASE_URL="postgresql://
salon_user:senha_segura@localhost/salon_booking"
```

```
# Executar migrações
python src/main.py db upgrade
```

4. Configuração de Variáveis de Ambiente

Criar arquivo `.env`:

```
# Configurações do Flask
FLASK_ENV=production
SECRET_KEY=sua_chave_secreta_muito_segura_aqui
DEBUG=False

# Banco de Dados
DATABASE_URL=sqlite:///salon_booking.db
# ou para PostgreSQL:
# DATABASE_URL=postgresql://salon_user:senha@localhost/
# salon_booking

# JWT
JWT_SECRET_KEY=sua_chave_jwt_muito_segura
JWT_ACCESS_TOKEN_EXPIRES=3600

# CORS
CORS_ORIGINS=http://localhost:3000,https://seudominio.com

# Email (opcional)
MAIL_SERVER=smtp.gmail.com
MAIL_PORT=587
MAIL_USE_TLS=True
MAIL_USERNAME=seu_email@gmail.com
MAIL_PASSWORD=sua_senha_app
```

5. Teste da Instalação

```
# Ativar ambiente virtual
source venv/bin/activate

# Executar aplicação
python src/main.py

# Testar API
curl http://localhost:5000/api/v1/health
```

Instalação do Frontend

1. Preparação

```
# Instalar Node.js (Ubuntu)
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash
-
sudo apt-get install -y nodejs

# Verificar instalação
node --version
npm --version
```

2. Configuração do Frontend

```
# Navegar para diretório do frontend
cd /opt/salon-booking/salon-booking-frontend

# Instalar dependências
npm install

# Configurar variáveis de ambiente
cp .env.example .env.local
```

Editar `.env.local`:

```
REACT_APP_API_URL=http://localhost:5000/api/v1
REACT_APP_ENVIRONMENT=production
```

3. Build para Produção

```
# Gerar build otimizado
npm run build

# Os arquivos estarão em ./build/
```

Configuração do Servidor Web (Nginx)

1. Instalação do Nginx

```
# Ubuntu
sudo apt install nginx -y

# Iniciar e habilitar
sudo systemctl start nginx
sudo systemctl enable nginx
```

2. Configuração do Site

Criar arquivo `/etc/nginx/sites-available/salon-booking:`

```
server {
    listen 80;
    server_name seudominio.com www.seudominio.com;

    # Frontend (React)
    location / {
        root /opt/salon-booking/salon-booking-frontend/build;
        index index.html;
        try_files $uri $uri/ /index.html;
    }

    # API (Flask)
    location /api/ {
        proxy_pass http://127.0.0.1:5000;
        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;
    }

    # Arquivos estáticos
    location /static/ {
        root /opt/salon-booking/salon-booking-frontend/build;
        expires 1y;
        add_header Cache-Control "public, immutable";
    }
}
```

3. Ativar Site

```
# Criar link simbólico
sudo ln -s /etc/nginx/sites-available/salon-booking /etc/nginx/sites-enabled/

# Remover site padrão
sudo rm /etc/nginx/sites-enabled/default

# Testar configuração
sudo nginx -t

# Reiniciar Nginx
sudo systemctl restart nginx
```

Configuração de SSL (HTTPS)

1. Instalar Certbot

```
# Ubuntu
sudo apt install certbot python3-certbot-nginx -y
```

2. Obter Certificado

```
# Gerar certificado SSL
sudo certbot --nginx -d seudominio.com -d www.seudominio.com

# Testar renovação automática
sudo certbot renew --dry-run
```

Configuração de Serviços Systemd

1. Serviço do Backend

Criar arquivo `/etc/systemd/system/salon-booking-api.service`:

```
[Unit]
Description=Salon Booking API
After=network.target
```

```
[Service]
Type=simple
User=www-data
WorkingDirectory=/opt/salon-booking/salon-booking-api
Environment=PATH=/opt/salon-booking/salon-booking-api/venv/bin
ExecStart=/opt/salon-booking/salon-booking-api/venv/bin/python
src/main.py
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target
```

2. Ativar Serviços

```
# Recarregar systemd
sudo systemctl daemon-reload

# Habilitar e iniciar serviço
sudo systemctl enable salon-booking-api
sudo systemctl start salon-booking-api

# Verificar status
sudo systemctl status salon-booking-api
```

Configuração de Backup

1. Script de Backup

Criar arquivo `/opt/salon-booking/backup.sh`:

```
#!/bin/bash

# Configurações
BACKUP_DIR="/opt/backups/salon-booking"
DATE=$(date +%Y%m%d_%H%M%S)
DB_NAME="salon_booking"

# Criar diretório de backup
mkdir -p $BACKUP_DIR

# Backup do banco de dados
if [ "$DATABASE_URL" == *"postgresql"* ]; then
    pg_dump $DB_NAME > $BACKUP_DIR/db_backup_$DATE.sql
else
```

```
cp /opt/salon-booking/salon-booking-api/salon_booking.db
$BACKUP_DIR/db_backup_$DATE.db
fi

# Backup dos arquivos
tar -czf $BACKUP_DIR/files_backup_$DATE.tar.gz /opt/salon-
booking

# Remover backups antigos (manter últimos 7 dias)
find $BACKUP_DIR -name "*.sql" -mtime +7 -delete
find $BACKUP_DIR -name "*.db" -mtime +7 -delete
find $BACKUP_DIR -name "*.tar.gz" -mtime +7 -delete

echo "Backup concluído: $DATE"
```

2. Agendar Backup

```
# Tornar script executável
chmod +x /opt/salon-booking/backup.sh

# Adicionar ao crontab (backup diário às 2h)
sudo crontab -e
# Adicionar linha:
0 2 * * * /opt/salon-booking/backup.sh >> /var/log/salon-
backup.log 2>&1
```

Monitoramento e Logs

1. Configuração de Logs

```
# Criar diretório de logs
sudo mkdir -p /var/log/salon-booking

# Configurar rotação de logs
sudo nano /etc/logrotate.d/salon-booking
```

Conteúdo do arquivo:

```
/var/log/salon-booking/*.log {
    daily
    missingok
    rotate 30
    compress
    delaycompress
```



```
    notifempty
    create 644 www-data www-data
}
```

2. Monitoramento Básico

Script de monitoramento `/opt/salon-booking/monitor.sh`:

```
#!/bin/bash

# Verificar se API está respondendo
if curl -f http://localhost:5000/api/v1/health > /dev/null
2>&1; then
    echo "$(date): API OK"
else
    echo "$(date): API ERROR - Reiniciando serviço"
    sudo systemctl restart salon-booking-api
fi

# Verificar espaço em disco
DISK_USAGE=$(df / | tail -1 | awk '{print $5}' | sed 's/%//')
if [ $DISK_USAGE -gt 80 ]; then
    echo "$(date): AVISO - Espaço em disco baixo: ${DISK_USAGE}%"
fi
```

Solução de Problemas

Problemas Comuns

API não inicia

```
# Verificar logs
sudo journalctl -u salon-booking-api -f

# Verificar permissões
sudo chown -R www-data:www-data /opt/salon-booking

# Verificar dependências
source /opt/salon-booking/salon-booking-api/venv/bin/activate
pip check
```

Frontend não carrega

```
# Verificar build
cd /opt/salon-booking/salon-booking-frontend
npm run build

# Verificar permissões Nginx
sudo chown -R www-data:www-data /opt/salon-booking/salon-booking-frontend/build

# Verificar configuração Nginx
sudo nginx -t
```

Banco de dados não conecta

```
# PostgreSQL
sudo systemctl status postgresql
sudo -u postgres psql -c "\l"

# SQLite
ls -la /opt/salon-booking/salon-booking-api/salon_booking.db
```

Comandos Úteis

```
# Verificar status dos serviços
sudo systemctl status salon-booking-api nginx postgresql

# Visualizar logs em tempo real
sudo journalctl -u salon-booking-api -f
sudo tail -f /var/log/nginx/access.log

# Reiniciar todos os serviços
sudo systemctl restart salon-booking-api nginx

# Verificar conectividade
curl -I http://localhost:5000/api/v1/health
curl -I http://seudominio.com
```

Atualizações

Processo de Atualização

1. **Backup:** Sempre fazer backup antes de atualizar

2. **Parar serviços:** `sudo systemctl stop salon-booking-api`
 3. **Atualizar código:** Substituir arquivos da aplicação
 4. **Atualizar dependências:** `pip install -r requirements.txt`
 5. **Migrar banco:** `python src/main.py db upgrade`
 6. **Rebuild frontend:** `npm run build`
 7. **Reiniciar serviços:** `sudo systemctl start salon-booking-api`
 8. **Testar:** Verificar se tudo está funcionando
-

Versão do Guia: 1.0

Última Atualização: Dezembro 2024