

Guia de Implantação e Configuração

Agente de IA para Atendimento de Farmácia

Versão 1.0.0

Sumário

1. [Introdução](#)
 2. [Requisitos do Sistema](#)
 3. [Instalação](#)
 4. [Configuração](#)
 5. [Configuração do Ambiente](#)
 6. [Configuração do Telegram](#)
 7. [Configuração do WhatsApp](#)
 8. [Configuração do Banco de Dados](#)
 9. [Implantação](#)
 10. [Implantação Local](#)
 11. [Implantação em Servidor](#)
 12. [Implantação em Nuvem](#)
 13. [Testes](#)
 14. [Manutenção](#)
 15. [Solução de Problemas](#)
 16. [Referências](#)
-

Introdução

Este guia fornece instruções detalhadas para a implantação e configuração do Agente de IA para Atendimento de Farmácia. O sistema consiste em um bot inteligente capaz de responder a perguntas sobre medicamentos, verificar disponibilidade de produtos, informar horários de funcionamento e muito mais, com integração para Telegram (testes) e WhatsApp (uso profissional).

Requisitos do Sistema

Hardware Recomendado

- CPU: 2 núcleos ou mais
- RAM: 2GB ou mais
- Armazenamento: 1GB de espaço livre

Software Necessário

- Python 3.8 ou superior
- MongoDB (opcional, para versões completas)
- pip (gerenciador de pacotes Python)
- Git (para clonar o repositório)
- Servidor web com suporte a HTTPS (para webhooks do WhatsApp)

Contas e Tokens Necessários

- Conta de desenvolvedor do Telegram e token de bot (via BotFather)
 - Conta de desenvolvedor do WhatsApp Business API e token de acesso
 - Número de telefone WhatsApp Business
-

Instalação

1. Clone o Repositório

```
git clone https://github.com/seu-usuario/farmacia-bot.git  
cd farmacia-bot
```

2. Crie e Ative um Ambiente Virtual (Recomendado)

```
# No Linux/macOS  
python3 -m venv venv  
source venv/bin/activate
```

```
# No Windows  
python -m venv venv  
venv\Scripts\activate
```

3. Instale as Dependências

```
pip install -r requirements.txt
```

4. Baixe o Modelo de Linguagem em Português para o spaCy

```
python -m spacy download pt_core_news_sm
```

Configuração

Configuração do Ambiente

1. Crie um arquivo `.env` na raiz do projeto (você pode copiar o arquivo `.env.example`):

```
cp .env.example .env
```

1. Edite o arquivo `.env` com suas configurações:

```
# Configurações do bot
TELEGRAM_TOKEN=seu_token_aqui
WHATSAPP_TOKEN=seu_token_aqui
WHATSAPP_PHONE_ID=seu_phone_id_aqui
VERIFY_TOKEN=seu_token_de_verificacao_aqui

# Configurações do banco de dados
MONGODB_URI=mongodb://localhost:27017/
DB_NAME=farmacia_bot

# Configurações da aplicação
DEBUG=False
PORT=5000
```

Configuração do Telegram

1. Crie um bot no Telegram usando o BotFather:
2. Abra o Telegram e pesquise por "@BotFather"
3. Inicie uma conversa e envie o comando `/newbot`
4. Siga as instruções para criar um novo bot

5. Copie o token fornecido pelo BotFather
6. Adicione o token ao arquivo `.env` : `TELEGRAM_TOKEN=seu_token_aqui`
7. Configure os comandos do bot (opcional):
8. Envie `/setcommands` para o BotFather
9. Selecione seu bot
10. Envie a lista de comandos: `start` - Iniciar ou reiniciar a conversa
`help` - Exibir mensagem de ajuda `horario` - Verificar horário de funcionamento `localizacao` - Ver a localização da farmácia
`falar_humano` - Solicitar atendimento humano

Configuração do WhatsApp

1. Crie uma conta no [Facebook Developers](#)
2. Configure o WhatsApp Business API:
3. Crie um aplicativo no Facebook Developers
4. Adicione o produto "WhatsApp" ao seu aplicativo
5. Configure um número de telefone para testes
6. Gere um token de acesso permanente
7. Obtenha o ID do número de telefone
8. Adicione o token e o ID do telefone ao arquivo `.env` :
`WHATSAPP_TOKEN=seu_token_aqui`
`WHATSAPP_PHONE_ID=seu_phone_id_aqui`
`VERIFY_TOKEN=seu_token_de_verificacao_aqui`
9. Configure o webhook:
10. O webhook deve estar acessível via HTTPS
11. Use um serviço como ngrok para testes locais: `bash ngrok http 5000`
12. Configure o webhook no painel do Facebook Developers:
 - URL: `https://seu-dominio.com/webhook`
 - Token de verificação: o mesmo definido em `VERIFY_TOKEN`
 - Eventos: `messages`

Configuração do Banco de Dados

1. Instale o MongoDB (se estiver usando a versão completa):

2. [Instruções de instalação do MongoDB](#)

3. Inicie o serviço MongoDB: `` bash # No Linux sudo systemctl start mongod

No macOS (com Homebrew) brew services start mongodb-community

No Windows # Inicie o serviço MongoDB pelo Gerenciador de Serviços ``

1. Configure a conexão no arquivo `.env`: `MONGODB_URI=mongodb://localhost:27017/ DB_NAME=farmacia_bot`

2. Inicialize os dados de exemplo (opcional):

`bash python data_initializer.py`

Implantação

Implantação Local

Para executar o bot localmente para testes:

Versão sem Banco de Dados

1. Execute o bot do Telegram: `bash python telegram_bot_no_db.py`

2. Para o WhatsApp, execute o servidor Flask: `bash python app.py`

E use ngrok para expor o servidor: `bash ngrok http 5000`

Versão Completa (com Banco de Dados)

1. Certifique-se de que o MongoDB está em execução

2. Execute o bot do Telegram: `bash python telegram_bot.py`

3. Para o WhatsApp, execute o servidor Flask: `bash python app.py`

Implantação em Servidor

Para implantar em um servidor dedicado:

1. Configure o servidor com os requisitos necessários (Python, MongoDB, etc.)

2. Clone o repositório e instale as dependências

3. Configure o arquivo `.env` com as configurações apropriadas

4. Configure um servidor web (como Nginx ou Apache) como proxy reverso:

Exemplo de configuração Nginx: ```nginx server { listen 80; server_name seu-dominio.com;

```
    location / {
        proxy_pass http://localhost:5000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
```

}```

1. Configure o SSL com Let's Encrypt: `bash sudo certbot --nginx -d seu-dominio.com`

2. Configure o bot como um serviço do sistema:

Crie um arquivo `/etc/systemd/system/farmacia-bot.service`: ```[Unit]
Description=Farmacia Bot After=network.target

[Service] User=seu-usuario WorkingDirectory=/caminho/para/farmacia-bot ExecStart=/caminho/para/farmacia-bot/venv/bin/python app.py Restart=always

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

1. Inicie e habilite o serviço: `bash sudo systemctl start farmacia-bot sudo systemctl enable farmacia-bot`

Implantação em Nuvem

Para implantar em serviços de nuvem:

Heroku

1. Crie um arquivo `Procfile` na raiz do projeto: `web: python app.py`

2. Crie um arquivo `runtime.txt`: `python-3.9.7`

3. Implante no Heroku: `bash heroku create farmacia-bot git push heroku main`

4. Configure as variáveis de ambiente no Heroku: `bash heroku config:set TELEGRAM_TOKEN=seu_token_aqui heroku config:set WHATSAPP_TOKEN=seu_token_aqui heroku config:set`

```
WHATSAPP_PHONE_ID=seu_phone_id_aqui heroku config:set  
VERIFY_TOKEN=seu_token_de_verificacao_aqui heroku config:set  
MONGODB_URI=sua_uri_mongodb_aqui
```

AWS

1. Crie uma instância EC2 com Amazon Linux 2
 2. Configure o servidor seguindo as instruções de "Implantação em Servidor"
 3. Configure o grupo de segurança para permitir tráfego HTTP/HTTPS
 4. Configure um Elastic IP para ter um endereço IP estático
 5. Configure o DNS para apontar para o Elastic IP
-

Testes

Teste do Processador de Linguagem Natural

```
python test_bot_no_db.py
```

Teste do Bot do Telegram

1. Inicie o bot do Telegram: `bash python telegram_bot_no_db.py`
2. Abra o Telegram e pesquise pelo nome do seu bot
3. Inicie uma conversa e teste os comandos e mensagens

Teste do Bot do WhatsApp

1. Inicie o servidor Flask: `bash python app.py`
 2. Configure o webhook no Facebook Developers
 3. Envie mensagens para o número de telefone configurado
-

Manutenção

Monitoramento

- Configure logs para monitorar o funcionamento do bot:

```
python import logging logging.basicConfig( filename='bot.log',
level=logging.INFO, format='%(asctime)s - %(name)s - %
(levelname)s - %(message)s' )
```
- Implemente um sistema de alertas para erros críticos

Atualizações

- Mantenha as dependências atualizadas: `bash pip install --upgrade -r requirements.txt`
- Verifique regularmente por atualizações nas APIs do Telegram e WhatsApp

Backup

- Configure backups regulares do banco de dados: `bash mongodump --db farmacia_bot --out /caminho/para/backup`
-

Solução de Problemas

Problemas Comuns

1. **Erro de conexão com o MongoDB**
2. Verifique se o serviço MongoDB está em execução
3. Verifique a URL de conexão no arquivo `.env`
4. Verifique as permissões de acesso ao banco de dados
5. **Webhook do WhatsApp não funciona**
6. Verifique se o servidor está acessível via HTTPS
7. Verifique se o token de verificação está correto
8. Verifique os logs do servidor para erros específicos
9. **Bot do Telegram não responde**

10. Verifique se o token do bot está correto
11. Verifique se o bot está em execução
12. Reinicie o bot e verifique os logs
13. **Erros no processamento de linguagem natural**
14. Verifique se o modelo spaCy foi baixado corretamente
15. Verifique se há erros nos logs relacionados ao NLP
16. Considere treinar o modelo com dados específicos da sua farmácia

Logs e Depuração

- Verifique os logs do aplicativo: `bash tail -f bot.log`
 - Ative o modo de depuração para mais informações: `DEBUG=True`
-

Referências

1. [Documentação do Python-Telegram-Bot](#)
 2. [Documentação da API do WhatsApp Business](#)
 3. [Documentação do spaCy](#)
 4. [Documentação do MongoDB](#)
 5. [Documentação do Flask](#)
-

Autor: Manus AI

Data: 06/06/2025

Versão: 1.0.0