

Jornada Heroica

Visão Geral

O projeto **Jornada Heroica** consiste em uma API desenvolvida em **ASP.NET Core** que implementa um bot conversacional baseado em uma **máquina de estados finitos**, modelada por meio de um **grafo de conversação**. O bot integra-se à **WAHA API** para envio de mensagens e utiliza um **webhook** para receber eventos de mensagens dos usuários.

Os estados e transições da conversa são definidos de forma declarativa no arquivo **appsettings.json**, permitindo alterar o comportamento do bot sem recompilar a aplicação.

Arquitetura Geral

A solução é composta pelos seguintes elementos principais:

- **ASP.NET Core Web API**: responsável por expor o endpoint de webhook.
- **ConversationService**: camada de serviço que processa a lógica da conversa.
- **ConversationGraph**: estrutura que representa o grafo de estados da conversa.
- **WAHA API**: utilizada para envio e recebimento de mensagens via WhatsApp.
- **IMemoryCache**: responsável por manter o estado da conversa de cada usuário.
- **appsettings.json**: arquivo responsável por definir os estados e transições da máquina de estados.
- **Docker / Docker Compose**: responsáveis pela orquestração dos serviços.

Definição do Grafo no appsettings.json

Os estados e transições da conversa são definidos no arquivo **appsettings.json** por meio da seção **ConversationGraph**, que é carregada na inicialização da aplicação:

```
var conversationGraph = builder.Configuration
    .GetSection("ConversationGraph")
    .Get<ConversationGraph>();

if (conversationGraph != null)
{
    builder.Services.AddSingleton(conversationGraph);
    builder.Services.AddScoped<IConversationService, ConversationService>();
}
```

Essa abordagem permite modificar fluxos conversacionais sem necessidade de recompilação.

Modelo de Máquina de Estados

O bot é implementado como uma **máquina de estados finitos**, onde:

- Cada **estado** corresponde a um `ConversationNode`.
- Cada **transição** ocorre por meio das opções informadas pelo usuário ou por uma transição automática (`Next`).
- O **estado atual** do usuário é armazenado em um objeto `ConversationState`.

Formalmente:

- **S**: conjunto de estados.
 - **Σ** : conjunto de entradas.
 - **δ** : função de transição.
 - **s₀**: estado inicial.
-

Gerenciamento de Estado

O estado de cada usuário é armazenado temporariamente em memória utilizando `IMemoryCache`, permitindo múltiplas sessões simultâneas sem interferência.

Integração com a WAHA API

A WAHA API é responsável por:

- Enviar mensagens ao usuário.
- Encaminhar eventos de mensagens para o webhook da aplicação.

A aplicação atua como intermediária, aplicando a lógica da máquina de estados antes de responder ao usuário.

Webhook

O endpoint principal é:

POST /Webhook

Esse endpoint recebe os eventos da WAHA e os encaminha ao serviço de conversação.

Execução do Projeto

Pré-requisitos

Para executar o projeto, é necessário possuir:

- **Docker 20+**
- **Docker Compose v2+**
- Acesso à internet para baixar imagens (quando necessário)

Docker Compose

O projeto é executado por meio do seguinte arquivo `docker-compose.yml`:

```
services:
  chatbot:
    restart: unless-stopped
    build:
      context: .
      dockerfile: ./src/JornadaHeroica.Api/Dockerfile
    ports:
      - 8998:8080
    environment:
      WAHA_API_URL: http://waha:3000

  waha:
    image: devlikeapro/waha
    pull_policy: never
    restart: unless-stopped
    ports:
      - 3627:3000
    environment:
      WHATSAPP_HOOK_URL: http://chatbot:8080/webhook
      WHATSAPP_HOOK_EVENTS: message
      WHATSAPP_HOOK_RETRIES_ATTEMPTS: 1
    volumes:
      - /tbtf/ifmg-waha:/app/.sessions
```

Build e Deploy

Para construir e iniciar os containers, execute:

```
docker compose build
docker compose up -d
```

Ou, em um único comando:

```
docker compose up -d --build
```

Verificação

Após a inicialização:

- A API estará disponível em: <http://localhost:8998> (<http://localhost:8998>).
- O serviço WAHA estará disponível em: <http://localhost:3627> (<http://localhost:3627>).

Os logs podem ser acompanhados com:

```
docker compose logs -f
```

Parada dos Serviços

Para interromper os containers:

```
docker compose down
```

Fluxo de Execução em Produção

1. O usuário envia uma mensagem via WhatsApp.
2. A WAHA recebe a mensagem e dispara o webhook.
3. A API Jornada Heroica processa a mensagem.
4. A resposta é enviada de volta ao usuário pela WAHA API.

Benefícios da Arquitetura

- Total desacoplamento entre definição de fluxo e código.
- Execução reproduzível por meio de containers.
- Facilidade de deploy em qualquer ambiente compatível com Docker.
- Escalabilidade horizontal da API.
- Clareza acadêmica na modelagem por máquina de estados.

Considerações Finais

O **Jornada Heroica** apresenta uma implementação sólida de um bot conversacional baseado em grafos e máquinas de estados finitos, combinando fundamentos teóricos da ciência da computação com práticas modernas de engenharia de software, como containerização e configuração declarativa.

Essa arquitetura torna o projeto adequado tanto para uso institucional quanto para fins acadêmicos, experimentais e didáticos.