Relatório de Execução - Interface para Ollama

Autor: Nicolas Amaral Lobo

Data: 16 de Junho de 2025

1. Objetivo do Exercício

Este relatório detalha a implementação de um servidor web em Python (Flask) e uma interface de usuário (HTML/JS) para interagir com um modelo de linguagem da plataforma Ollama. Os objetivos foram:

1. **Estender um servidor existente:** Adicionar uma nova função para processar um tipo diferente de resposta estruturada (dados de um personagem).
2. **Criar uma interface web:** Desenvolver uma página HTML que permite ao usuário enviar uma pergunta, escolher o tipo de consulta e visualizar a resposta formatada.
3. **Utilizar Template Strings:** Empregar template strings (``) em JavaScript para exibir dinamicamente os dados recebidos do servidor.

2. Arquitetura da Solução

A solução é composta por duas partes principais:

- **Backend (Servidor app.py):**
  - Utiliza a biblioteca Flask para criar um servidor web.
  - Define duas rotas de API (/ask_country e /ask_character) que recebem requisições POST.
  - Usa a biblioteca Pydantic para definir os schemas JSON (Country e Character) que o modelo Ollama deve retornar.
  - Comunica-se com o modelo llama3 através da biblioteca ollama.
  - Serve a interface de usuário através da rota principal (/).
- **Frontend (Interface templates/index.html):**
  - Fornece um formulário simples para o usuário inserir sua pergunta e selecionar o tipo de consulta.
  - Utiliza JavaScript para capturar o envio do formulário.
  - Usa a função fetch para enviar a pergunta do usuário para o endpoint apropriado no backend.
  - Renderiza a resposta JSON recebida em um formato legível usando Template Strings, exibindo o resultado na própria página sem recarregá-la.

3. Instruções para Execução

Pré-requisitos:

- Python 3.7+ instalado.
- Plataforma Ollama instalada e em execução.
- O modelo llama3 (ou outro de sua preferência) baixado no Ollama (ollama pull llama3).

Passos para Execução:

1. Estrutura de Pastas:
   Crie uma pasta para o projeto e, dentro dela, crie uma subpasta chamada templates.
   /seu-projeto/
   |-- app.py
   |-- /templates/
      |-- index.html

2. **Instalar Dependências:** Abra um terminal na pasta do projeto e instale as bibliotecas Python necessárias:
   pip install flask pydantic ollama

3. Executar o Servidor:
   Ainda no terminal, execute o script do servidor:
   python app.py

   O servidor iniciará e estará disponível em http://localhost:5000.
4. Acessar a Interface:
   Abra seu navegador web e acesse o endereço:
   http://localhost:5000

4. Resultados da Execução

Ao acessar a interface, o usuário se depara com um formulário. A imagem abaixo ilustra a tela inicial:

[Imagem da tela inicial com o formulário, campo de texto e botões de rádio para "País" e "Personagem"]

Exemplo de Consulta de País:

1. **Entrada:**
   - **Pergunta:** "Fale sobre o Japão"
   - **Tipo de Consulta:** "País" selecionado.
2. **Ação:** O usuário clica em "Enviar Pergunta".
3. **Saída na Tela:** Após um breve carregamento, a interface exibe os dados

estruturados recebidos do servidor:

Japão
Capital: Tóquio
Moeda: Iene japonês
População: 125.800.000
Idiomas:

- **Japonês**

[Imagem da tela com a resposta formatada para a pergunta sobre o Japão]

Exemplo de Consulta de Personagem:

1. **Entrada:**
   - **Pergunta:** "Descreva o Batman"
   - **Tipo de Consulta:** "Personagem" selecionado.
2. **Ação:** O usuário clica em "Enviar Pergunta".
3. **Saída na Tela:** A interface exibe os dados da nova função:

Batman
Origem: Gotham City
Descrição: Um vigilante mascarado que combate o crime em Gotham City, cuja identidade secreta é o bilionário Bruce Wayne.
Habilidades:

- **Mestre em artes marciais**
- Intelecto genial e habilidades de detetive
- Utilização de tecnologia e equipamentos avançados

[Imagem da tela com a resposta formatada para a pergunta sobre o Batman]

5. Conclusão

O exercício foi concluído com sucesso. A nova função de processamento de resposta foi implementada no servidor, e a interface web se mostrou funcional, capaz de se comunicar com ambos os endpoints e exibir os dados de forma clara e organizada, utilizando template strings em JavaScript como solicitado.