

## Detalhamento do Funcionamento da Aplicação Flask com HTML, CSS e JavaScript

A aplicação utiliza o **Flask** para a parte do backend (servidor em Python), enquanto o frontend é composto por **HTML**, **CSS** e **JavaScript**. A interação do usuário com a página é feita por meio de um formulário, e o backend retorna sugestões de filmes com base no gênero selecionado.

Aqui está uma explicação detalhada do funcionamento de cada arquivo mencionado:

---

### 1. app.py (Backend com Flask)

O arquivo app.py contém o código do servidor Flask. Este código lida com as requisições HTTP, processa as informações e retorna as respostas adequadas para o frontend.

#### Explicação do código:

python

```
from flask import Flask, render_template, request, jsonify
```

```
app = Flask(__name__)
```

```
# Conjunto de dados de filmes
```

```
filmes = [  
    {"nome": "Mad Max", "genero": "Ação"},  
    {"nome": "Toy Story", "genero": "Comédia"},  
    {"nome": "Invocação do Mal", "genero": "Terror"},  
    {"nome": "O Rei Leão", "genero": "Animação"},  
]
```

```
# Rota para a página principal
```

```
@app.route('/')
```

```
def index():
```

```
    return render_template('index.html')
```

```
# Rota para recomendar filmes com base no gênero
```

```
@app.route('/recomendar', methods=['POST'])
```

```
def recomendar():
```

```
    # Obtém o gênero preferido do formulário
```

```

genero_preferido = request.form['genero'].capitalize()

# Filtra os filmes com o gênero escolhido
sugestoes = [f['nome'] for f in filmes if f['genero'] == genero_preferido]

# Retorna a sugestão em formato JSON
if sugestoes:
    return jsonify(sugestoes=sugestoes)
else:
    return jsonify(sugestoes=[])

# Inicializa o servidor Flask
if __name__ == '__main__':
    app.run(debug=True)

```

#### Explicação detalhada:

- **Flask(\_\_name\_\_):** Cria a instância do Flask, que é a base do servidor.
- **@app.route('/'):** Define a rota para a página inicial (/). Quando o usuário acessa a raiz da aplicação, a função `index()` é chamada, que renderiza o template `index.html`.
- **@app.route('/recomendar', methods=['POST']):** Define uma rota POST para a URL `/recomendar`. O método POST é usado quando o formulário é enviado. Essa função processa o gênero selecionado e retorna uma lista de filmes daquele gênero.
- **request.form['genero']:** Obtém o valor do campo "genero" enviado pelo formulário.
- **jsonify(sugestoes=sugestoes):** Retorna a lista de sugestões em formato JSON, que será consumida pelo JavaScript no frontend.
- **app.run(debug=True):** Inicia o servidor Flask em modo de desenvolvimento, com o debug ativado, permitindo recarregar automaticamente as alterações feitas no código.

---

## 2. index.html (Frontend em HTML)

O arquivo `index.html` contém a estrutura da página web e a interface com o usuário. Ele usa o **Flask** para renderizar os dados e o formulário para interação.

#### Explicação do código:

html

```
<!DOCTYPE html>

<html lang="pt-br">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Sistema de Recomendação de Filmes</title>

  <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">

</head>

<body>

  <div class="container">

    <h1>Recomendação de Filmes</h1>

    <p>Escolha seu gênero favorito para receber uma sugestão de filme:</p>

    <form id="form-recomendacao">

      <label for="genero">Gênero:</label>

      <select id="genero" name="genero">

        <option value="Ação">Ação</option>

        <option value="Comédia">Comédia</option>

        <option value="Terror">Terror</option>

        <option value="Animação">Animação</option>

      </select>

      <button type="submit">Obter Recomendação</button>

    </form>

    <div id="sugestoes" class="sugestoes">

      <!-- Resultados serão exibidos aqui -->

    </div>

  </div>

  <script src="{{ url_for('static', filename='script.js') }}"></script>
```

</body>

</html>

#### Explicação detalhada:

- **<form id="form-recomendacao">**: Este formulário contém um campo de seleção (<select>) onde o usuário pode escolher um gênero de filme. O botão "Obter Recomendação" envia os dados ao servidor.
- **<select id="genero" name="genero">**: O campo de seleção permite que o usuário escolha um gênero de filme. O valor do gênero será enviado ao backend quando o formulário for submetido.
- **<div id="sugestoes" class="sugestoes">**: Este <div> é onde as sugestões de filmes serão exibidas, com base na resposta do backend.
- **{{ url\_for('static', filename='style.css') }}**: Usa o método url\_for para referenciar o arquivo CSS que está na pasta static, aplicando estilos à página.
- **{{ url\_for('static', filename='script.js') }}**: Similar ao CSS, o arquivo JavaScript é carregado da pasta static para adicionar interatividade à página.

---

### 3. script.js (Frontend em JavaScript)

O JavaScript é responsável pela interatividade na página. Ele captura o envio do formulário, envia os dados para o servidor e exibe as sugestões de filmes sem recarregar a página.

#### Explicação do código:

javascript

```
document.getElementById("form-recomendacao").addEventListener("submit", function (e) {
```

```
    e.preventDefault(); // Previne o envio do formulário
```

```
    // Obtém o gênero selecionado
```

```
    const generoSelecioneado = document.getElementById("genero").value;
```

```
    // Cria a requisição AJAX para enviar os dados para o backend (Flask)
```

```
    const formData = new FormData();
```

```
    formData.append("genero", generoSelecioneado);
```

```
    fetch("/recomendar", {
```

```

    method: "POST",

    body: formData
  })

  .then(response => response.json())

  .then(data => {

    const sugestaoDiv = document.getElementById("sugestoes");

    if (data.sugestoes.length > 0) {

      sugestaoDiv.textContent = `Recomendamos os seguintes filmes de
      ${generoSelecioneado}: ${data.sugestoes.join(", ")}`;

    } else {

      sugestaoDiv.textContent = "Desculpe, não temos sugestões para esse gênero.";

    }

  })

  .catch(error => {

    console.error('Erro ao fazer a requisição:', error);

  });

});

```

#### Explicação detalhada:

- **e.preventDefault();**: Previne o comportamento padrão do formulário (que seria o envio e recarregamento da página).
- **const generoSelecioneado = document.getElementById("genero").value;**: Captura o gênero selecionado no formulário.
- **fetch("/recomendar", { method: "POST", body: formData })**: Faz uma requisição POST para a URL /recomendar, enviando o gênero selecionado.
- **then(response => response.json())**: A resposta do servidor (em formato JSON) é processada. A lista de sugestões de filmes é extraída.
- **sugestoes.textContent = ...**: Exibe as sugestões de filmes no elemento <div id="sugestoes">. Se não houver sugestões, uma mensagem informando isso será exibida.

---

#### 4. style.css (Estilo para a Página)

O CSS é usado para estilizar a página, tornando-a visualmente atraente e organizada.

#### Explicação do código:

## CSS

```
body{  
  font-family: Arial, sans-serif;  
  background-color: #f4f4f9;  
  margin: 0;  
  padding: 0;  
}
```

```
.container{  
  max-width: 600px;  
  margin: 50px auto;  
  padding: 20px;  
  background-color: #fff;  
  border-radius: 8px;  
  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);  
  text-align: center;  
}
```

```
h1{  
  color: #333;  
}
```

```
label{  
  font-size: 1.1em;  
  margin-bottom: 10px;  
  display: block;  
}
```

```
select{  
  padding: 8px;  
  font-size: 1em;  
  border-radius: 5px;
```

```
border: 1px solid #ccc;
}

button {
  background-color: #4CAF50;
  color: white;
  padding: 10px 20px;
  font-size: 1.1em;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  margin-top: 15px;
}

button:hover {
  background-color: #45a049;
}

.sugestoes {
  margin-top: 20px;
  font-size: 1.2em;
  font-weight: bold;
  color: #444;
}
```

#### Explicação detalhada:

- **body {}**: Define o estilo geral do corpo da página, como fonte e cor de fundo.
  - **.container {}**: Estiliza a caixa que contém o conteúdo da página, definindo margens, bordas arredondadas e sombreamento.
  - **button:hover {}**: Adiciona um efeito de mudança de cor ao passar o mouse sobre o botão.
  - **.sugestoes {}**: Estiliza a área onde as sugestões de filmes são exibidas, tornando o texto mais visível.
-

## Estrutura de Diretórios

A estrutura de arquivos do projeto é a seguinte:

bash

/projeto

```
|
|
|— app.py      # Código Python (Flask)
|
|— templates/
|   └─ index.html  # HTML
|
|— static/
|   └─ style.css   # CSS
|
|   └─ script.js   # JavaScript
```

- **app.py:** Backend em Python com Flask.
- **templates/:** Pasta que contém o arquivo index.html, que é renderizado pelo Flask.
- **static/:** Pasta que contém os arquivos estáticos (CSS e JS), usados pelo frontend.

---

**Resumo:** A aplicação permite que o usuário escolha um gênero de filme e receba recomendações baseadas no seu gênero preferido. O frontend usa AJAX (JavaScript) para interagir com o backend Flask, que processa a requisição e retorna sugestões em formato JSON. O design é simples, com um formulário para escolha do gênero e exibição das sugestões na mesma página.