

RELATÓRIO TÉCNICO – TEMPLATE FLASK 'complaints.html' DO PROJETO FEELING IA

1. Introdução O presente relatório tem como objetivo explicar o funcionamento do arquivo **complaints.html** dentro da arquitetura do projeto **Feeling IA**. Esse arquivo é um **template Flask**, responsável por exibir na interface do site as opiniões e reclamações enviadas pelos usuários. Ele pertence à camada **Frontend dinâmico** do sistema, mas é controlado pelo **Backend Flask**, que envia os dados de forma automatizada.

2. O que é o Flask O **Flask** é um microframework em Python que serve para construir aplicações web. No projeto Feeling IA, ele é o intermediário entre o site (onde o cliente digita a opinião) e o sistema de Inteligência Artificial (que analisa os sentimentos). O Flask recebe os dados dos formulários, salva temporariamente e depois os envia para o template HTML, permitindo que os comentários fiquem visíveis na web.

3. Estrutura do Template 'complaints.html' O arquivo utiliza a linguagem de marcação **Jinja2**, que combina HTML com comandos Python. Os comandos aparecem entre `{% ... %}` e as variáveis dinâmicas entre `{{ ... }}`. O trecho principal é o seguinte:

```
{% extends 'base.html' %} {% block title %}Reclamações Recebidas{% endblock %} {% block content %} {% for c in complaints %} <h5>{{ c.nome }}</h5> <p>{{ c.descricao }}</p> {% endfor %} {% endblock %}
```

Explicação:

- `{% extends 'base.html' %}`: o template herda o layout principal do site.
- `{% block title %}`: define o título da página.
- `{% for c in complaints %}`: cria um loop que exibe cada reclamação enviada.
- `{{ c.nome }}` e `{{ c.descricao }}`: mostram o nome e o texto da reclamação.

4. Relação entre Flask e Web Scraping O arquivo **complaints.html** é gerado pelo Flask e exibe todas as opiniões enviadas. Já o módulo de **Web Scraping** é um script separado que acessa essa página e coleta as informações automaticamente. Fluxo completo: O usuário envia uma opinião através do formulário (Frontend). O Flask salva e exibe a opinião na página **/complaints**. O script de Web Scraping acessa essa página e extraí as tags `<h5>` e `<p>` com **BeautifulSoup**. Os dados coletados são enviados ao módulo de **Análise de Sentimentos**. Os resultados retornam em formato **JSON** para o painel do projeto.

5. Importância do Template O arquivo **complaints.html** é essencial, pois funciona como o ponto de integração entre o Flask e o Web Scraping. Ele apresenta os dados de forma estruturada, permitindo que o algoritmo de raspagem identifique facilmente os elementos HTML correspondentes aos comentários e autores.

6. Conclusão Portanto, o arquivo **complaints.html** faz parte do **Flask**, e não do Web Scraping. Ele atua como uma página gerada dinamicamente, exibindo os dados que o sistema coleta. O Web Scraping apenas lê o que o Flask disponibiliza nessa página. Essa divisão de funções garante clareza na arquitetura do projeto **Feeling IA**, separando o que é **exibição** (Flask/HTML) do que é **coleta e análise** (Python/IA).