

PRIMEIRO: O QUE ESSE ARQUIVO É?

👉 Ele é uma página do site feita em HTML + Jinja2.

Mas o que é Jinja2?

É um sistema que permite colocar dados do Python dentro do HTML.

Esse arquivo NÃO é:

- Python
- JavaScript
- Banco
- Backend

É **HTML misturado com comandos especiais** que o Flask usa.



2. PARA QUE ELE SERVE?

Ele serve para **mostrar a lista de reclamações** enviadas pelos usuários.

Ou seja:

- Alguém envia uma reclamação no formulário
- O Flask salva a reclamação
- O Flask monta uma lista de reclamações
- O Flask envia essa lista para este arquivo
- Este arquivo exibe as reclamações na tela

Pronto.

Esse é o papel dele.



3. EXPLICAÇÃO COM ANALOGIA (para iniciantes)

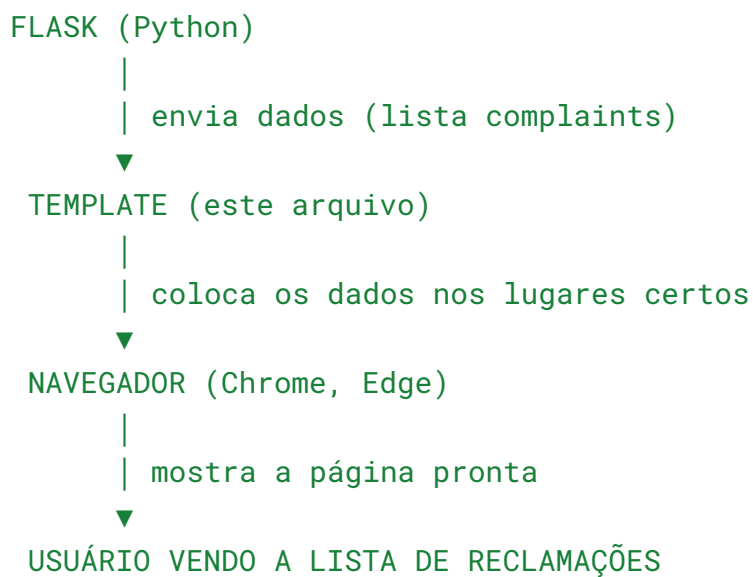
Imagine assim:

- **Python/Flask** = a cozinha
- **A lista de reclamações** = a comida
- **Este arquivo HTML** = o prato vazio
- **Jinja2** = a colher que coloca a comida no prato

Então:

- 👉 O Flask COZINHA os dados
 - 👉 O Jinja2 SERVE esses dados dentro do HTML
 - 👉 O navegador do usuário MOSTRA o prato pronto
-

4. DESENHO SUPER SIMPLES



5. E COMO ELE FAZ ISSO?

Ele faz isso com estas três ideias básicas:

- ✓ **1. Ele usa o layout principal do site**

```
{% extends 'base.html' %}
```

Significa:

👉 “Use o template base.html como molde.”

Então:

- o menu azul
- o rodapé
- os scripts
- o CSS

Tudo isso já vem pronto do `base.html`.

✓ 2. Ele diz onde colocar o conteúdo da página

```
{% block content %}  
    ... conteúdo aqui ...  
{% endblock %}
```

Isso fala para o Flask:

👉 “Este é o conteúdo específico desta página.”

No seu `base.html`, tem isso:

```
{% block content %}{% endblock %}
```

É ali que tudo aparece.

✓ 3. Ele mostra cada reclamação usando um loop

```
{% for c in complaints %}  
    ... usa c.nome, c.descricao, c.email ...  
{% endfor %}
```

Esse é o **coração** da página.

Explicação simples:

👉 Para cada reclamação c dentro da lista complaints, repita esse bloco.

Se tem 5 reclamações → ele cria 5 cartões.

Se tem 20 reclamações → 20 cartões.

Se tem zero → cai no ELSE e mostra a frase:

Nenhuma reclamação registrada até o momento.

6. AGORA A EXPLICAÇÃO DO CÓDIGO POR TRECHOS (SUPER DIDÁTICO)

Eu vou explicar como se fosse um professor explicando devagar:

A) Ele começa dizendo que usa o layout principal:

```
{% extends 'base.html' %}
```

Significa:

👉 “Pegue o base.html e coloque meu conteúdo dentro dele.”

B) Ele muda o título da aba:

```
{% block title %}Reclamações Recebidas{% endblock %}
```

Isso muda o nome que aparece lá em cima no navegador.

■ C) Ele inicia o conteúdo da página:

```
{% block content %}
```

Tudo que vem aqui dentro aparece no corpo da página.

■ D) Ele cria uma seção bonita e centralizada:

```
<section class="mx-auto" style="max-width: 900px;">
```

- `mx-auto` = centraliza
 - `max-width 900px` = deixa a página mais estreita (mais elegante)
-

■ E) Ele coloca o título visual:

```
<h2 class="mb-4 text-center fw-semibold">Reclamações Recebidas</h2>
```

■ F) Ele verifica se EXISTEM reclamações:

```
{% if complaints %}
```

- **Se tem reclamações** → entra aqui
 - **Se não tem** → pula para o ELSE
-

■ G) Ele cria uma lista bonita:

```
<div class="list-group">
```

Bootstrap deixa tudo organizado.

H) Ele faz um loop:

```
{% for c in complaints %}
```

Aqui acontece a mágica.

Pra cada reclamação, ele:

- cria um cartão
 - mostra o nome
 - mostra a data
 - mostra a descrição
 - mostra o email
-

I) Dentro do loop:

```
<h5>{{ c.nome }}</h5>
```

Mostra o nome.

```
{{ c.created_at[:10] }}
```

Mostra a data.

```
{{ c.descricao }}
```

Mostra o texto.

```
{{ c.email }}
```

Mostra o email.

J) Se NÃO tiver reclamações:

```
{% else %}
```

Mostra:

```
<p>Nenhuma reclamação registrada até o momento.</p>
```

K) Termina tudo:

```
{% endif %}  
{% endblock %}
```

7. RESUMO FINAL PRA VOCÊ NUNCA MAIS ESQUECER

O que ele é?

Um arquivo HTML com comandos Jinja2.

Pra que serve?

Para mostrar todas as reclamações já enviadas.

Quem manda os dados pra ele?

O Flask (Python).

O que ele faz exatamente?

Ele pega cada reclamação e cria um cartão bonito na tela.

```
{# --
----- #}
{# ESTE ARQUIVO É UMA PÁGINA DO SITE (TEMPLATE) QUE MOSTRA #}
{# TODAS AS RECLAMAÇÕES QUE JÁ FORAM ENVIADAS PELOS USUÁRIOS. #}
{# ELE NÃO É PYTHON, NÃO É FLASK. ELE É HTML + JINJA2. #}
{# O FLASK USA ESTE ARQUIVO PARA MONTAR A TELA QUE O USUÁRIO VÊ. #}
{# ----- #}
```

```
{% extends 'base.html' %}
{# "extends" significa "estender", "usar junto".
Ele diz: "Use o arquivo base.html como molde".
Isso quer dizer que:
• o cabeçalho (menu)
• o rodapé
• os estilos (Bootstrap)
já vêm prontos de lá.
```

Esta página só coloca seu próprio conteúdo no meio.

```
#}
```

```
{% block title %}Reclamações Recebidas{% endblock %}
{# Aqui mudamos o título da aba do navegador SÓ desta página.
No arquivo base.html existe um bloco reservado para isso.
Então este texto aparece lá no <title>.
#}
```

```
{% block content %}
{# Tudo daqui até o {% endblock %} será colocado no "miolo"
da página, dentro do template base.html.
É aqui que vai todo o conteúdo que aparece na tela.
#}
```

```
<section class="mx-auto" style="max-width: 900px;">
{# <section> é só uma caixa para organizar conteúdo.
mx-auto = centraliza essa caixa na tela.
max-width: 900px = limite de largura para ficar bonito.
#}
```

```
<h2 class="mb-4 text-center fw-semibold">Reclamações Recebidas</h2>
{# <h2> é um título grande.
mb-4 = espaço embaixo
text-center = centralizado
fw-semibold = semi-negrito
```

Esse é o título que aparece lá no topo da página.

```
#}
```



```
{% if complaints %}
{# "complaints" é uma lista que vem do Python.
O Flask envia essa lista para cá.
Aqui estamos perguntando:
"Tem reclamações? A lista tem alguma coisa?"
Se sim → entra no IF
Se não → mostra a mensagem do ELSE.
#}
```

```
<div class="list-group">
{# Bootstrap: "list-group" organiza a lista de forma bonita.

Cada reclamação vai virar um "item da lista".
#}
```

```
{% for c in complaints %}
{# "for" é um loop do Jinja (igual ao Python).
Aqui significa:
"Para cada reclamação c dentro da lista complaints,
repita o código abaixo".
Ou seja, se tiverem 5 reclamações,
tudo abaixo será repetido 5 vezes.
#}
```

```
<div class="list-group-item list-group-item-action flex-column align-items-start mb-2
rounded shadow-sm">
```

```
{# Este é o cartão da reclamação.
Cada reclamação vira um cartão bonito assim:
```

Nome	Data	
reclamação do cliente		
email do cliente		

```
#}
```

```
<div class="d-flex justify-content-between align-items-center mb-1">
```

```
{# d-flex = organiza em linha
justify-content-between = um item na esquerda, outro na direita
align-items-center = alinha tudo no meio
Isso aqui cria a linha com:
Nome da pessoa | Data
#}
```

```
<h5 class="mb-0">{{ c.nome }}</h5>
```

```
{# {{ c.nome }} mostra o nome da pessoa.
Isso vem do Python.
```

```
Se no Python c.nome = "João",
```

na tela aparece "João".

#}

<small class="text-muted">{{ c.created_at[:10] }}</small>

{# c.created_at é uma data completa, tipo:

"2025-11-10 14:30:00"

[:10] pega só os 10 primeiros caracteres:

"2025-11-10"

Ou seja:

Mostra só a DATA sem a hora.

#}

</div>

<p class="mb-1">{{ c.descricao }}</p>

{# Aqui aparece a reclamação em si:

"O atendimento foi ruim", etc.

#}

<small class="text-muted fst-italic">{{ c.email }}</small>

{# Aqui mostra o email da pessoa.

É opcional no seu sistema.

#}

</div>

{# FIM DO CARTÃO DE UMA RECLAMAÇÃO #}

{% endfor %}

{# FIM DO LOOP

Depois disso, todas as reclamações já foram exibidas.

#}

</div>

{% else %}

{# Se NÃO existir nenhuma reclamação #}

<p class="text-center fs-6 text-muted">

Nenhuma reclamação registrada até o momento.

</p>

{# Mensagem amigável quando a lista estiver vazia #}

{% endif %}

{# FIM DO IF #}

</section>

{% endblock %}

{# Aqui termina o conteúdo específico desta página.

O base.html vai colocar tudo isso no lugar certo.
#}