

```
from flask import Flask, render_template, request, make_response,
redirect, url_for

app = Flask(__name__)

@app.route("/")

def index():

    # Renderiza a página inicial com os links para votar e ver
    resultado

    return render_template('index.html')

@app.route("/votar", methods=['GET', 'POST'])

def votar():

    if request.method == 'GET':

        return render_template('votar.html')

    nome = request.form.get("nome") # coleta o nome digitado

    genero = request.form.get("genero")

    # Armazena o nome num cookie com validade de 5 minutos (60s x 5 =
    300s)

    response = make_response(redirect(url_for('resultado',
    genero=genero)))

    response.set_cookie('nome', nome, max_age=300)

    return response

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

```
def resultado():

    genero = request.args.get("genero", "não informado") # pega o gênero
da URL (string de consulta)

    nome = request.cookies.get("nome", "Visitante") # tenta pegar o nome
do cookie (ou "Visitante")

    return render_template('resultado.html', genero=genero, nome=nome)
```



Começo do arquivo

from flask import Flask, render_template, request, make_response, redirect, url_for

● Aqui você está **importando ferramentas do Flask**:

- **Flask**: para criar o app.
- **render_template**: para mostrar as páginas HTML.
- **request**: para acessar dados enviados pelo usuário (formulários, query strings).
- **make_response**: para montar uma resposta HTTP personalizada (com cookies, por exemplo).
- **redirect**: para redirecionar o usuário para outra rota.
- **url_for**: para gerar a URL de uma função automaticamente.

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

- ♦ Cria a **instância da aplicação Flask**.
- ♦ **__name__** serve para o Flask saber onde está o arquivo principal da aplicação.



Rota inicial: /

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

```
def index():
    # Renderiza a página inicial com os links para votar e ver resultado
    return render_template('index.html')
```

- ♦ Essa função é chamada quando o usuário acessa / (a página principal).
 - ♦ Ela **renderiza o template HTML `index.html`**, que tem os links para votar e ver resultado.
-

Rota **/votar**: GET e POST

```
@app.route("/votar", methods=['GET', 'POST'])
```

- ♦ Define a rota **/votar**, que aceita dois métodos:
 - **GET**: quando o usuário **abre a página de votação**.
 - **POST**: quando o usuário **envia o formulário**.
-

```
def votar():
    if request.method == 'GET':
        return render_template('votar.html')
```

- ♦ Se o método for **GET**, mostramos o formulário de votação (**votar.html**).
 - ✂ Isso acontece quando a pessoa **ainda não votou**.
-

```
nome = request.form.get("nome") # coleta o nome digitado
genero = request.form.get("genero")
```

- ♦ Aqui tratamos o envio do formulário (método POST).
 - ♦ **request.form.get("nome")** pega o que a pessoa digitou no campo de nome.
 - ♦ **request.form.get("genero")** pega o gênero escolhido (valor do botão "radio").
-

```
response = make_response(redirect(url_for('resultado', genero=genero)))
```

- ♦ Criamos uma **resposta personalizada** que redireciona para a rota **/resultado**.
- ♦ **url_for('resultado', genero=genero)** monta uma URL assim:

/resultado?genero=acao

📌 O gênero vai como **string de consulta** na URL.

```
response.set_cookie('nome', nome, max_age=300)
```

- ♦ Armazena o **nome do usuário em um cookie** chamado '**nome**'.
- ♦ **max_age=300** define que ele dura **5 minutos** (300 segundos).

```
return response
```

- ♦ Envia a resposta montada (com o redirecionamento e o cookie).



Rota **/resultado**

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

```
def resultado():
```

- ♦ Essa função é chamada quando a pessoa acessa **/resultado**.

```
genero = request.args.get("genero", "não informado") # pega o gênero da URL
```

- ♦ **request.args.get()** pega **valores da URL** (ex: **?genero=acao**).
- ♦ Se não tiver nada, mostra “não informado”.

```
nome = request.cookies.get("nome", "Visitante") # tenta pegar o nome do cookie
```

- ♦ Busca o nome no **cookie**.
- ♦ Se o cookie expirou ou nunca existiu, usa “Visitante”.

```
return render_template('resultado.html', genero=genero, nome=nome)
```

- ♦ Renderiza a página **resultado.html**, passando o nome e o gênero como **variáveis** para o template.

🥰 Resumo com amor

📁 Esse código cria:

1. Uma **página inicial** com links.
2. Uma **página de votação**, que:
 - coleta nome e gênero,
 - salva o nome num cookie,
 - redireciona para `/resultado?genero=...`
3. Uma **página de resultado**, que:
 - lê o gênero da URL,
 - lê o nome do cookie,
 - mostra a mensagem personalizada.

Se quiser, posso te ajudar a:

- salvar os votos num arquivo `.txt` ou `.csv`;
- contar votos por gênero;
- mostrar resultados com gráficos lindos;
- estilizar as páginas com Bootstrap!

```
<!DOCTYPE html>

<html lang="pt-br">

<head>

  <meta charset="UTF-8">
```

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

<title>Votar</title>

</head>

<body>

<h1>Votar</h1>

<form action="{{url_for('votar')}}" method="post">

    <label for="nome">

        Nome:

        <input type="text" name="nome" placeholder="nome">

    </label>

    <br><br>

    <label>

        <input type="radio" name="genero" value="acao">

        Ação

    </label><br>

    <label>

        <input type="radio" name="genero" value="comedia">

        Comédia

    </label><br>

    <label>

        <input type="radio" name="genero" value="drama">

        Drama

    </label><br>
```

```
        <button type="submit">Enviar voto</button>

    </form>

    <br>

    <a href="{url_for('index')}">Voltar para página inicial</a>

</body>

</html>
```

<!DOCTYPE html>

<html lang="pt-br">

♦ Essa é a **declaração do tipo do documento**. Diz ao navegador que é um documento HTML5.

♦ `lang="pt-br"` define o idioma principal como **português do Brasil**.

<head>

<meta charset="UTF-8">

♦ Define o **conjunto de caracteres** como UTF-8, permitindo acentuação, "ç", etc.

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

♦ Isso garante que a página seja **responsiva**, ou seja, funcione bem em celular, tablet e PC.

<title>Votar</title>

</head>

♦ O `<title>` define o **título da aba do navegador**, que aparecerá como "Votar".

```
<body>
```

```
    <h1>Votar</h1>
```

- ♦ Começa o corpo da página.
 - ♦ `<h1>` é um **título principal** visível na página, bem grande.
-

```
<form action="{{url_for('votar')}}" method="post">
```

- ♦ Cria um formulário HTML.
 - ♦ `action="{{ url_for('votar') }}"` indica que, quando o formulário for enviado, ele será mandado para a **rota /votar** definida no Flask.
 - ♦ `method="post"` diz que os dados serão enviados **de forma escondida**, no corpo da requisição (não aparecem na URL).
-

```
    <label for="nome">
```

```
        Nome:
```

```
        <input type="text" name="nome" placeholder="nome">
```

```
    </label>
```

- ♦ Um campo de **texto** para o usuário digitar o nome.
 - ♦ `name="nome"` é importante: é como o Flask vai pegar esse valor com `request.form['nome']`.
 - ♦ `placeholder="nome"` mostra um texto dentro da caixinha antes do usuário digitar.
-

```
    <br><br>
```

- ♦ Cria **espaços em branco** (quebra de linha) para separar os campos.
-

```
    <label>
```



```
<input type="radio" name="genero" value="acao">
```

Ação

```
</label><br>
```

- ♦ Cria um **botão de opção (radio)** com o valor "acao".
 - ♦ `name="genero"` faz com que todos esses botões façam parte do **mesmo grupo** — o usuário só pode escolher **um gênero**.
 - ♦ `value="acao"` é o valor que será enviado se o usuário escolher esse botão. O Flask vai acessar isso com `request.form['genero']`.
-

```
<label>
```

```
<input type="radio" name="genero" value="comedia">
```

Comédia

```
</label><br>
```

```
<label>
```

```
<input type="radio" name="genero" value="drama">
```

Drama

```
</label><br>
```

- ♦ Os outros dois botões de opção: um para "comedia" e outro para "drama". Funcionam exatamente como o primeiro.
-

```
<button type="submit">Enviar voto</button>
```

- ♦ Um **botão que envia o formulário**. Quando clicado, o nome e o gênero escolhidos vão para a rota `/votar` usando método POST.
-

```
</form>
```

```
<br>
```

```
<a href="{{url_for('index')}}">Voltar para página inicial</a>
```

```
</body>
```

```
</html>
```

- ♦ Um **link** para voltar à página inicial (a rota `/`).
 - ♦ Usa `url_for('index')` para que o Flask gere a URL automaticamente, sem você ter que escrever `/`.
-

🌟 Resumo fofo e útil:

Esse código cria uma página onde:

- a pessoa digita o nome,
 - escolhe o gênero do filme,
 - e envia os dados para o Flask tratar no `/votar`.
-

```
<!DOCTYPE html>

<html lang="pt-br">

<head>

    <meta charset="UTF-8">

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

    <title>Resultado</title>

</head>

<body>

    <h1>Resultado da Votação</h1>

    <p>Você votou no gênero: <strong>{{ genero }}</strong></p>

    <p>Obrigado por votar, {{ nome }}!</p>
```

```
<a href="{url_for('index')}">

    <button>Voltar para o início</button>

</a>

</body>

</html>
```

<!DOCTYPE html>

<html lang="pt-br">

- ♦ Declara que o tipo do documento é **HTML5**.
- ♦ `lang="pt-br"` diz que o conteúdo está em **português do Brasil** (útil para acessibilidade, dicionários, leitores de tela etc).

<head>

 <meta charset="UTF-8">

- ♦ Define a **codificação de caracteres UTF-8**, permitindo acentos e símbolos como "ç".

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

- ♦ Torna a página **responsiva**, ou seja, ajusta automaticamente o tamanho e a escala para funcionar bem em celulares e tablets.

 <title>Resultado</title>

</head>

- ♦ Define o **título da aba do navegador** como “Resultado”.

<body>

```
<h1>Resultado da Votação</h1>
```

- ♦ Abre o corpo da página.
- ♦ `<h1>` mostra um **título grande** visível: “Resultado da Votação”.

```
<p>Você votou no gênero: <strong>{{ genero }}</strong></p>
```

- ♦ Essa linha mostra o gênero que a pessoa votou.
- ♦ `{{ genero }}` é uma **variável do Flask**, passada do `app.py` para o template.
- ♦ `strong` deixa o texto em **negrito**.

👉 Exemplo de resultado real na tela:

Você votou no gênero: Comédia

```
<p>Obrigado por votar, {{ nome }}!</p>
```

- ♦ Mostra uma mensagem de agradecimento usando o nome da pessoa.
- ♦ `{{ nome }}` vem do **cookie** guardado no navegador.

👉 Se a pessoa se chama Ana:

Obrigado por votar, Ana!

👉 Se o cookie expirou:

Obrigado por votar, Visitante!

```
<a href="{{url_for('index')}}">
```

```
    <button>Voltar para o início</button>
```

```
</a>
```

- ♦ Cria um botão que leva de volta à **página inicial** (/).
 - ♦ `url_for('index')` gera automaticamente o link da função `index()` do Flask.
 - ♦ Isso evita que você precise escrever o caminho manualmente, e funciona mesmo se mudar a rota futuramente.
-

`</body>`

`</html>`

- ♦ Fecha o corpo e o documento HTML.
-



Resumo fofo e útil:

Esse template:

- Exibe o gênero escolhido,
- Mostra o nome da pessoa (ou "Visitante" se não tiver),
- E oferece um botão para voltar pra página inicial.