08

Gerando URLs dinâmicas

Transcrição

Com o nosso template funcionando, conseguimos resolver o problema da repetição de códigos. Agora, trabalharemos outro problema.

Atualmente, estamos com 3 arquivos HTML, mas no futuro poderemos ter muito mais, inclusive com mais pastas dentro da pasta "template".

Para acessar o CSS, estamos utilizando a URL `../static/bootstrap.css`. Isso é bom, pois estamos usando um caminho relativo para acessar o Bootstrap: utilizamos `..` para sair da pasta "template" e acessar a pasta superior, para então entrarmos na pasta "static", onde encontramos o arquivo `bootstrap.css`.

Se tivéssemos uma hierarquia muito grande, teríamos que tomar o cuidado de alterar esse caminho em cada um dos arquivos. Além disso, se mudássemos o `template.html` para uma outra pasta, por exemplo, se criássemos uma pasta somente para os *templates* que são herdados, teríamos que mudar esse caminho. Por essa razão, não é uma boa prática dependermos desse caminho.

A boa notícia é que o Flask, como um servidor, conhece as URLs que ele consegue servir. Portanto, podemos simplesmente instruí-lo a gerar dinamicamente a URL para nossa pasta de estáticos e pegar o arquivo indicado.

Para isso, o Flask nos fornece uma função chamada `url_for()`, que utilizaremos no nosso `template.html` por meio da diretiva do Flask (`{{ }}` , passando o nome do arquivo que estamos buscando (`filename-'bootstrap.css'):

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Jogoteca</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='bootstrap.css') }}">
```

Com essa alteração, nossas páginas continuarão funcionando normalmente no navegador. Porém, agora estamos gerando a URL do CSS dinamicamente, em vez de codificá-lo rigidamente.

E agora? Bom, temos uma aplicação na qual é possível criar novos jogos em uma lista. Mas não queremos que qualquer pessoa possa acessar nosso sistema e fazer isso, somente aqueles que possuam um usuário. Começaremos a implementar essa funcionalidade nos próximos vídeos.