

Adicionando conteúdo dinâmico

Transcrição

Até agora, nós conseguimos criar uma página, renderizá-la e exibi-la pelo Flask no nosso servidor. Porém, ainda falta dinâmica no conteúdo dessa página, e é nisso que trabalharemos agora.

Primeiramente, vamos fazer com que o título seja dinâmico, e mais tarde trabalharemos com os jogos. No nosso HTML, temos `<h1>Jogoteca</h1>` .

Para que o conteúdo de um HTML seja dinâmico, podemos utilizar uma diretiva do Flask, muito parecida com a de outras linguagens, como o Ruby e outros frameworks do Python, e que nos permite acessar uma variável do servidor — no caso, uma variável chamada "título".

Para isso, escreveremos `<h1>{{ titulo }}</h1>` . Então, no arquivo Python, precisaremos passar mais informações para o `render_template()` , que é justamente a variável que acessaremos na *view*. Para definirmos o valor de `titulo` , passaremos a string `Jogos` :

```
from flask import Flask, render_template

app = Flask(__name__)

@app.route('/inicio')
def ola():
    return render_template('lista.html', titulo='Jogos')

app.run()
```

Rodando novamente nossa aplicação e acessando a URL <http://127.0.0.1:5000/inicio> (<http://127.0.0.1:5000/inicio>), teremos:

Jogos

Nome

God of War

Pokemon Silver

Super Mario RPG

Pronto, nosso conteúdo é dinâmico!

Vamos entender um pouco melhor o que está acontecendo?

Dentre as várias *libs* utilizadas pelo Flask, temos a chamada **Jinja2**, responsável por fazer com que a *engine* de *templates* funcione. Ou seja, permite renderizar conteúdo dinâmico em uma página HTML.

Falaremos sobre isso mais detalhadamente, adiante. Por enquanto, faça os exercícios e teste a execução desse código na sua máquina. Até já!