

Atualizando dados no servidor

Transcrição

Agora vamos implementar a lógica de atualização dos dados no servidor por meio da rota `/atualizar`. Como vimos anteriormente, essa função será muito parecida com a `criar()`.

```
@app.route('/atualizar', methods=['POST',])
def atualizar():
    nome = request.form['nome']
    categoria = request.form['categoria']
    console = request.form['console']
    jogo = Jogo(nome, categoria, console)
    jogo_dao.salvar(jogo)
    return redirect(url_for('index'))
```

Nesse código, estamos recebendo `nome`, `categoria` e `console`. Seria ideal que o formulário também nos passasse a informação `id`. Assim, poderíamos passar para a variável `jogo` um `id=id`.

Para isso, precisamos primeiramente colocar essa informação no formulário - lembrando que, como o `id` não é uma informação que poderá ser alterada pelo usuário, ela não precisa ficar visível para ele.

Por isso, criaremos um campo escondido (`input type="hidden"`). Quando fazemos um `POST`, a aplicação procura um `name` para associar ao campo do formulário - nesse caso, `id`. Também precisamos passar o valor que deverá ficar armazenado nesse campo. Como estamos passando para o `template` `editar.html` um `jogo` diretamente do banco, podemos passar o `id` desse jogo (`jogo.id`) para esse `input`.

```
{% extends "template.html" %}
{% block conteudo %}
    <form action="{{ url_for('atualizar') }}" method="post">
        <fieldset>
            <input type="hidden" name="id" value="{{ jogo.id }}">
```

Com esse dado armazenado, poderemos acessá-lo do servidor com `request.form['id']`.

```
@app.route('/atualizar', methods=['POST',])
def atualizar():
    nome = request.form['nome']
    categoria = request.form['categoria']
    console = request.form['console']
    jogo = Jogo(nome, categoria, console, id=request.form['id'])
    jogo_dao.salvar(jogo)
    return redirect(url_for('index'))
```

Em seguida, estamos chamando a função `salvar()` do objeto `jogo_dao`. Se existe um `id`, essa função faz um `atualizar` (`SQL_ATUALIZA_JOGO`); se não, ela cria um jogo novo (`SQL_CRIA_JOGO`). Dessa forma, não precisamos alterar mais nada.

Acessando qualquer URL de edição, por exemplo <http://127.0.0.1:5000/editar/4> (<http://127.0.0.1:5000/editar/4>), poderemos alterar os campos e salvar as informações diretamente no servidor. Mas e se quiséssemos deletar um jogo? Veremos isso daqui a pouco. Até lá!