

Deletando jogos

Transcrição

Até o momento conseguimos criar, editar e listar os dados do nosso banco, e agora queremos implementar uma forma de deletá-los. Uma maneira de fazermos isso é criando um link para essa ação e passando o `id` do jogo, da mesma forma que fizemos para "Editar" - inclusive, se quisermos editar o oitavo item na lista, o respectivo link será <http://127.0.0.1:5000/editar/8> (<http://127.0.0.1:5000/editar/8>).

Para isso, adicionaremos um novo link em `lista.html`, que chamaremos de "Deletar":

```
{% extends "template.html" %}
{% block conteudo %}
    <table class="table table-striped table-responsive table-bordered">
        <thead class="thead-default">
            <tr>
                <th>Nome</th>
                <th>Categoria</th>
                <th>Console</th>
                <th></th>
            </tr>
        </thead>
        <tbody>
            {% for jogo in jogos %}
                <tr>
                    <td>{{ jogo.nome }}</td>
                    <td>{{ jogo.categoria }}</td>
                    <td>{{ jogo.console }}</td>
                    <td><a href = "{{ url_for('editar', id=jogo.id) }}">Editar</a>
                    <a href = "{{ url_for('deletar', id=jogo.id) }}">Deletar</a></td>
                </tr>
            {% endfor %}
        </tbody>
    </table>
{% endblock %}
```

Em seguida, criaremos a rota `/deletar` no projeto **jogoteca**. Essa rota receberá o parâmetro `id` na URL (`deletar/<int:id>`), que será passado para a lista de argumentos da função `deletar()`.

Para deletarmos um objeto do banco, podemos simplesmente chamar o método `delegar()` de `jogo_dao`, que recebe justamente o `id`. Depois de deletado o objeto, precisamos renderizar alguma outra tela, que pode ser a própria tela de lista recarregada (`return redirect(url_for('index'))`). Além disso, usando o `flash()`, mostraremos a mensagem "O jogo foi removido com sucesso!".

```
@app.route('/deletar/<int:id>')
def deletar(id):
    jogo_dao.deletar(id)
    flash('O jogo foi removido com sucesso!')
    return redirect(url_for('index'))
```

Atualizando a página da nossa aplicação, um novo "Deletar", completamente funcional, aparecerá ao lado de "Editar". Com mais essa função, nossa aplicação está cada vez mais interessante. Mas você consegue identificar um problema?

Para acessarmos a função "novo", por exemplo, temos que digitar a rota diretamente na barra de endereço. Nessa página, e também na página "Editar", temos que completar a operação se quisermos ser enviados de volta ao index, já que não criamos uma maneira de voltar à essa página. Ou seja, nossa usabilidade ainda não está muito boa.

O primeiro passo para solucionarmos esse problema é a criação de um botão para <http://127.0.0.1:5000/novo> (http://127.0.0.1:5000/novo) na tela "index", o que podemos fazer facilmente. Esse botão estará localizado no topo da página `lista.html`, antes da tabela:

```
{% extends "template.html" %}
{% block conteudo %}
    <a class="btn btn-primary" href="{{ url_for('novo') }}">Novo jogo</a>
    <table class="table table-striped table-responsive table-bordered">
        <thead class="thead-default">
            <tr>
                <th>Nome</th>
                <th>Categoria</th>
                <th>Console</th>
                <th></th>
            </tr>
```

Como estamos utilizando o **Bootstrap**, podemos usar `class=btn btn-primary` para criar um botão mais estilizado:

Já na tela "Novo", depois do formulário, adicionaremos um botão vermelho (`btn-danger`) que retornará o usuário para a página "index":

```
{% extends "template.html" %}
{% block conteudo %}
    <form action="{{ url_for('atualizar') }}" method="post">
        <fieldset>
            <input type="hidden" name="id" value="{{ jogo.id }}">
            <div class="form-group">
                <label for="nome">Nome</label>
                <input type="text" id="nome" name="nome" class="form-control" value="{{ jogo.nome }}">
            </div>
            <div class="form-group">
                <label for="categoria">Categoria</label>
                <input type="text" id="categoria" name="categoria" class="form-control" value="{{ jogo.categoria }}">
            </div>
            <div class="form-group">
                <label for="console">Console</label>
                <input type="text" id="console" name="console" class="form-control" value="{{ jogo.console }}">
            </div>
            <button type="submit" class="btn btn-primary btn-salvar">Salvar</button>
            <a class="btn btn-danger" href="{{ url_for('index') }}">Voltar</a>
        </fieldset>
    </form>
{% endblock %}
```

Novo jogo

Nome

Categoria

Console

Salvar

Voltar

Está ficando legal, não? Mas não seria ainda mais legal se tivéssemos imagens dos jogos da lista? Nos próximos vídeos veremos como o Flask nos ajuda com isso. Até lá!