



Flask

Programação Web III

Prof. Diego Max

Flask

Tópico 04.4: Alterando dados

CRUD com o banco de dados MySQL

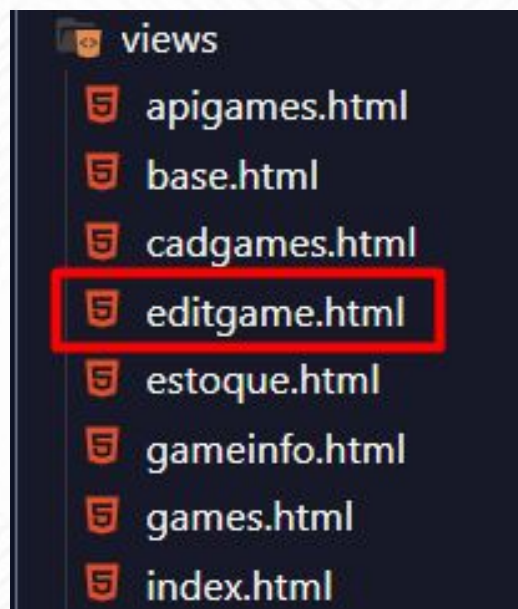
Alterando dados no banco



Para finalizarmos nosso CRUD, implementaremos agora a alteração de informações no banco de dados.

Para isso criaremos uma nova view com o nome “**editgame.html**”.

Essa view será responsável por conter um formulário na qual resgataremos as informações de um determinado registro e realizaremos as devidas alterações.



Alterando dados no banco



Antes de editarmos o arquivo “**editgame.html**”, abriremos o arquivo “**routes.py**”, nele criaremos a rota de edição “**/edit**” que receberá como **parâmetro** a **ID** do game que deseja-se editar.

Feito a seleção do game no banco através do método “**Game.query.get(id)**”, suas informações serão enviadas para a página “**editgame.html**” para serem mostradas no formulário de edição.

```
# CRUD - EDIÇÃO

@app.route('/edit/<int:id>', methods=['GET', 'POST'])
def edit(id):
    g = Game.query.get(id)
    return render_template('editgame.html', g=g)
```

Alterando dados no banco



Agora, já podemos criar o formulário de edição na página “**editgame.html**”:

```
{% extends 'base.html' %}
{% block content %}
<div class="col-md-6 m-auto">
  <h3 style="text-align: center;">Editar jogo:</h3><br>
  <form action="{{url_for('edit', id=g.id)}}" method="POST">
    <input type="text" name="titulo" class="form-control" value="{{g.titulo}}" required>
    <input type="number" name="ano" value="{{g.ano}}" min="1970" max="2099" class="form-control" required>
    <input type="text" name="categoria" value="{{g.categoria}}" class="form-control" required>
    <input type="text" name="plataforma" value="{{g.plataforma}}" class="form-control" required>
    <input type="number" name="preco" value="{{g.preco}}" min=0 step="0.01" class="form-control" required>
    <input type="number" name="quantidade" value="{{g.quantidade}}" class="form-control" required><br>
    <a href="{{url_for('edit', id=g.id)}}" onclick="return confirm('Confirma as alterações?')"><input
type="submit" value="Alterar" class="btn btn-warning"></a>
    <a href="{{url_for('estoque')}}"><input type="button" value="Cancelar" class="btn btn-danger"></a>
  </form>
</div>
{% endblock content %}
```

Alterando dados no banco



Após isso, no arquivo “**estoque.html**”, incluiremos o link com a **ID** do game na coluna da ação **Editar**.

```
<tbody>
{% for g in gamesestoque %}
  <tr>
    <td>{{g.titulo}}</td>
    <td>{{g.ano}}</td>
    <td>{{g.categoria}}</td>
    <td>{{g.plataforma}}</td>
    <td>{{g.preco}}</td>
    <td>{{g.quantidade}}</td>
    <td><a style="text-decoration: none; color: orange;" href="{{url_for('edit',
id=g.id)}}">Editar</a></td>
    <td><a style="text-decoration: none; color: red;" href="{{url_for('estoque',
id=g.id)}}" onclick="return confirm('Deseja mesmo excluir o jogo
{{g.titulo}}?')">Excluir</a></td>
  </tr>
{% endfor %}
</tbody>
```

Alterando dados no banco



Assim, já estamos conseguindo pegar as informações do game e mostra-las no formulário de edição.

Título:	Ano:	Categoria:	Plataforma:	Preço:	Quantidade:	Ação:	
Minecraft	2009	Sandbox	PC (Windows)	100	10	Editar	Excluir

Editar jogo:

Alterando dados no banco



Para finalizar, basta pegarmos as novas informações preenchidas no formulário e realizarmos a alteração do registro no banco. Para isso, incluiremos as seguintes linhas no arquivo “**route.py**”:

```
# CRUD - EDIÇÃO
@app.route('/edit/<int:id>', methods=['GET', 'POST'])
def edit(id):
    g = Game.query.get(id)
    # Edita o jogo com as informações do formulário
    if request.method == 'POST':
        g.titulo = request.form['titulo']
        g.ano = request.form['ano']
        g.categoria = request.form['categoria']
        g.plataforma = request.form['plataforma']
        g.preco = request.form['preco']
        g.quantidade = request.form['quantidade']
        db.session.commit()
        return redirect(url_for('estoque'))
    return render_template('editgame.html', g=g)
```

Feito isso, ao clicar no botão “**Alterar**” do formulário as **alterações serão efetivadas no banco de dados**.

Flask

Tópico 04.5: Paginação de registros *CRUD com o banco de dados MySQL*

Ao fazermos um SELECT no banco, podemos controlar o número de registros que serão exibidos por página. Faremos isso agora. A primeira alteração deve ser feita no arquivo “**routes.py**”, conforme a seguir:

```
# Cadastra um novo jogo
if request.method == 'POST':
    newgame = Game(request.form['titulo'], request.form['ano'],
request.form['categoria'], request.form['plataforma'], request.form['preco'],
request.form['quantidade'])
    db.session.add(newgame)
    db.session.commit()
    return redirect(url_for('estoque'))
else:
    # Captura o valor de 'page' que foi passado pelo método GET
    # Define como padrão o valor 1 e o tipo inteiro
    page = request.args.get('page', 1, type=int)
    # Valor padrão de registros por página (definimos 3)
    per_page = 3
    # Faz um SELECT no banco a partir da pagina informada (page)
    # Filtrando os registro de 3 em 3 (per_page)
    games_page = Game.query.paginate(page=page, per_page=per_page)
    return render_template('estoque.html', gamesestoque=games_page)
```

Paginação de registros



Feito isso, mais duas alterações serão realizadas no arquivo “**estoque.html**”, conforme a seguir:

```
        <th style="text-align: center;" colspan="2">Ação:</th>
    </tr>
</thead>
<tbody>
<!-- for g in gamesestoque -->
{% for g in gamesestoque.items %}
    <tr>
        <td>{{g.titulo}}</td>
```

```
    </tr>
{% endfor %}
</tbody>
</table>
<p style="text-align: center;">
    <!-- Se houver registros anteriores -->
    {% if gamesestoque.has_prev %}
        <a href="{{url_for('estoque', page=gamesestoque.prev_num)}}"><< Anterior</a>
    {% endif %}
    <!-- Se houver próximos registros -->
    {% if gamesestoque.has_next %}
        <a href="{{url_for('estoque', page=gamesestoque.next_num)}}">Próximo >></a>
</p>
{% endif %}
```

Paginação de registros



Agora os registros são buscados e exibidos na página, separados por uma paginação, de 3 em 3 registros:

Jogos em estoque:

Título:	Ano:	Categoria:	Plataforma:	Preço:	Quantidade:	Ação:	
FIFA 2019	2019	Esporte	XBOX 360	39.9	15	Editar	Excluir
CS GO	2012	FPS Online	PC	65.0	10	Editar	Excluir
Minecraft	2009	Sandbox	PC	35.9	20	Editar	Excluir

[Próximo >>](#)

Jogos em estoque:

Título:	Ano:	Categoria:	Plataforma:	Preço:	Quantidade:	Ação:	
Ragnarok	2007	RPG	PC	90.5	100	Editar	Excluir
Fifa Mobile 2	2024	Esporte	PC	100.5	32	Editar	Excluir
Overwatch 2	2022	Shooter	PC (Windows)	90.0	10	Editar	Excluir

[<< Anterior](#) [Próximo >>](#)



Flask

Programação Web III

Prof. Diego Max