

Mão na massa: Upload e preview de imagem

Chegou a hora de você executar o que foi visto na aula! Para isso, execute os passos listados abaixo.

Se você quiser o projeto completo feito no treinamento anterior, pode baixá-lo [aqui \(https://caelum-online-public.s3.amazonaws.com/739-python-flask2/02/aula2.zip\)](https://caelum-online-public.s3.amazonaws.com/739-python-flask2/02/aula2.zip).

1) Para fazer o upload do arquivo, adicione um novo *input*, como primeiro elemento do formulário, na página `novo.html`. Ele será do tipo `file` e de início só mostrará para o usuário os seus arquivos com a extensão `.jpg`:

```
<input type="file" name="arquivo" accept=".jpg">
```

2) E para poder tratar arquivos, adicione a propriedade `enctype` ao formulário, com o valor `multipart/form-data`:

```
<form action="{{ url_for('criar') }}" method="post" enctype="multipart/form-data">
```

3) Em `jogoteca.py`, na rota `criar`, capture o arquivo através do atributo `files` de `request`, e salve-o em uma variável:

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

    arquivo = request.files['arquivo']

    return redirect(url_for('index'))
```

4) Dentro do projeto, crie a pasta `uploads`, que será o local onde os arquivos serão salvos. E na configuração do seu projeto, adicione um caminho específico, fixo, para todos os arquivos de upload. Ele será o diretório do caminho absoluto do arquivo em que será executado, no caso o arquivo `jogoteca.py`, seguido da pasta `uploads`:

```
app.config['UPLOAD_PATH'] = os.path.dirname(os.path.abspath(__file__)) + '/uploads'
```

5) E para tal configuração funcionar, importe a biblioteca `os`:

```
import os
```

6) Voltando à rota `criar`, salve o diretório de upload em uma variável e salve o arquivo através do seu método `save`, passando para ele o seu destino:

```

@app.route('/criar', methods=['POST'])
def criar():
    nome = request.form['nome']
    categoria = request.form['categoria']
    console = request.form['console']
    jogo = Jogo(nome, categoria, console)
    jogo_dao.salvar(jogo)

    arquivo = request.files['arquivo']
    upload_path = app.config['UPLOAD_PATH']
    arquivo.save(f'{upload_path}/{arquivo.filename}')

    return redirect(url_for('index'))

```

7) Manter o nome do arquivo na hora de salv  -lo pode causar problemas, como por exemplo dois arquivos com nome **imagem.jpg**. Ent  o, customize o nome do arquivo com o seu **id**. Para isso funcionar, guarde o jogo retornado pelo m  todo salvar em uma vari  vel, possibilitando o acesso ao id do jogo salvo:

```

@app.route('/criar', methods=['POST'])
def criar():
    nome = request.form['nome']
    categoria = request.form['categoria']
    console = request.form['console']
    jogo = Jogo(nome, categoria, console)
    jogo = jogo_dao.salvar(jogo)

    arquivo = request.files['arquivo']
    upload_path = app.config['UPLOAD_PATH']
    arquivo.save(f'{upload_path}/capa{jogo.id}.jpg')

    return redirect(url_for('index'))

```

8) Para exibir uma *preview* da foto do jogo, dentro de **novo.html**, envolva o *input* em um **label**, dentro de um **figcaption** e de uma **figure**:

```

<figure>
    <figcaption>
        <label>
            <input type="file" name="arquivo" accept=".jpg">
        </label>
    </figcaption>
</figure>

```

9) Logo abaixo de **figure**, adicione uma tag **img** com a imagem a ser exibida. Como voc   est   mexendo na p  gina **novo.html**, n  o h   uma imagem para ser exibida, ent  o mostre uma imagem padr  o, que voc   pode baixar [aqui](https://caelum-online-public.s3.amazonaws.com/739-python-flask2/03/capa_padrao.zip) (https://caelum-online-public.s3.amazonaws.com/739-python-flask2/03/capa_padrao.zip) e colocar dentro de **uploads**:

```

<figure>
    
    <figcaption>
        <label>

```

```
<input type="file" name="arquivo" accept=".jpg">
</label>
</figcaption>
</figure>
```

10) Falta criar a rota `imagem`, então crie-a em `jogoteca.py`. Para ela funcionar corretamente, de `flask`, importe também `send_from_directory`:

```
@app.route('/uploads/<nome_arquivo>')
def imagem(nome_arquivo):
    return send_from_directory('uploads', nome_arquivo)
```

11) Por fim, falta exibir a imagem da capa do jogo no momento em que ele for editado. Para isso, em `editar.html`, como primeiro elemento do formulário, adicione o mesmo código de `novo.html`, mudando somente do nome do arquivo da capa:

```
<figure>
    
    <figcaption>
        <label>
            <input type="file" name="arquivo" accept=".jpg">
        </label>
    </figcaption>
</figure>
```

12) Agora, na rota `editar` em `jogoteca.py` remonte a estrutura do nome da capa do jogo e passe-a para o *template*:

```
@app.route('/editar/<int:id>')
def editar(id):
    if 'usuario_logado' not in session or session['usuario_logado'] == None:
        return redirect(url_for('login', proxima=url_for('editar', id=id)))
    jogo = jogo_dao.busca_por_id(id)
    return render_template('editar.html', titulo='Editando Jogo', jogo=jogo
                           , capa_jogo=f'capa{id}.jpg')
```