



Flask

Programação Web III

Prof. Diego Max

Flask

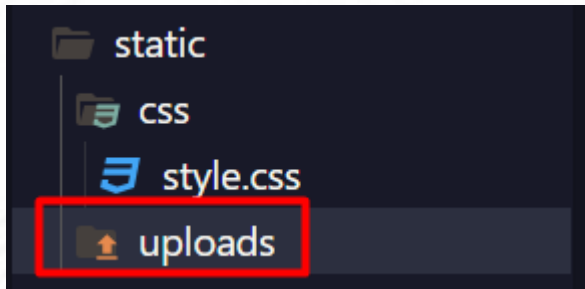
Aula 09.1: Upload de arquivos

Enviando arquivos para a aplicação

Upload de arquivos



Para criarmos um sistema de upload de arquivos, o primeiro passo é definir a pasta que armazenará os arquivos enviados para a aplicação. Nesse caso, criamos uma pasta chamada “**uploads**”, dentro da pasta “static”.



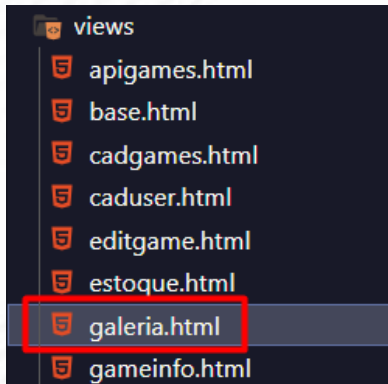
Em seguida, no arquivo “**base.html**”, inserimos um link na navbar que será responsável pelo redirecionamento para a rota de galeria, onde serão feitos os uploads dos arquivos.

```
<li class="nav-item">
  <a class="nav-link"
href="{{url_for(galeria)}}">Galeria</a>
</li>
</ul>
```

Upload de arquivos



Feito isso, iremos criar agora uma nova view chamada “**galeria.html**”, que será a página a ser exibida.



Agora no arquivo “**routes.py**”, iremos criar a rota para essa página:

```
# UPLOAD DE IMAGENS
@app.route('/galeria', methods=['GET', 'POST'])
def galeria():
    return render_template('galeria.html')
```

No arquivo “**app.py**”, precisamos configurar a pasta que receberá os arquivos e o tamanho máximo permitido dos mesmos. No nosso caso, o tamanho máximo permitido dos arquivos será de **16 megabytes**.

```
# Define pasta que receberá arquivos de upload
app.config['UPLOAD_FOLDER'] = 'static/uploads'
# Define o tamanho máximo de um arquivo de upload
app.config['MAX_CONTENT_LENGTH'] = 16 * 1024 * 1024
```

Upload de arquivos



Voltando ao arquivo “**galeria.html**”, iremos inserir o código responsável por criar um formulário para envio dos arquivos:

```
{% extends 'base.html' %}
{% block content %}
<h2 style="text-align: center;">Galeria</h2><br>
<div class="container col-6">
  <div class="card mt-4">
    <div class="card-body">
      <form action="{#{url_for('galeria')}}#" method="post" enctype="multipart/form-data">
        <label for="file">Envie sua imagem:</label>
        <input class="form-control" name="file" type="file" required><br>
        <button type="submit" class="btn btn-danger btn-block">Enviar</button>
      </form>
    </div>
  </div>
</div>
{% endblock content %}
```

Galeria

Envie sua imagem:

Escolher arquivo Nenhum arquivo escolhido

Enviar

Todos os direitos reservados ®

Upload de arquivos



Após isso, no arquivo **“routes.py”**, precisamos importar a biblioteca **“OS”**, que iremos utilizar para referenciar os diretórios e a biblioteca **“UUID”** responsável por criar identificadores únicos para os nomes dos arquivos enviados ao servidor.

```
import os
import uuid
```

Ainda no arquivo **“routes.py”**, criaremos uma função para verificar os tipos de arquivos permitidos, receber os arquivos e gravá-los no servidor com um identificador único, conforme a seguir:

```
# Definindo tipos de arquivos permitidos
FILE_TYPES = set(['png', 'jpg', 'jpeg', 'gif'])
def arquivos_permitidos(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in FILE_TYPES

# UPLOAD DE IMAGENS
@app.route('/galeria', methods=['GET', 'POST'])
def galeria():
    if request.method == 'POST':
        # Captura o arquivo vindo do formulário
        file = request.files['file']
        # Verifica se a extensão do arquivo é permitida
        if not arquivos_permitidos(file.filename):
            flash("Utilize os tipos de arquivos referentes a imagem.", 'danger')
            return redirect(request.url)
        # Define um nome aleatório para o arquivo
        filename = str(uuid.uuid4())
        # Salva o arquivo na pasta de uploads
        file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
        flash("Imagem enviada com sucesso!", 'success')
        return redirect(url_for('galeria'))
    return render_template('galeria.html')
```

Upload de arquivos



Agora no arquivo “**galeria.html**”, devemos inserir o código responsável por fazer a exibição das **flash messages** naquela página:

```
{% extends 'base.html' %}
{% block content %}

<h2 style="text-align: center;">Galeria</h2>

{% with messages = get_flashed_messages(with_categories=true) %}
  {% for category, message in messages %}
    <div class="alert alert-{{ category }}">
      {{ message }}
    </div>
  {% endfor %}
{% endwith %}
```

Ao enviar um tipo de arquivo não permitido a seguinte mensagem é exibida:

Utilize os tipos de arquivos referentes a imagem.

Envie sua imagem:

Escolher arquivo

Nenhum arquivo escolhido

Enviar

Upload de arquivos



Enviando um arquivo correto, o mesmo é recebido e armazenado na pasta upload da aplicação, conforme a seguir:

Imagem enviada com sucesso!

Envie sua imagem:

Escolher arquivo

Nenhum arquivo escolhido

Enviar

 uploads

 62a01963-84f3-496a-8e49-ab69d9903850

Flask

Aula 09.2: Cadastrando arquivos no banco

Upload de arquivos

Cadastrando arquivos no banco

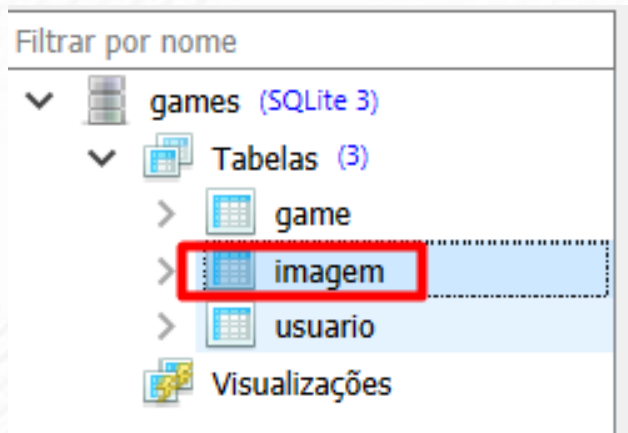


Agora que os arquivos já estão sendo recebidos, iremos gravar o nome desses arquivos no banco de dados, para que possamos montar uma **galeria de imagens**. O primeiro passo é criar uma nova entidade no banco de dados, para isso, devemos criar uma nova classe com o nome **"Imagem"** no arquivo **"database.py"**:

```
# Classe para imagens
class Imagem(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    filename = db.Column(db.String(120), unique=True, nullable=False)

    def __init__(self, filename):
        self.filename = filename
```

Ao rodar novamente a aplicação, uma nova tabela deve ser criada no banco:



Cadastrando arquivos no banco



Agora no arquivo “**routes.py**”, devemos importar a classe **Imagem**:

```
from models.database import db, Game, Usuario, Imagem
```

Ainda no arquivo “**routes.py**”, iremos inserir os códigos responsáveis por gravar o nome dos arquivos no banco e também buscar esses nomes para que a galeria de imagens seja montada, conforme a seguir:

```
# UPLOAD DE IMAGENS
@app.route('/galeria', methods=['GET', 'POST'])
def galeria():
    # Seleciona os nomes dos arquivos de imagens no banco
    imagens = Imagem.query.all()

    if request.method == 'POST':
        file = request.files['file']
        # Verifica se o tipo de arquivo é permitido
        if not arquivos_permitidos(file.filename):
            flash("Utilize os tipos de arquivos referentes a imagem.", 'danger')
            return redirect(request.url)

        # Criando um nome aleatório para o arquivo
        filename = str(uuid.uuid4())

        # Gravando o nome do arquivo no banco
        img = Imagem(filename)
        db.session.add(img)
        db.session.commit()

        # Gravando o arquivo na pasta de uploads
        file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
        flash("Imagem enviada com sucesso!", 'success')
        return redirect(url_for('galeria'))
    return render_template('galeria.html', imagens=imagens)
```

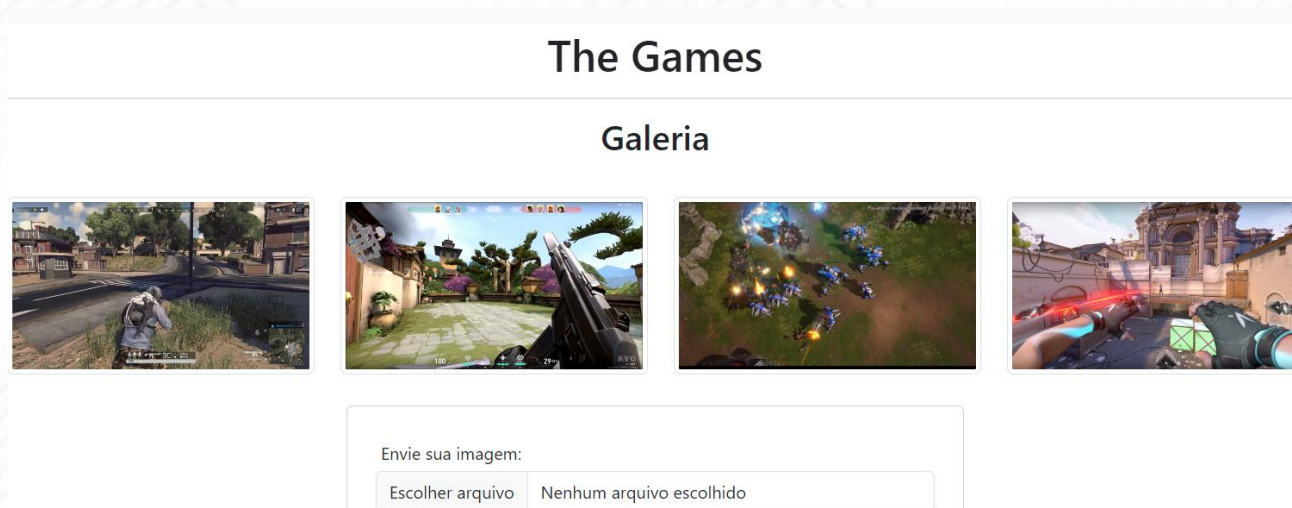
Cadastrando arquivos no banco



Por fim, no arquivo **“galeria.html”**, iremos montar uma estrutura para que as imagens sejam exibidas corretamente:

```
{% extends 'base.html' %}
{% block content %}
<h2 style="text-align: center;">Galeria</h2><br>
<div class="row">
  {% for img in imagens %}
    <div class="col-md-3 m-auto">
      
    </div>
  {% endfor %}
</div>
```

Agora ao enviar novas imagens, as mesmas já são exibidas na página, conforme a seguir:





Flask

Programação Web III

Prof. Diego Max