Utilizando o Flask

Instrutor: Tarik Ponciano

Senac Fecomércio Sesc

Links da Disciplina

1. Discord: https://discord.gg/wt5CVZZWJs

2. Drive: https://drive.google.com/drive/folders/1hOl0DaPeAor7gnhKBUIlZ5n8lLRDNvUY?usp=sharing

3. Github: https://github.com/TarikPonciano/Programador-de-Sistema-SENAC

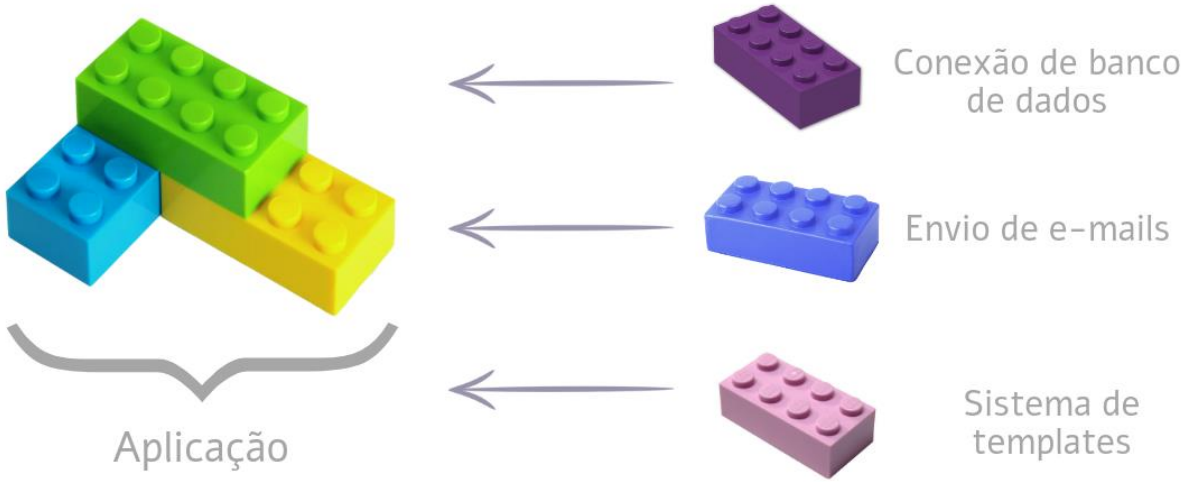
Flask – O que é?

Escrito em Python e disponível sobre a licença BSD (Licença de código aberto), o Flask é um micro-framework multiplataforma que provê um modelo simples para o desenvolvimento web.

Podemos dizer que o micro-framework é uma versão minimalista destes frameworks, sendo bastante utilizado para criação de microsserviços, como APIs RESTful.

Pense em um Micro-Framework como uma peça de lego. Inicialmente, um projeto criado com o micro-framework possui apenas o básico para funcionar, (normalmente, sistema de rotas), porém, ao decorrer do projeto, podem haver necessidades para utilização de outros recursos como, conexão de banco de dados, sistemas de templates, envio de email, etc. A partir desta necessidade, novas bibliotecas são “encaixadas” no projeto, como uma estrutura de lego.

Flask – O que é?


Conexão de banco de dados
Envio de e-mails
Sistema de templates
Aplicação

Flask – Características

•Simplicidade: Por possuir apenas o necessário para o desenvolvimento de uma aplicação, um projeto escrito com Flask é mais simples se comparado aos frameworks maiores, já que a quantidade de arquivos é muito menor e sua arquitetura é muito mais simples.

•Rapidez no desenvolvimento: Com o Flask, o desenvolvedor se preocupa em apenas desenvolver o necessário para um projeto, sem a necessidade de realizar configurações que muitas vezes não são utilizadas.

•Projetos menores: Por possuir uma arquitetura muito simples (um único arquivo inicial) os projetos escritos em Flask tendem a ser menores e mais leves se comparados a frameworks maiores.

•Aplicações robustas: Apesar de ser um micro-framework, o Flask permite a criação de aplicações robustas, já que é totalmente personalizável, permitindo, caso necessário, a criação de uma arquitetura mais definida.

Flask – Hello World

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def index():
return ‘Hello World’

if __name__ == "__main__":

app.run()
```

Senac
Fecomércio
Sesc

Flask – Utilizando Templates HTML

• No momento, seu aplicativo mostra somente uma mensagem simples sem HTML. Os aplicativos Web usam, em sua maioria, o HTML para exibir informações para os visitantes. Desta forma, você trabalhará na incorporação de arquivos HTML em seu aplicativo, que podem ser exibidos no navegador Web.

• O Flask permite o uso de Templates/Modelos HTML para a criação das visualizações desejadas de uma página WEB, além de permitir o uso de lógica e scripts nessas páginas HTML.

Flask – Utilizando Templates HTML

```
from flask import Flask, render_template

app = Flask(__name__)

@app.route('/')
def index():
    return render_template('index.html')
```

Experimente rodar o código acima

A linha @app.route(‘/’) representa a url requisitada em nossa página web e será a principal ferramenta para conectar a interação com nossa visualização e nosso servidor Flask

Senac
Fecomércio
Sesc

Flask – Utilizando Templates HTML

Para utilizar corretamente os templates, precisamos criar uma pasta chamada “templates” na mesma pasta em que está nosso código python.

Na nova pasta “templates” criamos o arquivo index.html, e finalmente invocamos esse arquivo no código do Flask.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>FlaskBlog</title>
</head>
<body>
   <h1>Welcome to FlaskBlog</h1>
</body>
</html>
```

index.html

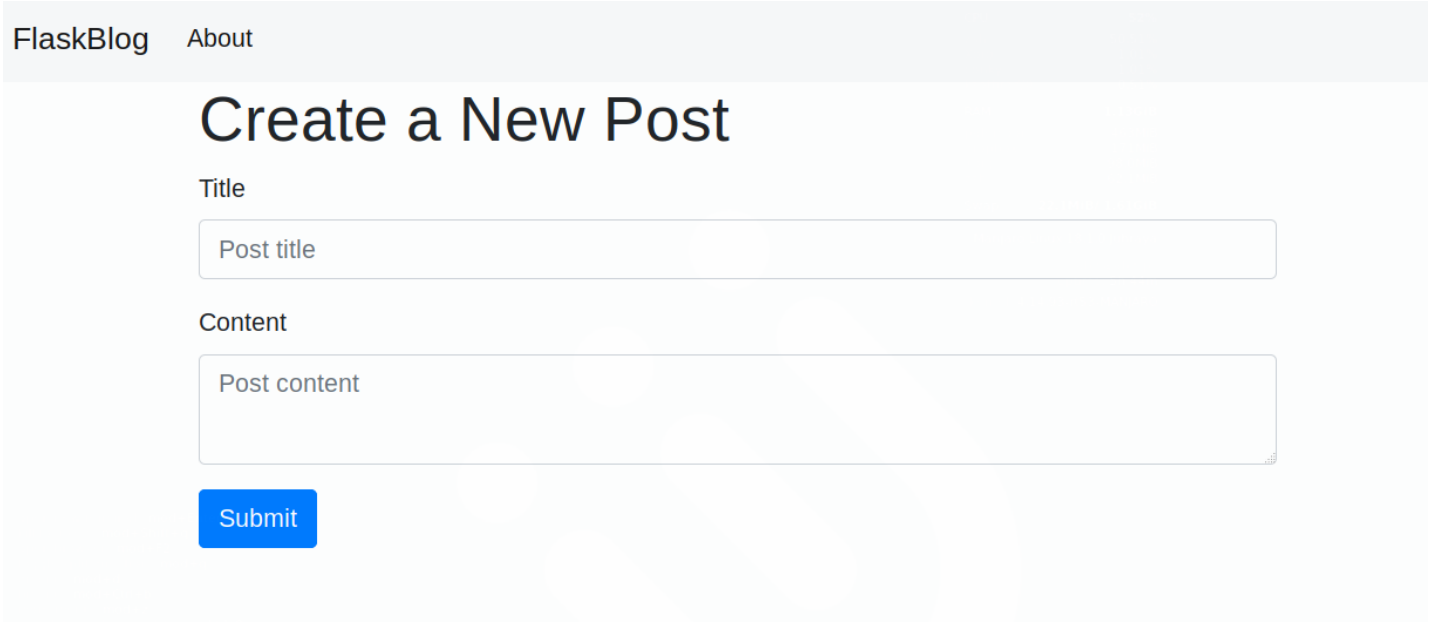
Flask – Formulário HTML

```
<form method="post">
    <div class="form-group">
        <label for="title">Title</label>
        <input type="text" name="title"
               placeholder="Post title" class="form-control"
               value="{{ request.form['title'] }}"></input>
    </div>

    <div class="form-group">
        <label for="content">Content</label>
        <textarea name="content" placeholder="Post content"
                  class="form-control">{{ request.form['content'] }}</textarea>
    </div>
    <div class="form-group">
        <button type="submit" class="btn btn-primary">Submit</button>
    </div>
</form>
```

Senac
Fecomércio
Sesc

Flask – Formulário HTML



Vamos continuar no VScode

Flask – Utilizando sem Templates

O Flask não precisa dos templates para agir como um servidor. É possível utilizá-lo como um servidor para requisições de uma API Rest

```
from flask import Flask, request

app = Flask(__name__)

@app.route('/', methods=['POST'])

def webhook():

data = request.get_json(force=True)

return f'Recebido: {data['data']}\n'

if __name__ == "__main__":

app.run()
```

Senac Fecomércio Sesc

Flask – Exemplo de edição de postagem

```python
@app.route('/<int:id>/edit', methods=('GET', 'POST'))
def edit(id):
    post = get_post(id)

    if request.method == 'POST':
        title = request.form['title']
        content = request.form['content']

        if not title:
            flash('Title is required!')
        else:
            conn = get_db_connection()
            conn.execute('UPDATE posts SET title = ?, content = ?'
                         ' WHERE id = ?',
                         (title, content, id))
            conn.commit()
            conn.close()
            return redirect(url_for('index'))

    return render_template('edit.html', post=post)
```

Senac
Fecomércio
Sesc

De volta ao VScode

Senac
Fecomércio
Sesc

Obrigado!!

Senac Fecomércio Sesc

https://www.treinaweb.com.br/blog/o-que-e-flask

https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3-pt

https://www.treinaweb.com.br/blog/consumindo-apis-com-python-parte-2

https://www.w3schools.com/python/ref_requests_post.asp

Senac Fecomércio Sesc