blue

Módulo 2
Flask

< thefutureisblue.me />

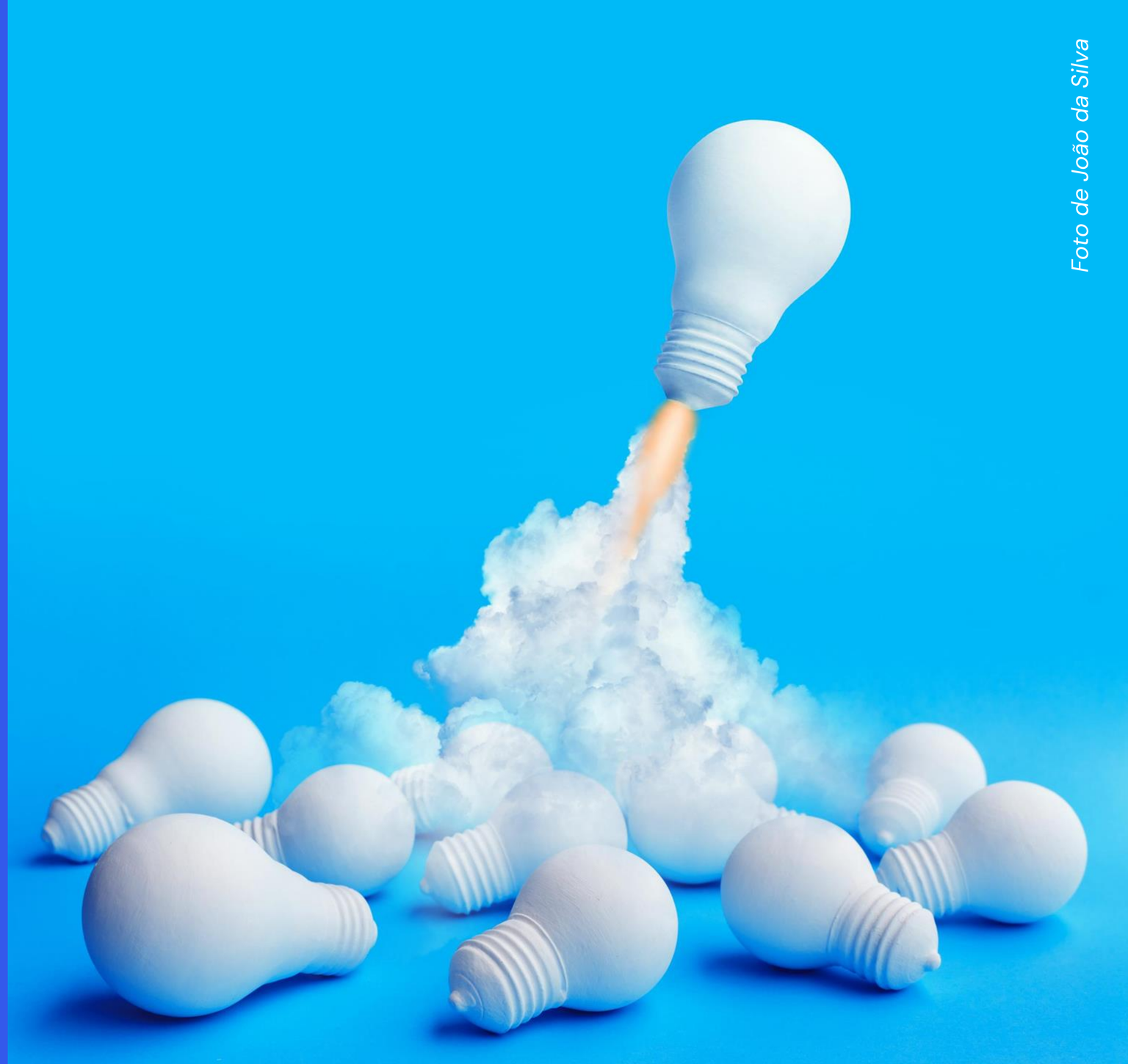

Foto de João da Silva

blue

O que é Flask?

Flask facilita o desenvolvimento de sites, APIs e qualquer sistema que rode na internet, através do protocolo HTTP.

Com apenas algumas linhas de código é possível rodar um arquivo Python para que interage com arquivos HTML, permitindo a integração entre esses dois ecossistemas.

Flask é um micro-framework. Esse termo refere-se a estruturas de aplicativos que são minimalistas.

O Flask oferece sugestões, mas não impõe quaisquer dependências ou layout do projeto.

blue

Framework x micro-framework?

Framework x micro-framework?

Framework Django e o Micro-framework Flask.

O Django é um framework web python de alto nível que permite rápido desenvolvimento de sites seguros e fácil manutenção. Ele é composto por uma grande estrutura de pastas e muitas funcionalidades, sendo utilizado geralmente para projetos maiores, em que são necessários muitos recursos. É gratuito e de código aberto; tem uma comunidade ativa; boa documentação; e opções de suporte gratuito e pago.

Flask que é um micro-framework que utiliza python para criar aplicativos Web de forma simples, rápida. Não possui estrutura definida de pastas; Pode ser importado de qualquer outro módulo do Python. Outras bibliotecas podem ser importadas para o flask, importando apenas recursos específicos que você queira utilizar.

Instalando o Flask

1) Abra o terminal

2) Instale o Flask:
py -m pip install flask

3) Crie um arquivo chamado **app.py**:

4) Adicione o conteúdo 'boilerplate' e execute o Flask:

flask run

blue

Botando para rodar!

the future is blue

Criando uma aplicação

- Conteúdo do arquivo **app.py**

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def rota1():
   return 'Hello, world!'

@app.route('/rota2')
def rota2():
       return  '<h1>Essa é a segunda rota de aplicação</h1>'

if __name__ == '__main__':
   app.run(debug=True)
```

FLASK – Aplicação | Rota

O parâmetro string passado para o decorator determina a extensão da URL que irá vincular a função.

Nossa página inicial ou domínio é representado localmente como **http://127.0.0.1:5000/** ou **http://localhost:5000/**

Usamos decorators para adicionar ao

@app.route ("/rota2")

http://127.0.0.1:5000/rota2

Assim que a página estiver na nuvem, 127.0.0.1 será substituído pelo domínio (www.site.com).

Renderizando arquivos HTML

```
from flask import (
    Blueprint, render_template, request
)

from flask import Flask
app = Flask(__name__)

bp = Blueprint('app', __name__)

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

app.register_blueprint(bp)

# Habilitando o hot reload (execução via python app.py em vez de flask run):
if __name__ == "__main__":
    app.run(debug=True)
```

Variáveis em arquivos HTML: app.py

```
@bp.route('/')
def index():
    nome_jogador = 'João Silva'
    premio = True

    return render_template(
        'index.html',
        nome_jogador=nome_jogador,
        premio=premio,
    )
```

Variáveis em arquivos HTML: index.html

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Variáveis em HTML</title>
    </head>
    <body>
        <div class="conteudo">
            {% if premio %}

            <div class="bloco">
                {{ nome_jogador }} recebeu o prêmio.
            </div>

            {% else %}

            <div class="bloco">
                {{ nome_jogador }} não ganhou o prêmio.
            </div>

            {% endif %}
        </div>
    </body>
</html>
```

blue

Botando para rodar!

the future is blue

Por hoje é só!
Obrigado! =)

the future is blue

blue