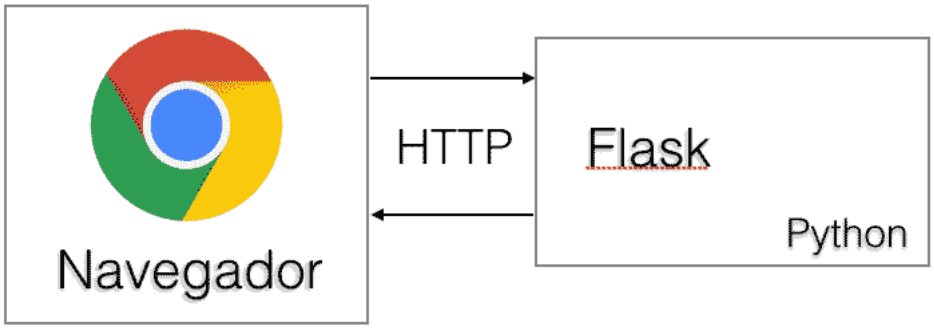
10

Entendendo e testando o Flask

Já vimos no vídeo que o Flask é um servidor web. Isso significa nada mais do que o Flask entende o protocolo HTTP:



Quando você roda o arquivo **pibot.py** na linha de comando, automaticamente sobe o servidor Flask. Por padrão, o servidor sobre na porta **5000**. Agora basta saber o endereço IP para acessar o Flask.

Lembrando que você consegue facilmente descobrir o IP do Raspberry na linha de comando, como mostramos no curso anterior:

```
hostname -I
```

```
pi@raspberrypi:~ $ hostname -I
192.168.0.170
pi@raspberrypi:~ $
```

Sabendo disso, e com Flask rodando, é possível chamar o Raspberry Pi pelo navegador (o navegador é um cliente HTTP).

```
http://192.168.0.170/
```

Quando você usa o navegador para chamar o Raspberry Pi, o navegador *envia uma requisição HTTP*. Uma requisição é nada mais do que uma forma padronizada de enviar e pedir informação do servidor. Os detalhes desse padrão são descritos no protocolo HTTP.

Mapeando uma requisição

Quando o servidor recebe essa requisição, a tarefa do Flask é achar uma função que deve ser executada. Então devemos dar algumas dicas para o Flask, para ele poder procurar e achar a função certa.

Para deixar mais claro, vamos dar um exemplo. Observe o seguinte código:

```
from flask import Flask

app = Flask(__name__)
```

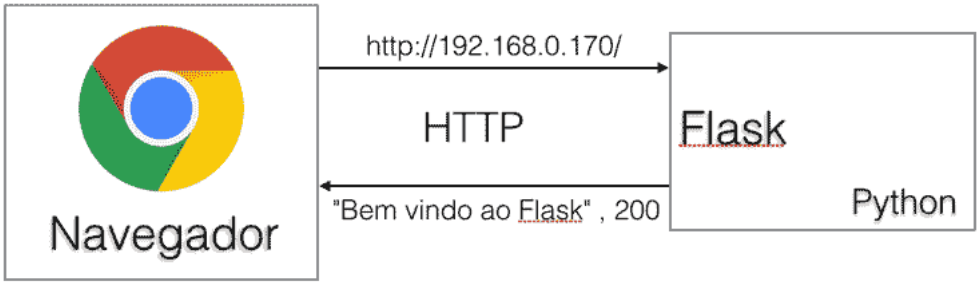
```
@app.route('/')
def bem_vindo():
    return 'Bem vindo ao Flask', 200

if __name__ == "__main__":
    app.run(host='0.0.0.0')
```

Além do `import` e a inicialização do Flask, existe apenas uma função nesse código:

```
@app.route('/')
def bem_vindo():
    return 'Bem vindo ao Flask', 200
```

Repare que a função possui uma configuração através do `@app.route` . Aqui estamos configurando a URL que deve ser utilizada no navegador para chamar essa função. Ou seja, ao chamar http://192.168.0.170:5000/ (http://192.168.0.170:5000/), o Flask recebe a requisição e vai chamar a função `bem_vindo` .

Aquela configuração também se chama *mapeamento*. Repare também que a função devolve uma string. É essa string que o navegador receberá como resposta. Aquele código 200 é o status do protocolo HTTP. 200 significa *sucesso*:



Mais mapeamento

Através do `@app.route` , podemos então configurar qual requisição queremos atender. Por exemplo, se quisermos executar algum código para a URL `/tchau` , podemos adicionar uma nova função:

```
@app.route('/tchau')
def tchau():
    return 'Até mais', 200
```

Ao testar no navegador http://192.168.0.170:5000/tchau (http://192.168.0.170:5000/tchau), deve ser executado a função `tchau` .

Testando o código

Se quiser testar esse pequeno código, crie um novo arquivo **teste-flask.py** no seu Raspberry Pi com o seguinte conteúdo:

```
## arquivo teste-flask.py

from flask import Flask

app = Flask(__name__)
```

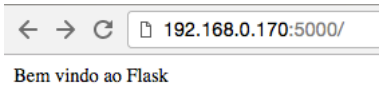
```
@app.route('/')
def bem_vindo():
    return 'Bem vindo ao Flask', 200

@app.route('/tchau')
def tchau():
    return 'Até mais', 200

if __name__ == "__main__":
    app.run(host='0.0.0.0')
```

Com Flask instalado, rode na linha de comando:

```
python3 teste-flask.py
```

Depois teste no navegador:

```
http://SEU-IP:5000/
```



e

```
http://SEU-IP:5000/tchau
```