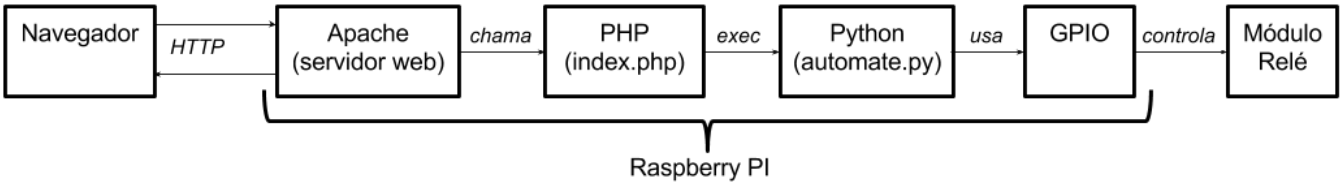
11

Para saber mais: Tudo com Python

Porque duas linguagens de programação?

Vamos olhar novamente nosso diagrama mais arquitetural:



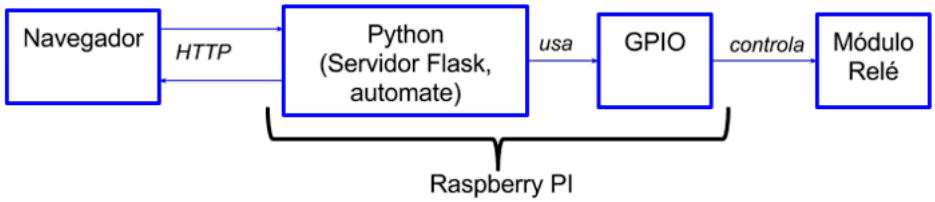
Repare que estamos usando duas linguagens diferentes: PHP e Python

Será que é possível fazer tudo com uma linguagem só? Já vimos que é possível fazer tudo com PHP, mas é claro que que também é possível fazer tudo com também com Python!!

Programação Web com Python

Podemos eliminar o PHP e usar o Python, não só para controlar os GPIOs, como também programar na web. Como?

O Python é uma linguagem universal e muito utilizada na programação web. Aliás, existem várias bibliotecas que facilitam o desenvolvimento Web. Uma famosa é o Django, mas talvez um pouco demais para resolver o problema do formulário. Por isso usaremos o Flask, que faz exatamente o papel do Apache e PHP. Algo leve e relativamente simples de usar.



Instalação do Flask

O primeiro passo é instalar o Flask. Abra um terminal e execute:

```
sudo apt-get install python3-flask
```

Criação das rotas

Uma vez o Flask instalado, podemos criar o código Python que representa o servidor e criar uma rota. Uma rota é algo que podemos chamar através do navegador. Em outras palavras, o navegador envia uma requisição HTTP e o servidor Flask responde. Vamos lá?

1) Crie uma arquivo app.py. Pode usar o editor Python:

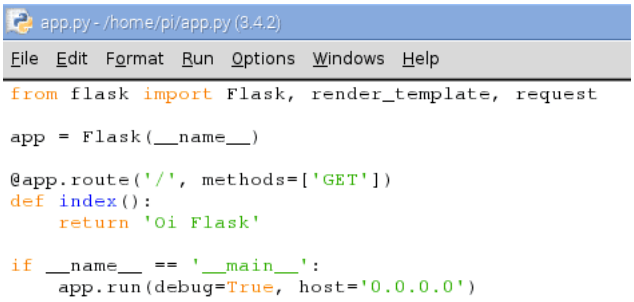
```
from flask import Flask, render_template, request
```

```
app = Flask(__name__)

@app.route('/', methods=['GET'])
def index():
    return 'Oi Flask'

if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0')
```

Como sempre tenha cuidado com as indentações.



O código importa e inicializa o Flask e define uma rota. A rota chama `/` e atende a requisição HTTP GET.

2) Salve o arquivo e abra um terminal. No terminal digite:

```
python3 app.py
```

Repare que estamos usando o Python 3, que já vem instalado no Raspberry PI. Esse comando sobe o servidor no porta 5000, que é o padrão do Flask.

3) Teste no navegador: http://192.168.0.170 (http://192.168.0.170)

No navegador você deve ver a mensagem "Oi Flask".

Trabalhando com HTML

Já vimos uma mensagem no navegador, mas na verdade queremos ver o formulário para controlar o Relé, mas primeiro pare o servidor Flask apertando `CTRL + C`.

Usando Flask, as páginas HTML devem ficar dentro de uma pasta **templates**, que é preciso criar antes. Execute no terminal:

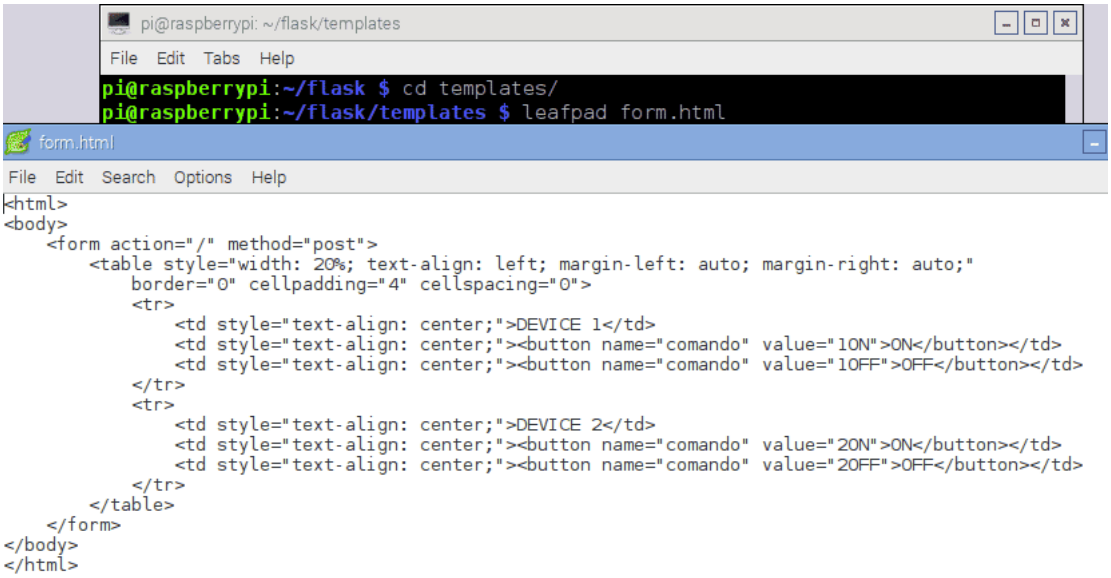
```
mkdir templates
```

```
pi@raspberrypi:~/flask $ mkdir templates
pi@raspberrypi:~/flask $ ls
app.py  templates
pi@raspberrypi:~/flask $
```

Dentro da pasta **templates**, crie um novo arquivo **form.html** com o conteúdo seguinte:

```
<html>
<body>
    <form action="/" method="post">
        <table style="width: 20%; text-align: left; margin-left: auto; margin-right: auto;"
            border="0" cellpadding="4" cellspacing="0">
            <tr>
                <td style="text-align: center;">DEVICE 1</td>
                <td style="text-align: center;"><button name="comando" value="1ON">ON</button></td>
                <td style="text-align: center;"><button name="comando" value="1OFF">OFF</button></tc
            </tr>
            <tr>
                <td style="text-align: center;">DEVICE 2</td>
                <td style="text-align: center;"><button name="comando" value="2ON">ON</button></td>
                <td style="text-align: center;"><button name="comando" value="2OFF">OFF</button></tc
            </tr>
        </table>
    </form>
</body>
</html>
```

O HTML é praticamente o mesmo, cada botão possui o mesmo atributo `name`, mudando somente o `value`. Isso vai facilitar a leitura depois.



Salve o arquivo e abra novamente o **app.py**. Agora queremos chamar o formulário pela rota. Para tal, altere a função `index`:

```
@app.route('/', methods=['GET'])
def index():
    return render_template('form.html')
```

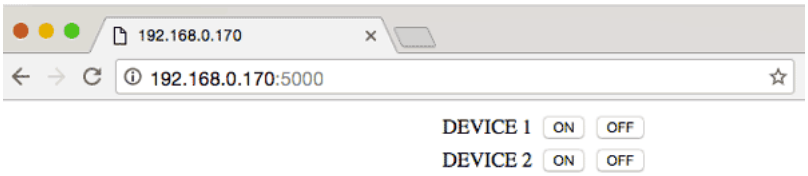
Repare que estamos retornando o HTML através da função `render_template` .

Agora reinicie o Flask no terminal e teste no navegador com seu IP:

```
python3 app.py
```

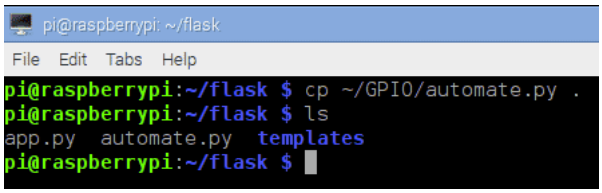
http://192.168.0.170:5000 (http://192.168.0.170:5000)

Deve aparecer o formulário:



Ajustando o script automate.py

Queremos chamar as funções do nosso script **automate.py** pelo Python, mas antes disso é preciso "fazer uma limpeza". Crie uma cópia do arquivo **automate.py** e coloque na mesma pasta do arquivo **app.py**:



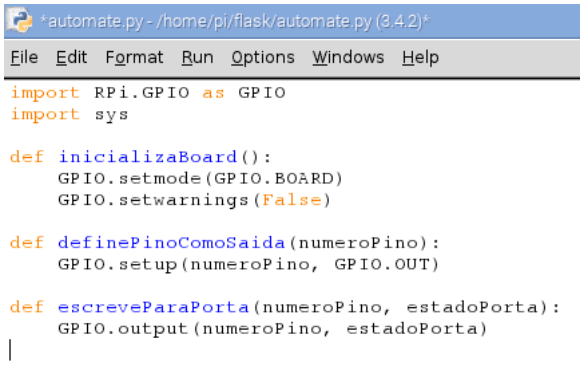
No arquivo **automate.py**, deixe apenas as importações e funções:

```
import RPi.GPIO as GPIO
import sys

def inicializaBoard():
    GPIO.setmode(GPIO.BOARD)
    GPIO.setwarnings(False)

def definePinoComoSaida(numeroPino):
    GPIO.setup(numeroPino, GPIO.OUT)

def escreveParaPorta(numeroPino, estadoPorta):
    GPIO.output(numeroPino, estadoPorta)
```



Automate no app.py

Agora vamos usar o **automate.py** dentro do arquivo **app.py**.

Abra o arquivo **app.py** e adicione mais um *import* no início:

```
from automate import inicializaBoard, definePinoComoSaida, escreveParaPorta
```

Repare que estamos importando as 3 funções do arquivo **automate.py**.

Logo depois desse *import*, chame a função `inicializaBoard` e defina os pinos 7 e 11 como saída:

```
inicializaBoard()
definePinoComoSaida(7)
definePinoComoSaida(11)
```

O último passo é adicionar mais uma rota no **app.py**, que atende o formulário e aciona o Relé. Logo após da função `index`, adicione:

```
@app.route('/', methods=['POST'])
def submit():
    comando=request.form['comando']
    print(comando)

    if(comando == '1ON'):
       escreveParaPorta(7,0)
    if(comando == '1OFF'):
       escreveParaPorta(7,1)
    if(comando == '2ON'):
       escreveParaPorta(11,0)
    if(comando == '2OFF'):
       escreveParaPorta(11,1)
    return render_template('form.html')
```

Repare que estamos lendo o parâmetro `comando` do formulário, para testar nos `if` s o valor, escrevendo para uma porta.

Segue uma vez o código completo do **app.py** para sua comparação:

```
from flask import Flask, render_template, request
from automate import inicializaBoard, definePinoComoSaida, escreveParaPorta

app = Flask(__name__)

inicializaBoard()
definePinoComoSaida(7)
definePinoComoSaida(11)

@app.route('/', methods=['GET'])
def index():
    return render_template('form.html')

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

```
def submit():
    comando=request.form['comando']
    print(comando)

    if(comando == '1ON'):
       escreveParaPorta(7,0)
    if(comando == '1OFF'):
       escreveParaPorta(7,1)
    if(comando == '2ON'):
       escreveParaPorta(11,0)
    if(comando == '2OFF'):
       escreveParaPorta(11,1)
    return render_template('form.html')

if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0')
```

Salve tudo e reinicialize o servidor:

```
python3 app.py
```

Teste o formulário clicando nos botões:

http://192.168.0.170 (http://192.168.0.170)