UNIVERSIDADE FEDERAL DE UBERLÂNDIA-UFU
FACULDADE DE ENGENHARIA MECÂNICA
ENGENHARIA MECATRÔNICA
SISTEMAS DIGITAIS

Projeto Final 2

Projeto IOT com RaspberryPi, Flask e ngrok

ENRICO SAMPAIO BONELA	11721EMT007
GUILHERME SALOMÃO AGOSTINI	11721EMT003
JADSON SILVA SOUZA	11721EMT017
LEONARDO FRANÇA DE CARVALHO	11821EMT012
RAFAEL LINS NOBRE	11811EMT002
VITOR HUGO VASCONCELOS DE MELO	11821EMT006

UBERLÂNDIA

2022

Sumário

Sumário **2**

Resumo **3**

1. Introdução **4**

2. Materiais utilizados: **5**

3. Aplicação Web **6**

Apêndice **9**

Codigo app.py: 9

Código master.css 11

Código index.html: 12

Resumo

Esse projeto tem como objetivo a construção de um projeto IOT (internet of things) para simularmos uma automatização em uma residência. Para isso, foi construído um protótipo composto por 3 Leds, que representam 3 luzes da casa e um sensor de presença. Para controlar tudo e funcionar como servidor, foi utilizado um Raspberry Pi 3 Model B+. Nossa aplicação foi feita na linguagem Python, com o framework Flask, para conectar o sistema com a internet, foi utilizado o serviço gratuito ngrok.

1. Introdução

O termo IOT, ou Internet das Coisas, refere-se à rede coletiva de dispositivos conectados à internet e à tecnologia que facilita a comunicação entre os dispositivos e entre estes e a nuvem. Isso significa que dispositivos do dia a dia, como escovas de dentes, aspiradores, carros e máquinas, podem usar sensores para coletar dados e responder de forma inteligente aos usuários.

A Internet das Coisas integra “coisas” cotidianas à Internet. Engenheiros de computação vêm adicionando sensores e processadores a objetos do cotidiano desde a década de 1990, no entanto, o progresso foi inicialmente lento porque os chips eram grandes e volumosos. Chips de computador de baixa potência chamados etiquetas RFID foram usados pela primeira vez para rastrear equipamentos caros mas, à medida que os dispositivos de computação diminuíram de tamanho, esses chips também se tornaram menores, mais rápidos e mais inteligentes ao longo do tempo.

Um típico sistema de IoT funciona por meio da coleta e troca de dados em tempo real. Um sistema IoT tem três componentes: dispositivos inteligentes, aplicação e interface.

- O dispositivo pode ser qualquer equipamento que recebeu recursos de computação. Ele coleta dados de seu ambiente, entradas do usuário ou padrões de uso e comunica dados pela Internet de e para sua aplicação de IoT.
- Uma aplicação de IoT é um conjunto de serviços e software que integra dados recebidos de vários dispositivos de IoT. Ela utiliza tecnologia de machine learning ou inteligência artificial (IA) para analisar esses dados e tomar decisões informadas. Essas decisões são comunicadas de volta ao dispositivo de IoT e esse dispositivo responde de forma inteligente às entradas.
- A interface gráfica para o usuário é o que gerencia o dispositivo de IoT ou a frota de dispositivos. Exemplos comuns incluem uma aplicação móvel ou site que pode ser usado para registrar e controlar dispositivos inteligentes.

2. Materiais utilizados:

Para a construção do protótipo, foram utilizados os seguintes componentes:

- Raspberry Pi 3
- 3 Leds
- 3 resistores de 100 Ω
- 1 sensor de presença PIR
- Jumpers

O protótipo final ficou da seguinte maneira:

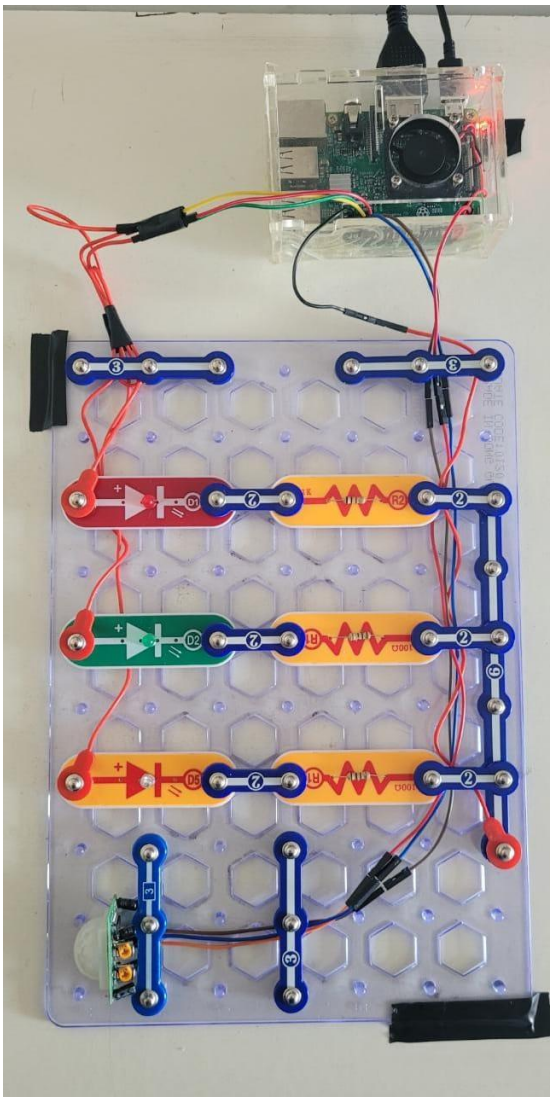

Figura 1: Projeto montado na Raspberry

3. Aplicação Web

No raspberry, temos como SO uma distribuição Linux, a Raspibian. Foi montado uma aplicação em python utilizando o framework aprendido durante as atividades, o Flask, para a construção da página Web, utilizamos também html e css.

A página Web ficou da seguinte forma:

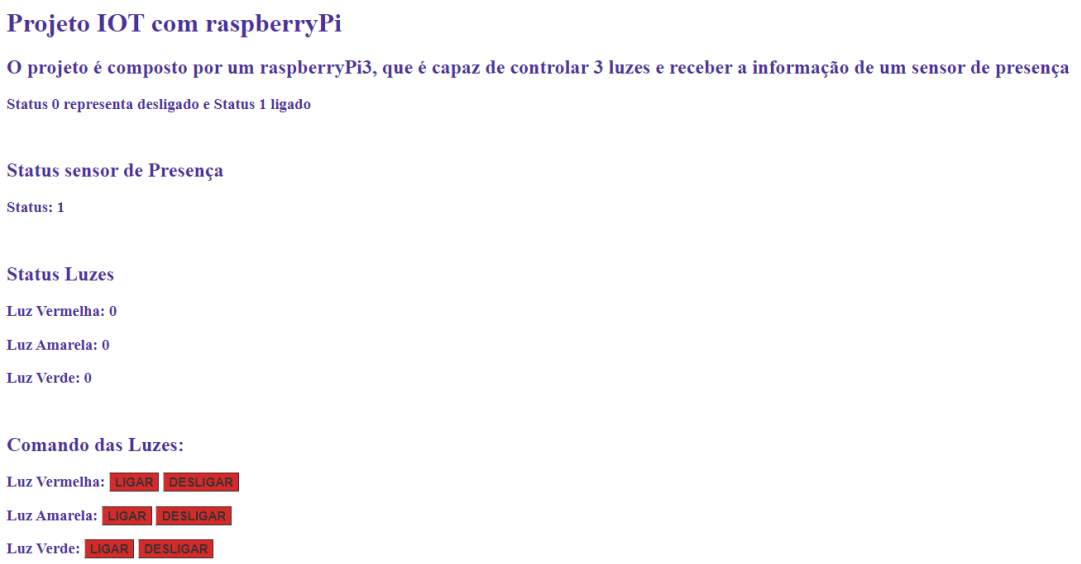

Figura 2: Interface da Aplicação

Nela é apresentado algumas informações, primeiramente um breve resumo sobre do que se trata o projeto. Depois disso, temos a informação do Sensor de presença, quando o status tem valor 0, nao temos movimento na frente do sensor, o que ocorre quando o status passa a ter o valor 1.

Depois, temos os status das 3 luzes, representadas no protótipo pelos LEDS, quando ligadas o status tem valor 1, desligadas, valor 0. No ultimo bloco de informações, temos a parte de comando das luzes. Nos botões podemos ligar e desligar os leds.

A organização do projeto foi feita da seguinte maneira:

Figura 3: Organização dos códigos do projeto

Temos 3 arquivos, sendo o principal o arquivo app.py, master.css que está a estilização da página web, e o index.html, com os textos a criação dos botões. Os códigos estarão no apêndice no final deste arquivo. Vale aqui ressaltar a seguinte parte do código:

```python
@app.route("/")
def index():
    #LE status da GPIO
    senPIRSts = GPIO.input(senPIR)
    ledRedSts = GPIO.input(ledRed)
    ledYlwSts = GPIO.input(ledYlw)
    ledGrnSts = GPIO.input(ledGrn)

    templateData = {
      'senPIR'  : senPIRSts,
      'ledRed'  : ledRedSts,
      'ledYlw'  : ledYlwSts,
      'ledGrn'  : ledGrnSts,
      }
    return render_template('index.html', **templateData)


@app.route("/<deviceName>/<action>")
def action(deviceName, action):
    if deviceName == 'ledRed':
        actuator = ledRed
    if deviceName == 'ledYlw':
        actuator = ledYlw
    if deviceName == 'ledGrn':
        actuator = ledGrn

    if action == "on":
        GPIO.output(actuator, GPIO.HIGH)
    if action == "off":
        GPIO.output(actuator, GPIO.LOW)

    senPIRSts = GPIO.input(senPIR)
    ledRedSts = GPIO.input(ledRed)
    ledYlwSts = GPIO.input(ledYlw)
    ledGrnSts = GPIO.input(ledGrn)

    templateData = {
      'senPIR'  : senPIRSts,
      'ledRed'  : ledRedSts,
      'ledYlw'  : ledYlwSts,
      'ledGrn'  : ledGrnSts,
    }
    return render_template('index.html', **templateData)

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

Figura 4: Trecho do código da aplicação referente ao direcionamento das páginas web

Quando um botão é clicado, ele nos direciona para uma outra página, tendo como rota, a composição de qual led o botão está comandando e o que é para ele realizar, um exemplo, queremos ligar o led verde:

O endereço ficaria: /ledGrn/on , a parte ledGrn será passada para a variável <deviceName> e o on será passado para a variável action. Com essas duas informações, podemos controlar a GPIO do Raspberry.

Para colocar nossa aplicação WEB online e assim podermos controlar nosso protótipo de qualquer lugar, utilizamos o serviço ngrok:

Figura 5: Logo do serviço ngrok

Foi uma boa alternativa ao serviço Heroku, que recentemente deixou de ser gratuito. O ngrok tem a vantagem de ser extremamente fácil a sua implementação e utilização, colocando nossa página disponível em poucos minutos.

Apêndice

Codigo app.py:

app.py > ...

```python
import RPi.GPIO as GPIO #importando bibliotecas para trabalhar com GPIO do rasp
from flask import Flask, render_template, request #Importando Flask

app = Flask(__name__)

GPIO.setmode(GPIO.BCM) #O referenciamento da GPIO sera por BCM
GPIO.setwarnings(False)

#definindo pinos
senPIR = 16

ledRed = 13
ledYlw = 19
ledGrn = 26

#Inicializa os status como desligado
senPIRSts = 0
ledRedSts = 0
ledYlwSts = 0
ledGrnSts = 0

#Definindo pinos como entrada ou saida
GPIO.setup(senPIR, GPIO.IN)

GPIO.setup(ledRed, GPIO.OUT)
GPIO.setup(ledYlw, GPIO.OUT)
GPIO.setup(ledGrn, GPIO.OUT)

GPIO.output(ledRed, GPIO.LOW)
GPIO.output(ledYlw, GPIO.LOW)
GPIO.output(ledGrn, GPIO.LOW)

@app.route("/")
def index():
    #LE status da GPIO
    senPIRSts = GPIO.input(senPIR)
    ledRedSts = GPIO.input(ledRed)
    ledYlwSts = GPIO.input(ledYlw)
    ledGrnSts = GPIO.input(ledGrn)

    templateData = {
      'senPIR'  : senPIRSts,
      'ledRed'  : ledRedSts,
      'ledYlw'  : ledYlwSts,
      'ledGrn'  : ledGrnSts,
      }
    return render_template('index.html', **templateData)

```

```python
@app.route("/")
def index():
    #LE status da GPIO
    senPIRSts = GPIO.input(senPIR)
    ledRedSts = GPIO.input(ledRed)
    ledYlwSts = GPIO.input(ledYlw)
    ledGrnSts = GPIO.input(ledGrn)

    templateData = {
      'senPIR'  : senPIRSts,
      'ledRed'  : ledRedSts,
      'ledYlw'  : ledYlwSts,
      'ledGrn'  : ledGrnSts,
      }
    return render_template('index.html', **templateData)


@app.route("/<deviceName>/<action>")
def action(deviceName, action):
    if deviceName == 'ledRed':
        actuator = ledRed
    if deviceName == 'ledYlw':
        actuator = ledYlw
    if deviceName == 'ledGrn':
        actuator = ledGrn

    if action == "on":
        GPIO.output(actuator, GPIO.HIGH)
    if action == "off":
        GPIO.output(actuator, GPIO.LOW)

    senPIRSts = GPIO.input(senPIR)
    ledRedSts = GPIO.input(ledRed)
    ledYlwSts = GPIO.input(ledYlw)
    ledGrnSts = GPIO.input(ledGrn)

    templateData = {
      'senPIR'  : senPIRSts,
      'ledRed'  : ledRedSts,
      'ledYlw'  : ledYlwSts,
      'ledGrn'  : ledGrnSts,
    }
    return render_template('index.html', **templateData)

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

```

Código master.css

static > # master.css > ...

```css
body {
  background: rgb(255, 255, 255);
  color: rgb(78, 49, 158);
  padding: 30px;
}

.button {
  font: bold 15px Arial;
  text-decoration: none;
  background-color: #d42a2a;
  color: #333333;
  padding: 2px 6px 2px 6px;
  border-top: 1px solid #571d1d;
  border-right: 1px solid #333333;
  border-bottom: 1px solid #333333;
  border-left: 1px solid #CCCCCC;
}
```

Código index.html:

templates > index.html > ...

```html
<!DOCTYPE html>
    <head>
        <title>RaspIOT</title>
        <link rel="stylesheet" href='../static/master.css'/>
    </head>

    <body>
        <h1>Projeto IOT com raspberryPi</h1>
        <h2>O projeto é composto por um raspberryPi3, que é capaz de controlar 3 luzes e receber a informação de um sensor de presença</h2>
        <h3>Status 0 representa desligado e Status 1 ligado</h3>
        <br>
        <h2>Status sensor de Presença</h2>
        <h3></h3>
        <h3> Status: {{ senPIR  }}</h3>
        <br>

        <h2> Status Luzes </h2>
        <h3> Luz Vermelha:   {{ ledRed  }}</h3>
        <h3> Luz Amarela:   {{ ledYlw  }}</h3>
        <h3> Luz Verde:   {{ ledGrn  }}</h3>
        <br>
        <h2> Comando das Luzes: </h2>
        <h3>
            Luz Vermelha:
            <a href="/ledRed/on" class="button">LIGAR</a>
            <a href="/ledRed/off"class="button">DESLIGAR</a>
        </h3>
        <h3>
            Luz Amarela:
            <a href="/ledYlw/on" class="button">LIGAR</a>
            <a href="/ledYlw/off"class="button">DESLIGAR</a>
        </h3>
        <h3>
            Luz Verde:
            <a href="/ledGrn/on" class="button">LIGAR</a>
            <a href="/ledGrn/off"class="button">DESLIGAR</a>
        </h3>

    </body>
</html>

```