

Projeto final 2

Enrico Sampaio Bonela

11721EMT007

Guilherme Salomão Agostini

11721EMT003

Jadson Silva Souza

11721EMT017

Leonardo França De Carvalho

11821EMT012

Rafael Lins Nobre

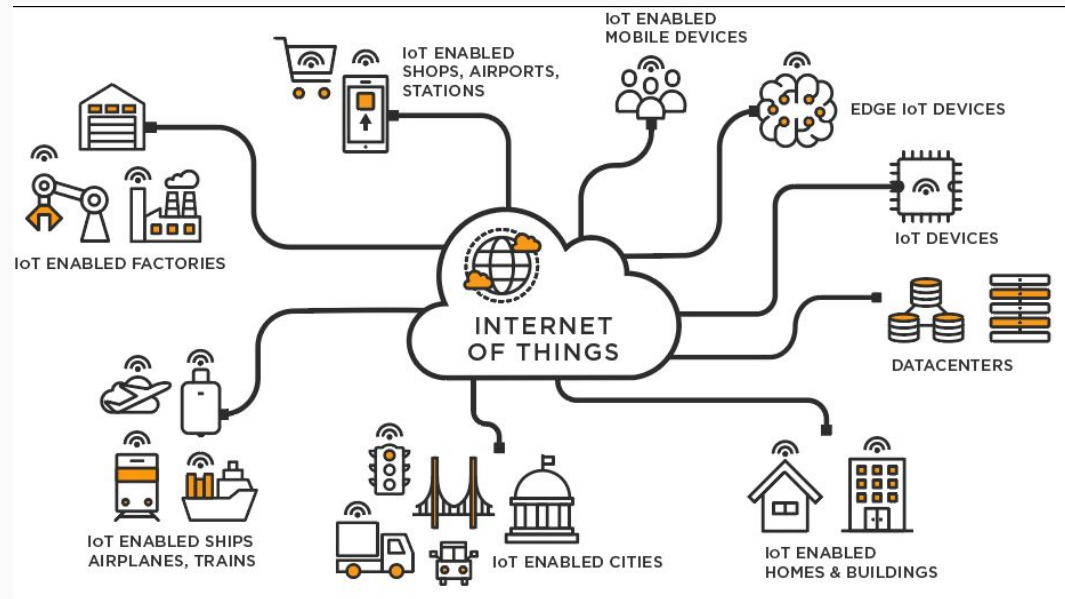
11811EMT002

Vitor Hugo Vasconcelos De Melo

11821EMT006

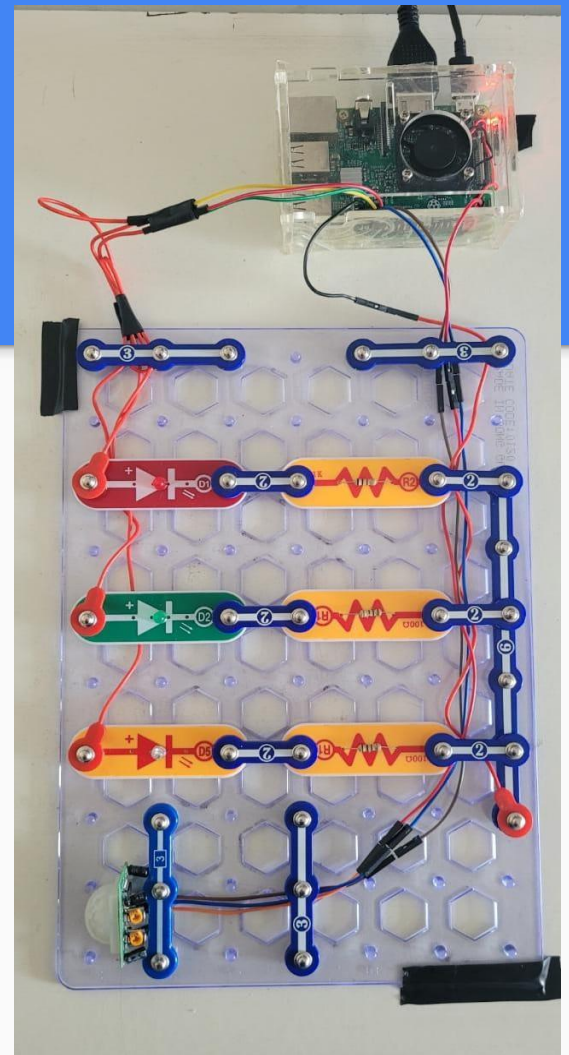
O que é IOT?

- Dispositivos Inteligentes;
- Aplicação;
- Interface gráfica.



Materiais utilizados

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



Interface da Aplicação Web

Projeto IOT com raspberryPi

O projeto é composto por um raspberryPi3, que é capaz de controlar 3 luzes e receber a informação de um sensor de presença

Status 0 representa desligado e Status 1 ligado

Status sensor de Presença

Status: 1

Status Luzes

Luz Vermelha: 0

Luz Amarela: 0

Luz Verde: 0

Comando das Luzes:

Luz Vermelha:

Luz Amarela:

Luz Verde:

Desenvolvimento e Organização da API

Para a marcação e estilização foram utilizados HTML e Css respectivamente. O app foi desenvolvido utilizando Python.

▼ PROJETO FINAL 2 - RASPBERRY IOT [WSL: UBUNTU]

- ▼ static
 - # master.css M
- ▼ templates
 - <> index.html M
- 🔗 app.py 2, M

```
import RPi.GPIO as GPIO #importando bibliotecas para GPIO
from flask import Flask, render_template, request

app = Flask(__name__)

GPIO.setmode(GPIO.BCM) #0 referenciamento da GPIO
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 saída
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)

@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)
```

Para colocar a aplicação WEB online e assim podermos controlar o protótipo de qualquer lugar, foi utilizado o serviço ngrok, que se mostrou uma boa alternativa ao Heroku que deixou de ser gratuito recentemente. O ngrok tem a vantagem de ser extremamente fácil a sua implementação e utilização, colocando nossa página disponível em poucos minutos.

The logo for ngrok, featuring the word "ngrok" in a bold, dark blue, sans-serif font. The letters are thick and modern, with a slight shadow effect behind them.

Funcionamento do Projeto