



Universidade Federal de Uberlândia
Faculdade de Engenharia Elétrica
Sistemas Embarcados II

Sistemas Embarcados II

Trabalho Prático 2

Aluno: Rafael Dias Pereira

nº de matrícula: 11911EAU003

Professor: Éder Alves de Moura

17 de agosto de 2022

Conteúdo

1	Objetivos	1
2	Desenvolvimento	2
2.1	Aplicação Flask	2
2.2	Templates	2
2.3	Hospedagem	3
2.4	Firmware	4
2.5	Circuito eletrônico	5

Objetivos

Neste trabalho, será demonstrado o funcionamento de um projeto de IoT através do uso de um microcontrolador ESP32, juntamente com ferramentas como Flask, HTML, CSS, JavaScript e AWS.

Desenvolvimento

Aplicação Flask

Em primeiro lugar, foi preciso criar as rotas para o compartilhamento de dados via HTTP. Para isso, utilizei o framework Flask, dentro do Python, para prover as rotas necessárias, conforme o exemplo a seguir. 1).

```
1 from flask import Flask, redirect, render_template, request, jsonify
2
3 gpio = {'lampada_quarto': ['off', '65535'], 'ventilador_quarto': ['off', '65535'],
4         'lampada_suite': ['off', '65535'], 'climatizador_suite': ['off', '65535'],
5         'lampada_sala': ['off', '65535'], 'ventilador_sala': ['off', '65535'],
6         'lampada_cozinha': ['off', '65535'], 'forno': ['off', '65535'], 'liquidificador': ['off', '65535']}
7
8 temperatura = 0
9
10 app = Flask(__name__)
11
12 @app.route('/', methods=['GET', 'POST'])
13 def index():
14     return render_template('index.html')
```

Figura 1 – Aplicação flask.

Templates

Em conjunto com o Flask, também precisamos de páginas web onde será a interface do projeto e de onde virão as informações a serem processadas. A página inicial é representada por uma barra ("/") e direciona para o arquivo index.html, conforme a seguir (Imagem 2).

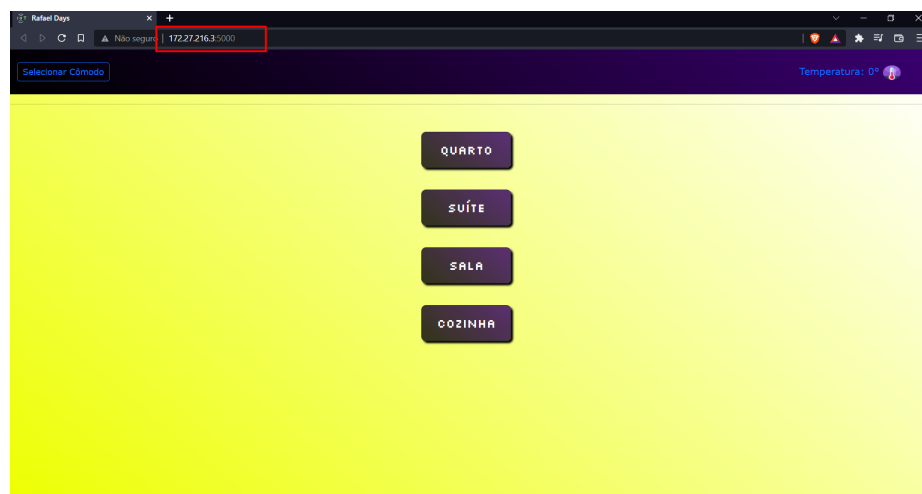


Figura 2 – Página inicial.

Observa-se que a página inicial já está sendo indexada no endereço da rede. A partir daí, apenas continuei programando em Flask e criei uma rota para cada cômodo da casa (quarto, cozinha, sala e suíte).

Na imagem a seguir, podemos ver como é a página de controle da cozinha (Imagem 3).

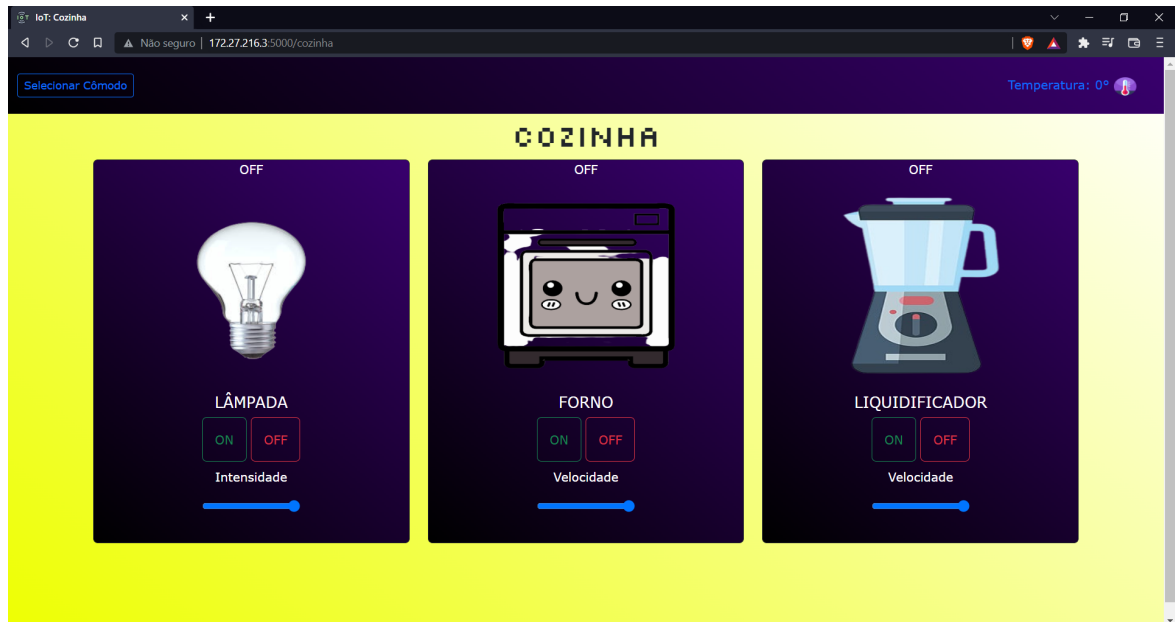


Figura 3 – Página de controle da cozinha.

Nota-se que para cada item, temos três possibilidades: ligar, desligar e controlar a sua intensidade enquanto ligado. O mesmo ocorre para os demais cômodos, a única diferença são os itens.

Hospedagem

Não basta criar a aplicação, também precisamos terceirizar sua execução. Para isso, utilizei o serviço AWS Elastic Beanstalk, que conseguiu manter o meu programa em execução na nuvem (Imagem 4).

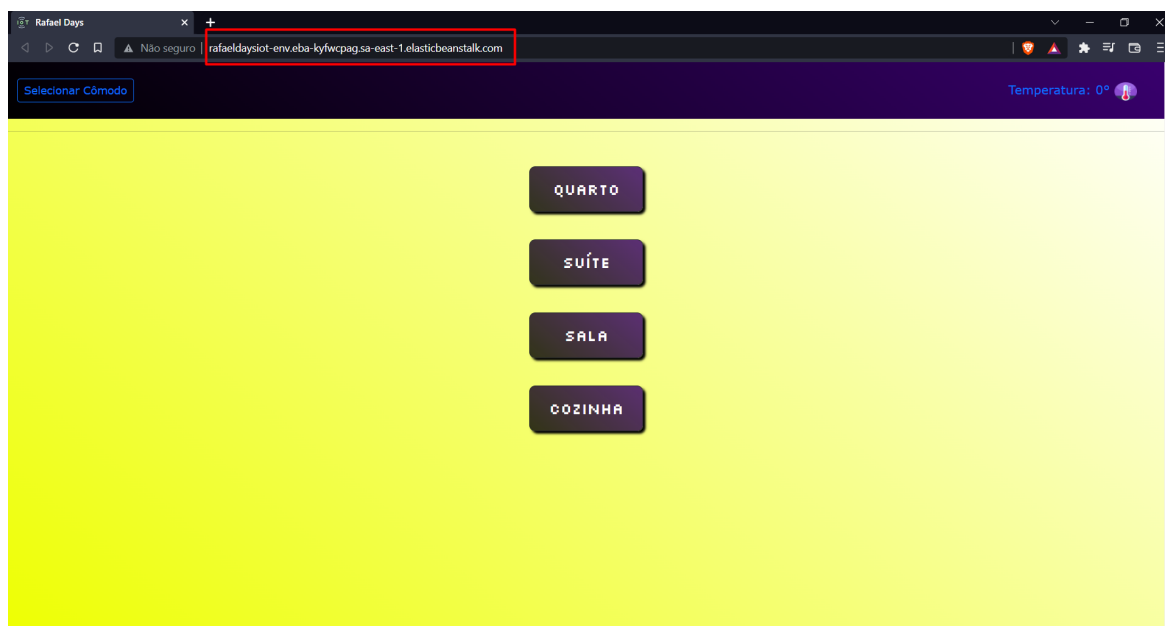


Figura 4 – Demonstração da aplicação na nuvem.

Firmware

Com a base front-end criada e hospedada, o próximo passo foi programar o hardware. O dispositivo escolhido foi o ESP32, capaz de se conectar à uma rede WiFi e acessar o website desenvolvido previamente. Além disso, uma ferramenta importante que também optei por usar é o MciroPython, um framework que possibilita o uso da linguagem Python na programação do microcontrolador. O código abaixo demonstra como realizei os requests (Imagem 5).

```
urequests.get('http://rafaeldaysiot-env.eba-kyfwcpag.sa-east-1.elasticbeanstalk.com/get-data')
```

Figura 5 – Trecho de código do firmware.

Através do método "get", foi possível receber as informações do rota Flask que continha todos os comandos necessários. Abaixo podemos observar como esses comandos são enviados (Imagem 6).

```
rafaeldaysiot-env.eba-kyfwcpag... x +
NÃO seguro | rafaeldaysiot-env.eba-kyfwcpag.sa-east-1.elasticbeanstalk.com/get-data

{"climatizador_suite":["off","65535"],"forno":
["off","65535"],"lampada_cozinha":["off","65535"],"lampada_quarto":
["off","40999"],"lampada_sala":["off","65535"],"lampada_suite":
["off","65535"],"liquidificador":["off","65535"],"ventilador_quarto":
["off","65535"],"ventilador_sala":["off","65535"]}
```

Figura 6 – Rota Flask de referência.

Circuito eletrônico

A implementação prática foi realizada com LEDs conectados em pinos diferentes que representam cada item da casa. A Imagem 7 apresenta o esquema geral.

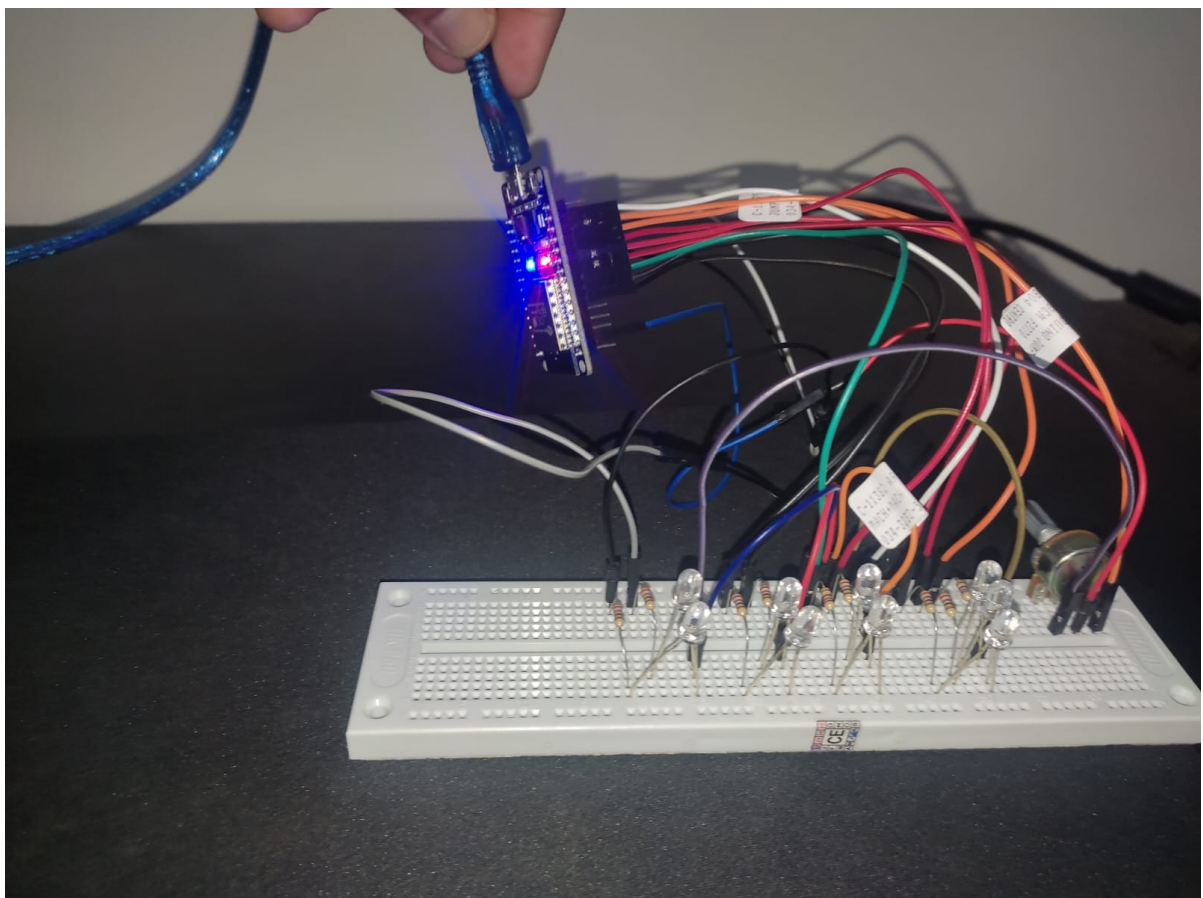


Figura 7 – Circuito eletrônico.