

**UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**AUTOMAÇÃO RESIDENCIAL UTILIZANDO RASPBERRY PI PARA AUXÍLIO A
PESSOAS COM DEFICIÊNCIA MOTORA**

Bruno Paiva Passini

Trabalho de Graduação apresentado ao Departamento de Engenharia Elétrica da Universidade Federal de São Carlos como parte dos requisitos para obter título de Bacharel em Engenharia Elétrica.

Orientador: Prof. **Robson Barcellos**

São Carlos – SP

2022

BANCA EXAMINADORA

Trabalho de graduação apresentado ao departamento de engenharia elétrica no dia 22 de Junho de 2022 perante a seguinte banca examinadora:

Orientador: Robson Barcellos, DEE/UFSCar.

Nome do avaliador	Avaliação (aprovado/reprovado)	Assinatura
Robson Barcellos		
Osmar Ogashawara		
Ricardo Augusto Souza Fernandes		

AGRADECIMENTOS

Gostaria de agradecer a meu pai e a minha irmã pelo apoio nos últimos anos. A ajuda para cuidar de minha mãe, que foi minha inspiração para o desenvolvimento do sistema desse trabalho.

Gostaria de agradecer a todos os meus amigos também pelo apoio em momentos difíceis.

Tenho que agradecer muito também ao meu orientador Robson, que por curiosidade foi o professor que me deu a primeira aula na UFSCar no meu primeiro dia na faculdade e acabou se tornando meu orientador na última matéria da faculdade.

Gostaria de agradecer também a todos os professores do curso de engenharia elétrica pela ajuda e todos os ensinamentos e por sempre orientar a encontrar os caminhos para resolver os problemas e dificuldades do dia a dia no mundo da engenharia elétrica.

RESUMO

Este trabalho apresenta um sistema que foi desenvolvido para permitir que pessoas com dificuldades motoras possam comandar, à distância, o funcionamento de dispositivos que atuam no bem-estar ambiente. Comandos são inseridos no sistema através de um telefone celular que se comunica com um computador de placa única, o Raspberry Pi, que recebe os comandos do usuário de ligar ou desligar. Um sensor de temperatura e umidade é usado para controlar a faixa de operação de um aquecedor e um umidificador, quando ligados.

ABSTRACT

This work presents a system that was developed to allow people with motor difficulties to remotely control the operation of devices that act on the well-being of the environment. Commands are entered into the system via a cellphone that communicates with a single-board computer, the Raspberry Pi, which receives user commands to turn it on or off. A temperature and humidity sensor are used to control the operating range of a heater and humidifier when turned on.

LISTA DE FIGURAS

Figura 1	Raspberry Pi Zero 2 W.....	11
Figura 2	Tela de edição de texto do Visual Studio Code.....	12
Figura 3	Exemplo de código em HTML.....	13
Figura 4	Exemplo de código em CSS.....	13
Figura 5	Exemplo de código em Javascript.....	14
Figura 6	Tela inicial do Iphone 6S com o ícone Assistente.....	16
Figura 7	Tela inicial da interface com o usuário.....	17
Figura 8	Deslizando tela inicial para cima com outras opções.....	18
Figura 9	Fluxograma operacional do sistema.....	19
Figura 10	Fluxograma de funcionamento do software.....	20
Figura 11	Fluxograma de funcionamento das campanhas.....	21
Figura 12	Fluxograma de funcionamento da lâmpada.....	22
Figura 13	Fluxograma de funcionamento do ventilador.....	23
Figura 14	Fluxograma de funcionamento do aquecedor.....	24
Figura 15	Fluxograma de funcionamento do umidificador.....	25
Figura 16	Foto do DHT11.....	27
Figura 17	Módulo relé.....	27
Figura 18	Diagrama elétrico.....	29

LISTA DE TABELAS

Tabela 1	Arquitetura e interfaces do Raspberry Pi Zero 2 W.....	11
Tabela 2	Datasheet DHT 11.....	26
Tabela 3	Especificações dos buzzers.....	28
Tabela 4	Preço dos componentes.....	30
Tabela 5	Medições de corrente do sistema.....	31

SUMÁRIO

INTRODUÇÃO	8
Motivação	8
Objetivos	8
REVISÃO BIBLIOGRÁFICA	9
MATERIAIS E MÉTODOS	10
Raspberry Pi e suas características eletrônicas	10
Programação do sistema	11
Linguagens utilizadas	12
HTML	12
CSS	13
Javascript	14
Python	14
Framework Flask	15
Interface gráfica	15
Quedas de energia e funcionamento automático	19
Funcionamento do sistema e seus dispositivos	19
Componente de hardware	26
Custo do sistema	30
RESULTADOS E DISCUSSÕES	31
CONCLUSÕES	32
TRABALHOS FUTUROS	33
REFERÊNCIAS BIBLIOGRÁFICAS	34

1 INTRODUÇÃO

1.1 Motivação

A automação residencial pode promover conforto e segurança para pessoas com deficiência motora, como alguém em uma cadeira de rodas. Ao toque de um botão, ela pode chamar um cuidador do outro lado da casa, ou ligar um ventilador, entre outros. Atualmente, são vendidos alguns dispositivos que fazem essa interface através da internet ou Bluetooth.

A umidade e temperatura, duas variáveis controladas pelo projeto, são de muita importância para certos tipos de pacientes, como pessoas acamadas e idosos que podem ter uma grande facilidade de desenvolver pneumonias [1].

Hospitais possuem sistemas que controlam temperatura e umidade, além de sistemas para chamar enfermeiros através de campainhas. Este projeto emula estes sistemas.

Tomadas e interruptores Wi-Fi costumam ter o preço muito elevado e quando um grande número destes equipamentos é necessário, o custo se torna grande demais e inviável para a grande parte da população. Isto é levado em consideração neste projeto e tem como objetivo ajudar essas pessoas construindo um sistema modular que atende pessoas com deficiência motora, da forma mais barata possível.

1.2 Objetivo

O objetivo do projeto é realizar o controle de dispositivos por meio de comandos dados em uma página de internet mostrada na tela de um telefone celular. Esses dispositivos são: umidificador, buzzer, ventilador, aquecedor e lâmpada.

2 REVISÃO BIBLIOGRÁFICA

Para realização do projeto, foram estudados os trabalhos de monografia de Matheus Hoffman Silva[2] e do José Guilherme Vitoratto[3], em que ambos utilizam o Raspberry Pi para uma solução para automação residencial.

O trabalho de Matheus usa um controlador ZWave que trabalha utilizando ZigBee, que é um protocolo de comunicação por onda de rádio.

O EaZy é o software de sistema de automação residencial usado no trabalho de Matheus, que junto com o ZWave Open, um protocolo de comunicação, torna possível controlar dispositivos como lâmpadas a distância.

Segundo Mateus, o EaZy pode ter algumas incompatibilidades com alguns sistemas operacionais. Além disso, o controlador ZWave e o software do ZWave Open podem não ser totalmente compatíveis.

Já o trabalho de José Vitoratto utiliza o Android SDK, um sistema de desenvolvimento de softwares para Android para criar aplicativos e ter o controle dos objetos controlados pelo projeto. Também utiliza uma página Web para comunicação e controlar objetos com dispositivos que não tenham sistema Android.

José utiliza a linguagem de programação Java para construção do servidor local e menciona que esta linguagem não é a mais recomendada por não ser otimizada no gasto de memória. Apesar destas questões, menciona que o servidor rodou sem problemas.

O seu projeto faz acionamento de lâmpadas e monitoramento de temperatura e umidade.

Dentre os dois trabalhos, o de José é o que mais se assemelha a este pois não utiliza materiais além do Raspberry Pi para o controle e opera através de software.

Os trabalhos mencionados funcionam de forma semelhante ao deste projeto, por criar um servidor local e fazer com que o Raspberry Pi controle os dispositivos, como lâmpadas e outros.

3 MATERIAIS E MÉTODOS

3.1 Raspberry Pi e suas características eletrônicas

Raspberry Pi é um microcomputador de baixo custo criado com o objetivo de possibilitar o aprendizado de programação sem grandes custos. Também possui GPIOs que tornam possíveis as criações de projetos eletrônicos como projetos utilizando microcontroladores.

ARM (Advanced RISC Machine) é um tipo de arquitetura de processadores. RISC (Conjunto Reduzido de Instruções de Computador) como o nome diz, são instruções que o processador de forma reduzida executa de forma eficiente para realizar suas funções como abrir um aplicativo. Isso faz com que menos energia seja necessária e que ele seja mais eficiente e esquente menos.

O Raspberry Pi usa um microprocessador ARM.

O projeto utiliza um microcomputador Raspberry Pi Zero 2 W, a versão mais barata dentro das versões com placa de rede Wi-Fi.

Por ser mais barata e por atender todos os requisitos do projeto, foi a escolhida. Porém existem outras versões como o Raspberry Pi 3B que conta com um processador mais rápido e conector USB do tipo A.

Pinos de propósitos gerais são chamados de GPIO (General Purpose Input Output). O Raspberry Pi possui 40 pinos sendo 28 GPIO (General Purpose Input Output) programáveis e 12 pinos que não são programáveis pelo usuário.

Além dos 28 GPIOs, também existem pinos para linha de 5V e pinos de terra, resultando em 40 pinos totais. Como o Raspberry é um computador funcional, ele também apresenta duas entradas micro-USB, saída HDMI, entrada de cartão de memória, além de saída de vídeo RCA para utilizar em televisões antigas. Para o funcionamento é necessária uma fonte de 2,5A e 5V[4].

Na Tabela 1 são mostradas mais algumas informações técnicas sobre o Raspberry Pi Zero 2 W.

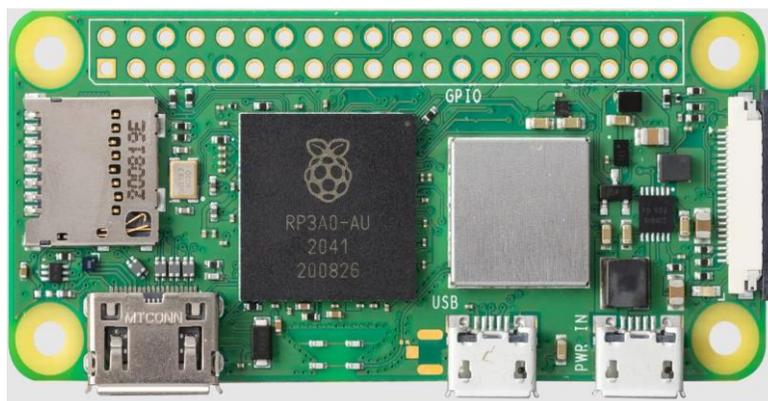
Tabela 1 – Arquitetura e interfaces do Raspberry Pi Zero 2 W.

Raspberry Pi Zero W	
Processador	1GHz quad-core 64-bit Arm Cortex-A53 CPU
GPU:	OpenGL ES 1.1, 2.0 graphics
Memória:	512MB SDRAM
Wi-Fi:	2.4GHz 802.11 b/g/n wireless LAN
Bluetooth:	Bluetooth 4.2, Bluetooth Low Energy (BLE), onboard antenna
Alimentação:	5V
Conector de vídeo:	mini HDMI/RCA
Porta USB OTG:	micro usb
GPIO:	40
Conector CSI:	Câmera

Fonte: Documentação do Raspberry Pi[4].

Na Figura 1 é mostrada uma foto de um Raspberry Pi Zero 2 W.

Figura 1: Raspberry Pi Zero 2 W.



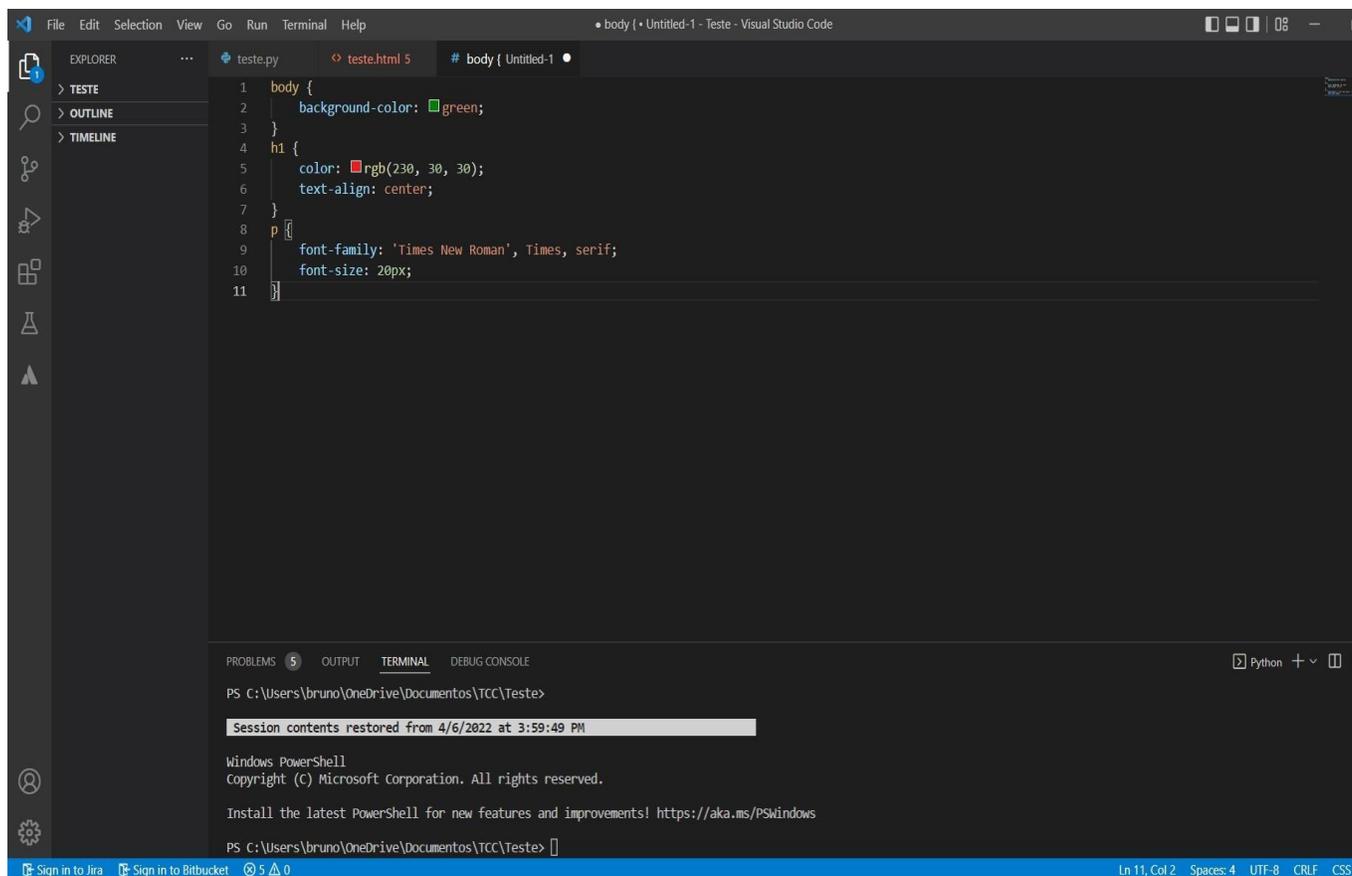
Fonte: Fundação Raspberry Pi[5].

3.2 Programação do sistema

O editor de texto escolhido para programação foi o Visual Studio Code por ser um dos mais utilizados e por possuir versão para o Raspberry Pi.

Na Figura 2 é mostrada a interface do editor de texto.

Figura 2: Tela de edição de texto do Visual Studio Code.



Fonte: Autoria própria.

3.2.1 Linguagens utilizadas

3.2.1.1 HTML

HTML (Linguagem de Marcação de HiperTexto) [6] é a codificação responsável pelo esqueleto da página de internet. Através dela definimos a formatação e textos que aparecem na tela do usuário.

Essa codificação foi escolhida por ser a mais utilizada em desenvolvimento de páginas de internet e ser compatível com diversos dispositivos, além disso, é de fácil compreensão e possui vários tutoriais de como usá-la.

Podemos definir através de HTML um título, onde este deve ser inserido, adicionar um botão e toda estrutura da página de internet.

Na Figura 3 é mostrado um exemplo de código em HTML feito no editor de texto VScode. Nele temos separado a estrutura da página por *head* (cabeça da página), *title* (título da página), *body* (corpo da página) e dentro do corpo um cabeçalho (<h1>) e um parágrafo

(<p>).

Figura 3: Exemplo de código em HTML.

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Título</title>
5  </head>
6  <body>
7  <h1>cabeçalho</h1>
8  <p>Parágrafo</p>
9  </body>
10 </html>

```

Fonte: Autoria própria

3.2.1.2 CSS

CSS (Folhas de Estilo em Cascata) [7] é a codificação que complementa o que foi codificado em HTML dando as características visuais da página de internet e por esse motivo também foi utilizada. Em CSS é possível dar cor a um texto, o tamanho da fonte, escolher uma fonte diferente.

Na Figura 4 é mostrado um exemplo de código em CSS.

Figura 4: Exemplo de código em CSS.

```

1  body {
2    background-color: green;
3  }
4  h1 {
5    color: rgb(230, 30, 30);
6    text-align: center;
7  }
8  p {
9    font-family: 'Times New Roman', Times, serif;
10   font-size: 20px;
11 }

```

Fonte: Autoria própria.

No trecho de código da Figura 4 o CSS faz com que o *body* criado em HTML tenha um fundo verde, e adiciona outras características visuais a página.

3.2.1.3 Javascript

Javascript é a linguagem de programação que adiciona funções para página de internet. Através do Javascript podemos fazer com que a página recarregue a cada 5 minutos para o caso de alguma queda de rede ou energia e ela se torne disponível novamente.

Esta linguagem foi escolhida por também ser uma linguagem que complementa HTML e CSS e é muito utilizada.

Na Figura 5 é possível ver um exemplo de código em Javascript complementando o código HTML.

Figura 5: Exemplo de código em Javascript.

```
<body onload="update_values(); JavaScript:AutoRefresh(35000);" >  
</body>
```

Fonte: Autoria própria.

Essa adição do Javascript ao elemento *body* do HTML faz com que ele se atualize sozinho a cada 35000ms.

3.2.1.4 Python

Python é uma linguagem de programação criada pelo holandês Guido van Rossum. É uma linguagem de alto nível possuindo uma curva de aprendizado mais fácil pois linguagens de alto nível se aproximam mais da linguagem falada e se distanciam da linguagem de máquinas.

É uma linguagem com orientação a objetos, ou seja, os dados podem possuir atributos e métodos que também podem ser chamados de procedimentos. Isso faz com que projetos em larga escala sejam muito mais simples de organizar e alterar sem causar grandes impactos na estrutura geral do código.

Python também foi utilizada por vir instalada no sistema operacional Raspbian, criado pela própria fundação Raspberry Pi, além de ser uma linguagem de fácil aprendizado.

Python também possui uma boa documentação [8] online.

3.2.2 Framework Flask

Um framework funciona como se fosse uma biblioteca de programação. Dentro desses frameworks existem combinações de funções simples em funções mais complexas. Frameworks têm como características a possibilidade de reutilização, modularização, entre outros, para se adaptar ao projeto que ele será implementado.

O Flask é um framework que utilizando a programação Python, HTML, CSS e Javascript possibilita a criação de uma página de internet ser disponível no servidor local rodando no Raspberry Pi.

O servidor local é necessário para que haja a troca de informações entre um dispositivo como celular, tablet ou televisão que está controlando os dispositivos ligados ao Raspberry Pi.

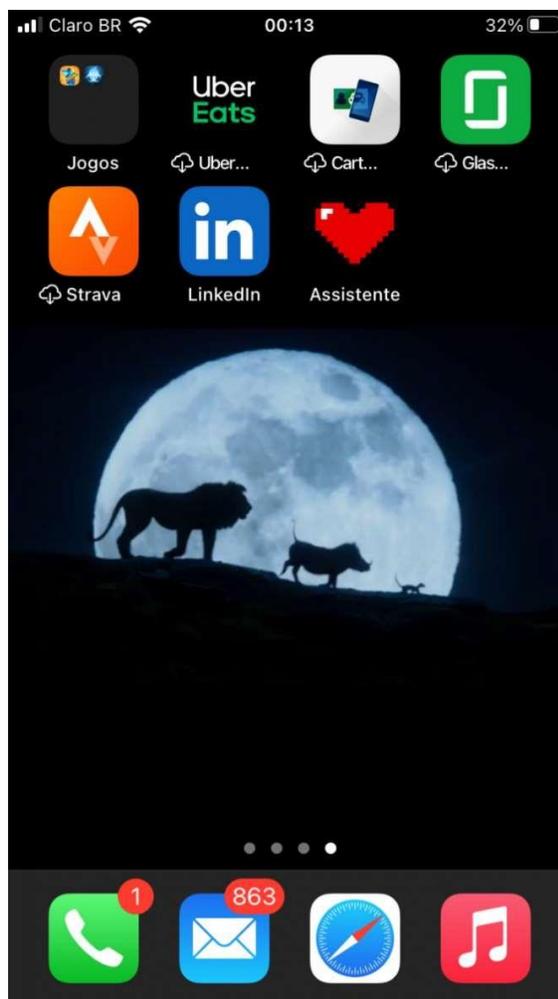
Por padrão, o Flask usa uma estrutura de arquivos para seu funcionamento, essa estrutura é dada por um arquivo e duas pastas: o arquivo é o código do sistema em Python, e duas pastas chamadas *static* e *templates*. Assim, foi criada uma pasta com o arquivo do código em Python e duas pastas chamadas *static* e outra chamada *templates*.

Na pasta *static* se encontram os arquivos do código em CSS e as figuras utilizadas na página. Já na pasta *templates* se encontra o código em HTML e Javascript da página.

3.2.3 Interface Gráfica

Na Figura 6 é possível ver como fica o ícone de coração chamado Assistente adicionado à tela inicial de um Iphone 6S rodando IOS14. Este ícone de coração funciona como um atalho para página de internet criada.

Figura 6: Tela inicial do Iphone 6S com o ícone Assistente.



Fonte: Autoria Própria.

A comunicação da página de internet da interface gráfica em HTML, CSS, Javascript se dá do Flask com a programação em Python através de requisições *GET* (receber) e *POST* (publicar).

GET acontece quando o usuário está com a página aberta e essa faz requisições de uma informação ao servidor. Nesse projeto a linguagem Javascript atualiza periodicamente os valores de temperatura e umidade requisitando através de *GET* para poder disponibilizar na página.

POST acontece quando uma informação é entrada na página e enviada ao servidor. Nesse projeto ao fazer uma entrada para ligar o ventilador, a requisição *POST* disponibiliza essa informação para que o código em Python possa interpretar o comando.

Para criação da interface gráfica primeiramente foi adicionado um título na

programação em HTML chamado “Boas-vindas :)!” como visto na Figura 5. Logo abaixo há uma referência de imagem (onde essa deve ser mostrada) utilizando HTML, e com a utilização de CSS podemos incluir a visualização da imagem.

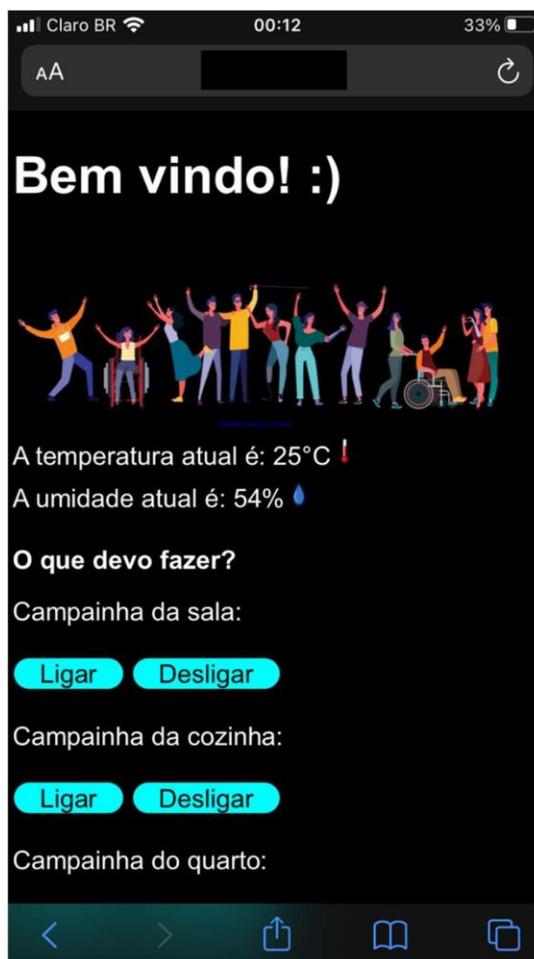
Na Figura 7 é mostrada a tela inicial da interface do usuário com as mensagens de boas-vindas, a apresentação da temperatura e umidade atual e botões de comando para ligar e desligar os equipamentos.

Ao apertar um botão da interface gráfica criada, o Flask faz com que a linguagem Python entenda que aquele botão é uma entrada para o código, fazendo com que o Raspberry Pi ligue ou desligue seus GPIO.

Cada GPIO tem uma frase ligada a ele que foi associada pelo código em Python. Essas frases são, por exemplo, “Ligar1”, “Ligar2”, etc.

Os botões da interface gráfica fazem o código em Python receber uma frase que liga os respectivos pinos de GPIO.

Figura 7: Tela inicial da interface com o usuário.



Fonte: Autoria Própria.

Posteriormente foi adicionado os parágrafos que indicam a temperatura e umidade com HTML e duas figuras representativas da temperatura e umidade como visto na Figura 7. Além disso, entre os dois pontos e as figuras há um espaço para as variáveis de temperatura e umidade.

Esses valores de temperatura e umidade são disponibilizados na página utilizando o Javascript. A cada 200ms o Javascript faz uma requisição do tipo GET ao servidor local rodando o código do sistema em Python que faz a medição de temperatura e umidade. O retorno dessa função são os valores de temperatura e umidade, e com Javascript, os valores do retorno são possíveis de serem mostrados na página.

Deslizando a tela inicial para cima são mostradas mais opções na Figura 8 onde temos também os botões do modo automático e os valores desejados de temperatura e umidade.

Figura 8: Deslizando tela inicial para cima com outras opções.



Fonte: Autorial Própria.

Para a entrada dos valores de temperatura e umidade desejados foram adicionadas duas entradas em HTML do tipo número.

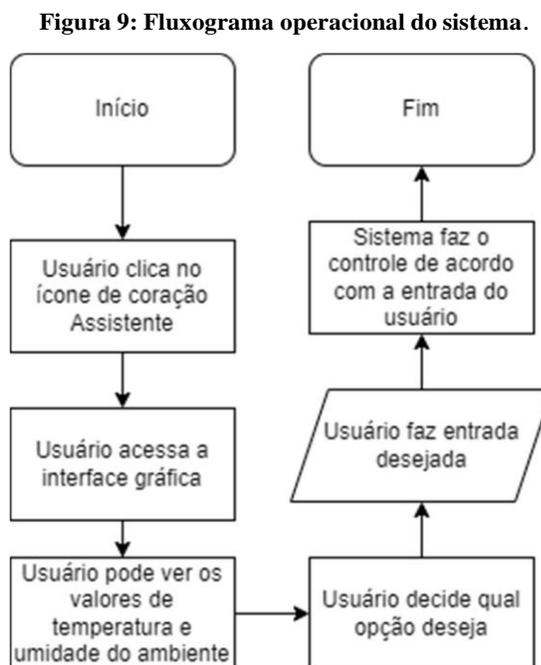
3.2.4 Quedas de energia e funcionamento automático

Caso haja queda de energia, o Raspberry Pi foi programado para que ao voltar a energia, inicie rodando o código Python do projeto para que não seja necessário inicializar manualmente.

Isso foi feito criando um serviço que sempre ao iniciar o Raspberry Pi esse serviço também seja iniciado e após estabelecer uma conexão com a internet o código em Python entre em execução automaticamente.

3.3 Funcionamento do sistema e seus dispositivos

Na Figura 9 é mostrado um fluxograma operacional do sistema.



Fonte: Autoria própria.

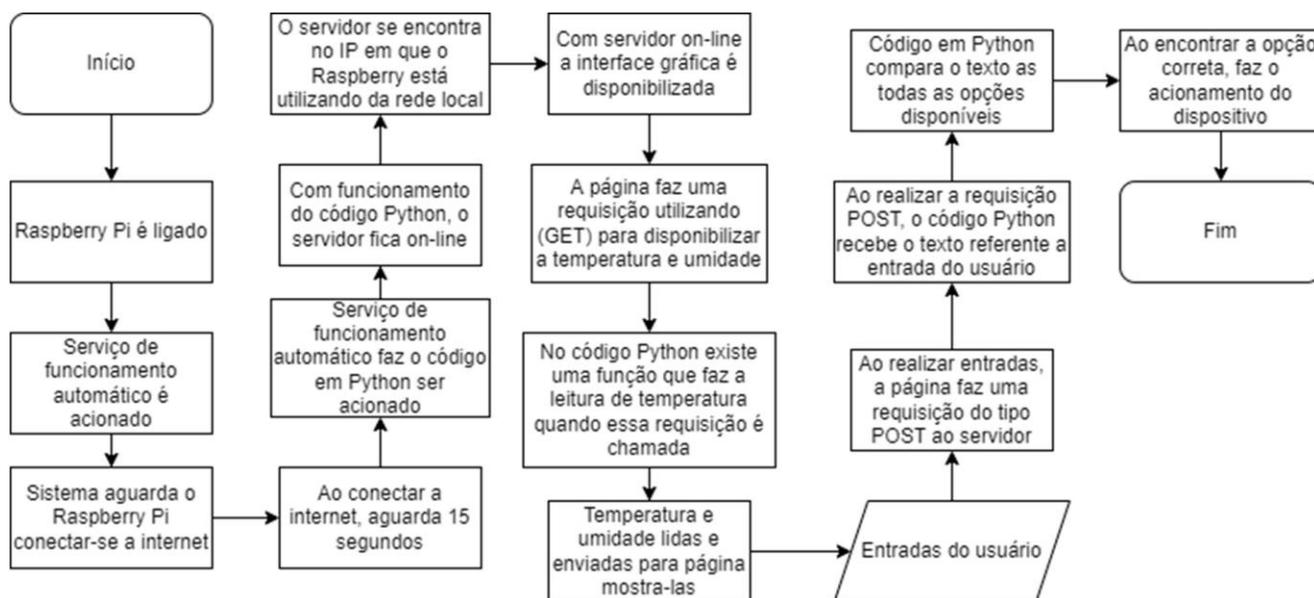
Como mostrado no fluxograma, o primeiro passo é clicar no ícone Assistente da interface gráfica, ao fazer isso, a página criada irá aparecer e o usuário tem acesso a interface gráfica. Na interface gráfica é possível ver os valores de temperatura e ambiente.

Com base na condição do ambiente que o usuário se encontra, ele deve decidir o que fazer, como ligar as campainhas, lâmpada, ventilador, entre outros. Após a decisão, o usuário faz a entrada utilizando os botões da interface gráfica.

Com as entradas registradas, o Raspberry fará o acionamento de acordo com a opção selecionada.

Na Figura 10 é mostrado um fluxograma de funcionamento do software.

Figura 10: Fluxograma de funcionamento do software.



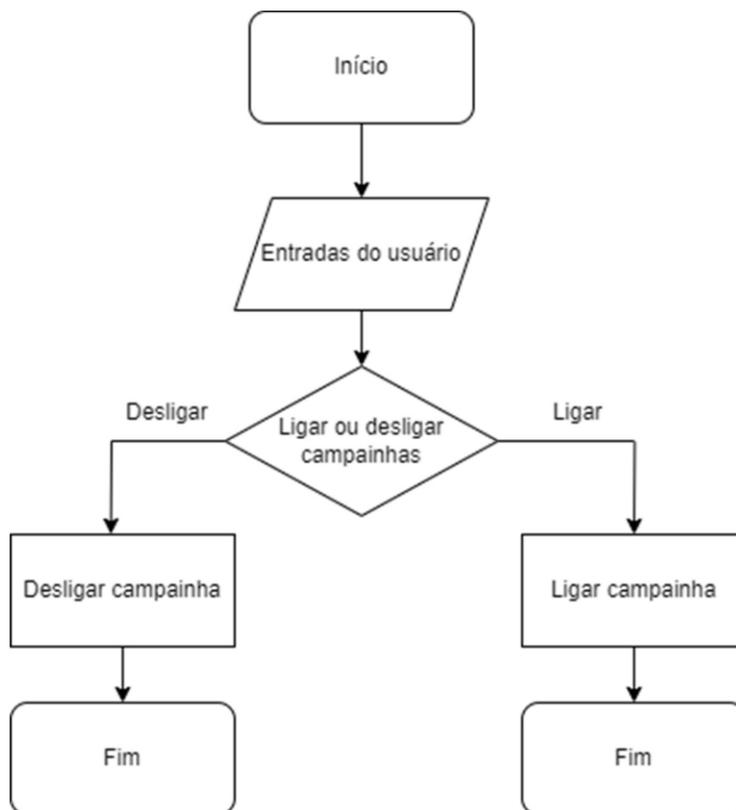
Fonte: Autoria própria.

O modo automático regula a temperatura da seguinte maneira: caso a temperatura de *setpoint* esteja 2 °C acima da temperatura medida, o ventilador deve ser ligado. Caso a temperatura *setpoint* esteja 4 °C abaixo da medida, deve ser ligado o aquecedor. Esses valores foram encontrados através de testes experimentais variando os valores acima e abaixo da temperatura de *setpoint* visando um controle ótimo de temperatura.

Já a umidade é regulada da seguinte maneira: caso a umidade medida seja maior que a umidade de *setpoint* deve ser desligado o umidificador. Caso a porcentagem de umidade *setpoint* seja 10% maior que a medida, o umidificador deve ser ligado. Caso a umidade medida seja 10% maior que a umidade *setpoint* deve ser ligado o aquecedor.

Na Figura 11 é mostrado o funcionamento de das campainhas.

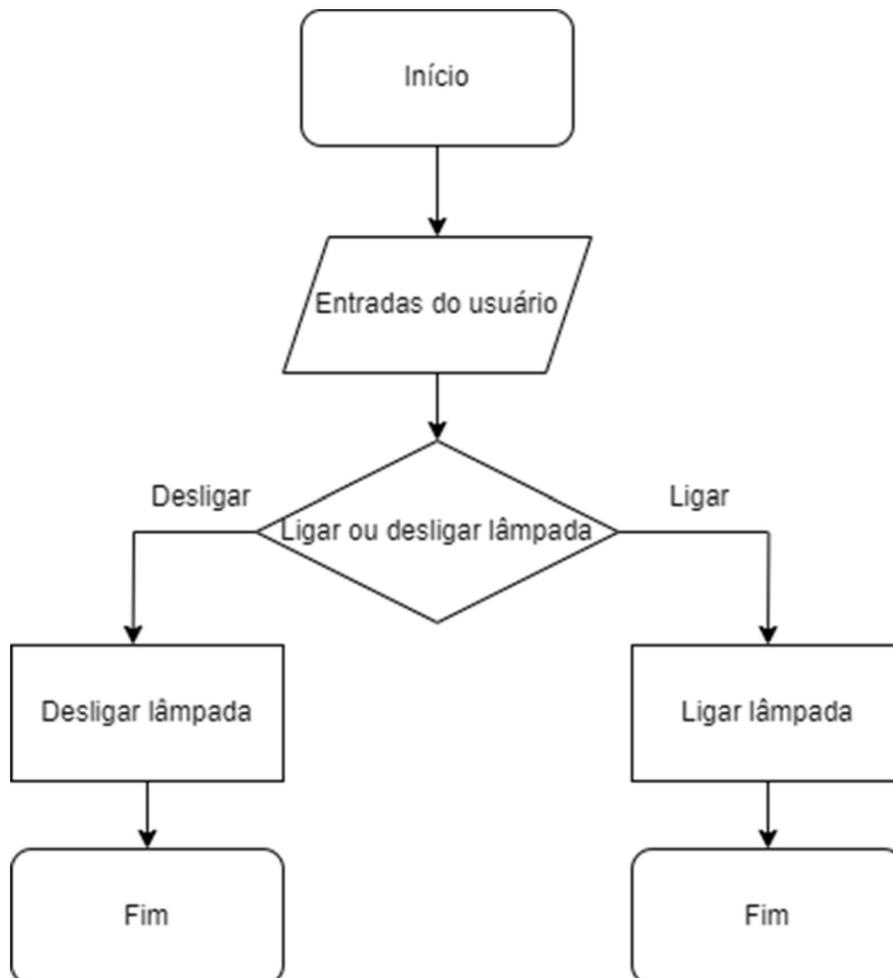
Figura 11: Fluxograma de funcionamento das campainhas.



Fonte: Autoria própria.

Na Figura 12 é mostrado o funcionamento da lâmpada.

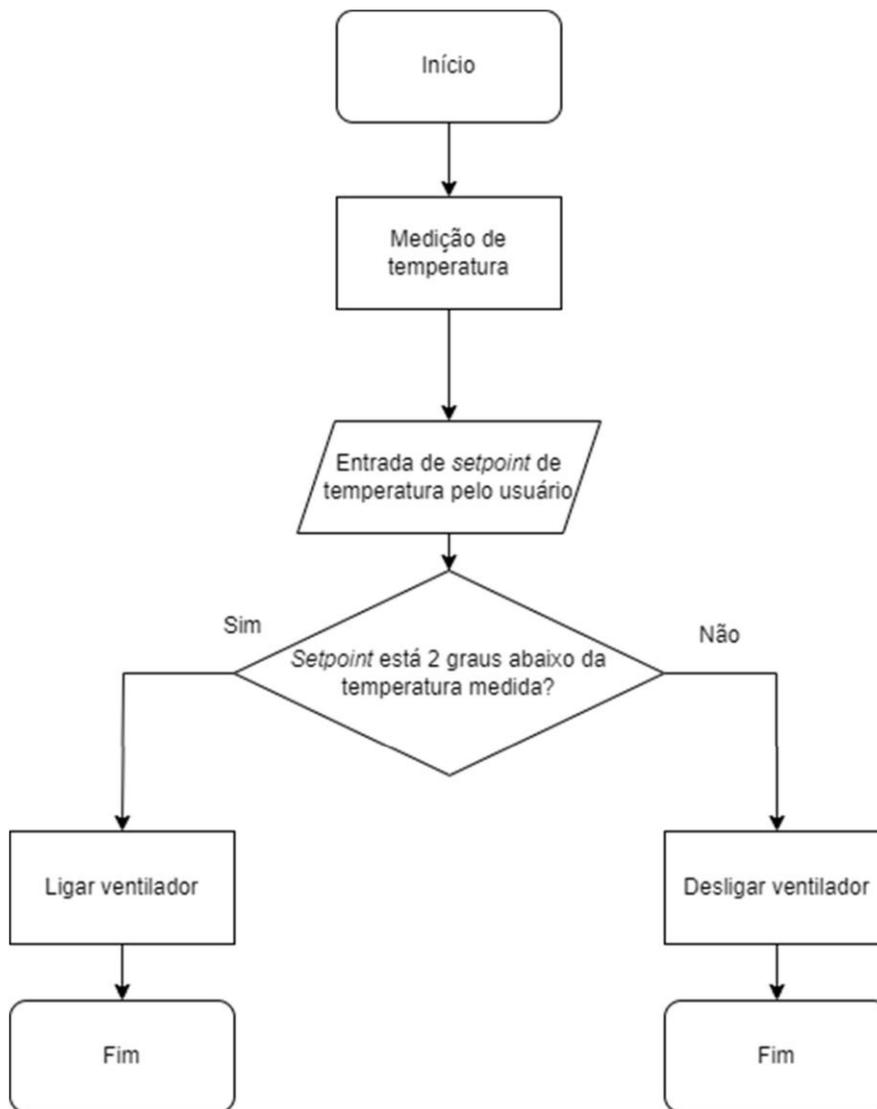
Figura 12: Fluxograma de funcionamento da lâmpada.



Fonte: Autoria própria.

Na Figura 13 é mostrado o funcionamento do ventilador.

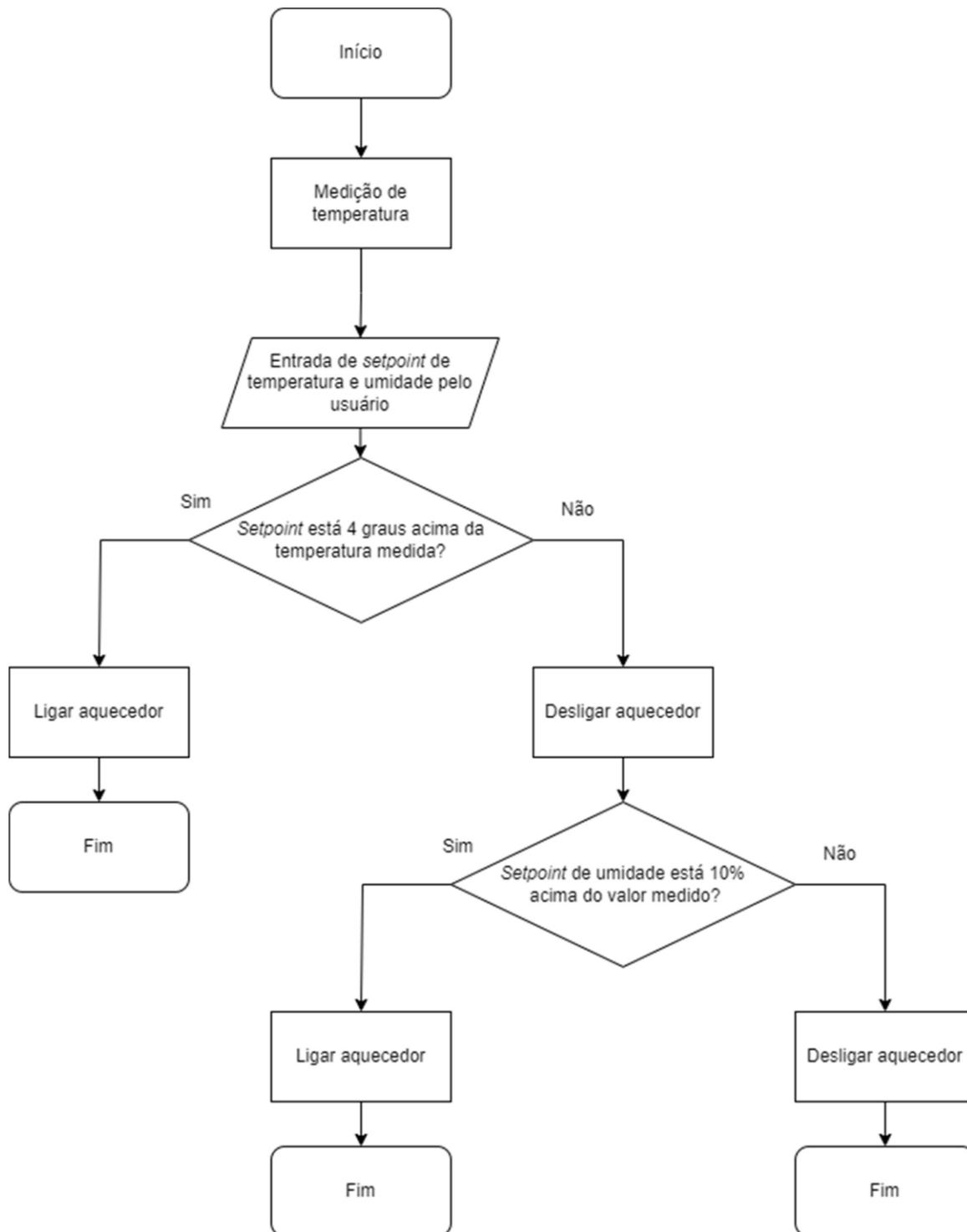
Figura 13: Fluxograma de funcionamento do ventilador.



Fonte: Autoria própria.

Na Figura 14 é mostrado o funcionamento do aquecedor.

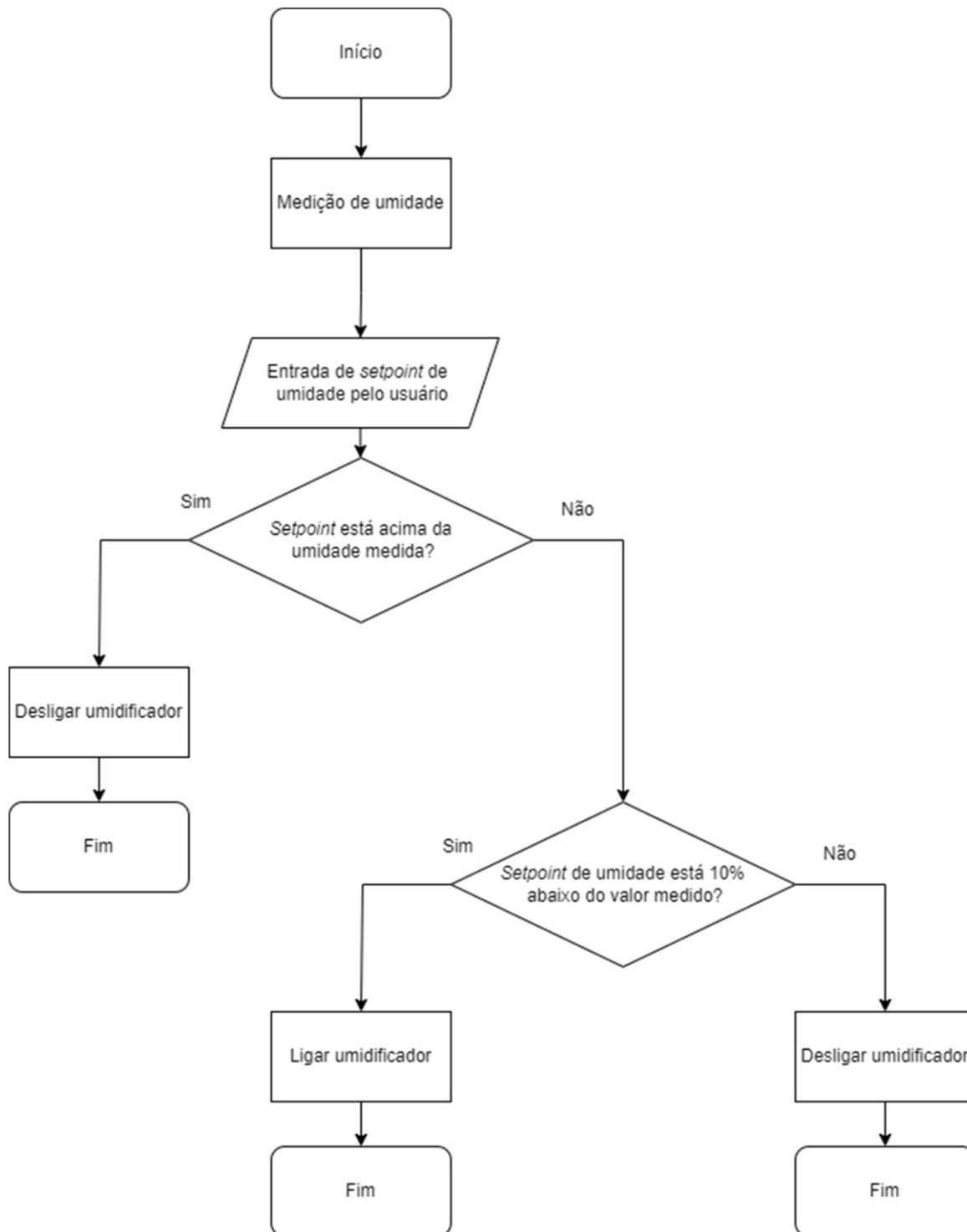
Figura 14: Fluxograma de funcionamento do aquecedor.



Fonte: Autoria própria.

Na Figura 15 é mostrado o funcionamento do umidificador.

Figura 15: Fluxograma de funcionamento do umidificador.



Fonte: Autoria própria.

3.4 Componentes de hardware.

Para medir a temperatura e umidade foi utilizado o sensor DHT11. Na Tabela 2 são mostradas algumas características do sensor obtidas do *datasheet* [9] do componente.

Tabela 2 – Datasheet DHT11.

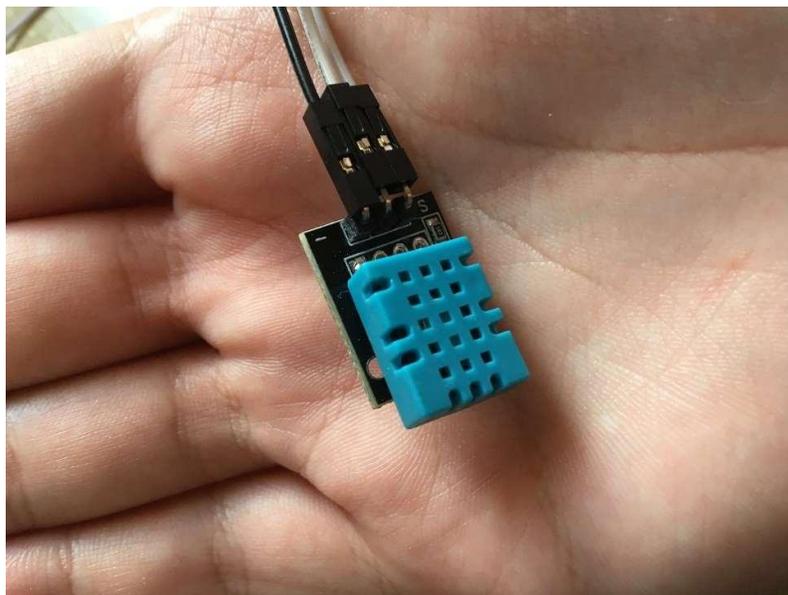
Model	DHT11	
Power supply	3.3-5.5V DC	
Output signal	digital signal	
Sensing element	Polymer humidity sensor	
Operating range	humidity 20-90% RH	temperature 0~50 Celsius
Accuracy	humidity +-5%RH	temperature +-2 Celsius
Resolution or sensitive	humidity 1%RH	temperature 1 Celsius
Repeatability	humidity +-2%RH	temperature +-1 Celsius
Humidity hysteresis	+1RH%	
Long-Term stability	+1RH/year	
Interchangeability	fully interchangeable	

Fonte: Datasheet DHT da Aosong.

A ligação eletrônica desse sensor é feita através de 3 pinos, sendo um pino o positivo da alimentação, ligado a linha de 3,3V do Raspberry Pi, um pino de sinal e um pino negativo que é ligado ao pino terra do Raspberry. O sensor consegue mandar uma resposta a cada 2 segundos aproximadamente.

Na Figura 16 é mostrado o sensor.

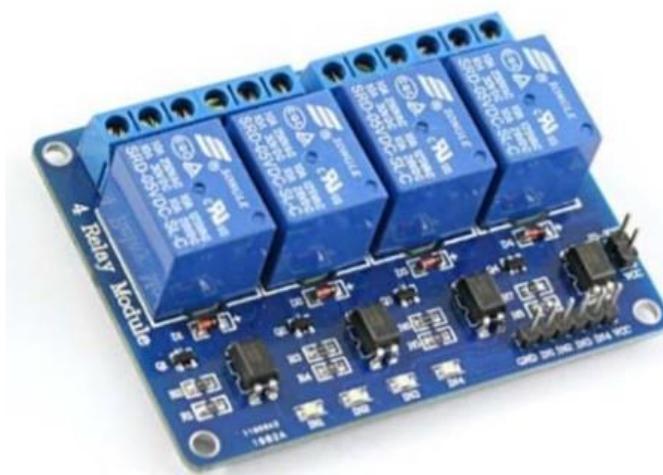
Figura 16: Foto do DHT11.



Fonte: Autorial Própria.

Para o acionamento dos equipamentos foram utilizados módulos relés. Os módulos relés são mostrados na Figura 17.

Figura 17: Módulo relé.



Fonte: Datasheet[10].

Por último, para realizar a função de campanha para os cuidadores foram utilizados 3

buzzers ativos simples de 5V DC. Na Tabela 3 são mostradas algumas características desses buzzers.

Tabela 3 – Especificações dos buzzers.

SPECIFICATIONS:		
Type	Unity	LTE12-05
Rated Voltage	V	5
Operating Voltage	V	4~8
*Rated Current(MAX)	mA	30
*Min Sound Output at 10cm	dB	85
*Resonant Frequency	Hz	2300±300
Operating Temperature	°C	-20 ~ +70
Storage temperature	°C	-30 ~ +105

Fonte: Datasheet[11].

Para acionar os buzzers foram utilizados transístores 2N2222 como chave, conectando a base dos transístores aos GPIO do Raspberry Pi para efetuar o chaveamento. Além disso, foram conectados resistores de 1,8k ohms entre a base do transistor e os buzzers para limitar a corrente de base do transistor.

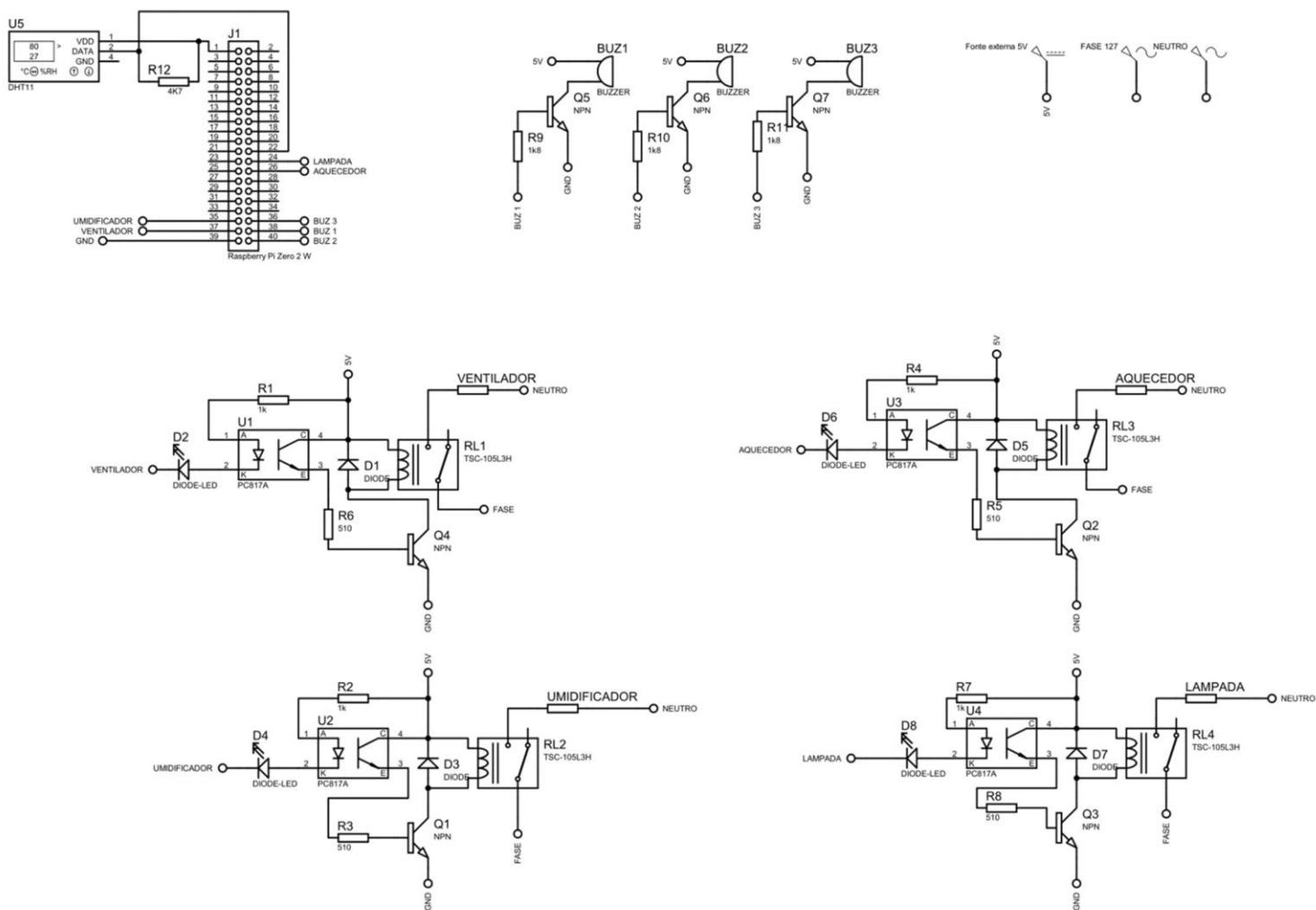
Também foi utilizada uma fonte de 5V DC e 3A para alimentação dos relés e buzzers.

Na Figura 18 é mostrado o diagrama elétrico do projeto. Onde o Raspberry Pi está representado no barramento de 40 pinos indicando a posição de seus GPIOs.

Quando há falta de energia apenas no Raspberry todos os GPIOs perdem sua energia, o que faz com que os relés que ativam os dispositivos desliguem, fazendo que esses também desliguem. Evitando com que haja ligamentos indevidos dos dispositivos.

Caso o sensor de temperatura falhe o código do sistema em Python não terá um valor válido para fazer as comparações entre valores desejados e valores medidos, e como os estados naturais dos dispositivos são desligados, todos os aparelhos são desligados evitando problemas de acionamentos indevidos.

Figura 18: Diagrama elétrico.



Fonte: Autoria própria.

3.5 Custo do sistema

Na Tabela 4 mostrado a relação de preço dos componentes. O dólar no dia de referência (16/05/2022) dos preços estava cotado em R\$ 5,06.

Tabela 4 – Preços dos componentes.

Componente:	Quantidade	Preço
Raspberry Pi Zero 2W	1	US \$ 17,25 [12]
DHT 11	1	US \$ 3,63 [13]
Buzzer	3	US \$ 2,23 [14]
Resistor 1,8kΩ	3	US \$ 0,03
Transistor 2N222	3	US \$ 0,20 [15]
Módulo relé 4 canais	1	US \$ 4,30 [16]
Fonte 5V 3A	2	US \$ 5,03 [17]
Jumpers	30	US \$ 3,87 [18]
Total:		US \$ 46,49

Fonte: Autoria Própria.

O custo do sistema está dentro do esperado, já que com o mesmo, é possível fazer o controle inteligente de quatro dispositivos e três campainhas.

As alternativas no mercado para o sistema são tomadas inteligentes, porém, essas possuem o custo em aproximadamente U\$ 24,70[19] cada. Esse valor multiplicado por quatro, para controlar os mesmos quatro dispositivos que esse projeto, ultrapassaria o valor do sistema em U\$ 52,31. Já campainhas sem fio custam U\$ 7,55 [20]. Como o sistema controla três campainhas, isso geraria um acréscimo de U\$ 22,65 para ter essa função.

Um sensor de temperatura e umidade vendido no mercado custa U\$ 5,90[21].

Para obter as mesmas funções com dispositivos encontrados no mercado se teria um valor de U\$ 127,35. Este valor é aproximadamente o triplo do valor gasto para construir todo o sistema desse projeto.

4 RESULTADOS E DISCUSSÕES

O projeto traz a facilidade de todos os dispositivos serem controlados por uma mesma interface gráfica.

É compatível com qualquer dispositivo que acesse a internet com navegador que possua suporte à HTML 5 e CSS 3. O sistema foi testado conectando os seguintes aparelhos: Smart TV, Ipad 2, Iphone 6S, Iphone 8, e realizando o acionamento dos dispositivos controlados pelo Raspberry Pi.

A resposta do sensor de temperatura, por limitação do próprio sensor, é de aproximadamente 2 segundos e, como visto no datasheet, sua resolução e precisão são suficientes para a aplicação

Outro fator importante é de não exceder a capacidade de corrente da fonte externa e do Raspberry Pi. Na Tabela 5 é possível ver como ficaram essas medições de corrente utilizadas pelos componentes, do Raspberry e da fonte externa.

Tabela 5 – Medições de corrente do sistema.

Correntes	Raspberry	Fonte Ex
Relé 1	4,6 mA	70,7 mA
Relé 2	4,6 mA	70,8 mA
Relé 3	4,6 mA	70,7 mA
Relé 4	4,6 mA	70,4 mA
Buzzer 1	1,38 mA	7,08 mA
Buzzer 2	1,33 mA	7,1 mA
Buzzer 3	1,33 mA	6,9 mA
DHT	GPIO	3.3v Raspberry
	0,08 mA	1,71 mA
	Total	
3.3v	24,23 mA	
Fonte	303,68 mA	

Fonte: Aatoria Própria.

Considerando que o projeto utiliza uma fonte de 3A, ela pode alimentar todos os periféricos mostrados no diagrama elétrico, e até permitir adicionar novos dispositivos.

A potência total máxima, considerando Raspberry Pi, sensor de temperatura, sensor de umidade, módulos relés e buzzers foi de 2,2W. Esse resultado pode ser considerado baixo e de

pouco impacto na conta de luz.

A máxima potência do dispositivo que o relé consegue controlar é 1.250W para 125V de tensão alternada e 1.680W para 240VAC de tensão alternada, como indicado no corpo do relé.

Essa potência é suficiente para ligar uma grande gama de dispositivos e pode ser aumentada caso haja necessidade trocando por um módulo relé que possa controlar cargas maiores.

O tempo de resposta entre acionar um botão na interface gráfica e o Raspberry Pi fazer o acionamento de um objeto é de aproximadamente 1 segundo. Esse teste foi feito cronometrando o tempo entre o acionamento do botão referente a lâmpada da interface gráfica até a lâmpada ligar. Um resultado responsivo e rápido.

O modo automático é capaz de alterar e controlar a umidade e temperatura do ambiente de acordo com os valores escolhidos pelo usuário. Porém existem limitações quanto ao resfriamento que o sistema é capaz de realizar, por utilizar um ventilador.

Como o Raspberry funciona como servidor local, o acesso à interface e ao controle só podem ser feitos quando o dispositivo que for controlar os objetos estiver conectado na mesma rede Wi-Fi que o Raspberry. Assim pessoas de fora não podem acessar os conteúdos armazenados no Raspberry Pi e fazer o controle de objetos de forma indevida.

5 CONCLUSÕES

Como foi visto, foi possível atingir um preço acessível e de maneira que possa ser modular. O controle pode ser feito de maneira simples e possui uma alta compatibilidade por fazer o acionamento elétrico direto, sem se preocupar com protocolos de comunicação.

Além disso, a resposta do sistema é boa, o que faz a navegação da interface e experiência do usuário mais agradável e responsivo por entregar os comandos aos objetos de forma rápida.

Os limites de corrente foram respeitados para essa aplicação em particular, ficando ainda abaixo da metade do recomendado o que abre possibilidades caso haja necessidade de controle de mais objetos.

Como toda interface e foi feita praticamente do zero, é possível alterá-la para atender novas configurações e interfaces, não ficando preso a códigos ou plataformas disponíveis na internet e suas possibilidades pré-definidas.

O controle de temperatura e umidade pode ser feito de maneira boa, porém em situações em que o clima é mais quente, apenas com o ventilador pode ser difícil resfriar o ambiente.

Como mencionado nas sugestões é possível ampliar o projeto para ter uma central

completa fazendo os controles já mencionados, adicionar comandos por voz e até uma câmera de segurança para ter ainda mais controle e apoio ao paciente.

6 TRABALHOS FUTUROS

Para o projeto ter um controle de temperatura melhor, seria interessante fazer a implementação de controle de um ar-condicionado pois este é capaz de oferecer uma refrigeração melhor.

É possível criar históricos de ligamento e desligamento para fazer a programação de horários que esses devem ligar, assim como históricos de temperaturas e umidades registradas.

Como o Raspberry Pi Zero 2 W possui uma entrada USB, é possível também adicionar uma câmera para monitorar por vídeo o estado da pessoa debilitada como uma babá eletrônica e fazer gravações de segurança.

Outra sugestão é adicionar controle por voz, tirando a necessidade de aparelhos externos para realizar o controle.

Como as ferramentas de programação utilizadas no projeto são reprogramáveis, é possível adicionar comandos e rotinas que possam atender melhor um usuário em específico, além de poder fazer o controle de novos dispositivos.

7 REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Gomes, Lucy. “Fatores de risco e medidas profiláticas nas pneumonias adquiridas na comunidade”. *Jornal de Pneumologia*, vol. 27, nº 2, março de 2001, p. 97–114. *DOI.org (Crossref)*. Disponível em: <<https://doi.org/10.1590/S0102-35862001000200008>>. Acesso em: 17 de agosto de 2021.
- [2] SILVA, M. H. Sistema de Automação Residencial baseado em Raspberry Pi e Open ZWave. Disponível em: <https://repositorio.ufsc.br/bitstream/handle/123456789/184698/TCC_MatheusHoffmannSilva.pdf?sequence=-1&isAllowed=y> Acesso em: 11 agosto. 2021.
- [3] VITORATTO. J. G. PROTÓTIPO DE AMBIENTE INTELIGENTE BASEADO EM RASPBERRY PI, WEB E MOBILIDADE. Disponível em: <<https://cepein.femanet.com.br/BDigital/arqTccs/1111330521.pdf>> Acesso em: 11 de agosto. 2021.
- [4] Raspberry Pi Documentation – Raspberry Pi Hardware. Disponível em: <<https://www.raspberrypi.com/documentation/computers/raspberry-pi.html>>.
- [5] Ltd, Raspberry Pi. “Raspberry Pi Zero 2 W”. *Raspberry Pi*. Disponível em: <<https://www.raspberrypi.com/products/raspberry-pi-zero-2-w/>>. Acessado 6 de fevereiro de 2022. Raspberry Pi.
- [6] *HTML: Linguagem de Marcação de Hipertexto | MDN*. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTML>>. Acessado 6 de fevereiro de 2022.
- [7] *CSS | MDN*. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/CSS>>. Acessado 6 de fevereiro de 2022.
- [8] *Python 3.10.2 Documentation*. Disponível em: <<https://docs.python.org/3/>>. Acessado 6 de fevereiro de 2022.
- [9] DHT 11 Datasheet. Disponível em: <https://www.electronicoscaldas.com/datasheet/DHT11_Aosong.pdf> Acesso em: 11 agosto. 2021.
- [10] “5V Four-Channel Relay Module”. *Components101*. Disponível em: <<https://components101.com/switches/5v-four-channel-relay-module-pinout-features-applications-working-datasheet>>. Acessado 6 de fevereiro de 2022.
- [11] Buzzer Datasheet. Disponível em: <<http://www.electronicoscaldas.com/datasheet/LTE12-Series.pdf>> Acesso em: 11 agosto.2021.
- [12] “Raspberry Pi Zero 2 W (Wireless)”. *Cytron Technologies*. Disponível em: <<https://www.cytron.io/p-raspberry-pi-zero-2-w>>. Acessado 6 de fevereiro de 2022.
- [13] “0.88US \$ |Módulo do sensor de temperatura e umidade relativa KY 015 dht11 com 3

- pinos embutidos + pcb kit de iniciantes diy|Circuitos integrados| - AliExpress”. aliexpress.com. Disponível em:
<http://pt.aliexpress.com/item/32956262831.html?src=ibdm_d03p0558e02r02&sk=&aff_platform=&aff_trace_key=&af=&cv=&cn=&dp=>. Acessado 6 de fevereiro de 2022.
- [14] “2.23US \$ 8% de desconto|Campanha alarme ativos ultrafina 5v, alarme ac 9mm * 4.2mm 9x4.2mm 0942 mini campanha piezo ativa apto para arduino diy eletromagnética com 10 peças|Componentes acústicos| - AliExpress”. aliexpress.com. Disponível em:
<http://pt.aliexpress.com/item/32948360045.html?src=ibdm_d03p0558e02r02&sk=&aff_platform=&aff_trace_key=&af=&cv=&cn=&dp=>. Acessado 6 de fevereiro de 2022.
- [15] “Transistor NPN 2N2222”. *Bau Eletrônica*. Disponível em:
<<https://www.baudaeletronica.com.br/transistor-npn-2n2222.html>>. Acessado 6 de fevereiro de 2022.
- [16] “Módulo Relé 5V 4 Canais”. *Arduino Ômega*, <https://www.arduinoomega.com/modulo-rele-5v-4-canais>. Acessado 6 de fevereiro de 2022. Módulo relé, Arduino Omega. Disponível em:
<https://www.arduinoomega.com/modulo-rele-5v-4-canais?utm_source=Site&utm_medium=GoogleMerchant&utm_campaign=GoogleMerchant&gclid=Cj0KCQjw_fiLBhDOARIsAF4khR1gNgwrT2a9I-ZQ_wGsPR2f_JX14SU4pNJjZhnjOQtrl2ZRD4u25-oaAtVbEALw_wcB> Acesso em: 31 outubro. 2021.
- [17] Fontes Chaveadas | MercadoLivre.com.br. Disponível em:
<<https://eletronicos.mercadolivre.com.br/peças-e-componentes-eletricos-fontes-chaveadas/#redirectedFromVip>>. Acessado 6 de fevereiro de 2022.
- [18] Kit Cabo Jumper Wire 20cm 60 Peças P/ Arduino Veja Anuncio - R\$ 21,5. Disponível em:
<https://produto.mercadolivre.com.br/MLB-1020024323-kit-cabo-jumper-wire-20cm-60-peças-p-arduino-veja-anuncio-_JM>. Acessado 6 de fevereiro de 2022.
- [19] Tomada Smart Plug Wi-fi Positivo 10a Casa Inteligente. Disponível em:<https://produto.mercadolivre.com.br/MLB-1923870058-tomada-smart-plug-wi-fi-positivo-10a-casa-inteligente-_JM?matt_tool=56291529&matt_word=&matt_source=google&matt_campaign_id=14303413604&matt_ad_group_id=133074303519&matt_match_type=&matt_network=g&matt_device=c&matt_creative=584156655498&matt_keyword=&matt_ad_position=&matt_ad_type=pla&matt_merchant_id=509501800&matt_product_id=MLB1923870058&matt_product_partition_id=1413191054866&matt_target_id=aud-378637574381:pla-1413191054866&gclid=CjwKCAjwj42UBhAAEiwACIhADpkqLYwe_5OErABiDA_8NIVV6_R3ZCcBDa1jQO3QZbn8ZKrdjbfMxoCFkAQAvD_BwE>. Acesso em: 17 maio. 2022.
- [20] Campanha Residencial Wireless Sem Fio Resistente Água Wifi. Disponível em:
<https://produto.mercadolivre.com.br/MLB-1859313108-campanha-residencial-wireless-sem-fio-resistente-agua-wifi-_JM>. Acesso em: 17 maio. 2022.
- [21] Termo Higrômetro Sensor Externo Medidor Temperatura Umidade. Disponível em:
<<https://produto.mercadolivre.com.br/MLB-2602958206-termo-higrmetro-sensor-externo>>

medidor-temperatura-umidade-_JM>. Acesso em: 17 may. 2022.