

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS  
ESCOLA POLITÉCNICA E DE ARTES  
GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO



**Sistema para controle e monitoramento  
de energia elétrica residencial**

ERICK RIAN BARBOSA DE SOUSA

GOIÂNIA

2023

ERICK RIAN BARBOSA DE SOUSA

**Sistema para controle e monitoramento  
de energia elétrica residencial**

Trabalho de Conclusão de Curso  
apresentado à Escola Politécnica e de  
Artes, da Pontifícia Universidade Católica  
de Goiás, como parte dos requisitos para  
a obtenção do título de Bacharel em  
Engenharia de Computação.

Orientadora:

Profa. Ma. Ludmilla Reis Pinheiro  
dos Santos

Banca examinadora:

Prof. Me. Fernando Gonçalves  
Abadia

Prof. M.e.e. Marcelo Antônio  
Adad de Araújo

GOIÂNIA

2023

ERICK RIAN BARBOSA DE SOUSA

**Sistema para controle e monitoramento  
de energia elétrica residencial**

Trabalho de Conclusão de Curso aprovado em sua forma final pela Escola Politécnica e de Artes, da Pontifícia Universidade Católica de Goiás, para obtenção do título de Bacharel em Engenharia de Computação, em \_\_\_\_/\_\_\_\_/\_\_\_\_.

---

Orientadora: Profa. Ma. Ludmilla Reis Pinheiro  
dos Santos

---

Prof. Me. Fernando Gonçalves Abadia

---

Prof. M.e.e. Marcelo Antônio Adad de Araújo

GOIÂNIA  
2023

A Deus pela minha vida e oportunidades.

Aos meus pais pelo apoio e por  
sempre acreditarem em mim.

## **AGRADECIMENTOS**

Aos meus pais Hérica e José por me ajudarem durante toda a minha vida, por investirem na minha educação, por me incentivarem a enfrentar todas as dificuldades e me apoiarem em tudo.

À Professora Ludmilla, orientadora acadêmica, pelas instruções feitas em cada reunião que contribuíram para o desenvolvimento do trabalho.

“Não é a força, mas a constância dos bons resultados que conduz os homens à felicidade.”

Friedrich Nietzsche

## RESUMO

Neste trabalho foi realizado uma pesquisa em relação ao tema de consumo energético e maneiras de alcançar uma maior eficiência do mesmo, após isso houve um estudo de projetos relacionados com o tema deste trabalho, em relação ao objetivo deles e como concluíram o mesmo. Posteriormente, foi descrito a proposta de solução, que tem como um dos objetivos, desenvolver um protótipo de controle e monitoramento do consumo de energia elétrica em uma residência, sendo que para isso, utilizou-se do sensor ACS712 para medir a corrente elétrica do eletroeletrônico, o sensor ZMPT101B para medir a tensão da rede, e um microcontrolador ESP32 para processar esses dados, e um relé para ligar ou desligar o dispositivo. Quanto a parte do software deste trabalho, foi desenvolvido um aplicativo Android na linguagem de programação Dart, que reuniu as informações obtidas pelo protótipo e as apresentou no aplicativo por meio de um gráfico, uma tabela, um campo referente ao consumo total em kW do aparelho monitorado e outro campo referente ao valor do consumo total em reais. Além disso, os resultados deste trabalho foram sim satisfatórios, visto que tanto a parte do hardware quanto a parte do aplicativo cumpriram com os respectivos objetivos propostos, contudo a parte do hardware apresentou delay tanto na questão de envio, quanto na obtenção de informação por meio da internet.

***Palavras-Chave:*** Consumo elétrico. ESP32. Sensor de corrente. Sensor de tensão. Aplicativo Android.

## ABSTRACT

*In this work, a research was carried out on the topic of energy consumption and ways to achieve greater efficiency in it. After that, a study was conducted on projects related to the theme of this work, regarding their objectives and how they concluded. Subsequently, the proposed solution was described, which aims to develop a prototype for controlling and monitoring the electricity consumption in a house. To achieve this, the ACS712 sensor was used to measure the electrical current of the electronic device, the ZMPT101B sensor to measure the network voltage, and an ESP32 microcontroller to process this data, and a relay to turn the device on or off. As for the software part of this work, an Android application was developed in the Dart programming language, which brought together the information obtained by the prototype and presented it in the application through a graph, a table, a field referring to the total consumption in kW of the device monitored and another field referring to the value of total consumption in reais. Furthermore, the results of this work were indeed satisfactory, since both the hardware part and the application part met their respective proposed objectives, however the hardware part presented delays both in terms of sending and obtaining information through the Internet.*

**Keywords:** *Electricity consumption. ESP32. Current Sensor. Voltage Sensor. Android Application.*

## LISTA DE FIGURAS

Figura 1 - Hardwares do projeto.....	21
Figura 2 - Aplicativo da autora.....	22
Figura 3 - Aplicativo hospitalar.....	24
Figura 4 - Teste de consumo com um desktop.....	25
Figura 5 - Resultado de consumo com desktop.....	26
Figura 6 - Hardware para controle de iluminação.....	27
Figura 7 - Aplicativo para controle de iluminação.....	28
Figura 8 - Hardware e Software do trabalho.....	30
Figura 9 - Diagrama de blocos do sistema proposto.....	32
Figura 10 - Microcontrolador ESP32.....	33
Figura 11 - Sensor de tensão ZMPT101B.....	35
Figura 12 - Sensor de corrente ACS712.....	36
Figura 13 - Circuito interno Relé.....	37
Figura 14 - Módulo Relé.....	37
Figura 15 - Código para ligar/desligar o LED.....	40
Figura 16 - Circuito para ligar/desligar o LED.....	41
Figura 17 - Código para ligar/desligar lâmpada com relé.....	42
Figura 18 - Circuito para ligar/desligar lâmpada com relé.....	42
Figura 19 - Código para medir corrente de um secador de cabelo.....	43
Figura 20 - Circuito para medir corrente de um secador de cabelo.....	44
Figura 21 - Valores medidos pelo sensor de corrente.....	44
Figura 22 - Código para medir tensão de uma tomada.....	45
Figura 23 - Valores medidos pelo sensor de tensão.....	46
Figura 24 - Circuito para medir tensão de uma tomada.....	46
Figura 25 - Código para medir tensão e corrente de um secador de cabelo e ligar uma lâmpada.....	47
Figura 26 - Circuito para medir tensão e corrente de um secador de cabelo e ligar uma lâmpada.....	48
Figura 27 - Valores medidos pelo sensor de tensão e corrente.....	49
Figura 28 - Valores medidos pelo sensor de tensão e corrente do ferro de passar.....	50
Figura 29 - Trecho de código para enviar o dado de ligar o LED para o Firebase.....	52

Figura 30 - Código para ligar e desligar o LED pelo aplicativo usando o Firebase.....	54
Figura 31 - Resultado do: (a) aplicativo Android; e (b) monitor serial.....	56
Figura 32 - Resultado do Firebase .....	56
Figura 33 - Resultado dos comandos de ligar e desligar o LED.....	57
Figura 34 - Tela de Login.....	59
Figura 35 - Entrar com Google .....	60
Figura 36 - Login via e-mail e senha.....	61
Figura 37 - Tela de criar conta.....	62
Figura 38 - Tela esqueci minha senha.....	63
Figura 39 - E-mail de recuperação de senha.....	64
Figura 40 - Tela Principal.....	65
Figura 41 - Aba Tabela.....	67
Figura 42 - Tela Definir Tarifa.....	68
Figura 43 - Filtro por Data e Hora.....	69
Figura 44 - Hardware e aplicativo para controle e monitoramento do consumo de energia elétrica.....	70
Figura 45 - Circuito para medir a tensão e corrente e poder desligar e ligar um secador de cabelo.....	71
Figura 46 - Teste utilizando o secador de cabelo: (a) Gráfico; (b) tabela de pontos.....	72
Figura 47 - Teste aplicando o filtro nos dados do secador: (a) Gráfico; (b) tabela de pontos.....	73
Figura 48 - Teste utilizando o ferro de passar: (a) Gráfico; (b) tabela de pontos.....	74
Figura 49 - Valores de potência calculada de um secador de cabelo por meio dos sensores de corrente e tensão.....	75
Figura 50 - Função para pegar dados para a primeira coluna da tabela.....	76

## LISTA DE TABELAS

Tabela 1 - Características do ESP32.....	34
Tabela 2 - Características do ZMPT101B.....	35
Tabela 3 - Características do ACS712.....	36
Tabela 4 - Características do Relé.....	37

## LISTA DE SIGLAS

A	Ampère
AC	<i>Alternating Current</i>
DC	<i>Direct Current</i>
EPE	Empresa de Pesquisa Energética
IDE	<i>Integrated Development Environment</i>
kWh	Quilowatt-hora
LED	<i>Light Emitting Diode</i>
V	Volt
VA	Volt-Ampère
<i>Wi-fi</i>	<i>Wireless Fidelity</i>
W	Watt

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
1.1	<i>Objetivo geral</i>	16
1.2	<i>Objetivos Específicos</i>	17
1.3	<i>Justificativa</i>	17
1.4	<i>Procedimentos Metodológicos</i>	17
1.5	<i>Organização do trabalho</i>	18
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>19</b>
2.1	<i>Eficiência energética</i>	19
2.2	<i>Trabalhos relacionados</i>	19
2.2.1	<i>Desenvolvimento de um aplicativo Android para monitoramento do consumo de energia elétrica de uma residência</i>	20
2.2.2	<i>Desenvolvimento de software para controle de dispositivos elétricos via Bluetooth e comando de voz</i>	23
2.2.3	<i>Desenvolvimento de um sistema de automação residencial baseado em IoT para controle e monitoramento de dispositivos elétricos</i>	24
2.2.4	<i>Sistema de controle de iluminação e monitoramento de consumo elétrico residencial via aplicativo</i>	26
2.2.5	<i>Sistema de automação residencial por comandos de voz utilizando Raspberry e aplicação móvel</i>	29
2.2.6	<i>Análise dos trabalhos relacionados</i>	30
<b>3</b>	<b>PROPOSTA DE SOLUÇÃO</b>	<b>32</b>
3.1	<i>Diagrama de blocos</i>	32
3.2	<i>Hardware</i>	33
3.2.1	<i>Microcontrolador</i>	33
3.2.2	<i>Sensor de Tensão</i>	34
3.2.3	<i>Sensor de Corrente</i>	36
3.2.4	<i>Relé</i>	37
3.3	<i>Software</i>	38
3.3.1	<i>Banco de dados</i>	38
3.3.2	<i>Linguagem de programação Dart</i>	39
3.3.3	<i>Android Studio</i>	39
3.3.4	<i>Arduino Software IDE</i>	39
<b>4</b>	<b>DESENVOLVIMENTO DO PROTÓTIPO</b>	<b>41</b>
4.1	<i>Testes Unitários</i>	41
4.1.1	<i>ESP32</i>	41
4.1.2	<i>Relé</i>	42
4.1.3	<i>Sensor de corrente ACS712</i>	43

4.1.4	Sensor de tensão ZMPT101B .....	46
4.1.5	Teste de integração .....	47
<b>5</b>	<b>DESENVOLVIMENTO DO APLICATIVO .....</b>	<b>53</b>
5.1	<i>Comunicação entre o aplicativo, o Firebase e a ESP32 .....</i>	<i>53</i>
5.2	<i>Versão final do aplicativo .....</i>	<i>58</i>
5.2.1	<i>Tela de Login .....</i>	<i>59</i>
5.2.2	<i>Entrar com Google .....</i>	<i>60</i>
5.2.3	<i>Entrar com E-mail .....</i>	<i>61</i>
5.2.3.1	<i>Opção de Criar Conta .....</i>	<i>62</i>
5.2.3.2	<i>Opção de Esqueci minha Senha .....</i>	<i>63</i>
5.3	<i>Tela Principal .....</i>	<i>65</i>
5.4	<i>Aba Tabela .....</i>	<i>67</i>
<b>6</b>	<b>RESULTADOS OBTIDOS .....</b>	<b>71</b>
6.1	<i>Análise dos resultados obtidos .....</i>	<i>71</i>
6.2	<i>Considerações sobre os resultados obtidos .....</i>	<i>78</i>
<b>7</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>79</b>
7.1	<i>Sugestões para trabalhos futuros .....</i>	<i>80</i>
<b>8</b>	<b>REFERÊNCIAS .....</b>	<b>82</b>
<b>APÊNDICE A – CÓDIGO DA ESP32 .....</b>		<b>86</b>
<b>APÊNDICE B – TERMO DE AUTORIZAÇÃO DE PUBLICAÇÃO DE PRODUÇÃO ACADÊMICA .....</b>		<b>89</b>

## 1 INTRODUÇÃO

De acordo com o Anuário Estatístico de Energia Elétrica de 2022 da Empresa de Pesquisa Energética (EPE), o Brasil é um dos 10 países que mais consome energia no mundo, e esse consumo tem aumentado ao longo dos anos, sendo que de 2010 a 2019 houve um aumento de 18,70% no consumo de energia elétrica (EPE, 2022).

Além disso, vale ressaltar que conforme o Balanço Energético Nacional de 2022 da EPE, uma das principais fontes de energia no Brasil, a fonte hídrica, corresponde a 53,4% da oferta interna de energia no país, sendo que as fontes não renováveis representam 22,6% do total nacional de geração elétrica e as demais fontes renováveis representam 24% desse total (EPE, 2022).

Em 2021, o Brasil passou pela pior crise hídrica em mais de 90 anos, que consequentemente resultou em uma crise energética e por consequência impactou em um aumento na conta de energia, devido a criação da tarifa de escassez hídrica (CNN BRASIL, 2021).

Maneiras que buscam reduzir o desperdício de energia, como políticas de sensibilização ou de estímulos a fim de substituir eletrodomésticos antigos por equipamentos que consomem menos energia, podem diminuir o gasto monetário com a mesma, sendo que isso não implica necessariamente na redução do nível de bem-estar de uma família (COSTA et al., 2017).

Para auxiliar na identificação de desperdício de energia em um eletrodoméstico, é possível medir o consumo energético desse aparelho a partir de um único ponto de medição, como a entrada de energia, e assim por meio da utilização de dados de um medidor inteligente, permitir monitorar com precisão o consumo de energia (RASHID et al., 2019).

De acordo com Ruano et al. (2019), por meio do uso contínuo de medidores inteligentes, diferentes dispositivos comerciais, e com base em uma única medida de corrente e tensão, é possível obter dados quanto ao gasto energético de cada aparelho utilizado em residências ou edifícios.

Portanto, diferentes tipos de tecnologia podem ser utilizadas para medir parâmetros de energia, sendo que tecnologias de medição inteligente são

apropriadas para checagem da qualidade da energia, medir a potência ativa e reativa da mesma, otimização do controle da rede e monitoramento do consumo de energia (MASSARO, 2022).

Devido ao avanço de certas tecnologias, como sensores, comunicação mobile, e aplicação das mesmas em uma residência, tem se como resultado uma automação em casa que proporciona tanto conforto e segurança, quanto a redução do consumo de energia (KYAS, 2017).

A utilização de aparelhos móveis como celulares para controle de dispositivos elétricos, é capaz de proporcionar a economia de energia em um ambiente, pois ao acessar equipamentos de forma remota, há por consequência, uma redução no consumo de energia, devido ao acionamento dos dispositivos ser feito somente quando for necessário (HENRIQUE, 2016).

De acordo com Chung (2002) a eficiência energética pode ser alcançada por meio de diversas técnicas, como a utilização de softwares que buscam reduzir custos relacionados ao consumo de energia. Além disso, a adoção desses softwares podem contribuir para a redução do impacto ambiental da produção e uso da energia utilizada pela população.

Conforme Vandome (2018), o conceito de uma casa cujos dispositivos eletrônicos como lâmpadas e tomadas, possam ser controlados ou acessados remotamente por aplicações web ou mobile por exemplo, é algo completamente possível, realista e acessível.

## **1.1 Objetivo geral**

Este trabalho objetiva monitorar o consumo de energia elétrica de aparelhos eletroeletrônicos de uma residência, medindo os valores de corrente e tensão dos aparelhos, por meio de sensores. Além disso, permitir que o usuário seja capaz de ligar ou desligar os dispositivos de sua residência por meio de uma aplicação *mobile* que manda um sinal para um atuador, permitindo realizar esse controle.

## **1.2 Objetivos Específicos**

- Obter valores de corrente e tensão de aparelhos eletroeletrônicos a serem recebidos por meio de sensores e de um microcontrolador;
- Apresentar o consumo mensal de energia de um aparelho eletroeletrônico monitorado ou de um determinado período indicado;
- Permitir que o usuário seja capaz de ligar ou desligar um aparelho conectado à tomada por meio de um aplicativo mobile;
- Desenvolver um aplicativo Android que mostre ao usuário informações de consumo de energia de um aparelho na residência, e que permita ligar e desligar o mesmo.

## **1.3 Justificativa**

Estudar este tema é crucial, pois, conforme indicado pela EPE, observa-se um crescimento constante nos gastos e nos custos de energia no Brasil, juntamente atrelado a questão da crise hídrica em 2021, considerada a maior em mais de 90 anos no país.

Nesse mesmo contexto, este trabalho permitiu o controle e o monitoramento dos gastos de energia em uma residência por meio da automação proveniente de um hardware e de um aplicativo Android. Com isso, busca-se não apenas reduzir os gastos com energia, mas também possibilitar uma utilização mais sustentável das fontes de energia do país, de forma a considerar a questão do impacto ambiental da produção e uso da energia.

## **1.4 Procedimentos Metodológicos**

O primeiro passo desse trabalho foi a realização de uma pesquisa bibliográfica, sendo que ela se caracterizou por não produzir conhecimento novo, mas sim suprir o pesquisador de informações públicas que ainda não possuía (WAZLAWICK, 2014).

Já o segundo passo desse trabalho foi a realização de uma pesquisa experimental, em que o pesquisador provocou mudanças no ambiente de pesquisa, observando se as alterações realizadas são de acordo com os resultados esperados (WAZLAWICK, 2014).

### **1.5 Organização do trabalho**

O capítulo 1 apresenta o referencial teórico deste trabalho, junto aos objetivos gerais, objetivos específicos, a justificativa do trabalho, os procedimentos metodológicos e quanto a organização do trabalho.

O capítulo 2 aborda a busca pela eficiência energética, discutindo suas implicações e apresentando métodos práticos para sua obtenção. Além disso, apresenta trabalhos com semelhanças ao tema deste trabalho.

Já o capítulo 3 descreve como foi solucionado a questão do tema deste trabalho.

O capítulo 4 descreve os testes realizados para a geração de um hardware que atendesse a alguns dos objetivos deste trabalho.

O capítulo 5 descreve os testes realizados para a geração de um aplicativo que atendesse aos demais objetivos deste trabalho e descreve como ficou a versão final do aplicativo Android.

Já o capítulo 6, descreve os resultados obtidos com o protótipo e aplicativo utilizados neste trabalho.

Sobre o capítulo 7, este retoma as questões envolvendo este trabalho, como os seus objetivos e a avaliação da eficácia do hardware e do aplicativo utilizados para atingir esses objetivos. No final deste capítulo foram apresentadas sugestões para trabalhos futuros.

## 2 REFERENCIAL TEÓRICO

Este capítulo descreve o que a busca pela eficiência energética proporciona, assim como maneiras de obtê-la, e apresenta trabalhos relacionados com o tema deste trabalho.

### 2.1 Eficiência energética

De acordo com Rio (2019), há uma busca por eficiência energética através da utilização de dispositivos mais modernos como pelo uso de lâmpadas de LED e aparelhos de ar-condicionado, sendo que essa atitude visa a redução de custos operacionais em uma residência, por exemplo. Para ele, uma maior eficiência energética significa a realização da mesma tarefa, porém com a redução do uso de energia elétrica. A Equação (1) apresenta o cálculo de consumo de energia, sendo que C = consumo de energia (kWh), P = potência (watt), e T = tempo (horas).

$$C = \frac{P}{1000} * T \quad (1)$$

Por meio da utilização de sensores para desligar eletroeletrônicos de forma autônoma, a variável Tempo da Equação (1), pode ter um menor valor, o que possibilita uma diminuição no consumo da energia, ou seja, se 10% de energia for menos utilizada pela população, serão menos 10% de energia a serem produzidos pela matriz energética do país para sustentá-los (RIO, 2019).

### 2.2 Trabalhos relacionados

Esta seção apresenta trabalhos relacionados ao tema de pesquisa e uma análise destes trabalhos.

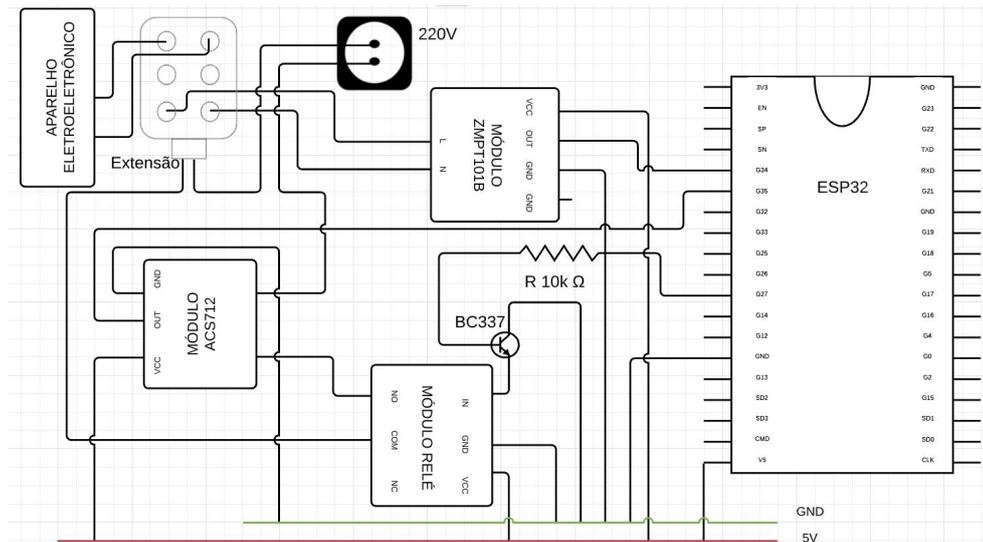
### *2.2.1 Desenvolvimento de um aplicativo Android para monitoramento do consumo de energia elétrica de uma residência*

Este trabalho teve como objetivo monitorar o consumo de energia elétrica de dispositivos eletroeletrônicos em uma casa, medindo tanto a corrente quanto a tensão. Os dados obtidos através dos medidores foram apresentados em um gráfico e tabela, permitindo que o consumidor visualizasse seus gastos e supervisionasse o consumo de energia para possibilitar que o mesmo possa melhorar a eficiência energética da residência (PIRES, 2021).

Para isso, foi utilizado um microcontrolador ESP32 que por meio de um programa em Arduino, processou as informações referentes aos valores de tensão e corrente obtidos através do sensor de corrente ACS712 e pelo sensor de tensão ZMPT101B, de cada aparelho eletroeletrônico. Além disso, para permitir o ligar e desligar desses aparelhos, utilizou-se um módulo relé, sendo que tanto esta função quanto as informações coletadas, foram reunidas no aplicativo Android (PIRES, 2021).

Assim, ao terminar de desenvolver a parte referente ao hardware do projeto, como mostra na Figura 1, o qual exibe que o aparelho eletroeletrônico foi conectado a uma extensão, sendo que a mesma foi ligada a uma tomada de 220 volts, ao módulo de corrente e tensão, ao relé e finalmente ao microcontrolador ESP32 (PIRES, 2021).

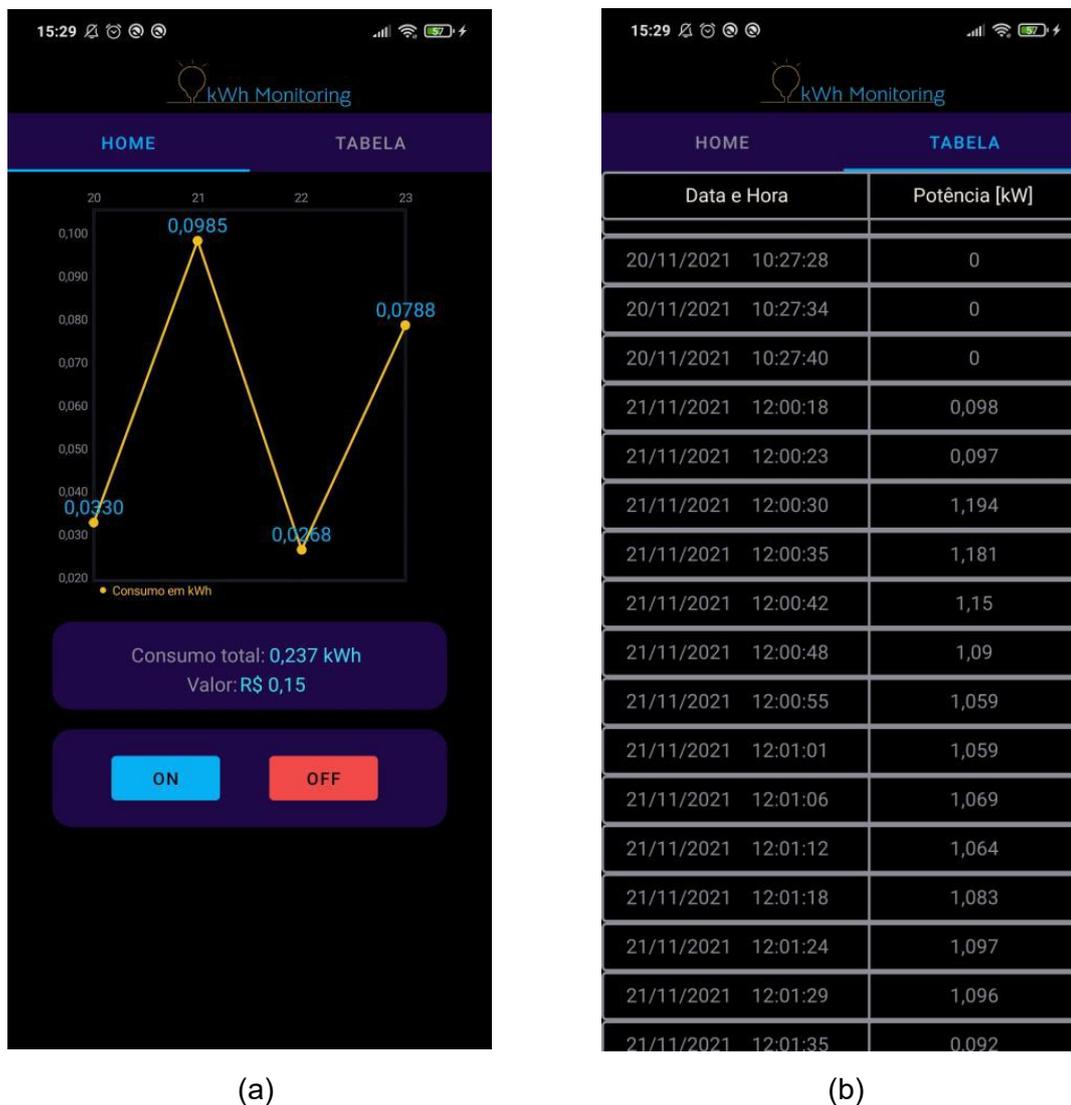
Figura 1 - Hardwares do projeto.



Fonte: Pires, 2021.

As imagens do software apresentados na Figura 2 exibem o consumo em Kw do dispositivo que está sendo monitorado, sendo que na Figura 2 (a) é exibido qual foi o consumo do dia 20 ao 23 de novembro de 2021, em que totalizou um consumo de energia de 0,237 kWh, cujo valor total em reais foi de R\$ 0,15, e sendo possível ver também os botões de ligar e desligar do dispositivo. Já na Figura 2 (b) é exibida uma tabela com as datas e horários do dia 20 ao 21 de novembro de 2021, e seus respectivos consumos em kW, sendo que no dia 20 o aparelho estava desligado (PIRES, 2021).

Figura 2 - Aplicativo da autora.



Fonte: Pires, 2021.

Neste trabalho o armazenamento dos valores obtidos foi feito no banco de dados em nuvem, e devido a velocidade da internet e do tempo que o ESP32 gastou para enviar e gravar os valores lidos, foi observado um atraso na gravação dos valores no banco de dados. Isso fez com que as medições de tensão e corrente sofressem interferência e não fosse calculado os valores exatos de consumo. Contudo, o trabalho apresentou resultados satisfatórios ao cumprir com o objetivo de realizar medições referentes a energia elétrica, de forma que os valores de consumo de kWh por dia e o valor total em reais gasto no mês foram apresentados no aplicativo (PIRES, 2021).

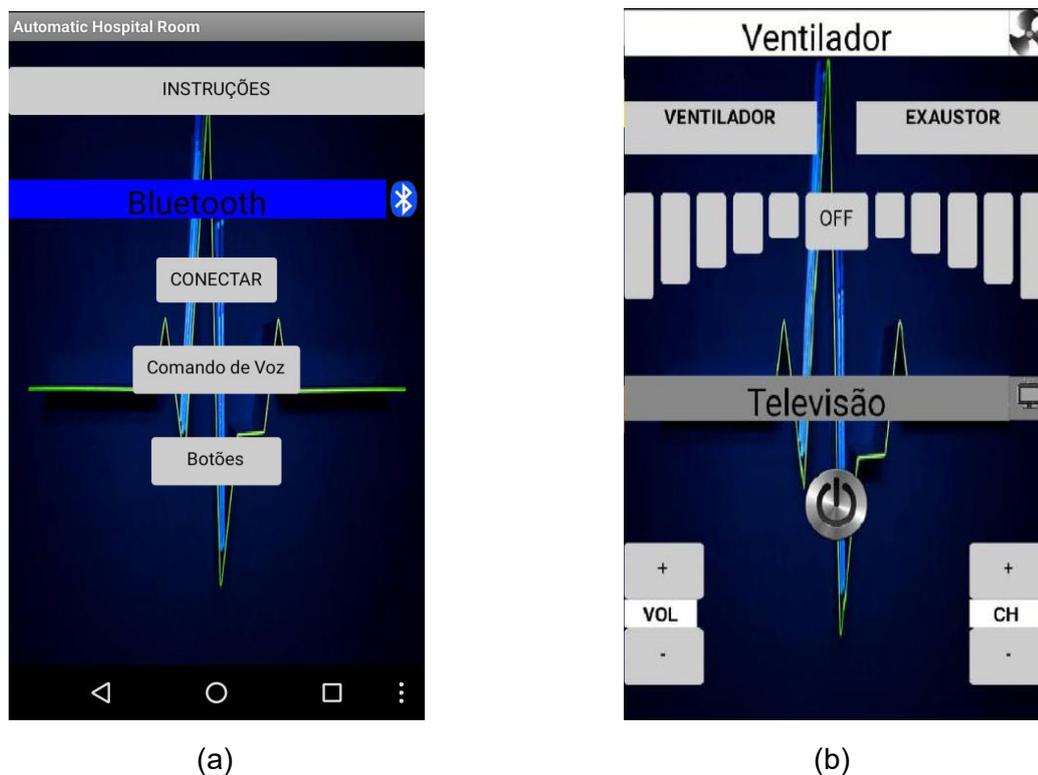
### 2.2.2 *Desenvolvimento de software para controle de dispositivos elétricos via Bluetooth e comando de voz*

O trabalho de desenvolvimento de software para controle de dispositivos elétricos via Bluetooth e comando de voz teve como foco a sua aplicação em um quarto hospitalar, e como objetivo, executar certas tarefas, como acionar uma lâmpada e ventilador sem precisar se levantar da cama, somente através de botões presentes no software ou de comandos de voz. O aplicativo foi criado para celulares e tablets Android, sendo que o software se comunicava com o hardware desenvolvido, por meio da tecnologia Bluetooth (HENRIQUE, 2016).

Para isso, foi utilizado uma placa Arduino UNO, que através do módulo HC-06, recebeu sinais do aplicativo para acionar os dispositivos do quarto como: campainha, lâmpada (utilizava um relé para ligar e desligar) e televisão. Vale dizer que o principal componente na placa Arduino foi o microcontrolador ATMEGA328 responsável por executar o comando de acionar os dispositivos. Já para garantir a comunicação do Arduino com o computador para inserção dos códigos (linguagem Arduino), foi usado o microcontrolador ATMEGA16U2 (HENRIQUE, 2016).

Na Figura 3 é exibido o aplicativo final, o qual na Figura 3 (a) é exibido 4 botões na tela, o de instruções (mostrava as frases reconhecidas pelo comando de voz), o de conectar (conectava o celular ou tablet com o módulo *Bluetooth*), o comando de voz (acionava uma tela para usuário requisitar um comando), e botões que mostrava as telas secundárias do aplicativo, como na Figura 3 (b), que exibia botões para acionar o ventilador e televisão (HENRIQUE, 2016).

Figura 3 - Aplicativo hospitalar.



Fonte: Henrique, 2016.

O resultado deste trabalho foi satisfatório, visto que cumpriu o objetivo de fazer com que a necessidade de auxílio de pequenas tarefas para o paciente em um quarto de hospital fosse minimizado, por conta da automatização dessas tarefas e devido a utilização da tecnologia *Bluetooth* para controle dos dispositivos elétricos que se mostrou eficaz (HENRIQUE, 2016).

### 2.2.3 Desenvolvimento de um sistema de automação residencial baseado em IoT para controle e monitoramento de dispositivos elétricos

O trabalho Desenvolvimento de um sistema de automação residencial baseado em IoT para controle e monitoramento de dispositivos elétricos, teve como objetivo o desenvolvimento de um protótipo de sistema de automação residencial, capaz de medir a corrente e a tensão de um dispositivo conectado à rede elétrica alternada. De forma que o usuário pôde visualizar esses dados por meio do site *ThingSpeak* (OLIVEIRA, 2019).

Para isso, foi utilizado o microcontrolador ESP32 que através de um programa em C, obteve os valores de corrente e tensão do dispositivo elétrico monitorado, e por meio do sensor de corrente ACS712 e de um circuito próprio para medir a tensão AC, o microcontrolador reuniu essas informações e às transmitiu para a plataforma *ThingSpeak*, por meio de sua conexão *Wi-Fi* (OLIVEIRA, 2019).

Foi realizado um teste para medir o consumo de energia de um *desktop* conectado aos sensores, e os resultados do mesmo podem ser vistos na Figura 4, sendo exibido quando o computador ficou em: modo de espera, modo ligado e modo desligado. Após isso, foi realizado o cálculo da energia consumida pelo *desktop*, o qual em 7 horas e meia consumiu o total de energia de 0,362 kWh, como pode ser visto na Figura 5 (OLIVEIRA, 2019).

Figura 4 - Teste de consumo com um desktop.



Fonte: Oliveira, 2019.

Figura 5 - Resultado de consumo com desktop.

Dados do teste realizado com um Desktop			
Data Inicial	Data Final	Hora Inicial	Hora Final
14/05/2019	15/05/2019	22:42:39	06:14:53
Tempo Total	07:32:14		
Tensão Média	126,758	V	
Corrente Média	0,366	A	
Potência Média	46,420	W	
Energia	0,362	KWh	

Fonte: Oliveira, 2019.

Quanto ao sensor de tensão utilizado no trabalho, ele apresentou resultados satisfatórios, já o sensor de corrente apresentou ruídos quando não havia corrente, sendo necessário a não medição quando o eletroeletrônico estava desligado, contudo sua medição era satisfatória quando havia corrente. Sobre o site *ThingSpeak*, o mesmo proporcionou segurança em seu acesso, e se mostrou uma ótima ferramenta gratuita para analisar os dados obtidos dos sensores (OLIVEIRA, 2019).

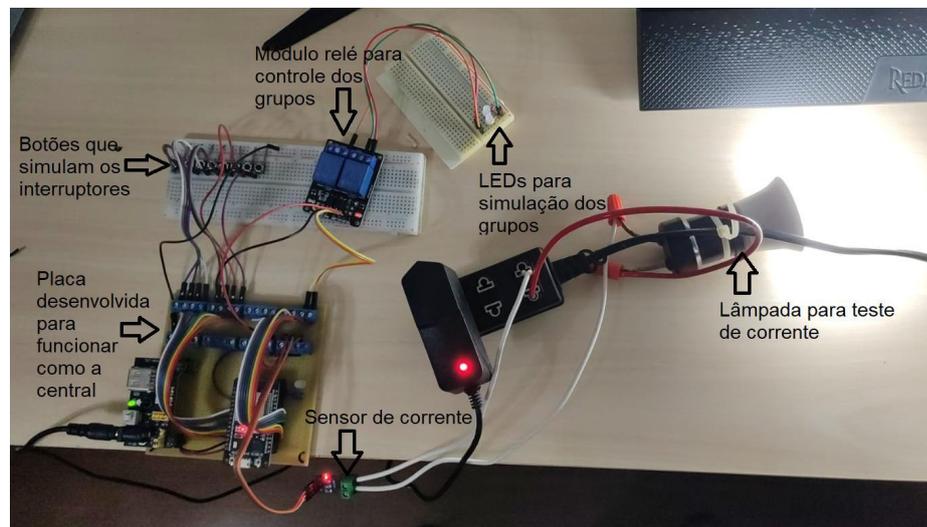
#### 2.2.4 Sistema de controle de iluminação e monitoramento de consumo elétrico residencial via aplicativo

O trabalho Sistema de controle de iluminação e monitoramento de consumo elétrico via aplicativo, teve como objetivo o desenvolvimento de uma central de controle de iluminação para diferentes espaços em uma casa, juntamente com o monitoramento em tempo real do consumo de energia das lâmpadas da residência, sendo que essas funções foram integradas a um aplicativo móvel, feito no site *Thunkable* (MERÇON, 2022).

O aplicativo foi a plataforma utilizada pelo usuário para monitorar sua residência e programar rotinas para ligar ou desligar as luzes em ambientes específicos. Para isso foi utilizado o Arduino Uno R3, junto ao módulo *Wi-Fi ESP32 Bluetooth*, que funcionaram como uma central para obter os valores de corrente consumidos pelas lâmpadas, proveniente do sensor ACS712, e sendo

que o ligar e desligar das lâmpadas foi feito usando um relé, conforme Figura 6 (MERÇON, 2022).

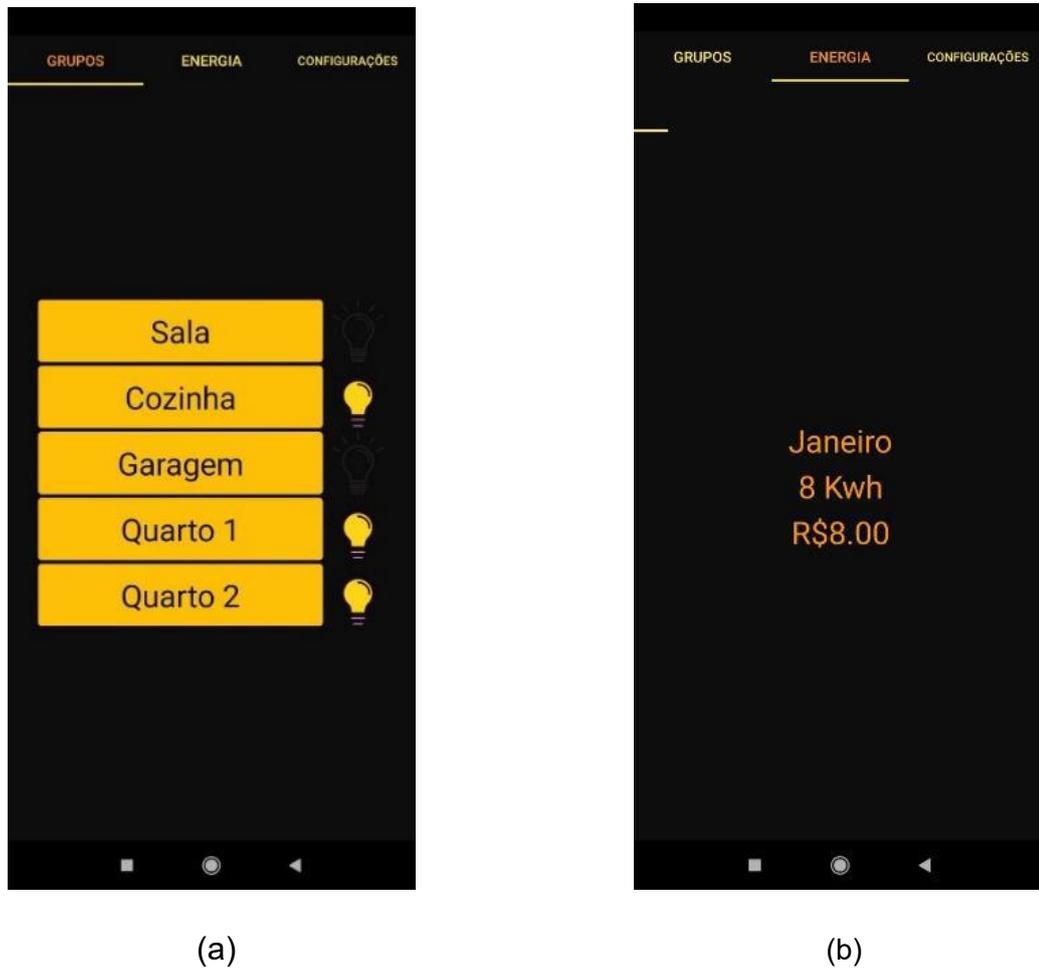
Figura 6 - Hardware para controle de iluminação.



Fonte: Merçon, 2022.

A Figura 7 (a) exibe a aba Grupos do aplicativo, o qual o usuário conseguia controlar a iluminação de cada cômodo da casa, ligando ou desligando as lâmpadas e podendo conferir o estado das luzes a partir dos ícones ao lado do nome dos grupos. Já na Figura 7 (b) na aba Energia é exibido o gasto de energia em Kwh e em reais do mês (MERÇON, 2022).

Figura 7 - Aplicativo para controle de iluminação.



Fonte: Merçon, 2022.

Este trabalho foi executado com sucesso, visto que o sistema desenvolvido permitiu ao usuário monitorar o consumo energético de sua residência e controlar as lâmpadas da residência de qualquer lugar, desde que seu dispositivo móvel estivesse conectado à internet. Além disso, o custo para construção do protótipo foi de R\$ 260,60 reais sendo que outras soluções comerciais com o mesmo propósito requeriam um investimento de pelo menos R\$ 722,90 reais (MERÇON, 2022).

### *2.2.5 Sistema de automação residencial por comandos de voz utilizando Raspberry e aplicação móvel*

O trabalho Sistema de automação residencial por comandos de voz utilizando Raspberry e aplicação móvel, teve como objetivo principal a criação de um sistema de automação residencial que podia controlar aparelhos eletroeletrônicos usando um sistema inteligente, sendo que o sistema utilizou conceitos da computação e da automação para realizar testes comportamentais em um protótipo de uma casa (LIMA, 2018).

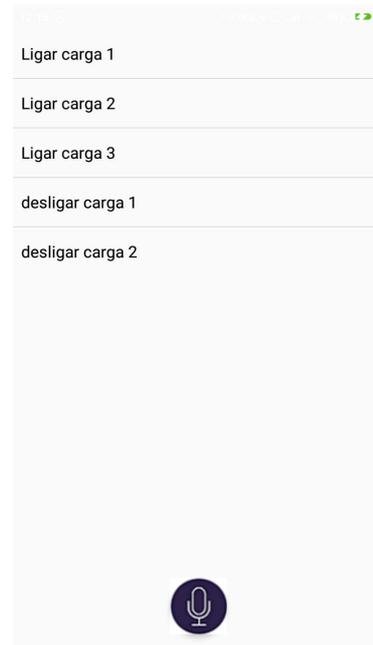
Para isso, o Raspberry Pi foi utilizado de forma que: o sistema recebia o comando de voz pela aplicação móvel, transcrevia a voz e mandava o texto via rede pelo roteador presente na residência para o Raspberry, que interpretava o comando e executava a ação de ligar ou desligar um aparelho, por meio de um módulo relé (LIMA, 2018).

Na Figura 8 (a) é exibido o sistema completo montado sobre a maquete construída, apresentando 4 lâmpadas conectadas aos seus respectivos relés. Já na Figura 8 (b) é exibida a aplicação mobile com seus respectivos botões para ligar ou desligar uma lâmpada, e uma opção para executar um comando por meio da voz (LIMA, 2018).

Figura 8 - Hardware e Software do trabalho.



(a)



(b)

Fonte: Lima, 2018.

Neste trabalho foi possível alcançar todos os objetivos desejados, contudo houve algumas dificuldades como durante a utilização do módulo *Wi-Fi* integrado do *Raspberry* que apresentou erros no socket, e o fato de que foi necessário implementar uma *Thread* para que o aplicativo móvel pudesse interagir com o usuário e realizar o envio dos comandos de forma simultânea (LIMA, 2018).

#### 2.2.6 Análise dos trabalhos relacionados

Os trabalhos relacionados atingiram seus objetivos por meio do desenvolvimento de: protótipos de monitoramento do consumo de energia elétrica com opção de ligar ou desligar um dispositivo da casa, ou com protótipo focado somente no controle de ligar ou desligar um eletroeletrônico. Os valores dos consumos de energia foram apresentados em aplicativos *mobile* ou plataformas *Web*, de forma que o usuário pôde monitorar o gasto de energia elétrica em determinado período e pôde ligar ou desligar os aparelhos da casa.

Contudo, houve trabalhos que diante de testes realizados para o envio de comandos liga/desliga de um eletroeletrônico e para gravação de valores em um banco de dados, apresentaram limitações devido ao atraso no envio das informações. Já em relação aos testes realizados para medição de valores de tensão e corrente dos aparelhos, o resultado obtido de um desses trabalhos, foi diferente do medido pela concessionária de energia, justamente devido ao atraso na gravação de valores em um banco de dados.

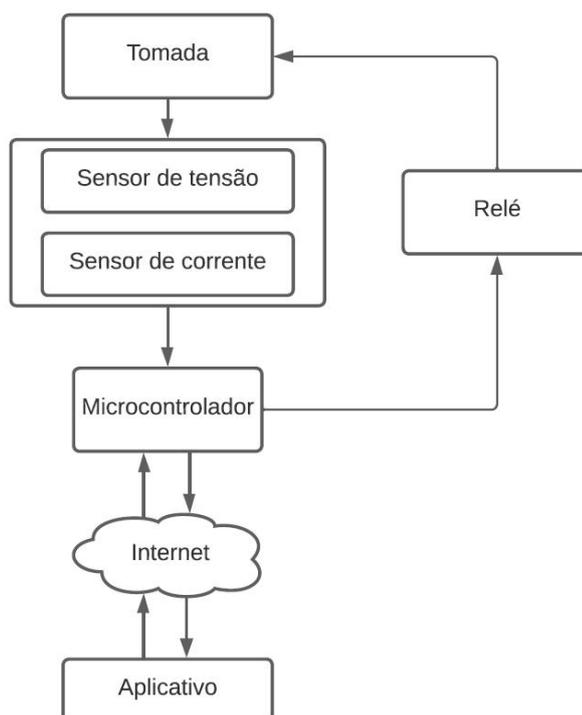
### 3 PROPOSTA DE SOLUÇÃO

Este capítulo apresenta a proposta de solução para o desenvolvimento de um protótipo de monitoramento e controle do consumo de energia elétrica de dispositivos eletroeletrônicos em uma residência.

#### 3.1 Diagrama de blocos

A Figura 9 exibe o diagrama de blocos do sistema proposto para este trabalho. Mais especificamente, foi desenvolvido um protótipo que através de sensores mediu a corrente e a tensão de aparelhos eletroeletrônicos conectados a uma tomada, utilizando um microcontrolador para processar estes dados. Além disso, foi utilizado um relé para controlar os dispositivos. Portanto, é calculado o consumo de energia elétrica e enviado, através da internet, para um banco de dados na nuvem capaz de armazenar essas informações. Esses dados são apresentados ao usuário por meio de um aplicativo Android.

Figura 9 - Diagrama de blocos do sistema proposto.



Fonte: Elaborado pelo autor.

## 3.2 Hardware

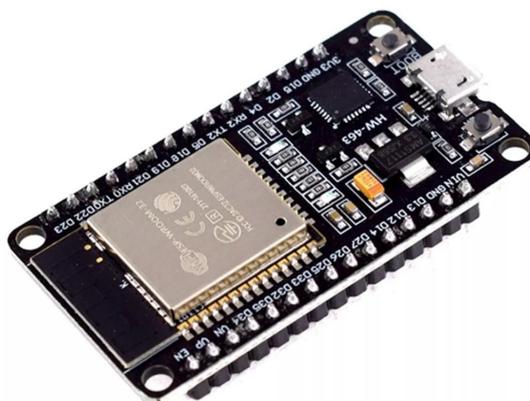
Esta seção apresenta os componentes de hardware utilizados para a montagem do protótipo de monitoramento e controle do consumo de energia de dispositivos eletroeletrônicos.

### 3.2.1 *Microcontrolador*

Um microcontrolador é um dispositivo eletrônico encapsulado em uma placa de silício. Ele é produzido em um único circuito eletrônico feito com semicondutores e pode executar tarefas de forma autônoma por meio de uma unidade de processamento, memórias para armazenamento de dados e portas de entrada e saída programáveis que permite interagir com outros dispositivos (MONTEIRO, 2019).

Neste trabalho foi utilizado o microcontrolador ESP32 desenvolvido pela empresa Espressif, exibido na Figura 10, e sendo que suas características podem ser vistas na Tabela 1. Este microcontrolador foi desenvolvido para ser utilizado em aplicações *mobile* e eletrônicos portáteis, contendo 30 pinos de entrada ou saída de uso geral. Além disso, ele apresenta conexão *Wi-Fi* de até 802.11 b/g/n, que transmite sinais sem fio através de ondas eletromagnéticas, e possui também conexão Bluetooth, versão 4.2, e Bluetooth de baixo consumo de energia (ESPRESSIF, 2023).

Figura 10 - Microcontrolador ESP32.



Fonte: Mercado Livre, 2023a.

Tabela 1 - Características do ESP32.

<b>CPU</b>	Xtensa Dual-Core 32-bit LX6
<b>MEMÓRIA ROM</b>	448 KBytes
<b>MEMÓRIA RAM</b>	520 KBytes
<b>Wireless padrão</b>	802.11 b/g/n
<b>Conexão Wi-Fi</b>	2.4Ghz (máximo de 150 Mbps)
<b>Quantidade de Portas</b>	30
<b>Bluetooth</b>	Classic, BLE 4.2
<b>Tensão de operação</b>	4,5 a 9 V
<b>Taxa de transferência</b>	110-460800bps
<b>Dimensões (comprimento x largura x espessura)</b>	5,5 x 2,8 x 1,3 cm

Fonte: Adaptado de Mercado Livre, 2023a.

### 3.2.2 Sensor de Tensão

Neste trabalho foi utilizado o sensor de tensão ZMPT101B para medir a tensão recebida em um eletroeletrônico e retornar o valor da mesma que foi processada pelo microcontrolador. Este sensor é capaz de medir tensões de até 250 V, sendo que suas principais características podem ser vistas na Tabela 2, e na Figura 11 é exibido o sensor (MERCADO LIVRE, 2023b).

O funcionamento do ZMPT101B baseia-se na utilização de um transformador para reduzir a tensão alternada proveniente da rede elétrica para um valor consideravelmente mais baixo, preservando ao máximo as formas de

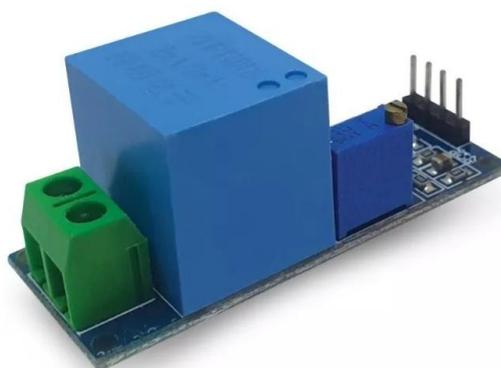
onda e os formatos originais. Ademais, o ZMPT101B é classificado como um módulo sensor de tensão AC, gerando uma saída analógica que varia conforme as mudanças na voltagem de entrada. Além disso, este sensor pode ser conectado a plataformas de código aberto, como Arduino, ESP8266 e Raspberry Pi (ROBOCRAZE, 2023).

Tabela 2 - Características do ZMPT101B.

<b>Marca</b>	OEM
<b>Modelo</b>	Zmpt101b
<b>Tensão de alimentação</b>	5 a 30VDC
<b>Tensão de medição</b>	0 a 250VAC
<b>Corrente nominal</b>	2mA
<b>Dimensões (comprimento x largura x espessura)</b>	22 x 20 x 51 mm
<b>Peso</b>	20g

Fonte: Adaptado de Mercado Livre, 2023b.

Figura 11 - Sensor de tensão ZMPT101B.

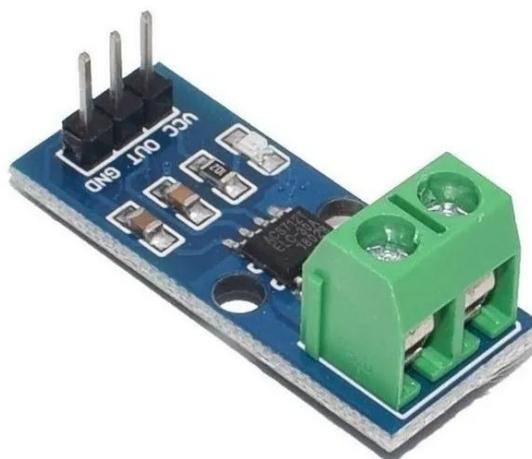


Fonte: Mercado Livre, 2023b.

### 3.2.3 Sensor de Corrente

O sensor de corrente ACS712 foi utilizado em conjunto com o microcontrolador para medir a corrente elétrica dos aparelhos eletrônicos. Esse sensor permite fazer leituras de corrente entre 0 à 30A, pode ser utilizado com corrente alternada (AC) e corrente contínua (DC), e sendo que seus bornes de ligação são completamente isolados da saída para o microcontrolador. Na Figura 12 é exibido um sensor, e na Tabela 3 é possível ver as características do mesmo (MERCADO LIVRE, 2023c).

Figura 12 - Sensor de corrente ACS712.



Fonte: Mercado Livre, 2023c.

Tabela 3 - Características do ACS712.

<b>Modelo</b>	ACS712
<b>Tensão de alimentação</b>	5V
<b>Corrente máxima</b>	5A, 20A, 30A
<b>Tempo de resposta</b>	5uS
<b>Margem de erro</b>	1.5% a 25C
<b>Dimensões (comprimento x largura x espessura)</b>	20 x 3 x 3 mm
<b>Peso</b>	4g

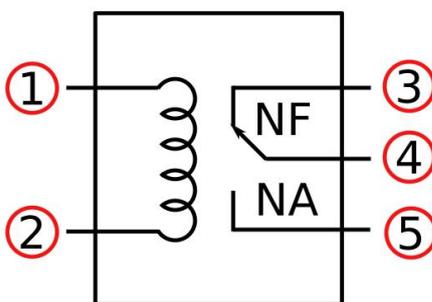
Fonte: Adaptado de Mercado Livre, 2023c.

### 3.2.4 Relé

O relé funciona com base no princípio eletromagnético, de forma a fechar ou abrir os contatos conectados aos terminais de um equipamento elétrico para acioná-lo. Assim como é exibido na Figura 13, o funcionamento do relé é da seguinte forma: quando a bobina interna (pinos 1 e 2) é alimentada por uma corrente, isso cria um campo magnético que atrai uma chapa metálica para um dos dois terminais (pino 3 ou pino 5) (FERREIRA, 2019).

Portanto, quando uma tensão é aplicada tem-se a mudança de contato, podendo estar em normalmente aberto (pino 5), quando tem uma tensão aplicada de até 5V na bobina, ou normalmente fechado (pino 3), quando não há tensão na bobina (FERREIRA, 2019).

Figura 13 - Circuito interno Relé.



Fonte: Isabella Ferreira, 2019.

Para este trabalho foi utilizado um módulo relé, o qual pode ser usado para realizar acionamentos de cargas de alta potência, de forma a ligar ou desligar um dispositivo eletrônico por exemplo. Na Figura 14 é exibido o módulo relé, e na Tabela 4 é apresentado as demais características desse componente (Mercado Livre, 2023d).

Figura 14 - Módulo Relé.



Fonte: Mercado Livre, 2023d.

Tabela 4 - Características do Relé.

<b>Modelo</b>	Módulo Relé 1 Canal 5v/10A
<b>Tensão de sinal</b>	5VCD
<b>Corrente típica do relé</b>	15~20mA
<b>Tempo de resposta</b>	5~10ms
<b>Dimensões</b>	17mm x 43mm x 20mm

Fonte: Adaptado de Mercado Livre, 2023d.

### 3.3 Software

Nesta seção serão apresentados os componentes de *software* para a montagem do protótipo de monitoramento e controle do consumo de energia.

#### 3.3.1 Banco de dados

O banco de dados armazena informações de forma a torná-las disponíveis quando solicitadas, sendo possível que o usuário realize operações de consulta, alteração, exclusão ou inserção de novos dados. Neste trabalho foi utilizado o *Firestore Realtime Database* que é um banco de dados não relacional na nuvem, sendo que seus dados são armazenados como JSON e sincronizados em tempo real para cada cliente conectado (FIREBASE, 2023a).

Para a criação de uma aplicação mobile que se conecte ao Firebase, no caso do ambiente de desenvolvimento integrado (IDE) Android Studio, isso pode ser feito de duas formas: utilizando o Console do Firebase, ou usar o Firebase Assistente do Android Studio (FIREBASE, 2023b).

### 3.3.2 Linguagem de programação Dart

Para este trabalho foi utilizado o Dart, que é uma linguagem otimizada para desenvolver aplicativos rápidos, sendo que ele é a base para o *framework Flutter* da Google. Além disso, a tecnologia do compilador Dart permite construir aplicações para diferentes plataformas, como *mobile*, *desktop* e plataformas web (DART, 2023).

Ademais, o Dart também oferece suporte a tarefas fundamentais para programadores, como formatação, análise, teste de código, uma tratativa para checar se uma variável pode ter valor nulo ou não, e uma verificação de tipo estático para garantir que o valor de uma variável sempre corresponda ao tipo estático da variável (DART, 2023).

### 3.3.3 Android Studio

O Android Studio é uma IDE utilizada para criar aplicativos para dispositivos Android. Ele é baseado no editor de código IntelliJ IDEA e oferece um ambiente unificado para desenvolvimento em todos os dispositivos Android. Um dos motivos para ele ser considerado uma boa IDE é devido ao seu emulador rápido com inúmeros recursos, ao seu sistema de *build* flexível baseado em Gradle e pela possibilidade de desenvolver aplicações em diferentes linguagens no Android Studio, usando o Dart por exemplo (ANDROID STUDIO, 2023).

### 3.3.4 Arduino Software IDE

O Arduino *Software IDE* foi utilizado para fazer o *upload* de programas para o ESP32 e permitir a comunicação com ele. Essa IDE contém um editor de texto, para escrever o código que foi utilizado, possui uma área de mensagem, que indica se o código deu erro por exemplo, e apresenta uma barra de ferramentas com diferentes botões e menus. Os programas escritos

em Arduino são feitos em um editor de texto, salvos em um arquivo com a extensão .ino e são chamados de *sketches* (ARDUINO, 2023).

## 4 DESENVOLVIMENTO DO PROTÓTIPO

Este capítulo apresenta os testes unitários e de integração para a geração do protótipo de monitoramento e controle do consumo de energia elétrica em uma residência.

### 4.1 Testes Unitários

Esta seção mostra os testes dos componentes de *hardware* e a montagem do protótipo de monitoramento e controle do consumo de energia. Os programas foram desenvolvidos usando um computador com processador AMD Ryzen 5, memória RAM de 8 GB e sistema operacional Windows 11 de 64 bits.

#### 4.1.1 ESP32

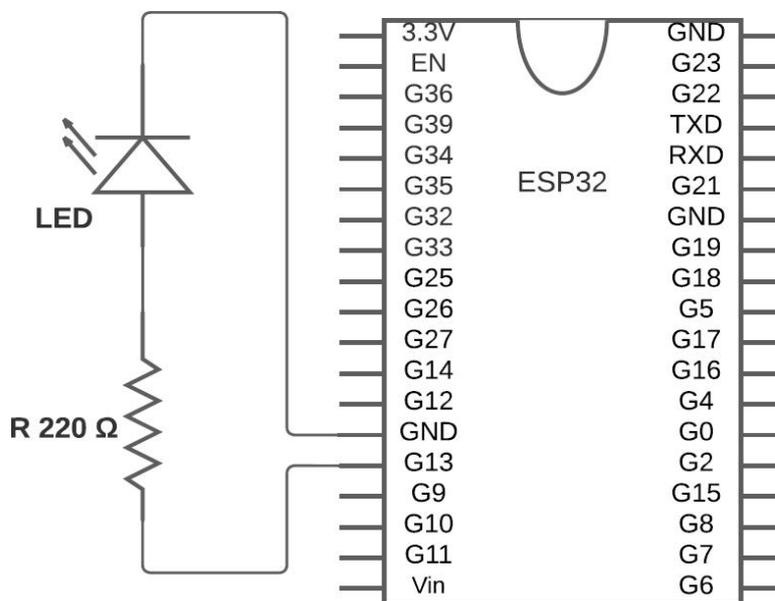
O ESP32 foi testado com o código na Figura 15 feito para ligar e desligar um LED a cada 1 segundo. Para isso, foi montado um circuito usando um ESP32, um resistor de 220 Ohms e um LED, exibido na Figura 16.

Figura 15 - Código para ligar/desligar o LED.

```
Led
//---Função Principal---//
void setup() {
  //Configura o pino GPIO13 como saída
  pinMode(13, OUTPUT);
}
//---Loop Infinito---//
void loop() {
  //'Liga' a GPIO13
  digitalWrite(13,HIGH);
  //Pausa o programa
  delay(1000);
  //'Desliga' a GPIO13
  digitalWrite(13,LOW);
  //Pausa o programa
  delay(1000);
}
```

Fonte: Elaborado pelo autor.

Figura 16 - Circuito para ligar/desligar o LED.



Fonte: Elaborado pelo autor.

#### 4.1.2 Relé

Para este teste foi usado o código na Figura 17, utilizando um módulo relé, um ESP32 e uma lâmpada de 12 *Watts*. Além disso, utilizou-se um transistor BC337, pois o relé possui um transistor 2TY que se conecta de forma que fica ativado com o GND e desativado com 5V, e por meio do transistor isso foi invertido. Na Figura 18 é exibido o circuito de teste do relé.

O transistor BC337 funciona como uma chave para permitir ou não a passagem de corrente, assim permitindo a passagem do sinal de entrada IN do relé ao GND. Isso permite que o relé seja ativado ou desativado dependendo do sinal enviado pelo ESP32 através da porta 13, dessa forma o resultado é um *loop* que liga e desliga a lâmpada a cada 1 segundo.

Figura 17 - Código para ligar/desligar lâmpada com relé.

```

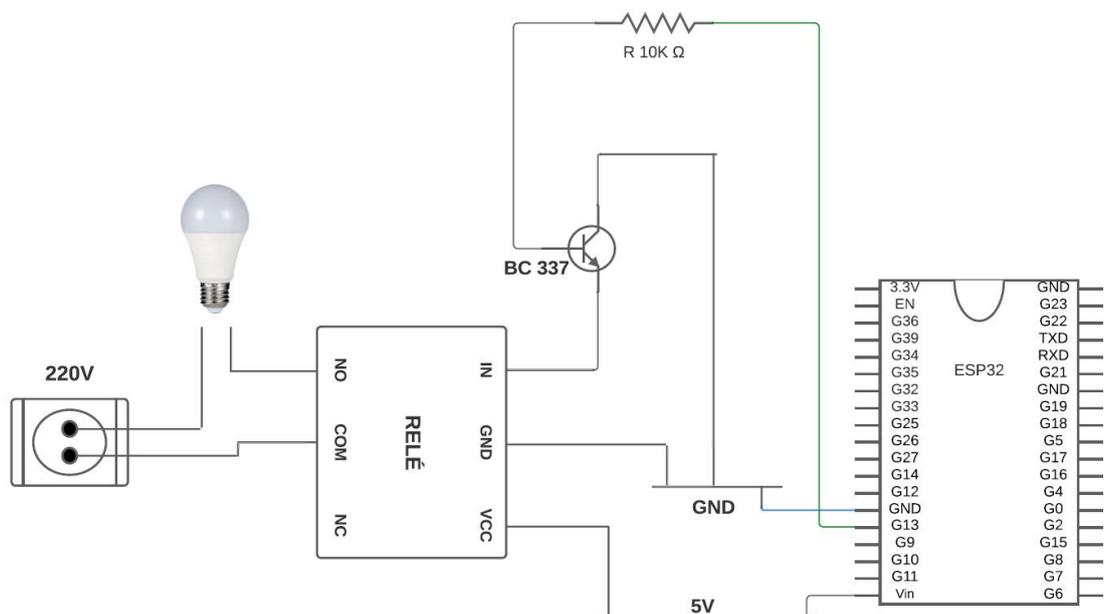
Lampada_Rele
void setup() {
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}

```

Fonte: Elaborado pelo autor.

Figura 18 - Circuito para ligar/desligar lâmpada com relé.



Fonte: Elaborado pelo autor.

#### 4.1.3 Sensor de corrente ACS712

Para este teste foi usado um sensor ACS712, um ESP32 e um secador de cabelo Taiff *Smart* 220-230V. Na Figura 19 é exibido o código utilizado, adaptado de Martins (2022), aproveitando-se da biblioteca *Emonlib* que

implementa a função *calcIrms* para medir valores de tensão e corrente, cujo valor de parâmetro foi de 1480 amostras.

Além disso, foi passado como parâmetro para função *current*, tanto o pino 13 do ESP32, quanto a constante de calibração de valor 1.0, que foi descoberta de forma empírica medindo com o multímetro os valores de corrente e fazendo a relação com os valores obtidos pelo sensor. Na Figura 20 é exibido o circuito montado para realizar este teste, já na Figura 21 é exibido os valores medidos de corrente entre 1.17 e 1.25A, e de potência entre 257.92 e 274.44 Volt-Ampére.

Figura 19 - Código para medir corrente de um secador de cabelo.

```

Corrente
#include "EmonLib.h"           // Include Emon Library
EnergyMonitor emon1;         // Create an instance

void setup()
{
  Serial.begin(115200);
  emon1.current(13, 1.0); // Constante de calibração 1.0
}

void loop()
{
  double Irms = emon1.calcIrms(1480);

  Serial.print("Corrente: ");
  Serial.print(Irms);
  Serial.println(" A");

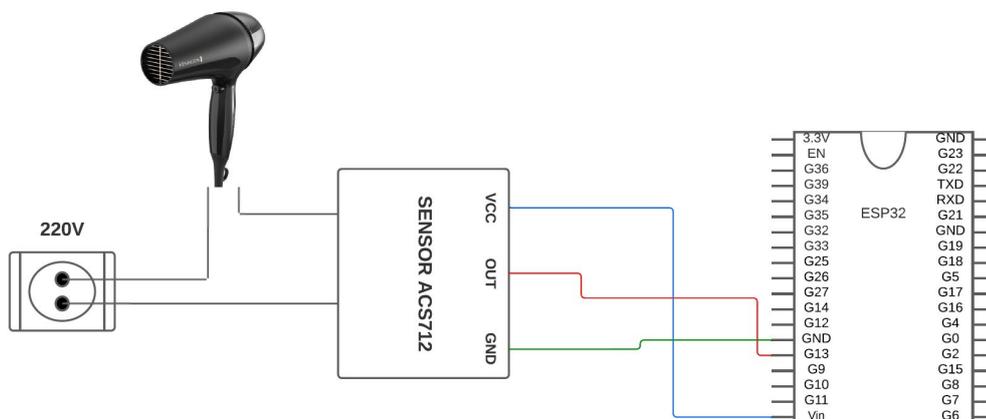
  Serial.print("Potencia: ");
  Serial.print(Irms * 220.00);
  Serial.println(" VA");

  Serial.println();
  delay(400);
}

```

Fonte: Elaborado pelo autor.

Figura 20 - Circuito para medir corrente de um secador de cabelo.



Fonte: Elaborado pelo autor.

Figura 21 - Valores medidos pelo sensor de corrente.

```

COM3
Corrente: 1.17 A
Potencia: 257.92 VA

Corrente: 1.25 A
Potencia: 274.43 VA

Corrente: 1.25 A
Potencia: 274.44 VA

Corrente: 1.25 A
Potencia: 273.98 VA

Corrente: 1.24 A
Potencia: 272.20 VA

Corrente: 1.23 A
Potencia: 270.96 VA

Corrente: 1.23 A
Potencia: 270.78 VA

Corrente: 1.24 A
Potencia: 272.98 VA

Corrente: 1.24 A
Potencia: 273.13 VA

Corrente: 1.23 A
Potencia: 270.76 VA

Corrente: 0.07 A
Potencia: 16.44 VA

Corrente: 0.04 A
Potencia: 9.02 VA

Corrente: 0.04 A
  
```

Auto-rolagem  Show timestamp

Fonte: Elaborado pelo autor.

#### 4.1.4 Sensor de tensão ZMPT101B

Para este teste foi utilizado um sensor ZMPT101B, um ESP32 e uma tomada. Para isso utilizou-se o código da Figura 22 para medir os valores de tensão de uma tomada, sendo que a função *voltage* da biblioteca *Emonlib* passa como parâmetros o pino 13 do ESP32, o valor de calibração 125.0 que foi obtido de forma empírica com a utilização de um multímetro, e o parâmetro de correção para os erros de amostragem da tensão cujo valor é 1.7.

Quanto a função *calcVI*, ela é responsável pelo cálculo da tensão da rede elétrica em volts. Nessa função, foram passados dois parâmetros essenciais: o valor 20, que representa o número de ciclos de rede elétrica utilizados para calcular a tensão, e o valor 1000, que indica o período de amostragem em milissegundos. Após a execução da função, ocorre a impressão da tensão medida em volts, exibida na Figura 23, já na Figura 24 é exibido o circuito montado.

Figura 22 - Código para medir tensão de uma tomada.

```
Tensao
#include "EmonLib.h"

EnergyMonitor emon1;

void setup() {
  Serial.begin(115200);

  emon1.voltage(13, 125.0, 1.7);
}

void loop() {
  emon1.calcVI(20, 1000);
  float voltagem = emon1.Vrms;
  Serial.print("Tensão: ");
  Serial.print(voltagem);
  Serial.println(" V");
}
```

Fonte: Elaborado pelo autor.

Figura 23 - Valores medidos pelo sensor de tensão.

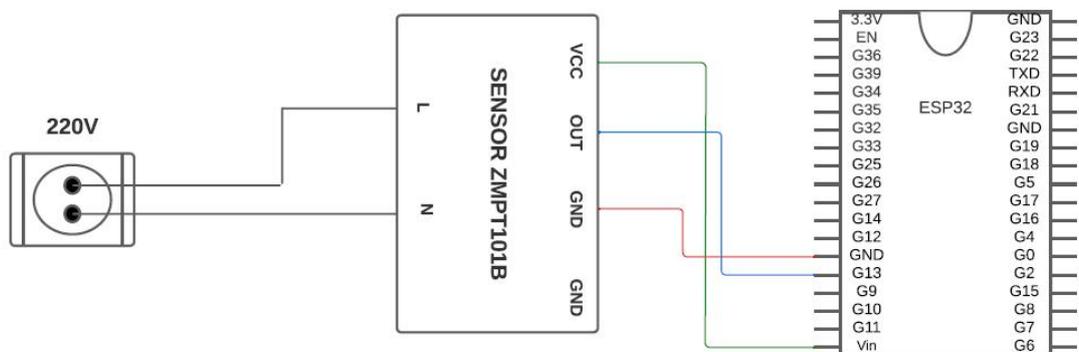
```

COM3
-----
Tensão: 219.98 V
Tensão: 219.44 V
Tensão: 219.88 V
Tensão: 219.95 V
Tensão: 220.49 V
Tensão: 220.71 V
Tensão: 220.59 V
Tensão: 219.41 V
Tensão: 219.74 V
Tensão: 219.76 V
Tensão: 217.31 V
Tensão: 219.76 V
Tensão: 218.21 V
Tensão: 220.23 V
Tensão: 219.80 V
Tensão: 218.05 V
Tensão: 219.31 V
Tensão: 220.09 V
Tensão: 218.94 V
Tensão: 221.02 V

```

Fonte: Elaborado pelo autor.

Figura 24 - Circuito para medir tensão de uma tomada.



Fonte: Elaborado pelo autor.

#### 4.1.5 Teste de integração

O teste de integração foi realizado com o ESP32, o sensor de corrente ACS712, o sensor de tensão ZMPT101B, um secador de cabelo, um ferro de passar e um relé. Para isso, foi utilizado o código na Figura 25, que reuniu os códigos feitos para os testes unitários dos sensores e do relé. Além disso, foi

utilizado o pino 13 e 12, sendo o primeiro para o sensor ZMPT101B e o segundo para o sensor ACS712. Na Figura 26 é exibido o circuito, e na Figura 27 é exibido os valores obtidos de corrente, tensão e potência.

Figura 25 - Código para medir tensão e corrente de um secador de cabelo e ligar uma lâmpada.

```
Tensao_Corrente_Rele
#include "EmonLib.h"
EnergyMonitor emon1;

void setup() {
  Serial.begin(115200);

  pinMode(14, OUTPUT);

  emon1.voltage(13, 125.0, 1.7);
  emon1.current(12, 1.0);
}

void loop() {

  digitalWrite(14, HIGH);
  delay(1000);
  digitalWrite(14, LOW);
  delay(1000);

  emon1.calcVI(20, 1000);
  double Irms = emon1.calcIrms(1480);

  float voltagem = emon1.Vrms;

  Serial.print("Corrente: ");
  Serial.print(Irms);
  Serial.println(" A");

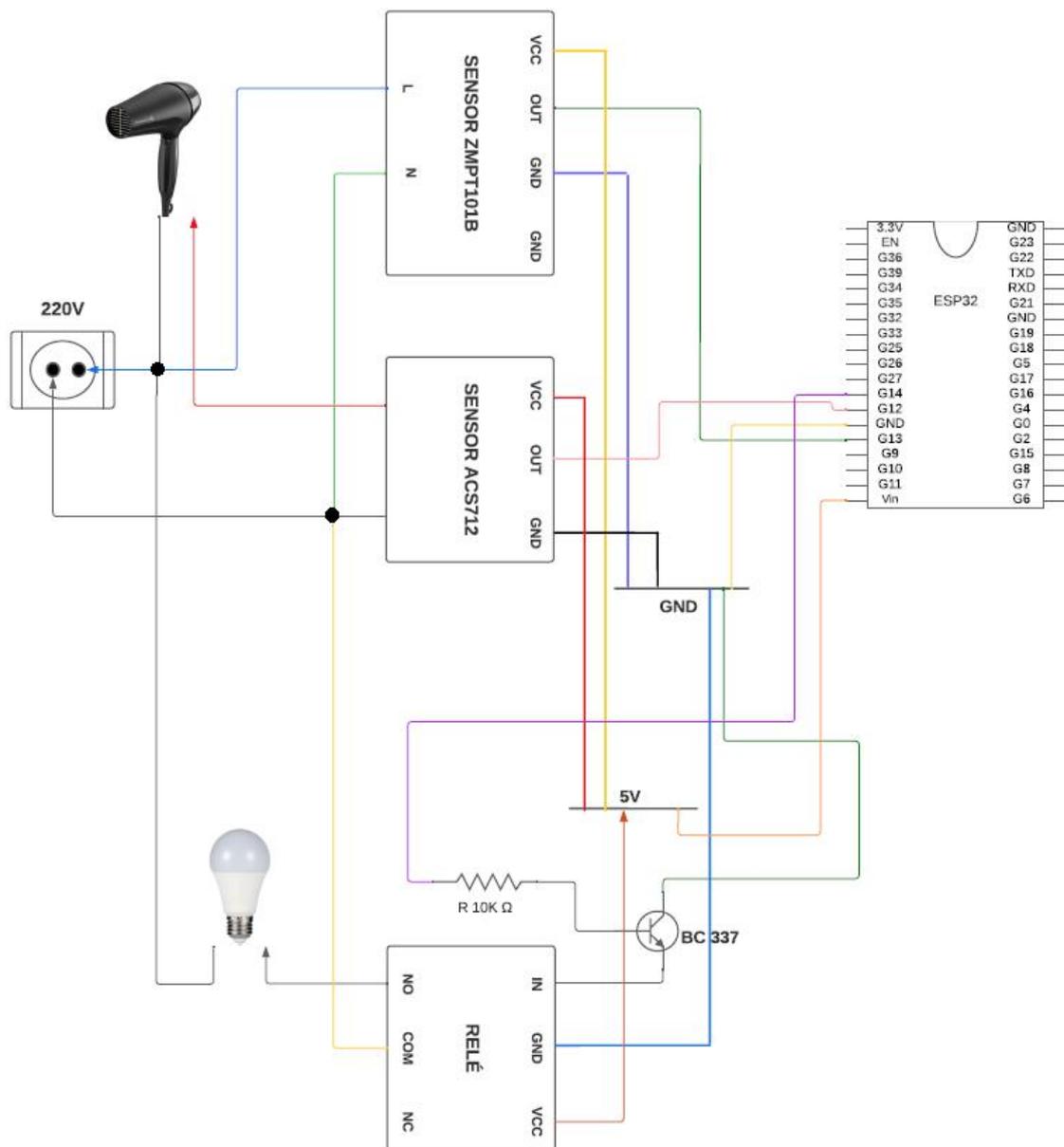
  Serial.print("Tensão: ");
  Serial.print(voltagem);
  Serial.println(" V");

  Serial.print("Potencia: ");
  Serial.print(Irms * 220.00);
  Serial.println(" VA");

  Serial.println();
  delay(400);
}
```

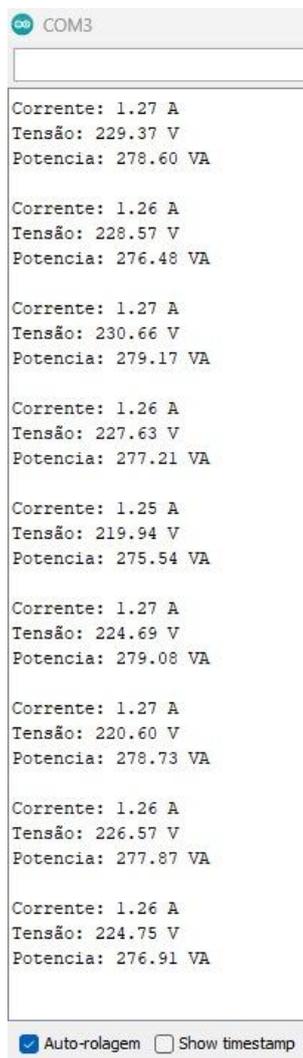
Fonte: Elaborado pelo autor.

Figura 26 - Circuito para medir tensão e corrente de um secador de cabelo e ligar uma lâmpada.



Fonte: Elaborado pelo autor.

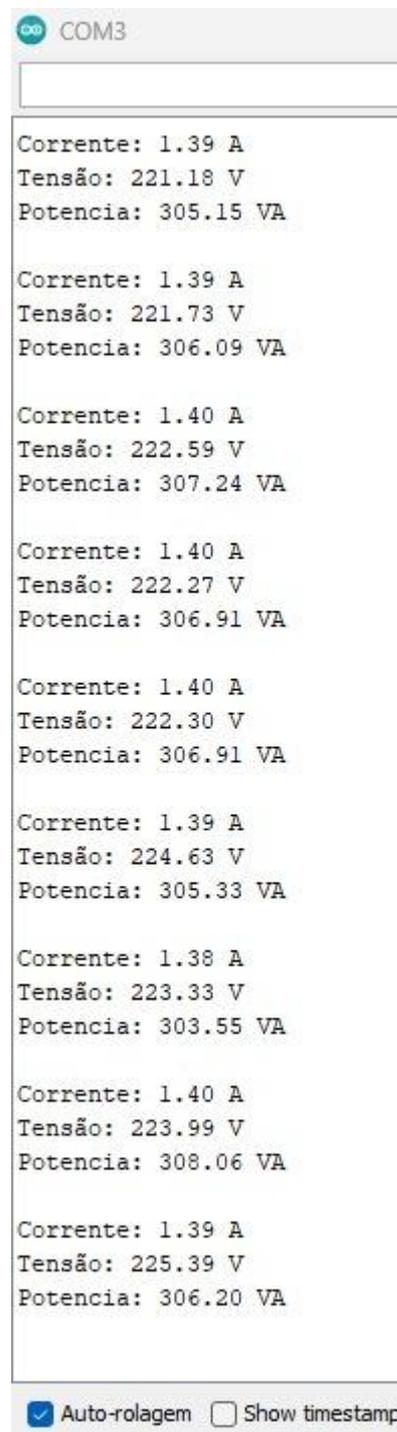
Figura 27 - Valores medidos pelo sensor de tensão e corrente.



Fonte: Elaborado pelo autor.

Utilizando-se do mesmo circuito e do mesmo código fonte, também foi realizado a medição do consumo de energia de um ferro de passar modelo Arno UltraGliss44. A Figura 28 exhibe os valores de corrente, tensão e potência medidos do aparelho. Vale ressaltar que, tanto o valor de corrente quanto o valor de potência do ferro de passar foram maiores que do secador de cabelo.

Figura 28 - Valores medidos pelo sensor de tensão e corrente do ferro de passar.



Fonte: Elaborado pelo autor.

Por meio dos testes unitários e do teste de integração realizado, foi possível comprovar a eficácia dos sensores para realizar a medição referente a

corrente e tensão dos eletroeletrônicos, sendo que os valores medidos pelos sensores foram próximos aos valores medidos por um multímetro. Com isso, o aplicativo Android desenvolvido neste trabalho pode utilizar dos valores obtidos pelos sensores e da função do relé para cumprir com o tema proposto, que é realizar o controle e monitoramento de energia de um eletroeletrônico.

## 5 DESENVOLVIMENTO DO APLICATIVO

Este capítulo apresenta o desenvolvimento de um aplicativo de controle e monitoramento de energia elétrica residencial.

### 5.1 Comunicação entre o aplicativo, o Firebase e a ESP32

Inicialmente para testar a comunicação do aplicativo com o Firebase foi implementado dois botões em uma tela, o botão 'Ligar' para ligar o LED e o botão 'Desligar' para desligar o LED, sendo que a variável *ref* recebe a referência do banco de dados do Firebase, conforme trecho de código exibido na Figura 29. Quando o usuário clica no botão 'Ligar', a variável *ref* aciona a função *update* (*ref.update*) para atualizar os dados do Firebase com a informação 'on', a fim de ligar o LED. De maneira semelhante, quando o usuário clicar no botão 'Desligar', ocorre uma atualização dos dados do Firebase com a informação 'off', para desligar o LED.

Figura 29 – Trecho de código para enviar o dado de ligar o LED para o Firebase.

```
final ref = FirebaseDatabase.instance.ref("");
Row(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: <Widget>[
    ElevatedButton(
      onPressed: () {
        ref.update({
          "Led_Status": 'on',
        });
      },
      style: ButtonStyle(
        backgroundColor: MaterialStateProperty.all(Colors.green),
        fixedSize: MaterialStateProperty.all(Size(100, 50)),
      ), // ButtonStyle
    child: const Text(
      'Ligar',
      style: TextStyle(fontSize: 18),
    ), // Text
  ), // ElevatedButton
```

Fonte: Elaborado pelo autor.

Em relação a comunicação entre o Firebase e a ESP32, foi desenvolvido o código exibido na Figura 30, adaptado de Pires (2021), que por meio da biblioteca Wifi.h permite a conexão da ESP32 à internet, e por meio da biblioteca FirebaseESP32.h possibilita a conexão da ESP com o Firebase. Foram definidos os parâmetros para conexão com o Firebase e, o identificador e senha do WiFi local.

Figura 30 – Código para ligar e desligar o LED pelo aplicativo usando o Firebase.

```

Led.ino
1  #include <WiFi.h>
2  #include <FirebaseESP32.h>
3  #define WIFI_SSID "S20"
4  #define WIFI_PASSWORD [REDACTED]
5  #define API_KEY [REDACTED]
6  #define DATABASE_URL "https://tcc-e7aa4-default-rtdb.firebaseio.com"
7
8  // Objeto de dados do Firebase
9  FirebaseData firebaseData;
10
11 // Objeto de autenticação do Firebase
12 FirebaseAuth auth;
13
14 // Objeto de configuração do Firebase
15 FirebaseConfig config;
16
17 //---Função Principal---//
18 void setup() {
19   Serial.begin(115200);
20   delay(1000);
21   pinMode(13, OUTPUT); //Configura o pino GPIO13 como saída
22   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
23   Serial.print("Conectando ao ");
24   Serial.print(WIFI_SSID);
25
26   while (WiFi.status() != WL_CONNECTED) {
27     Serial.print(".");
28     delay(500);
29   }
30
31   Serial.println();
32   Serial.print("Conectado ao ");
33   Serial.println(WIFI_SSID);
34
35   // Configura chave API do Firebase
36   config.api_key = API_KEY;
37
38   // Configura a Url do Firebase Realtime Database
39   config.database_url = DATABASE_URL;
40
41   // Habilita reconexão com WiFi
42   Firebase.reconnectWiFi(true);
43
44   Firebase.signUp(&config, &auth, "", "");
45
46   Firebase.begin(&config, &auth);
47 }
48
49 //---Loop Infinito---//
50 void loop() {
51
52   Firebase.getString(firebaseData, "Led_Status");
53
54   if (firebaseData.stringData() == "on") {
55     Serial.println("Led ON");
56     digitalWrite(13, HIGH); //Liga' a GPIO13
57   }
58   else if (firebaseData.stringData() == "off") {
59     Serial.println("Led OFF");
60     digitalWrite(13, LOW); //Desliga' a GPIO13
61   }
62 }
63

```

Fonte: Elaborado pelo autor.

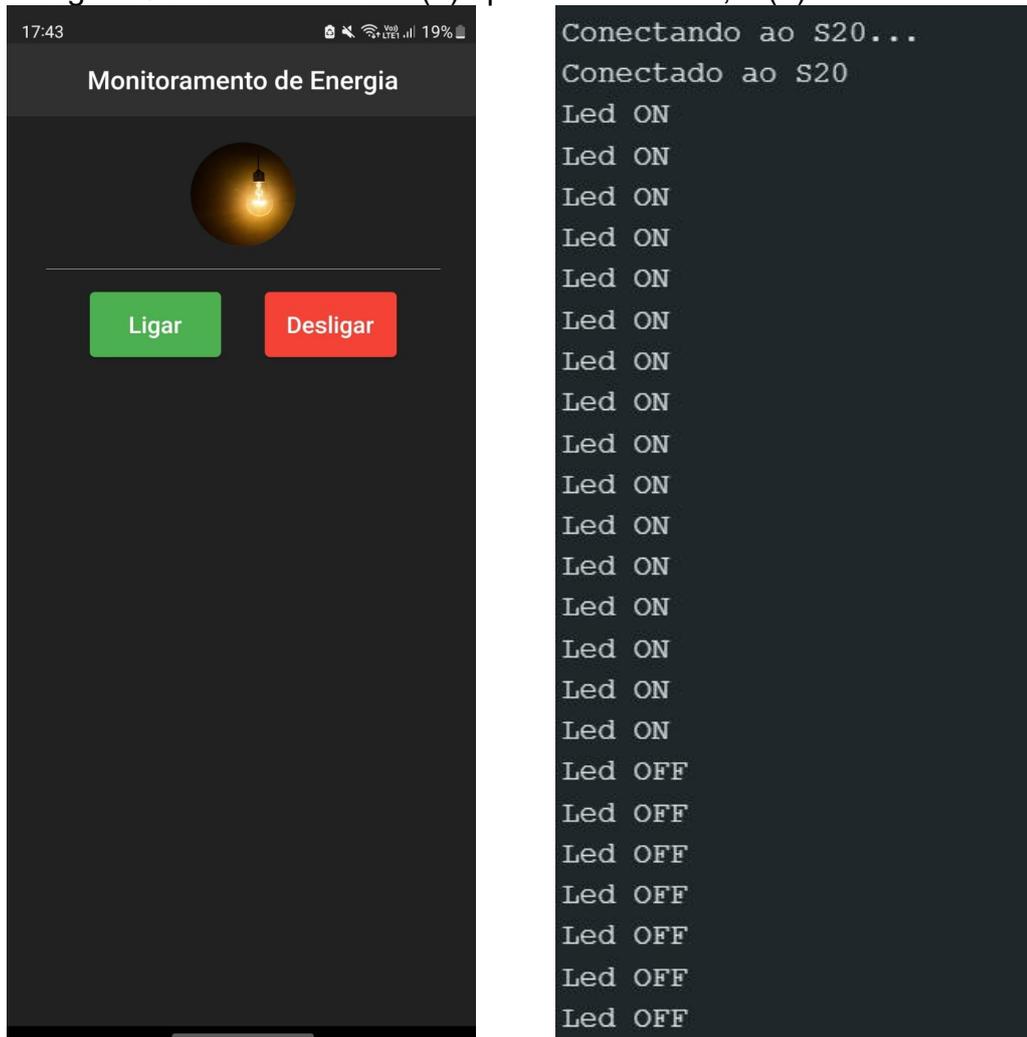
O pino 13 da ESP32 foi definido para ligar e desligar o LED, já a função WiFi.begin passa como parâmetros o identificador e a senha do Wi-fi

conectando a ESP32 à internet. A função `Firebase.signUp` recebe os parâmetros de configuração e autenticação para realizar o login no Firebase, já a função `Firebase.begin` passa como parâmetros o host e a autenticação, conectando a ESP32 ao Firebase.

Na função `loop`, o `Firebase.getString` obtém a variável `'Led_Status'` do Firebase e depois, utilizando a função `FirebaseData.stringData` obtém o valor da variável `'Led_Status'` para verificação. Se o valor da variável for `'on'`, liga-se o LED, se for `'off'`, o desliga.

Na Figura 31 (a) é exibida a tela com os dois botões `'Ligar'` e `'Desligar'`, na Figura 31 (b) é exibida a tela de resultados do monitor serial. Nas Figuras 32 (a) e (b) é exibida a tela do Firebase RealTime Database, demonstrando a transição do `Led_Status` de `'on'` para `'off'`. Isso ocorre quando o comando correspondente é enviado por meio do aplicativo, resultando na mudança do estado do LED de ligado para desligado, conforme exibido nas Figuras 33 (a) e (b).

Figura 31 – Resultado do: (a) aplicativo Android; e (b) monitor serial.



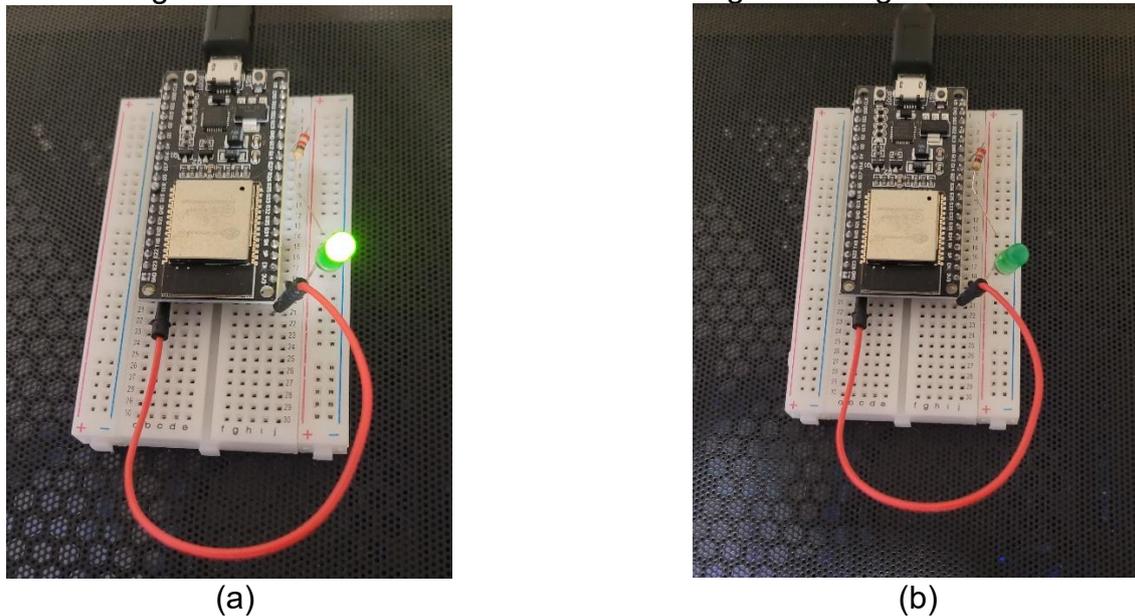
Fonte: Elaborado pelo autor.

Figura 32 – Resultado do Firebase.



Fonte: Elaborado pelo autor.

Figura 33 – Resultado dos comandos de ligar e desligar o LED.



Fonte: Elaborado pelo autor.

Para possibilitar que a ESP32 obtenha a data e a hora, a fim de relacionar estas informações com a potência em Watts medida do aparelho eletroeletrônico monitorado, foi necessário conectar a ESP32 à internet, o que permitiu a utilização do *Network Time Protocol* (NTP) para obter a data e a hora atual. Além disso, foi necessário fornecer o fuso horário como parâmetro na função `setTimeOffset()`, sendo que o valor utilizado foi `-10800`, que equivale ao *Greenwich Mean Time* (GMT) menos 3 horas (GMT-3) em segundos.

## 5.2 Versão final do aplicativo

O aplicativo Energy Monitoramento foi desenvolvido usando a linguagem de programação Dart, sendo que o app é dividido em tela de login e tela principal. A tela de login permite cadastrar e acessar o aplicativo por meio de e-mail e senha, e conta Google. Caso o usuário esqueça qual foi a senha cadastrada, há um botão 'Esqueci minha senha' para permitir cadastrar uma nova.

Após efetuar o acesso no aplicativo, o usuário é redirecionado para a tela principal, primeira aba, que apresenta os botões Ligar e Desligar para

habilitar e desabilitar a tomada, e apresenta um gráfico do consumo em kWh por dia do aparelho monitorado.

Já na segunda aba do aplicativo, há uma tabela referente aos dados medidos de consumo de energia em kW, um container referente ao consumo total em kW, outro container referente ao valor total em reais em relação ao consumo total de energia elétrica, um botão para poder alterar a tarifa de energia adotada pelo app, e um outro botão referente a um filtro de data e hora dos dados de consumo em kW.

### 5.2.1 *Tela de Login*

A tela de Login, Figura 34, tem como objetivo permitir que o usuário consiga acessar a tela principal do aplicativo por meio de conta Google ou criando um cadastro no aplicativo utilizando e-mail e senha. A tela contém a logo do aplicativo 'Energy Monitoramento', e os botões 'Entrar com Google' e 'Entrar com E-mail'.

Figura 34 – Tela de Login.

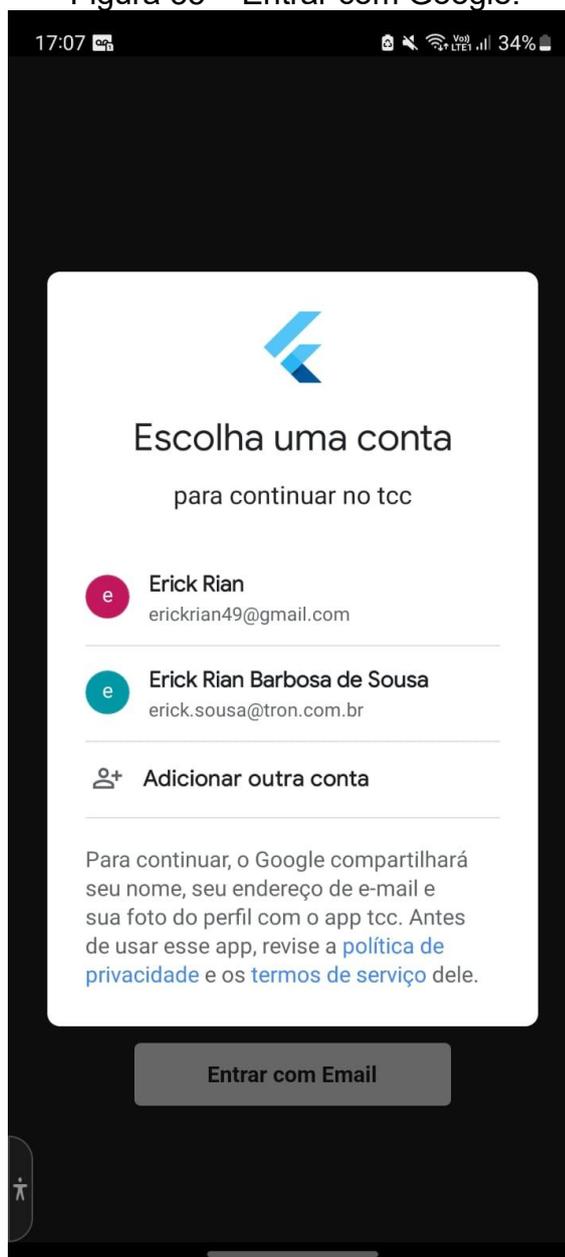


Fonte: Elaborado pelo autor.

### 5.2.2 Entrar com Google

Ao clicar em 'Entrar com Google', o usuário é direcionado para a tela na Figura 35, que exibe as opções de escolher uma conta já previamente cadastrada ou 'Adicionar outra conta', sendo que no final deste processo ele é redirecionado para a tela principal do aplicativo.

Figura 35 – Entrar com Google.



Fonte: Elaborado pelo autor.

### 5.2.3 Entrar com E-mail

Após clicar em 'Entrar com E-mail', o usuário é direcionado a uma tela composta pela logo do App, espaço para inserir as informações de e-mail e senha, e um botão para entrar na tela principal, como exibido na Figura 36. Além disso, há uma opção 'Criar conta', caso o usuário ainda não tenha uma

conta, e outra opção 'Esqueci minha senha', se o mesmo precisar cadastrar uma nova senha.

Figura 36 – Login via e-mail e senha.



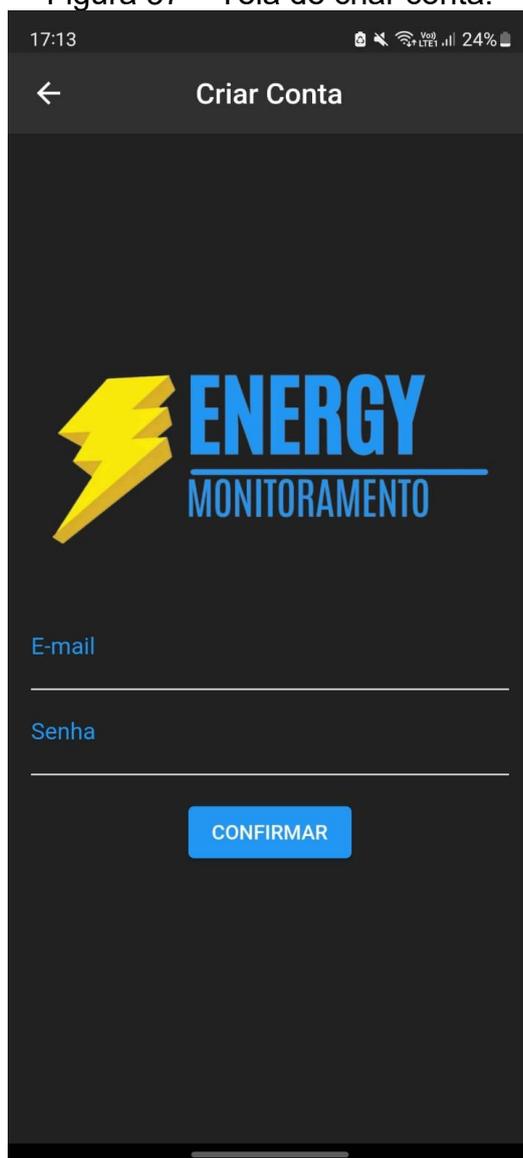
The image shows a mobile application login screen. At the top, the status bar displays the time 17:13, signal strength, VoLTE, and 24% battery. Below the status bar is a dark header with a back arrow and the title 'Login'. The main content area has a dark background. It features the 'ENERGY MONITORAMENTO' logo, which consists of a yellow lightning bolt icon and the text 'ENERGY' in blue above 'MONITORAMENTO' in blue. Below the logo are two input fields: 'E-mail' and 'Senha'. To the right of the 'E-mail' field is a link 'Criar conta'. To the right of the 'Senha' field is a link 'Esqueci minha senha'. At the bottom center is a blue button with the text 'ENTRAR'.

Fonte: Elaborado pelo autor.

### 5.2.3.1 Opção de Criar Conta

Nesta tela, Figura 37, o usuário pode informar o e-mail e a senha desejados para se cadastrar no aplicativo, posteriormente o mesmo deve clicar no botão 'CONFIRMAR' para cadastrar a conta.

Figura 37 – Tela de criar conta.



A imagem mostra a interface de usuário para a criação de uma conta no aplicativo 'ENERGY MONITORAMENTO'. No topo, há uma barra de status com o horário 17:13, ícones de notificação, Wi-Fi, LTE e bateria em 24%. Abaixo, uma barra de navegação com um ícone de seta para trás e o título 'Criar Conta'. O logotipo do aplicativo, que consiste em um relâmpago amarelo e o texto 'ENERGY MONITORAMENTO' em azul, está centralizado. Abaixo do logotipo, há dois campos de entrada de texto: 'E-mail' e 'Senha', cada um com uma linha de base branca. No final da tela, há um botão azul com o texto 'CONFIRMAR' em branco.

Fonte: Elaborado pelo autor.

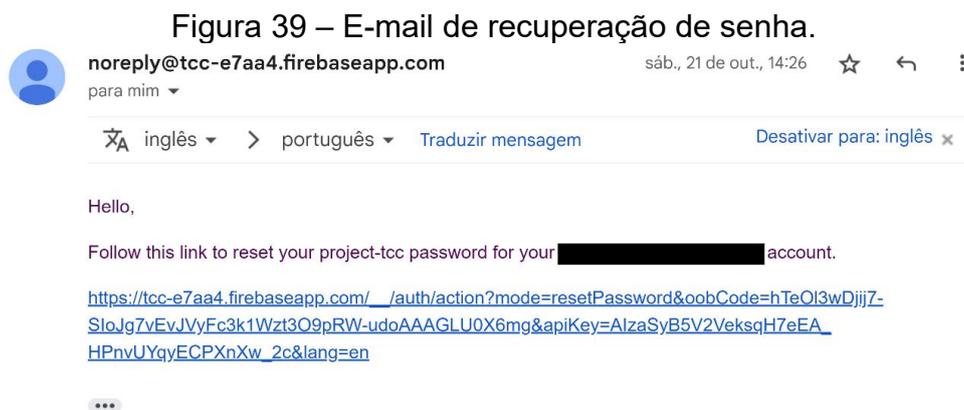
### 5.2.3.2 Opção de Esqueci minha Senha

Como exibido na Figura 38, há uma opção para informar o e-mail que deseja cadastrar uma nova senha. Após informar o e-mail e clicar no botão 'CONFIRMAR', o Firebase envia automaticamente um e-mail de recuperação de senha para o e-mail do usuário, assim como exibido na Figura 39.

Figura 38 – Tela esqueci minha senha.



Fonte: Elaborado pelo autor.



Fonte: Elaborado pelo autor.

### 5.3 Tela Principal

Na Figura 40 é exibida a tela principal do aplicativo que é composta por um botão de sair do aplicativo, item 1, além disso, a página apresenta duas abas, a aba 'HOME' (item 2) e a aba 'TABELA' (item 3). A aba 'HOME' contém dois botões, sendo um para ligar e outro para desligar o dispositivo monitorado (itens 4 e 5). Já no item 6, há um gráfico referente ao consumo de energia elétrica, com o eixo das abscissas  $x$  representando o tempo, e o eixo das ordenadas  $y$  representando a potência em kW.

Figura 40 – Tela Principal.



Fonte: Elaborado pelo autor.

Como exemplificado no gráfico na Figura 40, no tempo de 19:52:07 à 19:52:30 o aparelho eletroeletrônico monitorado (secador de cabelo) estava desligado e foi ligado na potência mínima (*Low*) do aparelho até 19:52:53. Posteriormente, o aparelho foi ajustado para a potência máxima do secador (*High*), como observado no intervalo de 19:52:53 a 19:53:53. Por fim, o aparelho foi desligado e ligado na potência mínima novamente.

## 5.4 Aba Tabela

Na Figura 41 é exibida a aba 'TABELA', nela foi implementado dois botões, sendo um deles mostrado no item 1, cuja funcionalidade é de definir o valor da tarifa de energia elétrica que o usuário deseja que o aplicativo considere. O outro botão é referente ao item 2, cuja funcionalidade é a de filtrar os dados do aplicativo por meio de Data e Hora informados pelo usuário.

Além disso, vale ressaltar que os objetivos dessa tela foram: (a) mostrar a informação do consumo total em kWh registrado pelo aplicativo, e o respectivo valor em reais referente ao consumo de energia elétrica, sendo visualizado pelos itens 3 e 4; e (b) mostrar informações de data e hora junto com o consumo de potência em kW daquele momento, podendo ser visto pelo item 5.

Em relação aos cálculos para se chegar no consumo total e valor, em relação ao item 3, levando em consideração que nenhum filtro de data e hora foi aplicado, o aplicativo somou todas as potências, em Watts, registradas até então, multiplicou por 1 hora e no final dividiu por 1000. Já no item 4 o cálculo foi multiplicar o consumo total em kWh pelo valor da tarifa de energia elétrica registrada no app, que caso o usuário não tenha definido o valor, o mesmo por padrão é de R\$ 0,89.

Figura 41 – Aba Tabela.



Fonte: Elaborado pelo autor.

Na Figura 42 é exibida uma janela para definição do valor em reais da tarifa de energia, sendo que ao alterar este valor, isso reflete nos dados do campo 'Valor' no item 4 na Figura 41.

Figura 42 – Tela Definir Tarifa.



Fonte: Elaborado pelo autor.

Na Figura 43 é exibida a tela de filtro que permite definir a data e horário de início e de fim para seleção dos dados, sendo que este filtro reflete no gráfico exibido na Figura 40, e nos dados do 'Consumo Total', 'Valor' e tabela na Figura 41. Nessa tela há dois botões, um para filtrar os dados e outro para limpar a formatação, de forma a mostrar todos os dados da base de dados.

Figura 43 – Filtro por Data e Hora.



Fonte: Elaborado pelo autor.

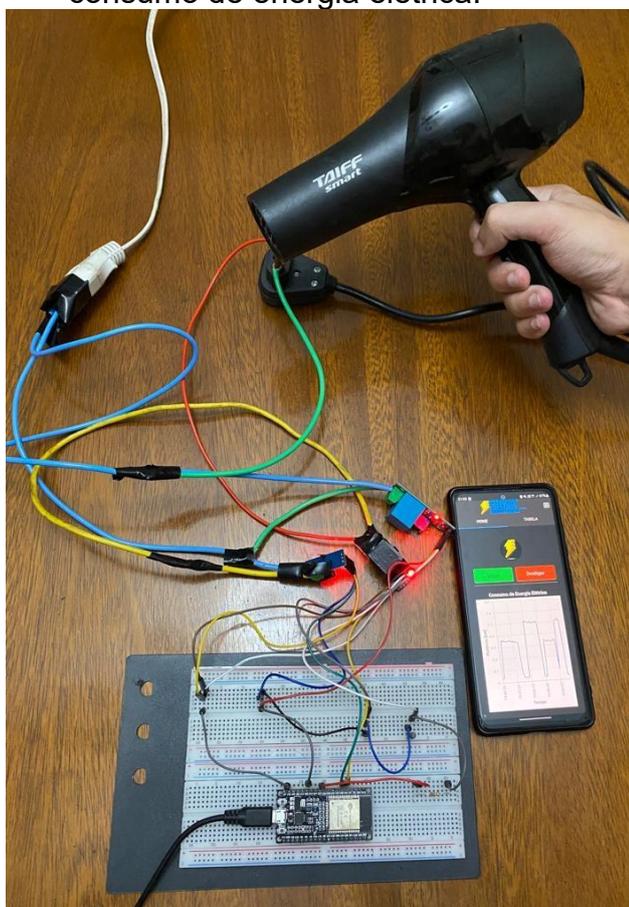
## 6 RESULTADOS OBTIDOS

Este capítulo mostra os resultados obtidos com o desenvolvimento do protótipo de controle e monitoramento de energia elétrica residencial.

### 6.1 Análise dos resultados obtidos

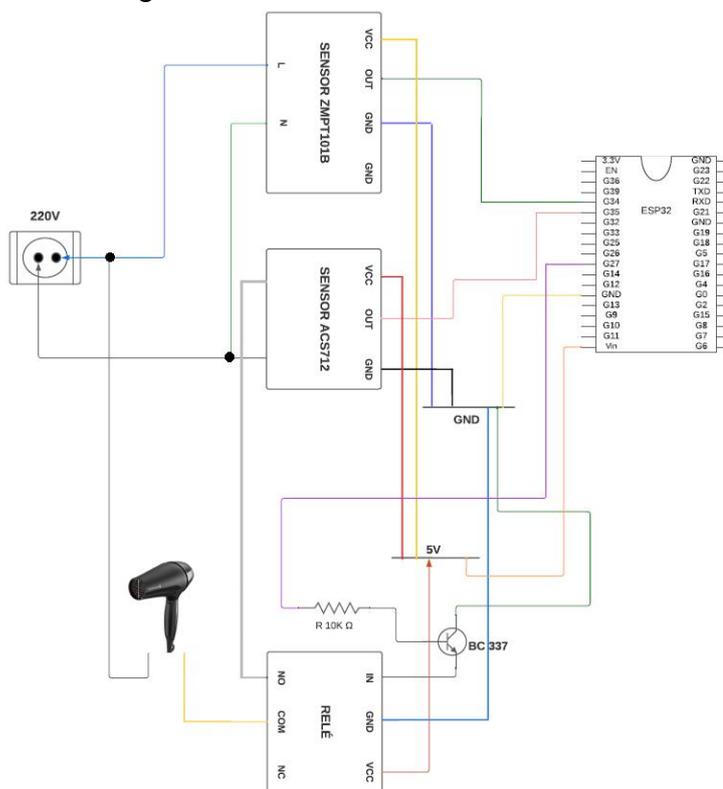
Na Figura 44 é exibida uma imagem do hardware e aplicativo utilizado para o controle e monitoramento do consumo de energia elétrica. A parte do hardware foi representado por meio do circuito na Figura 45 e foi montado com o propósito de medir a tensão da rede elétrica e a corrente do aparelho eletroeletrônico, a fim de calcular o consumo de energia elétrica. O custo total do hardware foi de R\$ 175,92.

Figura 44 – Hardware e aplicativo para controle e monitoramento do consumo de energia elétrica.



Fonte: Elaborado pelo autor.

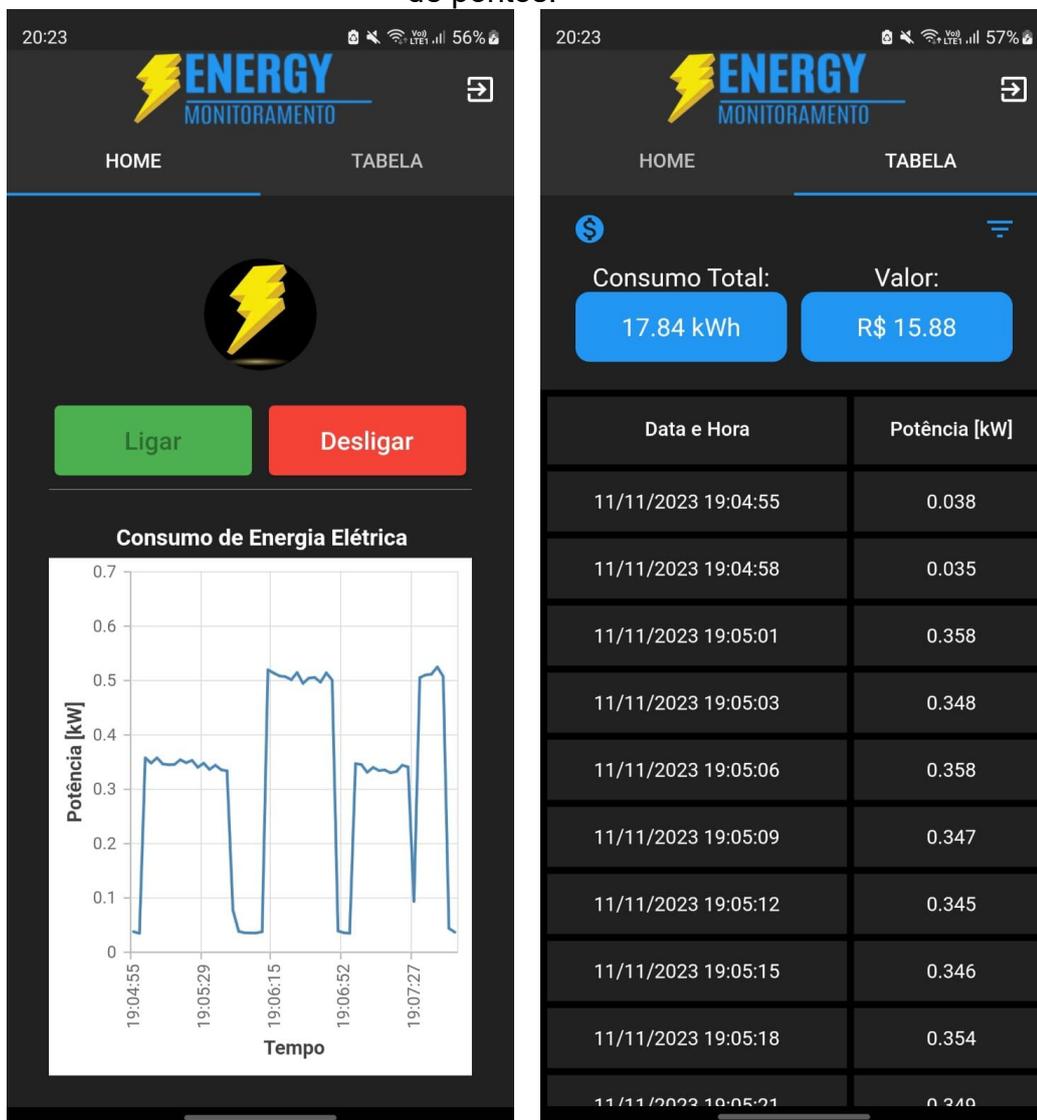
Figura 45 - Circuito para medir a tensão e corrente e poder desligar e ligar um secador de cabelo.



Fonte: Elaborado pelo autor.

Conseqüentemente, realizou-se a medição do consumo em quilowatt-hora (kWh), utilizando um secador de cabelo como exemplo. Os valores medidos, em 11 de novembro de 2023, foram exibidos no aplicativo por meio de um gráfico na Figura 46 (a), juntamente com informações sobre o consumo total e o custo em reais, conforme exibido na Figura 46 (b).

Figura 46 – Teste utilizando o secador de cabelo: (a) Gráfico; (b) tabela de pontos.

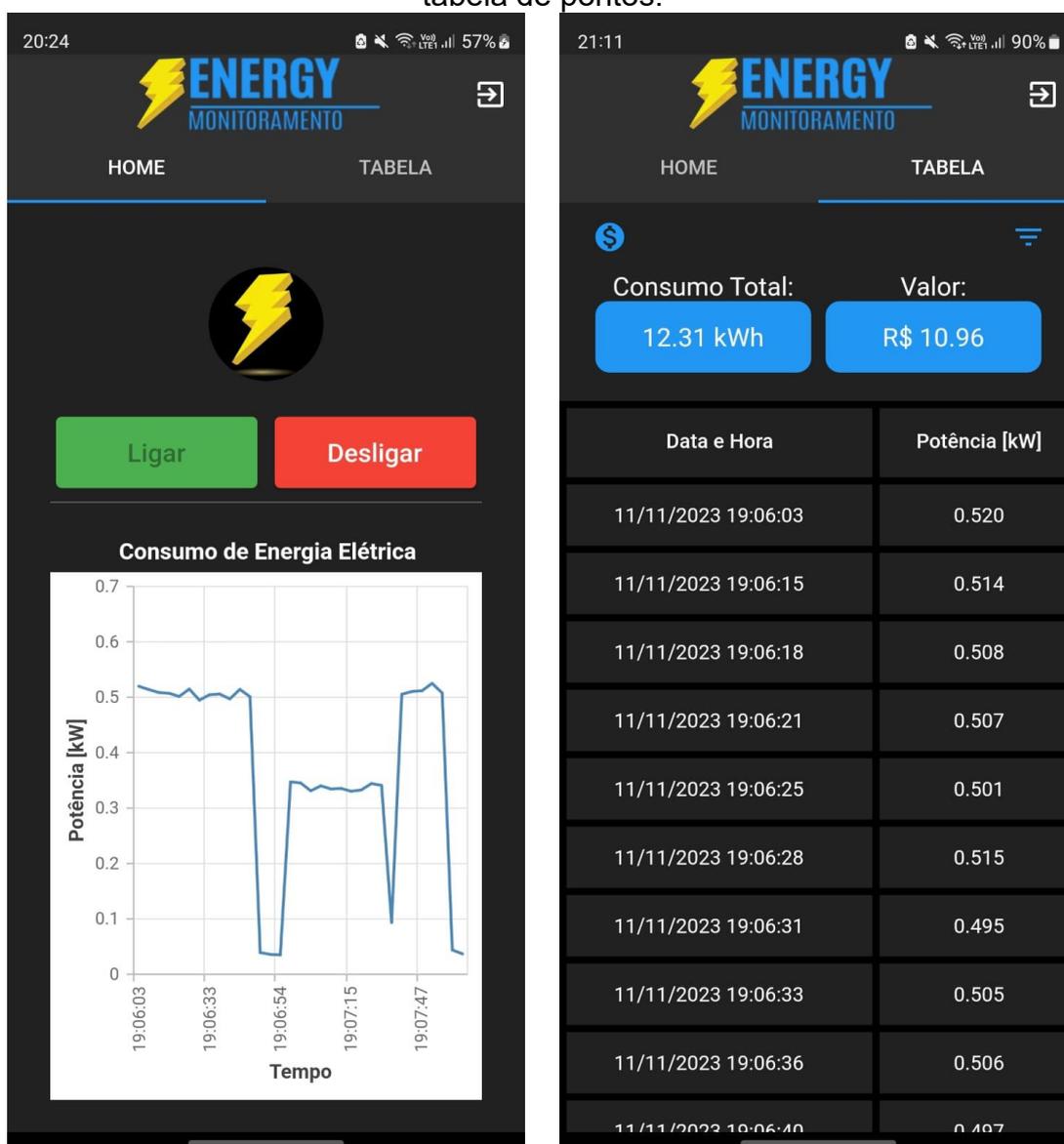


Fonte: Elaborado pelo autor.

Na Figura 46 (a) é exibido o gráfico referente ao consumo de energia elétrica do secador de cabelo, e por meio dele é possível visualizar que no eixo X (Tempo) em 19:04:55 o secador estava desligado e logo foi ligado na potência mínima (*Low*), atingindo uma potência inicial de 0,358 kW em 19:05:01, conforme a tabela na Figura 46 (b). Enquanto o secador estava ligado, a sua potência também foi ajustada para a máxima (*High*), podendo ser visualizado esse comportamento no gráfico, no tempo 19:06:15.

Para demonstrar a funcionalidade do filtro de data e hora presente no aplicativo, foi configurado a data de 11/11/2023 e tempo de aproximadamente 19:06 até um tempo de aproximadamente 19:08. Nas Figuras 47 (a) e (b) são apresentados o gráfico e a tabela com esse filtro. Vale ressaltar que, por conta do filtro aplicado na aba 'TABELA' do aplicativo, os valores mudaram na parte 'Consumo Total' e 'Valor', sendo atualizados com consumo total de 12.31 kWh e um valor em reais de R\$ 10.96.

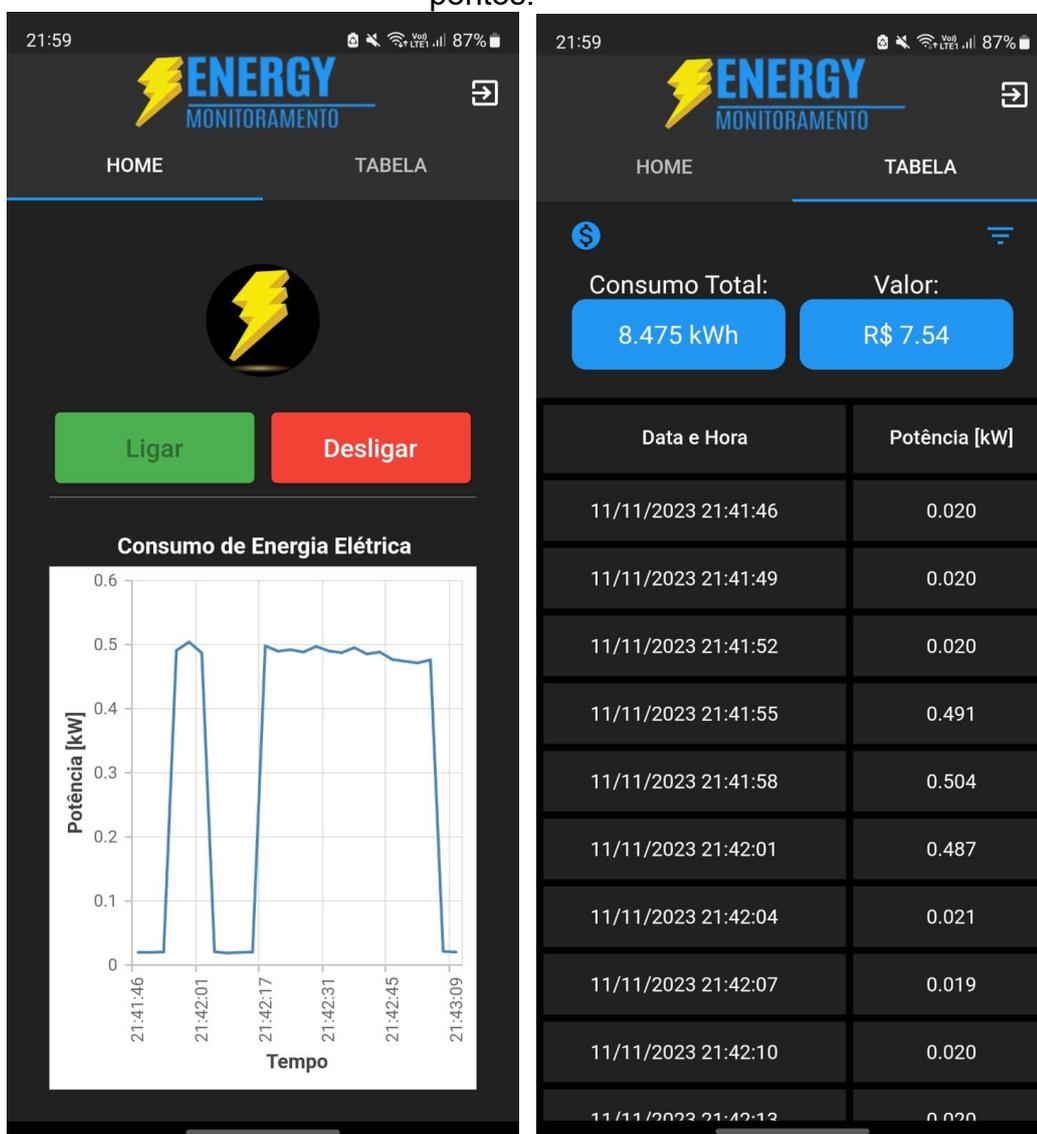
Figura 47 – Teste aplicando o filtro nos dados do secador: (a) Gráfico; (b) tabela de pontos.



Fonte: Elaborado pelo autor.

Além do teste realizado com o secador de cabelo, também foi utilizado um ferro de passar. Na Figura 48 (a) é exibido o gráfico correspondente ao consumo de energia elétrica. Observa-se que o ferro, inicialmente desligado, foi ligado por um curto período, desligado e, posteriormente, mantido ligado continuamente de um tempo de aproximadamente 21:42:17 até um tempo de aproximadamente 21:43:09, com uma potência variando entre 0.497 e 0.477. Na Figura 48 (b) na aba 'TABELA' é exibido os dados relacionados ao ferro de passar.

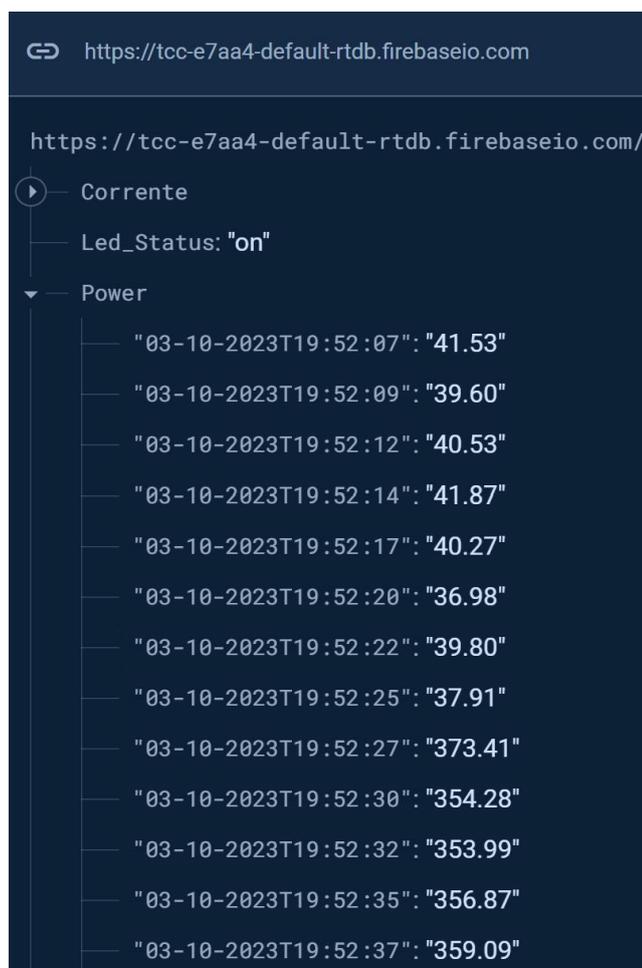
Figura 48 – Teste utilizando o ferro de passar: (a) Gráfico; (b) tabela de pontos.



Fonte: Elaborado pelo autor.

Vale ressaltar que, o banco de dados Firebase armazenou a data, o horário e a respectiva potência em Watts, obtidos por meio da ESP32 e dos sensores de corrente e tensão, conforme Figura 49. Já em relação a forma como os dados da potência em Watts são armazenados, eles são acompanhados por informação de data e hora, que seguem um padrão de 'dia-mês-anoThora:minuto:segundo', sendo que a letra 'T' foi utilizada como um separador entre a data e a hora, de forma a facilitar a obtenção dessas duas informações por parte do aplicativo.

Figura 49 – Valores de potência calculada de um secador de cabelo por meio dos sensores de corrente e tensão.



Fonte: Elaborado pelo autor.

Um exemplo de como o aplicativo separa as informações provenientes do Firebase, pode ser visto na Figura 50, na função 'formatador', cujo objetivo é formatar a parte da data e utilizar para a primeira coluna da tabela.

Mais especificamente, sobre essa função, ela recebe uma lista de strings representando datas e horas, juntamente com duas datas opcionais, alterDataIni e alterDataFim, sendo que essas datas opcionais só serão utilizadas se o usuário utilizou o filtro de data e hora do aplicativo. A função formata as datas de duas maneiras diferentes:

- Se o filtro foi utilizado pelo usuário, a função converte as strings para objetos DateTime, faz uma busca pelos dados considerando o filtro aplicado, substitui o símbolo de '-' por '/', e substitui o símbolo 'T' por um espaço;
- Se o filtro não foi aplicado pelo usuário, a função utiliza todos os dados armazenados pelo Firebase e substitui os mesmos símbolos indicados na maneira 1, nas strings originais.

O resultado é uma lista de strings formatadas, que é retornada pela função.

Figura 50 – Função para pegar dados para a primeira coluna da tabela.

```
//pega a data pra primeira coluna da tabela
List<String> formatador(List<String> dataHora, DateTime? alterDataIni, DateTime? alterDataFim) {
    List<String> formattedList = [];

    DateFormat dateTimeFormat = DateFormat('dd/MM/yyyy HH:mm:ss'); // Atualiza o formato com '/' ao invés de '-'

    if ((alterDataIni != DateTime(1999, 1, 1)) && (alterDataFim != DateTime(1999, 1, 1))) {
        for (String dateTimeString in dataHora) {
            List<String> components = dateTimeString.split('T');
            if (components.length == 2) {
                List<String> dateComponents = components[0].split('-');
                List<String> timeComponents = components[1].split(':');

                if (dateComponents.length == 3 && timeComponents.length == 3) {
                    int year = int.tryParse(dateComponents[2]) ?? 0;
                    int month = int.tryParse(dateComponents[1]) ?? 0;
                    int day = int.tryParse(dateComponents[0]) ?? 0;
                    int hour = int.tryParse(timeComponents[0]) ?? 0;
                    int minute = int.tryParse(timeComponents[1]) ?? 0;
                    int second = int.tryParse(timeComponents[2]) ?? 0;

                    DateTime dateTime = DateTime(year, month, day, hour, minute, second);

                    if (dateTime.isAfter(alterDataIni!) && dateTime.isBefore(alterDataFim!)) {
                        String formattedDateTime = dateTimeFormat.format(dateTime); // Formata date e time
                        formattedList.add(' $formattedDateTime');
                    }
                }
            }
        }
    } else {
        for (String dateTimeString in dataHora) {
            // Troca '-' por '/'
            String formattedDateTime = dateTimeString.replaceAll('-', '/');
            // Troca 'T' por espaço
            formattedDateTime = formattedDateTime.replaceAll('T', ' ');

            formattedList.add(' $formattedDateTime');
        }
    }

    return formattedList;
}
```

Fonte: Elaborado pelo autor.

## 6.2 Considerações sobre os resultados obtidos

Tanto o hardware quanto o aplicativo de controle e monitoramento do consumo de energia elétrica atenderam ao propósito deste trabalho. O hardware permitiu calcular a potência do aparelho eletroeletrônico monitorado, enquanto a aplicação exibiu os dados de consumo em kWh, por meio de um gráfico. Além disso, o aplicativo apresentou o consumo total junto com seu valor correspondente em reais e uma tabela com a data e a hora e o respectivo valor de potência calculado através dos valores obtidos pelos sensores de corrente e tensão.

Contudo, quanto a parte do hardware, observou-se um delay no armazenamento dos valores no Firebase, pois a velocidade do envio de informações depende tanto da velocidade da internet, quanto do tempo que a ESP32 gasta para enviar os valores lidos para o Firebase. Além disso, em relação ao comando de ligar ou desligar o aparelho eletroeletrônico, também foi observado um atraso para realizar a ação.

Vale ressaltar que, o APÊNDICE A - CÓDIGO DA ESP32 apresenta o código utilizado para o controle e monitoramento do consumo de energia elétrica. Já o código-fonte do aplicativo pode ser encontrado em <[https://github.com/3r1ckR1an/Energy\\_Monitor](https://github.com/3r1ckR1an/Energy_Monitor)>.

## 7 CONSIDERAÇÕES FINAIS

Segundo a EPE, o Brasil em 2022 foi um dos 10 países que mais consumiu energia no mundo, sendo que esse consumo tem aumentado ao longo dos anos (EPE, 2022). Neste trabalho algumas questões foram expostas referente ao consumo energético no geral, dentre elas sobre o consumo de energia em relação ao Brasil, e quanto a maneiras de alcançar uma maior eficiência energética no geral.

O objetivo deste trabalho foi monitorar o consumo de energia elétrica dos eletroeletrônicos e controlar (ligar ou desligar) os dispositivos de uma residência. Para obter conhecimento a fim de cumprir com esse objetivo, foi feito uma análise de trabalhos relacionados com o tema proposto, os quais desenvolveram protótipos de: monitoramento de consumo de energia elétrica, que apresentaram gráficos com valores referentes ao gasto energético mensal, e/ou protótipos que permitiram aos usuários controlar eletroeletrônicos. Sendo que essas opções foram reunidas por meio de aplicativos *mobile* ou plataformas *web*.

O objetivo do hardware deste trabalho foi concluído por meio da utilização de um microcontrolador, um sensor de corrente, um sensor de tensão, um relé, e por meio do código desenvolvido na plataforma Arduino IDE. Dessa forma, por meio dos testes unitários realizados, foram obtidos valores de corrente e tensão de um aparelho eletroeletrônico, como um secador de cabelo e um ferro de passar roupas por exemplo.

Em relação aos valores provenientes dos eletroeletrônicos testados, estes foram adquiridos pelos sensores e enviados para o microcontrolador, que por meio da opção Monitor Serial da plataforma Arduino foi possível visualizar o consumo de corrente e tensão do aparelho. Além disso, os valores enviados para o microcontrolador posteriormente foram enviados para o Firebase.

O objetivo de software deste trabalho foi concluído por meio do desenvolvimento de um aplicativo Android que fez a leitura dos valores gravados no banco de dados do Firebase e posteriormente calculando o consumo em kWh.

Com isso mostrou o gráfico referente ao consumo de energia elétrica do aparelho eletroeletrônico monitorado, apresentou uma tabela com os valores de potência em kW medidos a cada leitura, o consumo total em kWh e o respectivo preço em reais. Além disso, outra funcionalidade do aplicativo foi o de controlar o estado do aparelho eletroeletrônico, podendo ligá-lo ou desligá-lo, através dos botões 'Ligar' e 'Desligar'.

Desta forma, foi possível controlar e monitorar o consumo de energia elétrica do aparelho conectado a uma tomada, possibilitando uma automação e consequentemente um maior controle em relação a gastos com energia elétrica. Logo, este trabalho permitiu saber o consumo de energia elétrica de um aparelho em uma residência, e o preço deste consumo em reais.

### **7.1 Sugestões para trabalhos futuros**

Algumas considerações a serem feitas sobre as limitações deste trabalho são que, por conta da utilização de uma protoboard, mais especificamente em relação ao seu tamanho, o espaço ocupado pelo protótipo desenvolvido neste trabalho é maior do que deveria, portanto, a fim de possibilitar uma maior facilidade e simplicidade na instalação do mesmo, será benéfico o desenvolvimento de um módulo único do hardware.

Além disso, será benéfico permitir que tanto os dados do gráfico quanto os dados da tabela possam ser exportados para formato xlsx, para possibilitar que o usuário possa compartilhar essas informações com outros usuários de maneira mais simples.

Outra questão a ser comentada é o fato do gráfico do aplicativo mobile só ser atualizado com informações provenientes do banco de dados, caso o usuário mova da aba 'HOME' para a aba 'TABELA' ou vice versa. Portanto, para proporcionar uma maior comodidade ao usuário, será benéfico alterar o gráfico para que ele atualize seus dados de forma dinâmica.

Vale ressaltar que também será pertinente realizar uma integração do protótipo deste trabalho com a Alexa, assistente virtual desenvolvida pela

Amazon, de forma a adicionar comandos na mesma para informar o consumo de potência do eletroeletrônico monitorado e ligar ou desligar o mesmo.

## 8 REFERÊNCIAS

Android Studio. **Conhecer o Android Studio**. Disponível em: <https://developer.android.com/studio/intro?hl=pt-br>. Acesso em: 8 abr. 2023.

ARAUJO, Warley Monteiro; CAVALCANTE, Maxwell Machado; SILVA, Rogério Oliveira da. **VISÃO GERAL SOBRE MICROCONTROLADORES E PROTOTIPAGEM COM ARDUINO**. *Revista Tecnologias em Projeção*, Brasília, v. 10, n. 1, p. 36-46, 2019.

ARDUINO. **Arduino Integrated Development Environment (IDE) v1**. Disponível em: <https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics#toc4>. Acesso em: 8 abr. 2023.

CHUNG, Eui-Young. **Software Approaches for Energy Efficient System Design**. [S. L.]: Vdm Verlag, 2002. 144 p. Disponível em: <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjlkq2ZvsL6AhVBpZUCHfwtCmEQFnoECAcQAQ&url=https%3A%2F%2Fsi2.epfl.ch%2F~demichel%2Fgraduates%2Ftheses%2Fey.pdf&usg=AOvVaw0r353OWPjV1Ttu5OhxYfrl>. Acesso em: 2 out. 2022.

DART. **Dart overview**. Disponível em: <https://dart.dev/overview>. Acesso em: 8 abr. 2023.

EPE. **Balanco Energético Nacional**. 2022. Disponível em: <https://www.epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/PublicacoesArquivos/publicacao-675/topico-638/BEN2022.pdf>. Acesso em: 11 mar. 2023.

EPE. **Anuário Estatístico de Energia Elétrica**. 2022. Disponível em: <http://shinyepe.brazilsouth.cloudapp.azure.com:3838/anuario-livro/>. Acesso em: 11 mar. 2023.

ESPRESSIF. **ESP 32 Series: Datasheet**. 2023. Disponível em: [https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf). Acesso em: 02 abr. 2023.

FIREBASE. **Firestore Realtime Database**. 2023. Disponível em: <https://firebase.google.com/docs/database?hl=pt-br>. Acesso em: 08 abr. 2023a.

FIREBASE. **Adicionar o Firebase ao projeto para Android**. 2023. Disponível em: <https://firebase.google.com/docs/android/setup?hl=pt-br>. Acesso em: 08 abr. 2023b.

HENRIQUE, Sinezio. **Desenvolvimento de software para controle de dispositivos elétricos via Bluetooth e comando de voz**. 2016. 85 f. TCC (Graduação) – Curso de Engenharia Elétrica, Universidade Estadual Paulista “Júlio de Mesquita Filho”, Guaratinguetá, 2016.

KYAS, Othmar. **How to Smart Home**. [S. L.]: Key Concept Press, 2017. 432 p. Disponível em: <https://www.pdfdrive.com/how-to-smart-home-a-step-by-step-guide-for-smart-homes-building-automation-5th-edition-e199816422.html>. Acesso em: 2 out. 2022.

LIMA, Diogo Sebastião Silva de. **Sistema de Automação residencial por comandos de voz utilizando Raspberry e Aplicação Móvel**. 2018. 61 f. TCC (Graduação) - Curso de Engenharia de Computação, Instituto Federal Minas Gerais - Campus Bambuí, Bambuí, 2018.

MASSARO, Alessandro; STARACE, Giuseppe. **Advanced and Complex Energy Systems Monitoring and Control: a review on available technologies and their application criteria**. *Sensors*, [S.L.], v. 22, n. 13, p. 1-22, 29 jun. 2022. MDPI AG. <http://dx.doi.org/10.3390/s22134929>. Disponível em: <https://www.mdpi.com/1424-8220/22/13/4929>. Acesso em: 3 out. 2022.

MARTINS, Josias D. **Lendo sensor de corrente SCT100 no ESP com emonlib**. [S.L.]: Bobsien, 2022. 1 vídeo (13 min). Disponível em: [https://www.youtube.com/watch?v=V\\_K9ju9zQC8&ab\\_channel=Bobsien](https://www.youtube.com/watch?v=V_K9ju9zQC8&ab_channel=Bobsien). Acesso em: 16 abr. 2023.

MERÇON, Victor de Abreu. **Sistema de controle de iluminação e monitoramento de consumo elétrico residencial via aplicativo**. 2022. 100 f. TCC (Graduação) - Curso de Engenharia de Controle e Automação, Universidade Federal de Ouro Preto, Ouro Preto, 2022.

MERCADO LIVRE. **Placa Esp32 Pino Soldado Wifi Bluetooth Com Esp32-wroom-32**. Disponível em: [https://produto.mercadolivre.com.br/MLB-3282495240-placa-esp32-pino-soldado-wifi-bluetooth-com-esp32-wroom-32-\\_JM#is\\_advertising=true&position=2&search\\_layout=grid&type=pad&tracking\\_id=4b03f3c9-9316-4a48-bc4e-b3190d4c7d6a&is\\_advertising=true&ad\\_domain=VQCATCORE\\_LST&ad\\_position=2&ad\\_click\\_id=NGFIZWM0ZGItMGU3Ni00YTAyLTg3ZTIItNGUwMDRiINTJhNDIi](https://produto.mercadolivre.com.br/MLB-3282495240-placa-esp32-pino-soldado-wifi-bluetooth-com-esp32-wroom-32-_JM#is_advertising=true&position=2&search_layout=grid&type=pad&tracking_id=4b03f3c9-9316-4a48-bc4e-b3190d4c7d6a&is_advertising=true&ad_domain=VQCATCORE_LST&ad_position=2&ad_click_id=NGFIZWM0ZGItMGU3Ni00YTAyLTg3ZTIItNGUwMDRiINTJhNDIi). Acesso em: 02 abr. 2023a.

MERCADO LIVRE. **Sensor de Tensão Zmpt101b, 0 A 250 V Para Arduino**. Disponível em: [https://produto.mercadolivre.com.br/MLB-1761197143-sensor-de-tenso-zmpt101b-0-a-250-v-para-arduino-\\_JM?quantity=1](https://produto.mercadolivre.com.br/MLB-1761197143-sensor-de-tenso-zmpt101b-0-a-250-v-para-arduino-_JM?quantity=1). Acesso em: 08 abr. 2023b.

MERCADO LIVRE. **Sensor de Corrente Acs712 30A**. Disponível em: [https://produto.mercadolivre.com.br/MLB-1867352597-sensor-de-corrente-acs712-30a-\\_JM?quantity=1](https://produto.mercadolivre.com.br/MLB-1867352597-sensor-de-corrente-acs712-30a-_JM?quantity=1). Acesso em: 08 abr. 2023c.

MERCADO LIVRE. **Módulo Relé 1 Canal 5v/10A Led Indicador Arduino**. Disponível em: [https://produto.mercadolivre.com.br/MLB-1640327248-modulo-rele-1-canal-5v10a-led-indicador-arduino-\\_JM?quantity=1](https://produto.mercadolivre.com.br/MLB-1640327248-modulo-rele-1-canal-5v10a-led-indicador-arduino-_JM?quantity=1). Acesso em: 08 abr. 2023d.

OLIVEIRA, Isabella Ferreira de. **Desenvolvimento de um sistema de automação residencial baseado em IOT para controle e monitoramento de dispositivos elétricos**. 2019. 71 f. TCC (Graduação) - Curso de Engenharia de Controle e Automação, Universidade Federal de Ouro Preto, Ouro Preto, 2019.

PIRES, Laryssa Salustiana de Oliveira. **Desenvolvimento de um aplicativo Android para monitoramento do consumo de energia elétrica de uma residência**. 2021. 70 f. TCC (Graduação) - Curso de Engenharia de Computação, Pontifícia Universidade Católica de Goiás, Goiânia, 2021. Disponível em: <https://repositorio.pucgoias.edu.br/jspui/handle/123456789/3092>. Acesso em: 25 mar. 2023.

RASHID, Haroon; SINGH, Pushpendra; STANKOVIC, Vladimir; STANKOVIC, Lina. **Can non-intrusive load monitoring be used for identifying an appliance's anomalous behavior? *Applied Energy***, [S.L.], v. 238, p. 796-805, mar. 2019. Elsevier BV. <http://dx.doi.org/10.1016/j.apenergy.2019.01.061>.

RIO, Larissa Souto del; SILVA, João Octávio Barros Silva; AZEVEDO, Marcelo da Silva de; PEREIRA, Éder Paulo; FISCHER, Ivania Aline; MEDINA, Roseclea Duarte. **PROPOSTA DE AMBIENTES INTELIGENTES IOT SOB A ÓTICA DA EFICIÊNCIA ENERGÉTICA. A Produção do Conhecimento nas Ciências Exatas e da Terra 2**, [S.L.], p. 185-193, 4 abr. 2019. Atena Editora. <http://dx.doi.org/10.22533/at.ed.39519040418>.

ROBOCRAZE. **How to connect ZMPT101B to Arduino**. 2023. Disponível em: <https://robocraze.com/blogs/post/how-to-connect-zmpt101b-to-arduino#:~:text=2.-,How%20does%20ZMPT101B%20voltage%20sensor%20work%3F,to%20generate%20an%20analog%20output>. Acesso em: 21 nov. 2023.

RUANO, Antonio; HERNANDEZ, Alvaro; UREÑA, Jesus; RUANO, Maria; GARCIA, Juan. **NILM Techniques for Intelligent Home Energy Management and Ambient Assisted Living: a review. *Energies***, [S.L.], v. 12, n. 11, p. 2203, 10 jun. 2019. MDPI AG. <http://dx.doi.org/10.3390/en12112203>. Disponível em: <https://doi.org/10.3390/en12112203>. Acesso em: 9 set. 2022.

VANDOME, Nick. **Smart Homes**. [S. L.]: **In Easy Steps**, 2018. 293 p.  
Disponível em: <https://www.pdfdrive.com/smart-homes-in-easy-steps-master-smart-technology-for-your-home-e195285588.html>. Acesso em: 2 out. 2022.

WAZLAWICK, Raul Sidnei. **Metodologia de Pesquisa para Ciência da Computação**. 6. ed. [S. L.]: Elsevier, 2009. 124 p.

## APÊNDICE A – CÓDIGO DA ESP32

```
#include <WiFi.h>

#include <FirebaseESP32.h>
#include "EmonLib.h"
#include <NTPClient.h>
#include <WiFiUdp.h>

// Configurações de Wi-Fi e Firebase
#define WIFI_SSID "S20"
#define WIFI_PASSWORD "12345678"
#define API_KEY ""
#define DATABASE_URL ""

// Objeto para medir a energia usando a biblioteca EmonLib
EnergyMonitor emon1;

// Contador para atraso inicial
int i = 0;

// Objetos para interação com o Firebase
FirebaseData firebaseData;
FirebaseAuth auth;
FirebaseConfig config;

// Configuração do cliente NTP para obter a hora
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP);

// Função de configuração
void setup() {
  Serial.begin(115200);
  delay(1000);

  // Configuração de conexão Wi-Fi e configuração do sensor de energia
  conexao();
  emon1.voltage(34, 115.0, 1.7);
  emon1.current(35, 1.7);
  pinMode(27, OUTPUT); // Configura o pino GPIO27 como saída
}

// Loop principal
void loop() {
  // Funções para controlar a lâmpada e medir o consumo de energia
  lampada();
}
```

```

    consumo();
}

// Função para medir o consumo de energia e enviar dados para o Firebase
void consumo(){
    String power = "Power/\\" + tempo() + "\\";
    String current = "Corrente/\\" + tempo() + "\\";
    String voltage = "Voltagem/\\" + tempo() + "\\";

    // Calcula a corrente eficaz (Irms) e a voltagem eficaz (Vrms)
    emon1.calcVI(20, 1000);
    double Irms = emon1.calcIrms(1480);
    float voltagem = emon1.Vrms;

    Serial.println(Irms);
    Serial.println(voltagem);

    // Envio de dados para o Firebase após as primeiras 5 leituras para
    evitar dados iniciais imprecisos
    if (i > 5) {
        Firebase.setString(firebaseData, power, String(Irms * voltagem));
        Firebase.setString(firebaseData, current, String(Irms));
        Firebase.setString(firebaseData, voltage, String(voltagem));
    }
    else {
        i = i + 1;
    }
}

// Função para controlar o estado da lâmpada com base nas instruções do
// Firebase
void lampada(){
    Firebase.getString(firebaseData, "Led_Status");

    if (firebaseData.stringData() == "on") {
        Serial.println("Led ON");
        digitalWrite(27, HIGH); // Liga a GPIO27
    }
    else if (firebaseData.stringData() == "off") {
        Serial.println("Led OFF");
        digitalWrite(27, LOW); // Desliga a GPIO27
    }
}

// Função para realizar a conexão Wi-Fi e configurar o Firebase
void conexao(){
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
}

```

```

Serial.print("Conectando ao ");
Serial.print(WIFI_SSID);

// Aguarda a conexão Wi-Fi
while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  delay(500);
}

// Inicia o cliente NTP para obter a hora
timeClient.begin();
timeClient.setTimeOffset(-10800);

Serial.println();
Serial.print("Conectado ao ");
Serial.println(WIFI_SSID);

// Configuração da chave API e URL do Firebase
config.api_key = API_KEY;
config.database_url = DATABASE_URL;

// Habilita reconexão automática com o Wi-Fi
Firebase.reconnectWiFi(true);

// Inicia a autenticação e configuração do Firebase
Firebase.signUp(&config, &auth, "", "");
Firebase.begin(&config, &auth);
}

// Função para obter a data e hora formatada
String tempo(){
  timeClient.forceUpdate();
  String DateTime = timeClient.getFormattedTime();

  time_t rawtime = timeClient.getEpochTime();
  struct tm * ti;
  ti = localtime (&rawtime);

  uint16_t year = ti->tm_year + 1900;
  String yearStr = String(year);

  uint8_t month = ti->tm_mon + 1;
  String monthStr = month < 10 ? "0" + String(month) : String(month);

  uint8_t day = ti->tm_mday;
  String dayStr = day < 10 ? "0" + String(day) : String(day);
}

```

```
DateTime = dayStr + "-" + monthStr + "-" + yearStr + "T" + DateTime;  
Serial.println(DateTime);  
  
return DateTime;  
}
```

## APÊNDICE B – TERMO DE AUTORIZAÇÃO DE PUBLICAÇÃO DE PRODUÇÃO ACADÊMICA



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS  
GABINETE DO REITOR

Av. Universitária, 1089 • Setor Universitário  
Caixa Postal 06 • CEP: 74905-010  
Goiânia • Goiás • Brasil  
Fone: (62) 3946.1000  
www.pucgoias.edu.br • reitoria@pucgoias.edu.br

RESOLUÇÃO nº 038/2020 – CEPE

ANEXO I

APÊNDICE ao TCC

### Termo de autorização de publicação de produção acadêmica

O(A) estudante Erick Rian Barbosa de Sousa  
do Curso de Engenharia de Computação, matrícula 20191003300039,  
telefone: 62 996510628 e-mail erickrian49@gmail.com,  
na qualidade de titular dos direitos autorais, em consonância com a Lei nº 9.610/98 (Lei dos Direitos do Autor), autoriza a Pontifícia Universidade Católica de Goiás (PUC Goiás) a disponibilizar o Trabalho de Conclusão de Curso intitulado Sistema para controle e monitoramento de energia elétrica residencial, gratuitamente, sem ressarcimento dos direitos autorais, por 5 (cinco) anos, conforme permissões do documento, em meio eletrônico, na rede mundial de computadores, no formato especificado (Texto(PDF); Imagem (GIF ou JPEG); Som (WAVE, MPEG, AIFF, SND); Vídeo (MPEG, MWV, AVI, QT); outros, específicos da área; para fins de leitura e/ou impressão pela internet, a título de divulgação da produção científica gerada nos cursos de graduação da PUC Goiás.

Goiânia, 13 de Dezembro de 2023.

Assinatura do autor: Erick Rian Barbosa de Sousa

Nome completo do autor: Erick Rian Barbosa de Sousa

Assinatura do professor-orientador: Ludmilla R. P. dos Santos

Nome completo do professor-orientador: Ludmilla Rian Pinheiro dos Santos