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

DESENVOLVIMENTO DE UM APLICATIVO ANDROID PARA MONITORAMENTO DO CONSUMO DE ENERGIA ELÉTRICA DE UMA RESIDÊNCIA

LARYSSA SALUSTIANA DE OLIVEIRA PIRES

GOIÂNIA
2021

LARYSSA SALUSTIANA DE OLIVEIRA PIRES

DESENVOLVIMENTO DE UM APLICATIVO ANDROID PARA MONITORAMENTO DO CONSUMO DE ENERGIA ELÉTRICA DE UMA RESIDÊNCIA

Trabalho de Conclusão de Curso apresentado à Escola Politécnica, 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

GOIÂNIA
2021

LARYSSA SALUSTIANA DE OLIVEIRA PIRES

DESENVOLVIMENTO DE UM APLICATIVO ANDROID PARA MONITORAMENTO DO CONSUMO DE ENERGIA ELÉTRICA DE UMA RESIDÊNCIA

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

_____
Profa. Ma. Ludmilla Reis Pinheiro dos Santos
Coordenador de Trabalho de Conclusão de Curso

Banca examinadora:

_____
Orientadora: Profa. Ma. Ludmilla Reis Pinheiro dos Santos

_____
Prof. Dr. José Luiz de Freitas Júnior

_____
Prof. Me. Rafael Leal Martins

GOIÂNIA
2021

A Deus pela minha vida, pelas oportunidades e por ter me abençoado todos os dias.

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

AGRADECIMENTOS

A Deus que me deu força e saúde para desenvolver este trabalho.

Aos meus pais Nancy e Walter 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.

Aos meus amigos Brenno e Henrique pelo companheirismo, pela união, pelas risadas e por acreditarem no meu trabalho.

“Não é o que o mundo reserva para você, mas o que você traz para o mundo.”

Anne com e

RESUMO

O objetivo do trabalho foi desenvolver um protótipo de monitoramento do consumo de energia elétrica em uma residência, apresentando os valores em um aplicativo Android, para permitir o controle e possibilitar um consumo consciente de energia elétrica. Para isso, foram feitas leituras de tensão e corrente em uma tomada, com aparelhos eletroeletrônicos conectados. Utilizou-se os módulos: (a) ZMPT101B, para medir a tensão da rede; (b) ACS712, para medir a corrente elétrica do aparelho; e (c) relé, para permitir ao usuário habilitar ou desabilitar a tomada através de um aplicativo Android, desenvolvido em linguagem de programação Kotlin. Para processar os dados foi usado um ESP32, sendo os dados armazenados no banco de dados Firebase. O aplicativo permitiu a apresentação desses dados no formato gráfico e de tabela, para que o usuário pudesse monitorar o consumo, além de apresentar o consumo total e o preço em reais.

Palavras-Chave: *Monitoramento do consumo de energia elétrica. Firebase. ESP32. Aplicativo Android.*

ABSTRACT

The objective of the project was to develop a prototype for monitoring the consumption of eletrical energy, presenting the values in an Android application, to allow control and enable a conscious consumption of eletrical energy. To that, voltage and current readings were taken at a socket, with eletrical and electronic devices connected. The modules were used: (ZMPT101B), to measure the network voltage; (b) ACS712, to measure the electrical current of the device; and (c) relay, to allow the user to enable or disable the socket through an Android application, developed in programming language Kotlin. To process the data an ESP32 was used, with the data being stored in the Firebase database. The application allowed the presentation of this data in graphic and table format, so that the user could monitor consumption, in addition to presenting total consumption and price in real.

Keywords: *Monitoring of electricity consumption. Firebase. ESP32. Android Application.*

LISTA DE FIGURAS

Figura 1 – Histórico de consumo. 23
Figura 2 – Resultado da atividade. 24
Figura 3 – Gráfico de consumo em um período específico. 25
Figura 4 – Relatório de consumo em um período específico. 26
Figura 5 – Gráficos de consumo geral. 26
Figura 6 – Planilha com o relatório dos dados enviados. 27
Figura 7 – Tabela que apresenta o equipamento e o consumo total. 28
Figura 8 – Interface do aplicativo. 29
Figura 9 – Comparação do consumo medido pelo protótipo e pela Copel. 30
Figura 10 – Diagrama de blocos do sistema proposto. 31
Figura 11 – ESP32. 32
Figura 12 – ACS712. 33
Figura 13 – Módulo ACS712. 34
Figura 14 – Sensor de Tensão ZMPT101B. 34
Figura 15 – Funcionamento de um relé. 35
Figura 16 – Módulo relé. 36
Figura 17 – Código para piscar o LED. 39
Figura 18 – Circuito para piscar o LED. 40
Figura 19 – Código para piscar a lâmpada com relé. 41
Figura 20 – Circuito para piscar a lâmpada com relé. 41
Figura 21 – Código para medir corrente de uma prancha. 42
Figura 22 – Circuito para medir corrente de uma prancha. 43
Figura 23 – Valores medidos pelo sensor de corrente. 43
Figura 24 – Código para medir tensão de uma tomada. 44
Figura 25 – Circuito para medir tensão de uma tomada. 45
Figura 26 – Teste dos valores medidos pelo sensor de tensão. 45
Figura 27 – Código para medir tensão e corrente de um ferro elétrico. 47
Figura 28 – Circuito para medir tensão e corrente de um ferro elétrico. 47
Figura 29 – Teste dos valores medidos pelos sensores de tensão e de corrente. 48
Figura 30 – Implementação do aplicativo. 49
Figura 31 – Código para piscar o LED pelo aplicativo usando o Firebase. 50
Figura 32 – Circuito para piscar o LED pelo aplicativo. 50
Figura 33 – Trecho de código para enviar o dado de ligar/desligar o LED para o Firebase. 51
Figura 34 – Resultado do aplicativo Android e do monitor serial. 52
Figura 35 – Resultado do Firebase. 52
Figura 36 – Resultado do comando de ligar o LED. 52
Figura 37 – Circuito de monitoramento do consumo de energia. 54
Figura 38 – Tela Home do aplicativo. 54
Figura 39 – Partes da tabela do dia 20 na tela Table do aplicativo. 56
Figura 40 – Resultado do botão “OFF” no Firebase. 56
Figura 41 – Partes das leituras da data e hora do dia 20 no Firebase. 57
Figura 42 – Partes das leituras do consumo de potência do dia 20 no Firebase. 57
Figura 43 – Partes da tabela do dia 21 na tela Table do aplicativo. 58
Figura 44 – Partes da tabela do dia 22 na tela Table do aplicativo. 59
Figura 45 – Partes da tabela do dia 23 na tela Table do aplicativo. 60

LISTA DE TABELAS

Tabela 1 – Características do ESP32. 33
Tabela 2 – Características do módulo ACS712. 34
Tabela 3 – Características do módulo ZMPT101B. 35

LISTA DE SIGLAS

A	Ampère
ABNT	Associação Brasileira de Normas Técnicas
AC	*Alternating Current*
ANEEL	Agência Nacional de Energia Elétrica
DC	*Direct Current*
EPE	Empresa de Pesquisa Energética
GWh	Gigawatt-hora
IDE	*Integrated Development Environment*
IEA	*International Energy Agency*
IoT	*Internet of Things*
ISO	*International Organization for Standardization*
JSON	*JavaScript Object Notation*
kWh	Quilowatt-hora
LED	*Light Emitting Diode*
MWh	Megawatt-hora
NBR	Norma Brasileira Registrada
ODS	Objetivos de Desenvolvimento Sustentável
PIB	Produto Interno Bruto
V	Volt
VA	Volt-Ampère
W	Watt
XML	*eXtensible Markup Language*

SUMÁRIO

1 INTRODUÇÃO **15**
1.1 Objetivos **16**
1.1.1 Objetivo geral 17
1.1.2 Objetivos específicos 17
1.2 Justificativa **17**
1.3 Métodos **18**
1.4 Organização do trabalho **18**

2 REFERENCIAL TEÓRICO **19**
2.1 Eficiência energética **19**
2.2 Monitoramento **20**
2.3 *Internet of Things* **20**
2.4 *Big Data* **21**
2.5 Trabalhos relacionados **22**
2.5.1 Sistema supervisório para monitoramento de energia elétrica residencial 22
2.5.2 Sistema de monitoramento remoto de consumo de energia elétrica em residências 24
2.5.3 Desenvolvimento de um protótipo de sistema de monitoramento de energia elétrica via internet: um estudo de caso no IFMG-Campus Bambuí 25
2.5.4 Monitoramento do consumo de energia elétrica e controle de equipamentos via aplicativo 27
2.5.5 Análise do consumo de energia em uma unidade consumidora via Internet das Coisas 29
2.5.6 Análise dos trabalhos relacionados 30

3 PROPOSTA DE SOLUÇÃO **31**
3.1 Diagrama de blocos **31**
3.2 Hardware **31**
3.2.1 Microcontrolador 32
3.2.2 Sensor de Corrente 33
3.2.3 Sensor de Tensão 34
3.2.4 Atuadores 35
3.3 Software **36**
3.3.1 Banco de dados 36

3.3.2 Linguagem de programação 37
3.3.3 Android Studio 37
3.3.4 Arduino Software IDE 38

4 DESENVOLVIMENTO 39
4.1 Montagem do protótipo 39
4.1.1 Testes unitários 39
4.1.2 Teste de integração 46
4.2 Elaboração do aplicativo 48
4.3 Comunicação do protótipo com o aplicativo 49

5 RESULTADOS OBTIDOS 53
5.1 Análise dos resultados obtidos 53
5.2 Considerações sobre os resultados obtidos 61

6 CONSIDERAÇÕES FINAIS 62
6.1 Sugestões para trabalhos futuros 62

REFERÊNCIAS 63

APÊNDICE A – CÓDIGO PARA O MONITORAMENTO DO CONSUMO DE ENERGIA 69
APÊNDICE B – PROTÓTIPO DE MONITORAMENTO DO CONSUMO DE ENERGIA 70
ANEXO A – TERMO DE AUTORIZAÇÃO DE PUBLICAÇÃO DE PRODUÇÃO ACADÊMICA 71

1 INTRODUÇÃO

A matriz elétrica brasileira é dividida da seguinte forma: (a) usina hidrelétrica, com 51,16%; (b) usina termelétrica, com 25,36%; (c) central geradora eólica, com 11,87%; (d) central geradora solar fotovoltaica, com 6,06%; (e) pequena central hidrelétrica, com 3,50%; (f) usina termonuclear, com 1,65%; e (g) central geradora hidrelétrica, com 0,40%. O sistema elétrico brasileiro é formado por quatro grupos: (a) geração, que é responsável pela produção de energia; (b) transmissão, que transporta a energia dos geradores até os centros consumidores; (c) distribuição, que entrega a energia aos consumidores; e (d) consumo, que utiliza a energia através dos aparelhos eletroeletrônicos (ANEEL, 2016a; SUPERINTENDÊNCIA DE CONCESSÕES E AUTORIZAÇÕES DE GERAÇÃO, 2020).

O Brasil apresentou em 2019 o consumo de 633033 Gigawatt-hora (GWh), sendo os setores que apresentaram maiores gastos: (a) industrial, com 274774 GWh; (b) residencial, com 156546 GWh; (c) comercial, com 118416 GWh; e (d) outros, com 83297 GWh. As regiões brasileiras apresentaram, neste mesmo ano, os seguintes consumos: (a) Sudeste e Centro-Oeste, com 377355 GWh; (b) Sul, com 103160 GWh; (c) Nordeste, com 92561 GWh; (d) Norte, com 58152 GWh; e (e) Sistemas Isolados, que interliga Manaus, Macapá e a margem esquerda do Amazonas, com 1805 GWh (EPE, 2019b).

De acordo com a ANEEL (2019), para estimar o custo de energia elétrica, relativo aos aparelhos, são considerados a energia gerada, o transporte da mesma, os encargos setoriais e os tributos. Para o período de 01 de julho de 2019 a 30 de junho de 2020, a tarifa é de R$ 107,86 por Megawatt-hora (MWh).

Desta forma, a Agência Nacional de Energia Elétrica (ANEEL) criou o Programa de Eficiência Energética que possibilita o "uso eficiente de energia elétrica em todos os setores da economia por meio de projetos que demonstrem a importância e a viabilidade econômica de melhoria da eficácia energética de equipamentos, processos e usos finais de energia" (ANEEL, 2016b).

A ANEEL criou em 2015 o Sistema de Bandeiras Tarifárias que indicam se haverá ou não aumento no valor da energia, com o objetivo de "repassar mensalmente ao consumidor, de forma mais transparente, os custos adicionais causados pela necessidade de acionamento de usinas termelétricas na geração de energia". As bandeiras são das cores verde, amarela e vermelha, sendo que a primeira não sofre nenhum acréscimo, a segunda acrescenta R$ 0,01343 para cada quilowatt-hora (kWh) consumido, já a terceira apresenta dois patamares: (a) patamar 1, tem o acréscimo de R$ 0,04169 para cada kWh consumido; e (b) patamar 2,

tem o acréscimo de R$ 0,0624 para cada kWh consumido (SUPERINTENDÊNCIA DE GESTÃO TARIFÁRIA, 2019).

Assim, o consumo de energia de forma consciente tem se tornado um tema relevante, sendo que as tecnologias atuais têm sido utilizadas para solucionar este problema. Com a Internet das Coisas, ou do inglês *Internet of Things* (IoT), tem-se "uma extensão da Internet atual, que proporciona aos objetos do dia a dia (quaisquer que sejam), mas com capacidade computacional e de comunicação, se conectarem à Internet" (SANTOS et al., 2016).

Para um consumo eficiente de energia foi introduzido o conceito de *Smart Cities*, ou cidades inteligentes, que está atrelado tanto as tecnologias quanto ao bem-estar social. Assim, a Urban Systems (2019) tem desenvolvido um estudo desde 2015 sobre cidades inteligentes, criando um *ranking* composto pelas seguintes bases: mobilidade, urbanismo, meio ambiente, energia, tecnologia e inovação, economia, educação, saúde, segurança, empreendedorismo e governança.

Portanto, para se ter uma cidade sustentável foi criado um conjunto de regras definidas pela ISO NBR 37120, criada em 2017, que apresenta critérios de medições e indicadores a fim de se medir uma cidade sustentável, em relação a serviços urbanos e a qualidade de vida. Em complemento a esta norma, em 2019 foi criada a ISO NBR 37122 que objetiva mensurar a inteligência das cidades, utilizando IoT para se medir a performance de determinadas áreas para atender às necessidades dos cidadãos (EXATI, 2019).

Juntamente com a IoT, para aprimorar o conhecimento de negócios tem-se o conceito de *Big Data,* que faz a coleta de dados para tomada de decisões. O *Big Data* está relacionado aos "avanços recentes relativos à massificação da utilização de recursos tecnológicos e da farta produção de dados" (RAUTENBERG; CARMO, 2019).

Assim, pode-se ter a supervisão e controle de elementos elétricos de forma eficiente e sustentável, este sistema é chamado de *Smart Grids*. O surgimento deste tipo de rede elétrica inteligente tem propiciado ao consumidor o acesso às informações de uso de energia elétrica (SILVA, 2019).

Desta forma, este trabalho pretende responder a seguinte questão de pesquisa:

- **É possível desenvolver um aplicativo que monitore o gasto de aparelhos eletroeletrônicos de uma residência para um consumo consciente de energia elétrica?**

1.1 Objetivos

Esta seção apresenta os objetivos geral e específicos deste trabalho.

1.1.1 Objetivo geral

Este trabalho objetiva monitorar o consumo de energia elétrica de aparelhos eletroeletrônicos de uma residência medindo os valores de corrente e tensão destes aparelhos. Os valores das medições são armazenados para calcular o preço por kWh, sendo apresentados os dados em um gráfico e uma tabela para que o consumidor possa visualizar os gastos e supervisionar o consumo de energia, visando a eficiência energética.

1.1.2 Objetivos específicos

- Receber como entrada no microcontrolador os valores de tensão e corrente de cada aparelho eletroeletrônico;
- Realizar as medições e calcular os dados recebidos por hora, para se ter o consumo de potência do aparelho por hora;
- Armazenar os dados das medições em um banco de dados;
- Apresentar o preço do consumo de energia de um aparelho eletroeletrônico;
- Desenvolver um aplicativo Android que apresente os dados de consumo do aparelho eletroeletrônico em um gráfico e uma tabela;
- Permitir que o usuário tenha o controle da tomada, tornando possível deixá-la habilitada ou desabilitada, através do aplicativo.

1.2 Justificativa

Este trabalho permite o monitoramento dos gastos de energia em uma residência, facilitando a supervisão dos gastos por meio de um aplicativo Android.

Como a IoT permite a conexão de sensores e microcontroladores com a Internet, tornando possível enviar e receber dados, pode ser apresentado com este trabalho uma funcionalidade para o usuário. O aplicativo desenvolvido permite ao usuário ter o controle dos gastos de energia, utilizando um banco de dados com os dados das medições, bem como calculando o consumo de energia.

1.3 Métodos

Inicialmente foi realizada uma revisão bibliográfica, com a utilização de livros, artigos, revistas e similares. Posteriormente, foi realizado um procedimento prático, sendo feita a montagem de hardware para realizar a medição dos valores de tensão e corrente dos aparelhos eletroeletrônicos, medindo os gastos de energia desses e assim, foi desenvolvido um aplicativo para monitorar o consumo de energia com base nestas medições.

1.4 Organização do trabalho

O trabalho foi dividido em 6 capítulos. Neste capítulo foram apresentados o tema, os objetivos, a justificativa e os métodos usados para o desenvolvimento do projeto.

O capítulo 2 apresenta os conceitos e as definições utilizados para a construção do projeto.

O capítulo 3 apresenta a proposta de solução e uma descrição tanto de *hardware* como de *software*.

O capítulo 4 apresenta o desenvolvimento do protótipo de monitoramento do consumo de energia elétrica, sendo feita a montagem do protótipo, realizando testes unitários e de integração e a elaboração do aplicativo, e posteriormente foi feita a integração do protótipo com o aplicativo.

O capítulo 5 mostra os resultados dos testes realizados além de apresentar uma análise dos resultados obtidos.

O capítulo 6 apresenta as considerações finais do projeto.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta os conceitos e definições de termos utilizados para o desenvolvimento deste projeto, além de apresentar os trabalhos relacionados.

2.1 Eficiência energética

De acordo com o Encontro Anual de Tecnologia da Informação (2018), a eficiência energética está relacionada com a redução do uso de energia na realização de uma tarefa. A Equação (1) apresenta o cálculo de consumo de energia, sendo que C = consumo de energia, em kWh; P = potência, em watt; e t = tempo, em horas.

$$C = \frac{P}{1000} * t \tag{1}$$

Além da eficiência energética ter como objetivo atender a demanda de energia evitando o desperdício, também contribui para não faltar energia, para a cobrança de tarifas conforme a lei e para a redução de impactos ambientais. Com isso, a Empresa de Pesquisa Energética (EPE) em conjunto com a *International Energy Agency* (IEA) faz o planejamento da eficiência energética seguindo ações estruturadas, como: (a) formação das bases de dados, inclui os potenciais e custos; (b) preparação de estratégias, incentiva a disseminação de eficiência energética; e (c) monitoramento, acompanha o progresso de indicadores (EPE, 2019a).

O consumo de energia está associado à economia do país. Entre 2001 e 2018, o Produto Interno Bruto (PIB) e a oferta interna de energia tiveram uma taxa média de crescimento de 2,3% ao ano. Desta forma, a intensidade energética é a relação entre a energia e o PIB, sendo que a menor intensidade energética caracteriza maior eficiência de energia. Esta medição da intensidade pode ser calculada de duas maneiras: (a) de acordo com a Oferta interna de energia, caracterizada como a Intensidade Primária, conforme demonstrado na Equação (2); e (b) de acordo com o Consumo final de energia, descrito como a Intensidade Final, conforme demonstrado na Equação (3) (EPE, 2019a):

$$\text{Intensidade Primária} = \frac{\text{Oferta interna de energia}}{\text{PIB}} \tag{2}$$

$$\text{Intensidade Final} = \frac{\text{Consumo final de energia}}{\text{PIB}} \quad (3)$$

onde: oferta interna de energia, em tonelada equivalente de petróleo (tep); consumo final de energia, em tep; intensidade (primária e final), em tep por trilhões de dólares; e PIB, em trilhões de dólares. 1 tep equivale a aproximadamente 42 gigajoules ou 11,7 MWh (ANEEL, 2020).

Desta forma, a Aneel criou o Programa de Eficiência de Energia que promove "a transformação do mercado de eficiência energética, estimulando o desenvolvimento de novas tecnologias e a criação de hábitos e práticas racionais de uso da energia elétrica". Com isso, visa a sustentabilidade e a disponibilidade do setor elétrico (REVISTA DE EFICIÊNCIA ENERGÉTICA, 2017).

2.2 Monitoramento

O monitoramento do consumo de energia está relacionado com a eficiência energética, pois faz o acompanhamento de cada equipamento monitorado em busca de avanço nos Objetivos de Desenvolvimento Sustentável (ODS). Como por exemplo, com o monitoramento do consumo de energia, verificou-se que o Brasil teve maior gasto com o uso de chuveiros elétricos, e isto pode ser melhorado utilizando o sistema de aquecimento solar (EPE, 2019a).

O conceito de monitoramento remoto está associado a sistemas embarcados e transmissão de dados estando presente em diversas áreas, como industrial e segurança. Uma das aplicações de monitoramento remoto pode ser observada na indústria de lacticínios, em que se tem os dados referentes às quebras mecânicas e elétricas. O planejamento preventivo de manutenção impede que paradas inesperadas ocorram, que evita o problema de perda de eficiência (SILVEIRA JUNIOR; MOURA, 2018; REIS, 2019).

O monitoramento do consumo de energia elétrica, utiliza a potência ativa consumida a ser monitorada, sendo possível enviar esses dados para se ter uma análise e serem exibidos em gráficos e tabelas para que o usuário possa visualizar o consumo (MOURA; MONTEIRO SOBRINHO, 2019).

2.3 *Internet of Things*

A IoT pode ser definida como um paradigma que possibilita a interconexão de diferentes objetos, com conexão a Internet, para otimizar serviços. Além disso, pode ser

usados os sensores dos *smartphones,* que são utilizados em grande número atualmente, para monitorar algum ambiente (MEDEIROS et al., 2018).

Sendo assim, pode ser utilizada para monitoramento e controle para que seja feita a coleta dados relacionados ao uso de energia de dispositivos, permitindo gerenciar automaticamente e acompanhar o desempenho em tempo real. A IoT permite analisar uma grande quantidade de dados, pois faz o processamento de grandes quantidades dados resultantes dos dispositivos com sensores e atuadores e envia estas informações para ferramentas que irão servir de base para tomada de decisões (MEDEIROS et al., 2018).

Com essa conectividade é necessário que os dados coletados sejam mantidos em sigilo e em segurança. Portanto, para o desenvolvimento de aplicativos utilizando o conceito de IoT há alguns desafios para manter a segurança, privacidade e confidencialidade dos dados. Contudo, a maioria das soluções de confidencialidade em IoT enfatiza a segurança e a autorização dos canais de comunicação (JAYARAMAN et al., 2017).

2.4 *Big Data*

Segundo Ferlin e Rezende (2019), o *Big Data* é "o termo que descreve o grande volume de dados – estruturados e não estruturados – que impactam nas tomadas de decisões". E pode ser aplicado às *Smart cities* para o estudo de um conjunto de dados volumoso, vez que utilizam as informações e dados coletados para gerir os recursos de uma cidade, fazendo com que ela se torne inteligente e sustentável.

Assim, o *Big Data* apresenta dimensões denominadas de "5 Vs", sendo elas: (a) Volume, refere a informação e a acessibilidade aos dados; (b) Variedade, trata das diversas fontes de informação e tipos de dados; (c) Velocidade, agilidade de tomada de decisão e de troca de informações; (d) Valor, refere ao volume de dados obtidos em análises, gestão e coleta de dados; e (e) Veracidade, trata da autenticidade dos dados (MAÇADA; CANARY, 2014; MENDONÇA; ANDRADE, 2019).

Pode ser observado que nas últimas duas décadas teve um aumento no volume de dados, de kilobytes para zettabytes e essa quantidade de dados leva a um maior armazenamento e processamento de dados. Desta forma, processar e armazenar grandes quantidades de dados tem se tornado um desafio, podendo ser utilizada a computação em nuvem para solucionar este problema. Porém, esse processo pode apresentar lentidão, vez que a velocidade de *upload* pode ser até 10 vezes mais lenta que a de *download* (VENKATRAM; GEETHA, 2017; CIPOLI, 2020).

2.5 Trabalhos relacionados

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

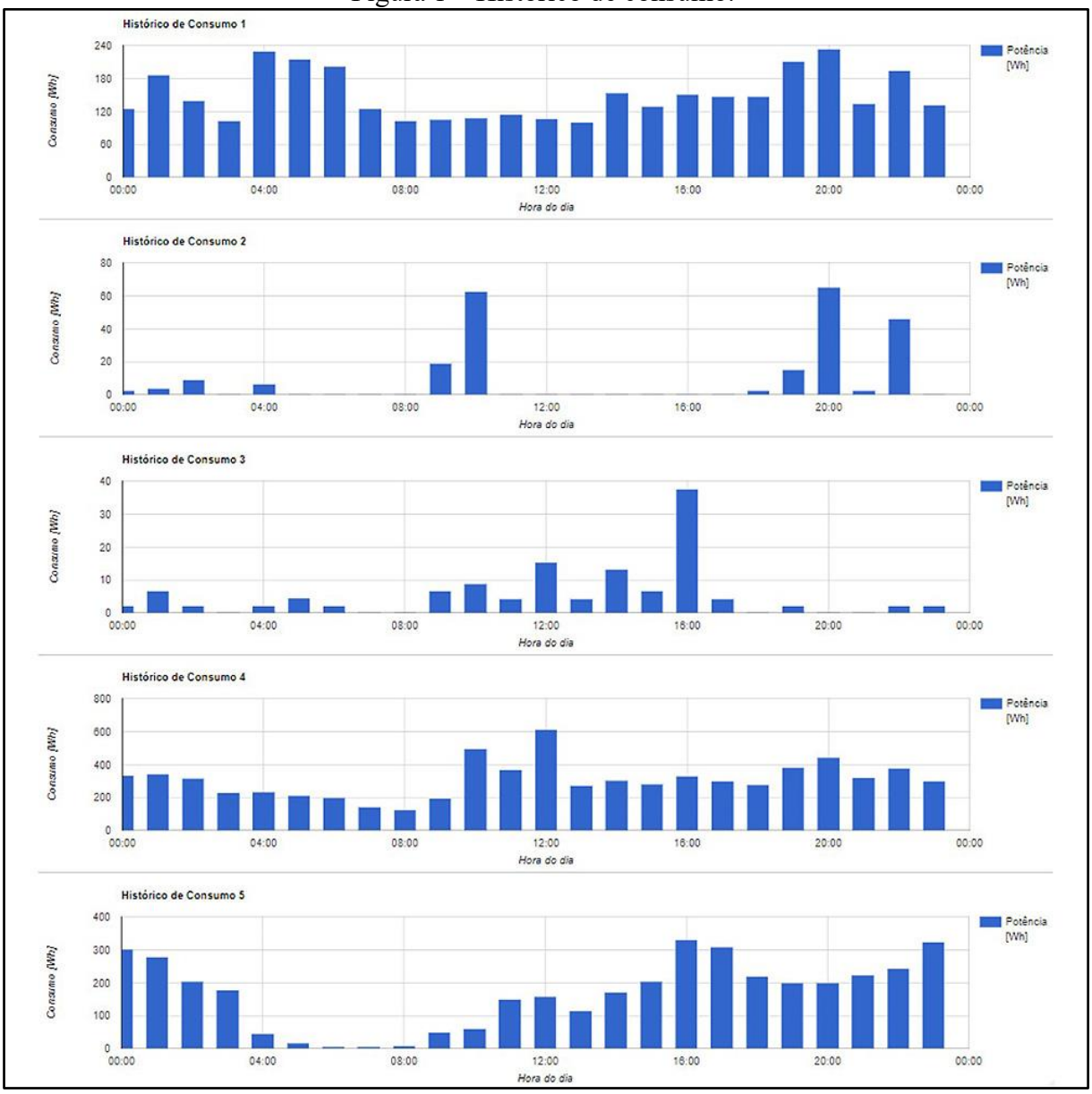
2.5.1 Sistema supervisório para monitoramento de energia elétrica residencial

O trabalho de sistema supervisório para monitoramento de energia elétrica residencial teve como objetivo desenvolver um sistema supervisório para monitorar o consumo de energia de uma residência com quadros elétricos de até 5 circuitos. Assim, o consumidor teve acesso a gráficos detalhados de consumo para analisar a conformidade com as tarifas adotadas. O sistema proposto foi de baixo custo e possibilitou ao usuário ter acesso aos dados através de qualquer dispositivo com suporte a navegação na internet para acessar a página *Web* (MAESTRELLI; NAPOLEÃO, 2018).

Para isso, foi utilizado o microcontrolador ESP32 que, através de um programa em linguagem C, fez a comunicação via internet com sensores de corrente, tensão. Foi utilizada a conexão *WiFi,* integrado no microcontrolador, e a cada 20 segundos eram enviados os dados processados que foram armazenados no banco de dados MySQL. Então, foi gerado um arquivo *Hypertext Prepocessor* (PHP) que através de uma plataforma *Web* demonstrou os resultados para o usuário (MAESTRELLI; NAPOLEÃO, 2018).

Assim, o teste realizado demonstrou que o usuário pôde ter acesso às informações das medições de tensão e corrente, e aos gráficos de consumo dos cinco circuitos, em um período específico. Sendo que: (a) Circuito 1, corresponde as tomadas da cozinha; (b) Circuito 2, iluminação; (c) Circuito 3, chuveiro elétrico; (d) Circuito 4, fase de entrada 1; e (e) Circuito 5, fase de entrada 2. Para demonstrar os resultados foi feito um teste do dia 27 de maio de 2018, conforme mostra o histórico de consumo da Figura 1 (MAESTRELLI; NAPOLEÃO, 2018).

Figura 1 – Histórico de consumo.



Fonte: Maestrelli; Napoleão, 2018.

Os gráficos apresentados na Figura 1 mostram o consumo, em Wh, de cada circuito e a partir destes dados foram calculados os valores para as tarifas convencional e branca. Além disso, foi demonstrado a comparação de custos do consumo entre as tarifas, apresentando qual tarifa foi mais vantajosa para determinado circuito. No teste realizado, se o usuário utilizasse a tarifa convencional teria um custo total de R$4,58 nos Circuitos de 1 a 3, sendo mais lucrativa do que a tarifa branca que foi de R$4,66. Da mesma forma nos Circuitos 4 e 5, a tarifa convencional foi mais vantajosa, com um custo total de R$9,21, enquanto a tarifa branca teve um custo total de R$9,59 (MAESTRELLI; NAPOLEÃO, 2018).

2.5.2 Sistema de monitoramento remoto de consumo de energia elétrica em residências

O trabalho de sistema de monitoramento remoto de consumo de energia elétrica em residências teve como objetivo desenvolver um sistema para medir a corrente elétrica, com isso foi calculado o consumo de energia em uma residência, mostrando em um aplicativo Android o custo total e os históricos de consumo. Para isso, foram utilizados sensores de medição de corrente invasivos ACS712 conectados em tomadas que fizeram a leitura das correntes, sendo que estes valores foram interpretados pelo Arduino Nano. Este microcontrolador, modelo ESP-01, possui *WiFi* integrado podendo ser utilizado como servidor, assim os dados foram enviados para um aplicativo Android (PRADO, 2018).

O teste apresentado no aplicativo mostrou que o usuário teve que inserir as informações de tarifa de energia elétrica, bandeira tarifária do mês e senha do aplicativo. Além disso, a aplicação recebeu os dados do servidor ESP-01 e foram feitos os cálculos que geraram a saída de kWh consumidos e o valor em reais, sendo o resultado apresentado na Figura 2 (PRADO, 2018).

Figura 2 – Resultado da atividade.

Fonte: Prado, 2018.

O resultado da leitura da corrente pelo sensor ACS712 apresentou uma diferença do valor real, portanto o cálculo do consumo, em kWh, e do gasto, em reais, teve um erro em relação ao valor real medido pela concessionária de energia. Mas a interface com o usuário foi intuitiva e mostrou os resultados do consumo de forma clara (PRADO, 2018).

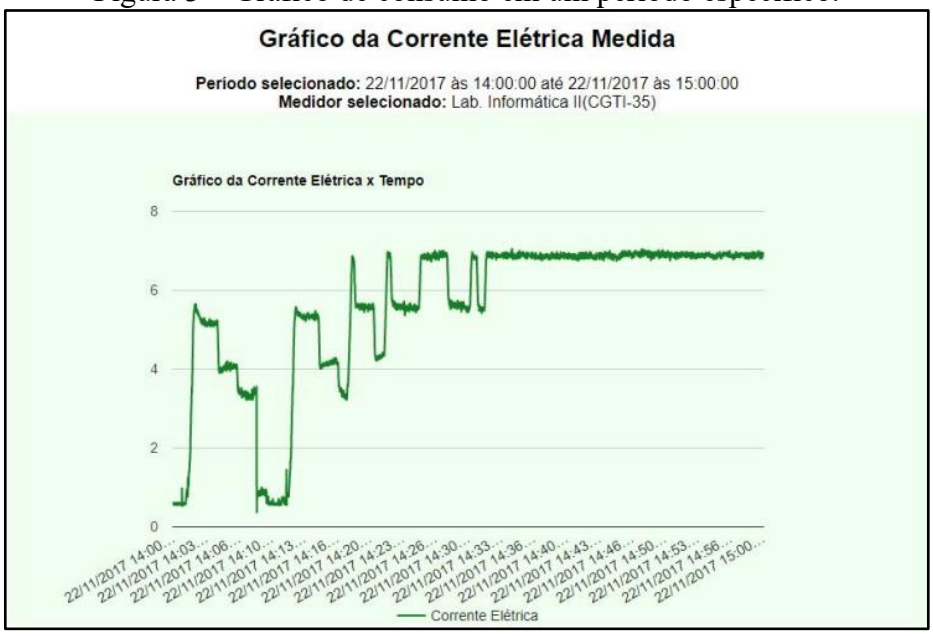
2.5.3 Desenvolvimento de um protótipo de sistema de monitoramento de energia elétrica via internet: um estudo de caso no IFMG-Campus Bambuí

O trabalho do desenvolvimento de um protótipo de sistema de monitoramento de energia elétrica via internet: um estudo de caso no IFMG-Campus Bambuí teve como objetivo desenvolver um sistema de monitoramento para reduzir gastos com o consumo de energia no Instituto Federal de Minas Gerais (IFMG), para isso foi feito o protótipo para monitorar o gasto do ar condicionado de um laboratório de informática. Foi utilizado Arduino, sensor de corrente ACS712 e *Ethernet Shield* W5100 para realizar as medições, estes dados foram enviados para uma aplicação *Web*, que utilizou a linguagem PHP (OLIVEIRA, 2017).

Como a medição foi feita a cada segundo, utilizou-se um Sistema de Gerenciamento de Banco de Dados (SGBD) para armazenamento dos dados e para gerenciar o banco de dados escolhido, o MySQL. Desta forma, foram desenvolvidas duas tabelas, uma que armazenou os dados de corrente elétrica, código do medidor, hora e data das medições, e a outra contendo o código de cada medidor e a descrição (OLIVEIRA, 2017).

Para visualizar o consumo, foi desenvolvido uma aplicação *Web* em que o usuário selecionou o medidor e teve a opção de visualizar o consumo por um período específico e teve acesso aos relatórios destes períodos, Figuras 3 e 4, e a outra forma é geral, conforme mostra a Figura 5 (OLIVEIRA, 2017).

Figura 3 – Gráfico de consumo em um período específico.



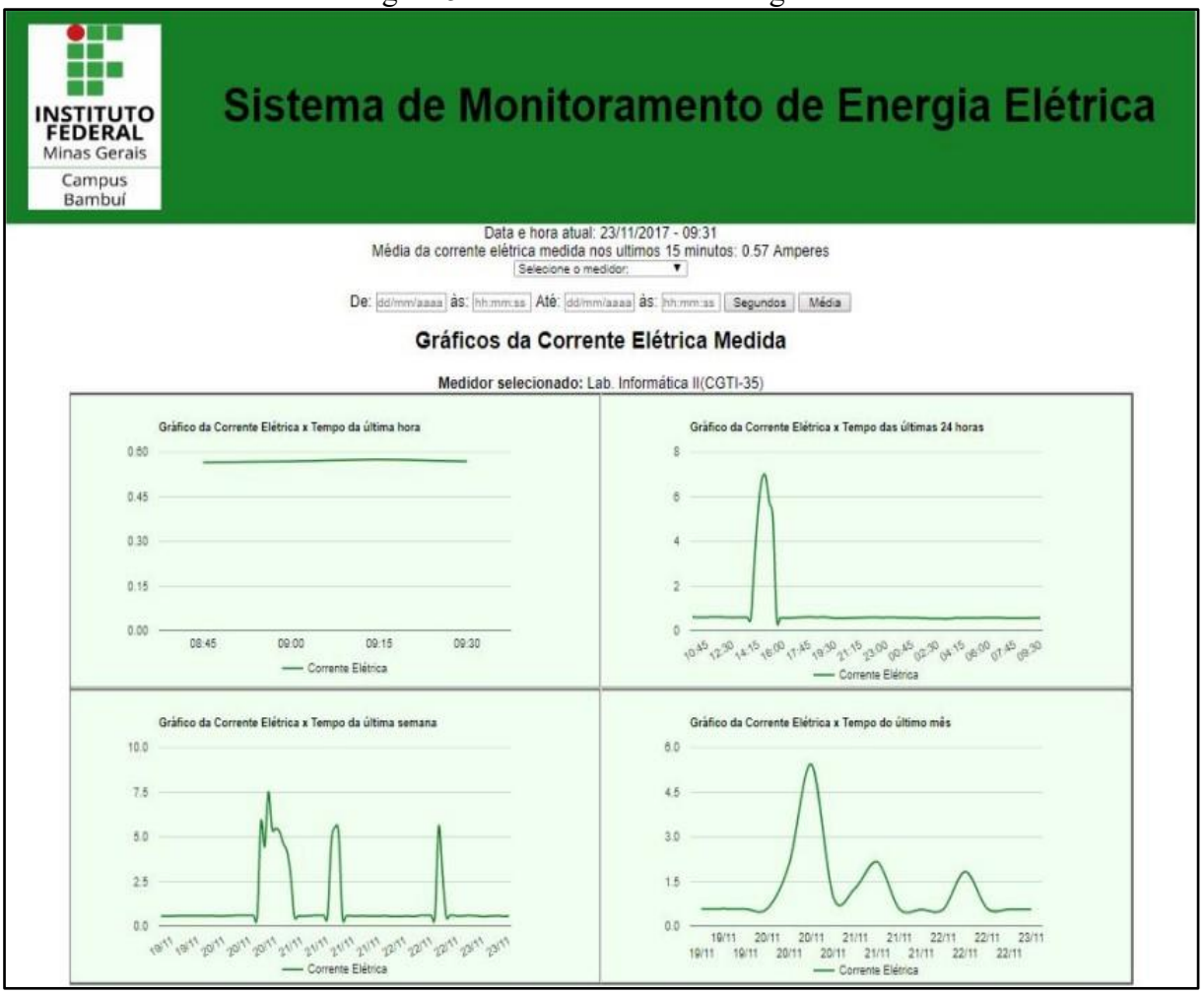
Fonte: Oliveira, 2017.

Figura 4 – Relatório de consumo em um período específico.

Data e Horário	Corrente (A)	Potência (W)	Consumo (kW/h)
22/11/2017 15:00	6.92	1,522.40	0.38
22/11/2017 14:45	6.90	1,517.75	0.38
22/11/2017 14:30	6.78	1,490.86	0.37
22/11/2017 14:15	5.47	1,204.02	0.30
22/11/2017 14:00	3.20	703.61	0.18

Fonte: Oliveira, 2017.

Figura 5 – Gráficos de consumo geral.



Fonte: Oliveira, 2017.

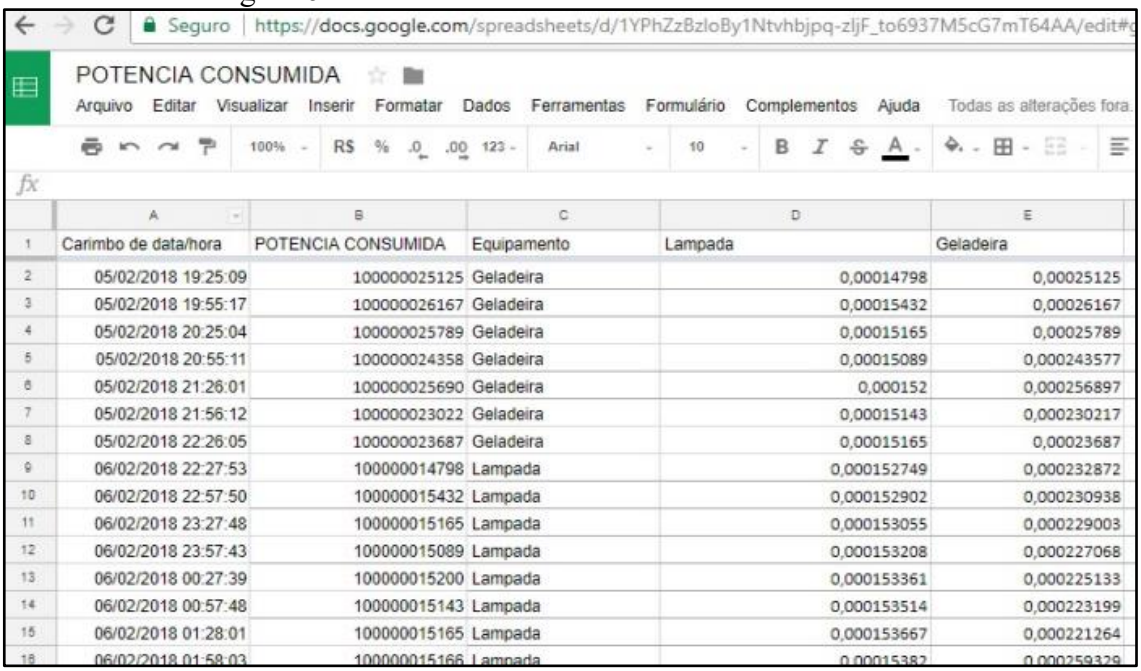
Os testes feitos em ambiente real comprovaram a funcionalidade do protótipo pois os resultados obtidos contribuíram para o monitoramento de energia do IFMG, sendo apresentados relatórios contendo o valor da corrente elétrica, a potência e o consumo em determinado período, e estes dados foram condizentes com o funcionamento do ar-condicionado. (OLIVEIRA, 2017).

2.5.4 Monitoramento do consumo de energia elétrica e controle de equipamentos via aplicativo

O trabalho de monitoramento do consumo de energia elétrica e controle de equipamentos via aplicativo teve como objetivo o desenvolvimento de um protótipo acessível que informou, em tempo real, o gasto de potência elétrica consumida e o preço do consumo de energia de um aparelho. Além disso, permitiu o acionamento do aparelho através da conexão *wireless,* utilizando qualquer dispositivo com conexão a *internet*. Portanto, foi utilizado o sensor ACS-712 que mediu a corrente elétrica, o microcontrolador ESP8266 *NodeMcu*, que possui *WiFi* integrado e utilizou os valores obtidos nas medições para calcular a potência consumida (PEREIRA, 2018).

Desta forma, foi necessário um banco de dados que armazenou estas informações e foi utilizada a ferramenta de planilhas do *Google Docs,* para supervisionar o consumo de cada equipamento testado. Foi gerado um relatório dos dados processados e uma tabela contendo equipamento e consumo, em kWh, além do preço do kWh, Figuras 6 e 7 (PEREIRA, 2018).

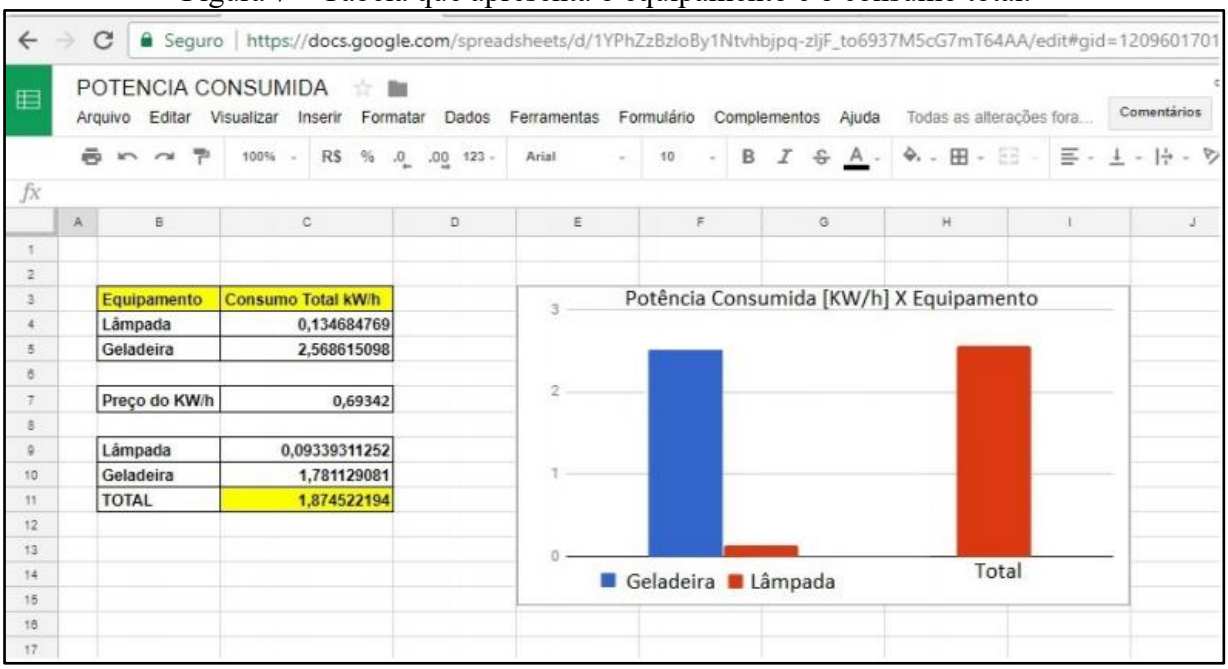
Figura 6 – Planilha com o relatório dos dados enviados.



	A	B	C	D	E
1	Carimbo de data/hora	POTENCIA CONSUMIDA	Equipamento	Lampada	Geladeira
2	05/02/2018 19:25:09	100000025125	Geladeira	0,00014798	0,00025125
3	05/02/2018 19:55:17	100000026167	Geladeira	0,00015432	0,00026167
4	05/02/2018 20:25:04	100000025789	Geladeira	0,00015165	0,00025789
5	05/02/2018 20:55:11	100000024358	Geladeira	0,00015089	0,000243577
6	05/02/2018 21:26:01	100000025690	Geladeira	0,000152	0,000256897
7	05/02/2018 21:56:12	100000023022	Geladeira	0,00015143	0,000230217
8	05/02/2018 22:26:05	100000023687	Geladeira	0,00015165	0,00023687
9	06/02/2018 22:27:53	100000014798	Lampada	0,000152749	0,000232872
10	06/02/2018 22:57:50	100000015432	Lampada	0,000152902	0,000230938
11	06/02/2018 23:27:48	100000015165	Lampada	0,000153055	0,000229003
12	06/02/2018 23:57:43	100000015089	Lampada	0,000153208	0,000227068
13	06/02/2018 00:27:39	100000015200	Lampada	0,000153361	0,000225133
14	06/02/2018 00:57:48	100000015143	Lampada	0,000153514	0,000223199
15	06/02/2018 01:28:01	100000015165	Lampada	0,000153667	0,000221264
16	06/02/2018 01:58:03	100000015166	Lampada	0,00015382	0,000259329

Fonte: Pereira, 2018.

Figura 7 – Tabela que apresenta o equipamento e o consumo total.



Fonte: Pereira, 2018.

O teste realizado apresentou as medições do consumo total de uma lâmpada, com 0,134684769 kWh e de uma geladeira, com 2,568615098 kWh. O preço do kWh, no período de março de 2017 para uma faixa de 52 a 300 kWh residencial, foi de R$0,69342, assim o preço do kWh da lâmpada foi de R$ 0,09339311252 e o da geladeira foi de R$ 1,781129081 (PEREIRA, 2018).

Para o acionamento dos aparelhos foi criado um aplicativo com a ferramenta *MQTT-Dash,* em que foram representados os equipamentos testados e foi possível ativá-los ou desativá-los. Neste teste, pode ser observado que a lâmpada e a geladeira estavam ativadas, conforme mostra a Figura 8 (PEREIRA, 2018).

Figura 8 – Interface do aplicativo.

Fonte: Pereira, 2018.

2.5.5 Análise do consumo de energia em uma unidade consumidora via Internet das Coisas

O trabalho Análise do consumo de energia em uma unidade consumidora via Internet das Coisas teve como objetivo identificar qual aparelho elétrico de uma residência possuía maior consumo de energia. Para isso, foi necessário o uso de sensor de corrente, modelo SCT – 013, para medir o consumo de energia e foi utilizado o microcontrolador ESP8266. Para auxiliar este microcontrolador, foi usado o WeMos D1 R1, que trata e converte os valores medidos para enviar as informações para o servidor. Para gravar programas na placa WeMos D1 R1, foi utilizado o *software* Arduino IDE (CIDADE JÚNIOR, 2018).

Para o servidor foi necessário um computador com acesso à *internet* que inseriu os arquivos em formato PHP no banco de dados MySQL. Estes dados foram inseridos a cada minuto e foram feitas comparações destes valores com o consumo diário, em kWh, medido pela concessionária de energia Copel, o que possibilitou uma supervisão remota do consumo dos equipamentos elétricos. Portanto, foi possível identificar os aparelhos que consumiram mais energia em determinados intervalos de tempo, conforme mostra a Figura 9 (CIDADE JÚNIOR, 2018).

Figura 9 – Comparação do consumo medido pelo protótipo e pela Copel.

Carga	Consumo Diário (kW/h)
Torneira	2,86
Chuveiro	3,05
Secadora	1,74
Máquina de Lavar	0,20
Geladeira	2,09
Média Diária Sensores	9,95
Media Diária Copel	10,41
Diferença (Demais Cargas	0,46

Fonte: Cidade Júnior, 2018.

O teste realizado foi feito no período de agosto a setembro de 2018 e mostrou que o chuveiro teve maior consumo diário, com 3,05 kWh. A potência foi lida a cada minuto e para calcular a média do consumo diário foram utilizadas as cargas monitoradas num tempo mínimo de 48 horas. Portanto, o protótipo apresentou a média de consumo diário de 9,95 kWh, e em comparação com o consumo medido pela concessionária Copel teve 0,46 kWh de diferença, pois o da Copel foi de 10,41 kWh (CIDADE JÚNIOR, 2018).

2.5.6 Análise dos trabalhos relacionados

Os resultados dos trabalhos relacionados atingiram os respectivos objetivos desenvolvendo protótipos de monitoramento do consumo de energia elétrica. Os valores dos consumos foram apresentados em aplicativos ou plataformas *Web,* de forma que o usuário pôde monitorar o gasto de energia elétrica em determinado período e/ou os respectivos consumos de cada aparelho monitorado.

Contudo, os trabalhos apresentaram erros de medições quando comparados com os valores medidos pelas concessionárias de energia, pois não utilizaram sensores de tensão para calcular o valor real da tensão recebida. Em algumas situações pôde ser observado que outros aparelhos eletrônicos, que não estavam conectados aos protótipos, estavam ligados durante os testes de monitoramento e podem ter causado diferença entre os resultados das medições dos consumos calculados pelas concessionárias de energia e pelos protótipos de monitoramento desenvolvidos.

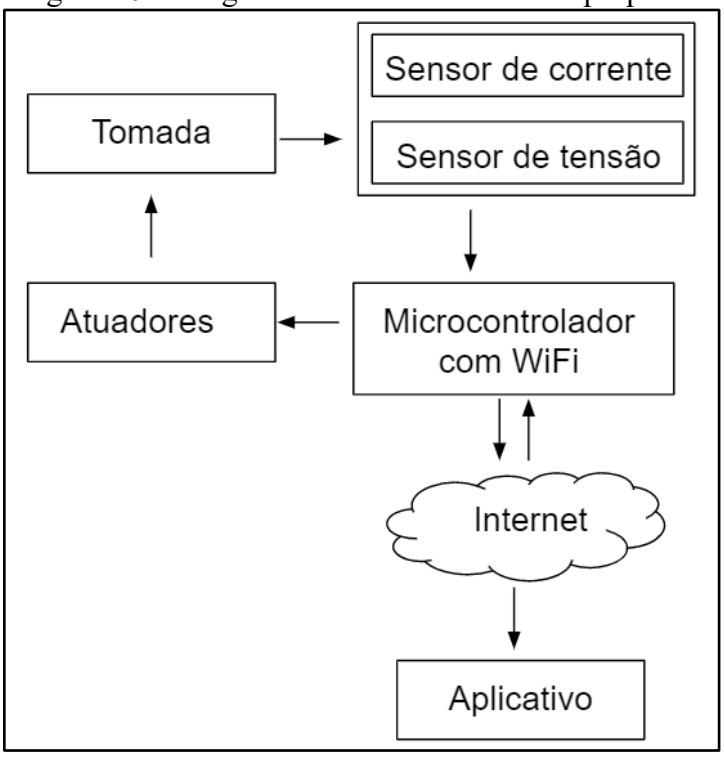
3 PROPOSTA DE SOLUÇÃO

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

3.1 Diagrama de blocos

Para este trabalho foi desenvolvido um protótipo que através de sensores mede a corrente e a tensão de aparelhos eletrônicos conectados a uma tomada, utilizando um microcontrolador para processar estes dados. Além disso, possui atuadores responsáveis pelo acionamento destes aparelhos. Portanto, foi calculado o consumo de energia elétrica e enviado, através da Internet, para um banco de dados na nuvem capaz de armazenar essas informações e apresentá-las ao usuário utilizando um aplicativo Android, como pode ser observado no diagrama da Figura 10.

Figura 10 – Diagrama de blocos do sistema proposto.



Fonte: Elaborado pela autora.

3.2 Hardware

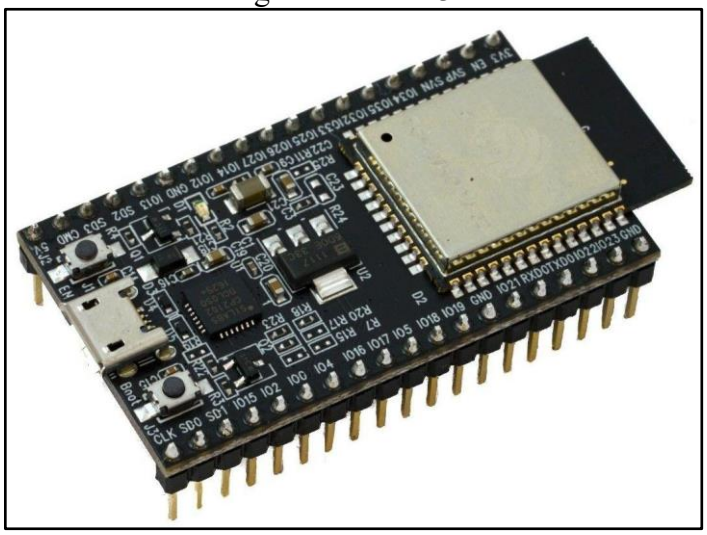
Esta seção apresenta os componentes de *hardware* utilizados para a montagem do protótipo de monitoramento do consumo de energia.

3.2.1 Microcontrolador

O microcontrolador é considerado um computador completo desenvolvido em um único circuito integrado. O circuito integrado é uma miniatura de um circuito eletrônico feito com semicondutores e encapsulado em silício, geralmente, assim ocupa menos espaço em um circuito que teria vários componentes, como resistores, capacitores e diodos. Desta forma, um microcontrolador tem a capacidade de processar informações e possui componentes que possibilitam a execução de tarefas de maneira autônoma. Possui memória para gravar dados, conversores de sinais e portas de entrada e saída programáveis para controlar outros dispositivos (ARAUJO; CAVALCANTE; SILVA, 2019).

Neste trabalho foi utilizado o microcontrolador ESP32, apresentado na Figura 11, desenvolvido pela empresa Espressif, sendo que as principais características são apresentadas na Tabela 1. Este microcontrolador foi projetado para aplicativos móveis, eletrônicos portáteis e IoT, ele contém 38 pinos de entrada ou saída de uso geral. Além disso, o ESP32 utiliza o Protocolo de Controle de Transmissão, do inglês *Transmission Control Protocol* (TCP) e o Protocolo de Internet, do inglês *Internet Protocol* (IP), formando o protocolo TCP/IP e o *WiFi MAC* 802.11 b/g/n, que transmite sinais sem fio através de ondas eletromagnéticas (ESPRESSIF, 2019).

Figura 11 – ESP32.



Fonte: Mercado Livre, 2021a.

Tabela 1 – Características do ESP32.

CPU	Xtensa Dual-Core 32-bit
Memória *EEPROM*	448 kB
Memória *RAM*	520 kB
Memória *Flash*	4 MB
***Clock* máximo**	240 MHz
***Wireless* padrão**	802.11 b/g/n
Conexão *Wi-fi*	2.4 Ghz (máximo de 150 Mbps)
Antena	Sim
Conector	Micro-USB
Bluetooth	BLE 4.2
Quantidade de portas	38
Tensão de operação	4,5 a 9 V
Taxa de transferência	110 a 460800 bps
Peso	9g
Dimensões (comprimento x largura x espessura)	50 x 26 x 13 mm

Fonte: Adaptado de Mercado Livre, 2021a.

3.2.2 Sensor de Corrente

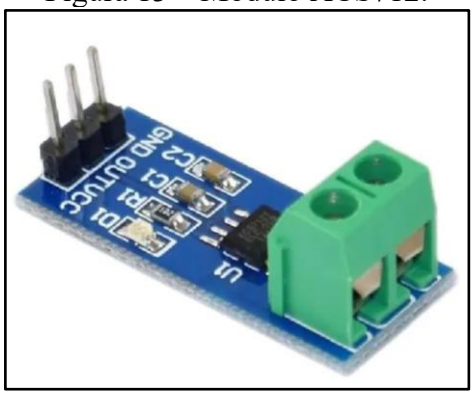
O sensor de corrente ACS712, apresentado na Figura 12, foi utilizado em conjunto com o microcontrolador para medir a corrente elétrica dos aparelhos eletrônicos. O ACS712, desenvolvido pela Allegro, detecta correntes contínuas e alternadas e é uma solução precisa e econômica, sendo utilizado em indústrias, sistemas de comunicação e aplicações no comércio. Este sensor usa o efeito Hall linear, preciso, de baixo *offset* e possui um caminho de condução de cobre. O efeito Hall é gerado quando a corrente passa pelo cobre e gera um campo magnético que resulta em uma tensão proporcional (ALLEGRO MICROSYSTEMS, 2020).

Figura 12 – ACS712.

Fonte: Allegro MicroSystems, 2020.

Para este projeto, foi utilizado o módulo ACS712, apresentado na Figura 13, que consiste em uma placa com o sensor de corrente ACS712 integrado, sendo as principais características apresentadas na Tabela 2 (SARAVATI, 2020).

Figura 13 – Módulo ACS712.



Fonte: Mercado Livre, 2021c.

Tabela 2 – Características do módulo ACS712.

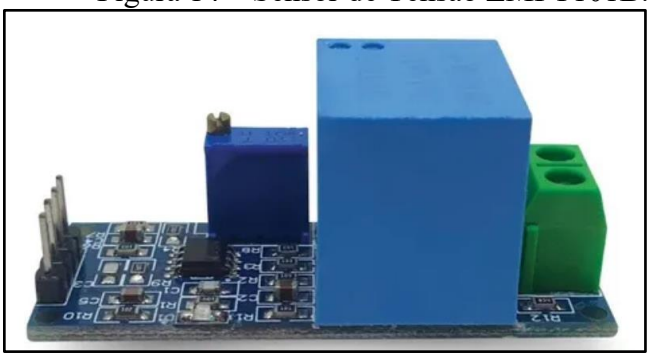
Faixa de medição	-30 a +30A
Alimentação	5V
Tempo de resposta	5 microssegundos
Saída analógica proporcional	66mV/A
Peso	30g
Dimensões (comprimento x largura x espessura)	30 x 12 x 12 mm

Fonte: Adaptado de Mercado Livre, 2021c.

3.2.3 Sensor de Tensão

O módulo de sensor de tensão ZMPT101B, apresentado na Figura 14, mede tensões de 0 a 250 V, sendo as principais características apresentadas na Tabela 3, e foi utilizado neste projeto para medir a tensão recebida em um aparelho e valores para serem processados pelo microcontrolador. Desta forma, permitiu cálculos precisos do consumo de energia elétrica ao invés de utilizar um valor padrão de tensão para realizar os cálculos (MERCADO LIVRE, 2021d).

Figura 14 – Sensor de Tensão ZMPT101B.



Fonte: Mercado Livre, 2021d.

Tabela 3 – Características do módulo ZMPT101B.

Tensão de operação	5 a 30 VDC
Tensão de entrada	0 a 250 VAC
Temperatura de operação	-40 a 70° C
Peso	20g
Dimensões (comprimento x largura x espessura)	51 x 20 x 22 mm

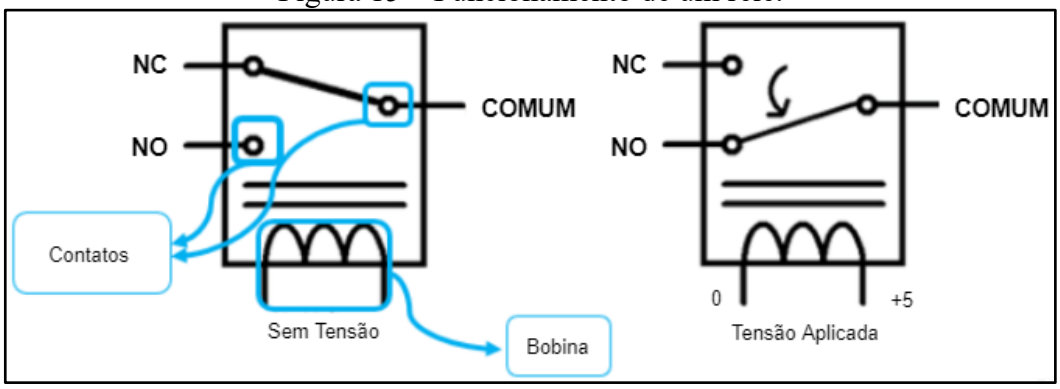
Fonte: Adaptado de Mercado Livre, 2021d.

3.2.4 Atuadores

Os atuadores transformam sinais elétricos em grandezas físicas realizando mudanças no ambiente em que estão inseridos. Desta forma, os atuadores realizam comandos manuais ou automáticos recebidos por um outro dispositivo, um dos tipos de atuadores é o relé, que atua com pequenas correntes e aciona circuitos que possuem correntes elevadas (LONGO, 2015).

O relé, Figura 15, funciona de forma a fechar ou abrir os contatos conectados aos terminais de um equipamento elétrico para acioná-lo. Sendo assim, é necessário que uma corrente elétrica percorra a bobina do relé, atraindo uma chapa metálica para um dos dois terminais através do campo magnético. Portanto, quando uma tensão é aplicada tem-se a mudança de contato, podendo estar em: (a) Normalmente Fechado, ou *Normally Close* (NC), quando não tem uma tensão energizando a bobina; ou (b) Normalmente Aberto, ou *Normally Open* (NO), quando tem uma tensão aplicada de até 5V (FREITAS; DIOGENES, 2019).

Figura 15 – Funcionamento de um relé.

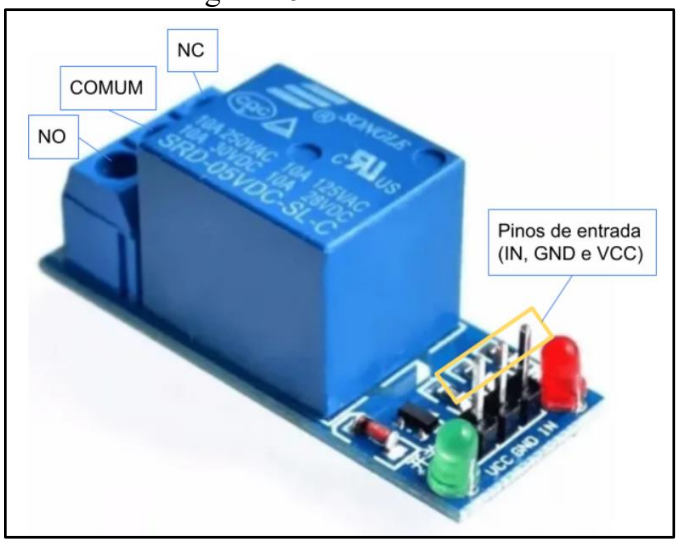


Fonte: Adaptado de Future Eletronic Corporation, 2020.

Para este trabalho foi utilizado um módulo relé, Figura 16, que suporta corrente de até 10A para tensões 250V AC ou 30V *Direct Current* (DC). O módulo tem dimensões de 46 mm de comprimento, 25 mm de largura e 18 mm de espessura, 16 gramas e é composto por 6

pinos, sendo três pinos de entrada (IN, GND e VCC): o IN tem uma tensão de operação que varia entre 3 e 5V, o GND em 0 e VCC em 5V e os três pinos de saída são os NO, NC e COMUM. Este módulo conectado ao microcontrolador, com a saída configurada para alto (5V), resulta no funcionamento da bobina que controla o processo de mudança de estado do contato dos terminais acionando um dispositivo remotamente (FUTURE ELETRONIC CORPORATION, 2020).

Figura 16 – Módulo relé.



Fonte: Adaptado de Mercado Livre, 2021b.

3.3 Software

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

3.3.1 Banco de dados

O banco de dados mantém informações organizadas de forma a torná-las disponíveis quando solicitadas, sendo possível que o usuário realize operações como consultar, alterar, excluir e inserir novos dados. Neste trabalho foi utilizado o Firebase *Realtime Database* que é um banco de dados na nuvem, sendo que os dados foram armazenados como *JavaScript Object Notation* (JSON) e sincronizados em tempo real. O Firebase é NoSQL, portanto tem funcionalidades diferentes de um banco de dados relacional, sendo desenvolvido para operações executadas com rapidez (DUTRA, 2017; FIREBASE, 2019c).

Os dados são armazenados na árvore JSON hospedada na nuvem na forma de objetos no formato de nós e podem ser acessados com uma chave associada, sendo que o usuário pode fornecer as próprias chaves ou gerá-las utilizando o “childByAutoId”. Os dados armazenados podem ser escritos como tipos nativos, correspondendo a tipos JSON existentes para que se tenha códigos simples de serem atualizados (FIREBASE, 2019b).

Para a criação de um aplicativo que se conecte ao Firebase pode-se utilizar o Android Studio, de duas formas: (a) utilizar o console do Firebase, que é a alternativa recomendada; ou (b) utilizar o Firebase Assistente do Android Studio, que necessita de uma configuração adicional (FIREBASE, 2019a).

3.3.2 Linguagem de programação

A linguagem de programação é uma abstração de conceitos complexos da matemática, sendo utilizada para resolver determinados problemas. Para a construção da linguagem de programação são necessários dois conceitos: sintaxe e semântica. A sintaxe está relacionada ao que está escrito e a semântica ao significado (BERTOLINI *et al.*, 2019).

Assim, foi utilizado o Kotlin que é uma linguagem executada sobre a máquina virtual Java. O Kotlin foi desenvolvido pela empresa JetBrains, sendo projetado para corrigir erros da linguagem Java tornando o desenvolvimento de software seguro. A principal característica é ser uma linguagem *null-safe*, isso evita erros em tempo de execução porque é possível detectar objetos nulos antes da execução do código. Um aspecto importante do Kotlin é a expressividade em comparação ao Java, diminuindo a quantidade de linhas de código que seria em Java, pois o Kotlin gera funções automaticamente para a manipulação de informações (TOLEDO, 2019).

3.3.3 Android Studio

O Android Studio é o ambiente de desenvolvimento integrado, ou *Integrated Development Environment* (IDE) utilizado para o desenvolvimento de aplicativos para Android. Os projetos criados podem ser visualizados em *Gradle Scripts,* sendo que cada módulo do aplicativo são separados em: (a) *Manifests,* contém o arquivo “AndroidManifest.xml” que descreve informações essenciais do aplicativo para as ferramentas de compilação; (b) *Java,* contém os arquivos do código-fonte; e (c) Recursos, que contém os *layouts* em *Extensible Markup Language* (XML), *strings* da interface de usuário, ou *User Interface* (UI) e imagens em bitmap, que é um conjunto de *pixels* que formam um elemento de cor (ANDROID STUDIO, 2020).

O Android Studio utiliza o sistema de compilação base *Gradle,* que é executado de forma independente na linha de comando, como uma ferramenta adicionada ao menu, permitindo que cada projeto tenha recursos de compilação. Além disso, tem um arquivo de compilação de nível superior que envolve todo o projeto e para cada módulo existem arquivos de compilação separados (ANDROID STUDIO, 2020).

3.3.4 Arduino Software IDE

O Arduino *Software* IDE foi utilizado para fazer o *upload* de programas para o *hardware* e permitir a comunicação com ele. Essa IDE contém um editor de texto para escrever o código e uma área que apresenta, por exemplo as mensagens de erros, além de conter barras de ferramentas e menu. Os programas escritos nesta IDE através do editor de texto são chamados de *sketches* e possuem a extensão de arquivo em ".ino" (ARDUINO, 2015).

Sendo assim, o Arduino *Software* IDE utiliza o conceito de um caderno de rascunhos, ou *sketchbook,* que é um local padrão para armazenar os *sketches.* O IDE cria um diretório automaticamente quando executa o programa pela primeira vez. O *upload* de um *sketch* para um microcontrolador é feito pelo Arduino *bootloader,* que fica ativo alguns segundos quando a placa é redefinida e inicia o último *sketch* carregado para o microcontrolador (ARDUINO, 2015).

Para que este IDE seja utilizado para programar o ESP32 foi necessário adicionar uma *Uniform Resource Locator* (URL) nos Gerenciadores de Placas, permitindo o acesso a configuração de várias placas no formato ".json", sendo possível instalar a placa "esp32" feita pela Espressif Systems. Desta forma, o módulo ESP32 foi adicionado na lista de placas disponíveis e selecionado, através da porta em que estava conectado, para que os códigos pudessem ser gravados nele (USINAINFO, 2019).

4 DESENVOLVIMENTO

Este capítulo apresenta o desenvolvimento de um protótipo de monitoramento do consumo de energia elétrica em uma residência.

4.1 Montagem do protótipo

Esta seção mostra os testes dos componentes de *hardware* e a montagem do protótipo de monitoramento do consumo de energia. Os testes foram feitos usando um computador com processador Intel Core i5, memória RAM de 8 GB, sistema operacional Windows 10 de 64 *bits*.

4.1.1 Testes unitários

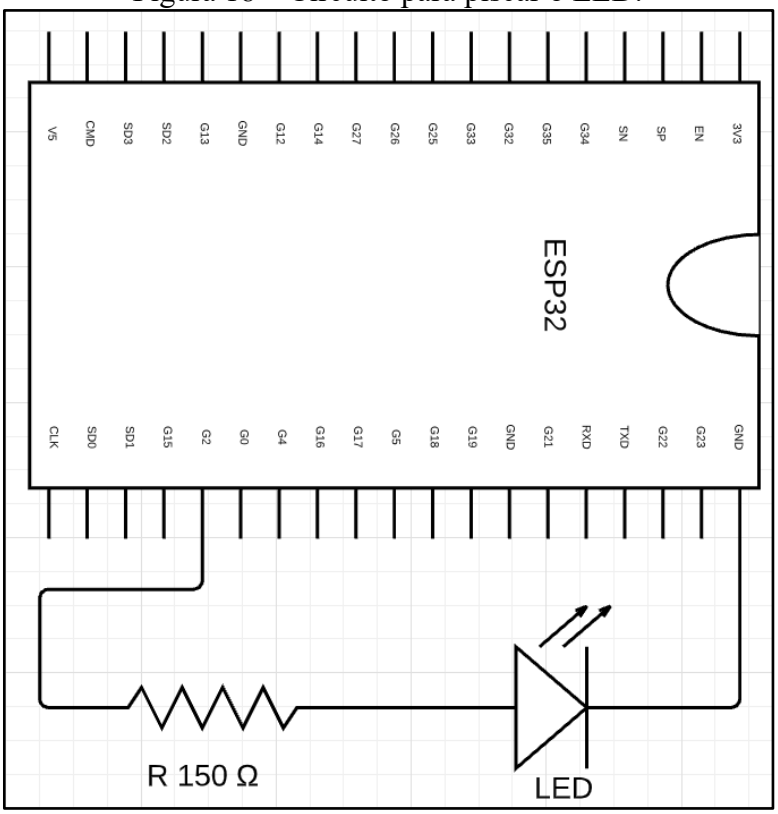
O ESP32 foi testado com o código da Figura 17 feito para piscar um LED. Para isso, foi montado um circuito usando um ESP32, um LED e um resistor de 150 Ohms, Figura 18, resultando em um *loop,* para ligar e desligar o LED por 1,5 segundo.

Figura 17 – Código para piscar o LED.

```
blink
const int ledPin = 2;
void setup() {
  pinMode (ledPin, OUTPUT);
}
void loop() {
  digitalWrite (ledPin, HIGH);
  delay(1500);
  digitalWrite (ledPin, LOW);
  delay(1500);
}
```

Fonte: Elaborado pela autora.

Figura 18 – Circuito para piscar o LED.



Fonte: Elaborado pela autora.

Para o segundo teste foi usado o código da Figura 19, usando um módulo relé, um ESP32 e uma lâmpada de 25 *Watts* (W). Esse módulo relé possui um transistor 2TY que se conecta de forma que o módulo fique ativado com o GND e desativado com 5V, por isso foi necessário um transistor para que seja feito o acionamento correto deste módulo, conforme apresentado no circuito da Figura 20.

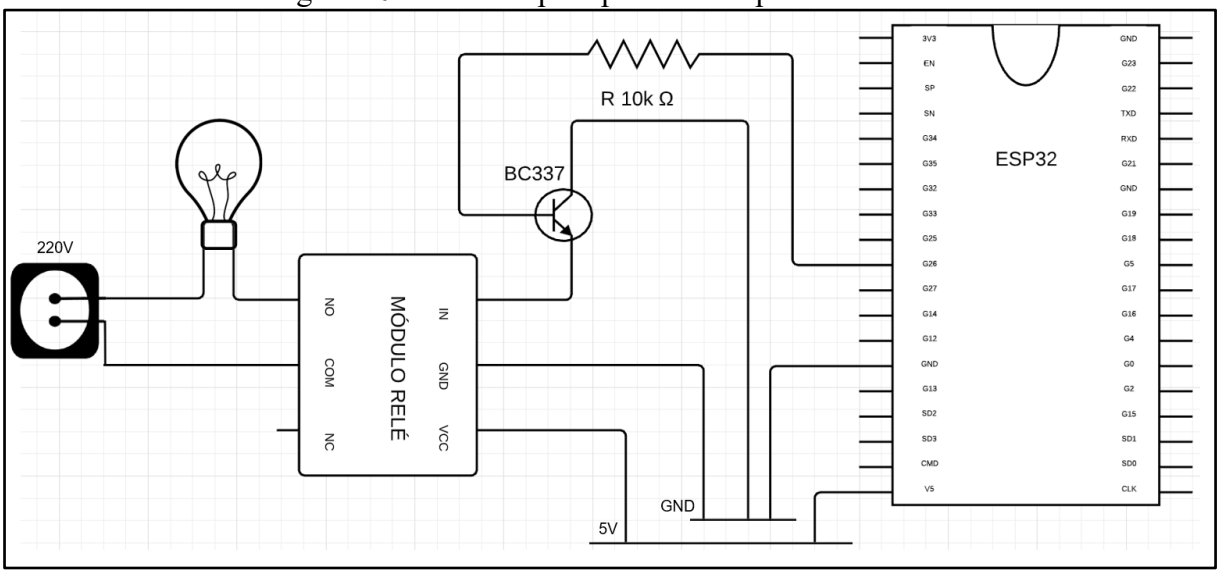
Foi utilizado o transistor BC337 que funciona como chave de contato entre o GND e o sinal de entrada IN do módulo relé, fazendo com que este módulo fique ativado ou desativado conforme o sinal enviado pelo ESP32 através da porta 26, resultando em um loop para ligar e desligar a lâmpada por 5 segundos.

Figura 19 – Código para piscar a lâmpada com relé.

```
blink_relay
void setup() {
  pinMode(26, OUTPUT);
}

void loop() {
  digitalWrite(26, HIGH);
  delay(5000);
  digitalWrite(26, LOW);
  delay(5000);
}
```

Fonte: Elaborado pela autora.

Figura 20 – Circuito para piscar a lâmpada com relé.



Fonte: Elaborado pela autora.

Para o terceiro teste foi usado um módulo ACS712, um ESP32 e uma prancha *Gama Italy bivolt* que atinge até 210º *Celsius.* Para isso foi utilizado o código da Figura 21, adaptado de Amaral (2017), sendo que a constante *SAMPLING_TIME* é o intervalo de amostragem de 166,8649 microssegundos, *LINE_FREQUENCY* se refere a frequência da rede com valor de 60 Hz e a *VOLTAGE_AC* que é a tensão de 220 V. Já a *ACS_MPY* é a constante de calibração e foi definida de forma empírica, medindo com o multímetro os valores de corrente e fazendo a relação com os valores medidos pelo sensor, obtendo 3,5.

A biblioteca *EmonLib* implementa um algoritmo *Root Mean Square* (RMS) para medir valores de tensão e corrente. A função *current* passa como parâmetros tanto o pino 26, usado no ESP32, quanto a constante de calibração. Como a frequência é de 60 Hz então a duração é

de 16,66 milissegundos (ms), e passando 1480 amostras por parâmetro para a função *calcIrms()*.

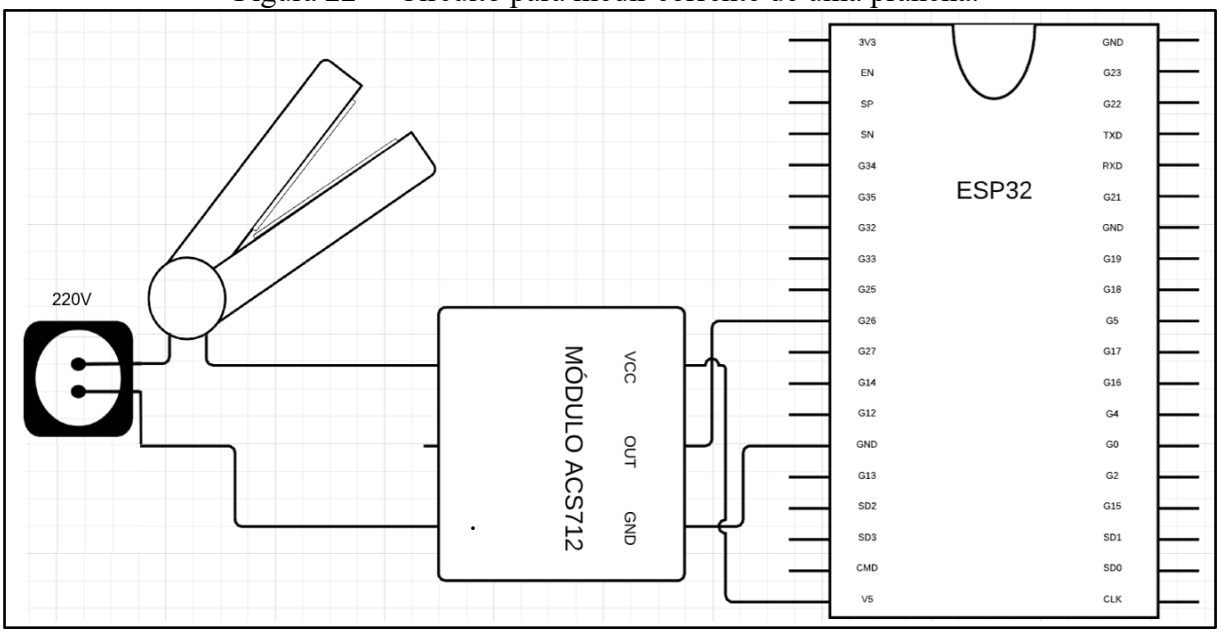
A Figura 22 apresenta o circuito montado para realizar este teste, obtendo valores de corrente entre 0.71 e 1.33 A e a potência aparente entre 156.92 e 293.20 *Volt-Ampère* (VA), conforme mostra a Figura 23.

Figura 21 – Código para medir corrente de uma prancha.

```
acs712
#include <EmonLib.h>
EnergyMonitor emon1;

#define   SAMPLING_TIME     0.0001668649
#define   LINE_FREQUENCY    60
#define   VOLTAGE_AC        220.00
#define   ACS_MPY           3.5

double Irms = 0;

void setup()
{
  Serial.begin(115200);
  emon1.current(26, ACS_MPY);
}

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

  Serial.print("Corrente: ");   Serial.print(Irms);   Serial.println(" A");
  Serial.println("Pot. Aparente: ");   Serial.print(Irms * VOLTAGE_AC); Serial.println(" VA");
  Serial.println();
  delay(500);
}
```

Fonte: Elaborado pela autora.

Figura 22 – Circuito para medir corrente de uma prancha.



Fonte: Elaborado pela autora.

Figura 23 – Valores medidos pelo sensor de corrente.

```
Corrente: 1.33 A
Pot. Aparente: 293.20 VA

Corrente: 1.17 A
Pot. Aparente: 258.49 VA

Corrente: 1.10 A
Pot. Aparente: 241.94 VA

Corrente: 1.03 A
Pot. Aparente: 226.22 VA

Corrente: 0.97 A
Pot. Aparente: 214.20 VA

Corrente: 0.90 A
Pot. Aparente: 197.70 VA

Corrente: 0.87 A
Pot. Aparente: 191.98 VA

Corrente: 0.86 A
Pot. Aparente: 189.58 VA

Corrente: 0.79 A
Pot. Aparente: 173.24 VA

Corrente: 0.81 A
Pot. Aparente: 177.95 VA

Corrente: 0.71 A
Pot. Aparente: 156.92 VA
```

Fonte: Elaborado pela autora.

Para o quarto teste foi utilizado o código da Figura 24, para medir os valores de tensão de uma tomada e foi montado o circuito, apresentado na Figura 25, com um módulo ZMPT101B e um ESP32.

No código, Figura 24, a constante *VOLT_CAL* foi definida com o valor 97.5, obtido empiricamente com a utilização de um multímetro para ajustar essa calibração. A função *voltage* da biblioteca *EmonLib* passa como parâmetros o pino 26 do ESP32, a constante de calibração *VOLT_CAL* e 1.7, sendo que o último é o *phase_shift,* que é a correção para os erros de amostragem da tensão.

Já a função *calcVI* passa como parâmetros 25 e 1000, sendo que o primeiro corresponde ao número de pontos e o segundo ao período de amostragem em ms. A função *Vrms* faz a leitura de tensão e armazena o valor na variável *supplyVoltage.* Posteriormente, faz a comparação se o valor lido é maior ou igual a 100, caso afirmativo apresenta os valores, caso contrário apresenta "Desligado". No teste realizado foi apresentado "Desligado", o que indica que a tomada estava desligada e quando a tomada estava ligada apresentou os valores entre 209.76 e 227.89 V, conforme mostrado na Figura 26.

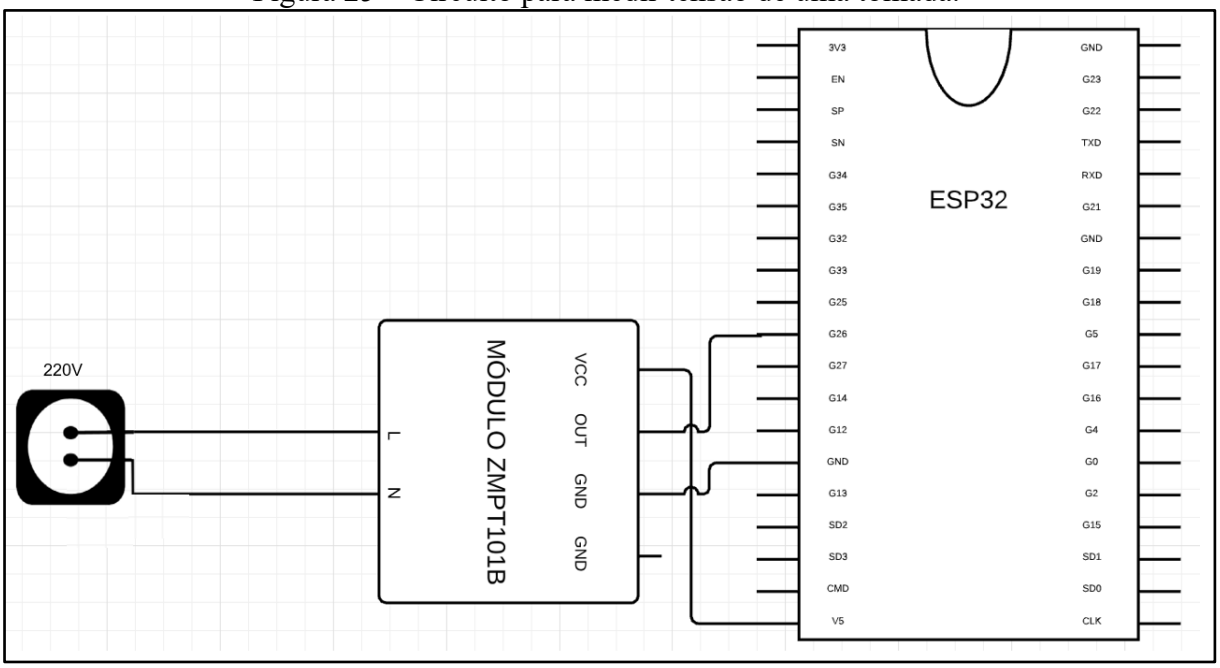
Figura 24 – Código para medir tensão de uma tomada.

zmpt101b

```
#include "EmonLib.h"

#define VOLT_CAL 97.5

EnergyMonitor emon1;

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

  emon1.voltage(26, VOLT_CAL, 1.7);
}

void loop()
{
  emon1.calcVI(25,1000);

  float supplyVoltage = emon1.Vrms;
  if(supplyVoltage >= 100){
    Serial.print("Tensão : ");
    Serial.print(supplyVoltage);
    Serial.println(" V");
  }
  else{
   Serial.println("Desligado");
  }
}
```

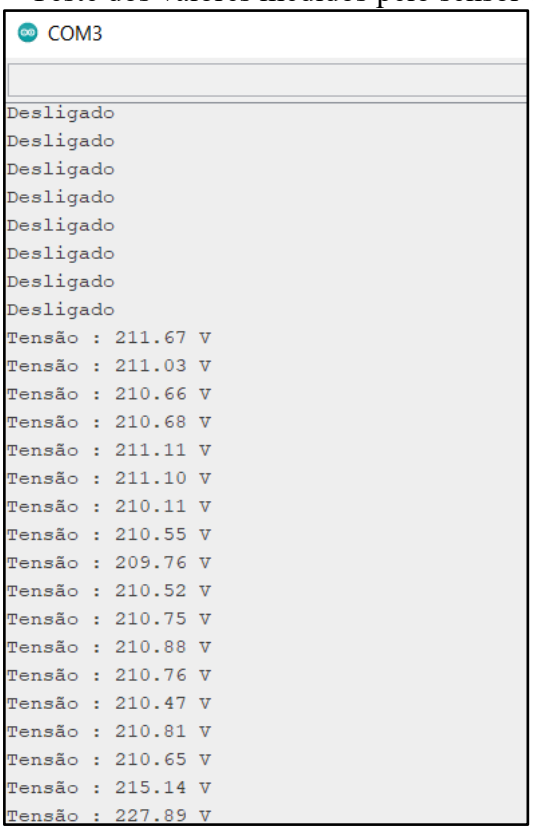
Fonte: Elaborado pela autora.

Figura 25 – Circuito para medir tensão de uma tomada.



Fonte: Elaborado pela autora.

Figura 26 – Teste dos valores medidos pelo sensor de tensão.



Fonte: Elaborado pela autora.

4.1.2 Teste de integração

Para a integração dos dispositivos de *hardware* foram utilizadas as seguintes bibliotecas:

- Emonlib, foi desenvolvida para diversos sensores e é voltada para o monitoramento de energia. Para configurar a instância da biblioteca devem ser indicados: o pino de entrada do microcontrolador e a constante de calibração (AMARAL, 2017);
- Wifi, permite a conexão do microcontrolador à Internet, sendo necessário fornecer a identificação da rede e a senha (ARDUINO, 2019);
- NTPClient e WifiUdp permitem a conexão do microcontrolador a um servidor externo, sincronizando o relógio do microcontrolador com a configuração do relógio de determinada região (CURTO CIRCUITO, 2018);
- IOXhop_FirebaseESP32, permite que o ESP32 manipule valores no Firebase (ARTRON SHOP, 2018).

O teste de integração foi realizado com o ESP32, o módulo ACS712, o módulo ZMPT101B e um ferro elétrico. Para isso, foi utilizado o código da Figura 27, que agrupou os códigos feitos para os testes unitários dos módulos. Foram utilizados os pinos 25 e 26, sendo o primeiro para o módulo ZMPT101B e o segundo para o módulo ACS712.

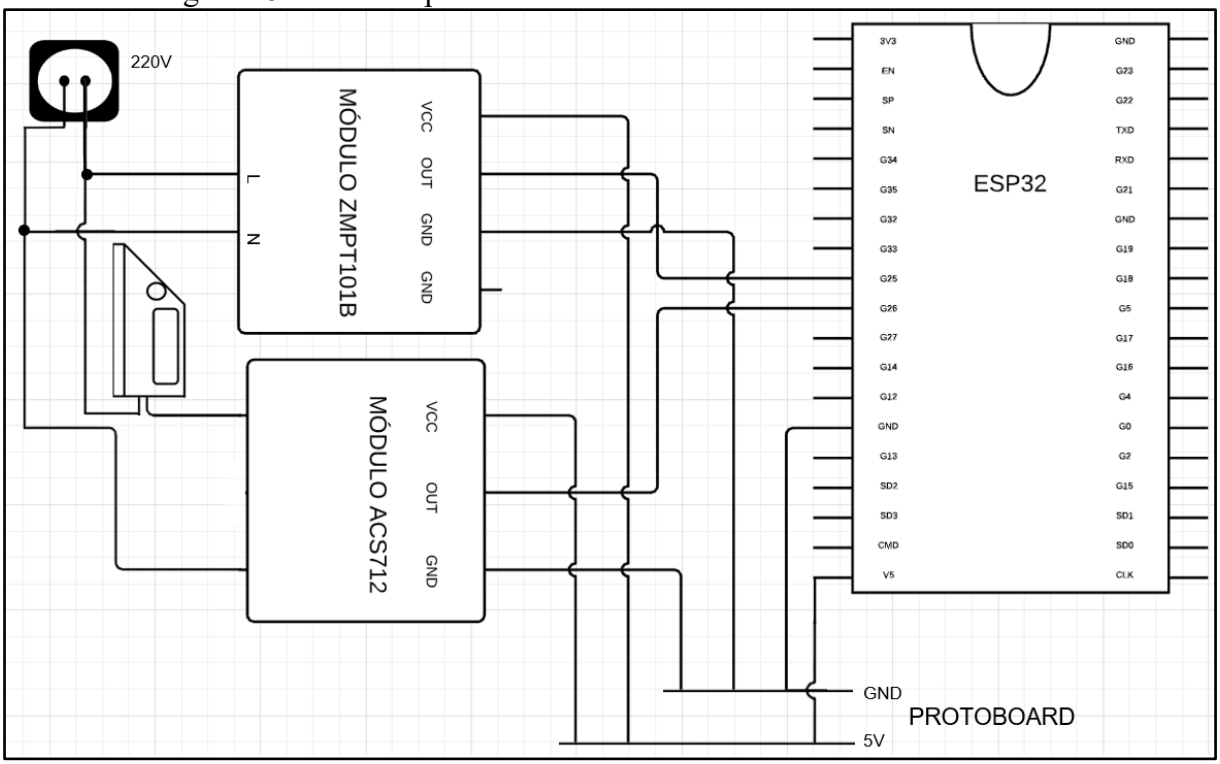
A Figura 28 apresenta o circuito montado para realizar este teste, obtendo os valores de corrente entre 0.37 e 5.49 A, de tensão apresentou os valores entre 205.50 e 217.99 V, e a potência entre 80.12 e 1163.64 W, conforme mostrado na Figura 29.

Figura 27 – Código para medir tensão e corrente de um ferro elétrico.

```
zmpt101b_acs712
#include "EmonLib.h"

#define   VOLT_CAL          97.5
#define   SAMPLING_TIME     0.0001668649
#define   LINE_FREQUENCY    60
#define   ACS_MPY           3.5

EnergyMonitor emon1;

double Irms = 0;
float power = 0;

void setup()
{
  Serial.begin(115200);
  emon1.voltage(25, VOLT_CAL, 1.7);
  emon1.current(26, ACS_MPY);
}

void loop()
{
  emon1.calcVI(25,1000);
  Irms = emon1.calcIrms(1480);

  float supplyVoltage = emon1.Vrms;
  power = Irms * supplyVoltage;
  Serial.print("Corrente: "); Serial.print(Irms); Serial.println(" A");
  Serial.print("Tensão: "); Serial.print(supplyVoltage);  Serial.println(" V");
  Serial.print("Potência: ");  Serial.print(power);  Serial.println(" W");
  Serial.println();
  delay(500);
}
```

Fonte: Elaborado pela autora.

Figura 28 – Circuito para medir tensão e corrente de um ferro elétrico.



Fonte: Elaborado pela autora.

Figura 29 – Teste dos valores medidos pelos sensores de tensão e de corrente.

```
Corrente: 5.46 A
Tensão: 213.23 V
Potência: 1163.64 W

Corrente: 5.47 A
Tensão: 208.49 V
Potência: 1140.91 W

Corrente: 5.40 A
Tensão: 209.22 V
Potência: 1129.15 W

Corrente: 5.49 A
Tensão: 205.50 V
Potência: 1129.13 W

Corrente: 0.38 A
Tensão: 214.61 V
Potência: 82.06 W

Corrente: 0.37 A
Tensão: 213.72 V
Potência: 80.12 W

Corrente: 0.39 A
Tensão: 214.66 V
Potência: 84.40 W

Corrente: 0.39 A
Tensão: 216.54 V
Potência: 84.52 W

Corrente: 0.39 A
Tensão: 217.99 V
Potência: 84.60 W

Corrente: 0.39 A
Tensão: 213.47 V
Potência: 82.53 W
```

Fonte: Elaborado pela autora.

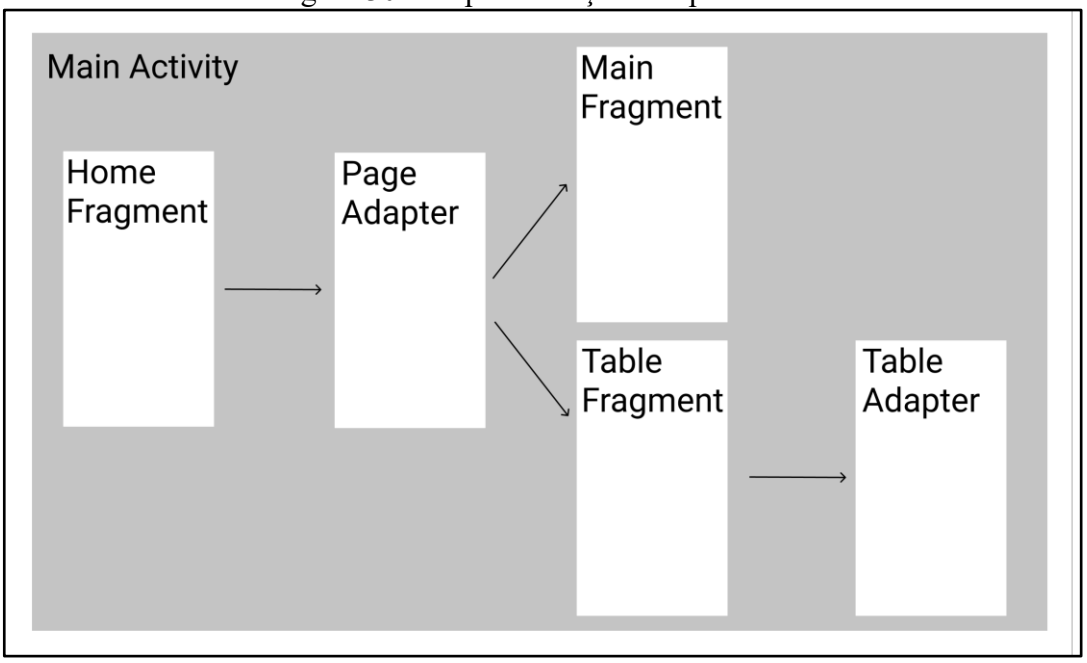
4.2 Elaboração do aplicativo

O aplicativo foi desenvolvido usando a linguagem de programação Kotlin, seguindo a arquitetura *single-activity,* ou seja, possui apenas uma *activity* que funciona como um contêiner com as *fragments* dentro para representar a tela. A *activity* é o componente onde se inicia o aplicativo, sendo que as *fragments* são diretamente impactadas pelo ciclo de vida desta, conforme exemplificado na Figura 30.

A *HomeFragment,* responsável pelo *TabLayout*, que é formado por duas *tabs,* foi implementada com um *PageAdapter* para fazer a transição das duas telas. A primeira tela é a *Home,* implementada na *MainFragment,* em que foi apresentado: (a) um gráfico, que mostra o consumo em kWh por dia; (b) um *card* de valores*,* contendo o consumo total em kWh, que é o somatório do consumo de todos os dias e o valor correspondente em reais; (c) um *card* de botões, que contém os botões *ON* e *OFF* para habilitar e desabilitar a tomada, respectivamente. A segunda tela é a *Tabela,* implementada na *TableFragment,* que apresenta uma tabela, com os valores lidos sendo mostrados em duas colunas, sendo a primeira de data

e hora, e a segunda com o consumo em kW. Para que os dados sejam mostrados na tabela foi implementada a *Table Adapter.*

Figura 30 – Implementação do aplicativo.



Fonte: Elaborado pela autora.

4.3 Comunicação do protótipo com o aplicativo

O teste de comunicação foi feito para piscar um LED via *wi-fi* usando um aplicativo Android. Para isso, foi desenvolvido o código da Figura 31, adaptado de Khan (2020), que inclui as bibliotecas *Wifi* e *IOXhop_FirebaseESP32*, a primeira conecta o ESP32 à internet e a segunda manipula os valores no Firebase através do ESP32. Foram definidos o *host* e autenticação do Firebase, e o identificador e a senha do *wi-fi* local.

A variável *LED* identifica o pino 26 do ESP32. A função *Wifi.begin* passa como parâmetros o identificador e a senha do *wi-fi* conectando o ESP32 à internet.

A função *Firebase.begin* passa como parâmetros o *host* e a autenticação conectando o ESP32 ao Firebase. A função *Firebase.setString("LED_STATUS", "OFF")* inicializa a variável "*LED_STATUS*" do Firebase com o valor "*OFF*".

Na função *loop,* a variável *fireStatus* armazena o valor recebido da variável "*LED_STATUS*", depois faz a verificação, se esse valor for "*ON*" liga o LED, se for "*OFF*" desliga o LED. Foi montado um circuito, apresentado na Figura 32, composto por um ESP32, um LED e um resistor de 150 ohms.

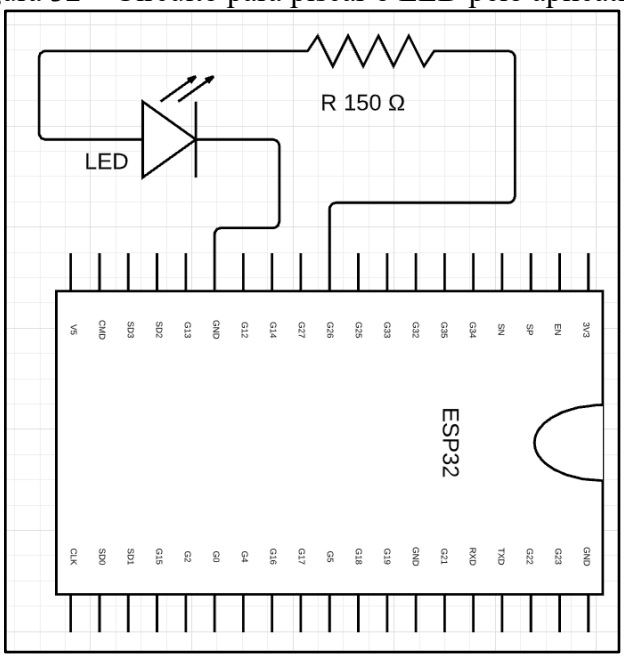
Figura 31 – Código para piscar o LED pelo aplicativo usando o Firebase.

FIREBASE_LED

```
#include <WiFi.h>
#include <IOXhop_FirebaseESP32.h>
#define FIREBASE_HOST "project-tcc-iot-default-rtdb.firebaseio.com/"
#define FIREBASE_AUTH "[redacted]"
#define WIFI_SSID "NET-V"
#define WIFI_PASSWORD "[redacted]"

String fireStatus = "";
int LED = 26;

void setup() {
  Serial.begin(115200);
  delay(1000);
  pinMode(LED, OUTPUT);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Conectando ao ");
  Serial.print(WIFI_SSID);

  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }

  Serial.println();
  Serial.print("Conectado ao ");
  Serial.println(WIFI_SSID);
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Firebase.setString("LED_STATUS", "OFF");
}

void loop() {
  fireStatus = Firebase.getString("LED_STATUS");
  if (fireStatus == "ON") {
    Serial.println("Led ON");
    digitalWrite(LED, HIGH);
  }
  else if (fireStatus == "OFF") {
    Serial.println("Led OFF");
    digitalWrite(LED, LOW);
  }
}
```

Fonte: Elaborado pela autora.

Figura 32 – Circuito para piscar o LED pelo aplicativo.



Fonte: Elaborado pela autora.

O aplicativo foi implementado com dois botões em uma tela, o botão "*ON*" para ligar o LED e o botão "*OFF*" para desligar o LED, no trecho de código apresentado na Figura 33, a variável *database* recebe a referência do banco de dados do Firebase. Quando o usuário clica no botão "*OFF*", *binding.off.setOnClickListener,* a variável do Firebase "*LED_STATUS*" recebe como valor "*OFF*" e quando clica no botão "*ON*", *binding.on.setOnClickListener,* a "*LED_STATUS*" recebe como valor "*ON*".

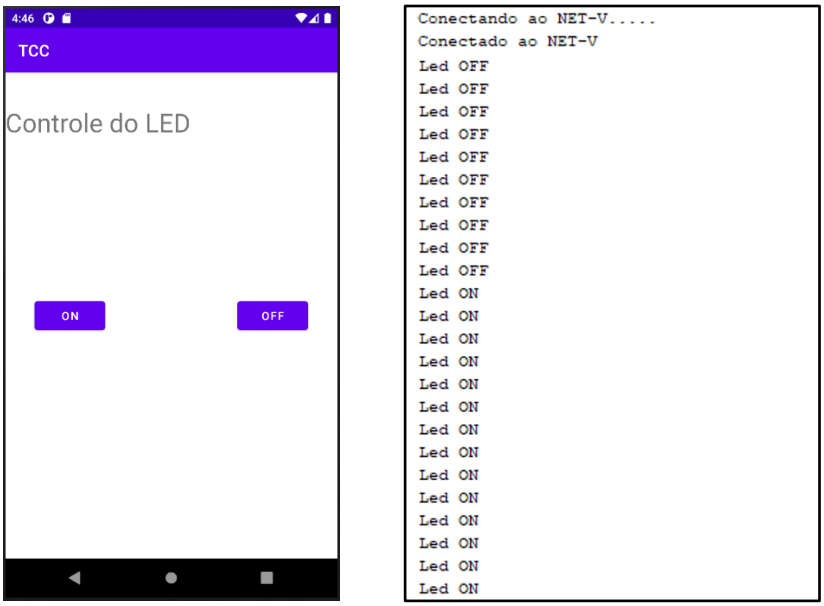
Figura 33 – Trecho de código para enviar o dado de ligar/desligar o LED para o Firebase.

```
    val database = Firebase.database

    binding.off.setOnClickListener {  it: View!
        val myRef = database.getReference( path: "LED_STATUS")
        myRef.setValue("OFF")
    }

    binding.on.setOnClickListener {  it: View!
        val myRef = database.getReference( path: "LED_STATUS")
        myRef.setValue("ON")
    }
}
```

Fonte: Elaborado pela autora.

O resultado desta comunicação, Figura 34 (a) mostra a tela com os dois botões "*ON*" e "*OFF*", a Figura 34 (b) mostra a tela de resultados do monitor serial. A Figura 35 (a) e (b) mostra a tela do Firebase com o dado "*LED_STATUS*" passando de "*OFF*" para "*ON*", conforme foi dado o comando através do aplicativo, fazendo com que o LED passe de desligado para ligado, Figura 36 (a) e (b).

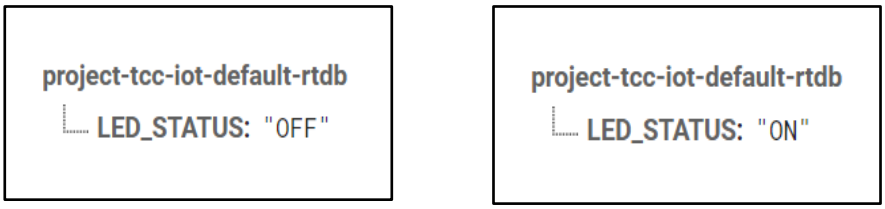
Figura 34 – Resultado do aplicativo Android e do monitor serial.



(a) (b)

Fonte: Elaborado pela autora.

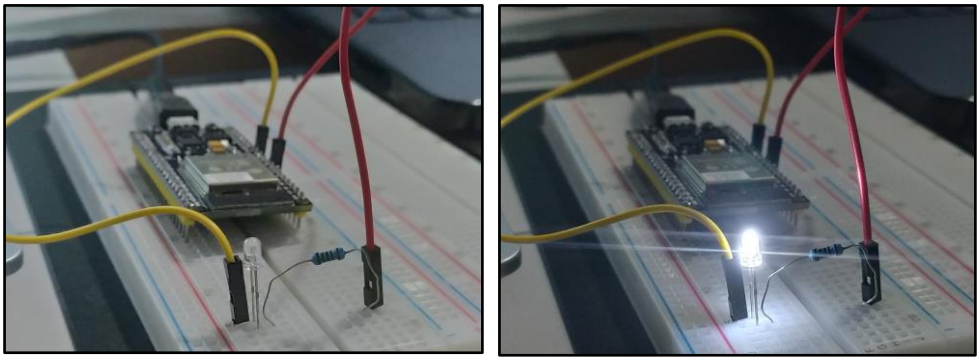
Figura 35 – Resultado do Firebase.



(a) (b)

Fonte: Elaborado pela autora.

Figura 36 – Resultado do comando de ligar o LED.



(a) (b)

Fonte: Elaborado pela autora.

5 RESULTADOS OBTIDOS

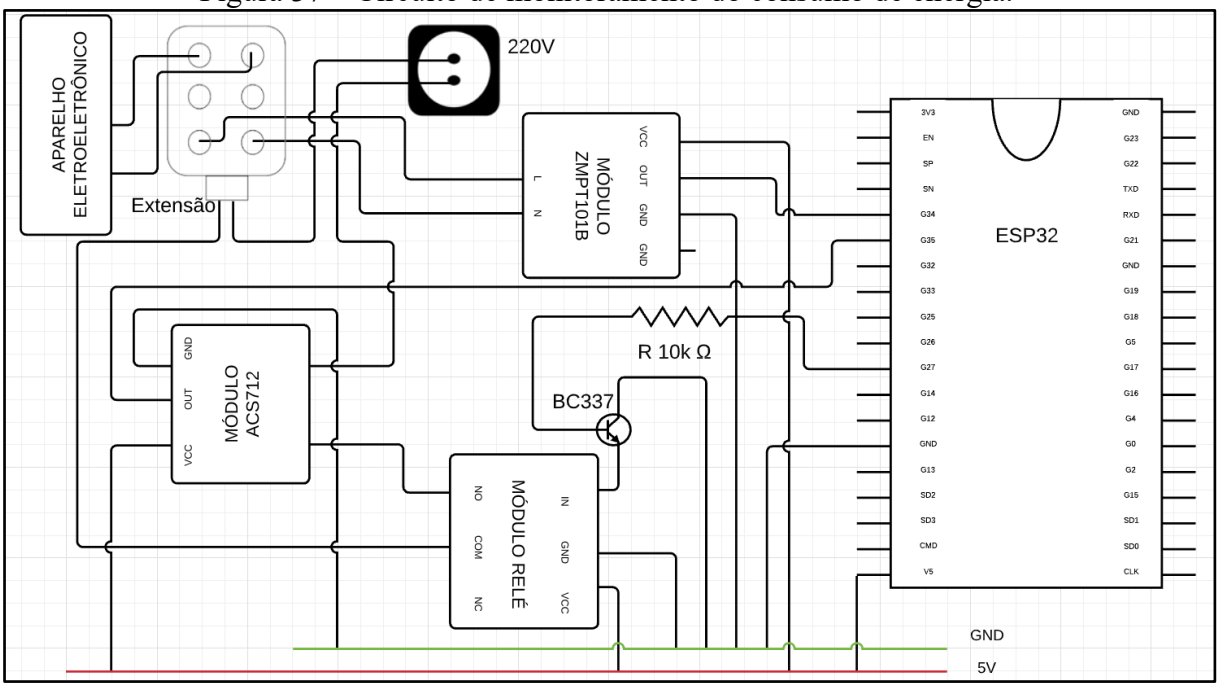
Este capítulo mostra os resultados obtidos com o desenvolvimento do protótipo de monitoramento do consumo de energia elétrica em uma residência.

5.1 Análise dos resultados obtidos

O código fonte do aplicativo está disponível em <https://github.com/laryssa-oliveira/kWh-Monitoring>. O Apêndice A contém o código para o monitoramento do consumo de energia. Para o ESP32 fornecer a data e a hora foi necessário conectá-lo à internet, o que tornou possível a utilização do *Network Time Protocol* (NTP), que retorna a data e a hora lida. Para isso, foi necessário fornecer o fuso horário, sendo passado como parâmetro na função *setTimeOffset()* o valor -10800, que é o equivalente ao *Greenwich Mean Time* menos 3 horas (GMT-3) em segundos.

O Apêndice B contém uma imagem do protótipo de monitoramento do consumo de energia, que foi representado através do circuito da Figura 37, sendo montado para medir a tensão da rede e a corrente do aparelho eletroeletrônico para calcular o consumo de energia. Logo, foi medido o consumo em kWh entre os dias 20 e 23 de novembro de 2021 de um ferro elétrico. Os valores medidos foram apresentados no aplicativo através do gráfico, além de mostrar o consumo total e o preço deste em reais, Figura 38.

Figura 37 – Circuito de monitoramento do consumo de energia.



Fonte: Elaborado pela autora.

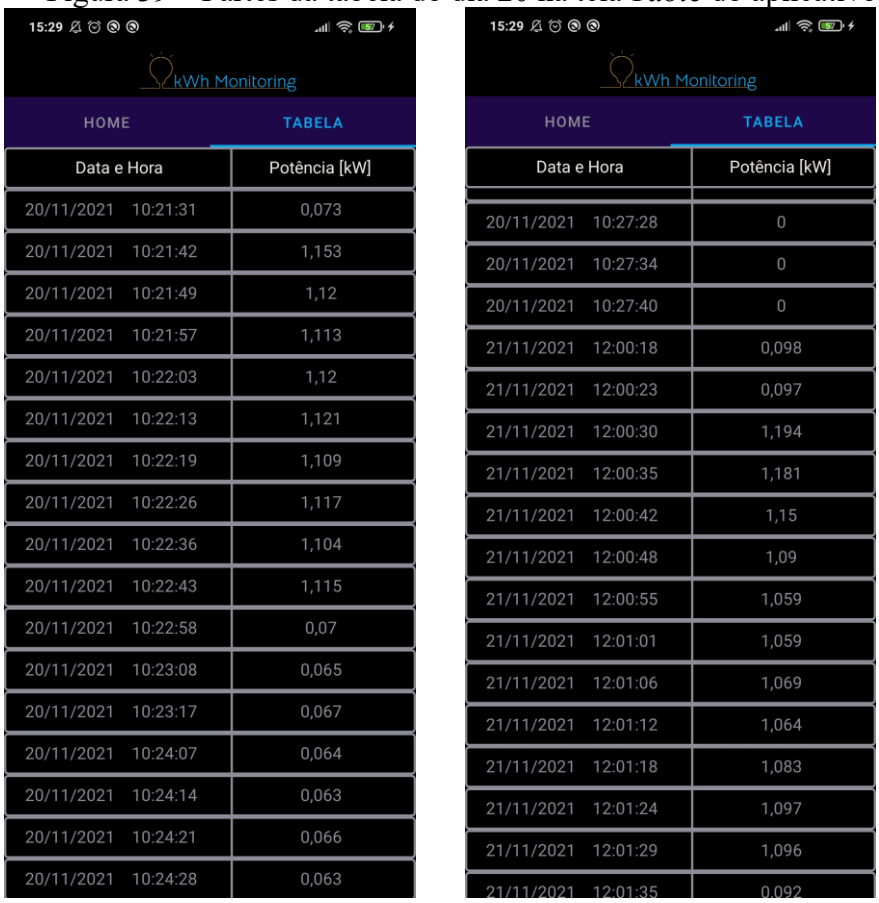
Figura 38 – Tela *Home* do aplicativo.

Fonte: Elaborado pela autora.

No dia 20, conforme apresentado no gráfico da Figura 38, foi observado um consumo de 0,0330 kWh, sendo que o ferro foi conectado à tomada no intervalo de tempo entre 10:21:31 e 10:27:40, totalizando 6 minutos e 9 segundos. Contudo, a tomada ficou desabilitada por 29 segundos e habilitada por 5 minutos e 40 segundos. Na tela Tabela foram apresentadas as datas e horários e as respectivas potências em kW.

A Figura 39 (a) mostra uma parte da tabela, sendo apresentadas as datas e horários do dia 20 e os respectivos consumos em kW. Logo, verificou que a primeira leitura apresentou a potência de 0,073 kW e a última leitura do dia 20 foi de 0 kW, Figura 39 (b). O consumo de 0 kW foi a leitura feita quando a tomada foi desabilitada pelo aplicativo, clicando no botão "*OFF*", conforme *control_status* no Firebase, mostrado na Figura 40.

Figura 39 – Partes da tabela do dia 20 na tela *Table* do aplicativo.



15:29

kWh Monitoring

HOME | TABELA

Data e Hora	Potência [kW]
20/11/2021 10:21:31	0,073
20/11/2021 10:21:42	1,153
20/11/2021 10:21:49	1,12
20/11/2021 10:21:57	1,113
20/11/2021 10:22:03	1,12
20/11/2021 10:22:13	1,121
20/11/2021 10:22:19	1,109
20/11/2021 10:22:26	1,117
20/11/2021 10:22:36	1,104
20/11/2021 10:22:43	1,115
20/11/2021 10:22:58	0,07
20/11/2021 10:23:08	0,065
20/11/2021 10:23:17	0,067
20/11/2021 10:24:07	0,064
20/11/2021 10:24:14	0,063
20/11/2021 10:24:21	0,066
20/11/2021 10:24:28	0,063

(a)

15:29

kWh Monitoring

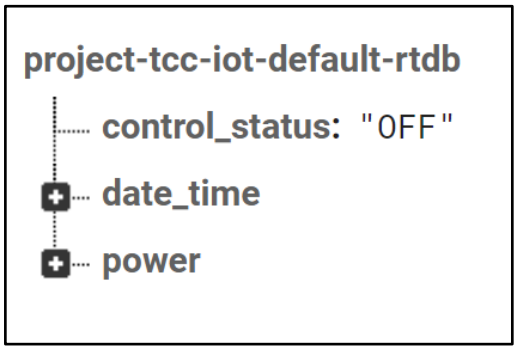
HOME | TABELA

Data e Hora	Potência [kW]
20/11/2021 10:27:28	0
20/11/2021 10:27:34	0
20/11/2021 10:27:40	0
21/11/2021 12:00:18	0,098
21/11/2021 12:00:23	0,097
21/11/2021 12:00:30	1,194
21/11/2021 12:00:35	1,181
21/11/2021 12:00:42	1,15
21/11/2021 12:00:48	1,09
21/11/2021 12:00:55	1,059
21/11/2021 12:01:01	1,059
21/11/2021 12:01:06	1,069
21/11/2021 12:01:12	1,064
21/11/2021 12:01:18	1,083
21/11/2021 12:01:24	1,097
21/11/2021 12:01:29	1,096
21/11/2021 12:01:35	0,092

(b)

Fonte: Elaborado pela autora.

Figura 40 – Resultado do botão “OFF” no Firebase.



Fonte: Elaborado pela autora.

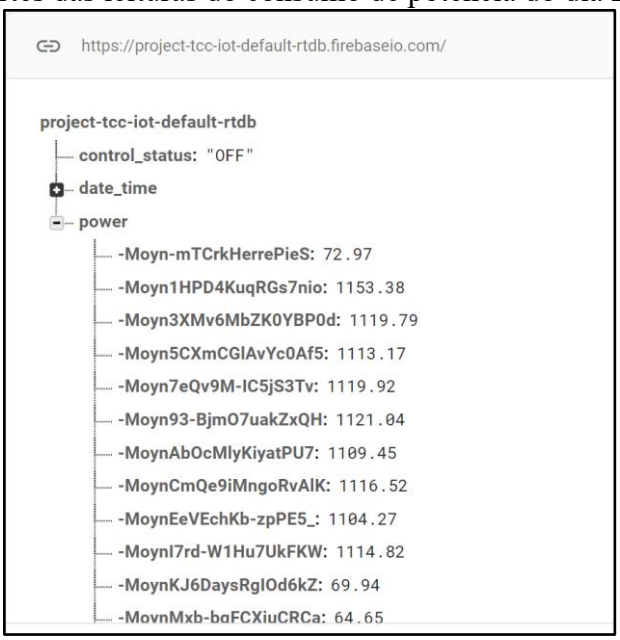
Assim, quando a tomada foi desabilitada não foi contado o tempo para que não alterasse o cálculo do kWh, vez que a potência foi de 0 kW. Como foi calculado o consumo de um ferro elétrico, o mesmo possui um funcionamento em que desliga automaticamente por um tempo, por isso pode ser observado leituras de potência menores do que 1 kW, como por exemplo a faixa de valores entre 0,063 e 0,073 kW apresentada na Figura 39 (a).

O banco de dados Firebase armazenou a data e o horário, Figura 41, e a respectiva potência em W, Figura 42, em que foram realizadas as leituras. Nas Figuras 41 e 42 pode ser observada uma parte das leituras feitas no dia 20.

Figura 41 – Partes das leituras da data e hora do dia 20 no Firebase.

Fonte: Elaborado pela autora.

Figura 42 – Partes das leituras do consumo de potência do dia 20 no Firebase.

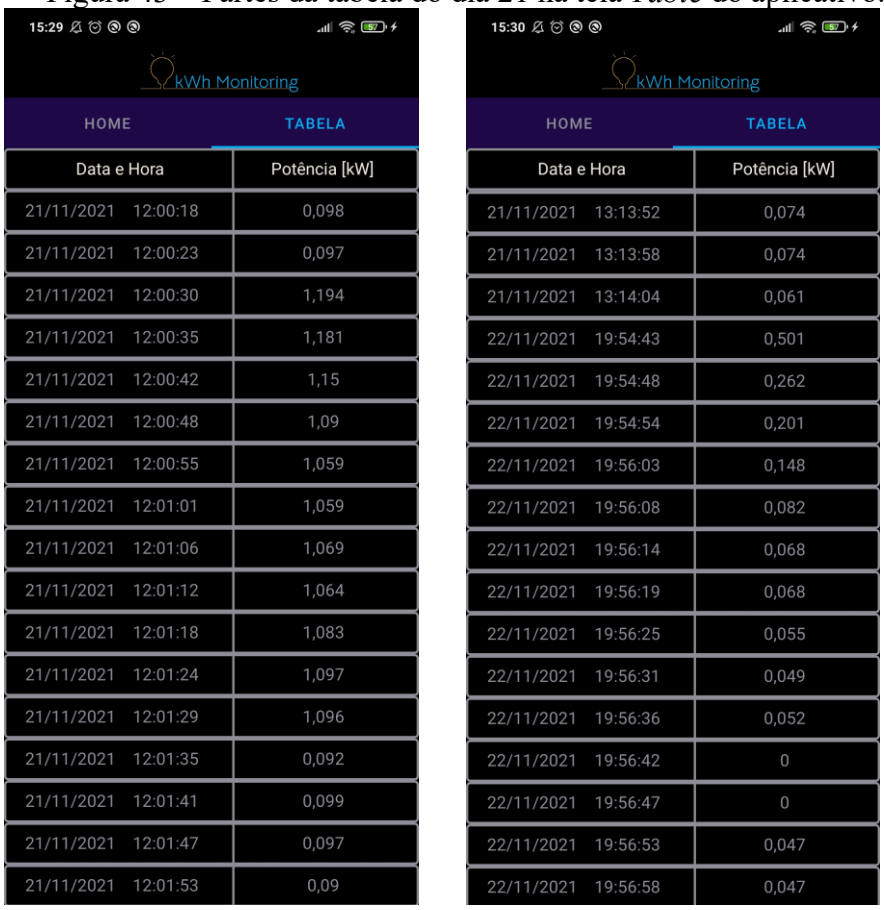


Fonte: Elaborado pela autora.

No dia 21, foi apresentado o consumo de 0,0985 kWh, conforme observado no gráfico da Figura 38, sendo que o ferro foi conectado à tomada entre o intervalo de tempo 12:00:18 e 13:14:04. Mas a tomada ficou desabilitada por 51 minutos e 16 segundos, e habilitada por 22

minutos e 30 segundos, totalizando 1 hora, 13 minutos e 46 segundos. A primeira leitura, conforme observado na Figura 43 (a) foi 0,098 kW e a última leitura do dia foi 0,061 kW, Figura 43 (b).

Figura 43 – Partes da tabela do dia 21 na tela *Table* do aplicativo.



Data e Hora	Potência [kW]
21/11/2021 12:00:18	0,098
21/11/2021 12:00:23	0,097
21/11/2021 12:00:30	1,194
21/11/2021 12:00:35	1,181
21/11/2021 12:00:42	1,15
21/11/2021 12:00:48	1,09
21/11/2021 12:00:55	1,059
21/11/2021 12:01:01	1,059
21/11/2021 12:01:06	1,069
21/11/2021 12:01:12	1,064
21/11/2021 12:01:18	1,083
21/11/2021 12:01:24	1,097
21/11/2021 12:01:29	1,096
21/11/2021 12:01:35	0,092
21/11/2021 12:01:41	0,099
21/11/2021 12:01:47	0,097
21/11/2021 12:01:53	0,09

(a)

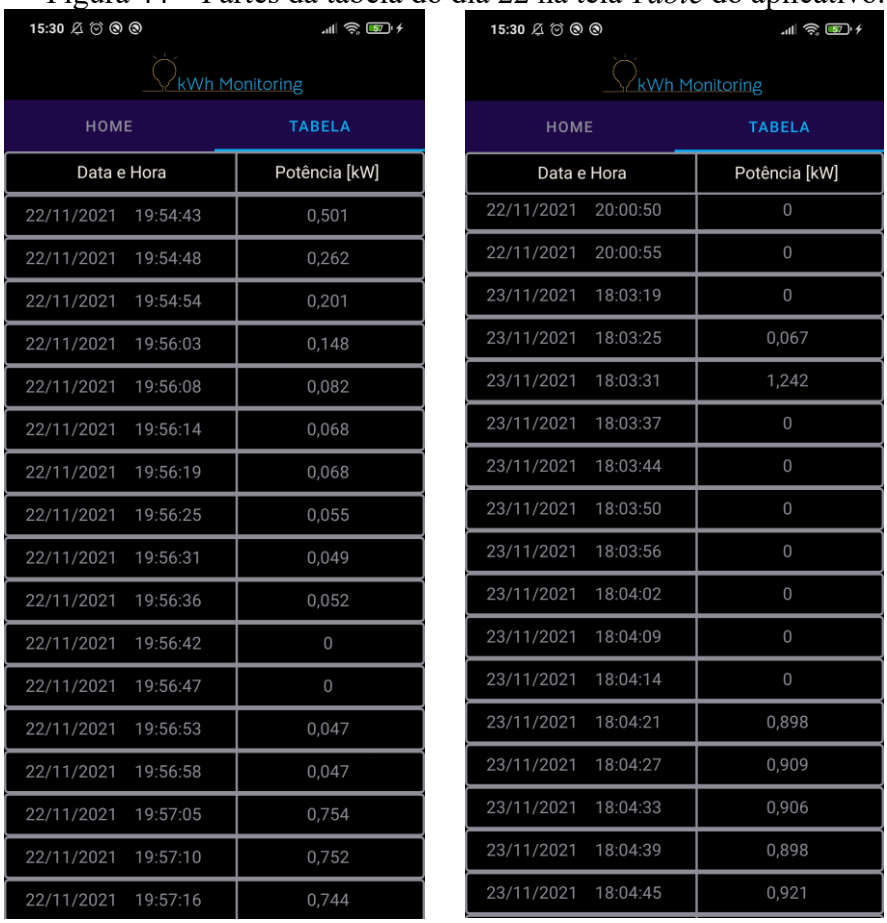
Data e Hora	Potência [kW]
21/11/2021 13:13:52	0,074
21/11/2021 13:13:58	0,074
21/11/2021 13:14:04	0,061
22/11/2021 19:54:43	0,501
22/11/2021 19:54:48	0,262
22/11/2021 19:54:54	0,201
22/11/2021 19:56:03	0,148
22/11/2021 19:56:08	0,082
22/11/2021 19:56:14	0,068
22/11/2021 19:56:19	0,068
22/11/2021 19:56:25	0,055
22/11/2021 19:56:31	0,049
22/11/2021 19:56:36	0,052
22/11/2021 19:56:42	0
22/11/2021 19:56:47	0
22/11/2021 19:56:53	0,047
22/11/2021 19:56:58	0,047

(b)

Fonte: Elaborado pela autora.

No dia 22, o consumo foi de 0,0268 kWh, como observado no gráfico da Figura 38, e as leituras foram realizadas entre 19:54:43 e 20:00:55, Figura 44 (a) e (b), sendo que a primeira leitura do dia mostrou um consumo de 0,501 kW e a última leitura um consumo de 0 kW. Assim como nos outros dias a tomada ficou desabilitada em alguns intervalos de tempo, somando 17 segundos, e ficou habilitada por 5 minutos e 55 segundos, totalizando um tempo de 6 minutos e 12 segundos.

Figura 44 – Partes da tabela do dia 22 na tela *Table* do aplicativo.



Data e Hora	Potência [kW]
22/11/2021 19:54:43	0,501
22/11/2021 19:54:48	0,262
22/11/2021 19:54:54	0,201
22/11/2021 19:56:03	0,148
22/11/2021 19:56:08	0,082
22/11/2021 19:56:14	0,068
22/11/2021 19:56:19	0,068
22/11/2021 19:56:25	0,055
22/11/2021 19:56:31	0,049
22/11/2021 19:56:36	0,052
22/11/2021 19:56:42	0
22/11/2021 19:56:47	0
22/11/2021 19:56:53	0,047
22/11/2021 19:56:58	0,047
22/11/2021 19:57:05	0,754
22/11/2021 19:57:10	0,752
22/11/2021 19:57:16	0,744

(a)

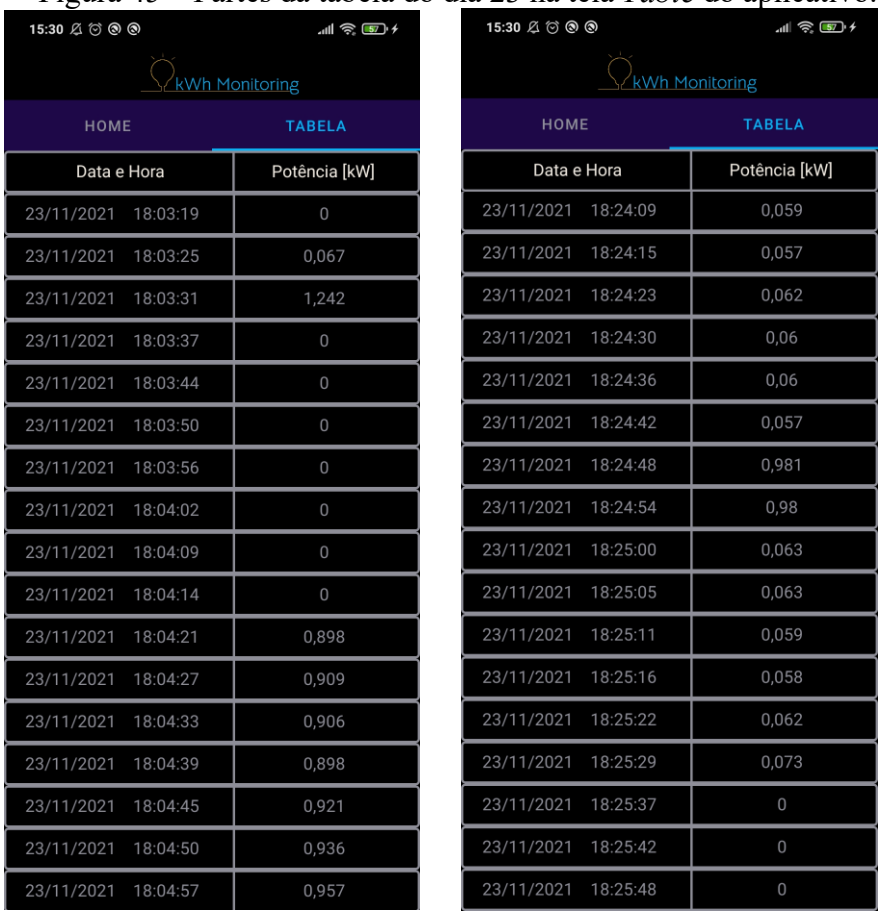
Data e Hora	Potência [kW]
22/11/2021 20:00:50	0
22/11/2021 20:00:55	0
23/11/2021 18:03:19	0
23/11/2021 18:03:25	0,067
23/11/2021 18:03:31	1,242
23/11/2021 18:03:37	0
23/11/2021 18:03:44	0
23/11/2021 18:03:50	0
23/11/2021 18:03:56	0
23/11/2021 18:04:02	0
23/11/2021 18:04:09	0
23/11/2021 18:04:14	0
23/11/2021 18:04:21	0,898
23/11/2021 18:04:27	0,909
23/11/2021 18:04:33	0,906
23/11/2021 18:04:39	0,898
23/11/2021 18:04:45	0,921

(b)

Fonte: Elaborado pela autora.

No dia 23, foram consumidos 0,0788 kWh, conforme apresentado no gráfico da Figura 38, sendo que as leituras foram realizadas entre 18:03:19 e 18:25:48, conforme apresentado na Figura 45 (a) e (b), como pode ser observado tanto a primeira quanto a última leitura do dia foi 0 kW. E o tempo total foi de 22 minutos e 29 segundos, sendo que a tomada ficou desabilitada por 3 minutos e 30 segundos, e ficou habilitada por 18 minutos e 59 segundos.

Figura 45 – Partes da tabela do dia 23 na tela *Table* do aplicativo.



15:30

kWh Monitoring

HOME | TABELA

Data e Hora	Potência [kW]
23/11/2021 18:03:19	0
23/11/2021 18:03:25	0,067
23/11/2021 18:03:31	1,242
23/11/2021 18:03:37	0
23/11/2021 18:03:44	0
23/11/2021 18:03:50	0
23/11/2021 18:03:56	0
23/11/2021 18:04:02	0
23/11/2021 18:04:09	0
23/11/2021 18:04:14	0
23/11/2021 18:04:21	0,898
23/11/2021 18:04:27	0,909
23/11/2021 18:04:33	0,906
23/11/2021 18:04:39	0,898
23/11/2021 18:04:45	0,921
23/11/2021 18:04:50	0,936
23/11/2021 18:04:57	0,957

15:30

kWh Monitoring

HOME | TABELA

Data e Hora	Potência [kW]
23/11/2021 18:24:09	0,059
23/11/2021 18:24:15	0,057
23/11/2021 18:24:23	0,062
23/11/2021 18:24:30	0,06
23/11/2021 18:24:36	0,06
23/11/2021 18:24:42	0,057
23/11/2021 18:24:48	0,981
23/11/2021 18:24:54	0,98
23/11/2021 18:25:00	0,063
23/11/2021 18:25:05	0,063
23/11/2021 18:25:11	0,059
23/11/2021 18:25:16	0,058
23/11/2021 18:25:22	0,062
23/11/2021 18:25:29	0,073
23/11/2021 18:25:37	0
23/11/2021 18:25:42	0
23/11/2021 18:25:48	0

(a) (b)

Fonte: Elaborado pela autora.

Logo, foi feito o somatório de consumo dos dias 20 a 23 de novembro de 2021 e foi apresentado o resultado desta soma no aplicativo, Figura 38, conforme observado no primeiro *card* da tela, o consumo total resultou em 0,237 kWh. Além disso, mostrou o preço deste consumo em reais, sendo calculado conforme a tarifa implantada a partir de 22 de outubro de 2021 no estado de Goiás com o valor de R$ 0,637/kWh, de acordo com a Enel (2021). Portanto, o preço do consumo entre os dias 20 e 23 de novembro de 2021, Figura 38, resultou em R$ 0,15.

O segundo *card* da tela, Figura 38, exibiu dois botões "*ON*" e "*OFF*" que foram usados para habilitar e desabilitar a tomada. Desta forma, os valores de leituras que apresentaram 0 kW de potência consumida, foram obtidos quando a tomada foi desabilitada através do botão "*OFF*" aplicativo. E os valores de consumo maiores de 0 kW foram obtidos quando a tomada estava habilitada pelo botão "*ON*".

5.2 Considerações sobre os resultados obtidos

O protótipo armazenou os valores no banco de dados na nuvem, então foi observado um *delay*, pois depende tanto da velocidade da internet quanto do tempo que o ESP32 gasta para enviar e gravar os valores lidos no Firebase, para que depois faça a próxima leitura.

Portanto, as medições sofreram interferência e não foram calculados os valores exatos de consumo, pois enquanto estava no *delay* o protótipo não realizou a leitura da potência consumida pelo aparelho. Além disso, o controle do estado da tomada também ficou comprometido, pois se o usuário clicar em um dos botões para habilitar ou desabilitar a tomada, foi observado um atraso para realizar a ação.

Contudo, tanto o protótipo de monitoramento do consumo de energia elétrica quanto o aplicativo cumpriram com o objetivo do trabalho, visto que o protótipo realizou as medições e o aplicativo mostrou os valores de consumo de kWh por dia em um gráfico e o consumo total juntamente com o seu respectivo valor em reais. Além disso, apresentou uma tabela em que foram apresentados todos os valores de potência lida em kW, durante a realização do monitoramento que foi feito entre os dias 20 e 23 de novembro de 2021.

6 CONSIDERAÇÕES FINAIS

Este trabalho teve o objetivo de monitorar o consumo de energia elétrica de um aparelho eletroeletrônico de uma residência, então foram lidos os valores de tensão da rede elétrica e de corrente de um ferro elétrico, com isso, foi calculada a potência do mesmo. Os valores das leituras foram armazenados no banco de dados Firebase, juntamente com a data e o horário, o que tornou possível calcular o consumo em kWh por dia e o preço deste em reais.

Os resultados do monitoramento do consumo de energia do ferro elétrico, realizado entre os dias 20 e 23 de novembro de 2021, mostraram que o protótipo realizou as medições de tensão e corrente e calculou a potência do ferro elétrico. Os valores de potência calculados foram armazenados no Firebase juntamente com a data e a hora em que foi realizada a leitura.

O aplicativo Android fez a leitura destes valores gravados no banco de dados e calculou o consumo em kWh, com isso mostrou o gráfico de consumo por dia, o consumo total e o preço deste em reais, e também apresentou a tabela com os valores de potência em kW medidos a cada leitura. Outra funcionalidade do aplicativo foi o de controlar o estado da tomada, podendo habilitá-la ou desabilitá-la, através dos botões "*ON*" e "*OFF*".

Desta forma, foi possível monitorar o consumo de energia do aparelho conectado a esta tomada, possibilitando uma supervisão para se alcançar um controle dos gastos. Logo, o projeto facilitou saber o consumo de energia elétrica consumida por um aparelho em uma residência, e o preço deste consumo em reais.

6.1 Sugestões para trabalhos futuros

- Filtrar os valores medidos do gráfico de consumo e da tabela, por mês, intervalo de dias e intervalo de horas;
- Alterar a tarifa de consumo de energia de acordo com o estado de residência do usuário;
- Montar um módulo único para que seja instalado em uma residência.

REFERÊNCIAS

ALLEGRO MICROSYSTEMS. **ACS712**: Fully Integrated, Hall-Effect-Based Linear Current Sensor IC with 2.1 kVRMS Voltage Isolation and a Low-Resistance Current Conductor. 2020. Disponível em: <https://www.allegromicro.com/en/Products/Sense/current-sensor-ics/Zero-to-fifty-amp-integrated-conductor-sensor-ICs/ACS712.aspx>. Acesso em: 01 maio 2020.

AMARAL, Haroldo. **Medidor de corrente AC com ACS712 e Emonlib**. 2017. Disponível em: https://www.filipeflop.com/blog/medidor-de-corrente-ac-acs712-emonlib/. Acesso em: 14 set. 2021.

AMPIRE, Eric. **Dependency Injection with Koin**. 2019. Disponível em: < https://medium.com/swlh/dependency-injection-with-koin-6b6364dc8dba>. Acesso em: 20 ago. 2021.

ANDROID STUDIO. **Conheça o Android Studio**. 2020. Disponível em: <https://developer.android.com/studio/intro?hl=pt-br>. Acesso em: 09 maio 2020.

ANEEL. **Fatores de Conversão**. Disponível em: <http://www2.aneel.gov.br/arquivos/pdf/atlas_fatoresdeconversao_indice.pdf>. Acesso em: 11 abr. 2020.

ANEEL. **Por dentro da conta de luz**: informação de utilidade pública. 7. ed. Brasília: ANEEL, 2016a.

ANEEL. **Programa de Eficiência Energética.** 2016b. Disponível em: <https://www.aneel.gov.br/programa-eficiencia-energetica>. Acesso em: 08 mar. 2020.

ANEEL. **Receita Anual de Geração para o ciclo 2019/2020 é fixada.** 2019. Disponível em: <https://www.aneel.gov.br/sala-de-imprensa-exibicao-2/-/asset_publisher/zXQREz8EVlZ6/content/receita-anual-de-geracao-para-o-ciclo-2019-2020-e-fixada/656877?inheritRedirect=false >. Acesso em: 08 mar. 2020.

ANNE com e. Roteiro: Moira Walley-beckett. Ontário: NETFLIX, 2017. Son., color. Disponível em: <https://www.netflix.com/br/title/80136311>. Acesso em: 30 abr. 2020.

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 Software (IDE)**. 2015. Disponível em: <https://www.arduino.cc/en/guide/Environment#toc4>. Acesso em: 09 maio 2020.

ARDUINO. **Wifi library**. 2019. Disponível em: <https://www.arduino.cc/en/Reference/WiFi>. Acesso em: 13 dez. 2021.

ARTRON SHOP. **IOXhop_FirebaseESP32**. 2018. Disponível em: <https://github.com/artronshop/IOXhop_FirebaseESP32>. Acesso em: 13 dez. 2021.

AUTOCORE ROBÓTICA. **Módulo Sensor de Tensão AC 110/220V**. Disponível em: <https://www.autocorerobotica.com.br/modulo-sensor-de-tensao-ac-110220v>. Acesso em: 02 maio 2020.

BABAGIL, Alperen. **Android Clean App Base Library (Clean Architecture + MVVM)**. 2020. Disponível em: <https://medium.com/huawei-developers/android-clean-app-base-library-clean-architecture-mvvm-part-i-e8614978877f>. Acesso em: 20 ago. 2021.

BERTOLINI, Cristiano *et al.* **Linguagem de programação I.** Santa Maria: UFMS, NTE, 2019.

BISHT, Ankit. **MVVM (Model View ViewModel) + Kotlin + Google Jetpack**. 2019. Disponível em: <https://medium.com/@er.ankitbisht/mvvm-model-view-viewmodel-kotlin-google-jetpack-f02ec7754854>. Acesso em: 20 ago. 2021.

CIDADE JÚNIOR, Darci. **Análise do consumo de energia em uma unidade consumidora via Internet das Coisas**. 2018. 41 f. Monografia (Especialização) - Curso de Especialização em Internet das Coisas, Universidade Tecnológica Federal do Paraná, Curitiba, 2018.

CIPOLI, Pedro. **Diferenças entre velocidade de upload e download**. 2020. Disponível em: <https://canaltech.com.br/internet/diferencas-entre-velocidade-de-upload-e-download/>. Acesso em: 11 abr. 2020.

CURTO CIRCUITO. **Protocolo NTP para ESP32**. 2018. Disponível em: <https://www.curtocircuito.com.br/blog/Categoria%20IoT/protocolo-ntp-para-ESP32>. Acesso em: 13 dez. 2021.

DUTRA, Luís Fernando Giacomini. **Aplicando o conceito de banco de dados relacional e temporal no aplicativo Calendar Tattoo**. 2017. 41 f. Monografia (Especialização) - II Curso de Especialização em Banco de Dados, Universidade Tecnológica Federal do Paraná, Pato Branco, 2017.

ENCONTRO ANUAL DE TECNOLOGIA DA INFORMAÇÃO, 2018, Santa Maria. **Proposta de ambientes inteligentes IoT sob a ótica da eficiência energética.** Frederico Westphalen: Novatec, 2018. 1 v.

ENEL. **Aneel aprova reajuste nas tarifas da enel distribuição goiás**. 2021. Disponível em: <https://www.enel.com.br/content/dam/enel-br/megamenu/m%C3%ADdia/releases-

2021/10/Aneel%20aprova%20reajuste%20nas%20tarifas%20da%20Enel%20Distribui%C3%A7%C3%A3o%20Goi%C3%A1s.pdf>. Acesso em: 20 nov. 2021.

EPE. **Atlas da Eficiência Energética**. 3. ed. [S.l.: s.n.], 2019a.

EPE. **Plano Decenal de Expansão de Energia.** 2019b. Disponível em: <http://www.epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/PublicacoesArquivos/publicacao-52/topico-89/Relat%C3%B3rio%20Final%20do%20PDE%202019.pdf >. Acesso em: 01 jun. 2020.

ESPRESSIF. **ESP 32 Series**: *Datasheet*. 2020. Disponível em: <https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf>. Acesso em: 01 maio 2020.

EXATI. **ISO 37122: a resolução das cidades inteligentes.** 2019. Disponível em: <https://blog.exati.com.br/iso-37122-a-resolucao-das-cidades-inteligentes/>. Acesso em: 27 fev. 2020.

FERLIN, Edson Pedro; REZENDE, Denis Alcides. Big Data aplicado à cidade digital estratégica: estudo sobre o volume de dados das aplicações Smart City. **Revista Gestão & Tecnologia**, [s.l.], v. 19, n. 2, p.175-194, 1 abr. 2019. Fundação Pedro Leopoldo - Revista Gestão & Tecnologia. <http://dx.doi.org/10.20397/2177-6652/2019.v19i2.1533>.

FIREBASE. **Adicionar o Firebase ao projeto para Android**. 2019a. Disponível em: <https://firebase.google.com/docs/android/setup?hl=pt-br>. Acesso em: 09 maio 2020.

FIREBASE. **Estruturar o banco de dados**. 2019b. Disponível em: <https://firebase.google.com/docs/database/ios/structure-data?hl=pt-br>. Acesso em: 09 maio 2020.

FIREBASE. **Firebase Realtime Database**. 2019c. Disponível em: <https://firebase.google.com/docs/database?hl=pt-br>. Acesso em: 09 maio 2020.

FREITAS, Ariel Alves de; DIOGENES, Daniel Pablo Dantas. **Sistema IoT de monitoramento e gerenciamento de equipamentos elétricos em uma sala de estudos**. 2019. 12 f. TCC (Graduação) - Curso de Ciência e Tecnologia, Universidade Federal Rural do Semiárido, Mossoró, 2019.

FUTURE ELETRONIC CORPORATION. ***Relay Modules***. Disponível em: <https://www.curtocircuito.com.br/datasheet/modulo/rele-1canal.pdf>. Acesso em: 09 maio 2020.

JAYARAMAN, Prem Prakash et al. ***Privacy preserving Internet of Things****: From privacy techniques to a blueprint architecture and efficient implementation.* ***Future Generation***

Computer Systems, [s.l.], v. 76, p.540-549, nov. 2017. Elsevier BV. <http://dx.doi.org/10.1016/j.future.2017.03.001>.

KHAN, Mohammed Arman. ***Turn LED ON/OFF from Firebase using ESP32***. 2020. Disponível em: <https://github.com/MohammedArmanKhan/Turn-LED-ON-OFF-from-firebase-using-ESP32>. Acesso em: 24 out. 2021.

LONGO, Lucas. **Internet das Coisas: uso de sensores e atuadores na automação de um protótipo residencial**. 2015. 97 f. Tese (Graduação) - Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná, Pato Branco, 2015.

MAÇADA, Antonio Carlos Gastaud; CANARY, Vivian Passos. A Tomada de Decisão no Contexto do Big data: Estudo de caso único. **XXXVIII EnANPAD 2014**, Rio de Janeiro, 13-17 Setembro 2014.

MAESTRELLI, Giovani Andrey; NAPOLEÃO, Guilherme Szemczak. **Sistema supervisório para monitoramento de energia elétrica residencial**. 2018. 95 f. TCC (Graduação) - Curso de Engenharia Elétrica, Universidade Tecnológica Federal do Paraná, Curitiba, 2018.

MEDEIROS, Flaviani Souto Bolzan et al. *Internet of Things*: uma investigação do conhecimento científico em artigos acadêmicos na última década. **Revista Eletrônica de Administração e Turismo-ReAT**, v. 12, n. 7, p. 1652-1674, 2018.

MENDONÇA, Claudio Marcio Campos de; ANDRADE, António Manuel Valente de. **Uso da IoT, Big Data e Inteligência Artificial nas capacidades dinâmicas**: um estudo comparativo entre cidades do Brasil e de Portugal. **Informação & Sociedade**: Estudos, [s.l.], v. 29, n. 4, p.37-60, 29 nov. 2019. Portal de Periódicos UFPB. <http://dx.doi.org/10.22478/ufpb.1809-4783.2019v29n4.47755>.

MERCADO LIVRE. **Esp32 Desenvolvimento Wifi + Bluetooth 38 Pinos**. Disponível em: <https://produto.mercadolivre.com.br/MLB-1736009841-esp32-desenvolvimento-wifi-bluetooth-38-pinos-_JM?quantity=1>. Acesso em: 30 ago. 2021a.

MERCADO LIVRE. **Modulo Rele 1 Canal Led Indicador Para Arduino Pi Pic 5v/10a**. Disponível em: <https://produto.mercadolivre.com.br/MLB-1364870549-modulo-rele-1-canal-led-indicador-para-arduino-pi-pic-5v10a-_JM?quantity=1>. Acesso em: 30 ago. 2021b.

MERCADO LIVRE. **Sensor De Corrente ACS 712 30a Arduino Pic Raspberry**. Disponível em: <https://produto.mercadolivre.com.br/MLB-1157509305-sensor-de-corrente-acs712-30a-arduino-pic-raspberry-_JM?quantity=1>. Acesso em: 30 ago. 2021c.

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>. Acesso em: 19 set. 2021d.

MOURA, Maycon Jean de; MONTEIRO SOBRINHO, Francisco Magno. **Sistema para monitoramento de consumo de energia elétrica conectado à internet de baixo custo**. 2019. 13 f. TCC (Graduação) - Curso de Bacharelado em Ciência e Tecnologia, Universidade Federal Rural do Semiárido, Mossoró, 2019.

OLIVEIRA, Bruno Alberto Soares. **Desenvolvimento de um protótipo de sistema de monitoramento de energia elétrica via internet: um estudo de caso no IFMG-Campus Bambuí**. 2017. 94 f. TCC (Graduação) - Curso de Engenharia de Computação, Instituto Federal de Minas Gerais, Bambuí, 2017.

PEREIRA, Luiz Henrique Junior. **Monitoramento do consumo de energia elétrica e controle de equipamentos via aplicativo**. 2018. 62 f. TCC (Graduação) - Curso de Controle e Automação, Universidade Federal de Ouro Preto, Ouro Preto, 2018.

PRADO, Paterson Anunciação. **Sistema de monitoramento remoto de consumo de energia elétrica em residências**. 2018. 54 f. TCC (Graduação) - Curso de Engenharia Elétrica, Universidade do Estado do Amazonas, Manaus, 2018.

RAUTENBERG, Sandro; CARMO, Paulo Ricardo Viviurka do. Big Data e ciência de dados: complementaridade conceitual no processo de tomada de decisão. ***Brazilian Journal of Information Studies.*** São Paulo, p. 56-67. abr. 2019.

REIS, Raphael Barbosa. **Servidor web para monitoramento de corrente, temperatura e umidade em máquinas elétricas**. 2019. 52 f. TCC (Graduação) - Curso de e Engenharia de Controle e Automação, Universidade Federal de Ouro Preto, Ouro Preto, 2019.

REVISTA DE EFICIÊNCIA ENERGÉTICA. [S.l.: s.n.], jun. 2017.

SANTOS, Bruno P. et al. Internet das coisas: da teoria à prática. **Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, v. 31, 2016.

SARAVATI. **Sensor de Corrente ACS712 30A**. 2020. Disponível em: <https://www.saravati.com.br/sensor-de-corrente-acs712-30a>. Acesso em: 03 maio 2020.

SILVA, Monaliza Mara Holanda. **Desenvolvimento de uma interface para medição, análise e supervisão do consumo de energia elétrica.** 2019. 34 f. TCC (Graduação) - Curso de Engenharia Elétrica, Universidade Federal Rural do Semiárido, Mossoró, 2019.

SILVEIRA JUNIOR, Wander Paulo de; MOURA, Luiz Gustavo Lourenço. COMUNICAÇÃO IoT APLICADO À SAÚDE ATRAVÉS DE DISPOSITIVOS DE MONITORAMENTO PESSOAL. ***Interdisciplinary Scientific Journal.*** Campos dos Goytacazes, p. 74-87. set. 2018.

SUPERINTENDÊNCIA DE CONCESSÕES E AUTORIZAÇÕES DE GERAÇÃO. **Sistemas de Informações de Geração da ANEEL SIGA:** Capacidade de Geração do Brasil. 2020.

Disponível em: <https://app.powerbi.com/view?r=eyJrIjoiNjc4OGYyYjQtYWM2ZC00YjllLWJlYmEtYzdkNTQ1MTc1NjM2IiwidCI6IjQwZDZmOWI4LWVjYTctNDZhMi05MmQ0LWVhNGU5YzAxNzBlMSIsImMiOjR9>. Acesso em: 04 abr. 2020.

SUPERINTENDÊNCIA DE GESTÃO TARIFÁRIA. **Bandeiras Tarifárias.** 2019. Disponível em: <https://www.aneel.gov.br/bandeiras-tarifarias>. Acesso em: 08 mar. 2020.

TOLEDO, Lucas Henrique Bento. **Desenvolvimento de aplicações Android adaptáveis utilizando a linguagem Kotlin.** 2019. 34 f. TCC (Graduação) – Curso de Ciência da Computação, Universidade Federal de Ouro Preto, Ouro Preto, 2021.

TORABI, Golnaz. **Clean Architecture with MVVM**. 2020. Disponível em: < https://levelup.gitconnected.com/clean-architecture-with-mvvm-34cc05ab3bc5>. Acesso em: 20 ago. 2021.

URBAN SYSTEMS. ***Ranking connected smart cities*.** 2019. Disponível em: <https://www.urbansystems.com.br/rankingconnectedsmartcities>. Acesso em: 27 fev. 2020.

USINAINFO. **Programar ESP32 com a IDE Arduino**: Tutorial Completo. 2019. Disponível em: <https://www.usinainfo.com.br/blog/programar-esp32-com-a-ide-arduino-tutorial-completo/>. Acesso em: 10 maio 2020.

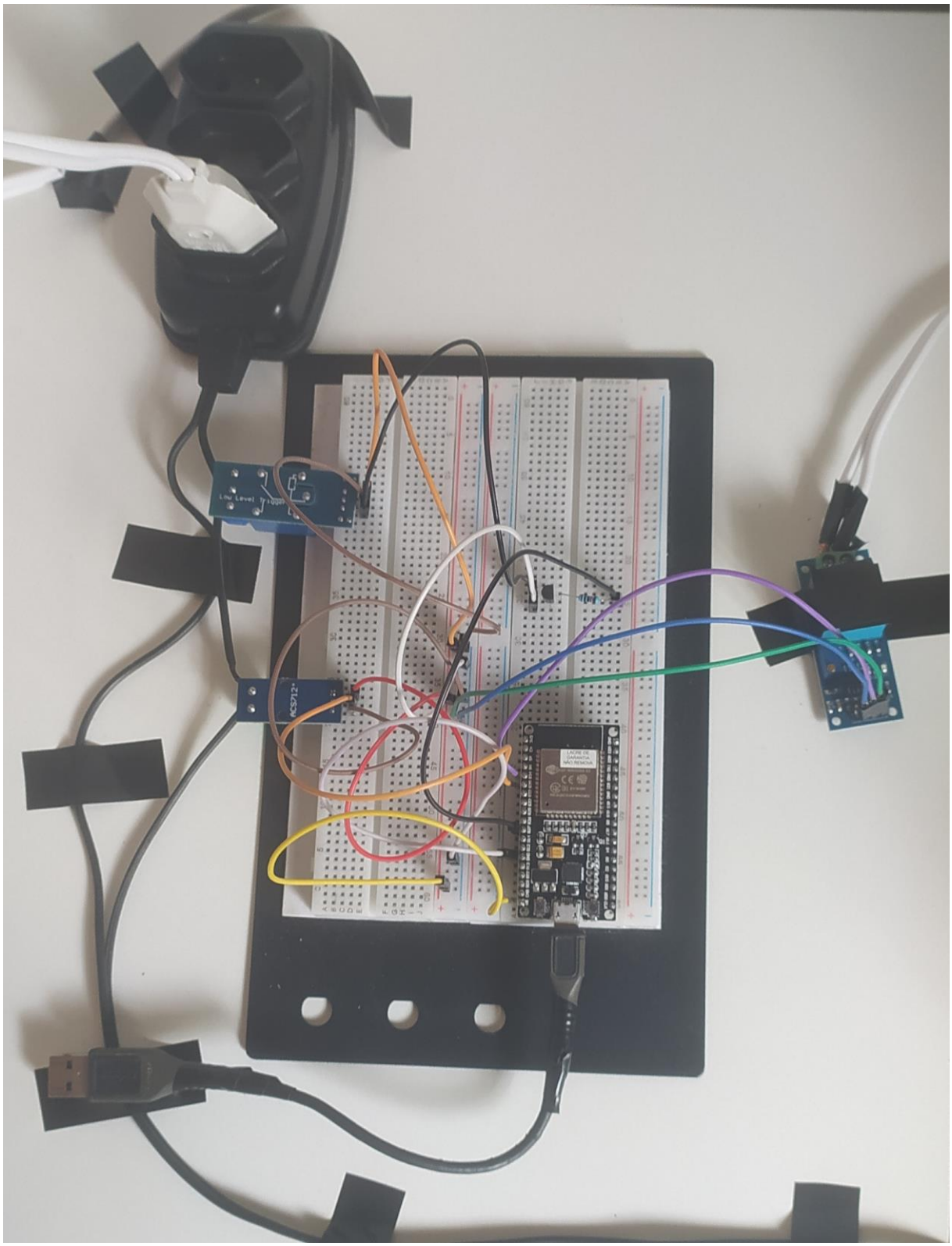
VENKATRAM, Kari; GEETHA, Mary A. ***Review on Big Data & Analytics – Concepts, Philosophy, Process and Applications***. ***Cybernetics And Information Technologies***, [s.l.], v. 17, n. 2, p.3-27, 1 jun. 2017. Walter de Gruyter GmbH. <http://dx.doi.org/10.1515/cait-2017-0013>.

APÊNDICE A – CÓDIGO PARA O MONITORAMENTO DO CONSUMO DE ENERGIA

kWh-monitoring

```
#include <WiFi.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include "EmonLib.h"
#include <IOXhop_FirebaseESP32.h>
#define   FIREBASE_HOST     "project-tcc-iot-default-rtdb.firebaseio.com/"
#define   FIREBASE_AUTH     "[redacted]"
#define   WIFI_SSID         "NET-V"
#define   WIFI_PASSWORD     "[redacted]"
#define   VOLT_CAL          97.5
#define   SAMPLING_TIME     0.0001668649
#define   LINE_FREQUENCY    60
#define   ACS_MPY           3.5

WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP);
String formattedDate;
String fireStatus = "";
EnergyMonitor emon1;
double Irms = 0;
float power = 0;

void setup() {
  Serial.begin(115200);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  timeClient.begin();
  timeClient.setTimeOffset(-10800);

  emon1.voltage(34, VOLT_CAL, 1.7);
  emon1.current(35, ACS_MPY);
  pinMode(27, OUTPUT);
}

void loop() {
  fireStatus = Firebase.getString("control_status");
  if (fireStatus == "ON") {
    digitalWrite(27, HIGH);
  }
  else if (fireStatus == "OFF") {
    digitalWrite(27, LOW);
  }

  emon1.calcVI(25,1000);
  float supplyVoltage = emon1.Vrms;
  if(supplyVoltage <= 100){
      supplyVoltage = 0;
  }

  Irms = emon1.calcIrms(1480);
  while(!timeClient.update()) {
    timeClient.forceUpdate();
  }
  formattedDate = timeClient.getFormattedDate();
  Firebase.pushString("date_time", formattedDate);
  power = Irms * supplyVoltage;
  Serial.print("Corrente: "); Serial.print(Irms); Serial.println(" A");
  Serial.print("Tensão: "); Serial.print(supplyVoltage);  Serial.println(" V");
  Serial.print("Potência: ");  Serial.print(power);  Serial.println(" W");
  Serial.println();
  Firebase.push("power", power);
  delay(10);
 }
```

APÊNDICE B – PROTÓTIPO DE MONITORAMENTO DO CONSUMO DE ENERGIA



ANEXO A – TERMO DE AUTORIZAÇÃO DE PUBLICAÇÃO DE PRODUÇÃO ACADÊMICA

PUC GOIÁS

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
GABINETE DO REITOR

Av. Universitária, 1069 ● Setor Universitário
Caixa Postal 86 ● CEP 74605-010
Goiânia ● Goiás ● Brasil
Fone: (62) 3946.1000
www.pucgoias.edu.br ● reitoria@pucgoias.edu.br

RESOLUÇÃO n° 038/2020 — CEPE

ANEXO I

APÊNDICE AO TCC

Termo de autorização de publicação de produção acadêmica

O (A) estudante Laryssa Salustiana de Oliveira Pires, do Curso de Engenharia de Computação, matrícula: 2016.1.0033.0098-3, telefone: (62) 98264-5730, e-mail: laryssa_olvgyn@hotmail.com, na qualidade de titular dos direitos autorais, em consonância com a Lei n 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 Desenvolvimento de um aplicativo Android para monitoramento do consumo de energia elétrica de uma residência, 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, 10 de dezembro de 2021.

Assinatura do(s) autor(es): Laryssa Salustiana de Oliveira Pires

Nome completo do autor: Laryssa Salustiana de Oliveira Pires

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

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