



Universidade Federal de Ouro Preto
Escola de Minas
CECAU - Colegiado do Curso de
Engenharia de Controle e Automação



Guilherme Iannini Dutra

Sistema de Monitoramento de Consumo de Energia Residencial e de Acionamento Remoto

Monografia de Graduação

Ouro Preto, 2025

Guilherme Iannini Dutra

Sistema de Monitoramento de Consumo de Energia Residencial e de Acionamento Remoto

Trabalho apresentado ao Colegiado do Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro(a) de Controle e Automação.

Universidade Federal de Ouro Preto

Orientador: Prof(a). Dra. Adrielle de Carvalho Santana

Ouro Preto

2025



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA CONTROLE E
AUTOMACAO



FOLHA DE APROVAÇÃO

Guilherme Iannini Dutra

Sistema de Monitoramento de Consumo de Energia Residencial e de Acionamento Remoto

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Engenharia de Controle e Automação

Aprovada em 16 de maio de 2025

Membros da banca

Dra. Adrielle de Carvalho Santana - Orientadora - Universidade Federal de Ouro Preto
Dr. Alan Kardek Rego Segundo - Convidado - Universidade Federal de Ouro Preto
Dr. Caio Fernando Teixeira Portela - Convidado - Universidade Federal de Ouro Preto

Adrielle de Carvalho Santana, orientadora do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 20/05/2025



Documento assinado eletronicamente por **Adrielle de Carvalho Santana, PROFESSOR DE MAGISTERIO SUPERIOR**, em 21/05/2025, às 17:51, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0912284** e o código CRC **3E40D723**.

Agradecimentos

Agradeço imensamente pela oportunidade de ter estudado no Curso de Engenharia de Controle e Automação na UFOP, Todos os conhecimentos e desafios adquiridos na Escola de Minas foram experiências inesquecíveis, cheia de altos e baixos.

Agradeço aos familiares e amigos por me acompanharem em minha jornada e por todo apoio, paciência e encorajamento ao longo da minha graduação. destacando pessoas como Arthur Gondim, Daniele Eva Cota Coluna, Junior campos, Raphael Iannini Dutra e minha mãe Andrea Maria Iannini Dutra.

Agradeço aos professores por todo o conhecimento adquirido, pela paciência, amizade e profissionalismo. um agradecimento especial a Prof(a). Dra. Adrielle de Carvalho Santana, por se disponibilizar como minha orientadora, toda a ajuda, apoio e encorajamento foi fundamental durante a realização deste projeto.

*Tudo é teoricamente impossível,
até que seja feito.*

— Robert A. Heinlein.

Resumo

Equipamentos modernos de todos os tipos estão cada vez mais presentes, tanto na indústria quanto na vida do cidadão comum, dessa forma, ter ciência e controle da quantidade de energia tem se tornado cada vez mais relevante. O presente trabalho consistiu no desenvolvimento de um sistema de monitoramento e controle de gasto energético controlado por ESP32, um microcontrolador que oferece diversas soluções devido à sua capacidade de ser programado para executar diferentes funções, integrando sensores e atuadores ao projeto, além de fornecer conexão via Wi-Fi e Bluetooth. O sistema conta com um sensor de corrente ACS712 20A, que realiza a medição da quantidade de corrente elétrica utilizada pelo equipamento. As informações coletadas são enviados ao uma base de dados, desenvolvida na plataforma FireBase e a um aplicativo mobile desenvolvido na plataforma Kodular. O aplicativo informa ao usuário a quantidade de energia consumida, o respectivo custo monetário, e possui opção para desligar o equipamento de forma remota, utilizando um relé como atuador. Testes do sistema com lâmpadas e um amperímetro permitiram demonstraram precisão e confiabilidade do sistema, que apresentou um erro médio de 5%.

Palavras-chaves: ESP32, Energia, Monitoramento, Controle, Consumo.

Abstract

Modern equipment of all kinds is becoming increasingly present, both in industry and in the daily life of ordinary citizens. As a result, being aware of and having control over energy consumption has become increasingly important. This work consisted of the development of an energy consumption monitoring and control system managed by an ESP32, a microcontroller that offers various solutions due to its ability to be programmed to perform different functions, integrating sensors and actuators into the project, as well as providing Wi-Fi and Bluetooth connectivity. The system includes an ACS712 20A current sensor, which measures the amount of electric current used by the equipment. The collected data is sent to a database developed on the Firebase platform and to a mobile application developed on the Kodular platform. The application informs the user about the amount of energy consumed, the corresponding monetary cost, and includes an option to remotely turn off the equipment using a relay as an actuator. System tests using light bulbs and an ammeter demonstrated the system's accuracy and reliability, with an average error within 5%.

Key-words: ESP32, Energy, Monitoring, Control, Consumption.

Lista de ilustrações

Figura 1 – Placa ESP32. Fonte: Systems (2025).	19
Figura 2 – Pinagem da placa ESP32. Fonte: Systems (2025).	20
Figura 3 – Tela de criação Blocos. Fonte: Autoria Própria	22
Figura 4 – Imagem auxiliar efeito Hall. Fonte: Instituto de Física - UFRGS (2025)	23
Figura 5 – Sensor de Corrente ACS712 Fonte: Robocore (2024b).	24
Figura 6 – Esquema elétrico ACS712 Fonte: Allegro MicroSistens (2006).	25
Figura 7 – Relé utilizado no projeto. Fonte: Robocore (2024a).	25
Figura 8 – Fonte regulável para microcontrolador. Fonte: (ROBOCORE, 2024).	26
Figura 9 – Circuito divisor de tensão. Fonte:Arduino e Cia (2025)	27
Figura 10 – Alteração de regras Firebase, Imagem 1. Fonte: Autoria Própria.	30
Figura 11 – Alteração de regras Firebase, Imagem 1. Fonte: Autoria Própria.	30
Figura 12 – Estrutura do banco de dados, Imagem 1. Fonte: Autoria Própria.	30
Figura 13 – Estrutura do banco de dados, Imagem 2. Fonte: Autoria Própria.	31
Figura 14 – Logo do aplicativo. Fonte: Criado por Daniele Eva Cota Coluna	32
Figura 15 – Tela de abertura. Fonte: Autoria Própria.	34
Figura 16 – Tela Menu. Fonte: Autoria Própria.	35
Figura 17 – Tela Configurar Equipamentos. Fonte: Autoria Própria.	36
Figura 18 – Tela Histórico. Fonte: Autoria Própria.	37
Figura 19 – Tela Monitorar Consumo. Fonte: Autoria Própria.	38
Figura 20 – Tela de Informações. Fonte: Autoria Própria.	39
Figura 21 – Circuito demonstrativo. Fonte: Autoria Própria.	40
Figura 22 – Montagem física do protótipo. Fonte: Autoria Própria.	41
Figura 23 – Montagem física do protótipo. Fonte: Autoria Própria.	42
Figura 24 – Gráfico respostas ADC do ESP32. Fonte: Autoria Própria.	44
Figura 25 – Montagem para testes corrente alternada. Fonte: Autoria Própria.	46
Figura 26 – Montagem para testes corrente alternada. Fonte: Autoria Própria.	47
Figura 27 – Gráfico obtido no teste de corrente alternada. Fonte: Autoria Própria.	49
Figura 28 – Gráfico obtido nos testes corrente contínua. Fonte: Autoria Própria.	50
Figura 29 – Tutorial configuração IDE, imagem 1 Fonte: Autoria Própria.	64
Figura 30 – Tutorial configuração IDE, imagem 2 Fonte: Autoria Própria.	65
Figura 31 – Tutorial configuração IDE, imagem 3 Fonte: Autoria Própria.	66
Figura 32 – Tutorial configuração IDE, imagem 4. Fonte: Autoria Própria.	66
Figura 33 – Chave de acesso. Fonte: Autoria Própria.	68
Figura 34 – Bibliotecas utilizadas na comunicação. Fonte: Autoria Própria.	68
Figura 35 – Explicação do passo 1, imagem 1. Fonte: Autoria Própria.	69
Figura 36 – Explicação do passo 1, imagem 2. Fonte: Autoria Própria.	69

Figura 37 – Explicação do passo 1, imagem 3. Fonte: Autoria Própria.	70
Figura 38 – Explicação do passo 2, imagem 1. Fonte: Autoria Própria.	70
Figura 39 – Explicação do passo 2, imagem 2. Fonte: Autoria Própria.	71
Figura 40 – Explicação do passo 3. Fonte: Autoria Própria.	71
Figura 41 – Download arquivo google-services.Json. Fonte: Autoria Própria.	72
Figura 42 – link identificação firebase. Fonte: Autoria Própria.	72
Figura 43 – Bloco Firebase DB. Fonte: Autoria Própria.	73
Figura 44 – Bloco Firebase DB. Fonte: Autoria Própria.	74
Figura 45 – Bloco Firebase DB. Fonte: Autoria Própria.	74
Figura 46 – Programação de blocos da Tela inicial do app. Fonte: Autoria Própria.	75
Figura 47 – Programação de blocos do Menu. Fonte: Autoria Própria.	75
Figura 48 – Programação de blocos tela configurar equipamentos (Imagem 1). Fonte: Autoria Própria.	76
Figura 49 – Programação de blocos tela configurar equipamentos (Imagem 2). Fonte: Autoria Própria.	76
Figura 50 – Programação de blocos tela configurar equipamentos (Imagem 3). Fonte: Autoria Própria.	77
Figura 51 – Programação de blocos tela configurar equipamentos (Imagem 4). Fonte: Autoria Própria.	78
Figura 52 – Programação de blocos tela configurar equipamentos (Imagem 5). Fonte: Autoria Própria.	78
Figura 53 – Programação de blocos tela histórico (Imagem 1). Fonte: Autoria Própria.	79
Figura 54 – Programação de blocos tela histórico (Imagem 2). Fonte: Autoria Própria.	79
Figura 55 – Programação de blocos tela histórico (Imagem 3). Fonte: Autoria Própria.	80
Figura 56 – Programação de blocos tela histórico (Imagem 4). Fonte: Autoria Própria.	81
Figura 57 – Programação de blocos tela histórico (Imagem 5). Fonte: Autoria Própria.	81
Figura 58 – Programação de blocos tela histórico (Imagem 6). Fonte: Autoria Própria.	82
Figura 59 – Programação de blocos tela histórico (Imagem 7). Fonte: Autoria Própria.	82
Figura 60 – Programação de blocos tela histórico (Imagem 8). Fonte: Autoria Própria.	83
Figura 61 – Programação de blocos tela histórico (Imagem 9). Fonte: Autoria Própria.	83
Figura 62 – Programação de blocos Monitorar Consumo (Imagem 1). Fonte: Autoria Própria.	84
Figura 63 – Programação de blocos Monitorar Consumo (Imagem 2). Fonte: Autoria Própria.	84
Figura 64 – Programação de blocos Monitorar Consumo (Imagem 3). Fonte: Autoria Própria.	85
Figura 65 – Programação de blocos Monitorar Consumo (Imagem 4). Fonte: Autoria Própria.	86

Figura 66 – Programação de blocos Monitorar Consumo (Imagem 5). Fonte: Autoria Própria.	87
Figura 67 – Programação de blocos Monitorar Consumo (Imagem 6). Fonte: Autoria Própria.	88
Figura 68 – Programação de blocos Monitorar Consumo (Imagem 7). Fonte: Autoria Própria.	88
Figura 69 – Programação de blocos Monitorar Consumo (Imagem 8). Fonte: Autoria Própria.	89
Figura 70 – Programação de blocos Monitorar Consumo (Imagem 9). Fonte: Autoria Própria.	90
Figura 71 – Programação de blocos Monitorar Consumo (Imagem 10). Fonte: Autoria Própria.	91

Lista de abreviaturas e siglas

AC	<i>Alternating Current</i> (Corrente Alternada)
A	Ampère
CA	Corrente Alternada
CC	Corrente Contínua
DC	Direct Current (Corrente Contínua)
IDE	Integrated Development Environment (Ambiente de Desenvolvimento Integrado)
ILM	Intrusive Load Monitoring (Monitoramento Intrusivo de Carga)
I	Corrente
IHM	Interface Homem-Máquina
IOT	<i>Internet of Things</i> (Internet das Coisas)
LEAD	Laboratório de Eletrônica Analógica e Digital
NA	Normalmente Aberto
NF	Normalmente Fechado
P	Potência
R	Resistência
RAM	<i>Random Access Memory</i> (Memória de Acesso Aleatório)
RMS	<i>Root Mean Square</i> (Raiz Média Quadrática)
SoC	<i>Security Operations Center</i> (Centro de Operações de Segurança)
TWh	Terawatt-hora
URL	<i>Uniform Resource Locator</i> (Localizador Uniforme de Recursos)
USB	<i>Universal Serial Bus</i> (Barramento Serial Universal)
V	Volt
Wi-Fi	<i>Wireless Fidelity</i> (Fidelidade Sem Fio)

Sumário

1	INTRODUÇÃO	13
1.1	Contextualização	13
1.1.1	Sistemas Embarcados	14
1.1.2	IOT	14
1.1.3	Automação residencial	15
1.2	Objetivos	15
1.2.1	Objetivo geral	15
1.2.2	Objetivos específicos	15
1.3	Justificativas e relevância	16
1.4	Organização do texto	16
2	REVISÃO DA LITERATURA	17
2.1	Trabalhos relacionados	17
2.2	Microcontroladores	18
2.2.1	Esp32	19
2.3	Tecnologia Wi-Fi	20
2.4	Softwares	21
2.4.1	FireBase	21
2.4.2	Plataforma Kodular	21
2.4.3	Plataforma Arduino IDE	21
2.5	Sensores para medição de corrente e Efeito Hall	23
2.6	Estratégia de medição de corrente: invasivo e não invasivo	24
2.7	Sensor de corrente ACS712	24
2.8	Relé	25
2.9	Fonte de alimentação	26
3	METODOLOGIA	27
3.1	Calculo de potência e corrente	27
3.1.1	Divisor de tensão	27
3.1.2	Método de cálculo	28
3.2	Configuração do relé	29
3.3	Banco de dados Firebase	29
3.3.1	Criação do banco de dados	29
3.3.2	Estrutura do banco de dados	30
3.4	Aplicativo do sistema de medição de energia (S.M.E)	32
3.4.1	Funções do aplicativo	32

3.4.2	Telas do aplicativo	32
3.5	Montagem do protótipo	40
3.6	Testes e calibração do sensor	43
3.6.1	Análise de linearidade do conversor analógico - digital do ESP32	43
3.6.2	Montagem para realização de testes	45
3.6.3	Testes corrente alternada	48
3.6.4	Testes correntes contínua	49
3.6.5	Calibração do sensor	51
4	RESULTADOS	52
5	CONSIDERAÇÕES FINAIS	54
5.1	Conclusão	54
5.2	Melhorias futuras	55
	Referências	56
	ANEXOS	59
	ANEXO A – DATASHEET DO SENSOR ACS712	60
	ANEXO B – DATASHEET DO RELÉ	61
	APÊNDICES	62
.1	código desenvolvido Arduino IDE.	63
.2	Tutorial de configurações	64
.2.1	Adaptação da Arduino IDE para o ESP32	64
.2.2	Configuração do Wi-Fi	67
.2.3	Comunicação entre Firebase e Arduino IDE	67
.2.4	Comunicação entre Firebase e Kodular	68
.3	Programação de blocos Kodular	75
.3.1	Tela Inicial	75
.3.2	Tela do Menu	75
.3.3	Tela Configurar Equipamentos	76
.3.4	Tela Histórico	79
.3.5	Monitorar Consumo	81

1 Introdução

1.1 Contextualização

A dependência da sociedade com relação à energia elétrica tem se ampliado com o avanço da ciência. De acordo com [Revista Ferramental \(2022\)](#)

“ A Terceira Revolução Industrial é o período posterior à Segunda Guerra Mundial, no qual o aprimoramento e os avanços tecnológicos começaram a englobar o campo da ciência e o incluindo no sistema produtivo. Esse período da Revolução Industrial também é chamado de Revolução Técnico Científica Informacional”

Essa revolução transformou profundamente o modo como a sociedade produz, consome e distribui bens e serviços. Esses avanços começaram a se desenvolver dentro das indústrias, com a integração de novas tecnologias e equipamentos mais sofisticados que permitiram, não apenas o aumento de produção, mas também a criação de novos produtos e realização de novos serviços que antes não eram possíveis. Contudo, tais mudanças se estenderam a toda a sociedade, tornando as pessoas cada vez mais dependentes das novas tecnologias, seja na área da medicina com a criação de novos equipamentos que auxiliam nos diagnósticos de pacientes, ou até mesmo no dia a dia do cidadão comum, com o uso de geladeiras, forno de micro-ondas, smartphones.

Assim, mostrou-se necessário também aumentar cada vez mais a produção de energia elétrica para atender às novas demandas, sejam elas fontes não renováveis (petróleo, carvão, gás natural e energia nuclear) ou renováveis (solar, eólica, hidrelétrica), bem como a criação de técnicas e equipamentos que auxiliem na economia de energia. Analisando dados mais atuais no Brasil, houve um aumento na geração de energia no país, a oferta interna de energia proveniente de fontes de energia renováveis no ano de 2013 era de 120.640 TWh, em comparação ao ano de 2022, o valor aumentou para 143.559 TWh ([EPE, 2024a](#)). Portanto é relevante pensar também em formas de monitorar o consumo de energia tanto em residências quanto em empresas. Estratégias deste tipo ajudam no planejamento econômico e também na percepção de onde o consumo de energia é mais evidente.

1.1.1 Sistemas Embarcados

O conceito de sistemas embarcados teve início por volta de 1960, na época em que existia um programa para controle de telefones, utilizando a linguagem de programação *Assembler*. Este tipo de estratégia foi adaptada para outros dispositivos, sem modificar o código base, adaptando apenas os sinais de entrada e saída definidos no programa para os novos equipamentos. Posteriormente na década de 1970, foram desenvolvidas bibliotecas e códigos para o desenvolvimento de sistemas embarcados ([CHASE; ALMEIDA, 2007](#)).

Atualmente, sistemas embarcados são conhecidos como circuitos integrados que possuem capacidade computacional e são utilizados em equipamentos. São criados circuitos eletrônicos quem atendem a demandas específicas, que dependem da aplicação. Tais circuitos podem conter microcontroladores com códigos que podem ser desenvolvidos em diferentes linguagens de programação. Além disso, são utilizados sensores que captam e interpretam diferentes informações e atuadores que realizam diferentes tarefas. As informações relevantes em cada aplicação são repassadas aos usuários utilizando alguma Interface Homem Máquina (IHM). Esse tipo de sistema é desenvolvido para não ser reprogramado e realizar uma ou mais tarefas específicas ([CUNHA, 2007](#)).

Alguns exemplos de equipamentos eletrônicos que utilizam sistemas embarcados incluem geladeiras, aparelhos de som, celulares e outros dispositivos. Sistemas embarcados podem atender a diferentes demandas devido a quantidade de recursos disponíveis, além de apresentarem soluções mais baratas em muitos casos, sendo assim, uma opção viável no monitoramento de energia, pois permitiria a medição e monitoramento de consumo de energia de forma prática, segura e moderna, ainda mais com a possibilidade da utilização de internet das coisas.

1.1.2 IOT

O conceito de Internet das Coisas é definido por [Santos, Silva et al. \(2016\)](#) como uma extensão da internet atual que proporciona a diferentes objetos a possibilidade de se conectarem à Internet, desde que esses objetos possuam capacidade computacional e de comunicação. A comunicação de diferentes dispositivos utilizando a internet é uma tendência cada vez maior na atualidade e é utilizada nas mais diversas áreas (economia, saúde, casas inteligentes), de acordo com [Carrion e Quaresma \(2019\)](#) dispositivos IoT estão alterando esferas diversas da sociedade, desde o gerenciamento doméstico e os cuidados com a saúde, à implementação nas indústrias, no planejamento urbano e de transportes, na geração de energia, dentre outros aspectos. A Internet das Coisas também tem sido amplamente utilizado na automação residencial, oferecendo diversas soluções inteligentes que promovem ampla praticidade no dia a dia.

1.1.3 Automação residencial

Automação residencial é um conceito relativamente novo. com início da década de 1970. Conforme ocorreram avanços na área de tecnologia, cada vez mais dispositivos têm se conectado à internet, oferecendo comunicação entre os mesmos o que gera diferentes possibilidades. De acordo com [Accardi e Dodonov \(2012\)](#) a Automação Residencial é também conhecida como Domótica, Residência Inteligente ou Casa do Futuro, cujo objetivo é tornar uma residência automatizada, através da integração de serviços e tecnologia, fornecendo conforto, praticidade e segurança. A automação residencial engloba diferentes áreas, conforme explicado por [Muratori e Dal Bó \(2011\)](#), algumas áreas citadas são:

- Instalação elétrica, que compreende: iluminação, persianas e cortinas, gestão de energia e outros;
- Sistema de segurança: alarmes de intrusão, alarmes técnicos (fumaça, vazamento de gás, inundação), circuito fechado de TV, monitoramento, controle de acesso;
- Sistemas multimídia: áudio e vídeo, som ambiente, jogos eletrônicos, além de vídeos, imagens e sons sob demanda;
- Sistemas de comunicações: telefonia e interfonia, redes domésticas, TV por assinatura;
- Utilidades: irrigação, aspiração central, climatização, aquecimento de água, bombas e outros.

1.2 Objetivos

1.2.1 Objetivo geral

Criar um sistema de monitoramento de gasto energético e acionamento de equipamentos, controlado por sistema embarcado.

1.2.2 Objetivos específicos

1. Construir um circuito que permita ao microcontrolador obter as informações de leitura de um sensor de corrente, para então calcular qual é o gasto de energia elétrica do equipamento monitorado, a potência consumida e o valor monetário do consumo.
2. Desenvolver um aplicativo para smartfone de uso simples, a fim de facilitar a utilização do sistema de monitoramento.
3. Desenvolver uma base de dados que irá salvar as informações processadas pelo sistema de medição de corrente.

4. Desenvolver a comunicação entre o Microcontrolador, a base de dados e o aplicativo.
5. Configurar o sistema para habilitar e desabilitar o uso do equipamento com um comando enviado por meio do aplicativo desenvolvido.

1.3 Justificativas e relevância

A demanda por energia no Brasil tem aumentado ao longo dos anos. Devido à criação recorrente de novas tecnologias e aumento da população, de acordo com LimaII (2004) “O consumo total de energia elétrica passou de 18.346 GWh em 1960 para 304.634 GWh em 2000, o que representa uma taxa de crescimento média anual no período de 7,35%”. Analisando dados mais atuais, tem-se que “O consumo de energia elétrica das residências no país foi de 46.242 no primeiro trimestre de 2024, com alta de 12,3% na comparação a igual trimestre de 2023” (EPE, 2024b).

Portanto, é relevante a criação de novas soluções inteligentes que possam amenizar o problema da alta demanda energética, uma vez que um sistema de análise de gastos de energia fornece ao consumidor informações que permitem a organização e controle do consumo de energia de cada equipamento em sua residência, podendo diminuir o consumo de energia e conseqüentemente seus custos financeiros. Ademais, ao monitorar o consumo de energia em uma residência, o sistema também ajudaria a detectar se há algum consumo indevido de energia, como os causados por fuga de corrente.

1.4 Organização do texto

No intuito de facilitar a organização e desenvolvimento, o presente trabalho foi separado em cinco capítulos distintos. O primeiro capítulo contextualiza o trabalho, explicando conceitos importantes e dados relevantes que constituem o problema abordado bem como a solução apresentada. O segundo capítulo apresenta a revisão de literatura realizada para a criação do trabalho, apresentando trabalhos relacionados com soluções similares, *hardwares*, *softwares* e demais ferramentas utilizadas na criação do protótipo que busca solucionar o problema proposto. O terceiro capítulo apresenta a metodologia utilizada na criação do trabalho, demonstrando os materiais e tecnologias selecionadas para a criação do trabalho, bem como alguns testes realizados e a calibração do sensor. No quarto capítulo são apresentados os resultados obtidos após a calibração do sensor bem como demais resultados alcançados pelo protótipo. Por fim, no quinto capítulo, são apresentadas as conclusões após a realização do trabalho, bem como uma lista de melhorias futuras.

2 Revisão da Literatura

2.1 Trabalhos relacionados

Esta etapa consistiu em pesquisas referentes a trabalhos semelhantes que adotam diferentes estratégias para o monitoramento de corrente elétrica utilizando microcontroladores a fim de auxiliar na escolha da estratégia de medição de corrente e auxiliar na escolha de ferramentas e recursos disponíveis.

No artigo “Monitoramento, Automação e Eficiência Energética, em Residência Familiar, para Redução da Conta de Energia Elétrica” [Moreno \(2021\)](#), o autor desenvolve o trabalho utilizando duas táticas distintas para o cálculo do gasto energético:

- **Monitoramento de Carga intrusiva** Consiste em medir a potência de cada equipamento diretamente no ponto em que está conectado a rede elétrica, fornecendo assim dados precisos, mas possui custo elevado e uma implementação mais complexa [Moreno \(2021\)](#), apud [Medeiros et al. \(2018\)](#).
- **Monitoramento de Carga Não intrusiva** Consiste em medir a potência de todos os equipamentos presentes na residência analisando um só ponto da rede elétrica. É um método mais simples e barato, porém não específica com precisão qual a potência de casa equipamento individualmente.

Ao mesclar ambas as técnicas em um período de 24 horas e analisar as contas elétricas da residência o autor conseguiu identificar os pontos com maior consumo de energia elétrica e automatizar o controle do uso de cargas não utilizadas ao longo do dia. O artigo inspirou a ideia de utilizar o método invasivo de medição de corrente elétrica no presente trabalho, uma vez que essa estratégia fornece dados específicos para cada equipamento monitorado.

O artigo “Protótipo Trifásico para Monitoramento de Consumo de Energia Elétrica Utilizando A Plataforma Arduino” de [Santos, Mendes e Martins \(2021\)](#) desenvolve a criação de um protótipo de monitoramento de energia elétrica controlado por Arduino, utilizando sensores não invasivos, uma estratégia mais abrangente e menos precisa, que fornece uma estimativa do gasto em tempo real. No desenvolvimento do projeto, os autores criam os programas embarcados na plataforma Arduino, primeiramente elaborando um protótipo inicial considerando apenas cargas monofásicas para posteriormente realizar a construção do protótipo final para medição do consumo de energia em sistemas trifásicos.

Os testes de ambos protótipos foram realizados em laboratórios e obtiveram resultados muito próximos aos valores teóricos esperados (SANTOS; MENDES; MARTINS, 2021). O artigo ilustrou possibilidade de se utilizar sistemas embarcados no monitoramento de energia elétrica, trazendo inspiração na utilização deste tipo de sistema no presente trabalho, adotando a estratégia de medição de corrente intrusiva.

2.2 Microcontroladores

Com o surgimento dos sistemas embarcados, ficou mais simples e barato criar dispositivos inteligentes com diferentes funções, esses circuitos são incorporados dentro de aparelhos eletrônicos e realizam funções específicas em cada caso. Nesse sentido, é necessário discutir a diferença entre microcontrolador e microprocessador. De acordo com Silva, Araujo e Cavalcante (2019), microprocessadores são considerados o cérebro de um sistema computacional. São circuitos integrados, dedicados à realização de cálculos, execução de instruções e tomadas de decisão, utilizam *Memória RAM (Random Access Memory)*, com dispositivos de entrada e saída externos. Já o microcontrolador por outro lado é um circuito integrado considerado um computador completo em miniatura. Dentro de um microcontrolador tem-se uma unidade de processamento e diversos outros componentes internos que executam tarefas conforme sua programação, como memórias para leitura de dados, conversores de sinal analógico/ digital, entradas e saídas programáveis que irão variar conforme os sensores e atuadores disponíveis no projeto (SILVA; ARAUJO; CAVALCANTE, 2019).

Hoje tem-se diferentes versões de microcontroladores com os mais diversos sensores e atuadores, como por exemplo, sensor de temperatura, corrente, luminosidade, relés, motores de passo, dentre outros. Tais dispositivos se tornaram populares devido a sua facilidade de programação, grande diversidade de aplicações e baixo custo.

2.2.1 Esp32

O ESP32 é um microcontrolador de código aberto baseada em *hardware* e *software* definido como um SoC (*software Security Operations Center*) fáceis de usar. O ESP32 pode ser programado utilizando a plataforma de código aberto Arduino IDE, uma plataforma de código aberto com diversos exemplos de códigos para programar o ESP32. Devido a quantidade de atuadores e sensores disponíveis, o ESP32 é considerado uma ferramenta extremamente versátil com diversas aplicações e baixo custo ([ARDUINO.CC](https://www.arduino.cc), 2024). Esse microcontrolador possui conexão Wi-fi e Bluetooth, e sua tensão de operação está na faixa de 2,3 a 3,2 V. Ele será o responsável por processar as informações coletadas pelo sensor de corrente no projeto, bem como enviar o sinal ao relé para o bloqueio de corrente. A Figura 1 demonstra algumas informações à respeito do pinos do ESP32, a Figura 2 complementa essas informações.

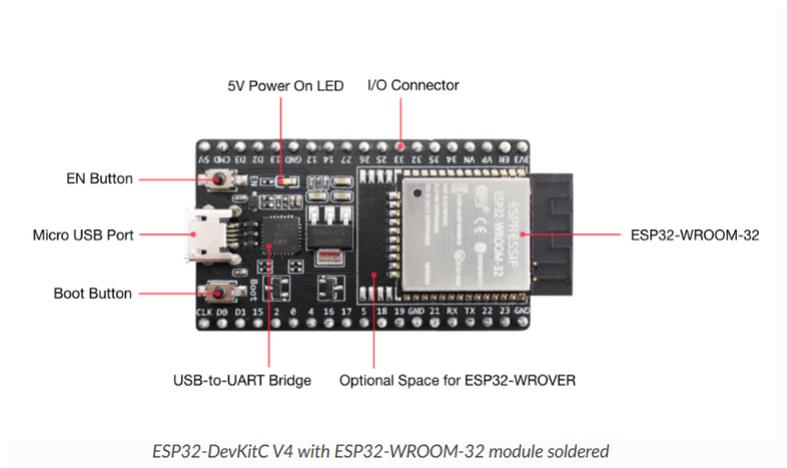


Figura 1 – Placa ESP32. Fonte: [Systems \(2025\)](#).

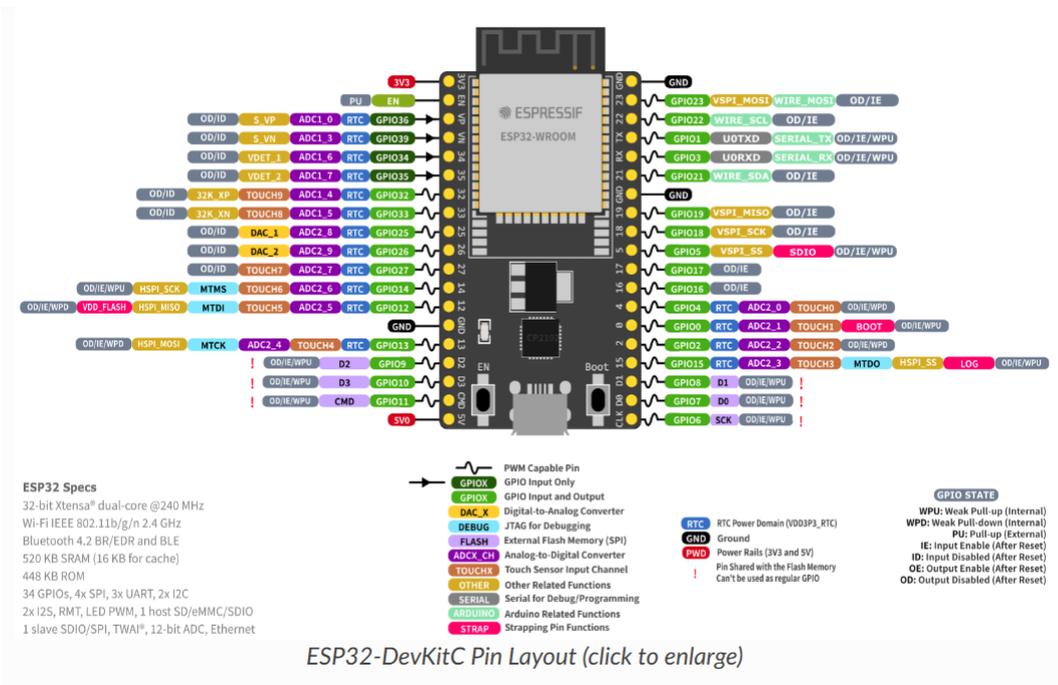


Figura 2 – Pinagem da placa ESP32. Fonte: [Systems \(2025\)](#).

2.3 Tecnologia Wi-Fi

O termo Wi-Fi significa “*Wireless Fidelity*” ou “fidelidade sem fios”. Refere-se a uma rede de comunicação entre dispositivos que utilizam ondas de rádio para transmitir informações de forma remota. A tecnologia Wi-Fi utiliza o padrão IEE 802.11 e frequências de 2,4 GHz, 5 GHz e 6 GHz para transmitir dados e permite a criação de redes domésticas ou empresárias, utilizando roteadores que distribuem sinais de internet utilizando essa tecnologia ([LUCAS BRAGA E ANA MARQUES, 2025](#)). A fim de promover o Wi-Fi, foi fundada a Wi-Fi Alliance, uma organização sem fins lucrativos fundada em 1999 por um grupo de empresas líderes no setor de tecnologia que se uniram para promover a tecnologia Wi-Fi e certificar os produtos que atendiam aos padrões estabelecidos para garantir a qualidade e segurança ([EXPRESS VPS, 2025](#)).

Para ter uma rede Wi-Fi, é necessário ter um roteador. Esse equipamento se conecta ao modem da operadora de internet via cabo de rede e é responsável por dividir a conexão com outros dispositivos, que podem se conectar ao roteador via Wi-Fi ou também via cabo de rede ([EXPRESS VPS, 2025](#)).

2.4 Softwares

2.4.1 FireBase

O Firebase é uma plataforma digital de desenvolvimento de aplicativos lançada pelo Google. A plataforma permite criar e expandir apps para diferentes ambientes como Android, iOS e para a Web, com diversos recursos que facilitam a criação de aplicações e as tornam mais seguras. Possui diversos recursos como, Realtime Database (banco de dados hospedado em nuvem), Authentication (facilita o desenvolvimento de um sistema de autenticação seguro), Crashlytics é (ferramenta que envia relatório de falhas em tempo real) dentre outros. O firebase é considerado como um Backend as a Service (BaaS), ou seja, o usuário consegue utilizar o sistema com mais facilidade, pois a infraestrutura interna necessária para o funcionamento da aplicação é fornecida (REMESSA ONLINE, 2023). O Firebase, apesar de possuir serviços gratuitos, possui limitações quanto ao número de envios de dados e consultas, contudo, para aplicações mais simples, consegue fornecer todas as ferramentas necessárias.

2.4.2 Plataforma Kodular

O Kodular é uma plataforma que oferece diversas ferramentas para o desenvolvimento de aplicativos, de forma online e gratuita. A aplicação permite a criação de aplicativos compatíveis com a plataforma Android utilizando a programação de blocos.

Trata-se de uma forma mais simples e intuitiva de programar que não necessita de conhecimentos prévios por parte do usuário, logo é possível criar aplicações de forma rápida e simples utilizando blocos lógicos e condicionais.

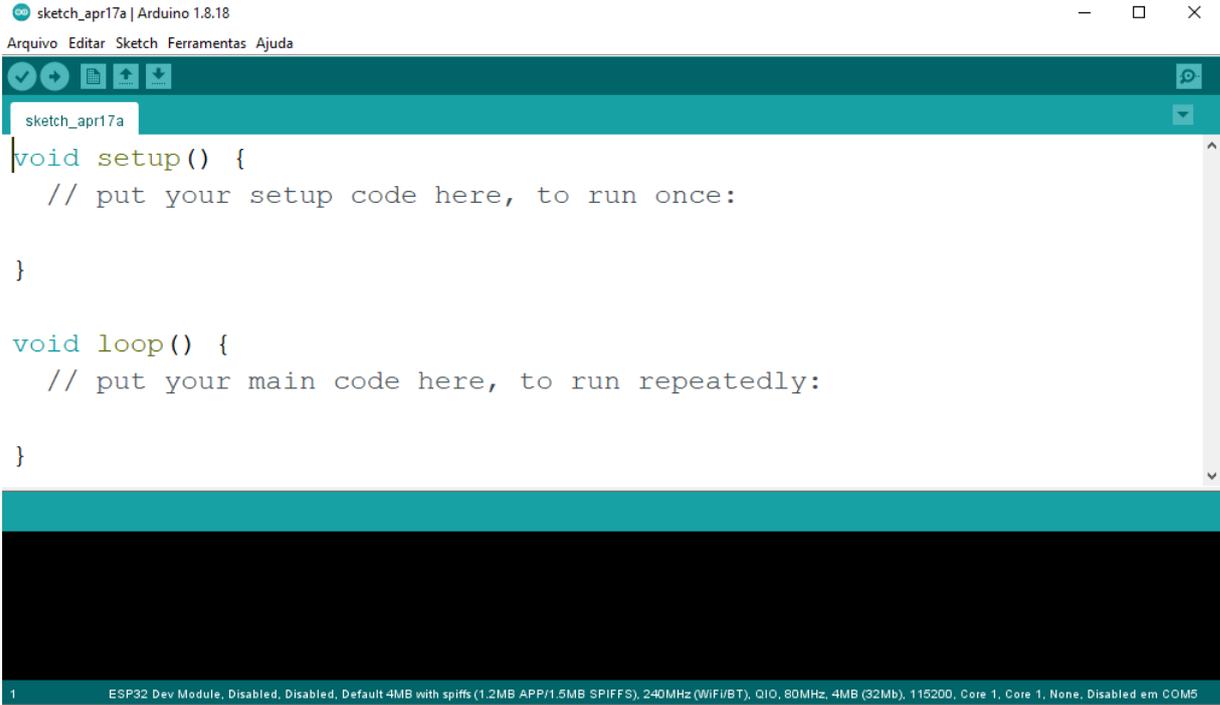
A plataforma conta com um *layout* simples e intuitivo, dividido em duas partes: Designer, onde o usuário pode configurar como será a interface de usuário do aplicativo, e Blocos, onde é feita a programação da lógica.

2.4.3 Plataforma Arduino IDE

A Arduino IDE (*Integrated Development Environment*) é uma plataforma de desenvolvimento de códigos, capaz de programar plataformas de prototipagem eletrônica de código aberto como o Arduino. A plataforma foi criada para facilitar a programação de microcontroladores usados em eletrônica, além de ser um *software* gratuito e de código aberto, com uma extensa comunidade e os mais diversos exemplos de códigos disponíveis para aprendizado e testes. A plataforma utiliza a linguagem de programação baseada em C/C++, possibilitando a programação de forma simples, além da IDE permitir a comunicação com as placas Arduino utilizando portas USB, a fim de realizar o envio do código para o microcontrolador. A plataforma também conta com uma coleção vasta de

bibliotecas com exemplos de códigos e outras bibliotecas que incluem mais possibilidades na programação (MAKERHERO, 2020).

Ademais, a plataforma Arduino IDE pode ser utilizada com outros microcontroladores, como é o caso do ESP32 utilizado no projeto, contudo, é necessário fazer algumas configurações adicionais na plataforma bem como a instalação de outras bibliotecas. Todo o processo foi detalhado no capítulo de desenvolvimento do projeto. A Figura 3 ilustra o ambiente de desenvolvimento fornecido pela plataforma.



The image shows a screenshot of the Arduino IDE interface. The window title is "sketch_apr17a | Arduino 1.8.18". The menu bar includes "Arquivo", "Editar", "Sketch", "Ferramentas", and "Ajuda". The toolbar contains icons for opening, saving, and running. The main editor area shows the following code:

```
sketch_apr17a
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

The status bar at the bottom displays hardware specifications: "1 ESP32 Dev Module, Disabled, Disabled, Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 115200, Core 1, Core 1, None, Disabled em COM5".

Figura 3 – Tela de criação Blocos. Fonte: Autoria Própria

2.5 Sensores para medição de corrente e Efeito Hall

Existem diferentes sensores de corrente que adotam diferentes estratégias para a medição, os mais comuns são os Sensores de Efeito Hall, Sensores de Transformadores de Corrente e Sensores de Bobina Rogowski ([CHAPECÓ LÍDER AUTOMAÇÃO, 2025](#)). O sensor utilizado no projeto, adota o efeito Hall.

O efeito Hall foi descoberto em 1879, por Edwin Herbert, durante seu doutorado em física na Universidade Johns Hopkins em Baltimore, Maryland.

Os princípios do efeito Hall, como explicado por [Instituto de Física - UFRGS \(2025\)](#) foram descobertos por meio de experimentos. A Figura 4 auxilia na descrição do Efeito Hall.

“ Aplicando-se um campo magnético na direção horizontal, conforme indicado na Figura 4, resulta numa força magnética na direção perpendicular ao movimento eletrônico, no sentido de cima para baixo. Esta força fará com que o movimento dos elétrons seja desviado para baixo. Com o tempo, cargas negativas acumulam-se na face inferior, e cargas positivas na face superior. O excesso de cargas positivas e negativas, funciona como um capacitor de placas paralelas, com um campo elétrico conhecido como campo Hall ”

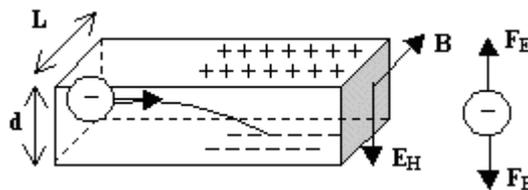


Figura 4 – Imagem auxiliar efeito Hall. Fonte: [Instituto de Física - UFRGS \(2025\)](#)

O efeito Hall permite determinar o sinal da carga dos portadores, medindo a diferença de potencial entre as superfícies superior e inferior. Alguns sensores de corrente usam o efeito Hall para medir a corrente presente nos circuitos.

2.6 Estratégia de medição de corrente: invasivo e não invasivo

Segundo [Medeiros et al. \(2018\)](#) o monitoramento de carga intrusivo consiste em dispositivos que, para realizar a medição, entram em contato físico com o sistema, assim é possível ter mais precisão nos resultados, apesar de ser uma estratégia normalmente com custo mais elevado. Já o monitoramento não intrusivo, usa dispositivos que medem sem contato direto, evitando interferências no sistema, resultando em uma estratégia mais barata mas com menos precisão. Para o projeto, a fim de garantir maior precisão, foi adotado a estratégia de medição intrusiva, onde o circuito medidor de corrente estará presente em uma tomada da residência.

2.7 Sensor de corrente ACS712

Sensor de Corrente ACS712 é um sensor capaz de medir correntes elétricas de equipamentos, fornecendo soluções precisas e econômicas para detecção de corrente CA ou CC de acordo com seu Datasheet:

“O Allegro® ACS712 fornece soluções econômicas e precisas para detecção de corrente CA ou CC em sistemas industriais, comerciais e de comunicações. As aplicações típicas incluem controle de motor, detecção e gerenciamento de carga, fontes de alimentação comutadas e proteção contra falhas de sobrecorrente” ([ALLEGRO MICROSISTENS INC., 2006](#)).

O sensor é capaz de medir correntes de -20A a 20A e utiliza o efeito Hall para medir a corrente. Sua resposta varia de 0V a 5V, de acordo com a corrente monitorada. Quando não há corrente no circuito, a tensão de saída é 2,5V, o que representa 0A. A imagem do sensor é apresentado na Figura 5, e seu esquema elétrico na Figura 6.

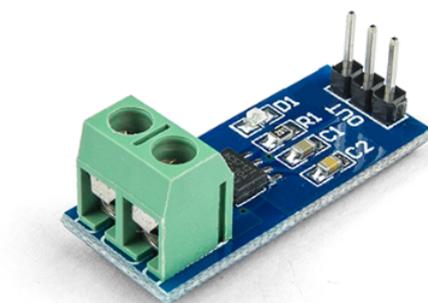


Figura 5 – Sensor de Corrente ACS712 Fonte: [Robocore \(2024b\)](#).

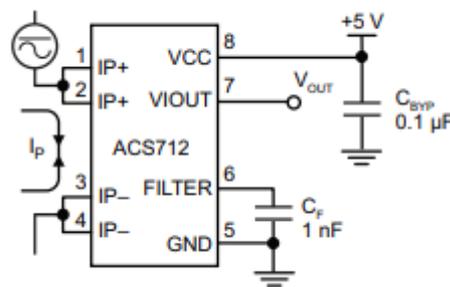


Figura 6 – Esquema elétrico ACS712 Fonte: [Allegro MicroSistens \(2006\)](#).

2.8 Relé

O módulo relé pode ser usado em diversos projetos, a fim de acionar cargas de até 250V AC @ 7A ou 125V AC @ 10A, permitindo o controle de equipamentos ligados a rede elétrica. Além disso possível até mesmo fazer uma ponte H baseada em relés com alguns destes módulos ([ROBOCORE, 2024a](#)). Neste trabalho será usado o Módulo Relé 5V - Relé Metaltex, a fim de controlar a passagem de corrente do componente que terá sua corrente medida. O relé pode ser ligado como NA (normalmente aberto) ou NF (normalmente fechado). Para este projeto, foi escolhida a ligação NF (normalmente fechado), o relé irá permitir a passagem de corrente, até que um comando externo seja enviado para bloqueá-la. O modelo do relé utilizado é representado na Figura 7.



Figura 7 – Relé utilizado no projeto. Fonte: [Robocore \(2024a\)](#).

2.9 Fonte de alimentação

Para o projeto, foi utilizada a fonte de alimentação para protoboard MB102, que consegue fornecer tensões de 3,3V a 5,0V, garantindo a alimentação do circuito. Juntamente a ela, foi utilizada uma fonte AC/DC De 9V modelo ZX-0910, essa fonte pode ser usada em conjunto com a fonte MB102 ou até mesmo diretamente em alguns microcontroladores como Arduino Uno. O modelo da fonte utilizada é representado na Figura 8.

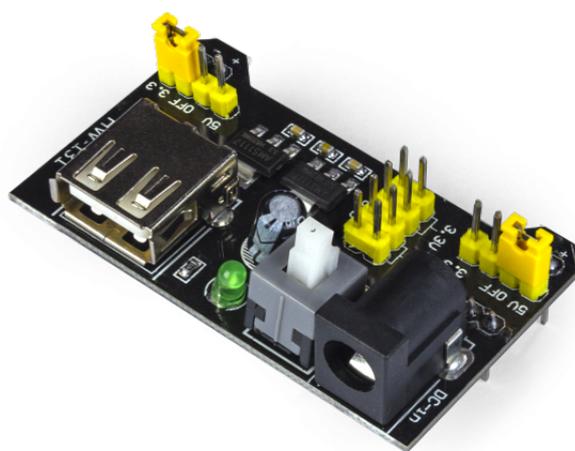


Figura 8 – Fonte regulável para microcontrolador. Fonte: ([ROBOCORE, 2024](#)).

3 Metodologia

3.1 Cálculo de potência e corrente

3.1.1 Divisor de tensão

No desenvolvimento do projeto, foi necessário a utilização de um divisor de tensão. O ESP32 suporta a tensão máxima de 3,3 V como entrada analógica, já o sensor de corrente envia sinais de 0V a 5V, sendo 2,5V a sua resposta quando não há leitura de corrente. Ao ler o valor máximo de corrente, o sensor enviará uma resposta equivalente a 0 ou 5 V, a depender do sentido da corrente (0V = -20A ; 5V = 20A). Portanto, para proteger o microcontrolador, foi utilizado um divisor de tensão.

De acordo com [Arduino e Cia \(2025\)](#), o circuito divisor de tensão é utilizado para reduzir a tensão de entrada a uma tensão menor proporcional, é um circuito muito utilizado na eletrônica, construído com pelo menos dois resistores ligados em série, como demonstrado na Figura 9.

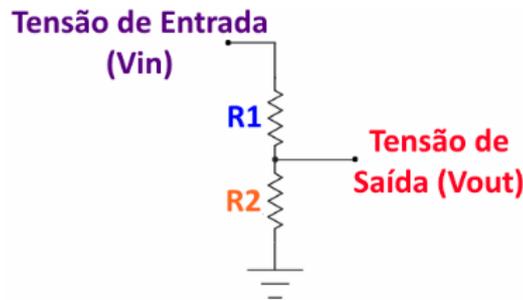


Figura 9 – Circuito divisor de tensão. Fonte: [Arduino e Cia \(2025\)](#)

A tensão de entrada será a enviada pelo sensor de medição de corrente, com valores de 0V a 5V, devido ao fato dos resistores estarem em paralelo com a alimentação, a tensão de entrada será dividida entre eles, seguindo o equacionamento:

$$\frac{V_{out}}{V_{in}} = \frac{R_2}{R_1 + R_2}$$

$$R_1 = R_2 = 300 \Omega$$

$$\frac{V_{out}}{V_{in}} = \frac{300 \Omega}{300 \Omega + 300 \Omega}$$

$$\frac{V_{out}}{V_{in}} = \frac{300}{600}$$

$$\frac{V_{out}}{V_{in}} = \frac{1}{2}$$

$$V_{out} = \frac{V_{in}}{2} \quad (3.1)$$

onde V_{in} é a tensão de entrada (fornecida pelo sensor de corrente) [0 - 5V] e V_{out} é tensão de saída (enviada para o ESP32 para o cálculo de corrente).

Ao utilizar duas resistências iguais, a tensão enviada pelo sensor de corrente será dividida pela metade, protegendo o circuito. Contudo, o alcance de medição também é dividido pela metade, então o projeto conseguirá medir apenas correntes de 0A a 10A. O valor de tensão enviado pelo sensor de corrente que representa uma leitura de corrente nula será dividido pela metade, portando, será igual a 1,25V conforme a relação demonstrada na equação 3.1.1.

3.1.2 Método de cálculo

O esp 32 tem uma resolução de de 12 bits em seu conversor analógico - digital, o que significa que ele pode converter um sinal analógico em um valor digital que varia de 0 a 4095 ($2^{12} = 4096$ níveis). para interpretação dos dados e cálculo da corrente, utiliza-se a seguinte relação:

$$I = \frac{V_{out} - V_{zero}}{S}$$

$$I = \frac{V_{out} - 1,25}{0,1} \quad (3.2)$$

onde, V_{out} representa a entrada analógica (V) fornecida ao microcontrolador, V_{zero} é o Valor de tensão referencial quando $I = 0$ (A), S é a sensibilidade do sensor (100 mV/A) e I é a corrente elétrica (A). Portanto, calculamos a corrente de acordo com o demonstrado na equação 3.2.

Para garantir uma maior precisão na medição da corrente, foram realizadas múltiplas leituras ao longo do tempo. A partir dessas amostras, foi calculado o valor RMS (Root Mean Square - Raiz Média Quadrática), que é uma métrica essencial para a análise de sinais alternados, como a corrente monitorada no projeto.

O cálculo de potência é descrito pela fórmula 3.3.

$$P = V \cdot I \quad (3.3)$$

Onde P é a potência (em watts), V é a tensão (em volts), I é a corrente (em amperes).

3.2 Configuração do relé

O relé utilizado no projeto permite a ligação NA ou NF. Foi escolhida a ligação NF para seguir a seguinte lógica: o status do aparelho é definido no aplicativo Kodular pelo usuário e então é enviado ao banco de dados. Caso o status seja igual a “Ligado” o relé não irá receber o sinal de ativação, permitindo a passagem de corrente. Caso o status seja igual a “Desligado” o relé será ativado através de um sinal digital enviado pelo ESP32, abrindo o circuito, bloqueando a passagem de corrente, até que o status mude novamente para “Ligado”.

3.3 Banco de dados Firebase

3.3.1 Criação do banco de dados

O banco de dados foi criado seguindo instruções simples e objetivas do próprio Firebase. Após a criação de uma conta, foi necessário a realização de algumas configurações adicionais. A primeira foi a alteração de regras do Realtime Database, pois ao alterar as regras existentes é possível alterar, enviar e receber dados do Firebase. O passo a passo seguido na alteração das regras é ilustrado nas Figuras 10 e 11.

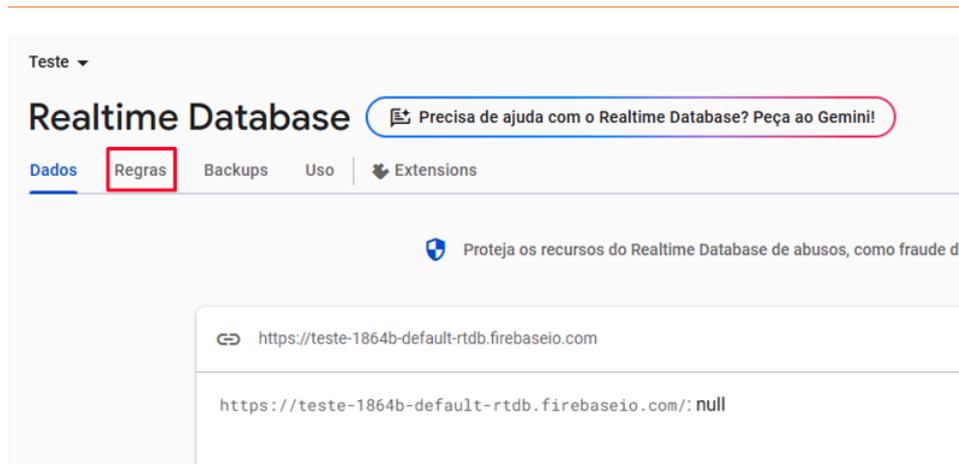


Figura 10 – Alteração de regras Firebase, Imagem 1. Fonte: Autoria Própria.

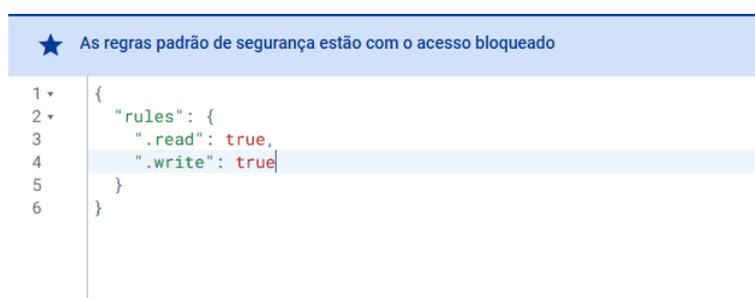


Figura 11 – Alteração de regras Firebase, Imagem 1. Fonte: Autoria Própria.

3.3.2 Estrutura do banco de dados

A estrutura adotada para o armazenamento de dados no Firebase é ilustrado nas Figuras 12 e 13.

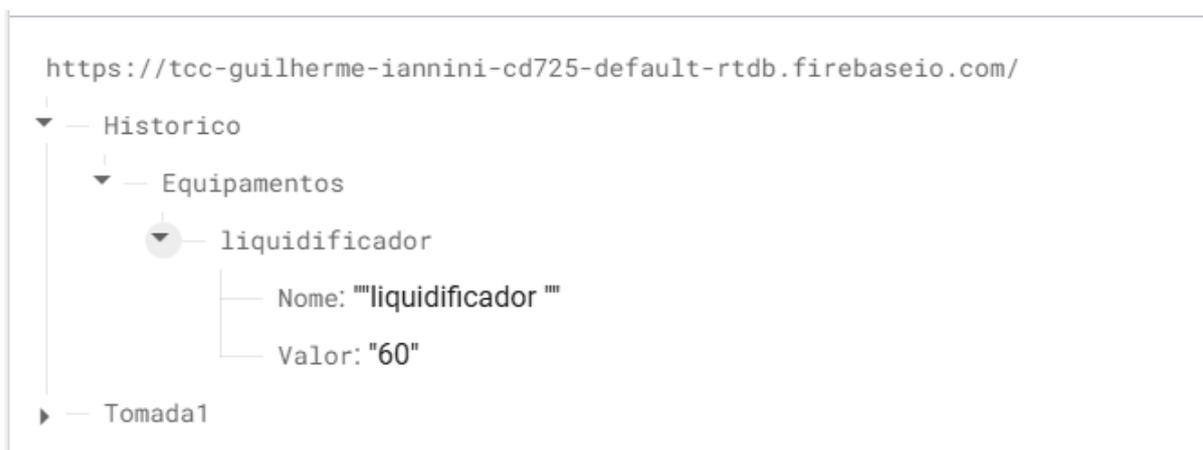


Figura 12 – Estrutura do banco de dados, Imagem 1. Fonte: Autoria Própria.

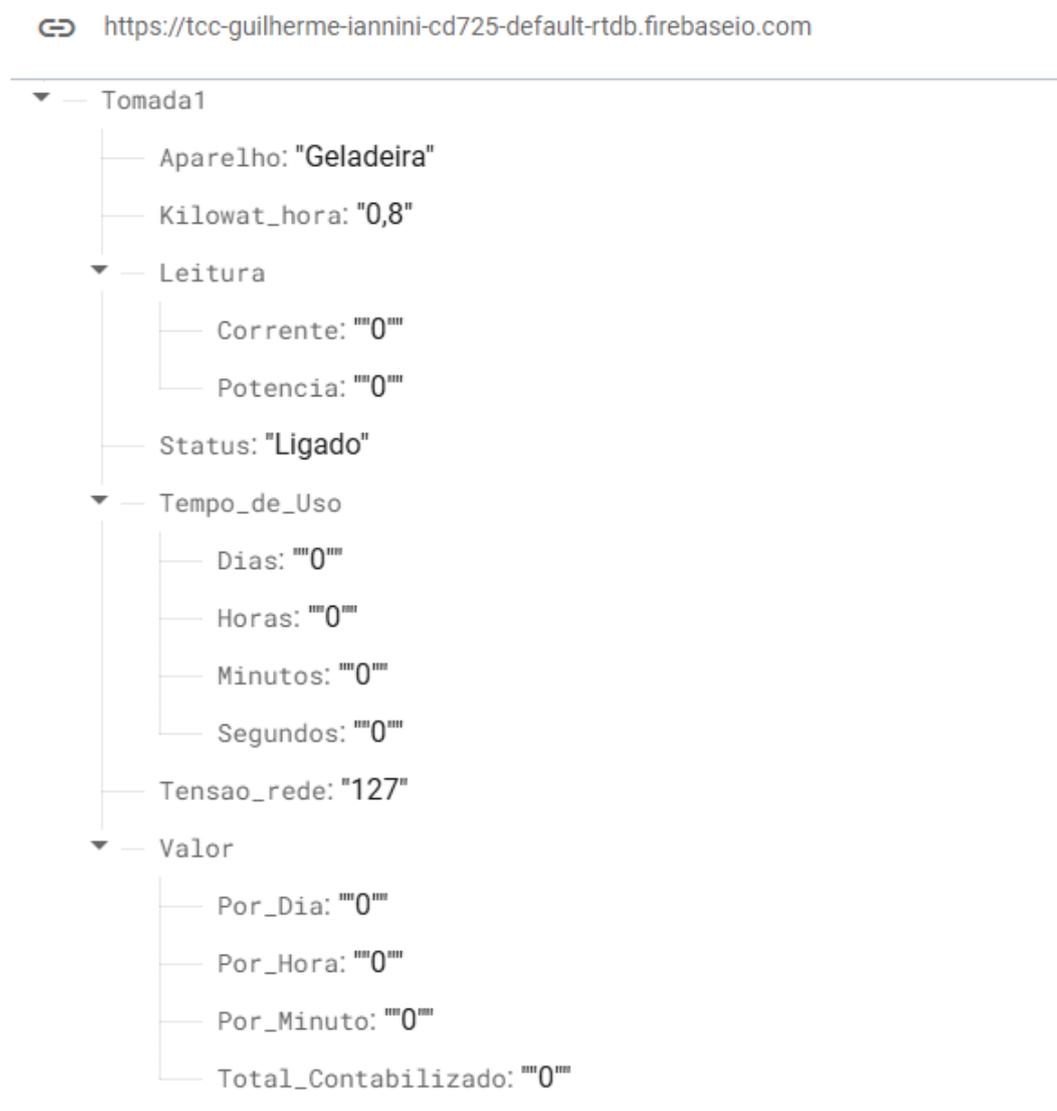


Figura 13 – Estrutura do banco de dados, Imagem 2. Fonte: Autoria Própria.

Na aba “Tomada1” constam as informações do equipamento que está sendo monitorado. A aba histórico contém as informações dos equipamentos salvos pelo usuário.

3.4 Aplicativo do sistema de medição de energia (S.M.E)

Foi criado um aplicativo na plataforma Kodular a fim de facilitar a visualização e alteração dos dados salvos no banco de dados Firebase.

3.4.1 Funções do aplicativo

Utilizando o aplicativo, o usuário tem acesso a várias funções:

- Visualização do cálculo de corrente, potência e custos monetários associados ao equipamento monitorado;
- Opção de salvar os dados do equipamento em um histórico;
- Barra de pesquisa para consulta de informações de equipamentos salvos no histórico;
- Limpeza do histórico ou dos valores do equipamento atual;
- Alteração do nome referente ao equipamento monitorado;
- Controle do relé clicando em botões rotulados “Ligar” e “Desligar”.

O aplicativo também possui uma imagem de logo própria, demonstrada na Figura 14.



Figura 14 – Logo do aplicativo. Fonte: Criado por Daniele Eva Cota Coluna

3.4.2 Telas do aplicativo

Foram desenvolvidas seis telas para a utilização do aplicativo, tela de abertura que mostra a logo do aplicativo, tela de menu para navegação, tela de histórico para consultas e tela de monitoramento para se obter informações. A Figura 15 demonstra como é a tela inicial do aplicativo, a Figura 16 ilustra como é a tela do menu, que redireciona o usuário

para as demais telas. A Figura 17 mostra a tela de configuração de equipamentos, onde o usuário consegue editar as informações pertinentes do mesmo. A Figura 18 mostra a tela de histórico, onde o usuário pode consultar informações salvas de equipamentos ou eliminar as informações que constam no histórico. A Figura 19 refere-se a tela de monitoramento, que exibe informações importantes referentes ao equipamento monitorado, também permite salvar as informações do equipamento em questão no histórico. Por fim, a Figura 20 mostra a tela de informações, que é uma tela que ensina como utilizar o aplicativo.



S.M.E.



Figura 15 – Tela de abertura. Fonte: Autoria Própria.

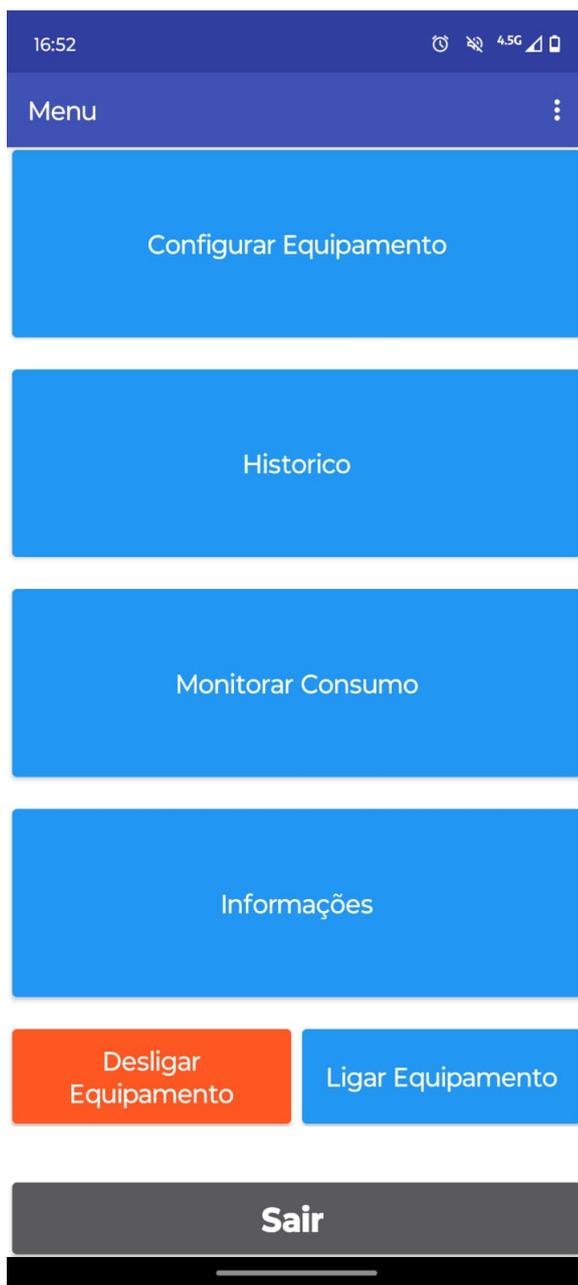


Figura 16 – Tela Menu. Fonte: Autoria Própria.

16:52 4.5G

Configurar Equipamento

Digite o Nome do Aparelho

Adicionar

Digite o Valor do KiloWatt/Hora

Adicionar

Digite a Tensão da Rede

Adicionar

Exemplo: 110 ou 127 ou 220

limpeza de leitura

Zerar Valores

Digite "zerar" e clique no botão para zerar os valores armazenados

Retornar ao Menu

Figura 17 – Tela Configurar Equipamentos. Fonte: Autoria Própria.

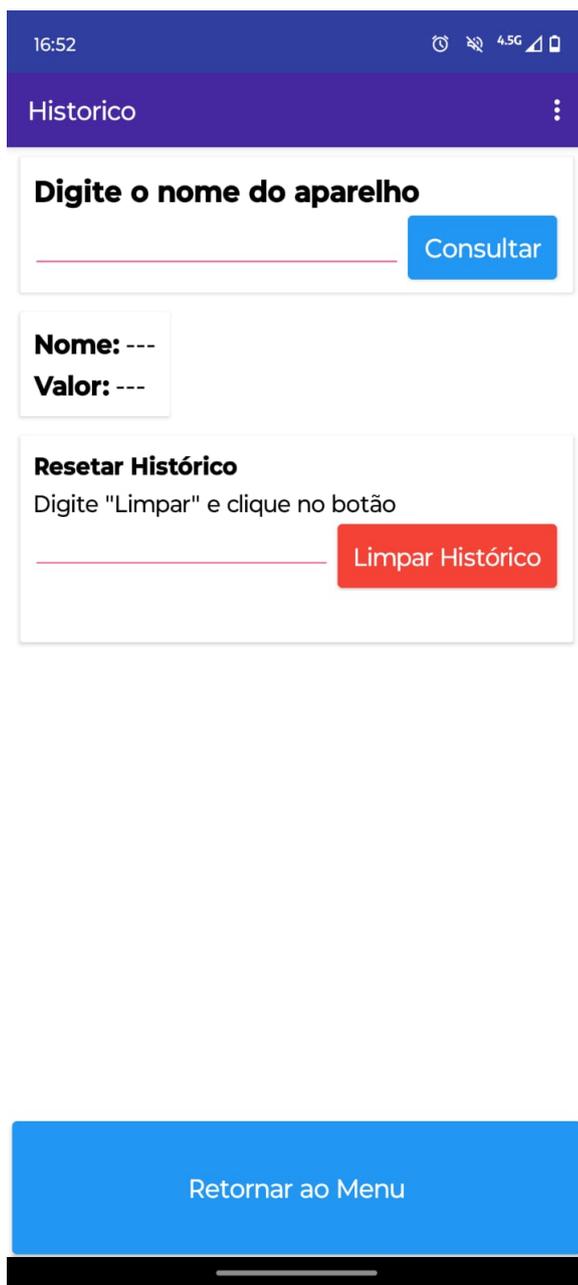


Figura 18 – Tela Histórico. Fonte: Autoria Própria.

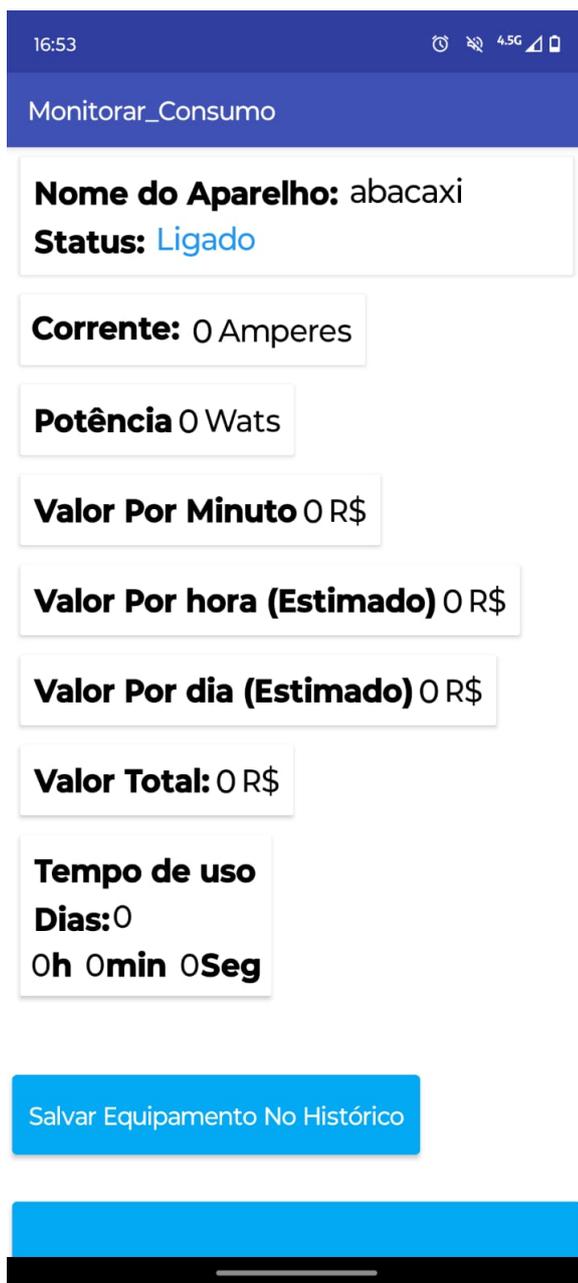


Figura 19 – Tela Monitorar Consumo. Fonte: Autoria Própria.

**Siga os Passos para Utilizar o APP:**

1- O projeto foi idealizado para medir apenas um equipamento por vês, para tal, primeiro clique em Configurar Equipamento.

2- Na tela "configurar Equipamento" Informe o nome do equipamento para a identificação. Digite o valor do Kilowatt e a tensão da rede (110, 127, 220 , etc.).

Caso deseje Resetar os valores ja medidos, utilize a barra "limpeza de leitura".

3- Clique em Monitorar Equipamento para obter as informações e salvar as mesmas no histórico se desejar.

4- O histórico mostra as informações de equipamentos salvos, tenha certeza de digitar o nome igual ao que consta no histórico (letras maiúsculas, etc).

5- Na tela de Histórico, existe a opção de deletar o histórico caso seja preciso.



Figura 20 – Tela de Informações. Fonte: Autoria Própria.

3.5 Montagem do protótipo

A montagem adotada do circuito está representada na Figura 21, e sua montagem física nas Figuras 22 e 23.

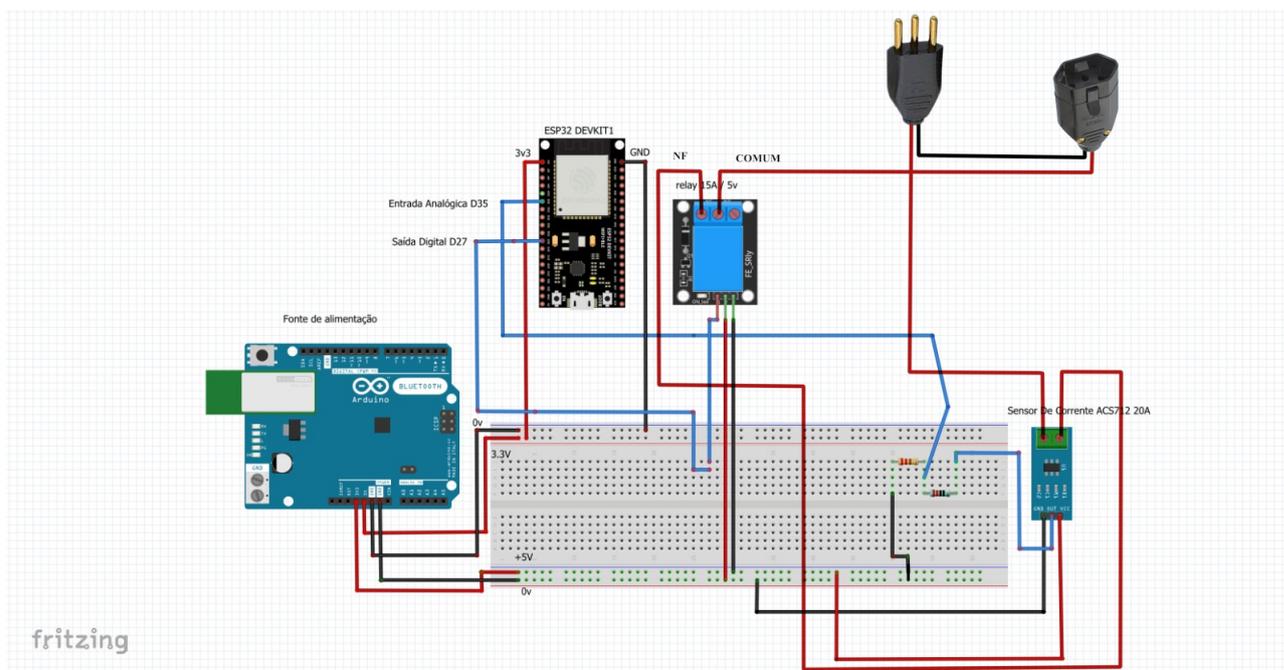


Figura 21 – Circuito demonstrativo. Fonte: Autoria Própria.



Figura 22 – Montagem física do protótipo. Fonte: Autoria Própria.

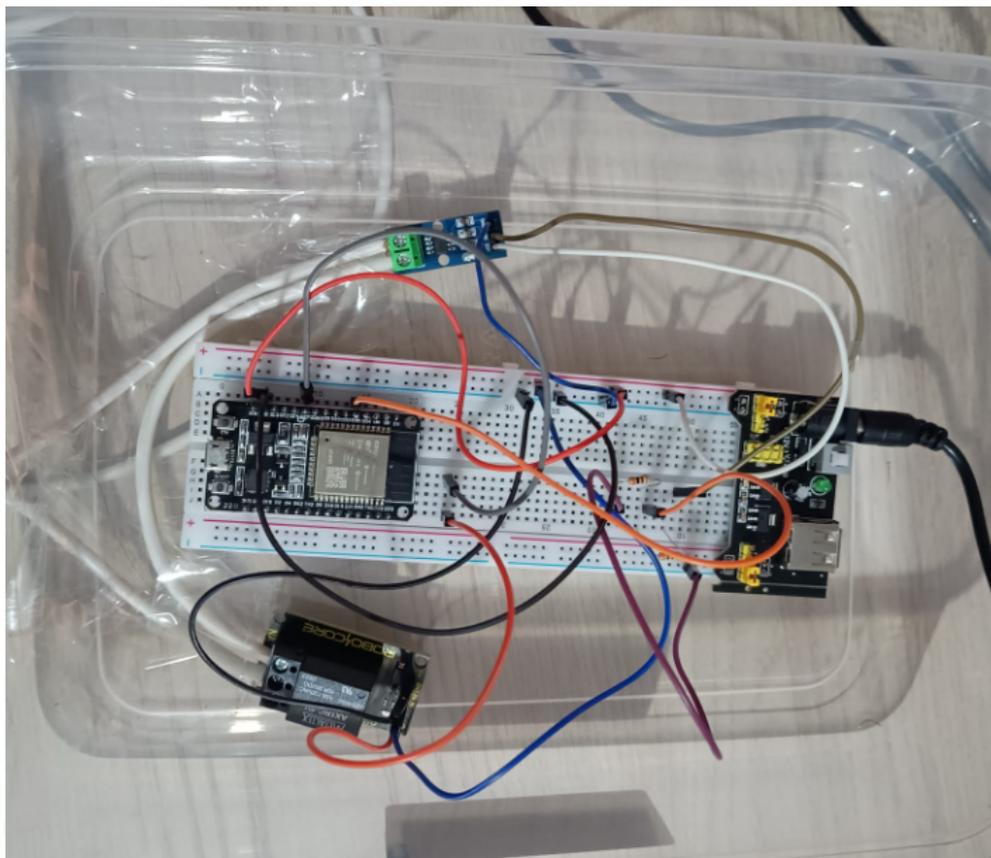


Figura 23 – Montagem física do protótipo. Fonte: Autoria Própria.

3.6 Testes e calibração do sensor

3.6.1 Análise de linearidade do conversor analógico - digital do ESP32

Foi construído um circuito de testes para monitorar a linearidade da resposta do conversor ADC do ESP32 e analisar como o mesmo interpreta diferentes valores analógicos, certificando a linearidade nas conversões de valores analógicos para digitais. Um potenciômetro foi utilizado para fornecer ao ESP32 diferentes valores de tensão para realização dos testes. Para cada valor de tensão entre 0V a 3,3V fornecido ao ESP32, seu conversor ADC gera uma resposta que varia de 0 a 4069, conforme demonstrado nas Tabelas 1 e 2 e na Figura 24.

Tabela 1 – Resposta do Conversor A/D (Autoria Própria)

Medições	Tensão Aplicada (V)	ESP32 (média 1000 amostras)	Diferença
1	0,20	126	—
2	0,30	253	127
3	0,40	381	128
4	0,50	502	121
5	0,60	626	124
6	0,70	756	130
7	0,80	876	120
8	0,90	1004	128
9	2,00	2407	—
10	2,10	2528	121
11	2,20	2658	130
12	2,30	2785	127
13	2,40	2919	134
14	2,50	3050	131
15	2,60	3182	132
16	2,70	3331	149
17	3,00	3844	—
18	3,05	3958	114
19	3,10	4054	96
20	3,15	4091	37
21	3,20	4093	2
22	3,25	4093	0
23	3,30	4096	3

Tabela 2 – Variação do Conversor A/D para Intervalos de Medida (Autoria Própria)

Medições	Média de Variação
2 a 9	125,43
12 a 18	132,00
20 a 25	42,00

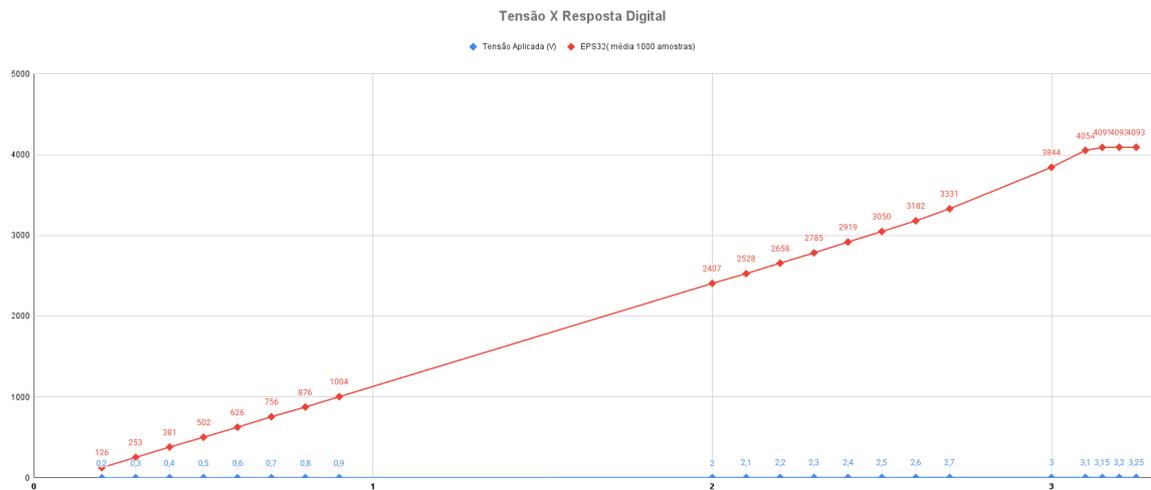


Figura 24 – Gráfico respostas ADC do ESP32. Fonte: Autoria Própria.

Analisando os dados foi concluído que o ESP32 tem um comportamento muito próximo ao linear para valores de tensão menores que 3,0V. Contudo, para valores entre 3,0V e 3,3V seu comportamento se afasta do linear. As informações coletadas foram pertinentes para avaliar o desempenho do sistema de medição de corrente desenvolvido.

3.6.2 Montagem para realização de testes

Para garantir que a medição de corrente estava correta, foram realizados testes no laboratório de Eletrônica Analógica e Digital (LEAD), localizado na UFOP, campus Ouro Preto. Os testes mostraram a diferença de valores medidos em comparação aos fornecidos, para corrente alternada e corrente contínua. A montagem de testes está demonstrada na Figura 25.

Houve uma alteração no protótipo inicial, para poder ligar um Dimmer em série com o sensor de corrente, para regular a quantidade de corrente fornecida a um filtro de linha. Foi adicionado também um multímetro em série para analisar a quantidade de corrente presente no circuito. No filtro de linha, foram ligadas três lâmpadas dimerizáveis, dessa forma, foi possível variar a corrente fornecida ao circuito, para comparar os valores lidos no amperímetro com os valores calculados no programa gravado no ESP32 pela plataforma Arduino IDE. Devido a limitações dos equipamento utilizados na medição e dos fios, foram medidas correntes baixas com valor máximo de 2,0A.

A Figura 26 auxilia na identificação de componentes, no quadrado de número 1, está localizado o multímetro. No quadrado de número 2, representa o sistema embarcado que realiza a medição e controle, o quadrado de número 3 mostra o dimmer utilizado para variar a corrente do circuito. O quadrado de número 4 mostra o filtro de linha onde as lâmpadas dimerizáveis foram ligadas.



Figura 25 – Montagem para testes corrente alternada. Fonte: Autoria Própria.

3.6.3 Testes corrente alternada

Primeiro foram realizados os testes para observar o comportamento do sensor ao medir correntes alternadas, os dados obtidos são demonstrados na Tabela 2.

Tabela 3 – Teste de Corrente Alternada

Medições	Corrente Fornecida (A)	Corrente Calculada (A)
1	0,42	0,58
2	0,55	0,68
3	0,69	0,78
4	0,77	0,84
5	0,85	0,89
6	0,96	0,97
7	1,05	1,01
8	1,12	1,05
9	1,17	1,07
10	1,27	1,11
11	1,37	1,16
12	1,47	1,20
13	1,56	1,23
14	1,63	1,25
15	1,76	1,28
16	1,87	1,30

Os dados foram registrados no *software* LibreOffice, após anotados os valores, foi observado que o comportamento do sensor se distancia do comportamento linear esperado em alguns momentos. Foi gerado um gráfico para análise de resultados, porém, devido ao comportamento se afastar do linear em alguns momentos, foi inserido uma linha de tendência para série de dados polinomial de grau 2, para se obter um modelo que aproxime os resultados medidos dos resultados reais. O gráfico criado é ilustrado na Figura 27, o valor do coeficiente de determinação R^2 ficou em torno de 0,9950, se encontra na equação 3.4, sendo representado como $f(x)$.

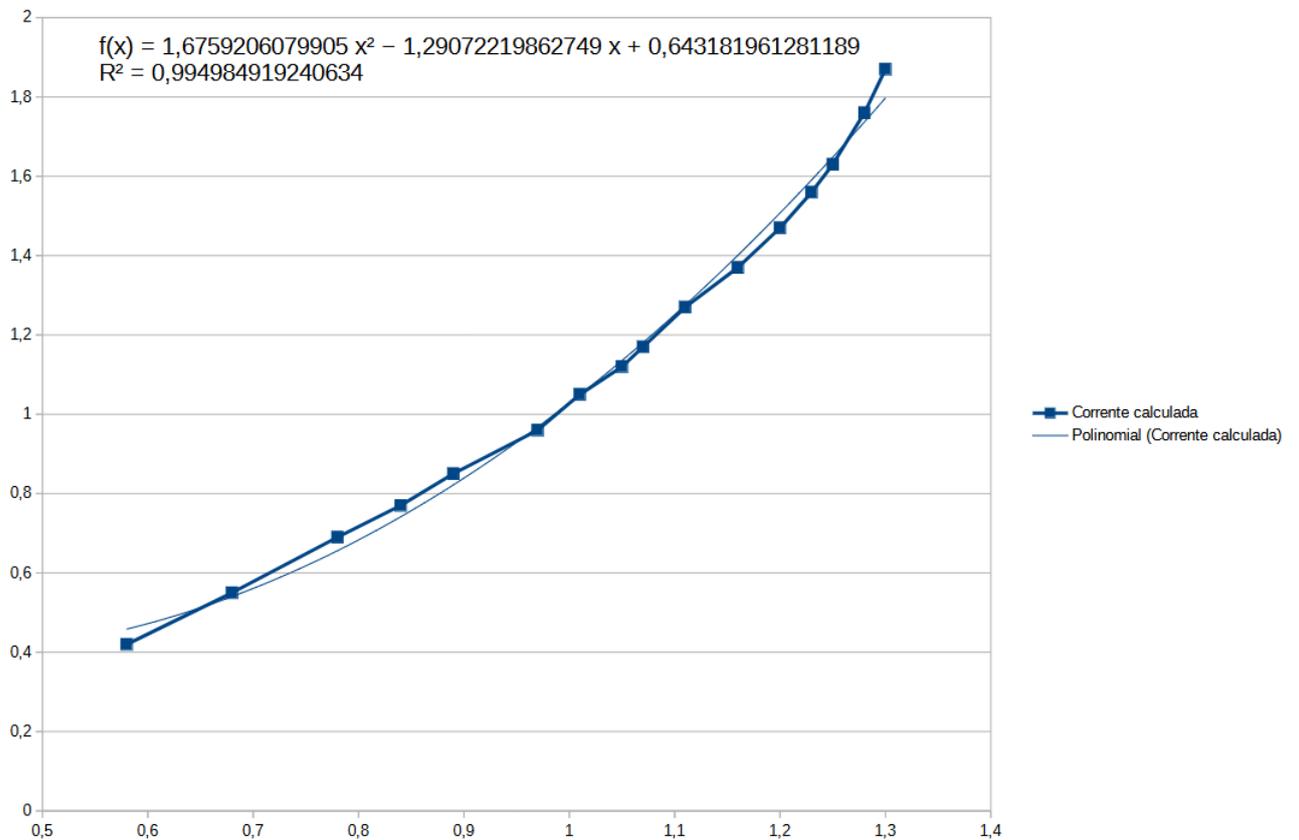


Figura 27 – Gráfico obtido no teste de corrente alternada. Fonte: Autoria Própria.

$$f(x) = 1,6759 x^2 - 1,2907 x + 0,6431 \quad (3.4)$$

3.6.4 Testes correntes contínua

Os testes com corrente contínua foram feitos de forma análoga ao teste de corrente alternada. A diferença principal é que a corrente foi fornecida ao sistema por uma Fonte de Alimentação DC Regulada 5A/250V Modelo FA 3005. Os resultados obtidos na medição de corrente contínua se aproximaram mais de um comportamento linear conforme ilustrado na Tabela 3. A Figura 28 demonstra o gráfico obtido a partir dos dados.

Tabela 4 – Teste Corrente Contínua

Medições	Corrente Fornecida (A)	Corrente Calculada (A)
1	0,20	0,13
2	0,26	0,17
3	0,29	0,18
4	0,32	0,22
5	0,40	0,29
6	0,52	0,37
7	0,84	0,61
8	1,02	0,74
9	1,23	0,89
10	1,41	1,02
11	1,56	1,13
12	1,70	1,24
13	1,86	1,35
14	2,05	1,51

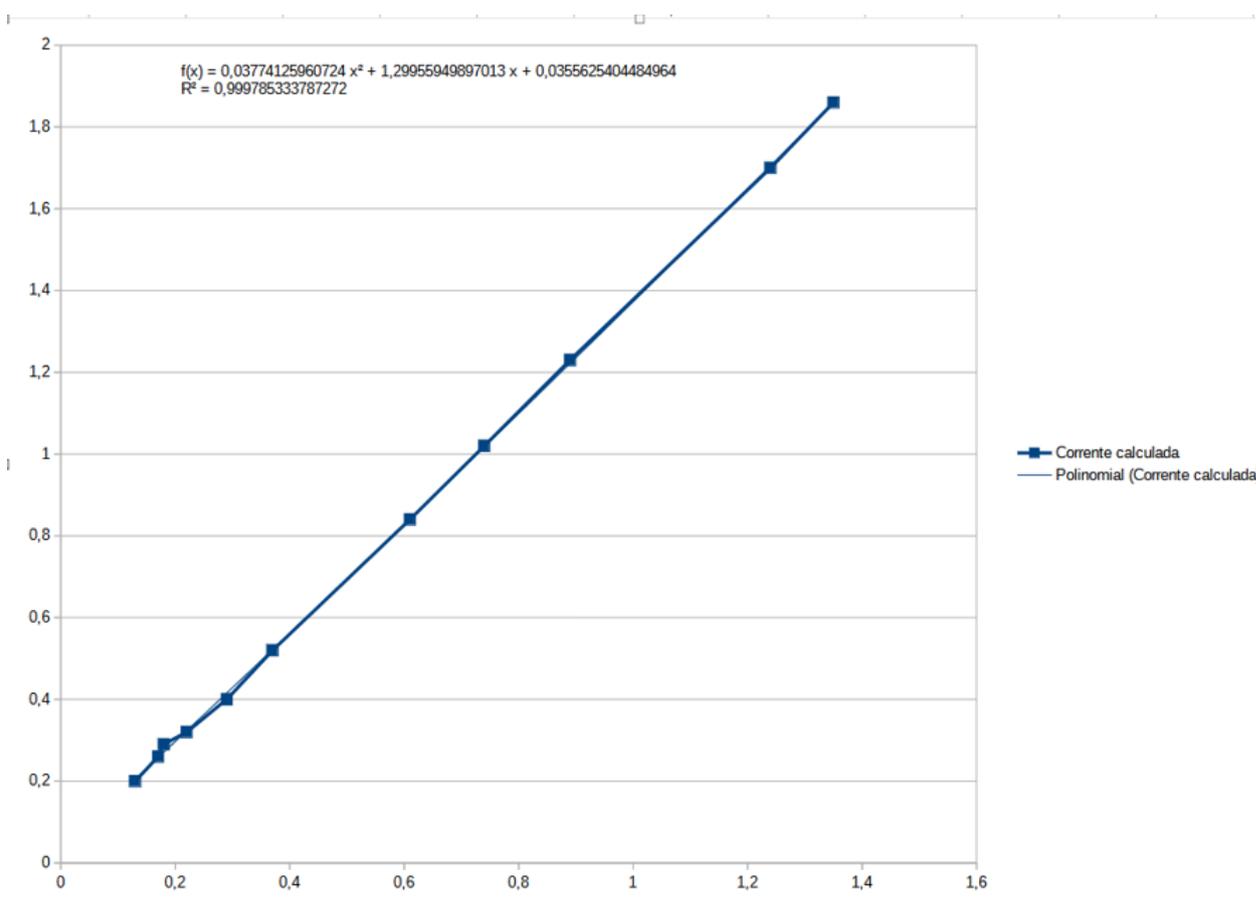


Figura 28 – Gráfico obtido nos testes corrente contínua. Fonte: Autoria Própria.

3.6.5 Calibração do sensor

Para a calibração final do sensor, foi utilizado o modelo obtido na linha de tendência polinomial gerado nos testes de corrente alternada. Conforme demonstrado no trecho de código a seguir.

```
1 float calcularCorrenteRMS() {.
2     float somaQuadrados = 0;
3     float filtro;
4     for (int i = 0; i < 500; i++) {
5         int leitura = analogRead(ANALOG_PIN_0) - 1414;
6         somaQuadrados += leitura * leitura;
7         delay(1);
8     }
9     float aux = sqrt(somaQuadrados / 500) * (2.5 / 4096);
10    float corrente = aux / sensibilidadeSensor;
11    if (corrente <= 0.5) {
12        return corrente;
13    } else {
14        filtro =(1.6756 * corrente * corrente) - (1.2907 * corrente) +
15            0.6432;
16        return filtro;
17    }
```

Listing 3.1 – Função para cálculo da corrente RMS

4 Resultados

Foi realizado um novo teste de desempenho da medição de corrente do sensor ACS712 após aplicado o modelo obtido na regressão polinomial.

O erro foi calculado utilizando a fórmula de erro quadrático médio, ficando dentro de 5%, para averiguar o desempenho antes e depois da calibração. Foi notório a melhoria no desempenho do sistema devido a diminuição do erro, fornecendo medições muito mais próximas do valor real.

Contudo o modelo não tem resultados satisfatórios para correntes muito baixas, onde a precisão do sensor também é bem instável (entre 0A e 0,3A). A solução implementada foi a criação de uma lógica condicional, onde o filtro não será considerado caso a corrente seja muito baixa.

O sistema teve resultados satisfatórios, com uma taxa de erro tolerável. Provando assim sua capacidade em auxiliar os usuários a monitorar o consumo de energia em suas residências bem como os custos monetários associados.

As tabelas 5 e 6 ilustram o comparativo de antes e depois da aplicação do modelo.

Tabela 5 – Teste Corrente Alternada

Medições	Corrente Fornecida (A)	Corrente Calculada (A)
1	0,42	0,58
2	0,55	0,68
3	0,69	0,78
4	0,77	0,84
5	0,85	0,89
6	0,96	0,97
7	1,05	1,01
8	1,12	1,05
9	1,17	1,07
10	1,27	1,11
11	1,37	1,16
12	1,47	1,20
13	1,56	1,23
14	1,63	1,25
15	1,76	1,28
16	1,87	1,30
Erro médio = 25,31%		

Tabela 6 – Teste Corrente Alternada com filtro

Medições	Corrente Fornecida (A)	Corrente Calculada (A)
1	0,00	0,55
2	0,31	0,41
3	0,63	0,62
4	0,99	0,95
5	1,11	1,11
6	1,23	1,24
7	1,50	1,52
8	1,63	1,62
9	1,75	1,75
10	1,80	1,91
11	1,98	2,06
Erro médio = 5,02%		

5 Considerações finais

5.1 Conclusão

O trabalho foi desenvolvido no intuito de desenvolver um sistema de monitoramento de consumo de energia elétrica residencial de baixo custo, para tal, foi utilizado o microcontrolador ESP32, um dispositivo de baixo custo com conectividade Wifi programado na plataforma Arduino IDE.

Para medição de corrente foi utilizado o sensor ACS712 que utiliza o efeito hall para medir correntes elétricas CA e CC. Também foi utilizado um módulo relé para permitir controle da ativação dos dispositivos de forma remota.

Para armazenar as informações, foi criado um banco de dados na plataforma Firebase, uma ferramenta simples e de fácil utilização que fornece diversas funções e soluções de forma gratuita. Os dados armazenados são enviados para um aplicativo criado na plataforma Kodular, a fim de facilitar o acesso às informações obtidas pelo sistema de medição bem como permitir o usuário controlar a ativação ou desativação de aparelhos de forma remota.

Para garantir a precisão dos cálculos, um circuito de testes foi desenvolvido para a realização das análises. Os resultados demonstraram a necessidade de uma calibração do sensor, realizada através da regressão polinomial. Após a aplicação da correção, os novos resultados demonstraram margens de erro dentro de 5%, contudo, para correntes pequenas, o sistema apresentou resultados imprecisos, devido as limitações do próprio sensor.

Por fim, para garantir maior aplicabilidade do sistema, o aplicativo desenvolvido possui a função de alterar o valor da tensão utilizada nos cálculos de potência, bem como a alteração do valor do preço do Kilowatt/Hora, uma vez que esses valores podem variar a depender da região em que serão utilizados e dependem da tensão da rede elétrica utilizada em cada casa. O sistema demonstrou resultados satisfatórios, com erro dentro de 5%. Portanto é uma solução de baixo custo e versátil para monitoramento e controle de acionamento de equipamentos residenciais.

5.2 Melhorias futuras

O projeto possui alguns pontos que podem ser considerados para melhorias futuras:

- O sistema foi projetado para ser utilizado em um único ponto de acesso, ou seja, uma tomada.

O acionamento remoto e cálculo de consumo de energia em mais de um ponto de acesso na residência só será possível com algumas alterações no protótipo.

- A calibração do sensor foi feita considerando correntes com valores de no máximo 2A.

Desenvolver uma calibração mais refinada considerando valores maiores de corrente proporcionaria um sistema com mais eficiência e confiabilidade.

- A utilização do divisor de tensão limitou a faixa de medição do sensor ACS712 a um valor de 10A.

A criação de um divisor de tensão que utilize valores máximos de 3,3v permitiria uma faixa de medição maior para o sensor de corrente. Contudo, vale ressaltar que o relé utilizado no projeto suporta correntes máximas de 15A.

Referências

- ACCARDI, Adonis; DODONOV, Eugeni. Automação residencial: elementos básicos, arquiteturas, setores, aplicações e protocolos. *Revista TIS*, v. 1, n. 2, 2012. Citado 1 vez na página 15.
- ALLEGRO MICROSISTENS, inc. *Fully Integrated, Hall Effect-Based Linear Current Sensor with 2.1 kVRMS Voltage Isolation and a Low-Resistance Current Conducto*. US: Allegro MicroSistens, inc, 2006. datasheet. Disponível em: <https://www.alldatasheet.com/datasheet-pdf/pdf/168326/ALLEGRO/ACS712.html>. Citado 0 vez na página 25.
- ALLEGRO MICROSISTENS INC. *Fully Integrated, Hall Effect-Based Linear Current Sensor with 2.1 kVRMS Voltage Isolation and a Low-Resistance Current Conductor*. US, 2006. Citado 1 vez na página 24.
- ARDUINO E CIA. *Calculador Divisor de Tensão*. 2025. Acessado em: 20 mar. 2025. Disponível em: <https://www.arduinoecia.com.br/calculador-divisor-de-tensao-function/>. Citado 1 vez na página 27.
- ARDUINO.CC. *What is Arduino?* US: Arduino.cc, 2024. site de informações do arduino. Disponível em: <https://www.arduino.cc/en/Guide/Introduction>. Citado 1 vez na página 19.
- CARRION, Patricia; QUARESMA, Manuela. Internet das Coisas (IoT): Definições e aplicabilidade aos usuários finais. *Human Factors in Design*, v. 8, n. 15, p. 049–066, 2019. Acessado em 15 de março de 2025. DOI: 10.5965/2316796308152019049. Disponível em: <https://www.revistas.udesc.br/index.php/hfd/article/view/2316796308152019049>. Acesso em: 15 mar. 2025. Citado 1 vez na página 14.
- CHAPECÓ LÍDER AUTOMAÇÃO. *O que é sensor de corrente?* 2025. Acessado em: 18 mar. 2025. Disponível em: <https://chapecali.com.br/glossario/o-que-e-sensor-de-corrente/>. Citado 1 vez na página 23.
- CHASE, Otavio; ALMEIDA, F. *Sistemas embarcados*. Mídia Eletrônica. 2007. Disponível em: <http://www.sbajovem.org/chase>. Acesso em: 10 nov. 2007. Citado 1 vez na página 14.
- CUNHA, Alessandro F. O que são sistemas embarcados. *Saber Eletrônica*, v. 43, n. 414, p. 1–6, 2007. Citado 1 vez na página 14.
- EPE. *Balanço Energético Nacional 2023*. Empresa de pesquisa energética. EPE. Jun. 2024. Disponível em: <https://www.epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/PublicacoesArquivos/publicacao-748/topico-687/BEN2023.pdf>. Citado 1 vez na página 13.

- EPE. *Boletim Trimestral do Consumo de Eletricidade*. empresa de pesquisa energética. EPE. Jun. 2024. Disponível em: <https://www.epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/PublicacoesArquivos/publicacao-483/topico-717/Boletim%20Trimestral%20de%20Consumo%20de%20Eletricidade%20ANO%20V%20-%20N%C2%BA17.pdf>. Citado 1 vez na página 16.
- EXPRESS VPS. *O que é Wi-Fi Alliance?* 2025. Acessado em: 16 mar. 2025. Disponível em: <https://expressvps.com.br/glossario/o-que-e-wi-fi-alliance/>. Citado 2 vezes na página 20.
- INSTITUTO DE FÍSICA - UFRGS. *Efeito Hall*. 2025. Acessado em: 18 mar. 2025. Disponível em: https://www.if.ufrgs.br/tex/fis142/mod08/m_s03.html. Citado 1 vez na página 23.
- LIMAI, Cristiane Alkmin Junqueira SchmidtI; Marcos A. M. *A demanda por energia elétrica no Brasil*. revista Brasileira de economia. Scielo Brazil. Set. 2004. Disponível em: <https://www.scielo.br/j/rbe/a/Bnvng4fCCnhvns9cRBYTSHB/#>. Citado 1 vez na página 16.
- LUCAS BRAGA E ANA MARQUES. *O que é Wi-Fi e como funciona?* 2025. Acessado em: 16 mar. 2025. Disponível em: <https://tecnoblog.net/respone/o-que-e-wi-fi-como-funciona/>. Citado 1 vez na página 20.
- MAKERHERO. *Arduino IDE: o que é e como instalar a ferramenta*. Acesso em: 16 abr. 2025. 2020. Disponível em: <https://www.makehero.com/blog/arduino-ide/>. Citado 1 vez na página 22.
- MEDEIROS, Aérton et al. Monitoramento Da Carga Elétrica Residencial Aplicado AO Gerenciamento De Resposta A Demanda. *Brazilian Journal of Development*, v. 1, p. 6, mai. 2018. Citado 2 vezes nas páginas 17, 24.
- MORENO, Luiz Fernando Novaes. Monitoramento, Automação e Eficiência Energética, Em Residência Familiar, Para Redução Da Conta De Energia Elétrica. *Repositório Institucional do Conhecimento - RIC-CPS*, Repositório Institucional do Conhecimento - RIC-CPS, p. 34–42, 2021. Citado 2 vezes na página 17.
- MURATORI, José Roberto; DAL BÓ, Paulo Henrique. Capítulo I Automação residencial: histórico, definições e conceitos. *O Setor elétrico*, p. 70–77, 2011. Citado 1 vez na página 15.
- REMESSA ONLINE. *Firestore: descubra para que serve, como funciona e como usar*. 2023. Disponível em: <https://www.remissaonline.com.br/blog/firebase-descubra-para-que-serve-como-funciona-e-como-usar/>. Acesso em: 15 mar. 2025. Citado 1 vez na página 21.

- REVISTAFERRAMENTAL. *Terceira revolução industrial: causas, características e consequências*. BR: RevistaFerramental, 2022. O site da Revista Ferramental é um veículo de comunicação que aborda temas relacionados à indústria metal-mecânica, como usinagem, estampagem, ferramentaria e indústria 4.0. Disponível em: <https://www.revistaferramental.com.br/artigo/terceira-revolucao-industrial-causas-caracteristicas-e-consequencias#>. Citado 1 vez na página 13.
- ROBOCORE (Ed.). *Fonte Ajustável para ProtoBoard*. 2024. Acessado em: 22 mar. 2024. Disponível em: <https://www.robocore.net/regulador-de-tensao/fonte-ajustavel-para-protoboard>. Citado 0 vez na página 26.
- ROBOCORE. *Atuadores e Relés*. BR: Robocore, 2024. site de compras. Disponível em: https://www.robocore.net/atuador-rele/modulo-rele?gad_source=1&gclid=Cj0KCQjwo8S3BhDeARIsAFRmk0PS3PFb0TRssSOP9sxSXBY08nJY_nypRXIoNkSC7dLHcTssK5oNgkUaAvgQEALw_wcB. Citado 1 vez na página 25.
- ROBOCORE. *Sensor de Corrente ACS712 30A*. BR: Robocore, 2024. site de compras componentes. Disponível em: <https://www.robocore.net/sensor-corrente-tensao/sensor-de-corrente-acs712-30a>. Citado 0 vez na página 24.
- SANTOS, Bruno P; SILVA, Lucas A et al. Internet das coisas: da teoria à prática. *Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, v. 31, p. 16, 2016. Citado 1 vez na página 14.
- SANTOS, Erik Thauan Mendes; MENDES, Douglas Heberte; MARTINS, Rodrigo Antonio. Eficiência energética e redução de custos utilizando equipamentos automatizados para gerenciar os gastos de energia elétrica em uma residência. *Brazilian Applied Science Review*, v. 5, n. 3, p. 1738–1752, 2021. ISSN 2447-0740. Disponível em: <https://ojs.brazilianjournals.com.br/ojs/index.php/BASR/article/view/17281/14046>. Citado 2 vezes nas páginas 17, 18.
- SILVA, Rogério Oliveira da; ARAUJO, Warley Monteiro; CAVALCANTE, Maxwell Machado. Visão geral sobre microcontroladores e prototipagem com Arduino. *Tecnologias Em Projeção*, v. 10, n. 1, p. 36–46, 2019. Citado 2 vezes na página 18.
- SYSTEMS, Espressif. *ESP32-DevKitC User Guide*. Acessado em: 17 de março de 2025. 2025. Disponível em: https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32/esp32-devkitc/user_guide.html#functional-description. Citado 0 vez nas páginas 19, 20.

Anexos

ANEXO A – Datasheet do Sensor ACS712

O Datasheet do sensor de corrente utilizado pode ser acessado pelo link: [datasheet do sensor](#)

ANEXO B – Datasheet do Relé

O Datasheet completo do relé utilizado pode ser acessado pelo link:

[Datasheet do Relé](#)

Apêndices

.1 código desenvolvido Arduino IDE.

O código desenvolvido para o ESP32 na plataforma Arduino IDE foi copiado para um bloco de notas e submetido na plataforma Github, disponível no link: [Clique aqui para acessar](#)

.2 Tutorial de configurações

.2.1 Adaptação da Arduino IDE para o ESP32

Após a instalação do Arduino IDE, para utilizar a plataforma na programação do ESP32, é necessário a realização de algumas configurações seguindo o passo a passo descrito a seguir.

Primeiro, clicar em preferências. Na aba de configurações, inserir a URL https://dl.espressif.com/dl/package_esp32_index.json e clicar em ok. Depois, clicar em Ferramentas / Placa: / Gerenciador de placas. Conforme descrito nas Figuras 29 e 30.

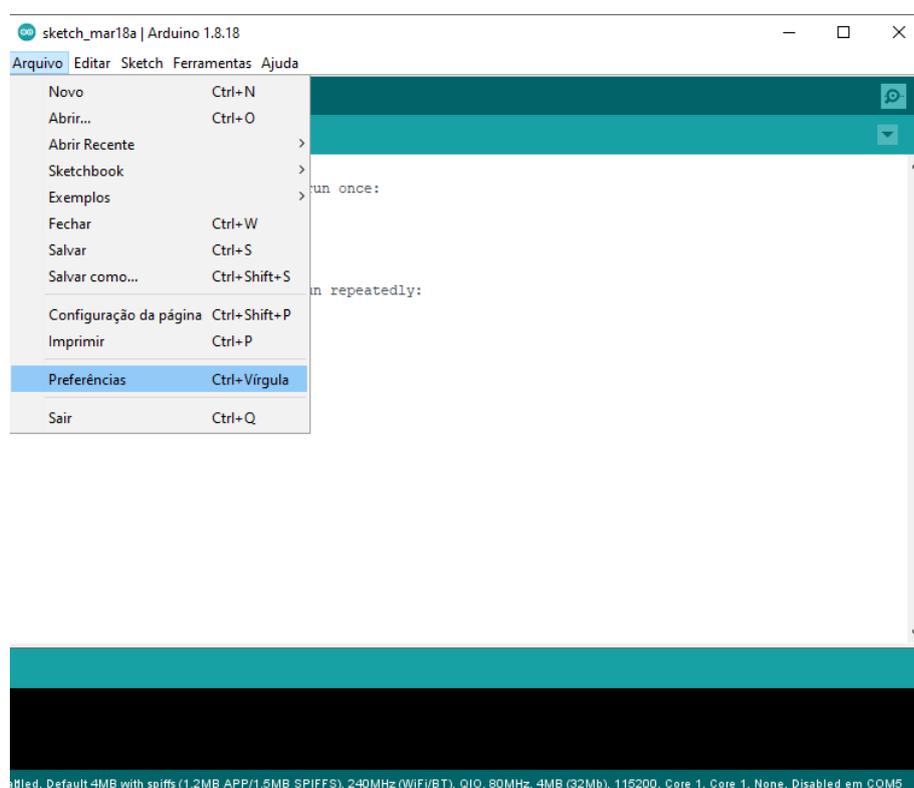


Figura 29 – Tutorial configuração IDE, imagem 1 Fonte: Autoria Própria.

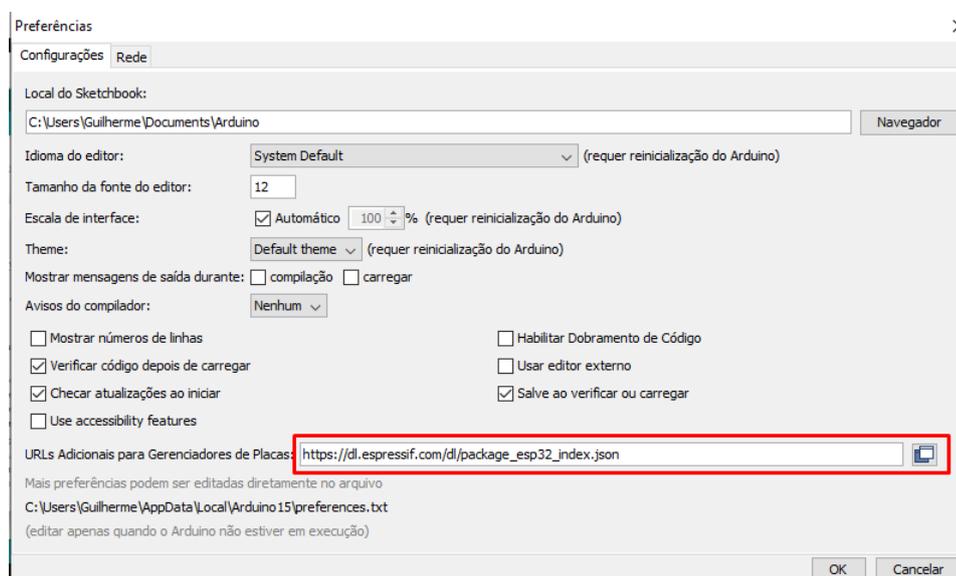


Figura 30 – Tutorial configuração IDE, imagem 2 Fonte: Autoria Própria.

Depois, clicar em Ferramentas / Placa / Gerenciador de placas conforme descrito nas Figura 31.

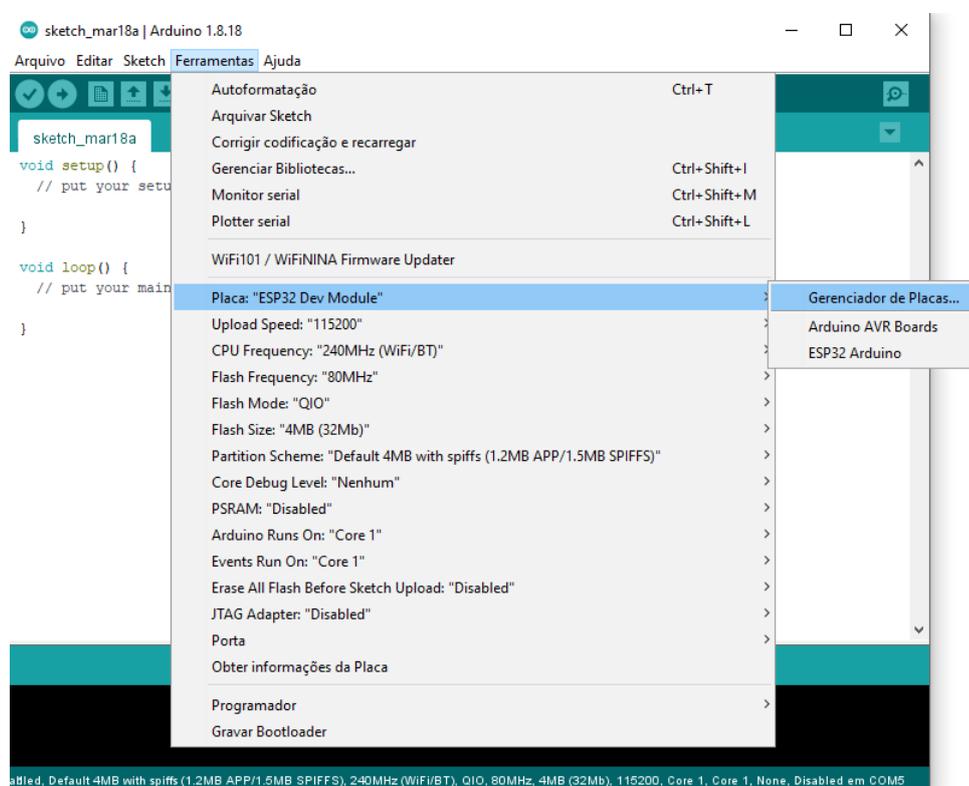


Figura 31 – Tutorial configuração IDE, imagem 3 Fonte: Autoria Própria.

Por fim, procurar por ESP32 e instalar a versão desejada, etapa ilustrada na Figura 13.

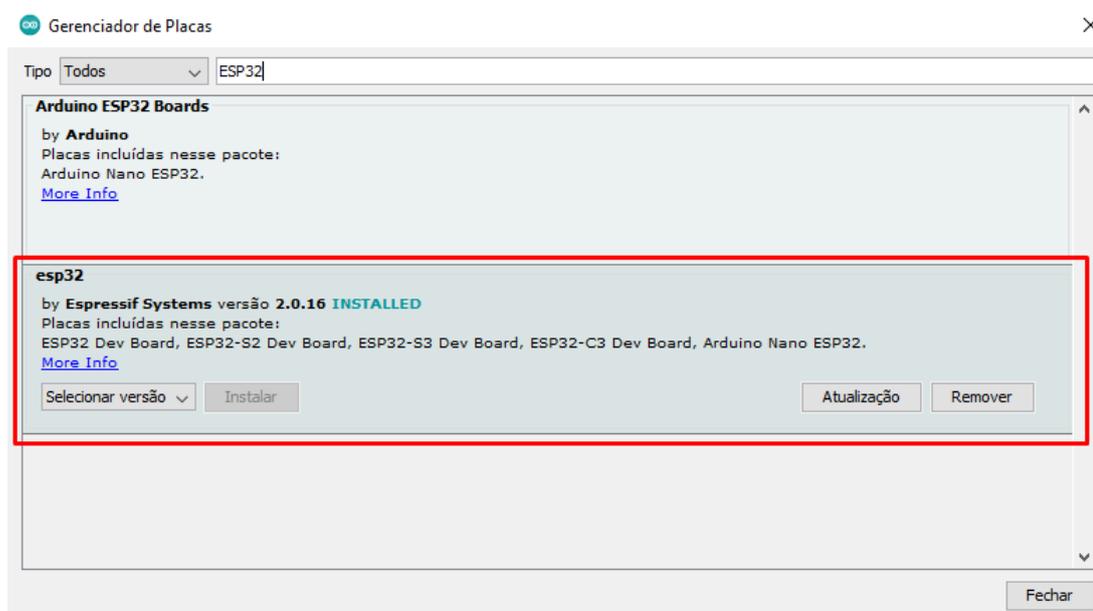


Figura 32 – Tutorial configuração IDE, imagem 4. Fonte: Autoria Própria.

.2.2 Configuração do Wi-Fi

Para a configuração do Wi-Fi, foi utilizada a biblioteca WiFi.h, utilizando o método de instalação de bibliotecas em arquivos “.zip”. A listagem 1 mostra a configuração da rede Wi-Fi utilizada no ESP32.

```
1 // Configura o da rede Wi-Fi
2 const char* ssid = "Gui";
3 const char* password = "guiga123";
```

Listing 1 – Configuração da rede Wi-Fi no ESP32

Para a utilização do Wi-Fi, foram criadas variáveis para o nome da rede e senha, bem como criada uma função que inicia o Wi-Fi. A função que inicia o Wi-Fi está ilustrada na listagem 2

```
1 void configurarWiFi() {
2     WiFi.begin(ssid, password);
3     Serial.print("Conectando ao Wi-Fi...");
4     while (WiFi.status() != WL_CONNECTED) {
5         delay(500);
6         Serial.print(".");
7     }
8     Serial.println("\nWi-Fi conectado.");
9 }
```

Listing 2 – Função para conectar o ESP32 à rede Wi-Fi

.2.3 Comunicação entre Firebase e Arduino IDE

Para o envio de dados, foram utilizadas duas bibliotecas distintas, mas, para a utilização correta das mesmas, antes é preciso obter a chave de acesso do projeto criado no Firebase. Ao entrar no projeto, ir para configurações do projeto/usuários e permissões/contas de serviço /chaves secretas do banco de dados. A Figura 33 ilustra o processo de aquisição da chave, a Figura 34 refere-se ao trecho de código utilizado para configurar o Firebase na plataforma de programação do microcontrolador.

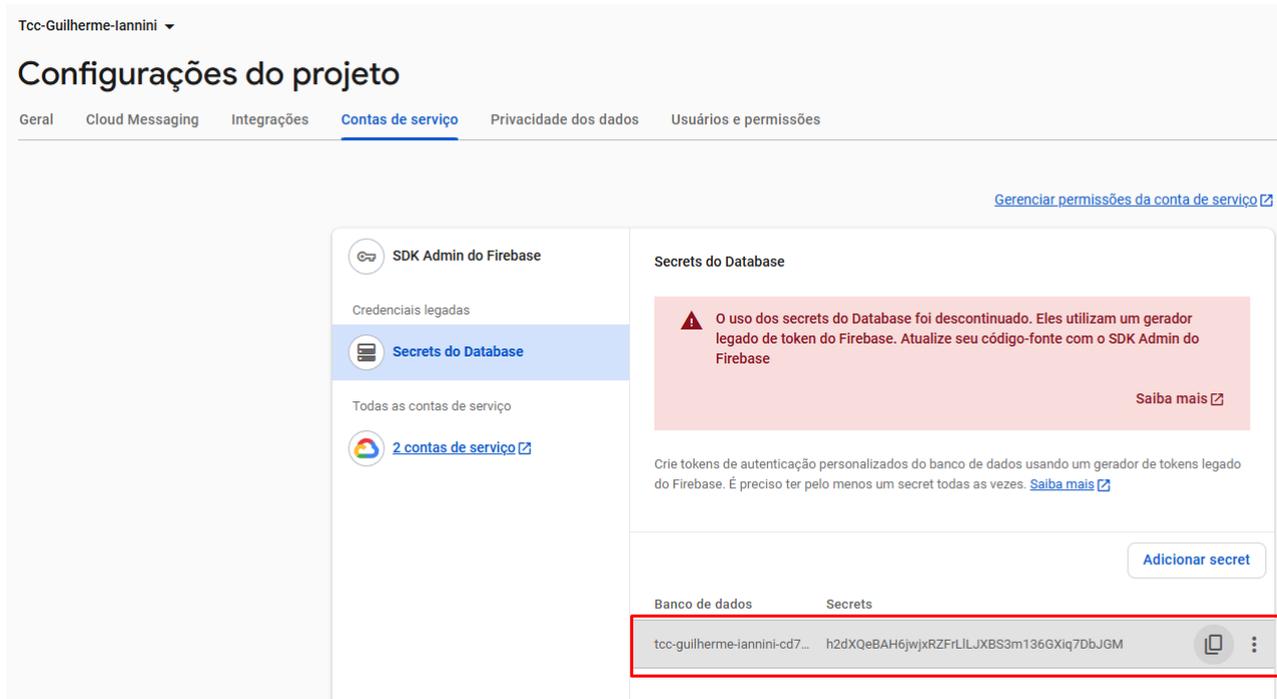


Figura 33 – Chave de acesso. Fonte: Autoria Própria.

```
#include <WiFi.h>
#include <IOXhop_FirebaseESP32.h>
#include <ArduinoJson.h>

// Configurações do Firebase
#define FIREBASE_HOST "https://tcc-guilherme-iannini-cd725-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "h2dXQeBAH6jwjxRZFrLlLJXBS3m136GXiq7DbJGM"
```

Figura 34 – Bibliotecas utilizadas na comunicação. Fonte: Autoria Própria.

2.4 Comunicação entre Firebase e Kodular

Foi necessário configurar tanto o projeto criado em Kodular, quanto o criado no firebase, para a comunicação de ambos. Seguindo o seguinte passo a passo:

1. Entrar no projeto criado no firebase, e clicar em adicionar aplicativo / aplicativo android, conforme as Figuras 35, 36 e 37.

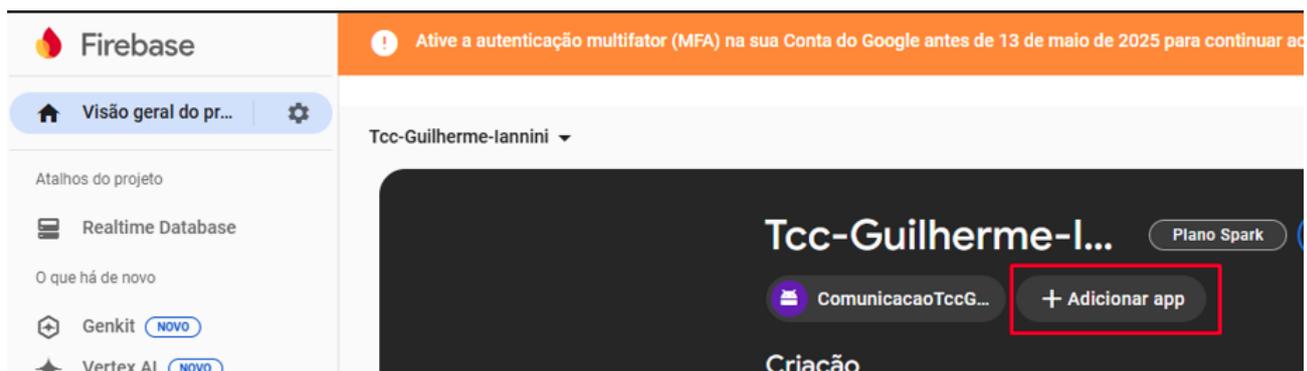


Figura 35 – Explicação do passo 1, imagem 1. Fonte: Autoria Própria.



Figura 36 – Explicação do passo 1, imagem 2. Fonte: Autoria Própria.

4. Realizar o *download* do arquivo `google-services.json`, dentro do arquivo, existe uma aba denominada *current key*, que será necessária no kodular. O *download* é feito conforme descrito na Figura 41.

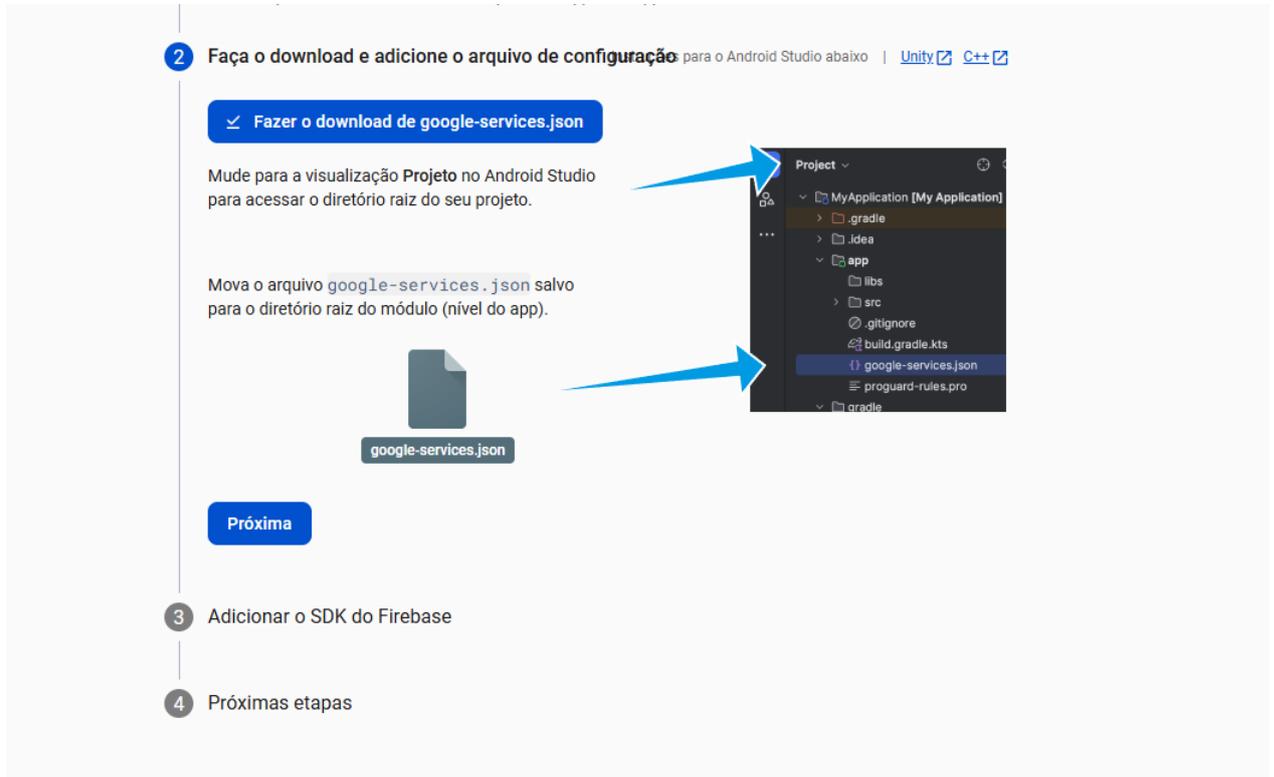


Figura 41 – Download arquivo `google-services.json`. Fonte: Autoria Própria.

5. Finalizar as configurações seguindo o passo a passo do Firebase.
6. Copiar o link de identificação do banco de dados firebase. O link em questão é demonstrado na Figura 42.



Figura 42 – link identificação firebase. Fonte: Autoria Própria.

7. Ir até o aplicativo criado na plataforma kodular / paleta / Google / firebase DB e adicionar o bloco firebase BD no projeto, clicando e arrastando até a tela do telefone, conforme a Figura 43.

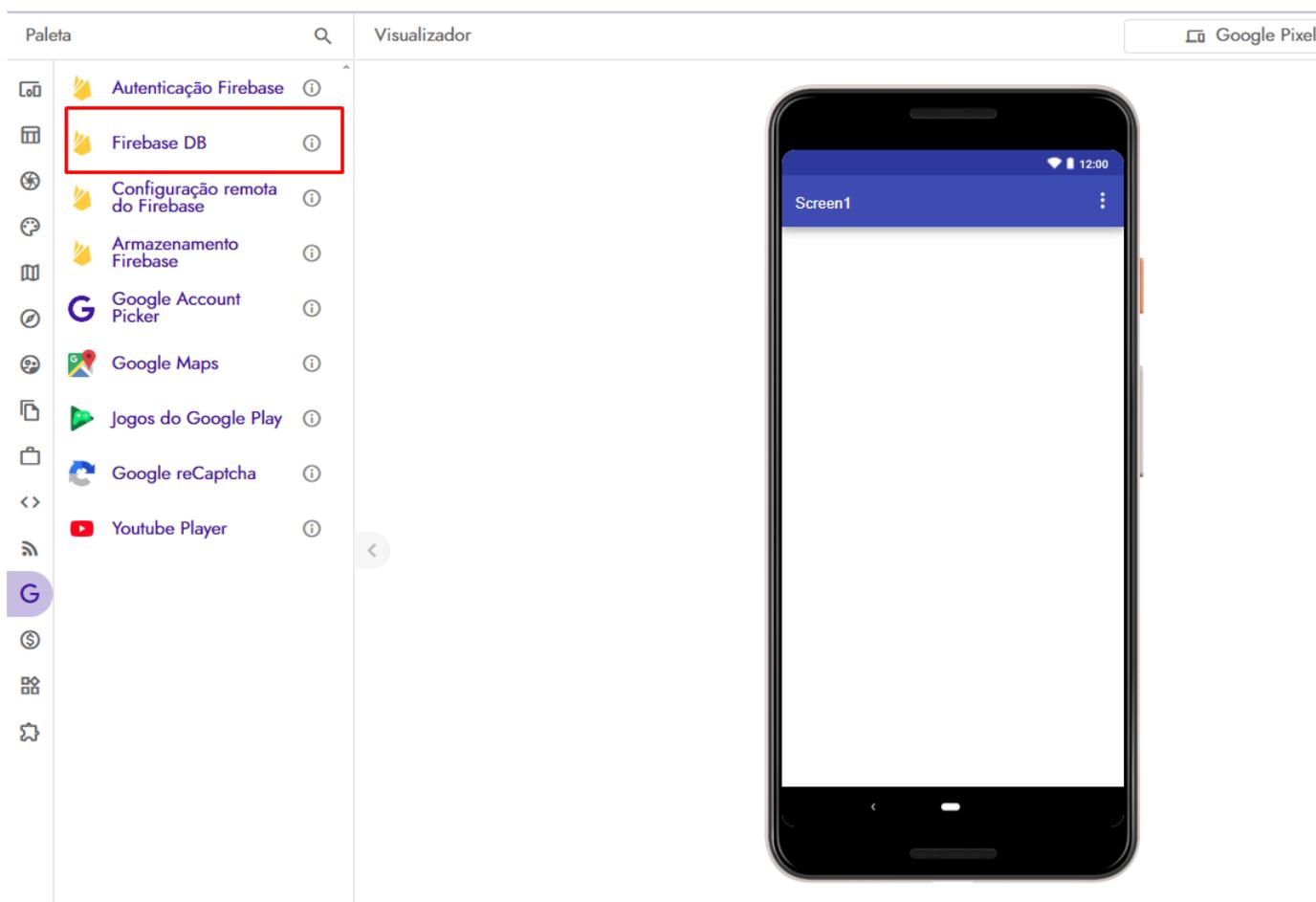


Figura 43 – Bloco Firebase DB. Fonte: Autoria Própria.

8. Copiar o link de de identificação do Firebase (passo 6) e então adicionar na aba Firebase *URL*. Copiar também o firebase token localizado no arquivo baixado no passo 4. A chave de acesso está destacada na Figura 44, e o local de sua utilização na Figura 45.

```
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
google-services.json X
C:\Users\Guilherme\Downloads\google-services.json > [ ] client > { } 0 > [ ] api_key > { } 0 > [ ] current_key
1
2 {
3   "project_info": {
4     "project_number": "499014704976",
5     "project_id": "teste-1864b",
6     "storage_bucket": "teste-1864b.firebaseioapp"
7   },
8   "client": [
9     {
10      "client_info": {
11        "mobilesdk_app_id": "1:499014704976:android:760aea938d335a725bd655",
12        "android_client_info": {
13          "package_name": "com.teste.kd"
14        }
15      },
16      "oauth_client": [],
17      "api_key": [
18        {
19          "current_key": "AIzaSyCSf98RgnTmsIjswiHcUEGqO6o_e2II1CQ"
20        }
21      ],
22      "services": {
23        "appinvite_service": {
24          "other_platform_oauth_client": []
25        }
26      }
27    }
28  ],
29  "configuration_version": "1"
}
```

Figura 44 – Bloco Firebase DB. Fonte: Autoria Própria.

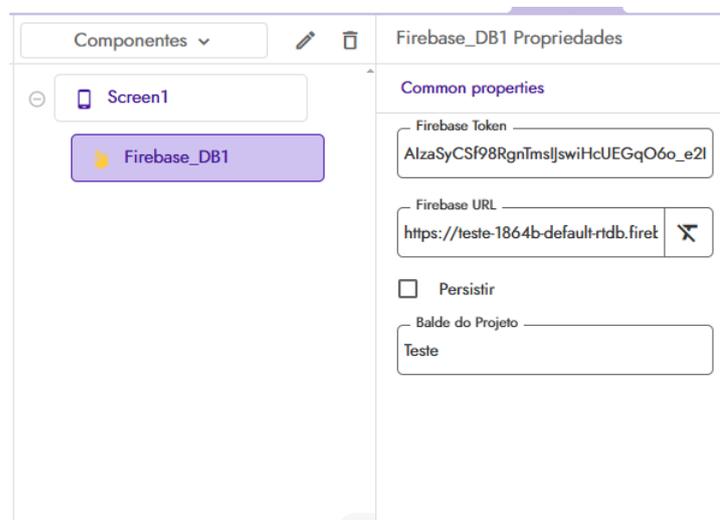


Figura 45 – Bloco Firebase DB. Fonte: Autoria Própria.

.3 Programação de blocos Kodular

.3.1 Tela Inicial

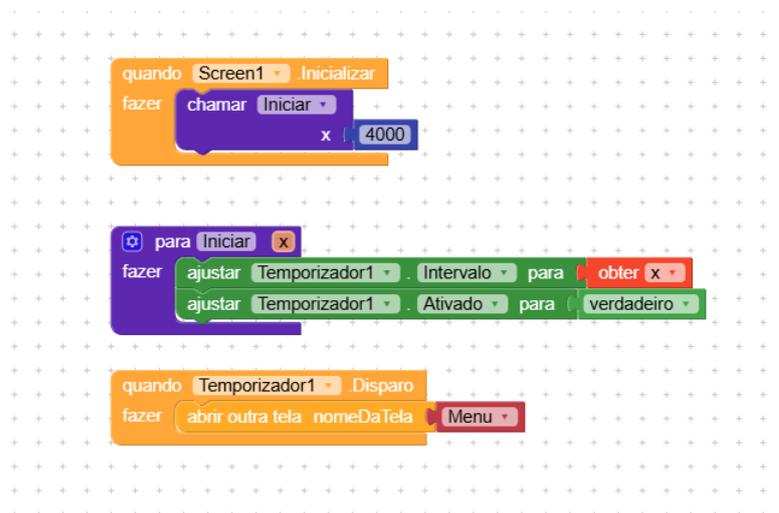


Figura 46 – Programação de blocos da Tela inicial do app. Fonte: Autoria Própria.

.3.2 Tela do Menu

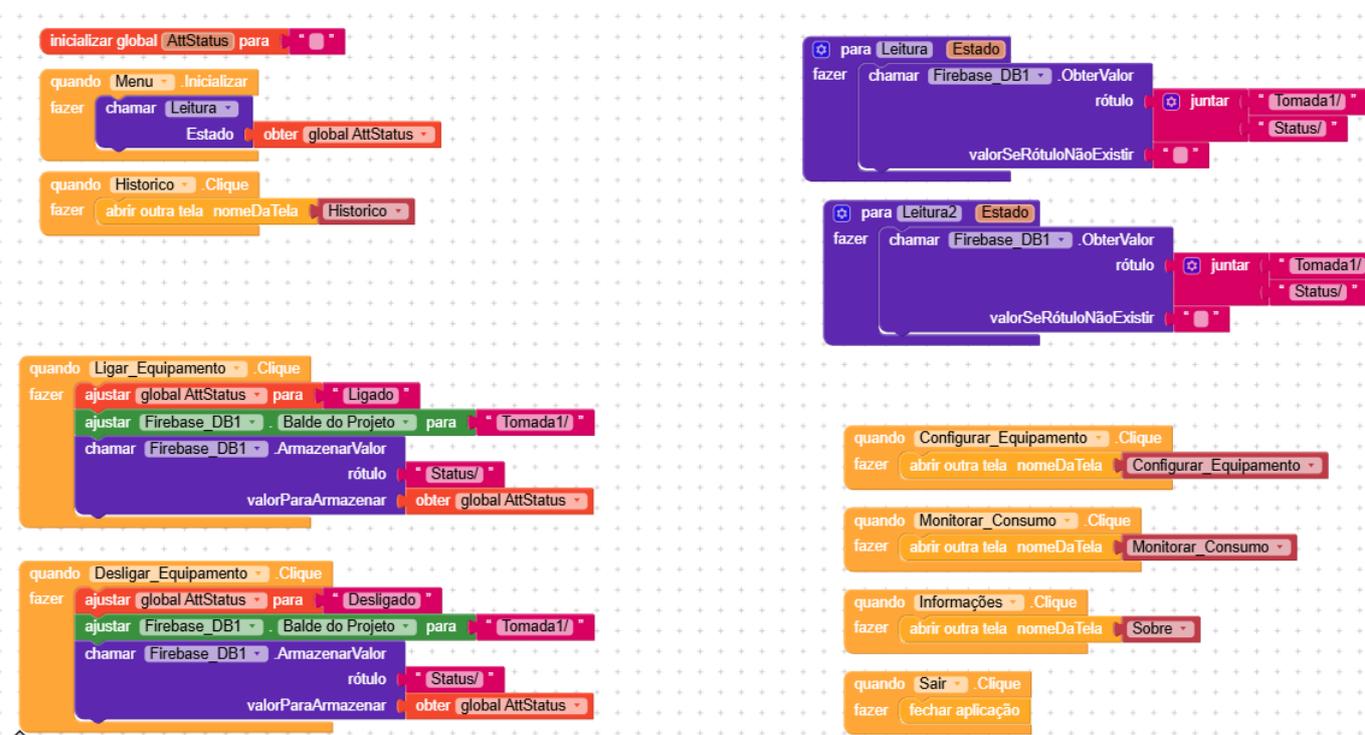


Figura 47 – Programação de blocos do Menu. Fonte: Autoria Própria.

.3.3 Tela Configurar Equipamentos



Figura 48 – Programação de blocos tela configurar equipamentos (Imagem 1). Fonte: Autoria Própria.

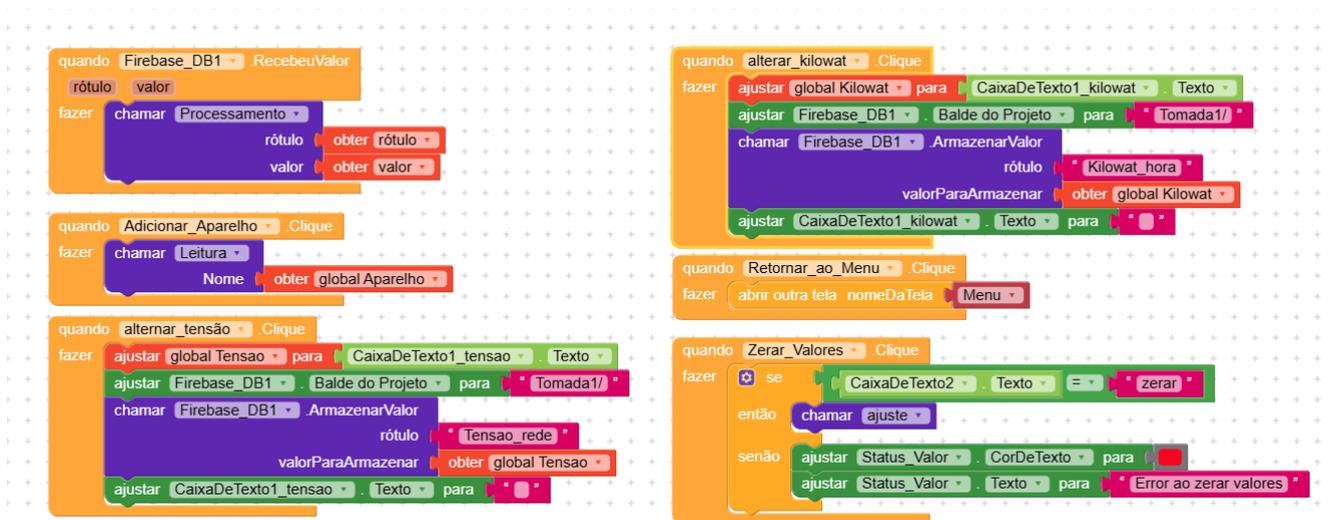


Figura 49 – Programação de blocos tela configurar equipamentos (Imagem 2). Fonte: Autoria Própria.



Figura 50 – Programação de blocos tela configurar equipamentos (Imagem 3). Fonte: Autoria Própria.

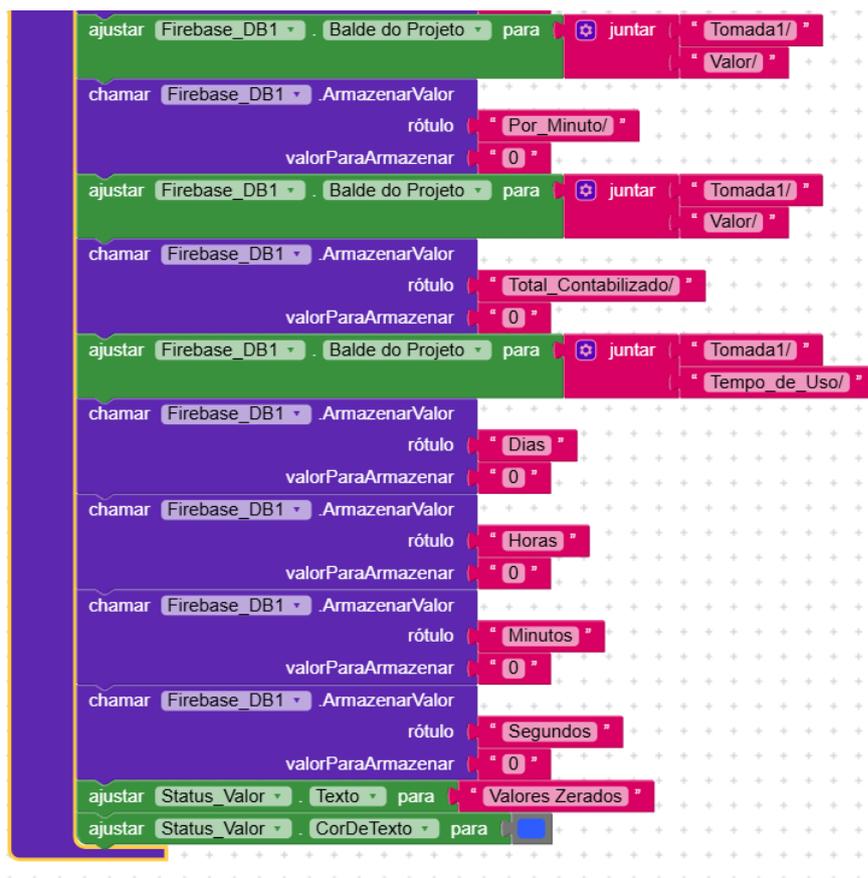


Figura 51 – Programação de blocos tela configurar equipamentos (Imagem 4). Fonte: Autoria Própria.

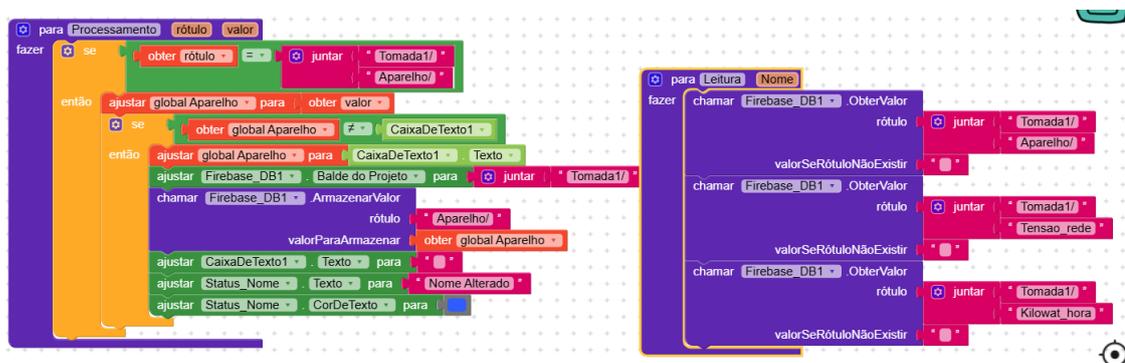


Figura 52 – Programação de blocos tela configurar equipamentos (Imagem 5). Fonte: Autoria Própria.

.3.4 Tela Histórico

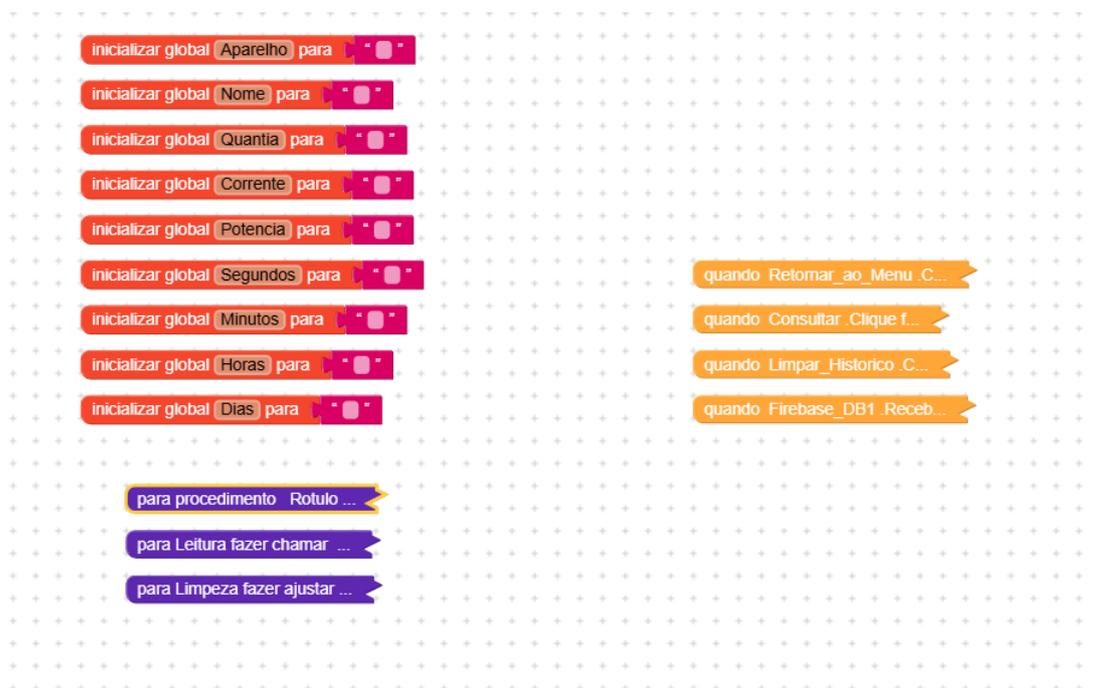


Figura 53 – Programação de blocos tela histórico (Imagem 1). Fonte: Autoria Própria.

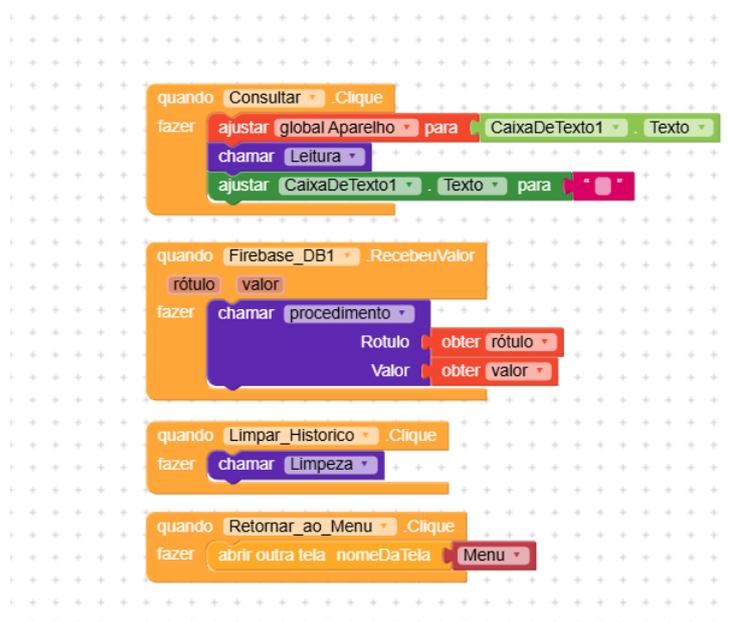


Figura 54 – Programação de blocos tela histórico (Imagem 2). Fonte: Autoria Própria.

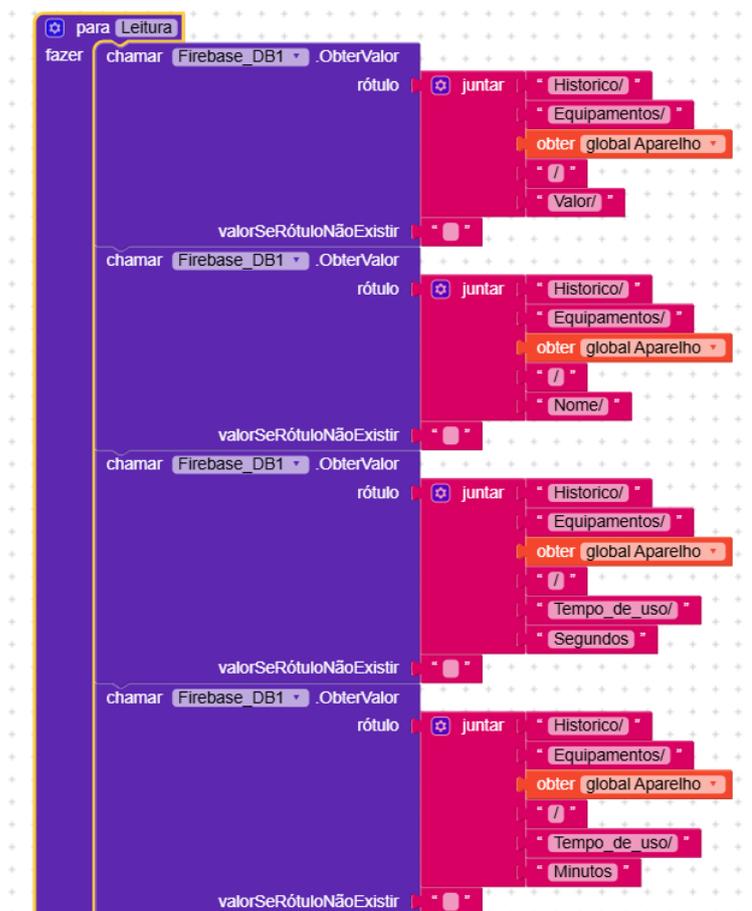


Figura 55 – Programação de blocos tela histórico (Imagem 3). Fonte: Autoria Própria.

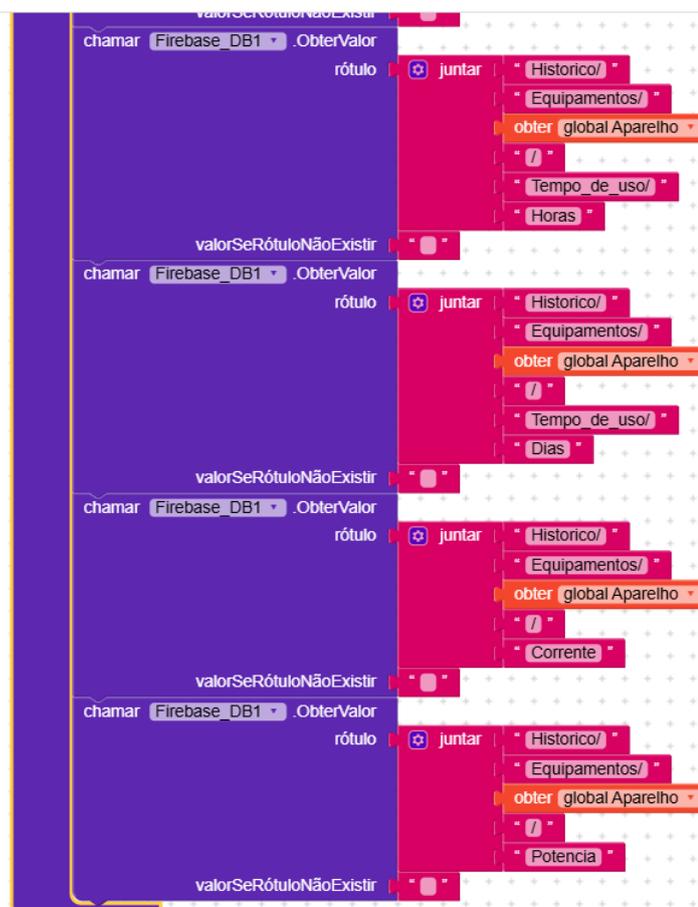


Figura 56 – Programação de blocos tela histórico (Imagem 4). Fonte: Autoria Própria.



Figura 57 – Programação de blocos tela histórico (Imagem 5). Fonte: Autoria Própria.

.3.5 Monitorar Consumo

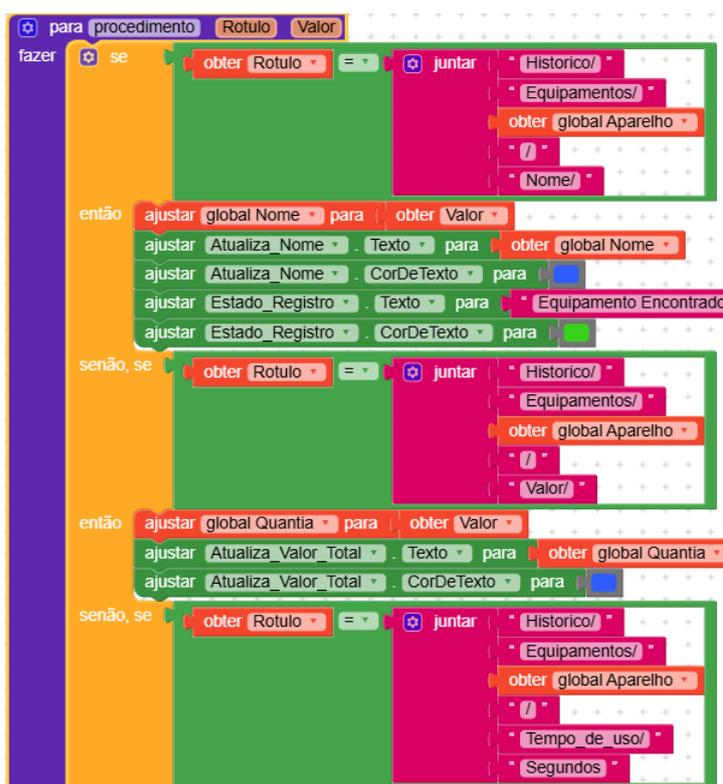


Figura 58 – Programação de blocos tela histórico (Imagem 6). Fonte: Autoria Própria.

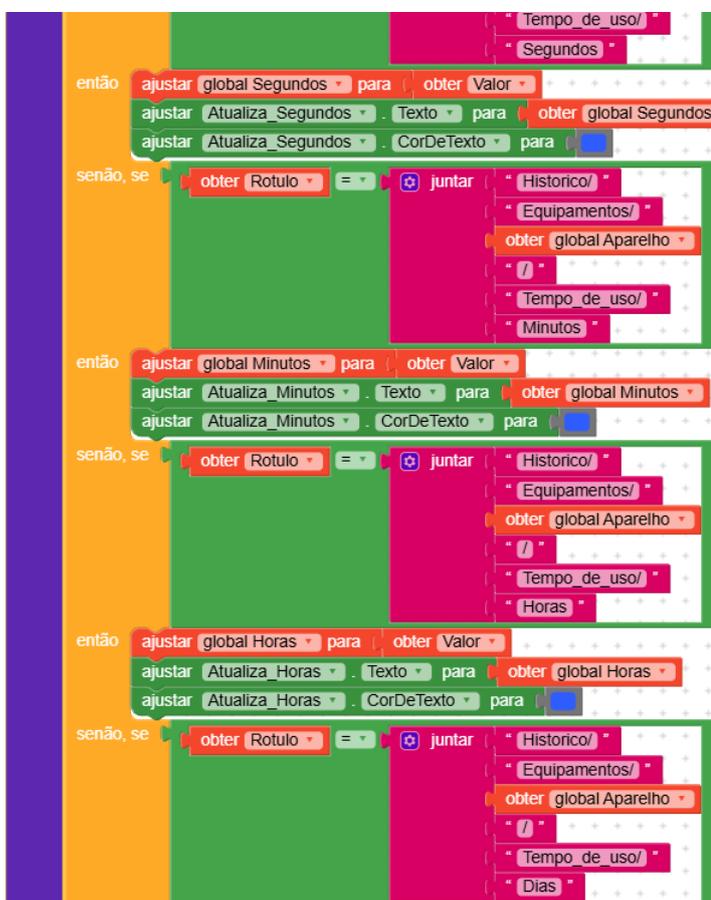


Figura 59 – Programação de blocos tela histórico (Imagem 7). Fonte: Autoria Própria.

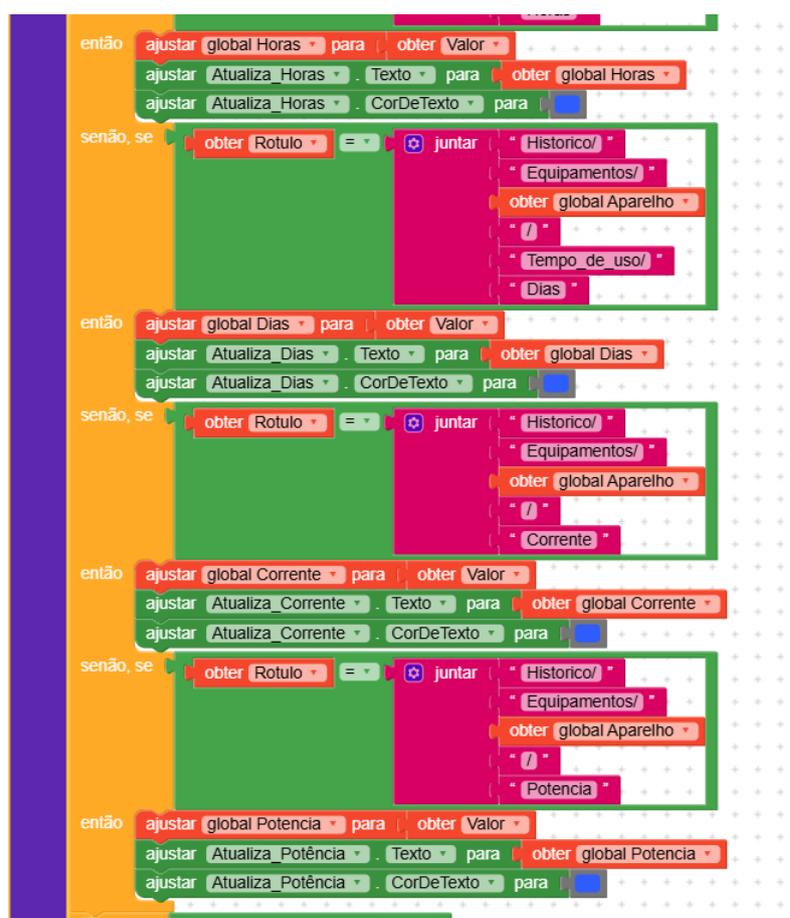


Figura 60 – Programação de blocos tela histórico (Imagem 8). Fonte: Autoria Própria.

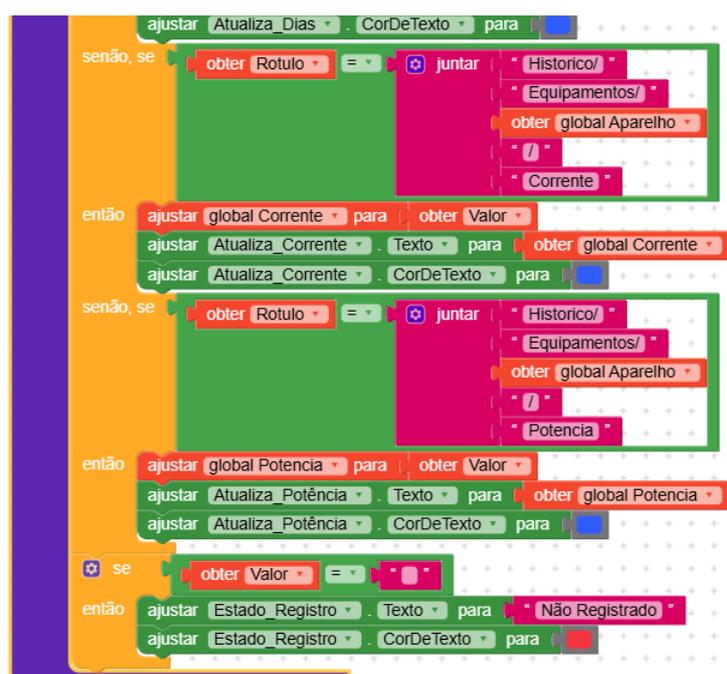


Figura 61 – Programação de blocos tela histórico (Imagem 9). Fonte: Autoria Própria.

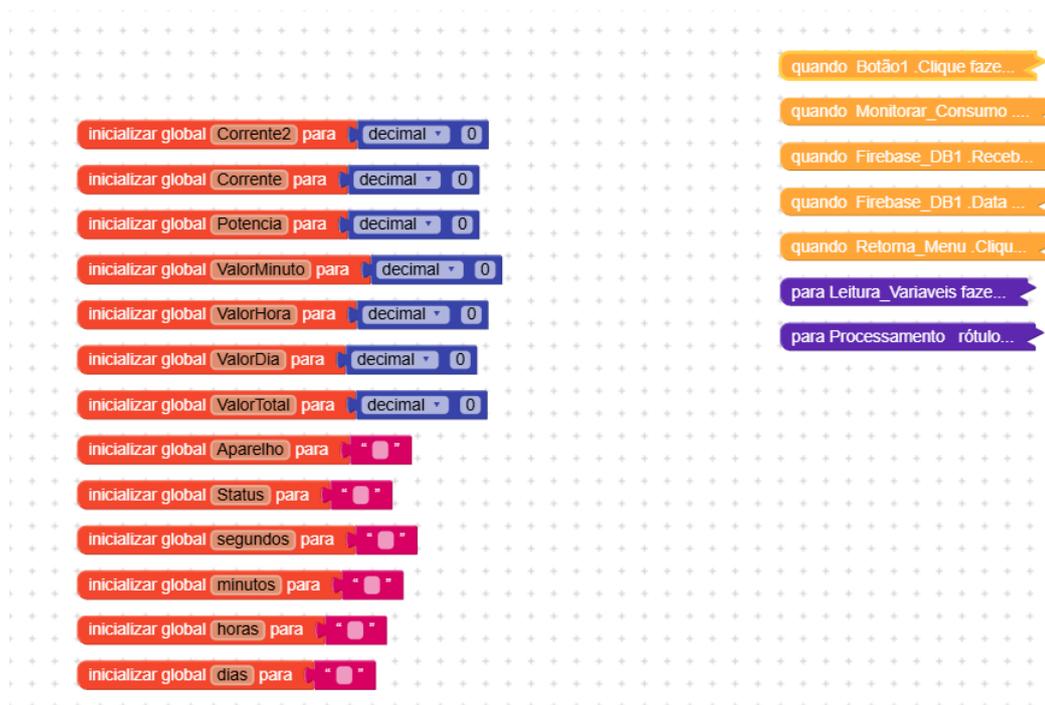


Figura 62 – Programação de blocos Monitorar Consumo (Imagem 1). Fonte: Autoria Própria.

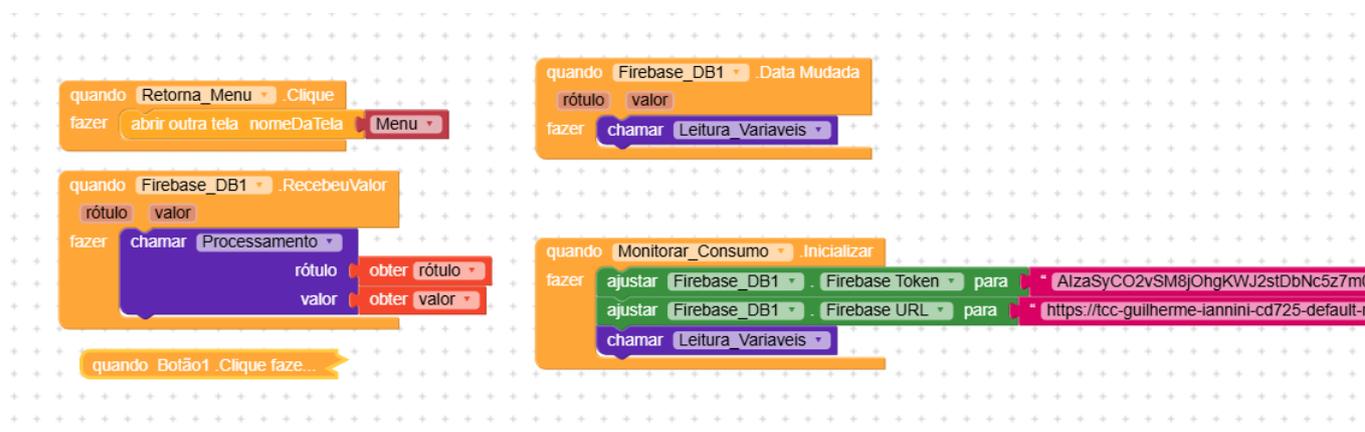


Figura 63 – Programação de blocos Monitorar Consumo (Imagem 2). Fonte: Autoria Própria.

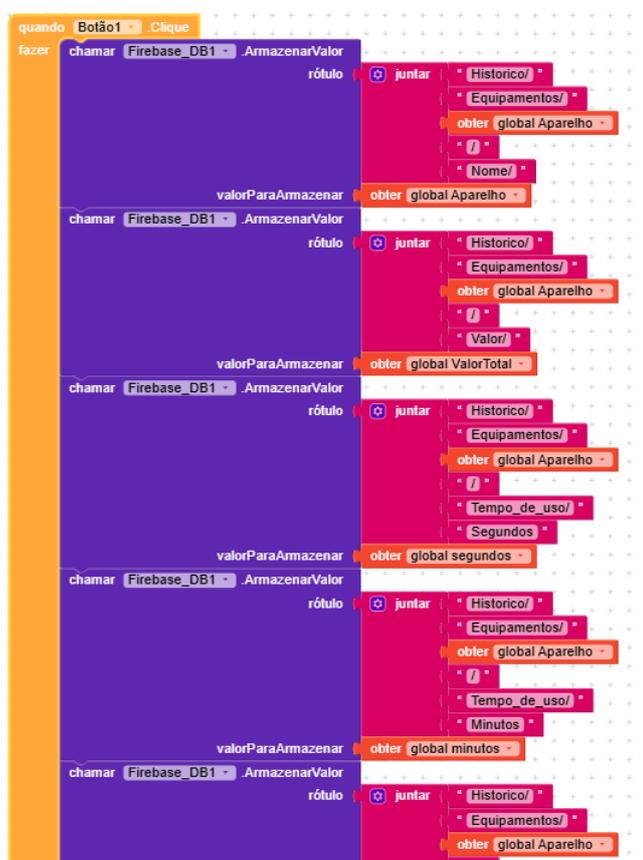


Figura 64 – Programação de blocos Monitorar Consumo (Imagem 3). Fonte: Autoria Própria.

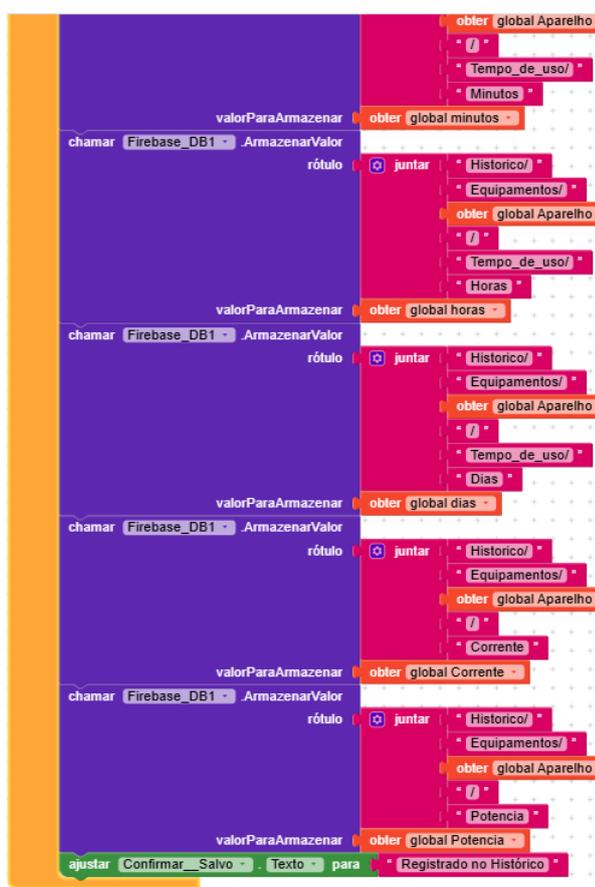


Figura 65 – Programação de blocos Monitorar Consumo (Imagem 4). Fonte: Autoria Própria.



Figura 66 – Programação de blocos Monitorar Consumo (Imagem 5). Fonte: Autoria Própria.

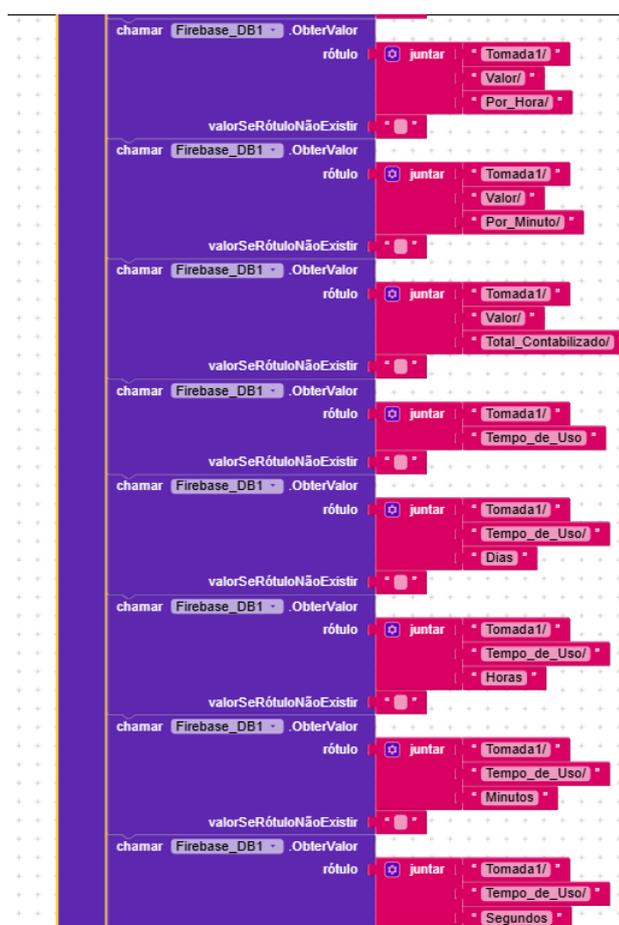


Figura 67 – Programação de blocos Monitorar Consumo (Imagem 6). Fonte: Autoria Própria.

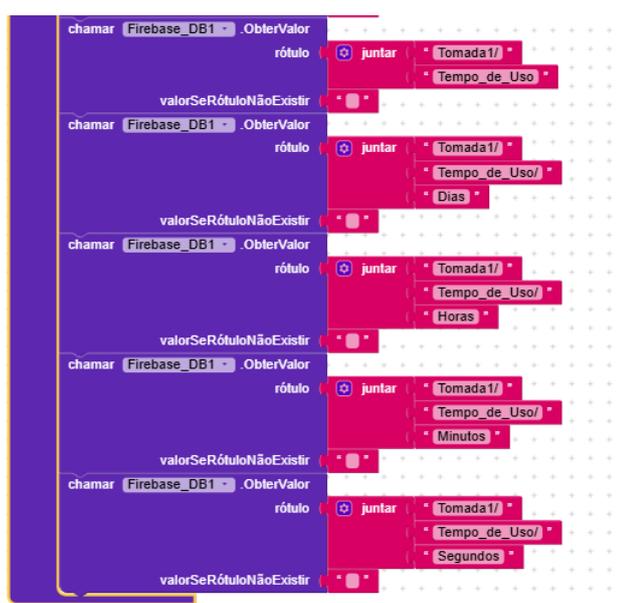


Figura 68 – Programação de blocos Monitorar Consumo (Imagem 7). Fonte: Autoria Própria.

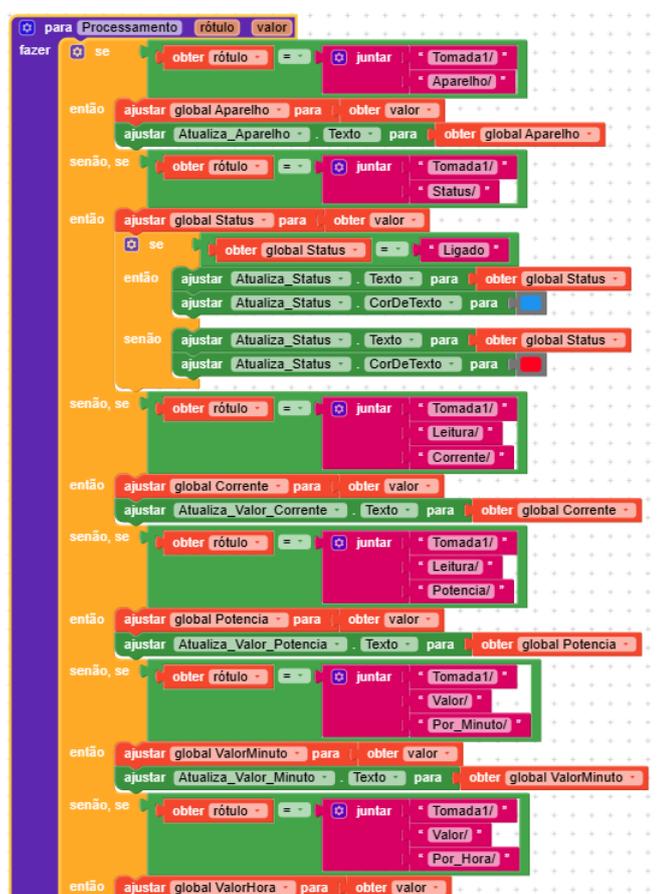


Figura 69 – Programação de blocos Monitorar Consumo (Imagem 8). Fonte: Autoria Própria.

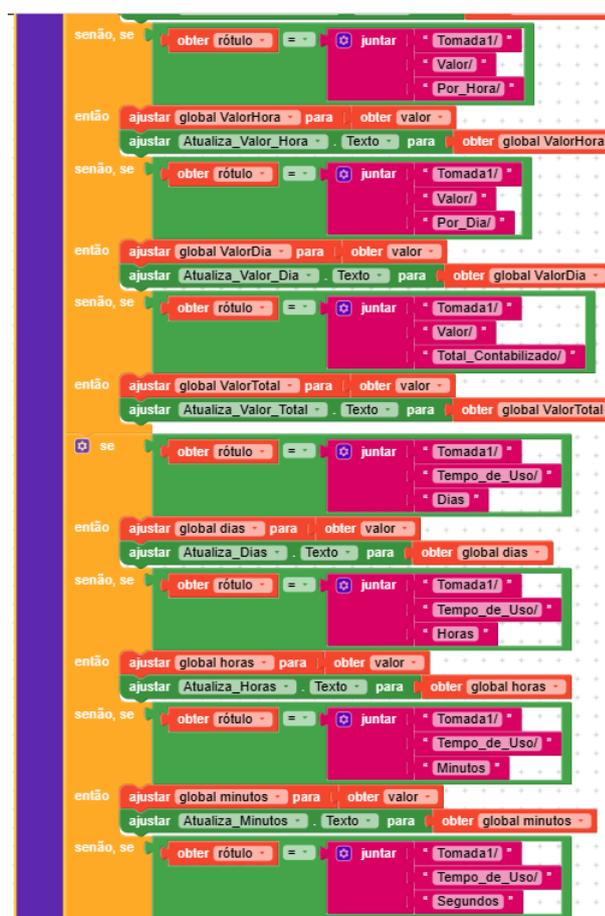


Figura 70 – Programação de blocos Monitorar Consumo (Imagem 9). Fonte: Autoria Própria.

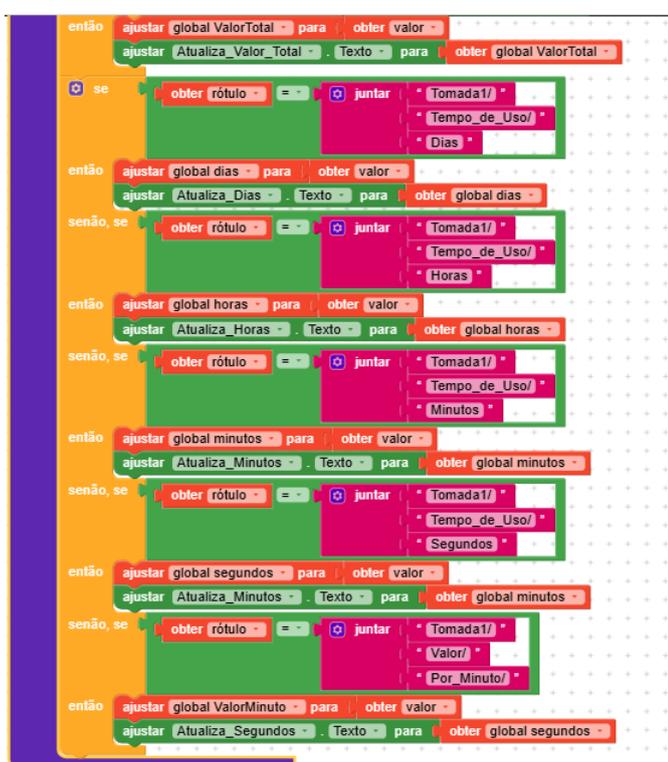


Figura 71 – Programação de blocos Monitorar Consumo (Imagem 10). Fonte: Autoria Própria.