

INSTITUTO FEDERAL DE SANTA CATARINA
CURSO DE ENGENHARIA ELÉTRICA

VINICIUS HERCILIO CORREA

**DESENVOLVIMENTO DE UM SISTEMA PARA MONITORAMENTO
DE CONSUMO DE ENERGIA ELÉTRICA RESIDENCIAL**

ITAJAÍ
2025

Vinicius Hercilio Correa

Desenvolvimento de um sistema para monitoramento de consumo de
energia elétrica residencial

Monografia apresentada ao curso de
Engenharia Elétrica do Instituto Federal de
Santa Catarina, para obtenção do título de
bacharel em engenharia elétrica.

Área de concentração: Medidas Elétricas

Orientadora: Prof.^a Dra. Fernanda Isabel
Marques Argoud da Silva

Itajaí
2025

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca do IFSC.

Correa, Vinicius Hercilio

Desenvolvimento de um sistema para monitoramento de consumo de energia elétrica residencial / Vinicius Hercilio Correa ; orientador, Fernanda Isabel Marques Argoud da Silva, 2025.

86 p.

Trabalho de Conclusão de Curso (graduação) - Instituto Federal de Santa Catarina, Campus Itajaí, Graduação em Engenharia elétrica, Itajaí, 2025.

Inclui referências.

1. Engenharia elétrica. 2. Monitoramento em tempo real. 3. Sistemas microcontrolados. 4. Eficiência energética. I. Silva, Fernanda Isabel Marques Argoud da. II. Instituto Federal de Santa Catarina. Graduação em Engenharia elétrica. III. Título.


DESENVOLVIMENTO DE UM APLICATIVO ANDROID PARA MONITORAMENTO DE CONSUMO DE ENERGIA ELÉTRICA RESIDENCIAL

Vinicius Hercílio Correa


Este trabalho foi julgado adequado para obtenção do Título de Engenheiro Eletricista e aprovado na sua forma final pela banca examinadora do curso de engenharia elétrica do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Itajaí, 13 de outubro de 2025.


BANCA EXAMINADORA:

Documento assinado digitalmente
 **FERNANDA ISABEL MARQUES ARGOUD DA SILVA**
Data: 10/11/2025 15:20:54-0300
Verifique em <https://validar.iti.gov.br>

Dr. Fernanda Isabel Marques Argoud da Silva
Instituto Federal de Santa Catarina – IFSC

Documento assinado digitalmente
 **DOUGLAS ALEXANDRE RODRIGUES DE SOUZA**
Data: 11/11/2025 18:17:26-0300
Verifique em <https://validar.iti.gov.br>

M.Sc. Douglas Alexandre Rodrigues de Souza
Instituto Federal de Santa Catarina – IFSC

Documento assinado digitalmente
 **SERGIO AUGUSTO BITENCOURT PETROVIC**
Data: 10/11/2025 15:35:22-0300
Verifique em <https://validar.iti.gov.br>

Dr. Sérgio Augusto Bitencourt Petrovic
Instituto Federal de Santa Catarina - IFSC

AGRADECIMENTOS

Primeiramente, gostaria de agradecer a minha mãe Patricia Janete da Conceição, uma mulher incrível, de que em todo o tempo que estive na faculdade, sempre me apoiou, sem ela e as dezenas garrafas de café que fez enquanto eu estudava, com certeza não teria chegado até esse momento.

Outra pessoa que eu gostaria de expressar minha gratidão é a minha namorada Paula Raylene Nolasco Moreno Fernandes, sempre me incentivando e não me deixando desistir, me apoiando nos meus sonhos e fazendo minha vida ser melhor a cada dia que passa. Obrigado.

Outro agradecimento vai para meus amigos que percorreram o caminho todo da faculdade junto comigo, Gustavo Avelar Cabral e Leonir José da Costa Junior, graças as várias vezes que estudamos, rimos e sofremos juntos. Foi uma trilha difícil, mas chegamos.

Obrigado a todas as outras pessoas que fizeram parte desse caminho, vou levar o aprendizado para sempre em minha vida.

“Quando você sorri pra vida, ela te sorri
de volta” (Autor desconhecido).

RESUMO

O elevado custo da energia elétrica no Brasil, somado à complexidade das faturas e à falta de informações em tempo real, dificulta o gerenciamento eficaz do consumo residencial e limita a capacidade dos consumidores de tomar decisões para reduzir o desperdício. Este trabalho apresenta o desenvolvimento de um sistema de monitoramento de consumo de energia elétrica residencial, composto por um *hardware* de medição e um aplicativo *Android*, chamado *EnergyCheck*, com foco em auxiliar os usuários em Santa Catarina. O *hardware* utiliza um microcontrolador esp32, programado via Arduino IDE, para coletar dados de sensores de corrente (SCT-013) e tensão (ZMPT101B). Os dados são enviados via *Wi-Fi* para o *Firebase Realtime Database*. O aplicativo, desenvolvido em *React Native* e *Expo*, apresenta esses dados ao usuário, permitindo visualizar o consumo diário, semanal e acumulado, estimar o valor da fatura e receber notificações personalizadas. Para avaliar o desempenho do sistema, os sensores foram calibrados contra um instrumento de referência (True RMS), apresentando um erro percentual máximo de 0,44 % para tensão e erros inferiores a 3,2 % para corrente nas faixas típicas de consumo. Em um teste de longa duração (duas semanas) em ambiente real, o consumo total medido pelo sistema (226 kWh) apresentou um desvio de apenas 2,21% em relação ao medidor da concessionária CELESC (231 kWh). Os resultados demonstram a viabilidade técnica e a efetividade do protótipo. O projeto atingiu seus objetivos técnicos ao entregar uma ferramenta útil e com precisão validada, confirmando seu potencial como um instrumento de apoio à gestão e conscientização sobre o consumo de energia elétrica.

Palavras-chave: consumo de energia elétrica, monitoramento em tempo real, eficiência energética, esp32.

ABSTRACT

This work presents the development of EnergyCheck, an Android application for real-time monitoring of residential electricity consumption, aiming to help users control their electricity usage in Santa Catarina. The application, developed using the React Native library and the Expo platform, uses an esp32 microcontroller programmed with Arduino IDE to collect consumption data through current and voltage sensors. The data is sent to a database, allowing users to view their daily, weekly, and accumulated consumption, estimate the bill amount, and track the monthly consumption history. EnergyCheck also offers features such as expense forecasting and personalized notifications, which can help users identify waste and adopt more efficient consumption practices. To evaluate the system's performance, multiple tests were conducted, including sensor calibration and real-environment testing, with a 24-hour experiment and another lasting two weeks. The results demonstrated the project's effectiveness in monitoring energy consumption, with accuracy compatible with the resolution of CELESC's meter. The project achieved its objectives of being a useful and practical tool for monitoring residential electricity consumption, promoting possible awareness and efficient use of electric energy.

Keywords: electricity consumption, real-time monitoring, energy efficiency, esp32.

LISTA DE FIGURAS

Figura 1 - Microcontrolador ESP32.	21
Figura 2 - Sensor de corrente SCT- 013-000.	22
Figura 3 - Sensor de tensão ZMPT101B.	23
Figura 4 - Esquemático do projeto.	24
Figura 5 - Projeto de telas login e registro.	25
Figura 6 - Projeto de telas internas do aplicativo.	26
Figura 7 – Estrutura do Firebase Realtime Database do Energy Check	28
Figura 8 - Alicata amperímetro HA-3120.	30
Figura 9 - Bancada de testes.	31
Figura 10 - Teste do sensor de tensão.	32
Figura 11 - Teste do sensor de tensão no monitor serial do Arduino IDE.	33
Figura 12 - Sistema EnergyCheck instalado na residência.	34
Figura 13 - Medidor de energia no início do teste.	35
Figura 14 - Medidor de energia após 24 horas.	35
Figura 15 - Medidor de energia antes do teste de duas semanas.	37
Figura 16 - Medidor de energia após o teste de duas semanas.	37
Figura 17 - Telas de login, cadastro e recuperação de senha.	39
Figura 18 - E-mail de redefinição de senha.	40
Figura 19 - Tela Home.	41
Figura 20 - Tela relatórios.	42
Figura 21 - Tela configurações.	43
Figura 22 - Notificação recebida.	43
Figura 23 - Tela perfil.	44
Figura 24 - Aplicativo EnergyCheck apresentando os valores medidos.	48
Figura 25 - EnergyCheck exibindo os valores medidos após duas semanas.	50

LISTA DE TABELAS

Tabela 1 - Teste do sensor SCT-013-000	45
Tabela 2 - Teste do sensor ZMPT101B	47
Tabela 3 - Comparação entre o consumo medido pelo <i>EnergyCheck</i> e o medidor da CELESC	49
Tabela 4 - Comparação entre o consumo medido pelo <i>EnergyCheck</i> e o medidor da CELESC	50

SUMÁRIO

1 INTRODUÇÃO	12
1.1 JUSTIFICATIVA.....	12
1.1.1 A importância da energia elétrica e seu impacto no desenvolvimento socioeconômico	13
1.1.2 A relação entre consumo de energia, desenvolvimento e renda	14
1.1.3 O consumo de energia elétrica no setor residencial e o impacto nas famílias brasileiras	14
1.1.4 A importância das soluções digitais no monitoramento de consumo de energia elétrica	15
1.1.5 Tarifas de energia elétrica e bandeiras tarifárias	16
1.1.6 Consumo de energia elétrica e unidade de medida	16
1.2 OBJETIVOS	17
1.2.1 Objetivo geral	17
1.2.2 Objetivos específicos	17
1.3 ORGANIZAÇÃO DO TRABALHO.....	18
2 REFERENCIAL TEÓRICO	19
2.1 SISTEMAS MICROCONTROLADOS PARA MONITORAMENTO ENERGÉTICO	19
2.2 APLICATIVO <i>ANDROID</i> PARA MONITORAMENTO DO CONSUMO DE ENERGIA ELÉTRICA DE UMA RESIDÊNCIA	19
2.3 MONITORAMENTO DO CONSUMO DE ENERGIA ELÉTRICA E CONTROLE DE EQUIPAMENTOS VIA APLICATIVO.....	20
2.4 ANÁLISE DOS TRABALHOS RELACIONADOS	20
3 METODOLOGIA	21
3.1 <i>HARDWARE</i>	21
3.2 <i>SOFTWARE</i>	24
3.2.1 Figma	24
3.2.2 Firebase realtime database e authentication	27
3.2.3 Desenvolvimento do aplicativo	29
3.3 PROCEDIMENTO DOS TESTES.....	29
3.3.1 Validação do Sensor de Corrente (SCT-013-000)	30
3.3.2 Validação do Sensor de Tensão (ZMPT101B)	31

3.3.3 Teste de Curta Duração (24 Horas) em Ambiente Real	33
3.3.4 Teste de Longa Duração (Duas Semanas) em Ambiente Real,	36
3.3.5 Métodos de Análise Estatística de Dados	38
4 ANÁLISE E DISCUSSÃO DOS RESULTADOS	39
4.1 DESENVOLVIMENTO DO APLICATIVO	39
4.2 TESTES INICIAIS	44
4.2.1 Calibração do Sensor de Corrente	44
4.2.2 Calibração do Sensor de Tensão	46
4.3 TESTE DE 24 HORAS EM AMBIENTE REAL	48
4.4 TESTE DE DUAS SEMANAS EM AMBIENTE REAL	49
5 CONCLUSÕES E CONSIDERAÇÕES FINAIS	51
5.1 TRABALHOS FUTUROS	51
5.2 CONSIDERAÇÕES FINAIS	52
REFERÊNCIAS	53
APÊNDICE A – CÓDIGO FONTE DO HARDWARE (ESP32)	56
A.1 MÓDULO PRINCIPAL (ABA MAIN)	56
A.2 MÓDULOS DE TAREFAS (ABAS TASK1 E TASK2)	60
A.3 MÓDULO DE FUNÇÕES (ABA FUNCOES)	63
APÊNDICE B – CÓDIGO-FONTE DO SOFTWARE (APLICATIVO)	68
B.1 DEPENDÊNCIAS DO PROJETO (PACKAGE.JSON)	68
B.2 CONFIGURAÇÃO DO FIREBASE (SRC/FIREBASE.CONFIG.JS)	70
B.3 NAVEGAÇÃO E PONTO DE ENTRADA (APP.JS)	71
B.4 TELA PRINCIPAL E LISTENER DE DADOS (SRC/PAGES/HOME.JS)	74
B.5 TELA DE RELATÓRIOS E GRÁFICOS (SRC/PAGES/RELATORIOS.JS)	78
B.6 TELA DE CONFIGURAÇÕES (SRC/PAGES/CONFIGURACOES.JS)	81
ANEXO A – ESPECIFICAÇÕES TÉCNICAS DO ALICATE AMPERÍMETRO HIKARI HA-3120	84

1 INTRODUÇÃO

Em um mundo cada vez mais consciente da finitude dos recursos naturais e dos impactos ambientais do consumo excessivo, a energia elétrica destaca-se como um pilar fundamental da vida moderna. Ela impulsiona desde as atividades cotidianas mais simples até os avanços tecnológicos mais complexos. No entanto, o acesso à energia elétrica, embora essencial, não deve ser sinônimo de desperdício. A utilização consciente e eficiente desse recurso é crucial não apenas para a saúde financeira das famílias, mas também para a preservação do meio ambiente e a construção de um futuro sustentável para as próximas gerações.

No cenário brasileiro, a questão da eficiência energética assume contornos ainda mais urgentes. Segundo levantamento da Associação Brasileira dos Grandes Consumidores de Energia (ABRACE), o Brasil apresenta o maior custo residencial de energia elétrica entre os 34 países da Organização para a Cooperação e Desenvolvimento Econômico (OCDE), quando relacionado à renda per capita. Isso deve-se, em grande parte, à elevada carga de tributos e subsídios embutidos na fatura, que representam aproximadamente 40 % do valor pago pelos consumidores, comprometendo a competitividade econômica e a eficiência do setor energético nacional (ABRACE, 2023).

A falta de informações claras e em tempo real sobre o consumo de energia elétrica, junto à dificuldade de compreender o sistema tarifário, limita a capacidade dos consumidores de tomar decisões efetivas. Essa realidade se traduz em surpresas desagradáveis nas faturas, desperdício de energia e, conseqüentemente, impactos negativos tanto no bolso dos brasileiros quanto no meio ambiente.

Apesar dos avanços tecnológicos e da crescente conscientização sobre a importância da sustentabilidade, ainda persistem desafios significativos no que diz respeito ao acesso as ferramentas de monitoramento de consumo de energia, tais quais, o alto custo para a obtenção destes equipamentos e a facilidade para utilizá-las.

1.1 JUSTIFICATIVA

A necessidade urgente de conscientização e controle dos gastos energéticos nas residências brasileiras justifica o desenvolvimento de um aplicativo Android para

monitoramento de consumo de energia elétrica residencial. Essa iniciativa se ancora em diversos fatores que evidenciam sua relevância e importância.

O aumento das tarifas de energia elétrica tem gerado impactos econômicos e sociais significativos, afetando especialmente as famílias de baixa renda. Apesar da existência da Tarifa Social de Energia Elétrica (TSEE), destinada a mitigar esse efeito, uma pesquisa realizada pelo Instituto Pólis em parceria com o Inteligência em Pesquisa e Consultoria Estratégica (IPEC) revelou que 57 % dos beneficiários consideram o desconto insuficiente para reduzir o peso da conta de luz em seus orçamentos. O estudo também destacou que, na região Sul, os gastos com energia elétrica comprometem 45 % da renda familiar, superando despesas essenciais como saúde. Esses dados evidenciam que, mesmo com políticas de subsídio, a energia elétrica continua sendo um fator de pressão financeira para as populações mais vulneráveis (PÓLIS, 2024).

O consumo excessivo de energia elétrica acarreta sérias consequências ambientais, contribuindo significativamente para as emissões de gases de efeito estufa. De acordo com a Empresa de Pesquisa Energética, a produção e o uso de energia foram responsáveis por 24 % das emissões brasileiras de gases de efeito estufa. Embora o Brasil tenha alcançado uma participação de 49,1 % de fontes renováveis na sua matriz energética, o desperdício de energia ainda representa um desafio crítico. Sendo assim, a promoção de práticas mais sustentáveis no contexto residencial é uma estratégia válida para mitigar os impactos ambientais negativos associados ao uso ineficiente de energia (EPE, 2023).

1.1.1 A importância da energia elétrica e seu impacto no desenvolvimento socioeconômico

A energia elétrica desempenha um papel fundamental no desenvolvimento da sociedade moderna, sendo amplamente reconhecida como um dos principais indicadores de progresso e qualidade de vida. Goldemberg (1998) destaca que o consumo de energia *per capita* está diretamente relacionado a indicadores sociais como expectativa de vida, taxa de alfabetização e mortalidade infantil, evidenciando a importância da energia elétrica para o desenvolvimento humano. Essa afirmação é corroborada pela análise do mercado de eletricidade, que apresenta taxas de crescimento superiores às do PIB (Produto Interno Bruto), evidenciando a crescente

demanda por energia elétrica em países em desenvolvimento (EPE, 2006).

De acordo com dados do Balanço Energético Nacional (BEN) de 2024, a geração de energia elétrica no Brasil em 2023 atingiu 708,1 TWh, um aumento de 4,6 % em relação ao ano anterior. Esse crescimento demonstra a importância estratégica da energia elétrica para a economia brasileira e a necessidade de garantir o suprimento energético para atender à crescente demanda. (EPE, 2024)

1.1.2 A relação entre consumo de energia, desenvolvimento e renda

O acesso à energia elétrica de uma população, é um marcador crucial do desenvolvimento social e econômico de um país, estando diretamente relacionado à qualidade de vida da nação. Greenpeace (2006) observa que o consumo de energia é superior em países desenvolvidos, impulsionado pela maior posse de aparelhos domésticos e pela demanda industrial. No entanto, a dependência de combustíveis fósseis para a geração de energia ainda é uma realidade global, o que acarreta impactos ambientais e desafios para a transição energética.

No Brasil, o consumo final de energia elétrica em 2023 foi de 616,3 TWh, com destaque para os setores industrial e residencial, que representaram 36,4 % e 27,5 % do consumo total, respectivamente (EPE, 2024). Esses dados expõem a importância do uso eficiente da energia elétrica em ambos os setores, visando a redução de custos e consequentemente, a minimização dos impactos ambientais.

1.1.3 O consumo de energia elétrica no setor residencial e o impacto nas famílias brasileiras

O consumo de energia elétrica no setor residencial varia conforme diversos fatores, como as condições climáticas, os hábitos de consumo e a renda familiar. Um estudo realizado no ano de 2015 pelo Laboratório de Eficiência Energética em Edificações (LABEEE) da Universidade Federal de Santa Catarina (UFSC) revelou que a região Sul apresentou o maior consumo médio residencial entre as regiões brasileiras, com 273 kWh/mês no verão e 261 kWh/mês no inverno. Para efeito de comparação, a média nacional de consumo residencial é de aproximadamente 160 kWh/mês, segundo dados da Empresa de Pesquisa Energética (EPE, 2023). Esses valores indicam que o consumo na região Sul pode ser até 70 % superior à média

nacional, o que pode ser atribuído, entre outros fatores, ao uso intensivo de aquecedores no inverno e de aparelhos de ar-condicionado no verão.

Outro fator relevante é o impacto do custo da energia elétrica no orçamento familiar. Segundo a Associação Brasileira de Grandes Consumidores Industriais de Energia e de Consumidores Livres (ABRACE), entre 2000 e 2022, o custo da conta de luz das famílias brasileiras aumentou 344,7 %. Esse aumento expressivo afeta principalmente as famílias de menor poder aquisitivo, que destinam uma parcela maior de sua renda para o pagamento das contas de luz e gás. A ABRACE aponta que, em 2018, famílias com renda de até R\$ 1.908 por mês gastavam 9,1 % de sua renda com energia elétrica, percentual que podia chegar a 18 % quando considerada a energia embutida no consumo de bens e serviços (ABRACE, 2023). Esses dados reforçam a necessidade de implementar medidas para reduzir o custo da energia e garantir o acesso à energia elétrica para todos.

1.1.4 A importância das soluções digitais no monitoramento de consumo de energia elétrica

O monitoramento digital do consumo residencial tem se mostrado uma ferramenta eficaz para incentivar o uso consciente da energia. De acordo com Darby (2006), o fornecimento de *feedback* direto e contínuo sobre o consumo tende a gerar economia média entre 5 % e 15 % na conta de energia, especialmente quando o usuário tem acesso fácil e compreensível às informações. Essa economia é potencializada quando os dados são apresentados por meio de soluções digitais que permitem visualizações claras, alertas personalizados e comparações históricas.

Além disso, estudos indicam que o uso de aplicativos e interfaces digitais aumenta a conscientização dos usuários sobre seus hábitos de consumo. Murtagh *et al.* (2011) destacam que o *feedback* personalizado, aliado a recomendações práticas, pode promover mudanças comportamentais sustentáveis e duradouras. Isso é particularmente relevante em contextos residenciais, nos quais o consumo é fortemente influenciado por rotinas diárias e decisões individuais.

O uso de soluções digitais também contribui para o empoderamento do consumidor, tornando-o um agente ativo na gestão de sua própria demanda energética. Segundo Froehlich *et al.* (2010), sistemas que oferecem dados detalhados e em tempo real sobre o consumo tendem a aumentar o engajamento e a motivação

dos usuários para reduzir o desperdício.

Portanto, a adoção de tecnologias digitais no monitoramento do consumo elétrico não apenas viabiliza uma gestão mais eficiente da energia, como também se apresenta como uma estratégia educativa e de conscientização, capaz de alinhar interesses econômicos e ambientais em um mesmo propósito.

1.1.5 Tarifas de energia elétrica e bandeiras tarifárias

O órgão responsável por regulamentar e fiscalizar o setor elétrico no Brasil é a Agência Nacional de Energia Elétrica (ANEEL), ela define diretrizes gerais de fornecimento e os critérios para o cálculo de tarifas aplicadas pelas distribuidoras. Em Santa Catarina, a principal distribuidora é a Centrais Elétricas de Santa Catarina S.A (CELESC), responsável pelo atendimento da maior parte dos municípios do estado.

As tarifas são compostas por diversos elementos, como os custos de geração, transmissão e distribuição da energia, encargos setoriais e tributos, como PIS, COFINS e ICMS. As bandeiras tarifárias têm relação com os custos de geração e indicam as condições de geração de energia. A bandeira verde indica condições favoráveis e não há acréscimo na tarifa. A bandeira amarela é acionada em condições menos favoráveis, incidindo um acréscimo de R\$ 1,885 para cada 100 kWh consumidos. Já a bandeira vermelha, para condições críticas, possui dois patamares: o Patamar 1 adiciona R\$ 4,463 e o Patamar 2 adiciona R\$ 7,877, ambos a cada 100 kWh consumidos (CELESC, 2025).

O calendário de acionamento das bandeiras tarifárias é atualizado mensalmente pela ANEEL, e seu acompanhamento pode auxiliar os consumidores a planejarem melhor o uso de energia (ANEEL, 2024).

1.1.6 Consumo de energia elétrica e unidade de medida

O consumo de energia elétrica é medido em quilowatt-hora (kWh), unidade que representa a quantidade de energia utilizada por um aparelho de 1.000 W durante uma hora. Esse valor é a base para o cálculo da fatura de energia e está diretamente relacionado aos hábitos de consumo dos usuários.

Para se calcular o valor do kWh de uma fatura de energia elétrica, a CELESC adota a equação 1:

$$\text{Tarifa final (R\$)} = \frac{\text{Tarifa homologada}}{\left(\frac{100 - \text{PIS} - \text{COFINS}}{100}\right)} \div \left(\frac{100 - \text{ICMS}}{100}\right) \quad (1)$$

Onde:

- Tarifa homologada é o valor do kWh aprovado pela ANEEL, sem impostos;
- PIS (Programa de Integração Cultural) e COFINS (Contribuição para o Financiamento da Seguridade Social) são tributos federais;
- ICMS (Imposto sobre Circulação de Mercadorias e Serviços) é um tributo estadual;
- Tarifa final é o custo do kWh da energia elétrica com impostos.

As alíquotas desses tributos são atualizadas e divulgadas mensalmente pela própria concessionária de energia elétrica em seu portal oficial (CELESC, 2025).

1.2 OBJETIVOS

Diante do cenário apresentado, este Trabalho de Conclusão de Curso propõe o desenvolvimento de um sistema técnico para endereçar a lacuna de monitoramento residencial. Os objetivos do projeto são detalhados a seguir.

1.2.1 Objetivo geral

Desenvolver um sistema de monitoramento de energia elétrica residencial, composto por um medidor físico e um aplicativo *Android*, para o cálculo de gastos com eletricidade no estado de Santa Catarina, e avaliar sua efetividade por meio de testes técnicos.

1.2.2 Objetivos específicos

- Desenvolver um circuito eletrônico para medir e registrar o consumo de energia elétrica, com atualização dos dados a cada 5 segundos.
- Programar o microcontrolador para processar os dados coletados pelo circuito e calcular o consumo de energia em quilowatt-hora.
- Aferir e calibrar o hardware de medição, comparando as leituras do protótipo com as de um instrumento de medição padrão para garantir sua acurácia.

- Desenvolver um aplicativo *Android* para análise dos dados coletados, que permita ao usuário visualizar seu consumo de energia elétrica e estimar os custos com base nas tarifas da CELESC.
- Validar o desempenho do sistema integrado em um ambiente residencial, confrontando os dados de consumo registrados pelo sistema de monitoramento com os do medidor da concessionária

1.3 ORGANIZAÇÃO DO TRABALHO

O Capítulo 2 – Referencial Teórico apresenta uma revisão das tecnologias e soluções existentes para o monitoramento de consumo de energia elétrica, analisando trabalhos correlatos e outras plataformas disponíveis no mercado. O Capítulo 3 – Metodologia detalha os materiais e métodos empregados no desenvolvimento do *EnergyCheck*, abrangendo tanto o *hardware* (microcontrolador ESP32, sensores de corrente e tensão) quanto o *software* (prototipação com *Figma*, banco de dados e autenticação via *Firebase*, e desenvolvimento do aplicativo *Android* utilizando *React Native* com *Expo*). O Capítulo 4 – Análise e Discussão dos Resultados apresenta os resultados dos testes de calibração dos sensores de corrente e tensão, bem como os dados obtidos em dois testes em ambiente real: um com duração de 24 horas e outro com duração de duas semanas, ambos comparando as medições do *EnergyCheck* com o medidor da CELESC. O capítulo inclui ainda uma discussão sobre a precisão do sistema e a efetividade do monitoramento. Por fim, o Capítulo 5 – Conclusões e Considerações Finais retoma os objetivos do trabalho, relacionando-os com os resultados obtidos, e discute as contribuições do *EnergyCheck* para o monitoramento e controle do consumo de energia elétrica residencial. O capítulo também apresenta sugestões para trabalhos futuros e considerações finais sobre o projeto.

2 REFERENCIAL TEÓRICO

Estudos com foco no monitoramento do consumo de energia elétrica tem se popularizado devido à necessidade de eficiência energética e à massificação de tecnologias, como microcontroladores e aplicativos móveis.

2.1 SISTEMAS MICROCONTROLADOS PARA MONITORAMENTO ENERGÉTICO

Schuler (2020) propôs a utilização de um sistema de monitoramento de consumo de energia elétrica para circuitos monofásicos de baixa tensão. O autor buscou interpretar as nuances do comportamento da energia elétrica e aplicar os conhecimentos com sensores e microcontroladores. Com componentes semelhantes aos adotados no presente trabalho, utilizou um sensor de corrente SCT 013, um microcontrolador ESP32 e um sensor de tensão ZMPT101B. Schuler teve como objetivo mensurar o consumo energético de um quiosque para a validação da eficiência do sistema. Os dados coletados foram enviados para uma página *web*, em que era possível visualizar grandezas como tensão, potência média e o consumo acumulado em kWh.

2.2 APLICATIVO *ANDROID* PARA MONITORAMENTO DO CONSUMO DE ENERGIA ELÉTRICA DE UMA RESIDÊNCIA

O trabalho teve como objetivo monitorar o consumo de energia elétrica de aparelhos eletrônicos de uma residência através de uma tomada, extraindo valores como tensão e corrente e podendo alterar o estado da tomada ON/OFF. Segundo Pires (2021), os valores das medições foram armazenados para calcular consumo de energia (em kWh) e, a partir deste, estimar o custo monetário (em R\$), foram apresentados em formato de gráfico e tabela via aplicativo *Android*, para que o consumidor pudesse visualizar os gastos e supervisionar o consumo de energia, visando a eficiência energética.

Para isso, foi utilizado um microcontrolador ESP32, responsável pela coleta e processamento de dados, um sensor de corrente ACS712 e um sensor de tensão ZMPT101B. Para o controle da tomada, foi utilizado um módulo relé, que conectado ao microcontrolador atua na passagem de corrente, alterando o estado

ligado/desligado. O ESP32, envia os dados para o *banco de dados* e o aplicativo criado via linguagem de programação *Kotlin*, faz a leitura desses dados, exibe ao usuário o consumo total em kWh e o valor a ser pago em reais.

O resultado obtido por Pires, foi o desenvolvimento de um aplicativo *Android* capaz de realizar medições em uma tomada e controlá-la, apesar da medição apresentar alguns problemas de *delay*, o que acabou por impactar negativamente nos valores exatos de consumo.

2.3 MONITORAMENTO DO CONSUMO DE ENERGIA ELÉTRICA E CONTROLE DE EQUIPAMENTOS VIA APLICATIVO

Pereira (2018) propôs-se a desenvolver um sistema acessível, conectado a algum equipamento alimentado por eletricidade. Tal sistema, seria capaz de fornecer em tempo real, por meio de uma aplicação *online*, a potência elétrica consumida e o seu custo monetário, permitindo o usuário o comando do acionamento (liga/desliga) do aparelho através de uma conexão wireless.

O microcontrolador utilizado foi o *NodeMCU*, que tinha como objetivo obter os dados coletados através do sistema de monitoramento. Para o sensor de corrente, utilizou o ACS712. Para se obter a tensão, foi utilizado um valor fixo baseado na tensão nominal da cidade de Ouro Preto. Já para o acionamento, foi utilizado um módulo relé. Para a conexão entre o sistema e o controle de equipamentos via aplicativo, foi utilizado a plataforma *CloudMQTT*, plataforma intuitiva e que possui um banco de dados e um aplicativo, apenas necessitando configurar o *MQTT Dash* e o *CloudMQTT*.

Os resultados encontrados por Pereira (2018), foram um protótipo capaz de chavear os estados *ON/OFF* através de um dispositivo *Android* e a capacidade de calcular o gasto energético em reais, utilizando a potência consumida pelo aparelho multiplicado pelo preço do kWh vigente na data do trabalho.

2.4 ANÁLISE DOS TRABALHOS RELACIONADOS

Os trabalhos relacionados atingiram os objetivos propostos, desenvolvendo protótipos capazes de mensurar grandezas elétricas como corrente e tensão elétrica. Os dados eram processados por um microcontrolador e armazenados em nuvem, os valores tratados de potência consumida e/ou preço do kWh eram apresentados aos

usuários através de plataformas web ou aplicativos de *smartphone*, de forma que o usuário tivesse autonomia para monitorar o gasto de energia elétrica em determinados períodos.

Contudo, apesar de os trabalhos atingirem seus objetivos, eles demonstraram erros de medições quando comparados com os valores reais medidos pela concessionária de energia, e a divergência se dá por causa exclusiva ou união de possíveis erros como, a precisão dos sensores, a não utilização de sensores de tensão, ruídos no sinal e as leituras estarem no mesmo *loop* da parte responsável pelo envio de dados, gerando um atraso na leitura e conseqüentemente erro na medição.

3 METODOLOGIA

A seguir, são descritos os materiais e métodos empregados no desenvolvimento do sistema, divididos em duas etapas principais: *hardware* e *software*.

3.1 HARDWARE

O desenvolvimento do *hardware* do *EnergyCheck* se baseou na utilização de um microcontrolador ESP32 (Figura 1), programado utilizando o Arduino IDE, com o objetivo de capturar e processar os dados de consumo de energia elétrica. O ESP32 foi escolhido por sua versatilidade, baixo custo e conectividade *Wi-Fi* integrada, permitindo a comunicação com o banco de dados e a interface do usuário.

Figura 1 - Microcontrolador ESP32



Fonte: ROBOCORE (2024).

O sensor de corrente utilizado é o modelo SCT-013-000 (Figura 2) um transformador de corrente do tipo toroidal, que opera com base no princípio da indução eletromagnética, segundo o qual uma corrente elétrica alternada ao circular por um condutor gera um campo magnético variável. Este, por sua vez, pode induzir uma corrente elétrica em um enrolamento secundário de um transformador, proporcional à corrente original (BOYLESTAD; NASHELSKY, 2014).

Figura 2 - Sensor de corrente SCT- 013-000



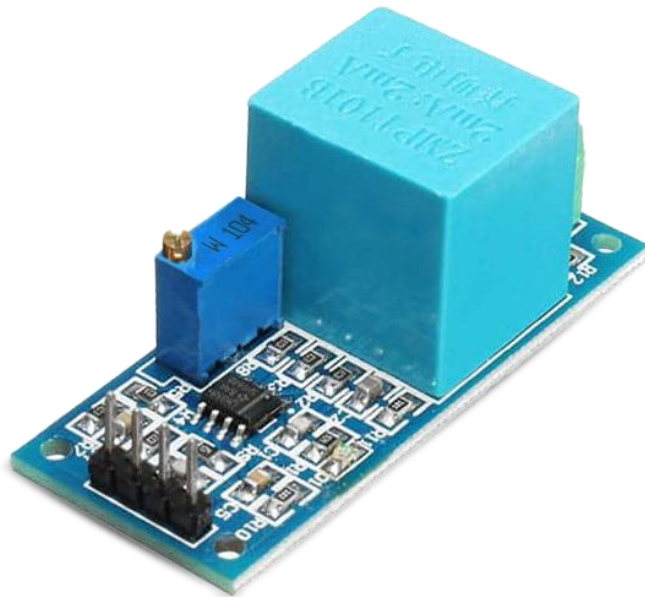
Fonte: ROBOCORE (2024).

Para medir o consumo de energia em tempo real, foi utilizado um sensor de corrente não invasivo, que permite a medição da corrente elétrica sem a necessidade de interromper o circuito. Este sensor, por sua vez, é acoplado ao cabo de fornecimento principal da residência, transformando 100 A em 50 mA.

Essa corrente secundária pode ser convertida em tensão por meio de um resistor de carga (*shunt*), possibilitando sua leitura por um microcontrolador — neste caso, o ESP32 — para posterior processamento, exibição e análise do consumo elétrico no sistema *EnergyCheck*.

Para medir a tensão da rede elétrica em tempo real, foi utilizado o sensor de tensão ZMPT101B (Figura 3). Este módulo permite a medição da tensão AC, oferecendo segurança e isolamento. O ZMPT101B é capaz de medir tensões de 0 a 250 V, tornando-o adequado para aplicações residenciais.

Figura 3 - Sensor de tensão ZMPT101B



Fonte: ROBOCORE (2024).

O funcionamento do ZMPT101B é baseado em um divisor de tensão capacitivo. A tensão da rede elétrica é aplicada a um divisor de tensão formado por capacitores, reduzindo-a a um nível seguro para o circuito interno do sensor. Um conversor analógico-digital (ADC) integrado ao módulo converte essa tensão reduzida em um sinal digital proporcional. Esse sinal digital é então transmitido através da interface de comunicação do sensor, que pode ser utilizada por um microcontrolador para leitura e processamento.

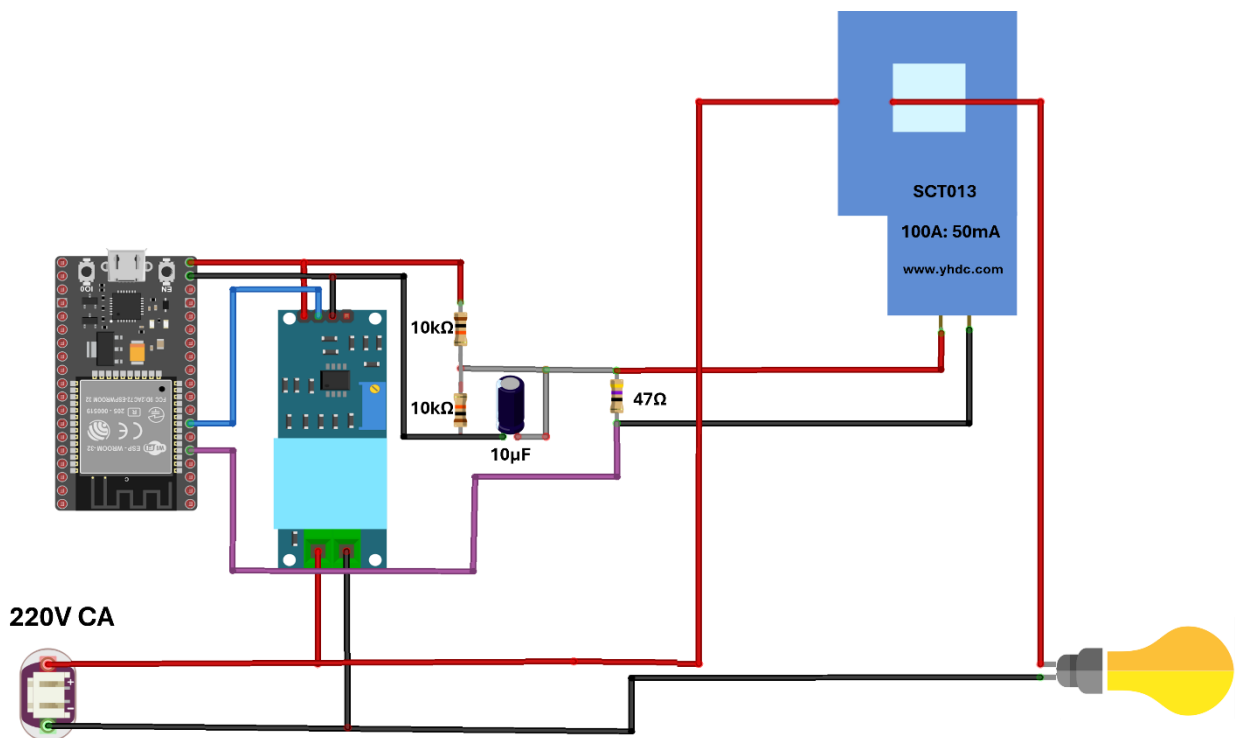
A conexão do ZMPT101B à rede elétrica é realizada através de dois terminais de entrada, indicados no terminal como "L" e "GND". O terminal "L" deve ser conectado ao cabo fase da rede elétrica, enquanto o terminal "GND" deve ser conectado ao cabo neutro.

O módulo possui pinos de saída para a comunicação com o microcontrolador: o pino VCC é conectado à alimentação de 5 V do ESP32; os pinos GND são conectados ao terra (*ground*) do microcontrolador; e o pino OUT, que fornece o sinal de saída analógico, é conectado a uma entrada analógica do ESP32, permitindo a leitura do valor da tensão medida.

A montagem do circuito final do projeto, cujo esquemático é apresentado na Figura 4, envolve a configuração do sensor de corrente e do circuito de tensão.

Para o sensor de corrente, é utilizado um divisor de tensão composto por dois resistores de $10\text{ k}\Omega$ ligados em serie. Essa configuração permite que o ESP32 leia as variações de tensão de forma adequada, transformando-as de negativas para positivas. Além disso, é adicionado um capacitor de $10\text{ }\mu\text{F}$ para filtrar o sinal recebido e reduzir interferências. O sensor de corrente é conectado a uma resistência de $47\text{ }\Omega$, isso porque o sensor de corrente transforma uma corrente de até 100 A em 50 mA , com isso, essa corrente ao percorrer a resistência de $47\text{ }\Omega$, gera uma tensão na qual o ESP32 é capaz de ler. É fundamental que o sensor seja instalado abraçando apenas o condutor de fase do circuito a ser medido. Isso é necessário pois o sensor opera como um Transformador de Corrente (TC), baseado na indução eletromagnética. Se o sensor abraçar o cabo completo (com fase e neutro), o campo magnético da corrente de ida anula o da corrente de volta, resultando em uma medição nula.

Figura 4 - Esquemático do projeto



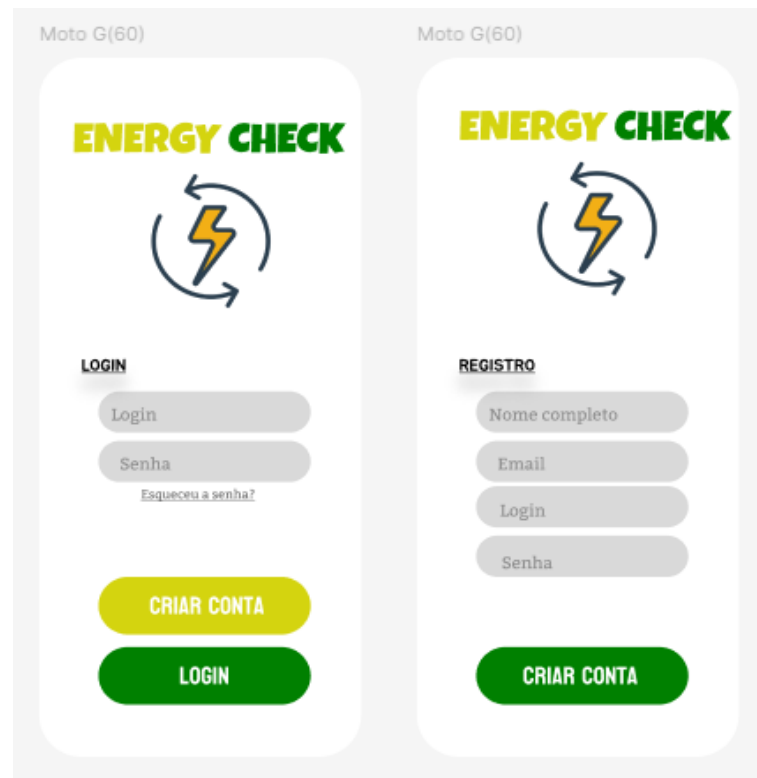
Fonte: elaborado pelo próprio autor.

3.2 SOFTWARE

3.2.1 Figma

Para se ter um esboço prévio de como ficaria o *layout* do aplicativo, foi utilizado do *software Figma* para projetar as telas, sendo elas inicialmente de "login" e "cadastro", conforme Figura 5.

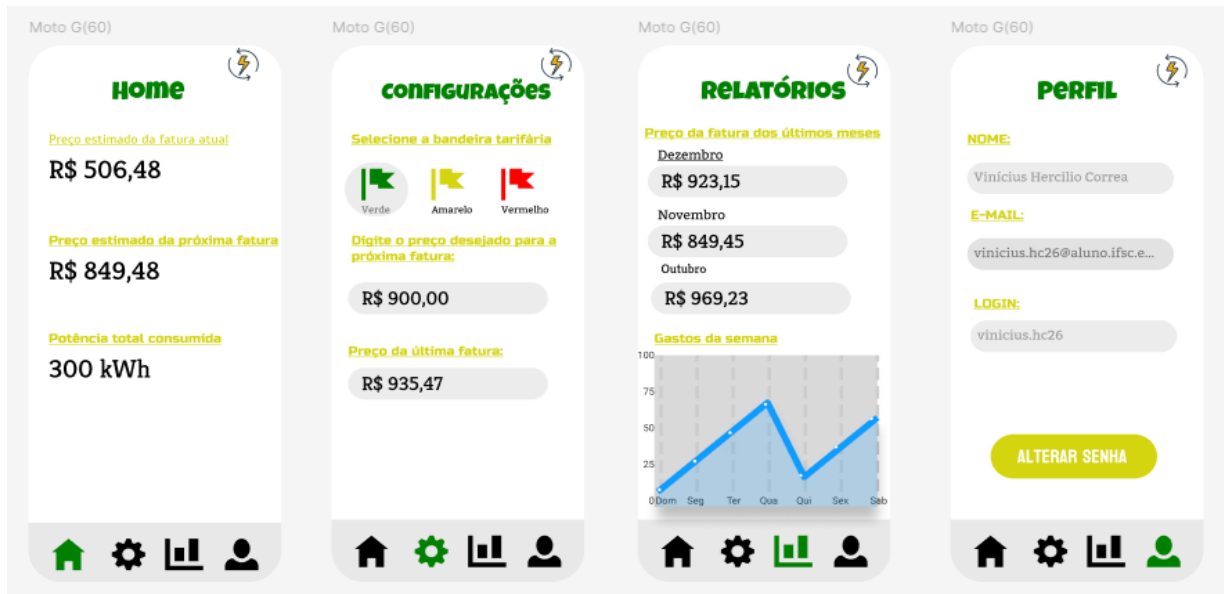
Figura 5 - Projeto de telas *login* e registro



Fonte: elaborado pelo próprio autor.

Após a criação das duas primeiras telas, as telas "Home", "Configurações", "Relatórios" e "Perfil", conforme apresentado na Figura 6 foram criadas, completando o esboço inicial do *layout*.

Figura 6 - Projeto de telas internas do aplicativo



Fonte: elaborado pelo próprio autor.

Com as telas no *Figma* prontas, consegue-se visualizar as principais ideias de funcionalidade até o momento, na tela *Home*, dados referentes a fatura atual, sejam eles o valor gasto em energia elétrica até o momento, a estimativa de gasto até o final do mês e a quantidade de potência consumida.

Já na tela *Configurações*, dados referentes à bandeira tarifária vigente e ao preço que se deseja pagar na fatura atual, para que quando esteja chegando próximo a esse valor (90 %), seja enviado uma notificação para o usuário, permitindo assim, racionalizar se assim fizer necessário.

Na aba *relatórios*, consegue-se visualizar as informações de faturas anteriores e um gráfico de gastos da semana.

E na parte referente ao perfil, encontra-se o nome, e-mail e *login* cadastrados no aplicativo.

Após a definição do *layout* do aplicativo no *Figma*, que serviu como protótipo visual, procedeu-se com o desenvolvimento do *EnergyCheck*. A interface foi codificada manualmente em *JavaScript*, traduzindo o design do *Figma* em componentes do *framework React Native*. A plataforma *Expo* foi adotada para gerenciar o ambiente de desenvolvimento e simplificar o processo de compilação do aplicativo. A escolha do *React Native* se deve à sua capacidade de construir interfaces de usuário com aparência nativa e bom desempenho tanto para *Android* quanto para *iOS*, a partir de uma única base de código *JavaScript*, o que aumenta a eficiência do

desenvolvimento e facilita a manutenção do aplicativo.

A plataforma *Expo*, por sua vez, foi escolhida para agilizar o processo de desenvolvimento, oferecendo ferramentas que simplificam a configuração do ambiente, a visualização do aplicativo em tempo real e a geração dos arquivos de instalação.

3.2.2 *Firestore realtime database e authentication*

O *Firestore Realtime Database* foi a escolha para o banco de dados do *EnergyCheck*. Ele oferece um banco de dados *NoSQL* hospedado na nuvem, com sincronização de dados em tempo real, o que é crucial para o funcionamento do aplicativo. A escolha por essa tecnologia se justifica pelas seguintes vantagens:

Sincronização em tempo real: Permite que o aplicativo receba atualizações instantâneas dos dados enviados pelo ESP32, garantindo que o usuário visualize as informações de consumo atualizadas.

Facilidade de integração: O *Firestore* oferece bibliotecas para *React Native* que facilitam a integração do aplicativo com o banco de dados.

Com o *Firestore Realtime Database* atuando como intermediário entre o ESP32 e o aplicativo, a comunicação entre o *hardware* e o *software* se torna eficiente e confiável. O ESP32 envia os dados de consumo para o banco de dados, e o aplicativo, por sua vez, monitora o *Firestore* e atualiza a interface do usuário com as novas informações em tempo real.

A estrutura do banco de dados foi organizada para otimizar o armazenamento e o acesso aos dados do sistema e dos usuários. O *Firestore Realtime Database* é composto por três nós principais, conforme ilustrado na Figura 7: *authCode*, *configuracoes* e *users*.

O nó *authCode* é responsável por armazenar códigos de autenticação. No processo de cadastro, o aplicativo verifica a validade do código inserido pelo usuário através deste nó, garantindo que o cadastro seja permitido apenas a usuários que possuam um código válido.

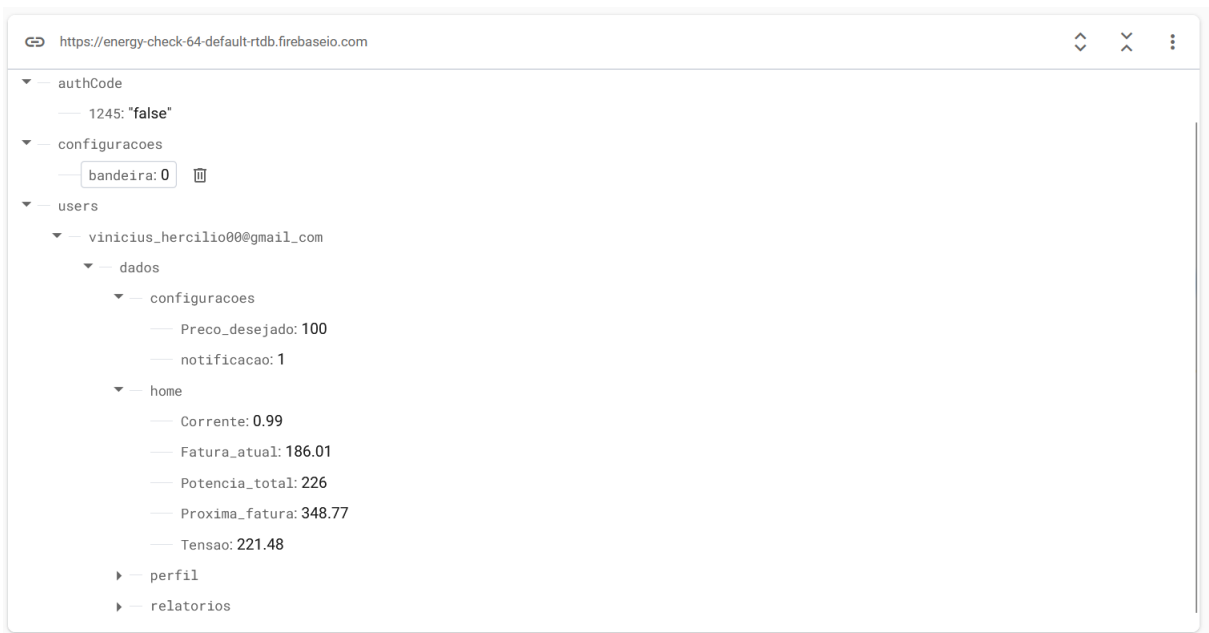
O nó *configuracoes* centraliza as configurações globais do aplicativo, como o status da bandeira tarifária atual. Os valores (0, 1, 2, 3) correspondem, respectivamente, às bandeiras Verde, Amarela, Vermelha patamar 1 e Vermelha patamar 2, permitindo que o aplicativo adapte os cálculos de custo em tempo real.

O nó `users` armazena os dados específicos de cada usuário, sendo cada entrada indexada pelo e-mail do usuário (com caracteres especiais substituídos para compatibilidade com o Firebase). Dentro de cada perfil de usuário, são organizados quatro sub-nós principais:

- **configurações:** Contém dados de configuração personalizados, como o `Preco_desejado` para alertas e o `status de notificacao` (1 para ativada, 0 para desativada).
- **home:** Armazena as informações de consumo em tempo real para exibição na tela inicial do aplicativo, incluindo `Corrente`, `Fatura_atual`, `Potencia_total`, `Proxima_fatura` e `Tensao`.
- **perfil:** Guarda os dados pessoais do usuário, como *e-mail*, nome e o *token* de cadastro.
- **relatorios:** Organiza os dados históricos de consumo para a tela relatórios, como o nó `Diario`, que detalha o consumo por hora (`d0h`, `d12h`, `d15h`, etc.) em um determinado dia.

Essa arquitetura permite uma gestão flexível e escalável dos dados, fundamental para o monitoramento contínuo e personalizado do consumo elétrico.

Figura 7 – Estrutura do *Firebase Realtime Database* do *Energy Check*



Fonte: elaborado pelo próprio autor.

Além do banco de dados em tempo real, foi utilizado o *Firebase Authentication*

para gerenciar a autenticação dos usuários do *EnergyCheck*. Com o *Firebase Authentication*, os usuários podem se cadastrar e autenticar no aplicativo usando e-mail e senha, ficando armazenadas nos servidores Google, isso garante a segurança das informações dos usuários e permite que cada usuário acesse apenas seus próprios dados de consumo.

3.2.3 Desenvolvimento do aplicativo

A plataforma *Expo*, por sua vez, complementou o *React Native*, agilizando o processo de desenvolvimento com suas ferramentas e serviços. O *Expo CLI* simplificou a configuração do ambiente de desenvolvimento e o gerenciamento de dependências, enquanto o *Expo Go* possibilitou a visualização e os testes do aplicativo em tempo real, sem a necessidade de compilação, ou seja, qualquer alteração feita durante o desenvolvimento do aplicativo, podia ser visualizada imediatamente por meio do *Android Studio*, simulador na qual foi utilizado para fazer a simulação de um celular *Android*.

A estrutura do aplicativo foi cuidadosamente planejada, com a organização do código em componentes reutilizáveis, o que facilita a manutenção e a evolução do projeto. As principais funcionalidades, como o monitoramento do consumo em tempo real, a estimativa da fatura, os gráficos de consumo e o histórico de faturas, foram implementadas utilizando os componentes do *React Native*, integradas às bibliotecas do *Firebase* para garantir a sincronização dos dados e a persistência das informações. A interface do usuário foi projetada para ser intuitiva e de fácil navegação, com o objetivo de proporcionar uma experiência agradável e eficiente aos usuários do *EnergyCheck*.

3.3 PROCEDIMENTO DOS TESTES

Para avaliar a eficácia técnica e a precisão do sistema desenvolvido, foram realizados quatro procedimentos experimentais distintos: a validação de bancada dos sensores de corrente e tensão, e a validação do sistema integrado em ambiente real em testes de curta e longa duração. A metodologia para cada procedimento é descrita a seguir.

3.3.1 Validação do Sensor de Corrente (SCT-013-000)

O procedimento descrito a seguir não é uma calibração primária, mas sim uma validação experimental da constante de calibração de corrente adotada no *software*. O objetivo foi assegurar que o sistema fornecesse leituras precisas sob diferentes condições de carga, comparando-as com as de um instrumento de referência.

Para este teste, utilizou-se como instrumento de referência o alicate amperímetro digital True RMS Hikari HA-3120 (Figura 8). A escolha deste modelo foi fundamentada em suas especificações técnicas (vide ANEXO A), que garantem uma base de comparação confiável: precisão de $\pm(2,0\%$ da leitura + 5 dígitos) na faixa de 20 A a 200 A, com resolução de 10 mA (HIKARI, [s.d.]). Como carga variável, foi empregado um secador de cabelo, cujos ajustes de temperatura e velocidade permitiram simular seis diferentes níveis de consumo de corrente.

Figura 8 - Alicate amperímetro HA-3120



Fonte: HIKARI (2024).

A bancada experimental (Figura 9) foi montada para permitir a medição simultânea da corrente pelo protótipo e pelo instrumento de referência. O procedimento de teste foi o seguinte:

1. O sensor SCT-013-000 foi instalado abraçando apenas um dos condutores de alimentação da carga (secador) e conectado ao circuito de medição do ESP32.
2. O alicate amperímetro de referência (Hikari HA-3120) foi posicionado no mesmo condutor.
3. A carga foi acionada em seus diferentes modos (variando temperatura e velocidade).
4. Para cada modo, foram registradas simultaneamente a leitura do instrumento de referência e a leitura gerada pelo sistema de monitoramento, esta última exibida via monitor serial.

Figura 9 - Bancada de testes



Fonte: elaborado pelo próprio autor.

3.3.2 Validação do Sensor de Tensão (ZMPT101B)

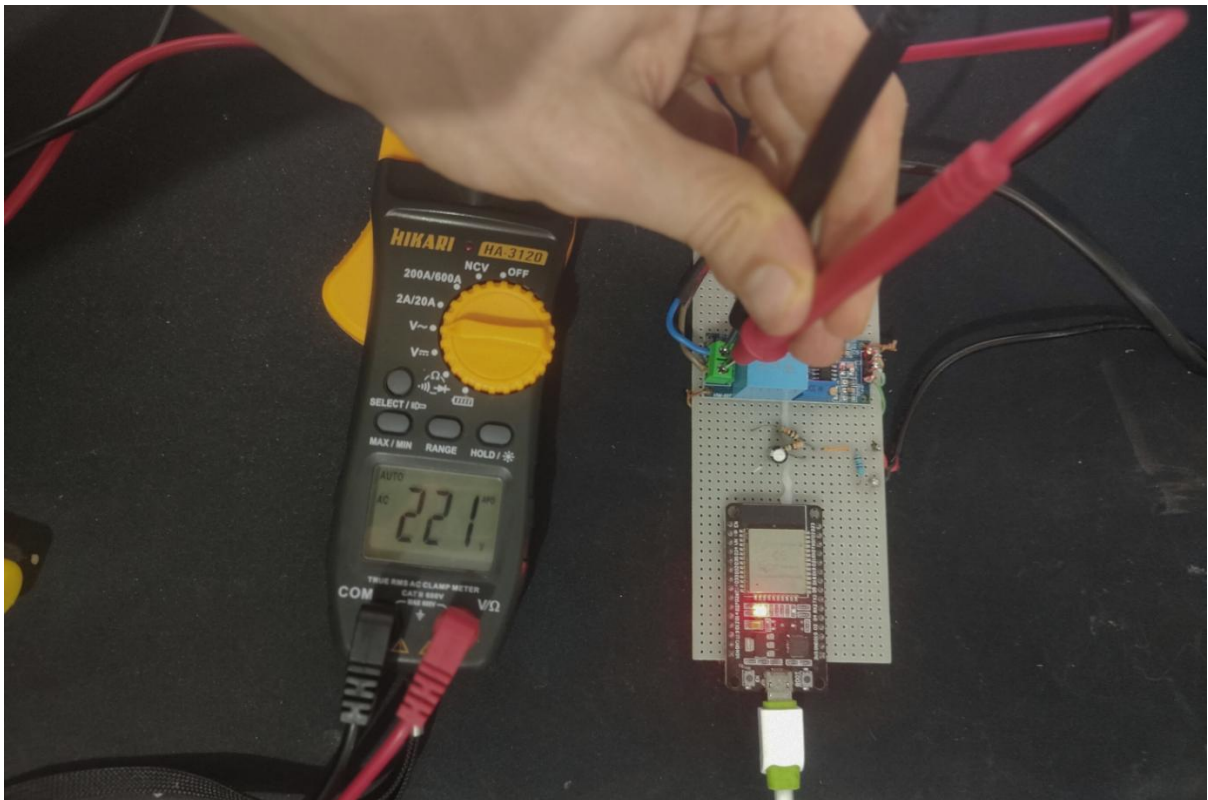
Assim como na medição de corrente, foi realizado um procedimento de validação experimental da constante de calibração adotada para o sensor de tensão ZMPT101B, garantindo que o sistema reporte valores de tensão com acurácia.

Para a validação, as leituras do protótipo foram comparadas com o mesmo instrumento de referência, o alicate amperímetro True RMS Hikari HA-3120 (cuja

figura já foi apresentada). Conforme seu manual técnico (HIKARI, [s.d.]), na escala de Tensão AC de 600 V, o instrumento possui uma precisão de $\pm(1,2\% \text{ da leitura} + 5 \text{ dígitos})$, especificação que serviu como base para a comparação.

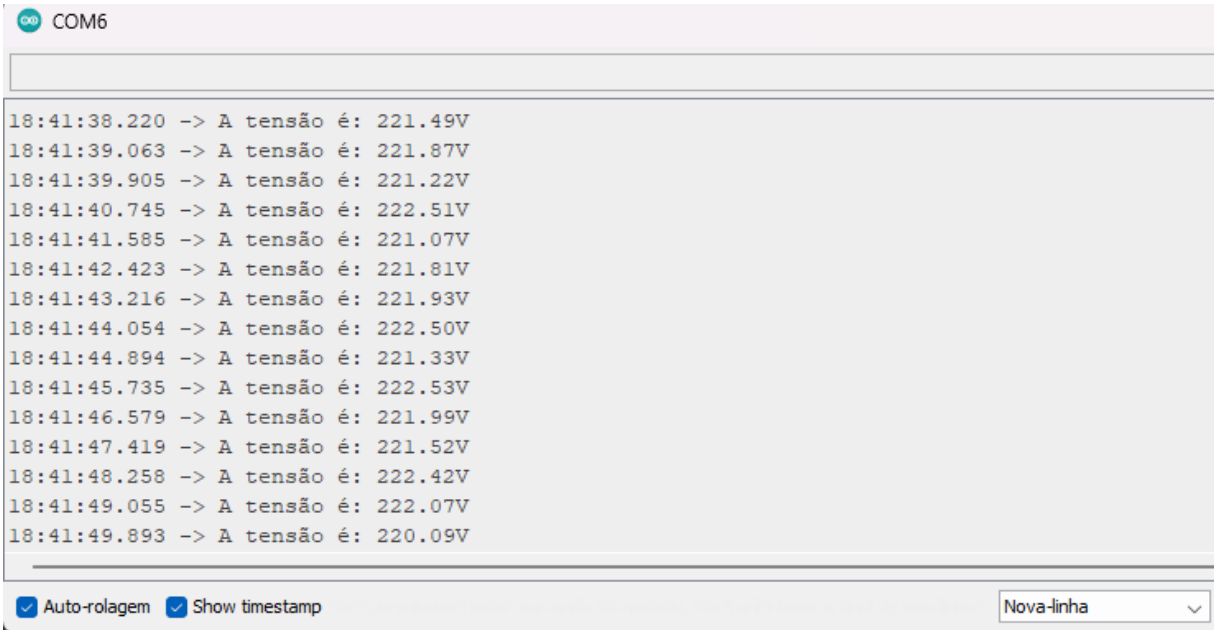
A montagem experimental (Figura 10) consistiu em conectar o sensor ZMPT101B e o alicate amperímetro de referência em paralelo com a rede elétrica de 220 VCA. As leituras do sistema EnergyCheck foram monitoradas via comunicação serial (Figura 11). O procedimento de coleta consistiu em realizar dez medições em intervalos de 5 minutos para registrar as flutuações naturais da rede, anotando-se simultaneamente os valores do instrumento de referência e do protótipo.

Figura 10 - Teste do sensor de tensão



Fonte: elaborado pelo próprio autor.

Figura 11 - Teste do sensor de tensão no monitor serial do Arduino IDE



```
COM6
18:41:38.220 -> A tensão é: 221.49V
18:41:39.063 -> A tensão é: 221.87V
18:41:39.905 -> A tensão é: 221.22V
18:41:40.745 -> A tensão é: 222.51V
18:41:41.585 -> A tensão é: 221.07V
18:41:42.423 -> A tensão é: 221.81V
18:41:43.216 -> A tensão é: 221.93V
18:41:44.054 -> A tensão é: 222.50V
18:41:44.894 -> A tensão é: 221.33V
18:41:45.735 -> A tensão é: 222.53V
18:41:46.579 -> A tensão é: 221.99V
18:41:47.419 -> A tensão é: 221.52V
18:41:48.258 -> A tensão é: 222.42V
18:41:49.055 -> A tensão é: 222.07V
18:41:49.893 -> A tensão é: 220.09V
 Auto-rolagem  Show timestamp Nova-linha
```

Fonte: elaborado pelo próprio autor.

3.3.3 Teste de Curta Duração (24 Horas) em Ambiente Real

Para avaliar o desempenho do sistema integrado em um ambiente real, foi realizado um teste de 24 horas. O método consistiu na comparação direta entre o consumo total de energia (kWh) medido pelo EnergyCheck e o consumo registrado pelo medidor da concessionária (CELESC, código SAP 27184), que foi utilizado como padrão de referência. O equipamento da CELESC é um medidor eletrônico monofásico Classe B, com precisão de 1,0% (ou melhor), qualificando-o como uma base confiável para a validação (CELESC, 2021).

O teste foi realizado em uma residência, com perfil de consumo típico, incluindo equipamentos como chuveiro elétrico, geladeira e máquina de lavar.

O sistema EnergyCheck foi instalado no circuito principal da residência, próximo ao quadro de distribuição, conforme ilustrado na Figura 12.

Figura 12 - Sistema *EnergyCheck* instalado na residência



Fonte: elaborado pelo próprio autor.

O procedimento de coleta foi o seguinte:

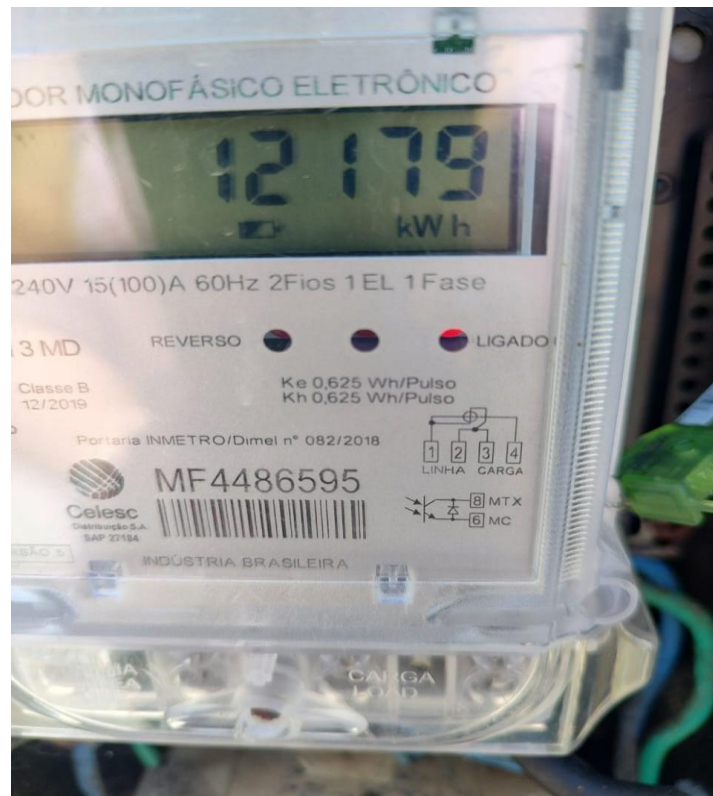
1. No dia 02/02/2025, às 12:00 h, foi anotado o valor inicial de consumo (em kWh) exibido no medidor da CELESC (Figura 13).
2. O sistema *EnergyCheck* foi iniciado para coletar dados de consumo continuamente durante 24 horas.
3. Após 24 horas (dia 03/02/2025, às 12:00 h), foi anotado o novo valor do consumo total (kWh) registrado no medidor da CELESC (Figura 14).
4. O consumo total registrado pelo *EnergyCheck* foi então comparado com o consumo total medido pela concessionária (diferença entre a leitura final e inicial do medidor da CELESC).

Figura 13 - Medidor de energia no início do teste



Fonte: elaborado pelo próprio autor.

Figura 14 - Medidor de energia após 24 horas



Fonte: elaborado pelo próprio autor.

3.3.4 Teste de Longa Duração (Duas Semanas) em Ambiente Real,

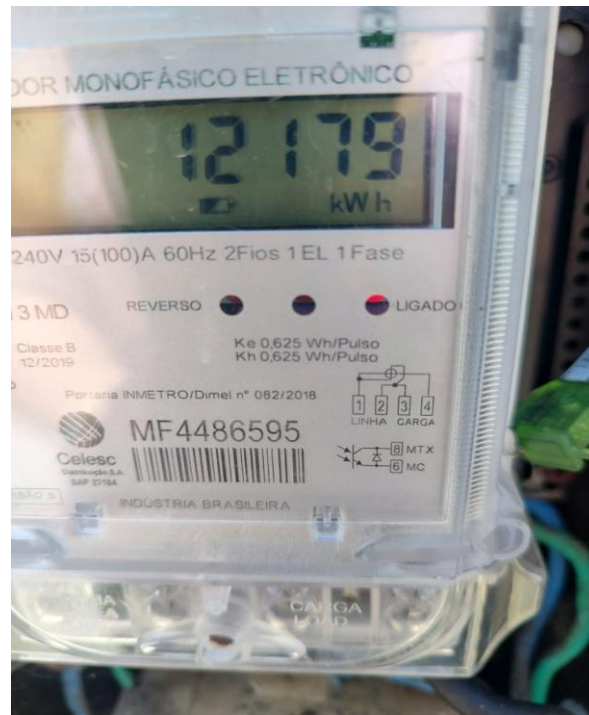
Para complementar o teste de curta duração, foi conduzida uma avaliação prolongada do desempenho do sistema com duração de duas semanas. O objetivo foi verificar a confiabilidade e estabilidade do sistema ao longo do tempo, mantendo a comparação com os registros do medidor da concessionária CELESC como padrão de referência.

O teste foi realizado na mesma residência do experimento anterior, mantendo-se as condições de instalação e perfil de consumo (descritos na seção 3.3.3).

O procedimento adotado foi o seguinte:

1. Antes do início do teste, todos os contadores de consumo do sistema EnergyCheck foram zerados para garantir uma medição isolada para o novo período.
2. No início do teste (03/02/2025, às 12h:10min), foi anotado o valor inicial de consumo (em kWh) exibido no medidor da CELESC (Figura 15).
3. O sistema EnergyCheck permaneceu instalado e em operação contínua, armazenando os dados de consumo no banco de dados durante todo o período.
4. Ao término das duas semanas (17/02/2025), foi registrado o valor final do medidor da CELESC (Figura 16).
5. O consumo total registrado pelo EnergyCheck foi então comparado com o consumo total medido pela concessionária (diferença entre a leitura final e inicial do medidor CELESC).

Figura 15 - Medidor de energia antes do teste de duas semanas



Fonte: elaborado pelo próprio autor.

Figura 16 - Medidor de energia após o teste de duas semanas



Fonte: elaborado pelo próprio autor.

3.3.5 Métodos de Análise Estatística de Dados

Para a análise quantitativa e validação dos dados coletados nos testes, dois cálculos estatísticos principais foram empregados.

Primeiramente, para avaliar a exatidão do protótipo, ou seja, a proximidade da medição do sistema em relação ao valor de referência, foi calculado o erro percentual relativo para cada ponto. A Equação 2 demonstra o cálculo utilizado.

$$\text{Erro(\%)} = \frac{|\text{Leitura do Sensor} - \text{Leitura do Alicate}|}{\text{Leitura do Alicate}} \times 100 \quad (2)$$

Em segundo lugar, para quantificar a precisão do sistema, que representa a repetibilidade ou a dispersão das medições, foi calculado o desvio padrão amostral. Esta métrica foi aplicada ao conjunto de erros percentuais obtidos em cada ensaio. Como demonstra a Equação 3:

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}} \quad (3)$$

Onde:

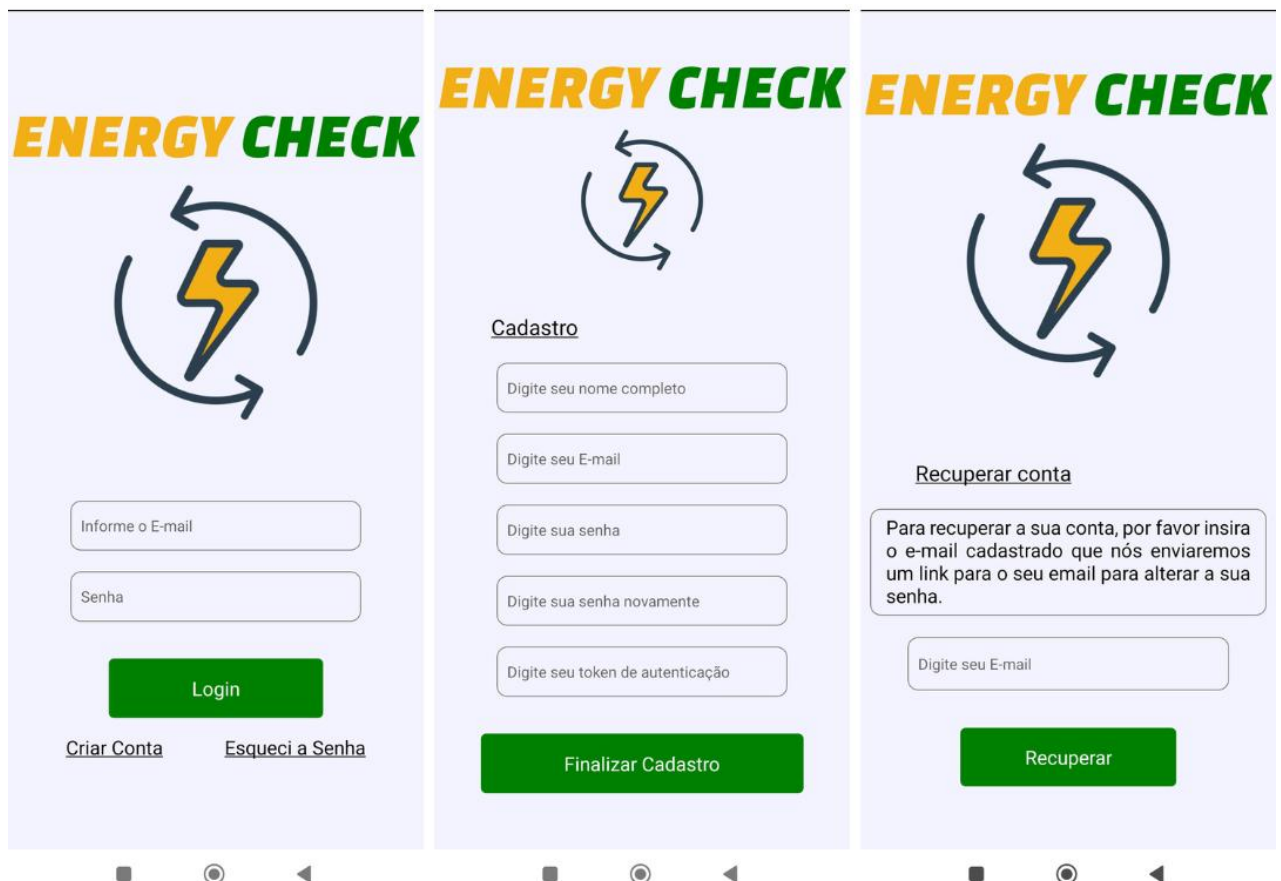
- s é o desvio padrão amostral;
- n é o número total de medições no ensaio (amostras);
- x_i o valor de um erro percentual individual;
- \bar{x} é a média aritmética de todos os erros percentuais do ensaio.

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

4.1 DESENVOLVIMENTO DO APLICATIVO

O aplicativo desenvolvido neste trabalho, buscou possibilitar ao usuário o acompanhamento em tempo real do consumo de energia elétrica residencial, fornecendo também outros dados como histórico de consumo e alguns recursos de configurações. A seguir, as principais telas do sistema são descritas, iniciando pelas telas de acesso e autenticação do usuário (Figura 17).

Figura 17 - Telas de *login*, cadastro e recuperação de senha



Fonte: elaborado pelo próprio autor.

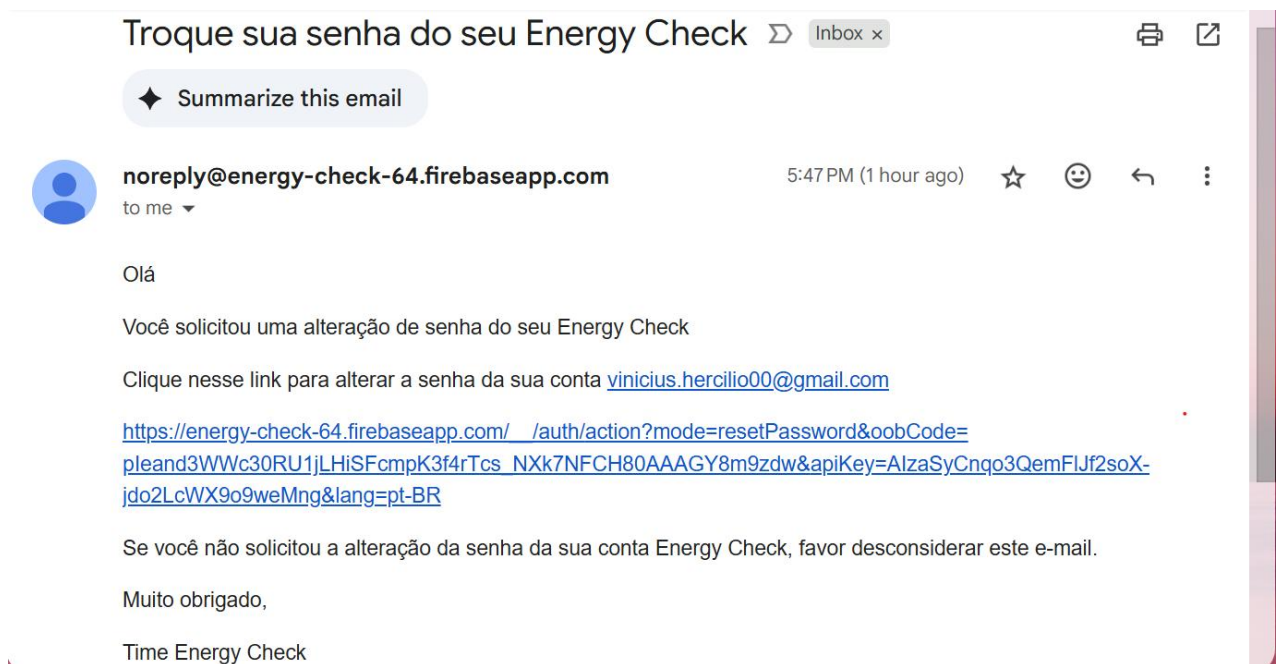
Na tela *login*, o usuário terá a opção de colocar suas credenciais, sendo elas um e-mail e senha cadastrados. Também terá a opção de “Criar Conta” e “Esqueci a Senha”. Essa etapa de validação garante que só usuários cadastrados tenham acesso aos dados do medidor de energia.

Já na tela de cadastro, o usuário a utilizará apenas uma única vez, onde nela, inserirá o seu nome completo, um e-mail válido, sua senha de acesso e o token de

autenticação, que vincula o módulo de medição (esp32), a conta que está sendo cadastrada, permitindo que o usuário visualize apenas os dados correspondentes ao seu dispositivo.

A tela de recuperação de senha, possibilita redefinir as credências a partir de um link enviado automaticamente para o e-mail cadastrado, permitindo o usuário a redefinir sua senha conforme a Figura 18.

Figura 18 - E-mail de redefinição de senha



Fonte: *Firebase* (2025).

Com isso, o indivíduo pode prosseguir com a alteração da senha apenas clicando no *hyperlink*.

A tela “Home” (Figura 19) é a tela principal do aplicativo, é onde o usuário ficaria ciente dos seus consumos.

- Preço da fatura atual estimada: esta informação representa o quanto em reais já foi gasto até o momento naquele mês.
- Preço estipulado da próxima fatura: estimativa do valor final da fatura de energia elétrica, com base nos gastos registrados até o momento.
- Potência total consumida: energia total acumulada (em kWh) pelo usuário desde o início do mês corrente até o presente momento.
- Corrente atual: corrente eficaz atual que está sendo consumida.
- Tensão atual: a tensão eficaz atual da residência.

Figura 19 - Tela *Home*

Fonte: elaborado pelo próprio autor.

A Figura 20 apresenta o relatório de consumo, onde exibe os gastos dos 3 últimos meses, extraídos diretamente das medições feitas pelo aplicativo. Na seção gasto da energia elétrica, é possível utilizar a opção de filtro, onde apresenta duas formas de visualização: semanal e diário. Na opção semanal, são exibidos os gastos ao longo de uma semana, permitindo a análise dos padrões de consumo em cada dia. Já na opção diário, é possível acompanhar o gasto de energia do dia atual, com capacidade de identificar picos de consumo específicos durante o dia.

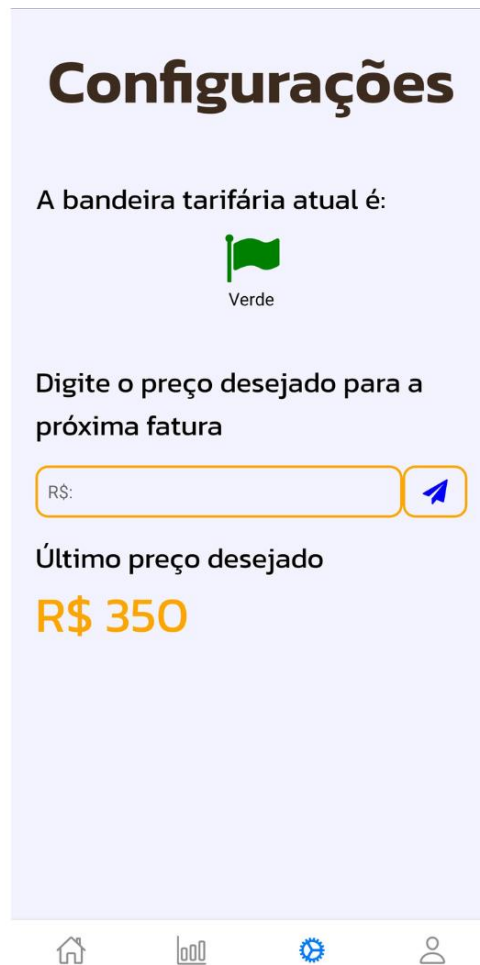
Figura 20 - Tela relatórios



Fonte: elaborado pelo próprio autor.

A tela “Configurações” (Figura 21) é a seção onde o usuário interage com as premissas do sistema. Nela, é exibida a bandeira tarifária vigente (verde, amarela, ou vermelha patamar 1 ou 2), cujos valores e definições foram detalhados na Seção 1.1.5. Esta tela também contém a funcionalidade “digite o preço desejado para a próxima fatura”, um campo onde o usuário define um valor de meta para sua fatura. Este valor é usado pelo sistema para acionar uma notificação quando o gasto estimado atingir 90% do preço estipulado.

Figura 21 - Tela configurações



Fonte: elaborado pelo próprio autor.

Figura 22 - Notificação recebida



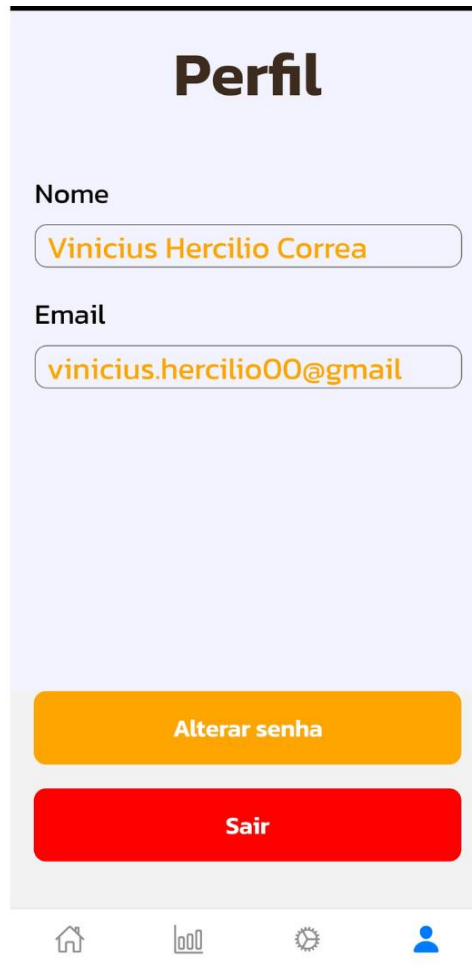
Fonte: elaborado pelo próprio autor.

A tela perfil (figura 23) exibe as informações cadastradas do usuário, como nome e endereço de e-mail, também oferece a opção de alterar a senha, que ao clicar

na opção, receberá um e-mail de alteração de senha conforme figura 18.

O botão de “sair”, o usuário será desconectado, encerrando sua seção.

Figura 23 - Tela perfil



Fonte: elaborado pelo próprio autor.

4.2 TESTES INICIAIS

4.2.1 Calibração do Sensor de Corrente

A confiabilidade de um sistema de monitoramento energético depende diretamente da acurácia de suas medições. Portanto, uma etapa crítica do projeto foi a aferição e validação do sistema de medição de corrente, composto pelo sensor não invasivo SCT-013-000 e pelo microcontrolador esp32.

O sensor SCT-013-000 opera como um transformador de corrente (TC), que fornece em seu secundário um sinal de corrente proporcional à corrente que flui no primário. Este sinal é então lido pela porta analógico-digital (ADC) do esp32. Para que

o microcontrolador possa converter o valor digital bruto lido em uma medição final em ampères (A), é necessário implementar no *software* uma constante de calibração. Esta constante é determinada analiticamente a partir da relação de transformação do sensor e dos componentes do circuito de condicionamento de sinal.

Após a realização do teste (cuja metodologia foi descrita na Seção 3.3.1), os dados coletados foram analisados para verificar a precisão do sensor de corrente SCT-013-000.

Os dados coletados no ensaio de validação estão compilados na Tabela 1. A tabela apresenta as leituras do instrumento de referência e do sensor, a diferença absoluta entre elas e o erro percentual relativo.

Tabela 1 - Teste do sensor SCT-013-000

Temperatura Velocidade		Leitura do Alicete (A)	Leitura do Sensor (A)	Diferença (A) (Alicete – Sensor)	Erro (%)
Frio	Baixa	0,85	0,89	-0,04	4,71 %
Frio	Alta	1,43	1,42	0,01	0,70 %
Morno	Baixa	2,59	2,56	0,03	1,16 %
Morno	Alta	5,30	5,21	0,09	1,69 %
Quente	Baixa	5,69	5,51	0,18	3,16 %
Quente	Alta	8,58	8,46	0,14	1,40 %

Fonte: elaborado pelo próprio autor.

A análise estatística dos erros percentuais, conforme a equação 2 e a equação 3 (Seção 3.3.5), revelou um erro médio (\bar{x}) de 2,14 % e um desvio padrão amostral (s) de 1,51 %.

A análise da Tabela 1 demonstra a alta correlação entre o sistema desenvolvido e o instrumento de referência. Observa-se que o erro máximo (4,71 %) ocorreu na medição de corrente mais baixa (0,85 A). Esta discrepância é esperada e consistente com o comportamento de transformadores de corrente, que intrinsecamente apresentam menor linearidade e precisão em valores muito baixos, próximos ao "fundo de escala" de sua capacidade de detecção.

O desvio padrão de 1,51 % reflete essa variação na exatidão do sensor em diferentes faixas de operação. Entretanto, para as faixas de operação mais representativas do consumo residencial típico (acima de 1,4 A, neste ensaio), o erro

percentual relativo manteve-se consistentemente abaixo de 3,2 %, com a maioria das leituras apresentando erros inferiores a 1,7 %. A diferença absoluta, mesmo na medição mais alta (8,58 A), foi de apenas 0,14 A.

Estes resultados validam experimentalmente a constante de calibração utilizada no *software* do esp32 e atestam que o sistema de medição possui acurácia e confiabilidade suficientes para os objetivos do projeto *EnergyCheck*, fornecendo informações precisas para o monitoramento de consumo e o gerenciamento eficiente dos gastos energéticos.

4.2.2 Calibração do Sensor de Tensão

Juntamente com a corrente, a medição precisa da tensão da rede elétrica é um pilar fundamental para o cálculo correto da potência instantânea, conforme equação 2, e conseqüentemente, do consumo de energia. O sensor ZMPT101B é um transformador de potencial miniatura que isola e reduz a tensão da rede (220 VCA) para um nível de sinal seguro e compatível com a porta analógico-digital (ADC) do esp32. Esta constante é a responsável por converter a leitura do ADC em um valor final representativo da Tensão Eficaz.

Os resultados da validação experimental, cuja metodologia foi descrita na Seção 3.3.2, estão compilados na Tabela 2.

Tabela 2 - Teste do sensor ZMPT101B

Tensão Nominal (V)	Leitura do Alicata (V)	Leitura do Sensor (V)	Diferença (V)	Erro (%)
220	227	228	1	0,44 %
220	223	224	1	0,44 %
220	227	227	0	0,00 %
220	221	220	-1	-0,44 %
220	225	225	0	0,00 %
220	221	222	1	0,44 %
220	224	223	-1	-0,44 %
220	223	223	0	0,00 %
220	225	226	1	0,44 %
220	224	224	0	0,00 %

Fonte: elaborado pelo próprio autor.

A análise dos dados da Tabela 2 demonstra uma notável exatidão e precisão do sistema de medição de tensão. A análise estatística dos erros percentuais, conforme as equações 2 e 3 (Seção 3.3.5), revelou um erro médio (\bar{x}) de apenas 0,088%, indicando que o sensor não possui um viés significativo de medição.

Adicionalmente, o desvio padrão amostral (s) foi de apenas 0,345%. Este valor, muito próximo de zero, comprova a alta precisão (consistência) das medições, atestando que o sistema é confiável e suas leituras são altamente repetíveis.

A análise qualitativa da tabela corrobora esses dados. Observa-se que o erro percentual relativo máximo foi de apenas 0,44%, com múltiplas medições apresentando concordância exata. Em todas as dez medições realizadas, a diferença absoluta entre o valor de referência e o valor medido pelo sensor não ultrapassou 1 V.

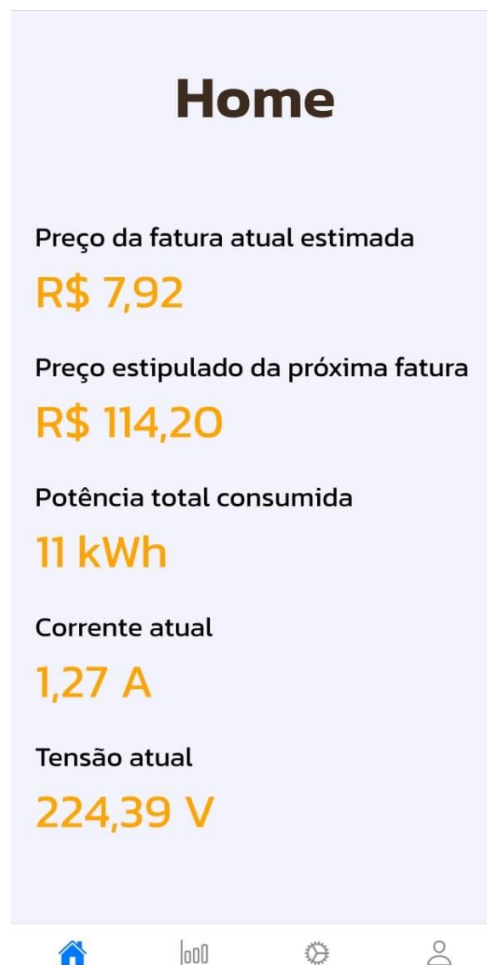
É fundamental destacar que a diferença máxima de 1 V (como no ensaio de 227 V contra 228 V) está confortavelmente dentro da própria margem de precisão do instrumento de referência. A precisão do Hikari de $\pm(1,2\% + 5$ dígitos) sobre uma leitura de 227 V, como descrito na metodologia, resulta em uma incerteza de medição que supera 1 V.

Este resultado demonstra que os valores lidos pelo sensor ZMPT101B são estatisticamente muito próximos aos do instrumento de referência. Os dados validam experimentalmente a constante de calibração utilizada no *software* e indicam que o sistema de medição de tensão é plenamente confiável para prover os dados necessários aos cálculos de potência e energia do aplicativo *EnergyCheck*.

4.3 TESTE DE 24 HORAS EM AMBIENTE REAL

Os resultados do teste de 24 horas, cuja metodologia foi detalhada na Seção 3.3.3, estão apresentados na Tabela 3. A Figura 24 exibe a tela do aplicativo *EnergyCheck* com o valor acumulado pelo protótipo ao final do período.

Figura 24 - Aplicativo *EnergyCheck* apresentando os valores medidos



Fonte: elaborado pelo próprio autor.

Tabela 3 - Comparação entre o consumo medido pelo *EnergyCheck* e o medidor da CELESC.

Medidor Utilizado	Consumo (kWh)
<i>EnergyCheck</i>	11
CELESC	11
Diferença	0

Fonte: elaborado pelo próprio autor.

Observa-se na Tabela 3 que o consumo medido pelo *EnergyCheck* foi de 11 kWh, enquanto o consumo registrado pelo medidor da CELESC foi o de mesmo valor.

Discussão: Os resultados demonstraram a capacidade do *EnergyCheck* em monitorar o consumo de energia elétrica em um ambiente residencial real, com resultados promissores. A concordância exata entre as medições do *EnergyCheck* (11 kWh) e do medidor da CELESC (11 kWh) indica a efetividade do sistema na coleta e processamento dos dados de consumo.

É importante reconhecer que a resolução do medidor da CELESC, que apresenta apenas números inteiros, limita a comparação a um nível de detalhamento mais aprofundado. Não é possível afirmar que o consumo foi exatamente 11,000 kWh em ambos; o resultado indica que o consumo em ambos os medidores esteve dentro da mesma faixa (entre 11,0 kWh e 11,99 kWh) durante o período. No entanto, a concordância observada sugere que o *EnergyCheck* fornece uma estimativa confiável do consumo de energia, suficiente para atender ao objetivo de monitoramento e auxiliar os usuários na gestão do seu consumo.

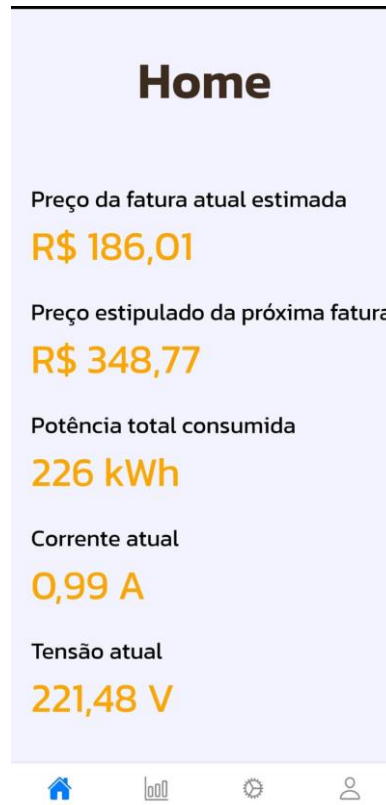
Os dados coletados pelo *EnergyCheck* podem ser utilizados pelos usuários para acompanhar seus hábitos de consumo, identificar possíveis desperdícios e tomar medidas para reduzir o consumo de energia. O sistema oferece uma ferramenta acessível e prática para o monitoramento do consumo de energia residencial, contribuindo para a conscientização e o uso eficiente da energia elétrica.

4.4 TESTE DE DUAS SEMANAS EM AMBIENTE REAL

Os resultados do teste de longa duração, cuja metodologia foi detalhada na Seção 3.3.4, estão compilados na Tabela 4. A Figura 25 apresenta a tela principal do

aplicativo EnergyCheck ao final do período de teste, exibindo o valor total acumulado pelo protótipo.

Figura 25 - *EnergyCheck* exibindo os valores medidos após duas semanas



Fonte: elaborado pelo próprio autor.

Tabela 4 - Comparação entre o consumo medido pelo *EnergyCheck* e o medidor da CELESC

Medidor Utilizado	Consumo (kWh)
<i>EnergyCheck</i>	226
CELESC	231
Diferença	5
Erro (%)	2,21 %

Fonte: elaborado pelo próprio autor.

Discussão: Os resultados do teste de duas semanas demonstram que o sistema *EnergyCheck* manteve um desempenho estável ao longo do tempo. O protótipo registrou 226 kWh, enquanto o medidor da concessionária registrou 231 kWh, resultando em uma diferença de 5 kWh e um erro percentual de 2,21% em relação ao padrão.

Esse nível de precisão é considerado aceitável para sistemas de monitoramento não homologados, especialmente em aplicações residenciais de conscientização. A diferença observada pode ser atribuída a um conjunto de fatores, como pequenas variações na calibração dos sensores, oscilações na rede elétrica, ruídos eletromagnéticos ou imprecisões na integração das leituras acumuladas ao longo do tempo. Apesar disso, o *EnergyCheck* apresentou um desempenho consistente e próximo da realidade, reafirmando seu potencial como uma ferramenta eficaz para o monitoramento do consumo energético.

5 CONCLUSÕES E CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo o desenvolvimento e a avaliação técnica do *EnergyCheck*, um sistema de monitoramento de consumo de energia elétrica composto por sensores de corrente e tensão, um microcontrolador ESP32 e um aplicativo móvel desenvolvido em *React Native*.

Foram realizados testes com os sensores SCT-013-000 e ZMPT101B, além de testes práticos em ambiente real, incluindo um monitoramento de 24 h e outro de duas semanas. Os resultados demonstraram que o sistema é capaz de medir o consumo de energia elétrica com boa precisão, apresentando erro percentual inferior a 2,5 % em comparação ao medidor oficial da concessionária CELESC.

Através dos testes, verificou-se também a estabilidade do sistema em funcionamento contínuo, bem como a consistência na transmissão e armazenamento dos dados. Todos os dados coletados são armazenados em nuvem, garantindo que as leituras não sejam perdidas mesmo em casos de interrupção de energia elétrica.

5.1 TRABALHOS FUTUROS

Com base nos resultados obtidos, algumas melhorias podem ser consideradas para versões futuras, do sistema *EnergyCheck*:

- Implementação de um sistema de calibração automática para maior precisão nas medições.
- Adição de alertas em tempo real via aplicativo para detecção de picos de consumo.
- Integração com painéis *web* administrativos para visualização em múltiplas

unidades.

- Expansão do sistema para monitoramento bifásico/trifásico.

5.2 CONSIDERAÇÕES FINAIS

Os códigos-fonte do sistema, incluindo o *firmware* do ESP32, o aplicativo móvel e os *scripts* de testes, estão disponíveis no seguinte repositório: <https://github.com/viniciushc26/EnergyCheck>

O *EnergyCheck* demonstrou ser uma ferramenta eficiente e acessível para o monitoramento do consumo elétrico residencial. A precisão obtida nos testes, aliada ao armazenamento em nuvem e à visualização centralizada dos dados no aplicativo móvel, evidencia o potencial de uso contínuo do sistema por usuários finais. Com isso, o projeto cumpre seu papel de fornecer uma solução técnica confiável para acompanhamento do consumo energético, contribuindo para o uso consciente da energia elétrica.

REFERÊNCIAS

AGÊNCIA NACIONAL DE ENERGIA ELÉTRICA (ANEEL). **Calendário de acionamento de bandeiras tarifárias para 2025 já está disponível**. Brasília, DF, 20 dez. 2024. Disponível em: <https://www.gov.br/aneel/pt-br/assuntos/noticias/2025/calendario-de-acionamento-de-bandeiras-tarifarias-para-2025-ja-esta-disponivel>. Acesso em: 18 out. 2025.

ASSOCIAÇÃO DOS GRANDES CONSUMIDORES INDUSTRIAIS DE ENERGIA E DE CONSUMIDORES LIVRES (ABRACE). **Brasil tem a conta de luz que mais pesa no bolso da população entre 34 países**. Brasília, DF: ABRACE, 2023. Disponível em: <https://abrace.org.br/site/brasil-tem-a-conta-de-luz-que-mais-pesa-no-bolso-da-populacao-entre-34-paises/>. Acesso em: 16 abr. 2025.

ASSOCIAÇÃO DOS GRANDES CONSUMIDORES INDUSTRIAIS DE ENERGIA E DE CONSUMIDORES LIVRES (ABRACE). **Os impactos do aumento dos custos de energia no orçamento dos brasileiros**. Brasília, DF: ABRACE, 2023. Disponível em: <https://abrace.org.br/site/os-impactos-do-aumento-dos-custos-de-energia-no-orcamento-dos-brasileiros/>. Acesso em: 16 abr. 2025.

BOYLESTAD, R. L.; NASHESKY, L. **Dispositivos Eletrônicos e Teoria de Circuitos**. 11. ed. São Paulo: Pearson Prentice Hall, 2014.

CENTRAIS ELÉTRICAS DE SANTA CATARINA (CELESC). **A Celesc Distribuição**. Florianópolis, [s.d.]. Disponível em: <https://www.celesc.com.br/a-celesc-distribuicao>. Acesso em: 18 out. 2025.

CENTRAIS ELÉTRICAS DE SANTA CATARINA (CELESC). **Como é definido o valor da conta de energia?** Florianópolis, [2025?]. Disponível em: <https://transparencia.celesc.com.br/pergunta-cidada/definido-valor-conta>. Acesso em: 1 maio 2025.

CENTRAIS ELÉTRICAS DE SANTA CATARINA (CELESC). **Manual Especial - Medidor Eletrônico Monofásico, Bifásico e Trifásico com Saída Serial**: Especificação Técnica E-321.0008. Florianópolis, 2021. Disponível em: <https://www.celesc.com.br/arquivos/normas-tecnicas/equipamentos-medicao/E3210008.pdf>. Acesso em: 18 out. 2025.

CENTRAIS ELÉTRICAS DE SANTA CATARINA (CELESC). **Tarifas de energia**. Florianópolis, [2025?]. Disponível em: <https://www.celesc.com.br/tarifas-de-energia>. Acesso em: 29 jul. 2025.

DARBY, S. **The effectiveness of feedback on energy consumption**. Oxford: Environmental Change Institute, University of Oxford, 2006. Disponível em: https://www.researchgate.net/publication/238785702_The_Effectiveness_of_Feedback_on_Energy_Consumption. Acesso em: 2 maio 2025.

EMPRESA DE PESQUISA ENERGÉTICA (EPE). **Anuário Estatístico de Energia Elétrica 2023**. Brasília, DF: EPE, 2023. Disponível em: <https://www.epe.gov.br/pt/publicacoes-dados-abertos/publicacoes/Anuario-Estatistico-de-Energia-Eletrica>. Acesso em: 16 abr. 2025.

EMPRESA DE PESQUISA ENERGÉTICA (EPE). **Balço Energético Nacional 2006**: Ano base 2005 - Relatório Final. Rio de Janeiro: Ministério de Minas e Energia, 2006. Disponível em: <https://www.epe.gov.br/pt/publicacoes-dados-abertos/publicacoes/Balanco-Energetico-Nacional-BEN>. Acesso em: 14 abr. 2025.

EMPRESA DE PESQUISA ENERGÉTICA (EPE). **Balço Energético Nacional 2024**: Ano base 2023 - Relatório Final. Rio de Janeiro: Ministério de Minas e Energia, 2024. Disponível em: <https://www.epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/PublicacoesArquivos/publicacao-819/topico-723/BEN2024.pdf>. Acesso em: 14 abr. 2025.

FROEHLICH, J.; FINDLATER, L.; LANDAY, J. A. The design of eco-feedback technology. In: PROCEEDINGS OF THE SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 2010, Atlanta. **Anais [...]**. Atlanta: ACM, 2010. p. 1999–2008. Disponível em: <https://dl.acm.org/doi/abs/10.1145/1753326.1753629>. Acesso em: 2 maio 2025.

GOLDEMBERG, J. Energia e desenvolvimento. **Estudos Avançados**, São Paulo, v. 12, n. 33, p. 7–15, 1998. Disponível em: <https://www.revistas.usp.br/eav/article/view/9402>. Acesso em: 14 abr. 2025.

GREENPEACE. **Mudanças do clima, mudanças de vidas**: como o aquecimento global já afeta o Brasil. São Paulo: Greenpeace Brasil, 2006. Disponível em: <https://www.greenpeace.org/brasil/blog/mudancas-do-clima-mudancas-de-vidas>. Acesso em: 16 abr. 2025.

HIKARI. **Alicate Amperímetro Digital Compacto True RMS**. [S. l.], 2024. Disponível em: <https://www.hikariferramentas.com.br/alicate-amperimetro-digital-compacto-true-rms/1216/104/>. Acesso em: 1 nov. 2024.

HIKARI. **Manual de instruções: Alicate Amperímetro Digital HA-3120**. [S. l.]: Hikari Instrumentos, [s.d.]. Disponível em: <https://hikari.nyc3.cdn.digitaloceanspaces.com/hhvozhy3zivq12mo7xzriux3w08u>. Acesso em: 16 out. 2025.

INSTITUTO PÓLIS. **Justiça Energética – Pesquisa de Opinião Pública**. São Paulo: Instituto Pólis, 2024. Disponível em: <https://polis.org.br/estudos-e-pesquisas-do-instituto-polis/justica-energetica-pesquisa-de-opiniao-publica/>. Acesso em: 14 abr. 2025.

LABORATÓRIO DE EFICIÊNCIA ENERGÉTICA EM EDIFICAÇÕES (LABEEE). **Usos finais de energia elétrica no setor residencial brasileiro**. Florianópolis: UFSC, 2015. Disponível em: <https://labeee.ufsc.br/pt-br/node/480>. Acesso em: 16 abr. 2025.

MURTAGH, N.; GATERSLEBEN, B.; UZZELL, D. Motivating energy conservation in the workplace: An evaluation of the use of group-level feedback and peer education. **Journal of Environmental Psychology**, v. 31, n. 1, p. 1–13, 2011. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0272494410001015>. Acesso em: 2 maio 2025.

PEREIRA, L. H. J. **Monitoramento do consumo de energia elétrica e controle de equipamentos via aplicativo**. 2018. 60 f. Monografia (Graduação em Engenharia de Controle e Automação) – Escola de Minas, Universidade Federal de Ouro Preto, Ouro Preto, 2018. Disponível em: https://www.monografias.ufop.br/bitstream/35400000/922/1/MONOGRAFIA_MoniotramentoConsumoEnergia.pdf. Acesso em: 16 out. 2025.

PIRES, Laryssa S.; O. **Análise do perfil de consumo de energia elétrica do bloco F da Pontifícia Universidade Católica de Goiás**. 2021. Trabalho de Conclusão de Curso (Engenharia Elétrica) – Pontifícia Universidade Católica de Goiás, Goiânia, 2021. Disponível em: https://repositorio.pucgoias.edu.br/jspui/bitstream/123456789/3092/1/TCC_II_LARYSSA%20SALUSTIANA%20DE%20OLIVEIRA%20PIRES_VersaoFinal.pdf. Acesso em: 12 ago. 2025.

ROBOCORE. **ESP32 - WiFi + Android**. [S. l.], 2024. Disponível em: <https://www.robocore.net/wifi/esp32-wifi-Android>. Acesso em: 3 out. 2024.

ROBOCORE. **Módulo Sensor de Tensão AC 0 A 250V ZMPT101B**. [S. l.], 2024. Disponível em: <https://www.robocore.net/sensor-corrente-tensao/modulo-sensor-de-tensao-ac-zmpt101b>. Acesso em: 3 out. 2024.

ROBOCORE. **Sensor de Corrente Não Invasivo 100A SCT-013**. [S. l.], 2024. Disponível em: <https://www.robocore.net/sensor-corrente-tensao/sensor-de-corrente-nao-invasivo-100a-sct-013>. Acesso em: 3 out. 2024.

SCHULER, R. **Sistema de Monitoramento de Consumo de Energia Elétrica para Circuitos Monofásicos de Baixa Tensão**. 2020. Trabalho de Conclusão de Curso (Graduação em Engenharia de Controle e Automação) – Universidade do Vale do Taquari, Lajeado, 2020. Disponível em: <https://www.univates.br/bdu/bitstream/10737/3196/1/2020RicardoSchuler.pdf>. Acesso em: 16 out. 2025.

APÊNDICE A – CÓDIGO FONTE DO HARDWARE (ESP32)

Este apêndice apresenta os trechos mais relevantes do código-fonte desenvolvido para o microcontrolador ESP32. O código foi modularizado para melhor organização, separando a inicialização (Main), as tarefas de processamento em tempo real (Task1 e Task2) e as funções de apoio (Funcoes).

O código-fonte completo e todos os arquivos do projeto estão disponíveis no repositório digital: <https://github.com/viniciushc26/EnergyCheck>

A.1 MÓDULO PRINCIPAL (ABA MAIN)

Esta seção é a mais importante. Ela mostra todas as bibliotecas, as configurações de hardware, a inicialização do sistema e a criação das duas tasks principais (a arquitetura FreeRTOS), além da leitura dos sensores no loop().

```

/*////////////////////////////////////*/
/*                                     */
/*          CONFIGURAÇÕES DO USUÁRIO          */
/*                                     */
/*////////////////////////////////////*/

/*

Insira seu endereço de e-mail cadastrado abaixo. Isso permite a
personalização e envio de dados para o banco de dados. Exemplo:

const char* usuarioEmail = "seu_email@example.com";
*/

const char* usuarioEmail = "usuario_exemplo@example.com";

```

```
#include <WiFi.h>
#include <IOXhop_FirebaseESP32.h>
#include <ArduinoJson.h>
#include "time.h"
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <EmonLib.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include <ZMPT101B.h>

#define WIFI_SSID "NOME_REDE_OCULTO"
#define WIFI_PASSWORD "*****"
#define FIREBASE_HOST "https://energy-check-64-default-
rtdb.firebaseio.com/"
#define FIREBASE_AUTH "*****"

#define SENSITIVITY 466.25f

#define PINO_CORRENTE 34 // Pino ao qual o sensor de corrente está
conectado

ZMPT101B voltageSensor(32, 60.0);

EnergyMonitor emon1;

TaskHandle_t Task1;
TaskHandle_t Task2;

String emailFormatado;

const long gmtOffset_sec = -3 * 3600;
```

```

        const int daylightOffset_sec = 0;                                // Sem horário
de verão
        WiFiUDP ntpUDP;                                              // Crie um objeto
UDP para comunicação com o servidor NTP
        NTPClient timeClient(ntpUDP, "pool.ntp.org",                gmtoffset_sec,
daylightOffset_sec); // Crie um objeto NTPClient

int aux = 0;
double tensao_Atual = 0, corrente_Atual = 0;
void setup(){
    Serial.begin(115200);

    emailFormatado = usuarioEmail;
    emailFormatado.replace(".", "_");
    emailFormatado.replace("-", "_");

    xTaskCreatePinnedToCore(
        Task1code, /* Task function. */
        "Task1", /* name of task. */
        50000, /* Stack size of task */
        NULL, /* parameter of the task */
        1, /* priority of the task */
        &Task1, /* Task handle to keep track of created task */
        0); /* pin task to core 0 */
    delay(500);

    xTaskCreatePinnedToCore(
        Task2code, /* Task function. */
        "Task2", /* name of task. */
        50000, /* Stack size of task */
        NULL, /* parameter of the task */
        1, /* priority of the task */
        &Task2, /* Task handle to keep track of created task */

```

```
        1);    /* pin task to core 0 */
delay(500);

    emon1.current(PINO_CORRENTE, 10.7); // Configura o sensor de corrente,
com fator de calibração de 10.7
    voltageSensor.setSensitivity(SENSITIVITY); // Configura o sensor de tensao

conectar_WiFi();
Ler_dados();
timeClient.begin();
hora_Atual();
data_Atual();
meses_Passados();
bandeira();
}

void loop(){
    float tensao_Somada = 0;
    float corrente_Somada = 0;

    for (int i = 0; i < 40; i++) {
        tensao_Somada += voltageSensor.getRmsVoltage();
    }
    tensao_Atual = tensao_Somada/40;

    // Serial.print("A tensão é: ");
    // Serial.print(tensao_Atual);
    // Serial.println("V");

    for (int i = 0; i < 5; i++) {
        corrente_Somada += emon1.calcIrms(1480);
    }
}
```

```

    corrente_Atual = corrente_Somada/5;
    corrente_Somada = 0;
    //Serial.print("A corrente é: ");
    //Serial.print(corrente_Atual);
    //Serial.println("A");

}

```

A.2 MÓDULOS DE TAREFAS (ABAS TASK1 E TASK2)

```

    double    fatura_Atual,    Gasto_ultimo_Mes,    Gasto_penultimo_Mes,
    Gasto_antepenultimo_Mes, potencia_Total, Proxima_fatura;
    double kWh_Seg = 0, kWh_Ter = 0, kWh_Qua = 0, kWh_Qui = 0, kWh_Sex =
    0, kWh_Sab = 0, kWh_Dom = 0;

    int Aux_hora = 300, mes, dia_Semana, mes_Passado1, mes_Passado2,
    mes_Passado3, dia_Mes;
    int bandeira_vigente = 0;
    int AuxDom = 0, AuxSeg = 0, AuxTer = 0, AuxQua = 0, AuxQui = 0, AuxSex
    = 0, AuxSab = 0;

    int Aux_dia_Semana = 100;
    float kWh_Diario = 0;
    String aux_ultimo_Mes;
    double preco_kWh =0;
    double kWh_Hora = 0;

    double potencia_Atual = 0;
    double kW_Consumido =0;
    double kWh_Instantaneo =0;

```

```
unsigned long lastmillis = millis();
int aux_bandeira = 0;
float restante_Mes = 0;
unsigned long Tempo = 0;

void Task1code(void * pvParameters) {
  for (;;) {

    if (aux == 0) {
      delay(45000);
      aux++;
    }

    if ((millis() - lastmillis) > 900) {
      Tempo = millis() - lastmillis;

      Serial.print("Tempo da iteraç o: ");
      Serial.print(Tempo);
      Serial.println(" ms");

      lastmillis = millis();

      // Serial.print("A potencia total consumida  : ");
      // Serial.print(potencia_Total);
      // Serial.println("kWh");

      potencia_Atual = corrente_Atual * tensao_Atual; // potencia
      kW_Consumido = potencia_Atual / 1000; // kw
      kWh_Instantaneo = kW_Consumido / 3600; // kwh
      potencia_Total += kWh_Instantaneo;
      fatura_Atual = potencia_Total * preco_kWh; // fatura_Atual
      kWh_Hora += kWh_Instantaneo * preco_kWh;
      kWh_Diario += kWh_Instantaneo * preco_kWh;
```

```

    restante_Mes = 30 - dia_Mes;
    Proxima_fatura = fatura_Atual / dia_Mes * 30;
    // fatura_Estimada = fatura_Atual + (kWh_Seg + kWh_Ter + kWh_Qua +
kWh_Qui + kWh_Sex + kWh_Sab + kWh_Dom) / 7 * restante_Mes * preco_kWh;
    }

```

```

    // Aguarda o próximo ciclo
    vTaskDelay(pdMS_TO_TICKS(100));
    }
}

```

```

unsigned long Tempo_Funcao = millis();
int aux_Task2 = 0;

```

```

void Task2code( void * pvParameters ){

```

```

    for(;;) {

```

```

        if(aux_Task2 == 0){
            aux_Task2 ++;
            delay(60000);
        }

```

```

        if(millis() - Tempo_Funcao > 5000){
            tela_Home();
            hora_Atual();
        }
    }
}

```

```

data_Atual();
meses_Passados();
bandeira();
Tempo_Funcao = millis(); // Atualiza Tempo_Funcao para o tempo atual
}

```

```

vTaskDelay(pdMS_TO_TICKS(1000)); // Atraso de 1000 milissegundos (1
segundo)

```

```

}
}

```

A.3 MÓDULO DE FUNÇÕES (ABA FUNCOES)

```

void conectar_WiFi() {
  unsigned long startTime = millis();
  const unsigned long timeout = 30000; // 30 segundos de timeout

  if (WiFi.status() != WL_CONNECTED) {
    while (WiFi.status() != WL_CONNECTED && millis() - startTime < timeout) {
      WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
      Serial.print("Connecting to Wi-Fi");

      while (WiFi.status() != WL_CONNECTED) {
        Serial.print(".");
        delay(300);
      }

      Serial.println();
      Serial.print("Conectado ao IP: ");
      Serial.println(WiFi.localIP());
      Serial.println();
    }
  }
}

```

```

    conectar_FireBase();

}
}
}

/*****
*****/

void conectar_FireBase() {
    Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
    Serial.println("Conectado ao Firebase");
    Serial.println();
}

/*****
*****/

void tela_Home(){
    if (WiFi.status() == WL_CONNECTED) {
        String path1 ="users/" + emailFormatado + "/dados/home/Fatura_atual";
        String path2 ="users/" + emailFormatado + "/dados/home/Potencia_total";
        String path3 ="users/" + emailFormatado + "/dados/home/Proxima_fatura";
        String path4 ="users/" + emailFormatado + "/dados/home/Corrente";
        String path5 ="users/" + emailFormatado + "/dados/home/Tensao";

        Firebase.setFloat(path1,fatura_Atual);
        Firebase.setFloat(path2,potencia_Total);
        Firebase.setFloat(path3,Proxima_fatura);
        Firebase.setFloat(path4,corrente_Atual);
        Firebase.setFloat(path5,tensao_Atual);
    }
}

```

```

    }
}

/*****
*****/

void meses_Passados(){

    // Serial.println("Começando a obter os meses passados ");

    const char *nomesMeses[] = {"", "Janeiro", "Fevereiro", "Março", "Abril",
"Maio", "Junho", "Julho", "Agosto", "Setembro", "Outubro", "Novembro", "Dezembro"};

    String path = "users/" + emailFormatado +
"/dados/relatorios/Mensal/Gasto_antepenultimo_Mes";
    String path2 = "users/" + emailFormatado +
"/dados/relatorios/Mensal/Gasto_penultimo_Mes";
    String path3 = "users/" + emailFormatado +
"/dados/relatorios/Mensal/Gasto_ultimo_Mes";
    String path4 = "users/" + emailFormatado +
"/dados/relatorios/Mensal/Ultimo_mes";
    String path5 = "users/" + emailFormatado +
"/dados/relatorios/Mensal/Penultimo_mes";
    String path6 = "users/" + emailFormatado +
"/dados/relatorios/Mensal/Antepenultimo_mes";
    String path7 = "users/" + emailFormatado + "/dados/home/Fatura_atual";
    String path8 = "users/" + emailFormatado + "/dados/home/Potencia_total";

    if (mes >= 1 && mes <= 12) {

```

```
String ultimo_Mes = nomesMeses[mes_Passado1];
String penultimo_Mes = nomesMeses[mes_Passado2];
String antepenultimo_Mes = nomesMeses[mes_Passado3];

if (aux_ultimo_Mes != ultimo_Mes) {
    aux_ultimo_Mes = ultimo_Mes;           // Atualiza a variável auxiliar

    Firebase.setFloat(path,Gasto_penultimo_Mes);

    Firebase.setFloat(path2,Gasto_ultimo_Mes);

    Gasto_ultimo_Mes = fatura_Atual;

    Firebase.setFloat(path3,Gasto_ultimo_Mes);

    fatura_Atual = 0;
    Firebase.setFloat(path7,fatura_Atual);

    potencia_Total = 0;
    Firebase.setFloat(path8,potencia_Total);

    Firebase.setFloat(path,Gasto_penultimo_Mes);

    Firebase.setString(path4, ultimo_Mes);

    Firebase.setString(path5, penultimo_Mes);

    Firebase.setString(path6, antepenultimo_Mes);
}

}
```

```

// Serial.println("Meses passados obtidos ");
// Serial.println();
}

/*****
*****/

void bandeira(){

//Serial.println("Obtendo a bandeira tarifária");

String path1 = "/configuracoes/bandeira";
bandeira_vigente = Firebase.getInt(path1);

switch (bandeira_vigente){
  case 0:
    if(potencia_Total <= 150){
      preco_kWh = 0.73;
    }else{
      preco_kWh = 0.77;
    }
  }

  case 1:
    //Serial.println("A bandeira tarifária é a amarela");
    if(potencia_Total <= 150){
      preco_kWh = 0.77;
    }else{
      preco_kWh = 0.82;
    }
    //Serial.print("O preço do kWh é: R$");
    //Serial.println(preco_kWh);
    break;
}

```

```
case 2:
  if(potencia_Total <= 150){
    preco_kWh = 0.83;
  }else{
    preco_kWh = 0.88;
  }
  break;

case 3:
  if(potencia_Total <= 150){
    preco_kWh = 0.96;
  }else{
    preco_kWh = 1.02;
  }
  break;

default:
  Serial.println("Não consegui encontrar o preço do kwh correto!");
  break;

}
}
```

APÊNDICE B – CÓDIGO-FONTE DO SOFTWARE (APLICATIVO)

Este apêndice apresenta os arquivos e trechos de código mais relevantes do aplicativo EnergyCheck, desenvolvido em React Native e Expo. O código-fonte completo está disponível no repositório digital do projeto: <https://github.com/viniciushc26/EnergyCheck>

B.1 DEPENDÊNCIAS DO PROJETO (PACKAGE.JSON)

O arquivo package.json define o projeto e lista todas as bibliotecas e *frameworks* de terceiros utilizados, demonstrando as tecnologias que fundamentam

o aplicativo.

```
{
  "name": "energycheck",
  "version": "1.0.0",
  "main": "node_modules/expo/AppEntry.js",
  "scripts": {
    "start": "expo start",
    "android": "expo start --android",
    "ios": "expo start --ios",
    "web": "expo start --web"
  },
  "dependencies": {
    "@expo-google-fonts/changa-one": "^0.2.3",
    "@expo-google-fonts/kanit": "^0.2.3",
    "@expo/vector-icons": "^14.0.0",
    "@react-native-async-storage/async-storage": "^1.21.0",
    "@react-navigation/bottom-tabs": "^6.5.11",
    "@react-navigation/drawer": "^6.6.6",
    "@react-navigation/native": "^6.1.9",
    "@react-navigation/native-stack": "^6.9.17",
    "expo": "~50.0.5",
    "expo-font": "~11.10.2",
    "expo-notifications": "~0.27.6",
    "expo-status-bar": "~1.11.1",
    "firebase": "^10.7.2",
    "react": "18.2.0",
    "react-native": "0.73.2",
    "react-native-exit-app": "^2.0.0",
    "react-native-gifted-charts": "^1.4.3",
    "react-native-linear-gradient": "^2.8.3",
    "react-native-safe-area-context": "4.8.2",
    "react-native-screens": "~3.29.0",
    "react-native-svg": "^14.1.0"
  }
}
```

```

},
"devDependencies": {
  "@babel/core": "^7.20.0"
},
"private": true
}

```

B.2 CONFIGURAÇÃO DO FIREBASE (SRC/FIREBASE.CONFIG.JS)

Este arquivo é responsável por inicializar a conexão com o banco de dados Firebase Realtime Database e o serviço de autenticação, utilizando persistência de dados local (AsyncStorage).

```

//bibliotecas importadas para o correto funcionamento
import { initializeApp } from "firebase/app";
import { getDatabase } from 'firebase/database';
import { initializeAuth, getReactNativePersistence } from 'firebase/auth';
import AsyncStorage from '@react-native-async-storage/async-storage';

//dados para a conexão ao firebase
// As chaves de API e configuração foram ocultadas por segurança.
const firebaseConfig = {
  apiKey: "*****",
  authDomain: "energy-check-64.firebaseio.com",
  databaseURL: "https://energy-check-64-default-rtdb.firebaseio.com",
  projectId: "energy-check-64",
  storageBucket: "*****",
  messagingSenderId: "*****",
  appId: "*****",
  measurementId: "*****"
};

//inicialização e autenticação do firebase
const firebaseApp = initializeApp(firebaseConfig);

```

```

const auth = initializeAuth(firebaseApp,{
  persistence: getReactNativePersistence(AsyncStorage)
});
const database = getDatabase();

//exportando database e auth
export { database, auth };

```

B.3 NAVEGAÇÃO E PONTO DE ENTRADA (APP.JS)

O arquivo App.js define a estrutura de navegação do aplicativo. Ele utiliza um StackNavigator para as telas de autenticação (AuthStack) e um BottomTabNavigator para as telas principais (MainTabNavigator), além de gerenciar a persistência de login.

```

import React, { useEffect, useState } from 'react';
import { NavigationContainer } from '@react-navigation/native';
import { StatusBar } from 'react-native';
import { createNativeStackNavigator } from '@react-navigation/native-stack';
import { createBottomTabNavigator } from '@react-navigation/bottom-tabs';
import { Home } from './src/pages/home';
import { Configuracoes } from './src/pages/configuracoes';
import { Relatorios } from './src/pages/relatorios';
import { Perfil } from './src/pages/perfil';
import { Auth } from './src/pages/auth';
import Criarconta from './src/pages/criarconta';
import { RecuperarConta } from './src/pages/recuperarconta';
import AsyncStorage from '@react-native-async-storage/async-storage';
import { signInWithEmailAndPassword, getAuth } from 'firebase/auth';
// ... Outros imports (fonts, icons) ...
import { Ionicons } from '@expo/vector-icons';

const Stack = createNativeStackNavigator();
const Tab = createBottomTabNavigator();

```

```

// Define a pilha de navegação para telas de autenticação
export function AuthStack() {
  return (
    <Stack.Navigator>
      <Stack.Screen name="auth" component={Auth} options={{ headerShown:
false }} />
      <Stack.Screen name="criarconta" component={Criarconta} options={{
headerShown: false }} />
      <Stack.Screen name="recuperarconta" component={RecuperarConta}
options={{ headerShown: false }} />
    </Stack.Navigator>
  );
}

```

```

// Define a navegação principal (abas) após o login
export function MainTabNavigator() {
  return (
    <Tab.Navigator>
      <Tab.Screen
        name="home"
        component={Home}
        options={{
          tabBarShowLabel: false,
          headerShown: false,
          tabBarIcon: ({ focused, size, color }) => (
            focused ? <Ionicons size={size} color={color} name="home" /> :
<Ionicons size={size} color={color} name="home-outline" />
          ),
        }}
      />
      <Tab.Screen
        name="relatorios"
        component={Relatorios}

```

```

    options={{
      tabBarShowLabel: false,
      headerShown: false,
      tabBarIcon: ({ focused, size, color }) => (
        focused ? <Icons size={size} color={color} name="bar-chart" /> :
<Icons size={size} color={color} name="bar-chart-outline" />
      ),
    }}
  />
<Tab.Screen
  name="configuracoes"
  component={Configuracoes}
  options={{ /* ... opções de ícone ... */ }}
/>
<Tab.Screen
  name="perfil"
  component={Perfil}
  options={{ /* ... opções de ícone ... */ }}
/>
</Tab.Navigator>
);
}

```

```

export default function App() {
  const [isLoggedIn, setIsLoggedIn] = useState(false);

  // Verifica se o usuário possui credenciais salvas no dispositivo
  useEffect(() => {
    const checkSavedCredentials = async () => {
      try {
        const userEmail = await AsyncStorage.getItem('userEmail');
        const userPassword = await AsyncStorage.getItem('userPassword');
        if (userEmail && userPassword) {
          const auth = getAuth();

```

```

    signInWithEmailAndPassword(auth, userEmail, userPassword)
      .then((userCredential) => {
        setIsLogged(true); // Loga o usuário automaticamente
      })
      .catch((error) => {
        console.error(error);
      });
  }
} catch (error) {
  console.error(error);
}
};
checkSavedCredentials();
}, []);

// ... Carregamento de fontes omitido ...
const [fontLoaded] = useFonts({ /* ... fontes ... */ });
if(!fontLoaded) { return null; }

// Renderiza a pilha de Autenticação ou a Navegação Principal
return (
  <NavigationContainer>
    <StatusBar backgroundColor="#000000" barStyle="light-content" />
    {isLogged ? <MainTabNavigator /> : <AuthStack />}
  </NavigationContainer>
);
}

```

B.4 TELA PRINCIPAL E LISTENER DE DADOS (SRC/PAGES/HOME.JS)

Este componente é o núcleo do aplicativo, responsável por "escutar" (onValue) as alterações de dados no Firebase em tempo real, exibir os valores de consumo e gerenciar o envio de notificações (expo-notifications) ao usuário.

```

import React, { useState, useEffect } from 'react';
import { View, Text, ActivityIndicator, Alert } from 'react-native';
import { getDatabase, ref, onValue, set } from "firebase/database";
import { styles } from './styles';
import * as Notifications from "expo-notifications";
import AsyncStorage from '@react-native-async-storage/async-storage';

export function Home() {

  // Busca o e-mail do usuário salvo localmente
  useEffect(() => {
    const fetchData = async () => {
      try {
        const Email = await AsyncStorage.getItem('userEmail');
        setUserEmail(Email);
      } catch (error) {
        console.error('Erro ao buscar dados:', error.message);
      }
    };
    fetchData();
  }, []);

  // Estados para armazenar os dados do Firebase
  const [faturaAtual, setFaturaAtual] = useState('Carregando...');
  const [proxFatura, setProxFatura] = useState('Carregando...');
  const [potTotal, setPotTotal] = useState('Carregando...');
  const [precoDesejado, setPrecoDesejado] = useState('Carregando...');
  const [notificacaoLocal, setNotificacaoLocal] = useState(0);
  const [userEmail, setUserEmail] = useState("");
  const [correnteAtual, setCorrenteAtual] = useState('Carregando...');
  const [tensaoAtual, setTensaoAtual] = useState('Carregando...');

  const db = getDatabase();
  const emailEditado = userEmail.replace(/./g, '_');

```

```

    // Referências (paths) para os nós no Firebase
    const fatura_atual = ref(db,
`/users/${emailEditado}/dados/home/Fatura_atual`);
    const prox_fatura = ref(db,
`/users/${emailEditado}/dados/home/Proxima_fatura`);
    const pot_total = ref(db,
`/users/${emailEditado}/dados/home/Potencia_total`);
    const preco_desejado = ref(db,
`/users/${emailEditado}/dados/configuracoes/Preco_desejado`);
    const notificacao_desejada = ref(db,
`/users/${emailEditado}/dados/configuracoes/notificacao`);
    const corrente_Atual =
ref(db,`/users/${emailEditado}/dados/home/Corrente`);
    const tensao_Atual = ref(db,`/users/${emailEditado}/dados/home/Tensao`)

    // Configura o handler de notificação
    Notifications.setNotificationHandler({
    handleNotification: async () => ({
    shouldShowAlert: true,
    shouldPlaySound: true,
    shouldSetBadge: true,
    }),
    });

    // Hook principal para "escutar" os dados do Firebase
    useEffect(() => {
    const fetchData = async () => {
    // Listener para Fatura Atual
    onValue(fatura_atual, (snapshot) => {
    const data = snapshot.val();
    const valorFormatado = parseFloat(data).toFixed(2).replace(/\./, ',');
    setFaturaAtual(valorFormatado);
    });
    });
  }

```

```

// Listener para Próxima Fatura
onValue(prox_fatura, (snapshot) => {
  const data = snapshot.val();
  const valorFormatado = parseFloat(data).toFixed(2).replace(/\./, ',');
  setProxFatura(valorFormatado);
});

// ... Outros listeners (pot_total, corrente_Atual, tensao_Atual) omitidos por
brevidade ...

onValue(preco_desejado, (snapshot) => {
  setPrecoDesejado(snapshot.val());
});

onValue(notificacao_desejada, (snapshot) => {
  setNotificacaoLocal(snapshot.val());
});

// Lógica de verificação para enviar notificação
if (parseFloat(faturaAtual) >= parseFloat(precoDesejado) * 0.9 &&
notificacaoLocal === 0) {
  sendNotification();
  EscreverNotificacao(); // Marca que a notificação foi enviada
}
};

fetchData();
}, );

// Função que agenda a notificação local
const sendNotification = async () => {
  await Notifications.scheduleNotificationAsync({
    content: {

```

```

    title: 'Atenção!',
    body: 'Sua fatura atual está próxima ao valor desejado!',
  },
  trigger: null, // Envia imediatamente
});
};

// Função que escreve no Firebase para não notificar novamente
const EscreverNotificacao = () => {
  set(notificacao_desejada, 1) // Salva o estado "notificado" (1)
  .then(() => {
    console.log('Notificação salva com sucesso.');
```

```
  })
```

```
  .catch((error) => {
```

```
    console.error('Erro ao salvar a notificação:', error.message);
```

```
  });
```

```
};
```

```
// A renderização da interface (JSX) foi omitida deste apêndice.
```

```
return (
```

```
  // [ ... UI (JSX) da tela Home omitido ... ]
```

```
);
```

```
}
```

B.5 TELA DE RELATÓRIOS E GRÁFICOS (SRC/PAGES/RELATORIOS.JS)

Este componente busca os dados históricos (diários, semanais e mensais) do Firebase e os formata para exibição na biblioteca de gráficos (react-native-gifted-charts), permitindo a visualização do consumo ao longo do tempo.

```

import React, {useState, useEffect} from 'react';
import { View, Text, Pressable, Modal, ScrollView } from 'react-native';
import { LineChart } from "react-native-gifted-charts";
import { getDatabase, ref, onValue } from "firebase/database";
import { styles } from '../styles'
```

```

import AsyncStorage from '@react-native-async-storage/async-storage';

export function Relatorios() {

  // ... Hook para buscar userEmail do AsyncStorage omitido por brevidade ...
  useEffect(() => { /* ... */ }, []);

  // Estados para os dados dos gráficos e faturas
  const [faturaUltimoMes, setUltimoMes] = useState("");
  // ... Outros estados de fatura (penultimo, antepenultimo) ...
  const [userEmail, setUserEmail] = useState("");
  const [dom, setDom] = useState(0);
  const [seg, setSeg] = useState(0);
  // ... Outros estados de dias (ter, qua, etc.) ...
  const [viewMode, setViewMode] = useState('semanal');
  const [h00, set00] = useState(0);
  const [h03, set03] = useState(0);
  // ... Outros estados de horas (h06, h09, etc.) ...

  const db = getDatabase();
  const emailEditado = userEmail.replace(/./g, '_');

  // Referências (paths) para os dados de relatórios no Firebase
  const fatura_ultimo_mes = ref(db,
`/users/${emailEditado}/dados/relatorios/Mensal/Gasto_ultimo_Mes`);
  // ... Outras referências de faturas ...
  const Domingo = ref(db,
`/users/${emailEditado}/dados/relatorios/Semanal/Dom`);
  // ... Outras referências de dias ...
  const d00h = ref(db, `/users/${emailEditado}/dados/relatorios/Diario/d0h`);
  // ... Outras referências de horas ...

  // Hook para "escutar" todos os dados de relatórios
  useEffect(() => {

```

```

// Listener para faturas mensais
const unsubscribe1 = onValue(fatura_ultimo_mes, (snapshot) => {
  const data1 = snapshot.val();
  const valorFormatado = parseFloat(data1).toFixed(2).replace(/\./, ',');
  setUltimoMes(valorFormatado);
});
// ... Outros listeners de faturas (unsubscribe2, 3, 4, 5, 6) ...

// Listeners para o gráfico semanal
const unsubsweek1 = onValue(Domingo, (snapshot) => {
setDom(snapshot.val()); });
const unsubsweek2 = onValue(Segunda, (snapshot) => {
setSeg(snapshot.val()); });
// ... Outros listeners semanais (unsubweek3 a 7) ...

// Listeners para o gráfico diário
const unsubsday0 = onValue(d00h, (snapshot) => { set00(snapshot.val());
});
const unsubsday3 = onValue(d03h, (snapshot) => { set03(snapshot.val());
});
// ... Outros listeners diários (unsubday6 a 2359) ...

// Função de limpeza (remove os listeners)
return () => {
  unsubscribe1();
  // ... Outros unsubscribes ...
  unsubsweek1();
  // ... Outros unsubscribes ...
  unsubsday0();
  // ... Outros unsubscribes ...
}
}, );

// Formatação dos dados para a biblioteca de gráfico (Semanal)

```

```

const data=[
  { value: dom, label: 'Dom', /* ... */ },
  { value: seg, label: 'Seg', /* ... */ },
  { value: ter, label: 'Ter', /* ... */ },
  { value: qua, label: 'Qua', /* ... */ },
  { value: qui, label: 'Qui', /* ... */ },
  { value: sex, label: 'Sex', /* ... */ },
  { value: sab, label: 'Sab', /* ... */ },
]

// Formatação dos dados para a biblioteca de gráfico (Diário)
const diario=[
  { value: h00, label: '0h', /* ... */ },
  { value: h03, label: '3h', /* ... */ },
  { value: h06, label: '6h', /* ... */ },
  { value: h09, label: '9h', /* ... */ },
  { value: h12, label: '12h', /* ... */ },
  { value: h15, label: '15h', /* ... */ },
  { value: h18, label: '18h', /* ... */ },
  { value: h21, label: '21h', /* ... */ },
  { value: h2359, label: '23h59', /* ... */ },
]

// A renderização da interface (JSX) foi omitida deste apêndice.
return (
  // [ ... UI (JSX) da tela Relatórios, incluindo o Modal e o LineChart, omitido
... ]
);
}

```

B.6 TELA DE CONFIGURAÇÕES (SRC/PAGES/CONFIGURACOES.JS)

Este componente demonstra a lógica de escrita de dados. Ele permite ao

usuário definir um `Preco_desejado` (meta de fatura), que é então salvo no Firebase usando a função `set()`.

```
import React, { useState, useEffect } from 'react';
import { View, Text, TouchableOpacity, TextInput } from 'react-native';
import { FontAwesome } from '@expo/vector-icons';
import { getDatabase, ref, set, onValue } from "firebase/database";
import { styles } from './styles';
import AsyncStorage from '@react-native-async-storage/async-storage';

export function Configuracoes() {

  // ... Hook para buscar userEmail do AsyncStorage omitido por brevidade ...
  useEffect(() => { /* ... */ }, );

  const [userEmail, setUserEmail] = useState("");
  const [precoDesejado, setPrecoDesejado] = useState(""); // Valor do input
  const [preco_Desejado, setPreco_Desejado] = useState(""); // Valor do DB
  const [bandeira, setBandeira] = useState("");

  const db = getDatabase();
  const emailEditado = userEmail.replace(/./g, '_');

  // Referências (paths) para os nós no Firebase
  const precoRef = ref(db,
`/users/${emailEditado}/dados/configuracoes/Preco_desejado`);
  const bandeiraRef = ref(db, `/configuracoes/bandeira`);
  const notificacao_desejada = ref(db,
`/users/${emailEditado}/dados/configuracoes/notificacao`);

  // Função que escreve o novo preço no Firebase
  const handleSalvarPreco = () => {
    const precoNumerico = parseFloat(precoDesejado);
    if (!isNaN(precoNumerico)) {
```

```
const precoReff = ref(db,
`/users/${emailEditado}/dados/configuracoes/Preco_desejado`);
// Salva o novo preço no banco de dados
set(precoReff, precoNumerico)
  .then(() => {
    console.log('Preço salvo com sucesso!');
  })
  .catch((error) => {
    console.error('Erro ao salvar o preço:', error.message);
  });
}
// Reseta o status da notificação para permitir novos alertas
set(notificacao_desejada, 0);
</a>
```

ANEXO A – ESPECIFICAÇÕES TÉCNICAS DO ALICATE AMPERÍMETRO HIKARI HA-3120

HIKARI
FERRAMENTAS

FICHA TÉCNICA

ALICATE AMPERÍMETRO DIGITAL HA-3120


CÓD. UCB: 21N239

O Alicate Amperímetro Digital modelo HA-3120 se destaca por ser True RMS, pelas medidas de Tensão DC/AC, Corrente AC, Resistência e pelo teste de Bateria, NCV, Diodo e Continuidade. Como característica adicional apresenta as funções: Lanterna, Iluminação no Display, Auto Power Off, Auto Range, Data Hold, Máx/Min e Indicador de Bateria Fraca.

O Alicate Amperímetro HA-3120 pode ser utilizado para medir circuitos eletrônicos e ramificações de rede elétrica.



Especificações Gerais

- Contagem Máxima do Display: 2000 contagens (3 ½ Dígitos);
- Iluminação do Display;
- Taxa de Amostragem: aproximadamente 3 vezes por segundo;
- Indicação de Polaridade: Automática;
- Indicação de Sobrefaixa: "OL" é mostrado no display;
- Indicação de bateria fraca:  é mostrado no display;
- Mudança de Faixa: Automática/ Manual;
- True RMS;
- Detecção de tensão sem contato NCV;
- Data Hold;
- Função Máximo/ Mínimo;
- Auto Power Off: Aproximadamente 15 minutos;
- Lanterna;
- Abertura da Garra: 28mm;
- Ambiente de Operação: 0°C a 40°C, RH<80%;
- Ambiente de Armazenamento: -10°C a 50°C, RH<80%;
- Segurança / Conformidade: IEC/EM 61010-1 Sobretensão e Dupla Isolação, CAT II 600V;
- Grau de poluição: 2;
- Tipo de Alimentação: 2 x 1,5V (Pilhas AAA);
- Dimensões: 185(A) x 45(L) x 24(P)mm;
- Peso: Aproximadamente 180g (incluindo bateria);

www.hikariferramentas.com.br

Tel: (11) 5070-1700

E-mail: hikari@unicoba.net

Especificações Técnicas

Tensão DC

Faixa: 200mV, 2V, 20V, 200V e 600V
 Resolução: 0,1mV, 1mV, 10mV, 100mV e 1V
 Precisão: 200mV~200V $\pm(0.5\% + 5 \text{ Dígitos})$
 600V $\pm(1.0\% + 5 \text{ Dígitos})$

- Impedância de Entrada: na faixa de 200mV>40M Ω , nas demais faixas é de 10M Ω .
- Proteção de Sobrecarga: 1000V DC/750V RMS.

Tensão AC

Faixa: 2V, 20V, 200V e 600V
 Resolução: 1mV, 10mV, 100mV e 1V
 Precisão: 2V~200V $\pm(1.0\% + 5 \text{ Dígitos})$
 600V $\pm(1.2\% + 5 \text{ Dígitos})$

- Impedância de Entrada: 10M Ω .
- Proteção de Sobrecarga: 1000V DC/750V RMS.
- Resposta em Frequência: na faixa de 600V: 40Hz~100Hz, nas demais faixas: 40Hz~400Hz.

Corrente AC

Faixa: 2A, 20A, 200A e 600A
 Resolução: 1mA, 10mA, 100mA e 1A
 Precisão: 2A $\pm(3.0\% + 5 \text{ Dígitos})$
 200A~600A $\pm(2.0\% + 5 \text{ Dígitos})$

- Proteção de Sobrecarga: 600A, tempo de entrada <1 minuto.
- Resposta em Frequência: Para onda Senoidal e Onda Triangular 40Hz~1kHz, para as demais formas de onda 40Hz~200Hz.

Resistência

Faixa: 200 Ω , 2k Ω , 20k Ω , 200k Ω , 2M Ω e 20M Ω
 Resolução: 0.1 Ω , 1 Ω , 10 Ω , 100 Ω , 1k Ω e 10k Ω
 Precisão: 200 Ω $\pm(0.8\% + 5 \text{ Dígitos})$
 2k Ω ~2M Ω $\pm(0.8\% + 1 \text{ Dígitos})$
 20M Ω $\pm(1.2\% + 5 \text{ Dígitos})$

- Proteção de Sobrecarga: 250Vrms em 15 segundos no máximo.
- Tensão Máxima de Circuito Aberto: 500mV.

Continuidade

Resolução: 0.1 Ω

Descrição: A buzina toca se a resistência medida for menor que 50 Ω .

- Proteção de Sobrecarga: 250Vrms em 15 segundos no máximo.
- Tensão Máxima de Circuito Aberto: 2V.

Diodo

Resolução: 1mV

Descrição: O display exibe o valor da queda de tensão aproximada do diodo

- Proteção de Sobrecarga: 250Vrms em 15 segundos no máximo.



FICHA TÉCNICA

- A corrente DC é de aproximadamente 0.8mA.
- A tensão reversa é de aproximadamente 2,2V.

Itens Inclusos

- Manual de Instruções
- Bolsa de Transporte
- Ponta de Prova (1 par)
- Pilha 1,5V AAA (2 peças)

Acessórios Opcionais / Reposição

- Ponta de Prova HK-3 (21N022)

Garantia

12 meses (consultar o manual do produto).

*Foto meramente ilustrativa. Especificações sujeitas a alterações sem aviso prévio.

www.hikariferramentas.com.br

Tel: (11) 5070-1700

E-mail: hikari@unicoba.net