



UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”
Instituto de Ciência e Tecnologia
Câmpus de Sorocaba

DANIEL FRANCISCO DA SILVA

PROJETO DE APLICATIVO PARA ANÁLISE E GESTÃO DE DADOS DE ENERGIA
ELÉTRICA DE MULTIMEDIDOR COM CONEXÃO MODBUS TCP/IP

Sorocaba

2025

DANIEL FRANCISCO DA SILVA

PROJETO DE APLICATIVO PARA ANÁLISE E GESTÃO DE DADOS DE ENERGIA
ELÉTRICA DE MULTIMEDIDOR COM CONEXÃO MODBUS TCP/IP

Trabalho de Conclusão de Curso apresentado à
Universidade Estadual Paulista (UNESP),
Instituto de Ciência e Tecnologia, Sorocaba,
como parte dos requisitos para obtenção do
grau de Bacharel(a) em Engenharia de Controle
e Automação.

Orientador(a): Prof. Dr. Eduardo Verri
Liberado

Sorocaba

2025

S586p

Silva, Daniel Francisco da

Projeto de aplicativo para análise e gestão de dados de energia elétrica de multimedidor com conexão modbus TCP/IP / Daniel Francisco da Silva. -- Sorocaba, 2025

102 p. : il., tabs., fotos

Trabalho de conclusão de curso (Bacharelado - Engenharia de Controle e Automação) - Universidade Estadual Paulista (UNESP), Instituto de Ciência e Tecnologia, Sorocaba

Orientador: Eduardo Verri Liberado

1. Software de aplicação. 2. Redes elétricas - Analisadores. 3. Automação industrial - Protocolos. 4. Aquisição de dados. 5. SQL (Linguagem de programação de computador). I. Título.

DANIEL FRANCISCO DA SILVA

PROJETO DE APLICATIVO PARA ANÁLISE E GESTÃO DE DADOS DE ENERGIA
ELÉTRICA DE MULTIMEDIDOR COM CONEXÃO MODBUS TCP/IP

Trabalho de Conclusão de Curso apresentado ao Instituto de Ciência e Tecnologia de Sorocaba, Universidade Estadual Paulista (UNESP), como parte dos requisitos para obtenção do grau de Bacharel(a) em Engenharia de Controle e Automação.

Data da defesa: 24/10/2025

BANCA EXAMINADORA:

Prof. Dr. Eduardo Verri Liberado

Orientador/UNESP – Instituto de Ciência e Tecnologia – Campus de Sorocaba

Prof. Dr. Luis Armando De Oro Arenas

UNESP – Instituto de Ciência e Tecnologia – Campus de Sorocaba

Me. Bruno Oliveira Zarpellon

UNESP – Instituto de Ciência e Tecnologia – Campus de Sorocaba

AGRADECIMENTOS

Chegar até aqui foi uma jornada marcada por desafios, aprendizados e, acima de tudo, por pessoas que fizeram toda a diferença. Por isso, quero expressar minha profunda gratidão a todos que estiveram ao meu lado nesse processo.

Primeiramente, agradeço ao meu pai, Manoel Francisco da Silva, e à minha mãe, Lindinalva Maria da Silva, por serem minha base, meu alicerce e minha inspiração. O apoio incondicional, os conselhos e o amor que recebi de vocês foram fundamentais para que eu pudesse seguir firme em cada etapa dessa caminhada.

Ao meu professor orientador, Eduardo Verri Liberado, minha eterna gratidão pela paciência e dedicação para me guiar neste projeto.

Ao meu amigo Giovanni Dotta Briganti, obrigado pela parceria e consideração. Sua amizade foi um presente nesse percurso que levarei para sempre.

Agradeço também à Eduarda Hadassa e ao Guilherme Folchetti, com quem tive o privilégio de estagiar. Compartilhar experiências com vocês foi enriquecedor e me fez crescer não só profissionalmente, mas também como pessoa.

A todos vocês e a comunidade da UNESP, meu sincero obrigado. Este trabalho é também fruto da presença e do apoio de cada um.

RESUMO

Este trabalho apresenta o desenvolvimento de um aplicativo para análise e gestão de dados de energia elétrica utilizando comunicação Modbus TCP/IP com multimedidores industriais. O projeto foi desenvolvido utilizando o framework Flutter com linguagem Dart, integrando funcionalidades de monitoramento em tempo real, análise de qualidade de energia e geração automatizada de relatórios. A metodologia adotada incluiu a criação de uma interface intuitiva baseada em protótipo Figma, implementação de comunicação Modbus TCP para aquisição de dados de grandezas elétricas, desenvolvimento de algoritmos de análise conforme normas PRODIST e IEEE, e integração com Firebase para armazenamento em nuvem e gestão de usuários. Para validação funcional, foram utilizados dados simulados via microcontrolador ESP32 e dados reais do projeto smart campus Unesp, abrangendo medições de tensão, corrente, fator de potência, distorção harmônica total e frequência. O sistema implementa análise estatística de 1.008 amostras por semana, cálculo de indicadores DRP, DRC, DTT95% e FD95%, além de funcionalidades de exportação para PDF e Excel. Os resultados demonstraram eficácia na identificação de padrões de qualidade energética, com capacidade de detectar desequilíbrios, distorções harmônicas e variações de tensão em diferentes setores de consumo. A solução desenvolvida oferece uma ferramenta robusta para supervisão energética industrial, contribuindo para otimização da eficiência energética e conformidade regulatória.

Palavras-chave: qualidade de energia; Modbus TCP; Flutter; análise energética; supervisão industrial.

ABSTRACT

This work presents the development of an application for electrical energy data analysis and management using Modbus TCP/IP communication with industrial multimeters. The project was developed using the Flutter framework with Dart language, integrating real-time monitoring functionalities, power quality analysis, and automated report generation. The adopted methodology included creating an intuitive interface based on Figma prototype, implementing Modbus TCP communication for electrical quantities data acquisition, developing analysis algorithms according to PRODIST and IEEE standards, and Firebase integration for cloud storage and user management. For functional validation, simulated data via ESP32 microcontroller and real data from UNESP Smart Campus project were used, covering voltage, current, power factor, total harmonic distortion, and frequency measurements. The system implements statistical analysis of 1,008 monthly samples, calculation of DRP, DRC, DTT95%, and FD95% indicators, plus PDF and Excel export functionalities. Results demonstrated effectiveness in identifying energy quality patterns, with capability to detect imbalances, harmonic distortions, and voltage variations across different consumption sectors. The developed solution provides a robust tool for industrial energy supervision, contributing to energy efficiency optimization and regulatory compliance.

Keywords: power quality; Modbus TCP; Flutter; energy analysis; industrial supervision.

LISTA DE ILUSTRAÇÕES

Figura 1 - Painel elétrico com multimedidor	11
Figura 2 - Tela do banco de dados com área útil destacada	17
Figura 3- Aquisição de dados por um painel conectado a sensores	23
Figura 4 - Diagrama de tráfego de dados	27
Figura 5 - Faixas de tensão de fase ideal, precária e crítica	30
Figura 6 - Representação dos componentes do fasor de tensão	36
Figura 7 - Página de configuração de alarmes.....	41
Figura 8 - Interface web do firebase	51
Figura 9 - Gráfico de consumo mensal.....	53
Figura 10 - Gráfico de tensão de linha	54
Figura 11 - Gráfico de corrente	54
Figura 12 - Gráfico de alarmes	55
Figura 13 - Página de configuração de conexão Modbus.....	56
Figura 14 - Exemplo de leitura dos registradores do multimedidor	58
Figura 15 - Mostradores dinâmicos de dados	59
Figura 16 - Comparativo de amostras de fator de distorção do smart campus Unesp	64
Figura 17 - Amostras precárias do smart campus Unesp	65
Figura 18 - Comparativo de amostras de distorção harmônica do smart campus Unesp.....	65
Figura 19 - Comparativo de amostras de fator de potência do smart campus Unesp.....	66

LISTA DE TABELAS

Tabela 1 - Limites de distorção harmônica DTT95%	33
Tabela 2 - Limite do fator distorção de tensão FD95%.....	34
Tabela 3 - Faixas de tensão de fase simuladas	60
Tabela 4 - Faixa de tensão de linha simuladas	60

SUMÁRIO

1 INTRODUÇÃO	11
1.1 Objetivos	13
2 CARACTERÍSTICAS DO PROJETO	14
2.1 Requisitos de projeto	14
2.2 Método de criação das páginas	15
2.3 Disposição do Menu Lateral.....	16
2.4 Área Útil do Aplicativo	17
3 REVISÃO BIBLIOGRÁFICA	18
3.1 Qualidade de energia.....	18
3.1.1 Principais Fatores de Degradação da Qualidade	19
3.1.2 Fundamentos e Definições	19
3.1.4 Monitoramento e Medição	21
3.1.5 Sistemas de Monitoramento Modernos	22
3.2 Supervisório	22
3.2.1 Protocolo Modbus TCP/IP.....	23
3.2.2 Formato da Mensagem e Cabeçalho MBAP	24
3.3 Flutter e Dart	25
4 METODOLOGIA	26
4.1 Interface Modbus TCP.....	26
4.1.1 Coleta das Leituras pelo Multimedidor	27
4.3 Armazenamento no Banco de Dados SQL Local.....	28
4.4 Análise de Qualidade da Energia	28
4.4.1 Variação de tensão em regime permanente	29
4.4.2 Critérios e Indicadores.....	29

4.4.4 Distorção harmônica.....	32
4.4.5 Desequilíbrio de tensão.....	34
4.4.6 Critério FD95%	34
4.4.7 Metodologia de Cálculo de Tensões de Linha usando Transformação Polar- Retangular	35
4.4.7.1 Fundamentos Teóricos da Representação Fasorial.....	36
4.4.7.2 Transformações entre Sistemas de Coordenadas.....	36
4.4.7.3 Cálculo de Tensões de Linha.....	37
4.4.8 Critérios de Avaliação da Frequência	37
4.5 Método de construção do servidor Firebase	38
4.6 Sistema de alarmes	40
4.6.1 Componentes Principais	41
4.6.2 Tipos de Alarmes Implementados	42
4.7 Gestão de usuários.....	42
4.8 Validação do Projeto	43
4.9 Teste com ESP32 e endereçamento modbus	44
5 RESULTADOS.....	50
5.1 População e visualização do banco de dados	50
5.2 Interação e construção das páginas	51
5.2.1 Padrão de Gerenciamento de Estado	52
5.3 Exibição dos gráficos.....	53
5.4 Conexão Modbus	55
5.4.1 Parâmetros de conexão	55
5.4.2 Teste funcional	57
5.5 Análise de qualidade.....	59
5.5.1 Análise de dados simulados	60

5.5.2 <i>Análise de dados reais</i>	61
5.5.2.1 <i>Metodologia de Coleta e Estruturação dos Dados</i>	61
5.5.2.2 <i>Parâmetros Analisados e Limitações</i>	62
5.5.2.3 <i>Processamento e Análise Estatística</i>	63
5.5.2.4 <i>Resultados da Análise</i>	63
5.5.2.5 <i>Fator de distorção</i>	64
5.5.2.6 <i>DRP - Duração Relativa de Transgressão de Tensão Precária</i>	64
5.5.2.7 <i>Distorção harmônica total</i>	65
5.5.2.8 <i>Comparativo do fator de potência</i>	66
6 CONCLUSÃO.....	68
REFERÊNCIAS	70
APÊNDICE A – Tabela de resultado da análise de dados simulados para 1008 amostras	72
APÊNDICE B – PDF de representação do relatório de análise de qualidade.....	73
APÊNDICE C – Link de download dos relatórios mensais.....	75
APÊNDICE D – Link de download do projeto dart	76
APÊNDICE E – Tela de login	77
APÊNDICE F – Tela inicial	78
APÊNDICE G – Página de monitoramento	79
APÊNDICE H – Página de gráficos	80
APÊNDICE I – Página de análise de qualidade	81
APÊNDICE J – Página de Banco de dados de alarmes.....	82
APÊNDICE K – Página de configuração – Firebase.....	83
APÊNDICE L – Página de configuração – usuário.....	84
APÊNDICE M - Código ESP32 de simulação de Multimedidor ABB M1M Modbus TCP	85
APÊNDICE N – Grupo dos registradores do aplicativo	96
APÊNDICE O – Faixas de classificação de qualidade da tensão de fase	102

1 INTRODUÇÃO

O crescimento da demanda por energia elétrica, aliado à necessidade de otimização de custos operacionais e redução de perdas energéticas, tem impulsionado o desenvolvimento de sistemas de monitoramento e análise da qualidade de energia elétrica. Segundo Dugan (2002), distúrbios na qualidade da energia podem representar perdas econômicas significativas para as indústrias, variando desde paradas não programadas até danos em equipamentos sensíveis.

Para um monitoramento eficiente, multimedidores como o da Figura 1 devem ser utilizados, integrados a sistemas de aquisição e análise que assegurem a coleta contínua e a interpretação dos principais indicadores de qualidade de energia.

Figura 1 - Painel elétrico com multimedidor



Fonte: Voltimum Brasil

A implementação de sistemas de supervisão energética baseados em protocolos de comunicação industrial, como o Modbus TCP/IP, tem se mostrado uma solução eficaz para o monitoramento contínuo de parâmetros elétricos essenciais. Conforme destacado por Clarke et al. (2004), o protocolo Modbus TCP oferece vantagens significativas em termos de simplicidade de implementação e interoperabilidade com equipamentos de diferentes fabricantes, características fundamentais para aplicações industriais.

No contexto da qualidade de energia elétrica, Ribeiro et al. (1997) ressaltam que "sistemas de monitoramento adequados são fundamentais para identificação rápida de

distúrbios e implementação de ações corretivas eficazes". Esta perspectiva reforça a importância, especialmente no contexto industrial, de soluções tecnológicas integradas que não apenas realizem a coleta de dados, mas também possibilitem sua análise e interpretação em tempo real. Em ambientes produtivos, onde pequenas falhas podem gerar grandes prejuízos, a capacidade de transformar medições em informações úteis é decisiva para a tomada de decisão rápida e eficaz. Nesse sentido, a integração entre sistemas de monitoramento e aplicativos de análise de dados surge como um elo fundamental, permitindo que os dados coletados sejam convertidos em diagnósticos e relatórios inteligentes.

No contexto do desenvolvimento de aplicações para análise de dados energéticos, o framework Flutter tem emergido como uma alternativa robusta para a criação de interfaces multiplataforma. Zammetti (2019) destaca que o Flutter, utilizando a linguagem Dart, permite o desenvolvimento de aplicações nativas de alta performance a partir de uma única base de código, reduzindo significativamente os custos de desenvolvimento e manutenção.

A integração entre sistemas de aquisição de dados industriais e plataformas de análise em tempo real representa um desafio técnico que demanda soluções arquiteturais adequadas. Kopec (2019) evidencia que a linguagem Dart, através de seu sistema de compilação *ahead-of-time (AOT)* — um processo no qual o código-fonte é convertido em código de máquina nativo antes da execução, reduzindo a necessidade de interpretações em tempo de execução e aumentando a performance — oferece características apropriadas para aplicações que necessitam processar grandes volumes de dados em tempo real.

A análise da qualidade de energia elétrica, fundamentada nas diretrizes estabelecidas pela IEEE Std 1159 (2019) e pelos Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional (PRODIST) da ANEEL (2021), requer ferramentas computacionais capazes de processar e interpretar indicadores como distorção harmônica total, desequilíbrio de tensão, variações temporárias de tensão e fator de potência. A figura 1 demonstra um painel com multimedidor:

Neste contexto, o presente trabalho propõe o desenvolvimento de um aplicativo integrado para análise e gestão de dados de energia elétrica, utilizando comunicação Modbus TCP/IP com multimedidores industriais. A solução visa proporcionar uma ferramenta completa para o monitoramento em tempo real, análise histórica da qualidade energética e geração

automatizada de relatórios técnicos, contribuindo para a otimização do desempenho energético em ambientes industriais.

1.1 Objetivos

O principal objetivo deste projeto é desenvolver um aplicativo voltado para a análise e gerenciamento de dados de energia elétrica, com base na leitura de informações obtidas de multimedidores industriais através do protocolo Modbus TCP/IP. A proposta busca criar uma ferramenta que viabilize o monitoramento contínuo da qualidade e do consumo de energia elétrica, oferecendo ao usuário visualizações em tempo real, análises inteligentes e geração automatizada de relatórios adaptáveis. A solução deverá proporcionar uma interface intuitiva e funcional, capaz de acompanhar indicadores energéticos com precisão, identificar padrões de consumo e antecipar possíveis anomalias na rede elétrica.

2 CARACTERÍSTICAS DO PROJETO

A seção 2.1 apresenta de forma concisa os requisitos funcionais e técnicos que orientam o desenvolvimento do aplicativo de gerenciamento e análise da qualidade de energia elétrica. Esses requisitos definem a plataforma e linguagem de implementação, as capacidades de aquisição, visualização e armazenamento de dados, os recursos de análise e exportação, além das exigências de segurança e gestão de usuários, servindo como base para o projeto, implementação e verificação das funcionalidades descritas nas demais subseções e nos capítulos a seguir.

2.1 Requisitos de projeto

O aplicativo de gerenciamento de dados e análise da qualidade de energia elétrica deve atender a uma série de requisitos funcionais e técnicos para garantir sua eficiência e confiabilidade. A seguir, são descritos os principais requisitos do projeto:

- Linguagem e Framework
 - I. O aplicativo deve ser desenvolvido na linguagem Dart, utilizando o framework Flutter
- Aquisição e Exibição de Dados em Tempo Real
 - I. Leitura dos parâmetros elétricos, incluindo tensão, corrente, consumo, frequência e fator de potência, através do protocolo Modbus TCP.
 - II. Exibição dos dados coletados de forma clara e acessível na interface do aplicativo.
- Visualização Gráfica e Monitoramento de Alarmes
 - I. Geração de gráficos para análise de consumo, corrente e tensão, permitindo a interpretação dos dados ao longo do tempo.

Exibição de eventos críticos e alarmes, possibilitando o monitoramento de variações inesperadas na rede elétrica.
- Análise e Armazenamento de Dados
 - I. Capacidade de análise dos dados da rede elétrica a partir de registros armazenados em um banco de dados SQL.

- II. Função para salvar relatórios de análise em formato PDF, facilitando a documentação e compartilhamento dos resultados.
- III. Exibição dos registros do banco de dados em formato de tabela, proporcionando maior organização e acessibilidade.
- Exportação e Integração de Dados
 - I. Exportação do banco de dados para arquivos Excel, permitindo o processamento e consulta dos dados em outras plataformas.
 - II. Armazenamento seguro dos dados no servidor Firebase, garantindo sincronização em nuvem e acesso remoto.
- Gestão de Usuários e Acessibilidade
 - I. Implementação de sistema de login com autenticação segura para controle de acesso.
 - II. Possibilidade de adicionar novos usuários ao sistema, permitindo gestão de diferentes perfis e níveis de permissão

2.2 Método de criação das páginas

O uso do Figma para criar o protótipo de um aplicativo de monitoramento e análise de qualidade de energia é uma escolha estratégica a ser feita antes da implementação em Flutter/Dart, pois permite validar layouts, fluxos de navegação e usabilidade, definir estilos e componentes reutilizáveis.

A plataforma apresenta características fundamentais que justificam sua escolha para desenvolvimento de interfaces industriais. Conforme destacado pela Turbo Partners (2025), o Figma viabiliza "colaboração em tempo real, onde diversos membros da equipe podem trabalhar simultaneamente no mesmo projeto, visualizando as alterações em tempo real". Essa capacidade de colaboração se mostra essencial no desenvolvimento de aplicações industriais, onde equipes multidisciplinares de engenheiros, designers e desenvolvedores precisam trabalhar de forma integrada.

Adicionalmente, a natureza baseada em nuvem do Figma permite acesso aos projetos de qualquer lugar, sem necessidade de instalação de software adicional, enquanto suas

funcionalidades de prototipagem interativa possibilitam simular a experiência do usuário antes do desenvolvimento, reduzindo custos e tempo de iteração. A plataforma também oferece recursos para design responsivo, garantindo que a interface se adapte adequadamente a diferentes dispositivos e resoluções, característica fundamental para aplicações industriais que podem ser acessadas via tablets, computadores e dispositivos móveis. A tela inicial da aplicação, que sintetiza a organização estrutural e funcional do sistema, encontra-se apresentada no Apêndice F, servindo como referência para a compreensão da interface disponibilizada ao usuário.

2.3 Disposição do Menu Lateral

O menu lateral do aplicativo é fundamental para a navegação entre os diferentes módulos, garantindo acesso rápido e organizado às funcionalidades principais. A disposição das opções no menu permite ao usuário acessar as informações desejadas sem dificuldades. As seções incluídas são:

Monitoramento – Exibe em tempo real as principais variáveis do sistema elétrico, como tensão, consumo, corrente e potência, permitindo uma visão dinâmica do desempenho da rede.

Gráficos – Apresenta as curvas de tensão, corrente, consumo e alarmes, possibilitando uma análise detalhada dos parâmetros elétricos ao longo do tempo.

Análise de Qualidade – Traz insights sobre a qualidade da energia com base nos dados armazenados em um banco de dados SQLite, permitindo avaliações criteriosas de estabilidade e eficiência energética.

Banco de Dados – Exibe uma tabela com os dados armazenados no banco de dados SQL local, fornecendo um histórico detalhado para consultas e estudos técnicos.

Histórico de alarmes – Exibe uma tabela com o histórico de alarmes registrado pela aplicação.

Configuração – Oferece opções de ajuste para a conexão Modbus TCP, armazenamento via Firebase, gestão de usuários e definição de limiares de alarmes, permitindo personalização e controle do sistema.

2.4 Área Útil do Aplicativo

A área útil é o espaço central onde as informações são apresentadas ao usuário. No protótipo, o dashboard principal exibe gráficos detalhados sobre o consumo mensal, além de indicadores de estado do multimedidor e consumo instantâneo. Esta organização facilita a tomada de decisão e otimiza o acompanhamento dos dados.

O Figma se destaca por permitir que todas essas seções sejam representadas visualmente, facilitando ajustes na interface antes do desenvolvimento final. Isso garante que o aplicativo seja intuitivo e funcional, proporcionando uma experiência fluida ao usuário.

Esse modelo estruturado, baseado em componentes organizados, fortalece a eficiência do sistema, oferecendo uma ferramenta robusta para análise e gestão da qualidade de energia. A figura 2 apresenta a disposição da área de informação útil destacada.

Figura 2 - Tela do banco de dados com área útil destacada

Dashboard

Monitoramento

Gráficos

Análise de Qualidade

Banco de Dados

Histórico de Alarmes

Configuração

admin@admin.com

Logout

Banco de Dados

Exportar ExcelAtualizarLimpar Histórico

Total de registros: 187771 | Use os filtros nas colunas para refinar a busca

Data	Hora	Variável	Valor
26/2/2025	9:20:39	V1N	210.19
26/2/2025	9:20:39	V2N	225.19
26/2/2025	9:20:39	V3N	198.19
26/2/2025	9:20:39	Vinef	210.29
26/2/2025	9:20:39	V12	357.33
26/2/2025	9:20:39	V23	382.83
26/2/2025	9:20:39	V31	336.93
26/2/2025	9:20:39	Vinefmed	359.03
26/2/2025	9:20:39	L1	127.75

Fonte: Autoria própria

3 REVISÃO BIBLIOGRÁFICA

A revisão bibliográfica apresenta os fundamentos teóricos e tecnológicos que sustentam o desenvolvimento deste trabalho, abordando conceitos, ferramentas e normas aplicáveis ao monitoramento e análise da qualidade de energia elétrica. São discutidos aspectos relacionados a frameworks de desenvolvimento multiplataforma, protocolos de comunicação industrial e sistemas supervisórios, bem como diretrizes normativas que orientam a medição e interpretação de indicadores elétricos. Essa base teórica é essencial para compreender as soluções propostas e justificar as escolhas metodológicas adotadas.

3.1 Qualidade de energia

A qualidade de energia elétrica (QEE) constitui um campo de crescente importância na engenharia elétrica moderna, especialmente em ambientes industriais, comerciais e hospitalares, onde a presença de equipamentos sensíveis e sistemas automatizados exige um fornecimento de energia estável, contínuo e confiável. Pequenas variações ou distorções nos parâmetros elétricos podem comprometer o desempenho, reduzir a vida útil ou até provocar falhas críticas em processos produtivos e serviços essenciais.

Segundo Dugan (2002), a qualidade de energia refere-se à capacidade do sistema elétrico de fornecer energia dentro de parâmetros aceitáveis de tensão, corrente, frequência e forma de onda, assegurando o funcionamento adequado dos equipamentos conectados. Esses parâmetros devem permanecer dentro de limites estabelecidos por normas técnicas, como a IEEE Std 1159 (2019) e, no contexto brasileiro, pelo Módulo 8 do PRODIST da ANEEL, que definem critérios objetivos para avaliação e monitoramento da QEE.

Mendonça et al. (2020) destacam que a qualidade da energia na indústria é fundamental tanto para a eficiência quanto para a vida útil dos componentes e maquinários. Os autores enfatizam que "uma energia apropriada é um problema a menos a ser enfrentado por uma fábrica e seus colaboradores em sua rotina", demonstrando a importância prática da manutenção de padrões adequados de qualidade energética em ambientes industriais.

3.1.1 Principais Fatores de Degradação da Qualidade

Conforme identificado por Mendonça et al. (2020), os problemas com a qualidade da energia elétrica geralmente são provenientes de três fatores principais: tensões e correntes harmônicas, baixo fator de potência e instabilidade da tensão. Esta classificação é fundamental para o desenvolvimento de estratégias adequadas de monitoramento e mitigação.

Distorções Harmônicas: As distorções harmônicas, segundo Kagan et al. (2009), são definidas como "ondas não-senoidais de corrente e tensão provocadas por equipamentos cuja carga é não-linear". Equipamentos como inversores de frequência, quando conectados à rede, contribuem significativamente para a distorção harmônica, sendo simultaneamente sensíveis ao fenômeno. A presença de distorções harmônicas em um circuito é indesejável, pois reduz a vida útil dos equipamentos, gerando custos adicionais de manutenção, produção e reinvestimentos.

Baixo Fator de Potência: O fator de potência, conforme explicado por Mendonça et al. (2020), é uma medida de quanto da potência elétrica consumida está efetivamente sendo convertida em trabalho útil. A ANEEL, através do Módulo 8 do PRODIST, estabelece que o valor do fator de potência deve estar compreendido entre 0,92 e 1 indutivo ou 1 e 0,92 capacitivo. Valores inferiores a 0,92 habilitam a concessionária a cobrar multa na fatura de energia sobre o consumo de potência reativa além dos 8% máximos permitidos.

Instabilidade de Tensão: Mendonça et al. (2020) identificam a instabilidade de tensão como consequência de um fornecimento instável de níveis de eletricidade. A baixa tensão causa ineficiência no circuito e pode provocar quedas de energia, enquanto a elevação na tensão, além de prejudicar o desempenho atual de equipamentos, reduz sua longevidade.

3.1.2 Fundamentos e Definições

A IEEE Std 1159 (2019) estabelece definições padronizadas para os fenômenos de qualidade de energia elétrica, classificando-os em categorias como variações de tensão de longa duração, variações temporárias de curta duração, desequilíbrios, distorções de forma de onda, variações de frequência, transientes elétricos e flutuação de tensão. Esta norma serve como referência mundial para a caracterização e análise de distúrbios elétricos, proporcionando uma linguagem comum entre fabricantes, concessionárias e consumidores.

No contexto brasileiro, os Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional (PRODIST) da ANEEL (2021) estabelecem os critérios nacionais para avaliação da qualidade do produto, definindo limites e metodologias de medição adequados às características do sistema elétrico nacional. O Módulo 8 do PRODIST especifica indicadores como DRP (Duração Relativa de Transgressão de Tensão Precária) e DRC (Duração Relativa de Transgressão de Tensão Crítica), fundamentais para a avaliação da conformidade do fornecimento de energia.

A experiência da CEMIG relatada por Ribeiro et al. (1997) demonstra que "a definição clara de responsabilidades entre concessionária e consumidor é fundamental para a resolução eficaz de problemas de qualidade de energia". O estudo evidencia que protocolos bem estabelecidos de medição e análise contribuem significativamente para a identificação precisa das causas de distúrbios e implementação de soluções adequadas.

3.1.3 Estudos de Caso em Ambientes Industriais

Ribeiro et al. (1997) apresentam um estudo de caso abrangente sobre monitoramento e diagnóstico da qualidade de energia em uma unidade industrial da Nestlé atendida em tensão de distribuição pela CEMIG. O estudo evidencia que equipamentos industriais modernos, especialmente controladores lógicos programáveis (CLP) e acionamentos a velocidade variável (AVV), possuem sensibilidades específicas que variam conforme marca e modelo, sendo essencial conhecer essas características para implementação de sistemas de proteção adequados.

Os autores demonstram que falhas no sistema elétrico podem resultar em perdas significativas de produção, citando que no processo de produção de leite condensado, "quando ocorre uma falta no sistema, interrompendo o processo, o leite queima entre as tubulações e caldeiras da esterilização, resultando em perdas da produção e tempo gasto para limpeza, que dura de 6 a 7 horas". Além das perdas diretas, o estudo destaca que "algumas faltas também resultam em perdas de programação dos controles do processo, sendo necessário também a reprogramação e reajustes dos controladores lógico programáveis".

Uma contribuição importante do trabalho é a caracterização de sensibilidades típicas de equipamentos industriais: "valores típicos de sensibilidade para acionamentos são para afundamento de 80% a 72% de tensão para 0,05 s (3 ciclos) a 0,2 s (12,5 ciclos), e os controles

para 90% da tensão entre 0,25 (15,6 ciclos) a 0,35 s (21 ciclos) podem apresentar problemas". Estas informações são fundamentais para desenvolvimento de sistemas de monitoramento que considerem as especificidades de cada aplicação industrial.

O trabalho também destaca a importância da curva CBEMA (Computer Business Equipment Manufacturers Association) na ausência de limites específicos para ocorrências transitórias, demonstrando como ferramentas padronizadas podem ser aplicadas na análise de distúrbios industriais. A experiência relatada pelos autores enfatiza que "o monitoramento de qualidade é muito importante tanto para a concessionária quanto para o consumidor, no sentido que estabelece responsabilidades", aspecto fundamental para resolução colaborativa de problemas de qualidade de energia.

Adicionalmente, Ribeiro et al. (1997) destacam que "a colaboração técnica entre concessionária e consumidor industrial resulta em benefícios mútuos, incluindo melhor compreensão dos fenômenos, otimização de investimentos e redução de conflitos". Esta perspectiva colaborativa tem se mostrado essencial para o desenvolvimento de soluções eficazes de qualidade de energia no setor industrial brasileiro.

3.1.4 Monitoramento e Medição

O monitoramento contínuo da qualidade de energia constitui elemento fundamental para identificação, caracterização e mitigação de distúrbios. O Módulo 8 do PRODIST da ANEEL (2021) estabelece metodologias padronizadas para medição e análise, incluindo requisitos para instrumentação, procedimentos de medição e critérios de avaliação.

Estudos como o de Almeida (2017) ressaltam a importância da localização estratégica dos pontos de medição e da duração adequada das campanhas de monitoramento. Já a metodologia proposta por Melo (2008) para "avaliação do impacto da qualidade de energia elétrica na produção industrial" complementa essas normas ao introduzir indicadores econômicos específicos que quantificam as perdas financeiras associadas a distúrbios de qualidade. Esta abordagem é particularmente relevante para ambientes industriais, onde a correlação entre qualidade de energia e produtividade precisa ser estabelecida de forma quantitativa. Para uma caracterização estatística confiável, recomenda-se períodos mínimos de medição de sete dias consecutivos, com amostragem de pelo menos 1.008 amostras por variável

analisada. Além disso, conforme estabelecido pelo Módulo 8 do PRODIST da ANEEL, os medidores utilizados devem estar em conformidade com normas técnicas específicas, como a IEC 61000-4-30 ou ABNT NBR IEC 61000-4-30 a fim de garantir a precisão e a confiabilidade das medições.

3.1.5 Sistemas de Monitoramento Modernos

A evolução dos sistemas de monitoramento de qualidade de energia tem sido impulsionada pelo desenvolvimento de tecnologias digitais e protocolos de comunicação industrial. Ribeiro dos Santos et al. (2019) ressaltam que a integração entre multimedidores inteligentes e plataformas de análise computacional, aliada ao uso de protocolos como Modbus TCP/IP, possibilita o processamento em tempo real de grandes volumes de dados, favorecendo diagnósticos rápidos e ações corretivas eficientes.

Rocha et al. (2024) desenvolveram um protótipo de sistema industrial para monitoramento de parâmetros elétricos e qualidade do fornecimento de energia elétrica, utilizando o microcontrolador ESP32 e o circuito integrado ADE7758. Os autores destacam que "o sistema deve adquirir os parâmetros elétricos, processar esses dados e enviá-los para armazenamento, permitindo a geração de gráficos com os parâmetros coletados". O protótipo demonstrou eficácia na coleta de dados como tensão, corrente, fator de potência, potências ativa, reativa e aparente, apresentando erro médio de aproximadamente 1% quando comparado a um analisador comercial FLUKE 434.

A implementação de sistemas baseados em protocolos como Modbus TCP/IP permite a integração de multimedidores inteligentes com sistemas supervisórios, possibilitando monitoramento contínuo, análise automatizada e geração de relatórios de conformidade regulatória.

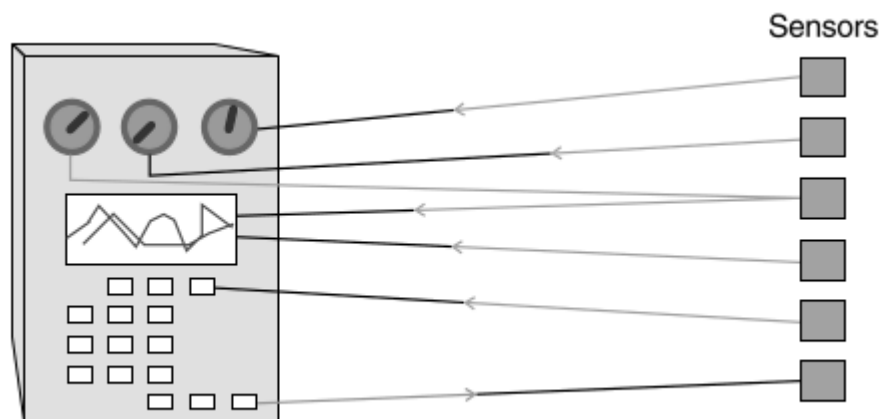
3.2 Supervisório

Os sistemas supervisórios, comumente denominados SCADA (Supervisory Control and Data Acquisition), desempenham papel central na automação industrial ao permitir o monitoramento, controle e análise de processos em tempo real. Esses sistemas são compostos por uma arquitetura distribuída que integra sensores, controladores lógicos programáveis

(CLPs), interfaces homem-máquina (IHMs) e servidores de dados, viabilizando a supervisão remota e a tomada de decisões baseada em dados operacionais.

A evolução dos sistemas SCADA tem sido marcada pela transição de protocolos proprietários para padrões abertos e interoperáveis, como o Modbus TCP/IP, que utiliza redes Ethernet para comunicação entre dispositivos. Essa mudança, conforme destacado em Clarke (2004), ampliou significativamente a flexibilidade e a escalabilidade dos sistemas supervisórios, permitindo sua integração com dispositivos de medição, como multimedidores industriais, e com plataformas de análise de dados. A figura 3 ilustra a obtenção de dados e sinais por um painel:

Figura 3- Aquisição de dados por um painel conectado a sensores



Fonte: Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems

3.2.1 Protocolo Modbus TCP/IP

O protocolo Modbus TCP/IP representa uma evolução significativa do protocolo Modbus RTU, oferecendo vantagens substanciais em termos de velocidade de transmissão, capacidade de endereçamento e integração nativa com redes Ethernet já existentes no ambiente industrial. Diferentemente do Modbus RTU, que utiliza camadas físicas seriais como RS-485 e RS-232, o Modbus TCP/IP opera sobre a pilha TCP/IP padrão, facilitando a interoperabilidade com sistemas corporativos e a integração com tecnologias de Indústria 4.0 e IoT Industrial (IIoT).

Conforme destacado por Tamboli (2015), a implementação do Modbus TCP em controladores programáveis como o Siemens S7-1200 demonstra sua versatilidade e aplicabilidade em processos industriais automatizados, permitindo comunicações robustas e expansíveis sem a necessidade de hardware adicional de conversão.

A especificação oficial Modbus TCP da Schneider Electric demonstrada por Swales (1999) enfatiza as vantagens da arquitetura baseada em TCP sobre UDP, destacando que "o protocolo TCP orientado à conexão é usado em vez do protocolo UDP orientado a datagramas para manter o controle de uma 'transação' individual, encapsulando-a em uma conexão que pode ser identificada, supervisionada e cancelada sem exigir ação específica por parte das aplicações cliente e servidor". Esta abordagem proporciona ampla tolerância a mudanças de desempenho da rede e permite que recursos de segurança como firewalls e proxies sejam facilmente adicionados.

O Modbus TCP utiliza uma arquitetura baseada no modelo cliente/servidor (equivalente aos termos master/slave na terminologia tradicional), onde o cliente inicia todas as transações e cada servidor responde exclusivamente às solicitações recebidas. Segundo Clarke (2004), tal abordagem proporciona controle determinístico sobre o fluxo de dados, atributo essencial em aplicações que exigem baixa latência e temporização previsível — como controle de malhas fechadas, supervisão de linhas de produção ou monitoramento de variáveis elétricas críticas.

3.2.2 Formato da Mensagem e Cabeçalho MBAP

Cada requisição Modbus TCP/IP é encapsulada em um frame TCP contendo o cabeçalho MBAP (Modbus Application Protocol Header), formado por:

- Transaction Identifier – identifica unicamente cada transação cliente/servidor.
- Protocol Identifier – fixo em zero para Modbus.
- Length – indica o tamanho restante da mensagem.
- Unit Identifier – identifica o dispositivo destino (quando há gateways ou múltiplos nós).
- Function Code – define a operação a ser executada.
- Data – carrega os parâmetros da requisição ou os dados de resposta.

O ciclo de comunicação no Modbus TCP segue um padrão bem definido:

1. O cliente gera e encapsula a requisição Modbus em um frame TCP/IP.
2. Essa requisição é transmitida via Ethernet (camadas física e de enlace IEEE 802.3).
3. O servidor interpreta o comando, executa a operação (como leitura de registradores ou escrita de saídas) e prepara a resposta.
4. A resposta é enviada ao cliente, completando o ciclo.

3.3 Flutter e Dart

O desenvolvimento de aplicações multiplataforma tem experimentado um crescimento substancial nos últimos anos, impulsionado pela necessidade de reduzir custos de desenvolvimento e acelerar o tempo de lançamento no mercado. Nesse contexto, o Flutter emerge como um framework de código aberto desenvolvido pelo Google, que utiliza a linguagem de programação Dart para criar interfaces nativas de alta qualidade para múltiplas plataformas a partir de uma única base de código (Zammetti, 2019).

A linguagem Dart, por sua vez, foi projetada especificamente para o desenvolvimento de interfaces de usuário, combinando recursos de linguagens orientadas a objetos com tipagem forte opcional e compilação ahead-of-time (AOT) para código nativo. Conforme documentado por Kopec (2019), essa combinação favorece a construção de aplicações industriais robustas que mantêm desempenho consistente mesmo ao processar grandes volumes de dados em tempo real. O autor destaca como o sistema de compilação em duas etapas do Dart permite otimizações substanciais para ambientes com recursos limitados, característica essencial para aplicações de monitoramento industrial que precisam garantir responsividade constante.

O padrão arquitetural Provider, amplamente utilizado no ecossistema Flutter, oferece uma solução eficiente para o gerenciamento de estado em aplicações que necessitam sincronizar dados entre diferentes componentes da interface como evidenciado por Biessek (2021). Esse modelo se mostra particularmente adequado para aplicações de monitoramento industrial, onde diversos painéis e visualizações precisam refletir simultaneamente as mudanças nos dados coletados. Como complemento prático, o Apêndice D apresenta o link de download do projeto desenvolvido em Dart, permitindo a reprodução e análise detalhada da implementação.

4 METODOLOGIA

A metodologia desenvolvida para o aplicativo de data manager e análise da qualidade de energia elétrica com conexão Modbus envolveu uma série de etapas fundamentais para garantir a eficiência e confiabilidade do sistema. Inicialmente, foi projetado o layout da aplicação, visando uma interface intuitiva e acessível.

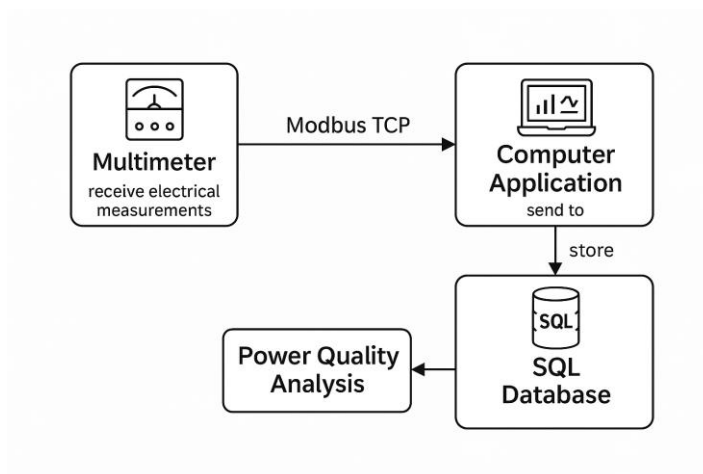
A comunicação e exibição dos dados do multimedidor via Modbus TCP foram implementadas para proporcionar uma visualização detalhada das informações elétricas. A análise da qualidade da energia da rede foi otimizada por meio da criação de um banco de dados, permitindo o armazenamento e processamento dos dados obtidos. Além disso, o Firebase foi utilizado não apenas para a comunicação e gestão de usuários, mas também para o armazenamento online dos dados adquiridos, garantindo maior segurança e acessibilidade das informações. A plataforma viabilizou a sincronização em tempo real e facilitou o gerenciamento remoto dos registros.

Um sistema de alarmes foi incorporado para alertar sobre eventos críticos, e funcionalidades adicionais foram desenvolvidas para exportação de relatórios em PDF e armazenamento de dados em Excel, assegurando a organização e disponibilidade das informações para análises futuras e tomadas de decisão estratégicas, conforme ilustrado na página de banco de dados de alarmes apresentada no Apêndice J.

4.1 Interface Modbus TCP

A comunicação Modbus TCP estabelece uma ligação contínua e confiável entre o multimedidor, responsável por medir variáveis elétricas críticas, e o aplicativo executado em um computador conectado à mesma rede local do multimedidor, onde o aplicativo atua como cliente solicitando e processando essas informações. A Figura 4 indica o fluxo das informações adquiridas:

Figura 4 - Diagrama de tráfego de dados



Fonte: Autoria própria

Esse aplicativo, por sua vez, organiza e envia os dados para o banco de dados SQL local, onde são armazenados de forma estruturada para consultas e análises posteriores. O fluxo é cuidadosamente sincronizado, com leituras periódicas e respostas no formato adequado, minimizando perdas ou inconsistências. Essa arquitetura garante que cada etapa — da coleta à gravação — preserve a integridade e a precisão das informações de qualidade de energia.

4.1.1 Coleta das Leituras pelo Multimetro

O multimetro realiza medições contínuas e de precisão das principais grandezas elétricas do sistema trifásico — tensão, corrente, potências ativa, reativa e aparente, consumo energético, fator de potência, frequência e distorção harmônica — e disponibiliza esses valores no formato apropriado para transmissão via protocolo Modbus TCP ao aplicativo cliente. Ressalta-se que, em função do mapa de registradores fixo utilizado pela aplicação, somente os modelos 15, 20 e 30 da série M1M da ABB são compatíveis com o sistema implementado; o conjunto de registradores empregados encontra-se detalhado no Apêndice N. Essa configuração assegura a integridade dos dados coletados, servindo de base para o armazenamento estruturado e para as subseqüentes análises de qualidade de energia. O multimetro realiza medições das principais variáveis elétricas, incluindo:

- Tensão (V)
- Corrente (A)

- Potências Ativa (kW), reativa (kVA) e aparente (kVAR)
- Consumo de energia Ativa(kWh), reativa (kVAh) e aparente (kVARh)
- Fator de potência
- Frequência (Hz)
- Distorção Harmônica

4.3 Armazenamento no Banco de Dados SQL Local

O aplicativo no computador atua como cliente Modbus TCP, solicitando ao multimedidor, em intervalos de 1 minuto, as leituras das variáveis elétricas. As requisições são enviadas pelo protocolo, e o multimedidor responde com os valores no formato adequado. Após processar as informações, o aplicativo as registra no banco de dados SQLite local, organizando-as em tabelas estruturadas para consultas rápidas e análises históricas. Caso não exista um banco de dados no momento da gravação, um novo arquivo será automaticamente criado na pasta “Documentos” do sistema, com o nome Local_data, garantindo a continuidade e integridade do registro dos dados.

4.4 Análise de Qualidade da Energia

O aplicativo também possui um módulo de análise de qualidade que consulta os dados armazenados no banco de dados e processa essas informações para identificar padrões, como variações excessivas de tensão, distorções harmônicas ou interrupções no fornecimento de energia; caso não exista a quantidade mínima de 1.008 amostras necessárias para o cálculo de determinado índice, o sistema interrompe a avaliação desse indicador e exibe a mensagem "Quantidade de amostras insuficiente para cálculo do índice".

Com base nesses dados, o sistema realiza uma avaliação detalhada e apresenta os resultados de forma visual na interface gráfica, permitindo ao usuário interpretar os diagnósticos e tomar decisões fundamentadas sobre a qualidade da energia disponível. Isso garante um acompanhamento eficiente e facilita a identificação de possíveis problemas na rede elétrica.

Para a análise da qualidade da energia elétrica, adotou-se como principal referência o Módulo 8 do Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional (PRODIST), elaborado pela ANEEL. Esse documento estabelece diretrizes técnicas

fundamentais, como os limites de variação de tensão em regime permanente, garantindo que o fornecimento de energia atenda aos padrões adequados de desempenho e confiabilidade (ANEEL, 2021).

A avaliação considerou cinco indicadores essenciais. O primeiro deles foi a variação de tensão em regime permanente, analisando se os valores registrados estão dentro dos limites permitidos. O fator de potência também foi verificado, pois sua eficiência influencia diretamente o consumo energético e a operação do sistema elétrico. Além disso, foram examinados os harmônicos, que podem causar distorções na forma de onda da tensão e corrente, afetando o desempenho de dispositivos elétricos.

O desequilíbrio de tensão foi outro parâmetro essencial, pois tensões desbalanceadas podem gerar sobrecarga e reduzir a vida útil dos equipamentos. Por fim, as variações de frequência foram avaliadas, garantindo a estabilidade da rede e evitando oscilações que poderiam comprometer o fornecimento de energia. Esses fatores foram considerados para assegurar uma análise detalhada e confiável da qualidade da energia, possibilitando ajustes e melhorias conforme necessário.

4.4.1 Variação de tensão em regime permanente

A variação de tensão em regime permanente refere-se à comparação da tensão medida no ponto de conexão com os níveis estabelecidos como adequados, precários e críticos. Essas medições são essenciais para evitar problemas como o mau funcionamento de dispositivos elétricos e garantir a segurança do sistema de distribuição.

As concessionárias de distribuição devem monitorar essas variações e atuar preventivamente para assegurar que a tensão permaneça dentro dos padrões adequados. A avaliação é feita por meio de medições periódicas e deve seguir metodologias padronizadas, garantindo a conformidade dos níveis de tensão.

4.4.2 Critérios e Indicadores

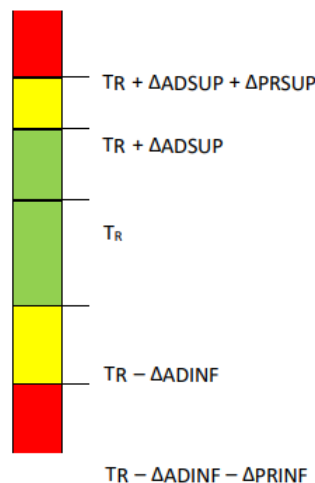
O PRODIST estabelece que as medições devem ser feitas nos pontos de conexão entre distribuidoras e unidades consumidoras. Para classificar a tensão, utiliza-se uma faixa de referência (Tensão de Referência - TR) e, com base nela, a tensão pode ser classificada como:

- Adequada: Intervalo dentro dos limites aceitáveis.

- Precária: Quando há pequenos desvios para cima ou para baixo da TR.
- Crítica: Quando os valores estão muito acima ou abaixo da TR, podendo comprometer o funcionamento dos equipamentos.

As leituras de tensão são classificadas segundo a faixa de referência conforme a figura 5:

Figura 5 - Faixas de tensão de fase ideal, precária e crítica



Fonte: PRODIST: Módulo 8 – Qualidade do Fornecimento de Energia Elétrica

Sendo:

$\Delta ADSUP$: Desvio limite superior da faixa adequada

$\Delta PRSUP$: Desvio limite superior da faixa precária

$\Delta ADINF$: Delta Adequada Inferior

$\Delta PRINF$: Delta Precária Inferior

O apêndice O demonstra os intervalos de classificação de acordo com a tensão que se está aferindo.

Os principais indicadores usados para medir a qualidade da tensão em regime permanente são:

1. Duração Relativa da Transgressão de Tensão Precária (DRP) – Mede o tempo em que a tensão ficou fora da faixa adequada, mas ainda dentro dos limites precários.
2. Duração Relativa da Transgressão de Tensão Crítica (DRC) – Mede o tempo em que a tensão permaneceu fora dos limites críticos.

Os indicadores são calculados com base em medições de tensão realizadas ao longo de um período de um mês civil. O cálculo leva em conta 1.008 leituras válidas ao longo de 7 dias, obtidas em intervalos consecutivos (período de integralização) de 10 minutos cada, totalizando 168 horas de medições. As equações de cálculo são:

$$DRP = \frac{nlp}{1008} \times 100\%$$

$$DRC = \frac{nlc}{1008} \times 100\%$$

Sendo:

nlp : maior valor entre as fases do número de leituras situadas na faixa precária

nlc : maior valor entre as fases do número de leituras situadas na faixa crítica

Para garantir a qualidade do fornecimento de energia elétrica, o PRODIST define limites máximos para os indicadores:

- Se o DRP for $\leq 3\%$, a tensão é considerada adequada.
- Se o DRC for $\leq 0,5\%$ a tensão é adequada
- Se ultrapassar 3% , a rede apresenta não conformidade, indicando possível necessidade de correção por parte da distribuidora.

Quando o processo da aplicação passa a registrar estes eventos, o cliente passa a ter visibilidade dessa não conformidade e das potenciais compensações financeiras por parte da distribuidora, transformando esse registro em uma vantagem adicional para o usuário.

4.4.3 Fator de potência

O PRODIST estabelece que o fator de potência das unidades consumidoras deve permanecer dentro de faixas específicas para garantir a eficiência do sistema elétrico. Os limites definidos são: entre 0,92 e 1,00 indutivo ou entre 1,00 e 0,92 capacitivo. Esses valores indicam a proporção entre a potência ativa — efetivamente utilizada para realizar trabalho útil — e a potência aparente, que inclui também a potência reativa, responsável por sustentar os campos eletromagnéticos em equipamentos como motores e transformadores.

O fator de potência (FP) é, portanto, uma medida da eficiência com que a energia elétrica é convertida em trabalho útil em sistemas de corrente alternada. Ele é definido como o cosseno do ângulo de defasagem entre tensão v e a corrente, sendo expresso pela equação:

$$FP = \cos (\theta_v - \theta_i)$$

onde θ_v representa o ângulo de fase da tensão e θ_i o da corrente. Alternativamente, o fator de potência pode ser calculado como a razão entre a potência ativa P e a potência aparente S :

$$FP = \frac{P}{S}$$

Essa relação evidencia que quanto mais próximo de 1 for o fator de potência, maior será a proporção de energia efetivamente convertida em trabalho útil em relação à energia total fornecida ao sistema. Valores inferiores indicam maior presença de potência reativa, o que implica em maior circulação de corrente reativa e, conseqüentemente, maiores perdas por efeito Joule e sobrecarga nos componentes da rede.

Quando o fator de potência se mantém dentro dos limites regulamentares, o sistema elétrico opera de forma mais eficiente, reduzindo perdas e evitando sobrecarga nas redes de distribuição. No entanto, caso o fator de potência fique abaixo de 0,92 (seja por excesso de potência reativa indutiva ou capacitiva), a distribuidora está autorizada a aplicar encargos adicionais na fatura do consumidor. Essa cobrança tem caráter corretivo e visa incentivar a adoção de medidas de compensação, como a instalação de bancos de capacitores ou outros dispositivos de correção, que reduzem o consumo de potência reativa e melhoram o desempenho energético.

4.4.4 Distorção harmônica

O critério de avaliação da Distorção Harmônica Total (DTT95%) está relacionado à qualidade do fornecimento de energia elétrica e mede a intensidade das distorções harmônicas presentes na tensão elétrica. O DTT95% é o valor que só 5% das medições alcançaram ou superaram o limite normativo, derivado da distribuição estatística das 1.008 medições válidas dentro do período de integralização, ou seja, fornece uma medida estatística que indica o nível de distorção harmônica que ocorre na maior parte do tempo dentro de um sistema elétrico.

Matematicamente, a Distorção Harmônica Total de tensão (DTT ou THD_V) é calculada pela expressão:

$$THD_V = \frac{\sqrt{\sum_{h=2}^n V_h^2}}{V_1}$$

onde:

- V_1 é o valor RMS da componente fundamental da tensão,
- V_h é o valor RMS da componente harmônica de ordem h,
- n é a ordem máxima de harmônicos considerada na análise.

Segundo Yousefpoor et al. (2012), essa relação representa a razão entre o valor eficaz de todas as componentes harmônicas e o valor eficaz da componente fundamental, sendo amplamente utilizada para quantificar a qualidade da forma de onda de tensão em sistemas elétricos.

Os limites aceitáveis de DTT95% variam de acordo com a tensão nominal do sistema e estão definidos na Resolução Normativa nº 956/2021 da ANEEL, conforme a Tabela 1:

Tabela 1 - Limites de distorção harmônica DTT95%

Tensão Nominal V_n	Limite de DTT95%
$V_n \leq 2,3 \text{ kV}$	10,0%
$2,3 < V_n < 69 \text{ kV}$	8,0%
$69 \leq V_n < 230 \text{ kV}$	5,0%

PRODIST: Módulo 8 – Qualidade do Fornecimento de Energia Elétrica

A avaliação desse indicador é crucial para garantir a qualidade da energia elétrica entregue aos consumidores, evitando danos a equipamentos eletrônicos sensíveis e assegurando a conformidade com os padrões regulatórios. Valores elevados de distorção harmônica podem

causar aquecimento excessivo, redução da vida útil dos equipamentos e interferências nos sistemas de medição e controle.

4.4.5 Desequilíbrio de tensão

O desequilíbrio de tensão ocorre quando há diferenças significativas entre as amplitudes das tensões de fase em um sistema trifásico ou quando a defasagem elétrica entre essas tensões não é exatamente 120°. Esse fenômeno pode resultar em funcionamento inadequado de equipamentos elétricos, principalmente motores trifásicos, causando aquecimento excessivo, perda de eficiência e até falhas prematuras. O FD% quantifica o nível de desequilíbrio de tensão utilizando a seguinte equação:

$$FD\% = 100 \times \sqrt{\frac{1 - \sqrt{3 - 6\beta}}{1 + \sqrt{3 - 6\beta}}}$$

Onde β é um parâmetro calculado a partir das tensões de linha V_{ab} , V_{bc} e V_{ca} , conforme:

$$\beta = \frac{V_{ab}^4 + V_{bc}^4 + V_{ca}^4}{(V_{ab}^2 + V_{bc}^2 + V_{ca}^2)^2}$$

4.4.6 Critério FD95%

O FD95% representa o valor do fator de desequilíbrio que foi superado por apenas 5% das 1.008 medições válidas. Os limites aceitáveis do FD95% variam de acordo com a tensão nominal do sistema, e estão definidas conforme Tabela 2.

Tabela 2 - Limite do fator distorção de tensão FD95%

Tensão Nominal	Limite de FD95%
$V_n \leq 2,3 \text{ kV}$	3,0%
$2,3 < V_n < 69 \text{ kV}$	2,0%

Fonte: PRODIST: Módulo 8 – Qualidade do Fornecimento de Energia Elétrica

Segundo a IEEE Std 1159 (2019) estes índices de qualidade de energia são importantes por estas razões:

- Transientes (sobretensões impulsivas e oscilatórias): podem provocar falhas imediatas em eletrônicos sensíveis e acelerar a degradação dielétrica de transformadores, cabos e isoladores.
- Variações de tensão de curta e longa duração (afundamentos e elevações): causam desligamentos momentâneos, aumento de perdas térmicas em máquinas e desarme de controladores, prejudicando a continuidade operacional.
- Desequilíbrio de fases: gera componentes de sequência negativa que aumentam correntes assimétricas nos enrolamentos, originando aquecimento adicional, perda de eficiência e maior risco de falhas prematuras em equipamentos rotativos.
- Distorção harmônica: deformação da forma de onda por cargas não lineares que eleva perdas térmicas, compromete medições e atuações de proteção e pode ser amplificada por ressonâncias no sistema.
- Fator de potência: fator de potência baixo aumenta correntes e perdas resistivas; correções reativas sem análise de impedância e do conteúdo harmônico podem provocar ressonância e agravar distorções.
- Monitoramento e validação: medição contínua e verificação pós-intervenção são essenciais para identificar causas, dimensionar soluções e confirmar a eficácia das medidas.

Por isso, a medição contínua e a adoção de medidas corretivas são essenciais para a manutenção da qualidade do fornecimento de energia.

4.4.7 Metodologia de Cálculo de Tensões de Linha usando Transformação Polar-Retangular

Em sistemas elétricos trifásicos, o cálculo preciso das tensões de linha a partir das tensões de fase é fundamental para análise da qualidade de energia, especialmente em condições de desequilíbrio. Este trabalho implementa uma metodologia baseada na transformação polar-retangular para calcular as tensões de linha, conforme descrito por Sadiku (2013) em análises em que elas não estão disponíveis.

4.4.7.1 Fundamentos Teóricos da Representação Fasorial

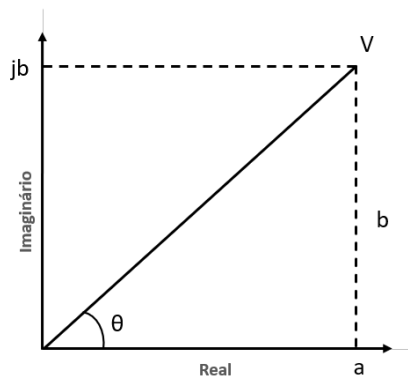
A representação fasorial permite expressar grandezas senoidais em função de sua magnitude e ângulo de fase, facilitando operações matemáticas complexas. Um fasor pode ser representado tanto em coordenadas polares quanto retangulares (Sadiku, 2013, p. 828):

Coordenadas Polares: $V = |V| \angle \theta$

Coordenadas Retangulares: $V = a + jb$

A Figura 6 demonstra graficamente as componentes vetoriais do fasor de tensão:

Figura 6 - Representação dos componentes do fasor de tensão



Fonte: Autoria própria

onde:

$|V|$ é a magnitude do fasor (valor RMS)

“ θ ” é o ângulo de fase em graus,

“a” é a componente real,

“b” é a componente imaginária,

“j” é a unidade imaginária.

4.4.7.2 Transformações entre Sistemas de Coordenadas

As transformações entre coordenadas polares e retangulares seguem as relações trigonométricas fundamentais (Sadiku, 2013, p. 829):

Polar para Retangular:

$$a = |V| \times \cos(\theta) \quad b = |V| \times \text{sen}(\theta)$$

Retangular para Polar:

$$|V| = \sqrt{(a^2 + b^2)} \quad \theta = \arctan\left(\frac{b}{a}\right)$$

4.4.7.3 Cálculo de Tensões de Linha

As tensões de linha em um sistema trifásico são normalmente obtidas pela diferença vetorial entre tensões de fase adjacentes. A aplicação oferece uma opção para calcular automaticamente essas tensões; quando as medidas de tensão de linha já estiverem disponíveis, o cálculo é ignorado e os valores informados são usados diretamente. Para um sistema com fases A, B e C, as tensões de linha são:

$$V_{ab} = V_a - V_b \quad V_{bc} = V_b - V_c \quad V_{ca} = V_c - V_a$$

A metodologia desenvolvida neste trabalho utiliza a transformação polar-retangular para calcular tensões de linha considerando desequilíbrios reais do sistema. O algoritmo implementado segue os seguintes passos:

1. **Conversão para Coordenadas Retangulares:** Cada tensão de fase é convertida de coordenadas polares (magnitude, ângulo) para retangulares (real, imaginária).
2. **Subtração Vetorial:** A diferença entre tensões de fase é calculada em coordenadas retangulares, onde a operação de subtração é direta.
3. **Conversão para Coordenadas Polares:** O resultado da subtração é convertido de volta para coordenadas polares, fornecendo a magnitude da tensão de linha.

Adotou-se, para a análise, um ângulo fixo de 120° entre as fases. Ressalta-se que, em desenvolvimentos futuros, poderá ser incorporada a funcionalidade de armazenamento e leitura dos ângulos de fase, possibilitando um cálculo com maior precisão.

4.4.8 Critérios de Avaliação da Frequência

O PRODIST define que, em condições normais de operação e em regime permanente, o sistema de distribuição deve operar dentro da faixa de 59,9 Hz a 60,1 Hz. Essa frequência nominal de 60 Hz é essencial para a estabilidade dos processos industriais, o funcionamento

eficiente de motores e a operação correta de dispositivos eletrônicos. Entretanto, distúrbios no sistema elétrico podem causar desvios na frequência, exigindo uma ação rápida para reequilibrar carga e geração. Os limites definidos pelo PRODIST para esses casos são:

- Após um distúrbio, a frequência deve retornar ao intervalo de 59,5 Hz a 60,5 Hz dentro de 30 segundos.
- Em situações extremas, a frequência não pode exceder 66 Hz ou ser inferior a 56,5 Hz.
- Caso seja necessária uma redução na geração ou no consumo de energia para estabilização do sistema, os seguintes limites devem ser respeitados:
 - o Frequência acima de 62 Hz por no máximo 30 segundos.
 - o Frequência acima de 63,5 Hz por no máximo 10 segundos.
 - o Frequência abaixo de 58,5 Hz por no máximo 10 segundos.
 - o Frequência abaixo de 57,5 Hz por no máximo 5 segundos.

Desvios significativos da frequência nominal podem gerar efeitos adversos graves, como:

- Danos a equipamentos sensíveis, incluindo eletrônicos industriais e sistemas de controle.
- Sobrecarga e aquecimento em motores e transformadores devido ao desbalanceamento eletromagnético.
- Comprometimento da estabilidade do sistema elétrico, aumentando a chance de falhas sistêmicas ou apagões.

4.5 Método de construção do servidor Firebase

O Google Firebase Realtime Database usa uma estrutura em árvore JSON para sincronizar dados em tempo real entre clientes e servidores. Ele fornece flexibilidade e performance para leituras rápidas, enquanto bancos SQL suportam transações e relacionamentos complexos; as duas abordagens podem ser combinadas via replicação ou eventos de download e sincronização, mantendo o SQL como fonte primária e o Firebase para acesso em tempo real. Ele é amplamente utilizado em aplicações móveis e web devido à sua facilidade de integração, escalabilidade e capacidade de atualização instantânea dos dados em múltiplos dispositivos conectados.

No contexto deste trabalho, o Realtime Database foi utilizado como repositório central para consolidar dados coletados localmente, oferecendo redundância que permite acesso remoto e preservação dos dados como backup. A sincronização entre o banco de dados local e o Firebase foi realizada por meio de requisições HTTP, utilizando o formato JSON para a troca de informações. Esse processo garante que os dados armazenados localmente sejam enviados para o banco remoto, evitando duplicidade e respeitando as restrições de estrutura impostas pelo Firebase. O banco de dados local utilizado é exclusivamente do tipo SQLite 3, contendo as seguintes colunas: data, hora, variável (representando a grandeza medida) e valor. As variáveis devem ser identificadas exatamente conforme os seguintes códigos, respeitando letras maiúsculas: FP, FREQ, L1, L2L3, THDc A, THDc B, THDc C, THDv A, THDv B, THDv C, V12, V23, V31, V1N, V2N, V3N, kW1, kW2, kW3, kVA1, kVA2, kVA3, kVAR1, kVAR2, kVAR3, kVARtot, kVAtot, kWtot.

Para garantir a integridade e a consistência dos dados, cada registro local é identificado por uma chave única, composta por informações como data, hora e variável monitorada. As chaves dos dados salvos no servidor Firebase são geradas para garantir a unicidade de cada registro, evitando duplicidade e facilitando a busca e atualização dos dados. O processo de geração dessas chaves segue um padrão que utiliza a combinação de três campos principais: data, hora e variável.

No código, a chave é construída da seguinte forma:

- dataPadronizada: A data é sanitizada para substituir caracteres que não são permitidos em chaves do Firebase (como ., /, #, \$, [,]) por _.
- hora: O horário da medição.
- variável: O nome da variável medida (por exemplo, kWh, V1N etc.).

A chave final é a concatenação desses três elementos, separados por sublinhados (_).

Exemplo: Suponha que tenha sido gravado um registro com os seguintes valores:

- data: 10/7/2025
- hora: 08:00:00
- variável: kWh
- valor: 1234.56

A chave gerada será:

10_7_2025_08:00:00_kWh

No Firebase, o dado será salvo sob o nó:

MonitorDB_remote/10_7_2025_08:00:00_kWh

E o conteúdo desse nó será:

```
{
  "data": "10/7/2025",
  "hora": "08:00:00",
  "variavel": "kWh",
  "valor": 1234.56
}
```

Quando a aplicação recebe novos dados via conexão Modbus, ela compara a chave de cada novo dado com todos os registros existentes no banco de dados. Antes de enviar um novo dado ao Firebase, o sistema verifica se já existe um registro correspondente no banco remoto. Caso não exista, o dado é transmitido e armazenado; caso contrário, a sincronização daquele item é ignorada, evitando redundância.

Além disso, o processo de sincronização contempla a sanitização das chaves utilizadas no banco de dados, substituindo caracteres proibidos pelo Firebase, como pontos, barras e colchetes, por caracteres permitidos. Essa etapa é fundamental para garantir que os dados sejam aceitos e corretamente organizados na estrutura do Realtime Database.

A utilização do Firebase Realtime Database neste projeto proporciona uma solução robusta e eficiente para o armazenamento centralizado e a sincronização de dados, facilitando o acesso remoto, a escalabilidade e a integridade das informações em aplicações distribuídas.

4.6 Sistema de alarmes

Sistemas de alarme representam um componente crítico em aplicações de monitoramento de energia elétrica, fornecendo alertas automáticos sobre condições anômalas que podem indicar falhas iminentes ou operação inadequada de equipamentos elétricos. A

detecção precoce de distúrbios como sobretensões, subtensões, sobrecorrentes e desvios no fator de potência é essencial para manter a continuidade operacional e prevenir danos a equipamentos sensíveis.

Tradicionalmente, sistemas de supervisão dependem de operadores para monitoramento contínuo e identificação manual de condições de alarme Cravo (2022). Esta abordagem apresenta limitações significativas em termos de tempo de resposta e disponibilidade, especialmente em instalações remotas ou durante períodos de operação desatendida. A figura 7 demonstra a página de configuração dos alarmes.

Figura 7 - Página de configuração de alarmes

Fonte: Autoria Própria

4.6.1 Componentes Principais

O sistema de alarmes implementado é composto por quatro subsistemas integrados:

Subsistema de Configuração: Interface para definição de limites de alarme, habilitação/desabilitação de tipos específicos de alarme e configuração de parâmetros de notificação.

Subsistema de Detecção: Algoritmos de comparação contínua entre valores medidos e limites configurados, com validação de dados e filtragem de falsos alarmes.

Subsistema de Notificação: Mecanismo de envio automático de emails com controle de frequência e formatação específica por tipo de evento.

Subsistema de Persistência: Armazenamento local de eventos de alarme com informações completas para rastreabilidade e análise posterior.

4.6.2 Tipos de Alarmes Implementados

O sistema monitora quatro categorias principais de distúrbios elétricos:

Alarmes de Tensão: Detecção individual para cada fase (V1N, V2N, V3N) com limites mínimos e máximos configuráveis. O sistema verifica continuamente se os valores medidos estão dentro da faixa operacional definida.

Alarmes de Corrente: Monitoramento de sobrecorrente e subcorrente para cada fase (L1, L2, L3), permitindo detecção de sobrecargas.

Alarmes de Fator de Potência: Supervisão do fator de potência total com limites configuráveis para atendimento às regulamentações de qualidade de energia.

Alarmes de Interrupção: Detecção automática de interrupções de fornecimento baseada na identificação de tensões zeradas em qualquer fase.

Para prevenir spam de notificações durante eventos prolongados, o sistema implementa controle de frequência baseado em intervalos configuráveis.

4.7 Gestão de usuários

A gestão de usuários em sistemas digitais é um componente essencial para garantir controle de acesso, segurança da informação e hierarquia de permissões. Neste projeto, foi implementado um sistema em Flutter com integração ao Firebase Authentication para controle de credenciais e níveis de acesso. O método adotado permite a distinção entre usuários comuns e administradores, sendo que este último possui privilégios estendidos para manutenção das contas.

Cada usuário registrado pode acessar a interface de gestão, na qual é possível visualizar informações básicas de login e executar ações específicas. Um dos recursos disponíveis para todos os usuários é a redefinição de senha, realizada por meio do envio automático de um link

de redefinição para o e-mail cadastrado. Este processo segue as diretrizes de segurança do Firebase e responde a uma limitação da plataforma: a senha só pode ser modificada diretamente pelo próprio usuário autenticado, como pode ser visualizado no apêndice L, página de configuração de usuário, não sendo permitida a alteração via aplicativo por terceiros. A representação visual dessa funcionalidade pode ser observada no Apêndice E, que apresenta a tela de login do sistema.

Administradores, por sua vez, têm acesso a funcionalidades como:

- Adição de novos usuários via diálogo interativo
- Deleção da própria conta, com redirecionamento imediato à tela de login
- Configuração de conexão Modbus com o multimedidor
- Habilitação e definição dos limiares dos alarmes

O uso do Firebase Authentication como método de persistência garante que as ações de autenticação, validação de sessão e controle de permissões sejam realizadas com alta confiabilidade, conformidade com padrões da indústria e integração transparente com o restante das funcionalidades do aplicativo. Essa abordagem assegura que alterações de configuração, incluindo conexões Modbus e parâmetros de alarme, fiquem restritas a usuários devidamente autorizados.

4.8 Validação do Projeto

A validação funcional constitui uma etapa essencial no ciclo de desenvolvimento de sistemas, tendo como finalidade assegurar que o projeto atenda aos requisitos previamente especificados e opere de forma adequada em condições de uso reais. No presente trabalho, a validação do projeto do aplicativo de gerenciamento de dados de energia foi realizada por meio de duas abordagens complementares: simulação de dispositivos físicos e análise de dados reais.

Na primeira abordagem, foi utilizada a placa ESP32 para simular um multimedidor de energia elétrica, empregando-se o protocolo de comunicação Modbus TCP, amplamente utilizado em sistemas industriais. A simulação foi implementada por meio da IDE Arduino, permitindo a transmissão de dados simulados de medição energética ao aplicativo em desenvolvimento. Essa etapa visou validar a comunicação entre o aplicativo e dispositivos físicos simulados, bem como verificar a correta interpretação e exibição dos dados na interface de usuário.

A segunda estratégia adotada consistiu na utilização do banco de dados referente ao monitoramento do consumo energético do Smart Campus da Unesp, conforme descrito por Zarpellon (2024), contendo amostras reais de consumo energético. A partir desses dados, foram aplicadas técnicas de análise estatística e visualização, a fim de validar a integridade das leituras e a capacidade do sistema em tratar eventos de amostras fora do padrão. Essa etapa proporcionou uma simulação do desempenho do aplicativo em um ambiente baseado em dados reais, conferindo maior robustez e realismo ao processo de validação.

A combinação dessas duas estratégias — uma baseada em simulação de hardware e outra fundamentada em dados reais — permitiu verificar a funcionalidade essencial do projeto e sua adequação aos objetivos estabelecidos, proporcionando confiança para o prosseguimento das fases subsequentes de desenvolvimento e refinamento do sistema.

4.9 Teste com ESP32 e endereçamento modbus

O uso do microcontrolador ESP32 para simulação de um multimedidor Modbus representa uma solução eficiente e flexível para testes, validação de sistemas e desenvolvimento de aplicações industriais. O ESP32, equipado com conectividade Wi-Fi e capacidade de processamento adequada, pode ser programado para emular o comportamento de equipamentos reais de medição, como multimedidores de energia, facilitando o desenvolvimento de sistemas de aquisição e monitoramento de dados.

No contexto apresentado, o ESP32 executa um firmware baseado na biblioteca ModbusIP, que permite ao dispositivo atuar como um servidor Modbus TCP. Os registradores Modbus são configurados para armazenar valores simulados de grandezas elétricas, como tensões, correntes, potências e energia acumulada (kWh). Cada registrador é mapeado conforme a especificação do protocolo Modbus definidos no manual do multimedidor ABB M1M, utilizando offsets e tipos de dados compatíveis com os equipamentos reais. O código fonte utilizado está disponível no apêndice M.

Os valores atribuídos aos registradores são gerados aleatoriamente ou calculados a partir de funções matemáticas, simulando variações reais das grandezas elétricas. Por exemplo, tensões de fase, tensões de linha, correntes e potências são atualizadas periodicamente, e o valor absoluto de energia (kWh) é incrementado para representar o consumo acumulado.

A comunicação entre o ESP32 e o sistema de monitoramento ocorre via protocolo Modbus TCP/IP. O ESP32, conectado à rede Wi-Fi, aguarda requisições de leitura (Read Holding Registers) provenientes de clientes Modbus, como softwares de supervisão ou sistemas embarcados (por exemplo, aplicações em Flutter/Dart).

O cliente Modbus estabelece uma conexão TCP com o ESP32, especificando o endereço IP e a porta configurada. Ao enviar uma requisição de leitura, o ESP32 responde com os valores presentes nos registradores solicitados, o conjunto de registradores cadastrados estão listados no apêndice N. Essa troca de informações segue o padrão do protocolo Modbus, garantindo interoperabilidade com sistemas industriais e comerciais.

No lado do cliente, o sistema de aquisição (por exemplo, uma aplicação Flutter/Dart) realiza leituras periódicas dos registradores Modbus do ESP32, obtendo os valores simulados das grandezas elétricas. Esses dados são processados, armazenados em bancos de dados locais e podem ser sincronizados com o servidor remoto Firebase cadastrado como mostrado na página de configuração exibido no apêndice K.

A simulação do multimedidor via ESP32 permite testar toda a cadeia de aquisição, armazenamento e visualização de dados, sem a necessidade de equipamentos físicos reais. Isso é especialmente útil em ambientes de desenvolvimento, validação de algoritmos e treinamento de equipes técnicas. O processo de aquisição dos dados segue o seguinte processo:

1. Configuração dos Endereços Modbus no ESP32

No firmware do ESP32, cada grandeza elétrica (tensão, corrente, potência, energia etc.) é associada a um par de registradores Modbus do tipo Holding Register como definido pelo manual do multimedidor ABB M1M. Por exemplo, para armazenar um valor de tensão de fase (VIN), são usados dois registradores consecutivos:

```
RegPhaseL1_0 = 23298, RegPhaseL1_1 = 23299;
```

```
// Registradores convertidos de hexadecimal (5B02, 5B03) para decimal
```

Esses registradores são adicionados ao servidor Modbus do ESP32 com:

```
mb.addHreg(RegPhaseL1_0, 0x0000); // Parâmetros de endereço e offset
```

```
mb.addHreg(RegPhaseL1_1, 0x0000);
```

Cada registrador armazena 16 bits, então para valores de ponto flutuante (float ou int32), o valor é convertido para hexadecimal e dividido em dois blocos de 16 bits.

2. Simulação e Escrita dos Dados nos Registradores

No loop principal do ESP32, valores simulados são gerados e escritos nos registradores. Por exemplo, para simular uma tensão de fase:

Gera um valor aleatório dentro do intervalo:

```
int randVph = random(2200, 2250);
```

Exemplo: 2234 representa 223,4 V

Converte o valor gerado em hexadecimal de 32 bits:

```
intToHex32(randVph, V1Nh, sizeof(V1Nh));
```

Separa em dois vetores para armazenar 16 bits usando a representação big-endian:

```
strncpy(V1Nh_0, &V1Nh[0], 4);
```

```
strncpy(V1Nh_1, &V1Nh[4], 4);
```

```
mb.Hreg(RegPhaseL1_0, (uint16_t)strtoul(V1Nh_0, &endPtr, 16)); // 0x0000
mb.Hreg(RegPhaseL1_1, (uint16_t)strtoul(V1Nh_1, &endPtr, 16)); // 0x08BA
```

Exemplo numérico:

Valor simulado: 2234

Hexadecimal: 0x000008BA

Registrador 0: 0x0000

Registrador 1: 0x08BA

3. Leitura dos Dados pelo Cliente Modbus (Flutter/Dart)

No lado do cliente (aplicação Flutter/Dart), é feita a configuração dos registradores a serem lidos, conforme o mapeamento do ESP32:

```
ModbusUint32Register(
  name: "V1N",
  address: 23298,
  type: ModbusElementType.holdingRegister,
  multiplier: 0.1,
),
```

A aplicação envia uma requisição Modbus TCP para o ESP32, solicitando a leitura dos endereços configurados. O ESP32 responde com os valores dos registradores.

4. Conversão dos Dados Recebidos

esses valores são combinados e multiplicados pelo fator de escala 0,1:

5. Armazenamento dos Dados no Banco Local

Após a leitura e conversão, os dados são armazenados no banco SQLite local através da seguinte função:

```
List<Map<String, dynamic>> dataList = dataToMap(data, formattedDate,
formattedTime);

for (var item in dataList) {

  await dbHelper.insertData(item);

}
```

Cada registro contém:

Data e hora da leitura

Nome da variável (ex: "V1", "kWh")

Valor convertido

Exemplo de registro salvo:

```
{
  "data": "11/1/2025",
  "hora": "14:23:00",
  "variavel": "V1N",
  "valor": 223.4
}
```

4.9.1 Teste com banco de dados UNESP Smart Campus

No contexto deste trabalho, foram desenvolvidas e aplicadas funções de análise de qualidade da energia elétrica utilizando dados provenientes do banco de dados do smart campus Unesp. O objetivo dessas funções é avaliar, de forma automatizada e quantitativa, diversos indicadores de qualidade, tais como tensão, frequência, fator de potência, distorção harmônica total e eventos de variações temporárias de tensão.

As funções de análise operam sobre séries temporais extraídas do banco de dados, compostas por amostras integralizadas em intervalos de 10 minutos. O processamento desses dados é realizado conforme os limites e critérios estabelecidos por normas técnicas, garantindo consistência e representatividade nas avaliações realizadas. Dentro deste contexto as seguintes funções em dart foram desenvolvidas:

1. calculateFD

Calcula o fator de desequilíbrio (FD%) entre as tensões de linha (Vab, Vbc, Vca).

Para cada amostra, utiliza uma fórmula baseada em potências das tensões para obter o FD%

2. calculateDTT

Avalia a distorção harmônica total (THD) das tensões. Para cada valor, verifica se está acima ou abaixo de 10%.

3. FPAnalyze

Analisa o fator de potência das amostras. Classifica cada valor como adequado (≥ 0.92 e ≤ 1) ou precário (fora desse intervalo).

4. FrequencyAnalyze

Avalia a frequência das amostras. Classifica cada valor como adequado (entre 59.5 Hz e 60.5 Hz) ou precário (fora desse intervalo).

5. IdentifyVoltageEvents

Identifica eventos de interrupção, afundamento e elevação temporária de tensão, usando como referência o módulo 8 do PRODIST.

Analisa cada amostra em relação ao valor nominal para classificar o tipo de evento.

Registra início, fim e duração de cada evento.

Atualiza as variáveis de estado com as quantidades de cada evento.

6. EvaluateData

Avalia as amostras de tensão de fase (V1N, V2N, V3N) conforme a tensão nominal selecionada. Classifica cada valor como adequado, precário ou crítico de acordo com faixas específicas para cada tensão nominal.

Os resultados dessas análises são apresentados em indicadores visuais e tabelas, permitindo uma avaliação clara e objetiva da qualidade da energia elétrica registrada smart campus Unesp.

O método adotado para testar e validar essas funções consistiu na utilização de dados reais e simulados armazenados no banco de dados do projeto, garantindo a representatividade dos resultados e a aderência às condições típicas de sistemas elétricos industriais e acadêmicos. Dessa forma, o conjunto de funções implementado contribui para o monitoramento, diagnóstico e melhoria da qualidade da energia elétrica, sendo aplicável tanto em ambientes de pesquisa quanto em aplicações práticas de gestão energética.

5 RESULTADOS

Este capítulo apresenta os resultados do projeto oriundos da aquisição e armazenamento dos dados obtidos via simulação, execução da análise de qualidade de energia via dados reais e simulados.

5.1 População e visualização do banco de dados

A análise da qualidade da energia elétrica foi realizada por meio de uma abordagem híbrida, combinando simulação em tempo real com dados reais coletados ao longo de 2024. Para a simulação, utilizou-se um microcontrolador ESP32, configurado para atuar como um multimedidor virtual. Os dados gerados pelo dispositivo foram armazenados localmente no banco de dados Local_data.db, permitindo testes e validações em ambiente controlado.

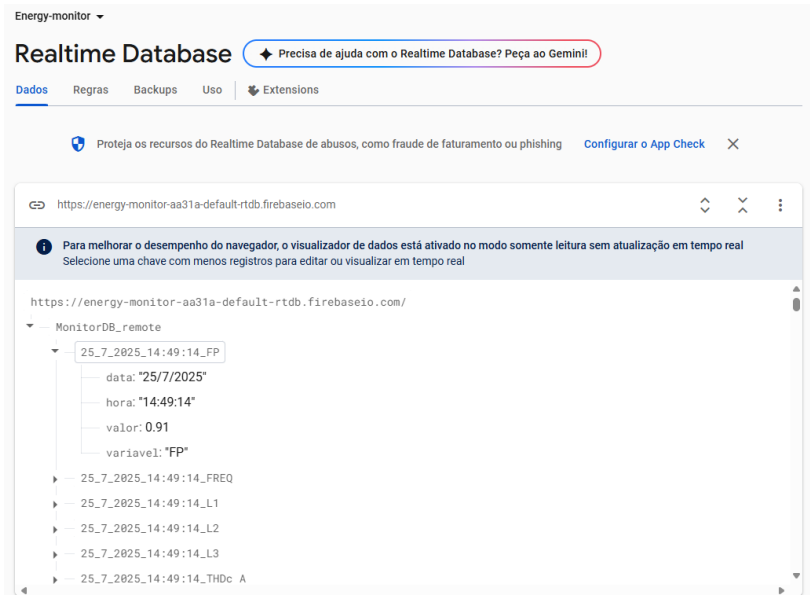
Paralelamente, foram incorporados dados reais provenientes do smart campus Unesp (Zarpellon,2024), coletados em formato csv. A amostragem abrangeu um intervalo 1008 amostras por semana de cada mês ao longo do ano de 2024. Esses dados foram inseridos no banco de dados Unesp_data.db, ampliando a robustez da análise ao incluir medições reais de campo.

As variáveis monitoradas incluíram:

- Tensões de fase (fase A, B e C)
- Fator de potência das três fases
- Distorção harmônica total (THD) por fase

Essas variáveis são fundamentais para avaliar a eficiência energética e a conformidade com padrões de qualidade, como os definidos pela ANEEL. A plataforma Firebase foi utilizada como interface de backup dos dados simulados pelo ESP32. Por meio da integração com o banco de dados local, os dados foram transmitidos e exibidos em tempo real sendo acessíveis via web. Essa visualização dinâmica permite o acompanhamento contínuo das variáveis elétricas, facilitando a identificação de anomalias, padrões de consumo e oportunidades de otimização. A Figura 8 demonstra a interface web do firebase com a estrutura de dados das variáveis:

Figura 8 - Interface web do firebase



Fonte: Autoria Própria

5.2 Interação e construção das páginas

A aplicação implementa uma arquitetura hierárquica baseada em módulos funcionais claramente delimitados, refletidos na organização do código e na experiência de navegação do usuário. Cada módulo funcional é encapsulado em um diretório específico contendo seus componentes, providers e helpers relacionados. Esta abordagem está alinhada com as recomendações de Zammetti (2019) para aplicações Flutter complexas, favorecendo a manutenibilidade e escalabilidade do código. A seguir será exibido a estrutura de organização da aplicação:

lib/

- └ Alarm_componentes/ # Componentes relacionados a alarmes
- └ Monitor_componentes/ # Componentes do módulo de monitoramento
- └ Trends_componentes/ # Componentes de visualização de tendências
- └ database/ # Camada de acesso a dados
- └ menu/ # Componentes de navegação global

|— provider/ # Providers e gerenciadores de estado

|— screen/ # Páginas principais da aplicação

└— quality_componentes/ # Componentes de análise de qualidade

Esta organização estrutural reflete a divisão conceitual da aplicação em sete módulos principais:

1. Dashboard: Visão consolidada dos principais indicadores
2. Monitoramento: Visualização em tempo real de grandezas elétricas
3. Gráficos: Análise histórica e gráficos de tendências
4. Banco de Dados: Consulta e exportação de registros históricos
5. Alarmes: Configuração e histórico de eventos de alarme
6. Qualidade: Análise técnica de parâmetros de qualidade de energia
7. Configuração: Ajustes do sistema e conectividade

5.2.1 Padrão de Gerenciamento de Estado

A aplicação implementa o padrão Provider para gerenciamento de estado centralizado, conforme evidenciado no arquivo main.dart. Este padrão, recomendado pela documentação oficial do Flutter para aplicações de média complexidade, permite a separação entre lógica de negócio e apresentação visual enquanto facilita o compartilhamento de estado entre widgets hierarquicamente distantes (FLUTTER DEV TEAM, 2023).

Esta implementação estabelece quatro providers principais:

- ModbusNotifier: Gerencia comunicação Modbus e dados em tempo real
- MenuAppController: Controla o estado da navegação e menu lateral
- AlarmProvider: Gerencia estado e configuração de alarmes
- AuthProvider: Controla autenticação e permissões de usuário

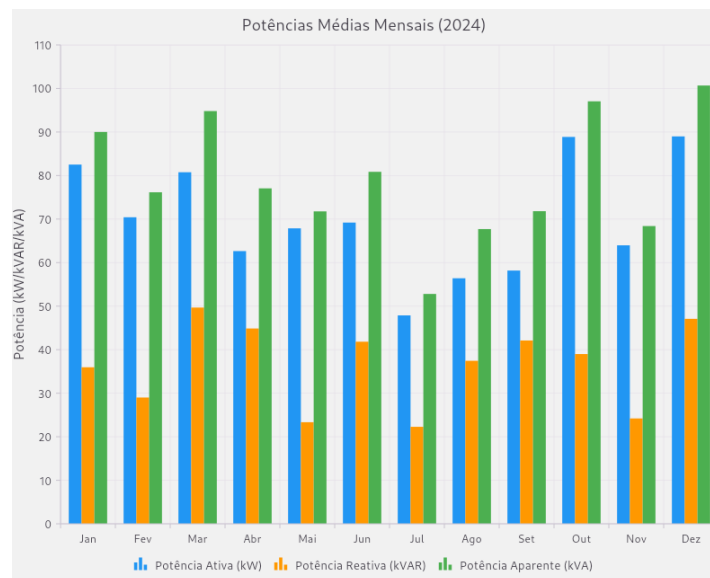
Este modelo facilita a propagação eficiente de mudanças de estado através da hierarquia de widgets. Por exemplo, quando novos dados são recebidos via Modbus, múltiplas páginas podem ser atualizadas simultaneamente sem necessidade de callbacks complexos ou passagem manual de estado.

5.3 Exibição dos gráficos

O sistema de visualização gráfica implementado no aplicativo representa um avanço na forma como dados elétricos são apresentados e analisados, permitindo tanto análises técnicas detalhadas quanto avaliações operacionais de alto nível.

Os gráficos de consumo oferecem visualização multi-escala (horária, diária e mensal) com recursos de agregação automática, facilitando a identificação de padrões de utilização e anomalias. A Figura 9 demonstra o gráfico de consumo mensal simulado.

Figura 9 - Gráfico de consumo mensal

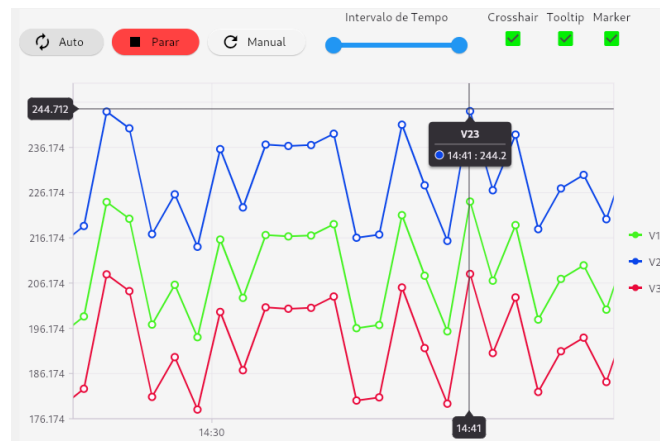


Fonte: Autoria própria

A implementação de filtros temporais nestes gráficos possibilita análises comparativas, fundamentais para decisões estratégicas relacionadas à eficiência energética.

Os gráficos de tensão implementam visualização simultânea das três fases com codificação por cores, permitindo identificação imediata de desequilíbrios e eventos transitórios. A figura 10 demonstra a funcionalidade de curvas de tensão:

Figura 10 - Gráfico de tensão de linha



Fonte: Autoria própria

Os recursos de zoom e pan incorporados possibilitam análises detalhadas de perturbações específicas, facilitando a correlação com parâmetros de qualidade de energia.

Para correntes, a implementação de escalas adaptativas facilita a visualização simultânea de cargas com diferentes magnitudes. A Figura 11 apresenta o gráfico de correntes trifásicas durante um evento de sobrecarga, demonstrando a capacidade do sistema em destacar visualmente eventos críticos.

Figura 11 - Gráfico de corrente



Fonte: Autoria própria

Além disso, os gráficos contam com recursos interativos como crosshair, tooltip e marker, que aprimoram a experiência de análise ao permitir a inspeção precisa de valores em pontos específicos, facilitando a identificação de variações súbitas.

O sistema de visualização de alarmes integra elementos gráficos com indicadores numéricos. A Figura 12 ilustra o histórico temporal de alarmes por categoria, evidenciando a distribuição temporal e a classificação por tipo de evento.

Figura 12 - Gráfico de alarmes



Fonte: Autoria própria

A arquitetura gráfica implementada demonstrou robustez mesmo ao processar grandes volumes de dados, mantendo responsividade satisfatória. Os resultados evidenciam que a combinação de Flutter com bibliotecas especializadas de visualização proporciona uma plataforma eficaz para interfaces gráficas industriais, equilibrando rigor técnico, facilidade de interpretação e desempenho consistente, como pode ser observado na página completa de gráficos apresentada no apêndice H.

5.4 Conexão Modbus

Esta seção apresentará o resultado do processo de configuração e comunicação através do protocolo Modbus TCP.

5.4.1 Parâmetros de conexão

A conexão modbus foi desenvolvida visando a conexão com um único multimedidor de grandezas elétricas através do protocolo Modbus TCP com aquisições de dados em intervalos de 1 minuto. Os parâmetros de conexão são definidos através do menu de configuração como indicado na Figura 13:

Figura 13 - Página de configuração de conexão Modbus

Fonte: Autoria própria

Os campos necessários para que seja feita a conexão computador/Multimedidor são:

ID (Unit ID ou Slave ID)

É o número que identifica o dispositivo Modbus dentro da rede. Ele permite que o mestre saiba para qual escravo enviar a requisição, especialmente em redes que compartilham múltiplos dispositivos. Os valores geralmente vão de 1 a 247.

IP Address

É o endereço exclusivo atribuído ao dispositivo na rede Ethernet. Ele funciona como o “RG” digital do equipamento, permitindo que o cliente encontre e se comunique com ele. Cada dispositivo conectado ao sistema precisa de um IP único que esteja na mesma sub-rede para que a comunicação ocorra corretamente.

Porta

Porta refere-se ao número de porta TCP usada para estabelecer a comunicação entre cliente e servidor. No Modbus TCP, a porta padrão é a 502, e é por meio dela que as mensagens são enviadas e recebidas. É possível modificar esse número em alguns sistemas, mas a 502 é amplamente utilizada e recomendada para garantir compatibilidade.

Tempo limite de conexão (Connection Timeout)

Define o período máximo que o cliente aguardará para estabelecer comunicação com o servidor. Se a conexão não ocorrer dentro desse tempo, a tentativa é abortada. Esse parâmetro é essencial para evitar que o sistema fique travado tentando acessar dispositivos offline ou com problemas de rede.

Tempo limite de resposta (Response Timeout)

Especifica quanto tempo o cliente espera pela resposta do servidor após o envio de uma requisição. Se esse tempo for excedido, ocorre um erro de timeout e a operação é considerada falha. Esse parâmetro deve ser ajustado conforme a velocidade da rede e o tempo de processamento dos dispositivos, garantindo o equilíbrio entre tempo de processamento e agilidade na comunicação.

Estes dados são salvos através da função Shared Preferences do dart, forma esta simples e eficiente para armazenar dados localmente no dispositivo sem a necessidade de banco de dados, ficando disponíveis mesmo após a aplicação ser fechada.

Teste de erro (CRC)

O teste CRC no Modbus RTU é uma verificação de integridade que usa o algoritmo CRC-16. O transmissor calcula o CRC sobre todos os bytes da mensagem (exceto os dois bytes finais), anexa os dois bytes do CRC (low byte primeiro) e o receptor recalcula e compara; se diferir, descarta o quadro. Isso detecta erros de transmissão simples de forma rápida e eficiente.

5.4.2 *Teste funcional*

Como mencionado anteriormente, utilizou-se um microcontrolador ESP32 para simular o multimedidor ABB M1M criando um cliente Modbus com o mesmo conjunto de variáveis com seu respectivo endereços e tipos. A figura 14 demonstra um exemplo de aquisição de dados de tensão de fase:

Figura 14 - Exemplo de leitura dos registradores do multimetro

Example

Request (read Phase voltage L1~L3)

Slave address	0x01
Function code	0x03
Start address, high byte	0x5B
Start address, low byte	0x02
No. of registers, high byte	0x00
No. of registers, low byte	0x06
Error check (CRC), high byte	0x77
Error check (CRC), low byte	0x2C

Response frame

Slave address	0x01
Function code	0x03
Byte count	0x0C
Value of registers	0x00 00 08 CA 00 00 08 CB 00 00 08 CC
Error check (CRC), high byte	0x9F
Error check (CRC), low byte	0x32

Autor: ABB. Modbus Manual – M1M Power Meters

Neste exemplo a partir da requisição dos dados os registradores L1, L2 e L3 é obtido respectivamente 0x08CA, 0x08CB e 0x08CC em hexadecimal, que quando convertidos para base decimal e multiplicados pela resolução do registrador (0,1) as tensões obtidas serão 225,0 V; 225,1 V e 225,2 V. Para validar o correto recebimento dos dados transmitidos pelo microcontrolador ESP32 à aplicação desenvolvida, empregou-se a funcionalidade de monitoramento em tempo real disponibilizada pela interface gráfica.

Os mostradores dinâmicos exibem os valores atualizados continuamente, permitindo uma análise visual imediata, conforme ilustrado na Página de Monitoramento apresentada no Apêndice G. Para garantir a observação das variações e simular alterações nos parâmetros monitorados, foram utilizados integradores de valores aleatórios. Essa abordagem viabilizou a validação da comunicação entre dispositivos, tornando perceptível a atualização contínua dos dados em cada variável representada. A Figura 15 apresenta a interface de monitoramento em tempo real utilizada na aplicação, evidenciando os indicadores visuais que representam as variáveis recebidas pelo microcontrolador ESP32.

Figura 15 - Mostradores dinâmicos de dados



Fonte: Autoria própria

Através dos mostradores dinâmicos, foi possível acompanhar as variações dos parâmetros simulados, graças à integração de valores aleatórios, conforme mencionado anteriormente. Essa representação gráfica contribui significativamente para a validação funcional do sistema, oferecendo suporte à análise do comportamento das variáveis ao longo do tempo e facilitando a identificação de atualizações e eventuais falhas na comunicação.

5.5 Análise de qualidade

A funcionalidade de análise de qualidade implementada no aplicativo tem como principal objetivo avaliar o comportamento da rede elétrica a partir dos dados registrados no banco de dados. Utilizando algoritmos de interpretação e cruzamento de informações, o sistema identifica padrões, variações e possíveis anomalias nas medições das variáveis elétricas. O modelo de relatório que pode ser gerado pela aplicação encontra-se descrito no Apêndice B, em formato PDF, servindo como referência para a apresentação estruturada dos resultados da análise de qualidade. A página de análise de qualidade está apresentada integralmente no Apêndice I.

Para ampliar a flexibilidade e autonomia da análise, foi incorporado à interface do aplicativo um conjunto de controles que permite ao usuário selecionar o banco de dados SQLite de origem, definir o intervalo temporal da análise por meio de um seletor de datas, e especificar

a tensão nominal de fase correspondente ao sistema monitorado. Esses parâmetros oferecem maior adaptabilidade às diferentes condições operacionais e contextos de aplicação, tornando a ferramenta mais robusta e versátil.

Os testes de validação da funcionalidade foram conduzidos utilizando dois bancos de dados distintos: um proveniente do smart campus Unesp (Zarpellon,2024), com registros reais de consumo energético da universidade, e outro composto por dados simulados gerados por um microcontrolador ESP32. Essa abordagem permitiu verificar a eficácia da análise tanto em cenários práticos quanto em ambientes simulados, garantindo que o sistema consegue identificar com precisão padrões relevantes e potenciais inconsistências nos dados, independentemente da fonte.

5.5.1 *Análise de dados simulados*

Para validação da proposta e verificação da capacidade analítica do sistema, foram utilizados dados simulados gerados aleatoriamente por meio da plataforma ESP32. Os valores atribuídos às variáveis elétricas foram configurados para apresentar variações fora dos parâmetros ideais de operação para redes trifásicas com tensão nominal de fase de 127 V, permitindo identificar desvios e instabilidades relevantes ao monitoramento. As tabelas 4 e 5 apresentam respectivamente as faixas de tensão de fase e linha simuladas:

Tabela 3 - Faixas de tensão de fase simuladas

Fase	Faixa simulada
L1	99 V a 134 V
L2	97 V a 132 V
L3	97 V a 132 V

Fonte: Autoria própria

Tabela 4 - Faixa de tensão de linha simuladas

Linha	Faixa Simulada
L1	190 V a 225 V
L2	210 V a 245 V
L3	174 V a 209 V

Fonte: Autoria própria

Esses valores representam variações em relação ao padrão de 127 V, simulando cenários de tensão precária e críticas. Tais desvios podem indicar sobrecargas locais, deficiências no

fornecimento ou variações sazonais da rede. As tensões de linha idealmente deveriam se manter em torno de 220 V. As faixas simuladas mostram variações de até ± 25 V, o que permite identificar desequilíbrio entre fases, característica que pode impactar diretamente a performance de motores, equipamentos sensíveis e a eficiência energética da instalação. Os demais parâmetros a serem simulados e usados para realizar a análise foram o THD de tensão e o fator de potência, configurados para serem gerados nos seguintes intervalos:

- THD L1: 7% a 12%
- THD L2: 6,5% a 11,5%
- THD L3: 10,3% a 17,3%
- FP: 0,8 até 1,00

Segundo o módulo 8 do PRODIST, o limite aceitável de THD de tensão é de até 10% e o fator de potência entre 0,92 e 1 indutivo ou capacitivo. Com isso, os dados simulados excedem esse limite em várias situações, o que permitiu testar a capacidade do sistema em detectar irregularidades e perdas de qualidade na rede. Os resultados da análise, apresentados no Apêndice A, evidenciam que o sistema foi capaz de estimar corretamente todos os índices exigidos pela norma, conforme o esperado.

5.5.2 Análise de dados reais

Este tópico trata sobre o recurso de análise de energia, apresentando sua aplicação, funcionalidades e importância para a avaliação da qualidade do fornecimento elétrico usando dados reais do Smart Campus da Unesp Sorocaba. Esta base de dados oferece registros históricos de medições elétricas da infraestrutura universitária, proporcionando um conjunto robusto e representativo para validação das funcionalidades analíticas implementadas.

5.5.2.1 Metodologia de Coleta e Estruturação dos Dados

O conjunto de dados utilizado abrange todo o ano de 2024; os registros originais foram coletados com amostragem de 1 minuto e, para análise estatística, foram integralizados em intervalos de 10 minutos, resultando em 1.008 amostras por semana. A análise foi realizada calculando-se a média de cada conjunto semanal de 1.008 amostras, assegurando conformidade com os critérios normativos e maior confiabilidade dos resultados. Entretanto, é importante destacar que, em cenários operacionais típicos, esse nível de detalhamento nas medições originais nem sempre está disponível. Essa metodologia de amostragem estratificada por 7 dias

por mês garante representatividade estatística adequada para a avaliação dos indicadores de qualidade de energia, contemplando variações sazonais e operacionais típicas de instalações universitárias.

Os dados foram organizados em arquivos CSV individuais correspondentes aos diferentes setores de consumo da infraestrutura universitária, refletindo a segmentação natural do sistema elétrico do campus:

- Consumo A, B e C: Biblioteca, blocos acadêmicos principais
- Consumo D, E, F e G: Complexo de laboratórios e salas de aula
- Consumo ar-condicionado D, E, F e G: Sistema de climatização
- Consumo H: Refeitório, salas de aula e dos professores centro de convivência, laboratórios de elétrica e ambiental
- Consumo ar condicionado H: Sistema de climatização

Cada arquivo foi importado para bancos de dados SQL individuais, mantendo a integridade e rastreabilidade dos dados por setor, facilitando análises comparativas e identificação de padrões específicos por tipo de carga.

5.5.2.2 Parâmetros Analisados e Limitações

As variáveis disponíveis para análise incluíram:

Distorção Harmônica Total (THD) de Tensão: Medições por fase permitindo avaliação da qualidade da forma de onda conforme limites estabelecidos pela Resolução Normativa nº 956/2021 da ANEEL.

Tensões de Fase: Registros trifásicos RMS instantâneos (VAN, VBN, VCN) para avaliação de adequação, precariedade e criticidade conforme faixas de tensão de referência.

Fator de Potência: Valores instantâneos por fase, com análise baseada na média aritmética dos fatores de potência em cada instante de medição.

A seguir é destacado os índices que não foram analisados por não estarem disponíveis no site do smart campus:

Ausência de Tensões de Linha: A indisponibilidade de medições diretas das tensões de linha (VAB, VBC, VCA) limitou inicialmente a capacidade de calcular o fator de desequilíbrio

de tensão (FD%). Para contornar esta limitação em futuras análises, foi implementada a funcionalidade de estimativa de tensões de linha a partir das tensões de fase utilizando a conversão retangular polar, conforme descrito na metodologia de cálculo desenvolvida.

Indisponibilidade de Dados de Frequência: A ausência de registros de frequência não comprometeu as análises principais de qualidade de energia, uma vez que os demais parâmetros disponíveis são suficientes para caracterizar a maior parte dos distúrbios relevantes em sistemas de distribuição de baixa tensão.

5.5.2.3 Processamento e Análise Estatística

O processamento dos dados seguiu metodologia padronizada, com cada mês sendo analisado individualmente para identificação de padrões temporais e sazonais. Os algoritmos implementados calcularam os indicadores estatísticos DTT95% e DRP/DRC conforme procedimentos estabelecidos pelo PRODIST, utilizando as 1.008 amostras semanais para determinação dos percentis apropriados.

Para o fator de potência, adotou-se a abordagem de análise da média dos valores instantâneos por fase, proporcionando uma avaliação representativa do comportamento geral do sistema em cada período de medição. Esta metodologia é adequada para identificação de tendências de operação e conformidade regulatória.

5.5.2.4 Resultados da Análise

Os resultados obtidos através da análise dos dados reais do projeto Smart Campus UNESP demonstraram a eficácia das funcionalidades implementadas em identificar padrões de qualidade de energia em instalações reais. As análises revelaram características distintas entre os diferentes setores de consumo, evidenciando a importância da segmentação por tipo de carga para diagnósticos precisos.

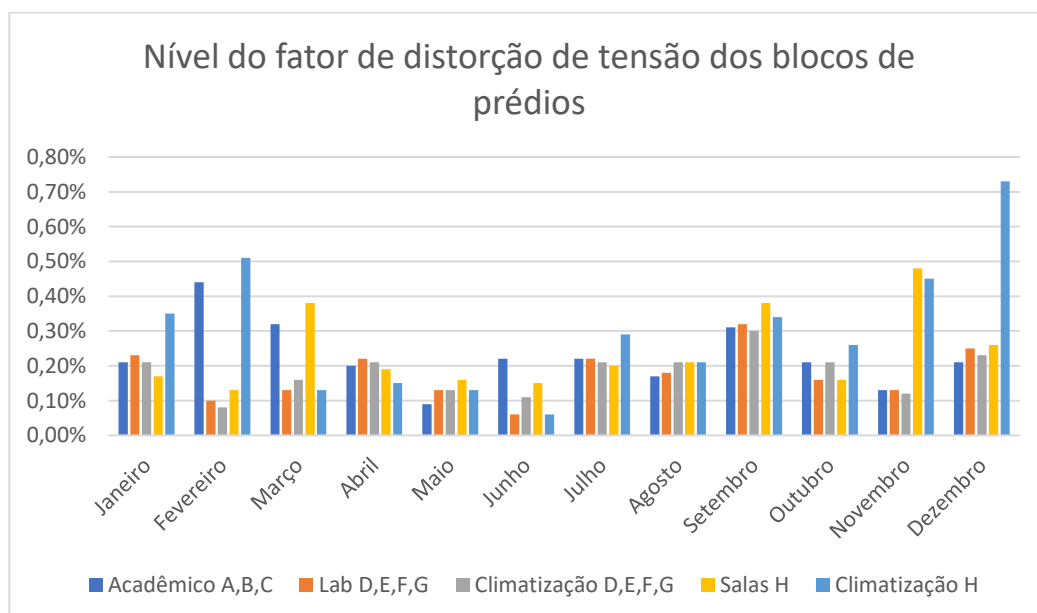
A implementação da estimativa de tensões de linha mostrou-se fundamental para viabilizar o cálculo do fator de desequilíbrio, fornecendo informações valiosas sobre a operação trifásica do sistema mesmo na ausência de medições diretas. Esta funcionalidade amplia significativamente a capacidade analítica do sistema quando aplicado a bases de dados com limitações instrumentais.

As subseções seguintes apresentam os resultados detalhados para cada bloco de consumo analisado, com as respectivas tabelas de indicadores sendo disponibilizadas no apêndice C. Esta estruturação permite análises comparativas entre diferentes tipos de carga e identificação de oportunidades de melhoria específicas para cada setor da infraestrutura universitária.

5.5.2.5 Fator de distorção

A figura 16 a seguir demonstra a comparação do nível de distorção dos blocos ao longo ano de 2024:

Figura 16 - Comparativo de amostras de fator de distorção do smart campus Unesp



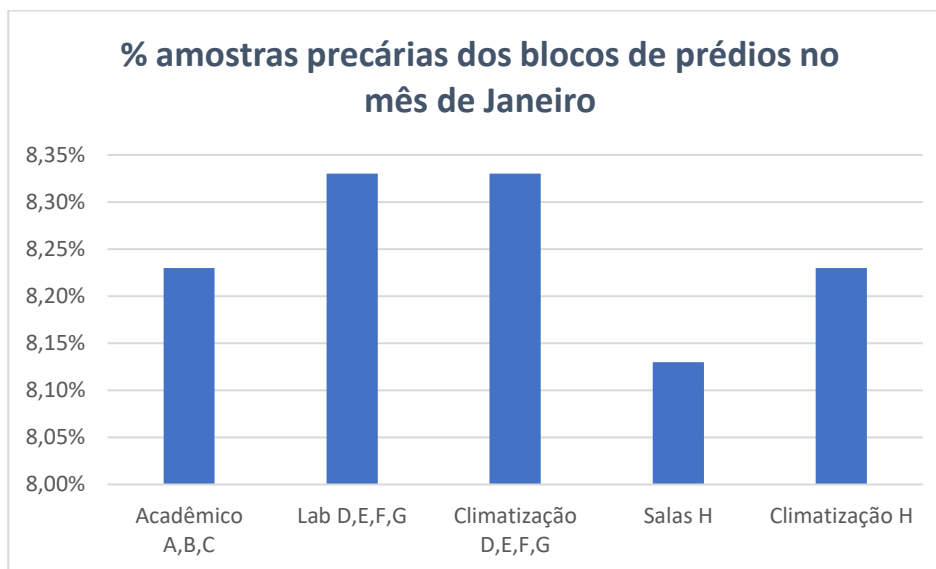
Fonte: Autoria própria

A análise dos dados dos blocos revelou características consistentes de qualidade de energia ao longo do período analisado. O sistema apresentou operação equilibrada entre as três fases durante todos os meses de 2024, não registrando valores de fator de desequilíbrio (FD%) acima do limite regulatório de 3% estabelecido pelo PRODIST.

5.5.2.6 DRP - Duração Relativa de Transgressão de Tensão Precária

A figura 17 a seguir compara o percentual de amostras de tensão encontradas em nível precário com tensão de referência de 127 V:

Figura 17 - Amostras precárias do smart campus Unesp



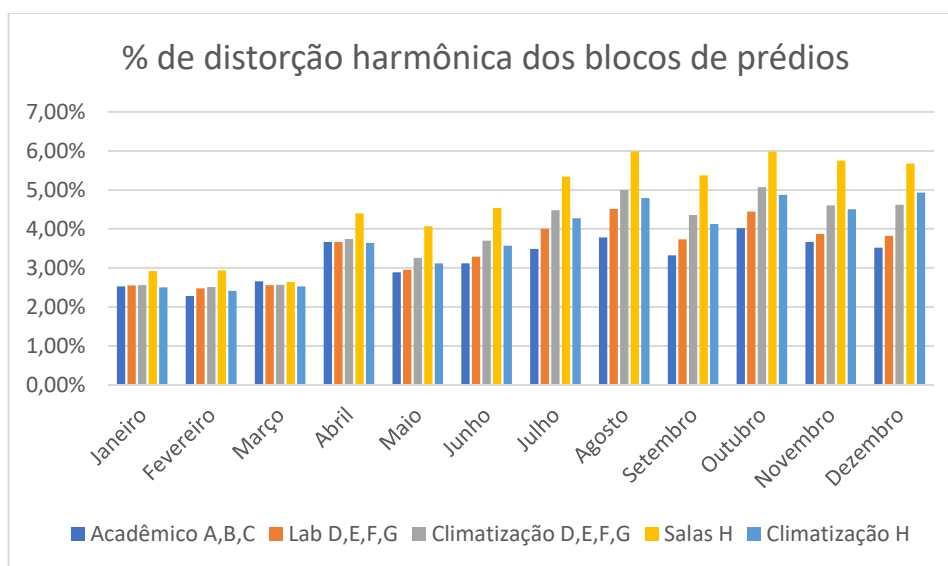
Fonte: Autoria própria

A análise da tensão em regime permanente mostrou conformidade em geral; apenas janeiro registrou DRP acima do ideal, indicando que os episódios de subtensão foram pontuais e que, na maior parte do ano, a amplitude eficaz da tensão permaneceu estável.

5.5.2.7 Distorção harmônica total

A figura 18 a seguir exibe um gráfico comparativo do índice de distorção harmônica dos blocos do período amostrado:

Figura 18 - Comparativo de amostras de distorção harmônica do smart campus Unesp



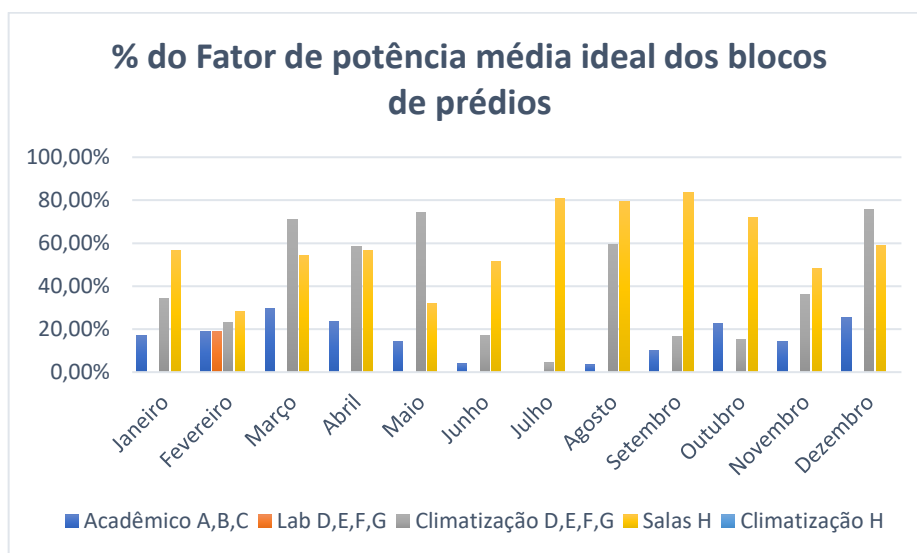
Fonte: Autoria própria

A qualidade da forma de onda mostrou-se exemplar, com todas as amostras de distorção harmônica total de tensão (DTT%) mantendo-se dentro dos limites adequados estabelecidos pela Resolução Normativa nº 956/2021 da ANEEL em todos os meses analisados. Este resultado indica ausência de distorções significativas que poderiam comprometer o funcionamento de equipamentos sensíveis.

5.5.2.8 Comparativo do fator de potência

O comportamento do fator de potência apresentou variações sazonais significativas, com percentuais de amostras em condição ideal variando consideravelmente ao longo do ano. A figura 19 a seguir demonstra o comparativo do percentual de amostras do fator de potência média em nível ideal:

Figura 19 - Comparativo de amostras de fator de potência do smart campus Unesp



Fonte: Autoria própria

O bloco dos prédios D, E, F e G apresentou comportamento atípico, com somente o mês de fevereiro apresentando amostras em níveis ideais (18,95%). Este comportamento sugere cargas com características predominantemente indutivas ou capacitivas, típicas de equipamentos de laboratório. De maneira geral o bloco de salas H apresentou comportamento mais consistente em comparação aos demais setores. O fator de potência do sistema de climatização do bloco H apresentou comportamento atípico, com 0% de amostras em condição ideal durante todos os meses do ano, ao contrário do sistema do bloco dos prédios D,E,F e G, que chegou a apresentar 75% de amostras em estado ideal no mês de dezembro. Este resultado

sugere características operacionais específicas do sistema de climatização, possivelmente relacionadas ao tipo de equipamento utilizado ou estratégias de controle implementadas, indicando a necessidade de investigação mais detalhada para otimização da eficiência energética deste setor.

Em um caso real, o setor não necessariamente requer medidas de correção do fator de potência se o valor medido no medidor geral de saída não apresentar irregularidades; entretanto, a análise indica que este setor é um bom candidato à adoção de estratégias de correção, caso elas se mostrem necessárias.

Ressalta-se, contudo, que a conformidade normativa não implica, necessariamente, a garantia de elevada qualidade do fornecimento. Nesse sentido, o sistema pode ser empregado como instrumento de aprimoramento da qualidade, mesmo quando os parâmetros observados se mantêm em conformidade com os requisitos normativos.

A validação com dados reais complementou satisfatoriamente os testes realizados com dados simulados, demonstrando a robustez e aplicabilidade prática das funcionalidades analíticas desenvolvidas. A capacidade de processar dados de diferentes fontes e contornar limitações instrumentais confirma a versatilidade da solução implementada para aplicações de monitoramento energético em ambientes acadêmicos e industriais.

6 CONCLUSÃO

O desenvolvimento do aplicativo para análise e gestão de dados de energia elétrica com comunicação Modbus TCP/IP atingiu plenamente os objetivos propostos, resultando em uma solução tecnológica robusta e versátil para supervisão energética industrial. A escolha do framework Flutter com linguagem Dart mostrou-se acertada, proporcionando uma plataforma multiplataforma eficiente, com interface responsiva e capacidade de processamento adequada para grandes volumes de dados em tempo real.

A implementação da comunicação Modbus TCP demonstrou excelente compatibilidade com equipamentos industriais, validada através dos testes com microcontrolador ESP32 simulando um multimedidor ABB M1M. Esta abordagem permitiu validar todo o fluxo de aquisição, processamento e armazenamento de dados antes da implementação em ambiente real, reduzindo riscos e custos de desenvolvimento.

A integração com Firebase Realtime Database proporcionou uma solução escalável para armazenamento em nuvem, sincronização de dados e gestão de usuários, enquanto o sistema de alarmes implementado oferece monitoramento proativo de eventos críticos. Estas funcionalidades ampliam significativamente a aplicabilidade da solução em ambientes industriais distribuídos.

A validação com dados reais do projeto smart campus Unesp confirmou a eficácia dos algoritmos de análise de qualidade de energia desenvolvidos. O sistema demonstrou capacidade de processar adequadamente as amostras semanais conforme estabelecido pelo PRODIST, calculando corretamente os indicadores DRP, DRC, DTT95% e FD95%. A implementação da estimativa de tensões de linha por meio da transformação entre coordenadas polares e retangulares mostrou-se fundamental para contornar limitações instrumentais, permitindo a conversão de magnitudes e ângulos em componentes cartesianas e vice-versa. Essa abordagem ampliou a versatilidade analítica do sistema, facilitando cálculos vetoriais e a interpretação precisa das relações fasoriais entre as grandezas elétricas.

Os resultados das análises dos diferentes setores de consumo da infraestrutura universitária evidenciaram a importância da segmentação por tipo de carga para diagnósticos mais precisos, fornecendo insights mais detalhados sobre o comportamento das instalações. Ressalta-se, contudo, que, caso os indicadores no Ponto de Conexão Comum (PCC) estejam

em conformidade, não há caracterização de não conformidade regulatória. Ainda assim, essa segmentação se mostra valiosa, pois permite identificar de forma mais ágil e direcionada eventuais falhas ou anomalias quando problemas de qualidade de energia são observados. As variações identificadas no comportamento do fator de potência entre sistemas de climatização, laboratórios e blocos acadêmicos demonstram o potencial da ferramenta para identificação de oportunidades específicas de otimização energética.

O projeto desenvolvido contribui significativamente para o avanço das ferramentas de supervisão energética, oferecendo uma solução integrada que combina aquisição de dados industriais, análise técnica especializada e interface moderna. Um dos principais diferenciais da proposta é o uso de ferramentas de código aberto, que além de ampliarem a flexibilidade e a possibilidade de customização, reduzem significativamente os custos de implementação. A metodologia adotada e os resultados obtidos validam a aplicabilidade da solução tanto em ambientes acadêmicos quanto industriais, representando um avanço importante na democratização do acesso a ferramentas avançadas e de baixo custo para a gestão energética.

Como trabalhos futuros, sugere-se a implementação de algoritmos de aprendizado de máquina para predição de eventos de qualidade de energia e o desenvolvimento de funcionalidades de controle automático baseadas nos indicadores analisados. Além disso, propõe-se a ampliação do sistema para suportar a integração de um número maior de multimedidores, com possibilidade de configuração flexível de registradores. Por fim, destaca-se a relevância de investigar mecanismos de comunicação direta com a companhia elétrica, de modo a favorecer tanto o monitoramento contínuo quanto a adoção de estratégias colaborativas de gestão da qualidade de energia.

REFERÊNCIAS

- AGÊNCIA NACIONAL DE ENERGIA ELÉTRICA (Brasil). Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional – PRODIST: Módulo 8 – Qualidade do Fornecimento de Energia Elétrica. Resolução Normativa nº 956. Brasília: ANEEL, 2021.
- ALEXANDER, C. K.; SADIKU, M. N. O. Fundamentos de Circuitos Elétricos. 5. ed. Porto Alegre: AMGH, 2013.
- ALMEIDA, D. F. U.; Análise da qualidade de energia no sistema elétrico da Universidade Tecnológica Federal do Paraná – campus Pato Branco. Orientador: Giovanni Alfredo Guarneri. 2017. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica) – Universidade Tecnológica Federal do Paraná, Pato Branco, 2017.
- BEIJING, ABB. Modbus Manual – M1M Power Meters. Version 1.3B. ABB LV Installation Materials Company Limited, 2025.
- BIESSEK, A. Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter and Dart 2. Birmingham: Packt Publishing, 2019.
- CLARKE, G.; REYNDERS, D.; WRIGHT, E. Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems. Oxford: Newnes, 2004.
- CRAVO, E. Sistema supervisorio: o que é e quais as vantagens. 2022. Disponível em: <https://blog.kalatec.com.br/sistema-supervisorio>. Acesso em: 6 out. 2025
- DUGAN, R. C.; MCGRANAGHAN, M. F.; SANTOSO, S.; BEATY, H. W. Electrical Power Systems Quality. 2. ed. New York: McGraw-Hill Education, 1996.
- FLUTTER DEV TEAM. Flutter Documentation State Management. 2023. Disponível em: <https://flutter.dev/docs/development/data-and-backend/state-mgmt/intro>. Acesso em: 12 mar. 2025.
- INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. IEEE Recommended Practice for Monitoring Electric Power Quality. New York: IEEE, 2019
- KAGAN, N.; ROBBA, E. J.; SCHMIDT, H. P. Estimção de indicadores de qualidade da energia elétrica. São Paulo: Blucher, 2009.
- KOPEC, D. Dart for Absolute Beginners. New York: Apress, 2014. *E-book*.
- MELO, G. S. Avaliação do impacto da qualidade de energia elétrica na produção industrial: Proposta de metodologia 2008. Dissertação (Mestrado em Engenharia Elétrica) - Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Pernambuco, Recife, 2008.
- MENDONÇA, G.; SOUZA, P. A.; DE CÁSSIO, E.; Análise da qualidade de energia na indústria. Revista Mythos, [S. l.], v. 12, n. 1, p. 76–86, 2021. DOI:

10.36674/mythos.v13i1.448. Disponível em:
<https://ojs.periodicos.unis.edu.br/mythos/article/view/448>. Acesso em: 24 ago. 2025.

RIBEIRO DOS SANTOS, F. H.; BECERRA SABLON, V. I.; Sistema de monitoramento, testes e medição da qualidade da energia através de redes móveis. In: CONGRESSO NACIONAL DE INICIAÇÃO CIENTÍFICA – CONIC SEMESP, 19., 2019, São Paulo. Anais [...]. São Paulo: SEMESP, 2019. Tema: A pesquisa de mãos dadas com os Direitos Humanos.

RIBEIRO, T. N.; BUENO, A. F. M.; Monitoramento e Diagnóstico da Qualidade da Energia Elétrica Fornecida a um Consumidor Industrial Atendido em Tensão de Distribuição. Belo Horizonte, 1997.

ROCHA, G. S.; NASCIMENTO, P. H. G. do; SILVA, D. O.; PIMENTA, T. C.; Protótipo de sistema industrial para monitoramento de parâmetros elétricos da qualidade de fornecimento de energia elétrica. Revista de Gestão e Secretariado – GeSec, São José dos Pinhais, Paraná, v. 15, n. 6, p. 1-14, 2024. DOI: 10.7769/gesec.v15i6.3755. Disponível em:
<https://ojs.revistagesec.org.br/secretariado/article/view/3651>. Acesso em: 10 ago. 2025

SWALES, A.; Open Modbus/TCP Specification. Schneider Electric, v. 29, n. 3, p. 19, 1999.

TAMBOLI, S.; RAWALE, M.; THORAIET, R.; AGASHE, S.; Implementation of Modbus RTU and Modbus TCP communication using Siemens S7-1200 PLC for batch process, International conference on smart technologies and management for computing, communication, controls, energy and materials (ICSTM). IEEE, 2015. p. 258-263.

TURBO PARTNERS. Vantagens e Desvantagens do Figma: Uma Análise Completa. 2025. Disponível em: <https://blog.turbopartners.com.br/vantagens-e-desvantagens-do-figma/>. Acesso em: 10 jun. 2025.

VOLTIMUM BRASIL. Multimedidor de energia: o que são e para que servem? 2023. Disponível em: <https://www.voltimum.com.br/noticias/multimedidor-de-energia>. Acesso em: 12 ago. 2025.

ZAMMETTI, F. Practical Flutter: Improve your Mobile Development with Google's Latest Open-Source SDK. Berkeley: Apress, 2019.

ZARPELLON, B. O. Sistema integrado para monitoramento de variáveis energéticas do Instituto de Ciência e Tecnologia de Sorocaba no contexto de Smart Campus. 2024. Dissertação (Mestrado em Engenharia Elétrica) – Instituto de Ciência e Tecnologia, Universidade Estadual Paulista, Sorocaba, 2024.

APÊNDICE A – Tabela de resultado da análise de dados simulados para 1008 amostras

Tensão nominal de Fase A	Adequada:474	Precária:214	Crítica:320
Tensão nominal de Fase B	Adequada:451	Precária:200	Crítica:357
Tensão nominal de Fase C	Adequada:448	Precária:201	Crítica:359
Fator de desequilíbrio	Adequada:0%	Precária:100%	
Interrupção temporária de tensão	Eventos:0		
Afundamento temporário de tensão	Eventos:1		
Elevação temporária de tensão	Eventos:0		
DRP A	21,23%		
DRC A	31,75%		
DRP B	19,84%		
DRC B	35,42%		
DRP C	19,94%		
DRP C	35,62%		
Fator de potência	Adequada:435	Precária:572	
Distorção harmônica Total A	Adequada:71,8%	Precária:28,2%	
Distorção harmônica Total B	Adequada:76,37%	Precária:23,63%	
Distorção harmônica Total C	Adequada:0%	Precária:100%	
Frequência	Adequada:1008	Precária:0	

APÊNDICE B – PDF de representação do relatório de análise de qualidade

Relatório de Análise de Qualidade Elétrica

Usuário: Daniel Francisco

Banco de dados: Local_data.db

Data de criação: 22/08/2025

Mês de análise: Janeiro 2024

Tensão nominal de fase: 127 V

Tabela 1 - Condição das amostras de tensão de fase

Tensão	Adequado	Precário	Critico
Fase A	938	70	0
Fase B	933	75	0
Fase C	924	84	0

Tabela 2 - Nível de desequilíbrio entre fases

Fator de desequilíbrio	Adequado	Precário
FD95%	100.00%	0.00%

Tabela 3 - Eventos de temporários de tensão

Eventos	Quantidade de eventos
Interrupção	0
Afundamento	1
Elevação	0

Tabela 4 - Duração relativa de transgressão de tensão precária - DRP

DRP	Quantidade de eventos
Fase A	6.94%
Fase B	7.44%
Fase C	8.33%

Tabela 5 - Duração relativa de transgressão de tensão precária - DRC

DRC	Quantidade de eventos
Fase A	0.00%
Fase B	0.00%
Fase C	0.00%

Tabela 6 - Fator de potência

Fator de potência	Adequado	Precário
FP	283	725

Tabela 7 - Distorção harmônica total

Distorção harmônica total	Adequado	Precário
DTT 95% - A	100.00%	0.00%
DTT 95% - B	100.00%	0.00%
DTT 95% - C	100.00%	0.00%

Tabela 8 - Desequilíbrio de frequência

Frequência	Adequado	Precário
Hz	0	0

Resultado

Energia NÃO adequada: Duração relativa de transgressão precária fase A: mais de 3% das amostras estão em estado precário, Duração relativa de transgressão precária fase B: mais de 3% das amostras estão em estado precário, Duração relativa de transgressão precária fase C: mais de 3% das amostras estão em estado precário, Fator de potência precário: foi identificado fator de potência abaixo de 0,92 indutivo ou capacitivo

APÊNDICE C – Link de download dos relatórios mensais

https://drive.google.com/file/d/1MKRvWUF3yprmN03pPlnNRaRv3X4ZGg0H/view?usp=drive_link

APÊNDICE D – Link de download do projeto dart

https://drive.google.com/file/d/1d4Jq5vKOYqoxufLIUQa3WhAznx6Lcsfn/view?usp=drive_link

APÊNDICE E – Tela de login



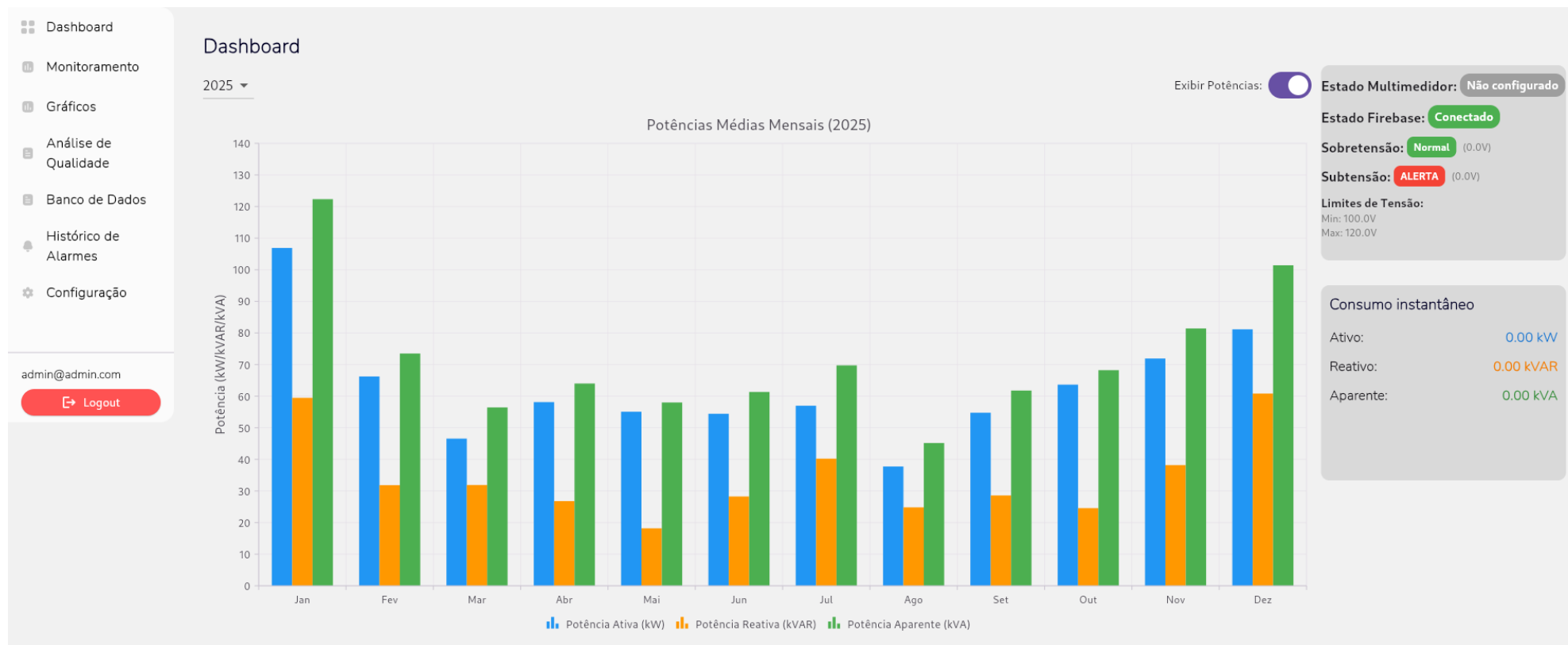
Usuário

Senha

Log in

Esqueci minha senha

APÊNDICE F – Tela inicial



APÊNDICE G – Página de monitoramento



APÊNDICE H – Página de gráficos



APÊNDICE I – Página de análise de qualidade

Dashboard

Monitoramento

Gráficos

Análise de Qualidade

Banco de Dados

Histórico de Alarmes

Configuração

admin@admin.com

Logout

Análise de Qualidade

Tensão nominal Fase A	Adequada 0	Precária 0	Crítica 0
Tensão nominal Fase B	Adequada 0	Precária 0	Crítica 0
Tensão nominal Fase C	Adequada 0	Precária 0	Crítica 0
Fator de desequilíbrio - FD95%	Adequada 0.00%	Precária 0.00%	
Interrupção temporária de Tensão	Eventos 0		
Afundamento temporário de Tensão	Eventos 0		
Elevação temporária de Tensão	Eventos 0		
Duração Relativa da Transgressão de Tensão Precária - DRP A	Eventos 0.00%		

Resultado da Análise

Intervalo de análise

☐ Habilitar Navegação

September 2025

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

☐ Estimar tensões de linha (V12, V23, V31) para cálculo fator de desequilíbrio

Selecionar banco de dados

Nenhum banco de dados carregado

Nome do Usuário:

Digite seu nome

Tensão nominal de fase: 220 V

Iniciar análise

PDF Salvar Relatório PDF

APÊNDICE J – Página de Banco de dados de alarmes

Dashboard

Monitoramento

Gráficos

Análise de Qualidade

Banco de Dados

Histórico de Alarmes

Configuração

admin@admin.com

Logout

Histórico de Alarmes

Exportar Excel

Atualizar

Limpar Histórico

Total de alarmes: 7 | Use os filtros nas colunas para refinar a busca

☐ Data	Hora	Tipo de Alarme	Variável	Valor	Limite Min	Limite Max
Contains	Contains	Contains	Contains	Contains	Contains	Contains
☐ 29/08/2025	13:28	sobretensao	V1N	245.5	200	240
☐ 29/08/2025	13:28	sobretensao	V1N	245.5	200	240
☐ 29/08/2025	13:28	sobretensao	V1N	245.5	200	240
☐ 29/08/2025	13:28	sobretensao	V1N	245.5	200	240
☐ 29/08/2025	13:28	sobretensao	V1N	245.5	200	240
☐ 29/08/2025	13:28	corrente	L1	15.8	0	15
☐ 29/08/2025	13:28	sobretensao	V1N	245.5	200	240

APÊNDICE K – Página de configuração – Firebase

The screenshot displays a web application interface with a sidebar on the left and a main content area. The sidebar contains a list of menu items: Dashboard, Monitoramento, Gráficos, Análise de Qualidade, Banco de Dados, Histórico de Alarmes, and Configuração. The 'Configuração' item is selected and highlighted. Below the menu items, the user's email 'admin@admin.com' is displayed, and a red 'Logout' button is visible. The main content area is titled 'Configuração' and features four tabs: MODBUS, FIREBASE (which is active and highlighted in blue), USUÁRIO, and ALARMES. Below the tabs, there is a form for configuring Firebase. It includes a label 'Url Firebase:' followed by a text input field containing the URL 'https://energy-monitor-aa31a-default-rtdb.firebaseio.co'. Below the input field are two buttons: a light purple 'Download' button and a blue 'Testar Conexão' button. At the bottom of the form, the 'Status Firebase:' is shown as 'Conectado' in a green pill-shaped button.

Dashboard

Monitoramento

Gráficos

Análise de Qualidade

Banco de Dados

Histórico de Alarmes

Configuração

admin@admin.com

Logout

Configuração

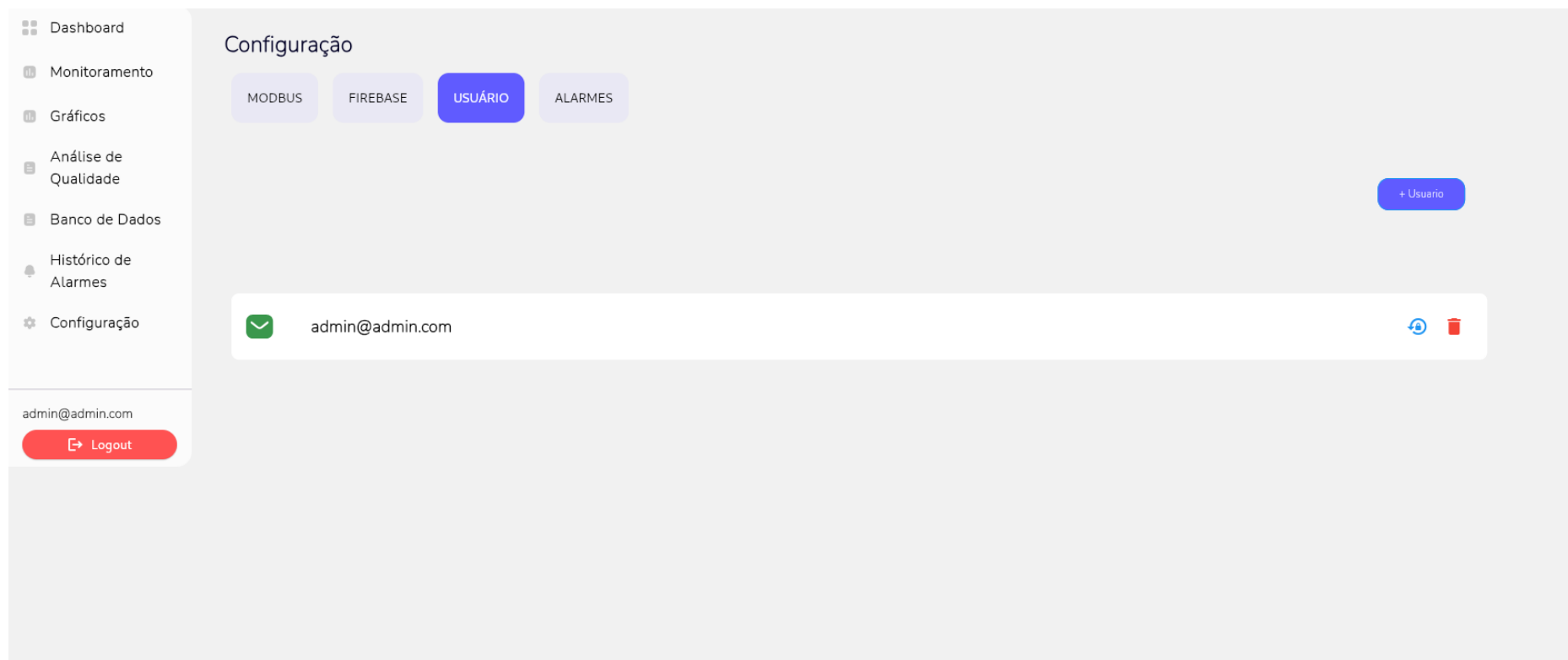
MODBUS FIREBASE USUÁRIO ALARMES

Url Firebase:

Download Testar Conexão

Status Firebase: Conectado

APÊNDICE L – Página de configuração – usuário



APÊNDICE M - Código ESP32 de simulação de Multimedidor ABB M1M Modbus TCP

```
#include <WiFi.h>
#include <ModbusIP_ESP8266.h>

// Modbus Registers Offsets ABB M1M

const int
// Voltage
//RegPhaseL1_0 = 2,RegPhaseL1_1 = 3, RegPhaseL2_0 = 4,RegPhaseL2_1 = 5,RegPhaseL3_0
= 6,RegPhaseL3_1 = 7,
//RegPhaseAVGL1_0 = 220,RegPhaseAVGL1_1 = 221,RegPhaseAVGL2_0 =
222,RegPhaseAVGL2_1 = 223,RegPhaseAVGL3_0 = 224,RegPhaseAVGL3_1 = 225,
//RegLineL1_0 = 8,RegLineL1_1 = 9, RegLineL2_0 = 10,RegLineL2_1 = 11, RegLineL3_0 =
12,RegLineL3_1 = 13,
//RegLineAVGL1_0 = 226,RegLineAVGL1_1 = 227,RegLineAVGL2_0 =
228,RegLineAVGL2_1 = 229,RegLineAVGL3_0 = 230,RegLineAVGL3_1 = 231,

//Voltage
RegPhaseL1_0 = 23298,RegPhaseL1_1 = 23299, RegPhaseL2_0 = 23300,RegPhaseL2_1 =
23301,RegPhaseL3_0 = 23302,RegPhaseL3_1 = 23303,
RegPhaseAVGL1_0 = 23516,RegPhaseAVGL1_1 = 23517,RegPhaseAVGL2_0 =
23518,RegPhaseAVGL2_1 = 23519,RegPhaseAVGL3_0 = 23520,RegPhaseAVGL3_1 =
23521,
RegLineL1_0 = 23304,RegLineL1_1 = 23305, RegLineL2_0 = 23306,RegLineL2_1 = 23307,
RegLineL3_0 = 23308,RegLineL3_1 = 23309,
RegLineAVGL1_0 = 23522,RegLineAVGL1_1 = 23523,RegLineAVGL2_0 =
23524,RegLineAVGL2_1 = 23525,RegLineAVGL3_0 = 23526,RegLineAVGL3_1 = 23527,

// Current
RegCurrentL1_0 = 23312, RegCurrentL1_1 = 23313,
RegCurrentL2_0 = 23314, RegCurrentL2_1 = 23315,
RegCurrentL3_0 = 23316, RegCurrentL3_1 = 23317,
RegCurrentAVGL1_0 = 23508,RegCurrentAVGL1_1 = 23509,RegCurrentAVGL2_0 =
23510,RegCurrentAVGL2_1 = 23511,RegCurrentAVGL3_0 = 23512,RegCurrentAVGL3_1 =
23513,

// Active Power
RegTotkW_0 = 23322, RegTotkW_1 = 23323,
```

RegkW1_0 = 23324, RegkW1_1 = 23325,
 RegkW2_0 = 23326, RegkW2_1 = 23327,
 RegkW3_0 = 23328, RegkW3_1 = 23329,

// Active Power Consumption

RegTotKWh_0 = 20480,
 RegTotKWh_1 = 20481,
 RegTotKWh_2 = 20482,
 RegTotKWh_3 = 20483,

// Reactive Power

RegTotkVAR_0 = 23330, RegTotkVAR_1 = 23331,
 RegkVAR1_0 = 23332, RegkVAR1_1 = 23333,
 RegkVAR2_0 = 23334, RegkVAR2_1 = 23335,
 RegkVAR3_0 = 23336, RegkVAR3_1 = 23337,

// Reactive Power Consumption

RegTotkVARh_0 = 20492,
 RegTotkVARh_1 = 20493,
 RegTotkVARh_2 = 20494,
 RegTotkVARh_3 = 20495,

// Apparent Power

RegTotkVA_0 = 23338, RegTotkVA_1 = 23339,
 RegkVA1_0 = 23340, RegkVA1_1 = 23341,
 RegkVA2_0 = 23342, RegkVA2_1 = 23343,
 RegkVA3_0 = 23344, RegkVA3_1 = 23345,

// Apparent Power Consumption

RegTotkVAh_0 = 20504,
 RegTotkVAh_1 = 20505,
 RegTotkVAh_2 = 20506,
 RegTotkVAh_3 = 20507,

// THD Voltage

RegTHDPhaseL1 = 23808,
 RegTHDPhaseL2 = 23936,
 RegTHDPhaseL3 = 24064,

// THD Current

RegTHDCurrentL1 = 24576,
 RegTHDCurrentL2 = 24704,

```

RegTHDCurrentL3 = 24832,

// Frequency
RegFreq = 23346,

RegPF = 23360;

char* endPtr;

//ModbusIP object
ModbusIP mb;

void converterParaHexadecimal(unsigned long numero, char* buffer, int bufferSize) {
    snprintf(buffer, bufferSize, "%lx", numero);
}

void intToHex32(int number, char* hexString, int bufferSize) {
    // Converte o número inteiro para uma string hexadecimal
    snprintf(hexString, bufferSize, "%08x", number);
}

void setup() {

    // Conexão Wifi e inicio do Servidor Modbus
    Serial.begin(115200);
    WiFi.begin("SSID", "password");
    reconnect();
    mb.setTransactionId(1); // ID do dispositivo
    mb.server();

    //for(int range = 20480; range < 24833; ++range){
    // mb.addHreg(range, 0x0000);
    // printf("register: %i\n", range);
    //}
    //mb.addHreg(30, 0x0000);
    for (int range_voltage = 23298; range_voltage < 23310; range_voltage++){
        mb.addHreg(range_voltage, 0x0000);
    }
    for (int range_current = 23312; range_current < 23318; range_current++){
        mb.addHreg(range_current, 0x0000);
    }
    for (int range_PWR = 23322; range_PWR < 23346; range_PWR++){

```

```

mb.addHreg(range_PWR, 0x0000);
}

mb.addHreg(23346, 0x0000); //frequency
mb.addHreg(23360, 0x0000); // PF
mb.addHreg(23808, 0x0000); // "THDVoltage_L1",
mb.addHreg(23936, 0x0000); // "THDVoltage_L2",
mb.addHreg(24064, 0x0000); // "THDVoltage_L3",
}

void reconnect() { // Nova tentativa de conexão Wifi

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

Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}

void loop() {
int randVph;
mb.task();
delay(2000); // Envia novos valores a cada minuto
char PhaseL1[9],PhaseL1_0[5],PhaseL1_1[5],
PhaseL2[9],PhaseL2_0[5],PhaseL2_1[5],
PhaseL3[9],PhaseL3_0[5],PhaseL3_1[5],
LineL1[9],LineL1_0[5],LineL1_1[5],
LineL2[9],LineL2_0[5],LineL2_1[5],
LineL3[9],LineL3_0[5],LineL3_1[5],
CurrentL1[9],CurrentL1_0[5],CurrentL1_1[5],
CurrentL2[9],CurrentL2_0[5],CurrentL2_1[5],
CurrentL3[9],CurrentL3_0[5],CurrentL3_1[5],
TotkW[9],TotkW_0[5],TotkW_1[5],
kW1[9],kW1_0[5],kW1_1[5],
kW2[9],kW2_0[5],kW2_1[5],
kW3[9],kW3_0[5],kW3_1[5],

TotkVAR[9],TotkVAR_0[5],TotkVAR_1[5],

```

```
kVAR1[9],kVAR1_0[5],kVAR1_1[5],
kVAR2[9],kVAR2_0[5],kVAR2_1[5],
kVAR3[9],kVAR3_0[5],kVAR3_1[5],
```

```
TotkVA[9],TotkVA_0[5],TotkVA_1[5],
kVA1[9],kVA1_0[5],kVA1_1[5],
kVA2[9],kVA2_0[5],kVA2_1[5],
kVA3[9],kVA3_0[5],kVA3_1[5],
Freq[9],Freq_0[5],Freq_1[5],
PF[9],PF_1[5],
```

```
THDPhaseL1[9],THDPhaseL1_0[5],
THDPhaseL2[9],THDPhaseL2_0[5],
THDPhaseL3[9],THDPhaseL3_0[5],
```

```
THDCurrentL1[9],THDCurrentL1_0[5],
THDCurrentL2[9],THDCurrentL2_0[5],
THDCurrentL3[9],THDCurrentL3_0[5]
```

```
;
```

```
if (WiFi.status() != WL_CONNECTED) {
  reconnect();
}
```

```
// Inicializa o gerador de números aleatórios
srand(millis());
randVph = random(1900,2250); //gera tensão de fase 2200 2250
```

```
// Grava os valores gerados no registrador de tensão de fase L1
intToHex32(randVph-910,PhaseL1,sizeof(PhaseL1)); //converte para hexadecimal
strncpy(PhaseL1_0, &PhaseL1[0], 4);
PhaseL1_0[4] = '\0'; // completa com \0 ao final do vetor
strncpy(PhaseL1_1, &PhaseL1[4], 4);
PhaseL1_1[4] = '\0';
mb.Hreg(RegPhaseL1_0,(uint16_t)strtoul(PhaseL1_0, &endPtr, 16));
mb.Hreg(RegPhaseL1_1,(uint16_t)strtoul(PhaseL1_1, &endPtr, 16));
// Grava os valores gerados no registrador de tensão de fase L2
intToHex32(randVph-928,PhaseL2,sizeof(PhaseL2)); //converte para hexadecimal
strncpy(PhaseL2_0, &PhaseL2[0], 4);
PhaseL2_0[4] = '\0'; // completa com \0 ao final do vetor
```

```

strncpy(PhaseL2_1, &PhaseL2[4], 4);
PhaseL2_1[4] = '\0';
mb.Hreg(RegPhaseL2_0,(uint16_t)strtoul(PhaseL2_0, &endPtr, 16));
mb.Hreg(RegPhaseL2_1,(uint16_t)strtoul(PhaseL2_1, &endPtr, 16));

// Grava os valores gerados no registrador de tensão de fase L3
intToHex32(randVph-929,PhaseL3,sizeof(PhaseL3)); //converte para hexadecimal
strncpy(PhaseL3_0, &PhaseL3[0], 4);
PhaseL3_0[4] = '\0'; // completa com \0 ao final do vetor
strncpy(PhaseL3_1, &PhaseL3[4], 4);
PhaseL3_1[4] = '\0';
mb.Hreg(RegPhaseL3_0,(uint16_t)strtoul(PhaseL3_0, &endPtr, 16));
mb.Hreg(RegPhaseL3_1,(uint16_t)strtoul(PhaseL3_1, &endPtr, 16));

printf("valor PhaseL1_0: %s\n", PhaseL1_0);
printf("valor PhaseL1_1: %s\n", PhaseL1_1);
printf("valor PhaseL1: %s\n", PhaseL1);

////////////////////////////////////

// Grava os valores gerados no registrador de tensão de linha L1
intToHex32(randVph,LineL1,sizeof(LineL1)); //converte para hexadecimal
strncpy(LineL1_0, &LineL1[0], 4);
LineL1_0[4] = '\0'; // completa com \0 ao final do vetor
strncpy(LineL1_1, &LineL1[4], 4);
LineL1_1[4] = '\0';
mb.Hreg(RegLineL1_0,(uint16_t)strtoul(LineL1_0, &endPtr, 16));
mb.Hreg(RegLineL1_1,(uint16_t)strtoul(LineL1_1, &endPtr, 16));

// Grava os valores gerados no registrador de tensão de linha L2
intToHex32(randVph+200,LineL2,sizeof(LineL2)); //converte para hexadecimal
strncpy(LineL2_0, &LineL2[0], 4);
LineL2_0[4] = '\0'; // completa com \0 ao final do vetor
strncpy(LineL2_1, &LineL2[4], 4);
LineL2_1[4] = '\0';
mb.Hreg(RegLineL2_0,(uint16_t)strtoul(LineL2_0, &endPtr, 16));
mb.Hreg(RegLineL2_1,(uint16_t)strtoul(LineL2_1, &endPtr, 16));
//

// Grava os valores gerados no registrador de tensão de linha L3
intToHex32(randVph-160,LineL3,sizeof(LineL3)); //converte para hexadecimal
strncpy(LineL3_0, &LineL3[0], 4);
LineL3_0[4] = '\0'; // completa com \0 ao final do vetor

```

```

strncpy(LineL3_1, &LineL3[4], 4);
LineL3_1[4] = '\0';
mb.Hreg(RegLineL3_0,(uint16_t)strtoul(LineL3_0, &endPtr, 16));
mb.Hreg(RegLineL3_1,(uint16_t)strtoul(LineL3_1, &endPtr, 16));

// atalho para terminal
//printf("valor V1 dec: %i\n", randVph-930);
//printf("valor LineL1_0: %s\n", LineL1_0);
//printf("valor LineL1_1: %s\n", LineL1_1);
/////////////////////////////////////////////////////////////////
// Grava os valores gerados no registrador de corrente L1 OK
intToHex32(randVph-1124,CurrentL1,sizeof(CurrentL1)); //converte para hexadecimal
strncpy(CurrentL1_0, &CurrentL1[0], 4);
CurrentL1_0[4] = '\0'; // completa com \0 ao final do vetor
strncpy(CurrentL1_1, &CurrentL1[4], 4);
CurrentL1_1[4] = '\0';
mb.Hreg(RegCurrentL1_0,(uint16_t)strtoul(CurrentL1_0, &endPtr, 16));
mb.Hreg(RegCurrentL1_1,(uint16_t)strtoul(CurrentL1_1, &endPtr, 16));
//
intToHex32(randVph-1091,CurrentL2,sizeof(CurrentL2)); //converte para hexadecimal
strncpy(CurrentL2_0, &CurrentL2[0], 4);
CurrentL2_0[4] = '\0'; // completa com \0 ao final do vetor
strncpy(CurrentL2_1, &CurrentL2[4], 4);
CurrentL2_1[4] = '\0';
mb.Hreg(RegCurrentL2_0,(uint16_t)strtoul(CurrentL2_0, &endPtr, 16));
mb.Hreg(RegCurrentL2_1,(uint16_t)strtoul(CurrentL2_1, &endPtr, 16));
//
intToHex32(randVph-1173,CurrentL3,sizeof(CurrentL3)); //converte para hexadecimal
strncpy(CurrentL3_0, &CurrentL3[0], 4);
CurrentL3_0[4] = '\0'; // completa com \0 ao final do vetor
strncpy(CurrentL3_1, &CurrentL3[4], 4);
CurrentL3_1[4] = '\0';
mb.Hreg(RegCurrentL3_0,(uint16_t)strtoul(CurrentL3_0, &endPtr, 16));
mb.Hreg(RegCurrentL3_1,(uint16_t)strtoul(CurrentL3_1, &endPtr, 16));

/////////////////////////////////////////////////////////////////
intToHex32(24215,TotkW,sizeof(TotkW)); //converte para hexadecimal
strncpy(TotkW_0, &TotkW[0], 4);
TotkW_0[4] = '\0'; // completa com \0 ao final do vetor
strncpy(TotkW_1, &TotkW[4], 4);
TotkW_1[4] = '\0';
mb.Hreg(RegTotkW_0,(uint16_t)strtoul(TotkW_0, &endPtr, 16)); //23322 RegTotkW_0

```

```
mb.Hreg(RegTotkW_1,(uint16_t)strtoul(TotkW_1, &endPtr, 16)); //23323 RegTotkW_1
```

```
intToHex32(8071,kW1,sizeof(kW1)); //converte para hexadecimal
```

```
strncpy(kW1_0, &kW1[0], 4);
```

```
kW1_0[4] = '\0'; // completa com \0 ao final do vetor
```

```
strncpy(kW1_1, &kW1[4], 4);
```

```
kW1_1[4] = '\0';
```

```
mb.Hreg(RegkW1_0,(uint16_t)strtoul(kW1_0, &endPtr, 16));
```

```
mb.Hreg(RegkW1_1,(uint16_t)strtoul(kW1_1, &endPtr, 16));
```

```
intToHex32(8071,kW2,sizeof(kW2)); //converte para hexadecimal
```

```
strncpy(kW2_0, &kW2[0], 4);
```

```
kW2_0[4] = '\0'; // completa com \0 ao final do vetor
```

```
strncpy(kW2_1, &kW2[4], 4);
```

```
kW2_1[4] = '\0';
```

```
mb.Hreg(RegkW2_0,(uint16_t)strtoul(kW2_0, &endPtr, 16));
```

```
mb.Hreg(RegkW2_1,(uint16_t)strtoul(kW2_1, &endPtr, 16));
```

```
//
```

```
intToHex32(8071,kW3,sizeof(kW3)); //converte para hexadecimal
```

```
strncpy(kW3_0, &kW3[0], 4);
```

```
kW3_0[4] = '\0'; // completa com \0 ao final do vetor
```

```
strncpy(kW3_1, &kW3[4], 4);
```

```
kW3_1[4] = '\0';
```

```
mb.Hreg(RegkW3_0,(uint16_t)strtoul(kW3_0, &endPtr, 16));
```

```
mb.Hreg(RegkW3_1,(uint16_t)strtoul(kW3_1, &endPtr, 16));
```

```
////////////////////////////////////
```

```
intToHex32(24218,TotkVAR,sizeof(TotkVAR)); //converte para hexadecimal
```

```
strncpy(TotkVAR_0, &TotkVAR[0], 4);
```

```
TotkVAR_0[4] = '\0'; // completa com \0 ao final do vetor
```

```
strncpy(TotkVAR_1, &TotkVAR[4], 4);
```

```
TotkVAR_1[4] = '\0';
```

```
mb.Hreg(RegTotkVAR_0,(uint16_t)strtoul(TotkVAR_0, &endPtr, 16));
```

```
mb.Hreg(RegTotkVAR_1,(uint16_t)strtoul(TotkVAR_1, &endPtr, 16));
```

```
intToHex32(8072,kVAR1,sizeof(kVAR1)); //converte para hexadecimal
```

```
strncpy(kVAR1_0, &kVAR1[0], 4);
```

```
kVAR1_0[4] = '\0'; // completa com \0 ao final do vetor
```

```
strncpy(kVAR1_1, &kVAR1[4], 4);
```

```
kVAR1_1[4] = '\0';
```

```
mb.Hreg(RegkVAR1_0,(uint16_t)strtoul(kVAR1_0, &endPtr, 16));
```

```

mb.Hreg(RegkVAR1_1,(uint16_t)strtoul(kVAR1_1, &endPtr, 16));
//
intToHex32(8072,kVAR2,sizeof(kVAR2)); //converte para hexadecimal
strncpy(kVAR2_0, &kVAR2[0], 4);
kVAR2_0[4] = '\0'; // completa com \0 ao final do vetor
strncpy(kVAR2_1, &kVAR2[4], 4);
kVAR2_1[4] = '\0';
mb.Hreg(RegkVAR2_0,(uint16_t)strtoul(kVAR2_0, &endPtr, 16));
mb.Hreg(RegkVAR2_1,(uint16_t)strtoul(kVAR2_1, &endPtr, 16));
//
intToHex32(8072,kVAR3,sizeof(kVAR3)); //converte para hexadecimal
strncpy(kVAR3_0, &kVAR3[0], 4);
kVAR3_0[4] = '\0'; // completa com \0 ao final do vetor
strncpy(kVAR3_1, &kVAR3[4], 4);
kVAR3_1[4] = '\0';
mb.Hreg(RegkVAR3_0,(uint16_t)strtoul(kVAR3_0, &endPtr, 16));
mb.Hreg(RegkVAR3_1,(uint16_t)strtoul(kVAR3_1, &endPtr, 16));

////////////////////////////////////

intToHex32(24215,TotkVA,sizeof(TotkVA)); //converte para hexadecimal
strncpy(TotkVA_0, &TotkVA[0], 4);
TotkVA_0[4] = '\0'; // completa com \0 ao final do vetor
strncpy(TotkVA_1, &TotkVA[4], 4);
TotkVA_1[4] = '\0';
mb.Hreg(RegTotkVA_0,(uint16_t)strtoul(TotkVA_0, &endPtr, 16));
mb.Hreg(RegTotkVA_1,(uint16_t)strtoul(TotkVA_1, &endPtr, 16));

intToHex32(8071,kVA1,sizeof(kVA1)); //converte para hexadecimal
strncpy(kVA1_0, &kVA1[0], 4);
kVA1_0[4] = '\0'; // completa com \0 ao final do vetor
strncpy(kVA1_1, &kVA1[4], 4);
kVA1_1[4] = '\0';
mb.Hreg(RegkVA1_0,(uint16_t)strtoul(kVA1_0, &endPtr, 16));
mb.Hreg(RegkVA1_1,(uint16_t)strtoul(kVA1_1, &endPtr, 16));
//
intToHex32(8071,kVA2,sizeof(kVA2)); //converte para hexadecimal
strncpy(kVA2_0, &kVA2[0], 4);
kVA2_0[4] = '\0'; // completa com \0 ao final do vetor
strncpy(kVA2_1, &kVA2[4], 4);
kVA2_1[4] = '\0';
mb.Hreg(RegkVA2_0,(uint16_t)strtoul(kVA2_0, &endPtr, 16));

```

```

mb.Hreg(RegkVA2_1,(uint16_t)strtoul(kVA2_1, &endPtr, 16));
//
intToHex32(8071,kVA3,sizeof(kVA3)); //converte para hexadecimal
strncpy(kVA3_0, &kVA3[0], 4);
kVA3_0[4] = '\0'; // completa com \0 ao final do vetor
strncpy(kVA3_1, &kVA3[4], 4);
kVA3_1[4] = '\0';
mb.Hreg(RegkVA3_0,(uint16_t)strtoul(kVA3_0, &endPtr, 16));
mb.Hreg(RegkVA3_1,(uint16_t)strtoul(kVA3_1, &endPtr, 16));

/////////////////////////////////////////////////////////////////

intToHex32(6000,Freq,sizeof(Freq)); //converte para hexadecimal
strncpy(Freq_1, &Freq[4], 4);
Freq_1[4] = '\0';
mb.Hreg(RegFreq,(uint16_t)strtoul(Freq_1, &endPtr, 16));

/////////////////////////////////////////////////////////////////

//RegPF = 23360;
int randomFP = random(800, 1000);

intToHex32(randomFP,PF,sizeof(PF)); //converte para hexadecimal
strncpy(PF_1, &PF[4], 4);
PF_1[4] = '\0'; // completa com \0 ao final do vetor
mb.Hreg(RegPF,(uint16_t)strtoul(PF_1, &endPtr, 16)); //

/////////////////////////////////////////////////////////////////

// THD Voltage
int randomTHD = random(70, 120);

// Grava os valores gerados no registrador de thd tensão de fase L1
intToHex32(randomTHD,THDPhaseL1,sizeof(THDPhaseL1)); //converte para hexadecimal
strncpy(THDPhaseL1_0, &THDPhaseL1[4], 4);
THDPhaseL1_0[4] = '\0'; // completa com \0 ao final do vetor
mb.Hreg(RegTHDPhaseL1,(uint16_t)strtoul(THDPhaseL1_0, &endPtr, 16));

intToHex32(randomTHD-5,THDPhaseL2,sizeof(THDPhaseL2)); //converte para hexadecimal
strncpy(THDPhaseL2_0, &THDPhaseL2[4], 4);
THDPhaseL2_0[4] = '\0'; // completa com \0 ao final do vetor

```

```

mb.Hreg(RegTHDPhaseL2,(uint16_t)strtoul(THDPhaseL2_0, &endPtr, 16));

intToHex32(randomTHD+3,THDPhaseL3,sizeof(THDPhaseL3));      //converte      para
hexadecimal
strncpy(THDPhaseL3_0, &THDPhaseL3[4], 4);
THDPhaseL3_0[4] = '\0'; // completa com \0 ao final do vetor
mb.Hreg(RegTHDPhaseL3,(uint16_t)strtoul(THDPhaseL3_0, &endPtr, 16));

////////////////////////////////////
// THD Current
//RegTHDCurrentL1 = 24576, THDCurrentL1[9],THDCurrentL1_0[5],
//RegTHDCurrentL2 = 24704, THDCurrentL2[9],THDCurrentL2_0[5],
//RegTHDCurrentL3 = 24832, THDCurrentL3[9],THDCurrentL3_0[5]

intToHex32(randomTHD-3,THDCurrentL1,sizeof(THDCurrentL1));  //converte      para
hexadecimal
strncpy(THDCurrentL1_0, &THDCurrentL1[4], 4);
THDPhaseL1_0[4] = '\0'; // completa com \0 ao final do vetor
mb.Hreg(RegTHDCurrentL1,(uint16_t)strtoul(THDCurrentL1_0, &endPtr, 16));

intToHex32(randomTHD+2,THDCurrentL2,sizeof(THDCurrentL2));  //converte      para
hexadecimal
strncpy(THDCurrentL2_0, &THDCurrentL2[4], 4);
THDCurrentL2_0[4] = '\0'; // completa com \0 ao final do vetor
mb.Hreg(RegTHDCurrentL2,(uint16_t)strtoul(THDCurrentL2_0, &endPtr, 16));
//
intToHex32(randomTHD-4,THDCurrentL3,sizeof(THDCurrentL3));  //converte      para
hexadecimal
strncpy(THDCurrentL3_0, &THDCurrentL3[4], 4);
THDPhaseL3_0[4] = '\0'; // completa com \0 ao final do vetor
mb.Hreg(RegTHDCurrentL3,(uint16_t)strtoul(THDCurrentL3_0, &endPtr, 16));

// atalho para terminal
//printf("valor V1 dec: %i\n", randVph-930);
//printf("valor THDCurrentL1_0: %s\n", THDCurrentL1_0);
//printf("valor THDCurrentL1: %s\n", THDCurrentL1);

}

```

APÊNDICE N – Grupo dos registradores do aplicativo

```

var m1mModbusGroup_1 = ModbusElementsGroup(
[
ModbusUint32Register(
name: "V1",
address: 23298,
type: ModbusElementType.holdingRegister,
multiplier: 0.1,
// onUpdate: (self) => print(self)
),
ModbusUint32Register(
name: "V2",
address: 23300,
type: ModbusElementType.holdingRegister,
multiplier: 0.1,
// onUpdate: (self) => print(self)
),
ModbusUint32Register(
name: "V3",
address: 23302,
type: ModbusElementType.holdingRegister,
multiplier: 0.1,
// onUpdate: (self) => print(self)
),
ModbusUint32Register(
name: "V12",
address: 23304,
type: ModbusElementType.holdingRegister,
multiplier: 0.1,
// onUpdate: (self) => print(self)
),
ModbusUint32Register(
name: "V23",
address: 23306,
type: ModbusElementType.holdingRegister,
multiplier: 0.1,
// onUpdate: (self) => print(self)
),
ModbusUint32Register(
name: "V31",
address: 23308,

```

```

type: ModbusElementType.holdingRegister,
multiplier: 0.1,
// onUpdate: (self) => print(self)
),

```

```

ModbusUint32Register(
name: "L1",
address: 23312,
type: ModbusElementType.holdingRegister,
multiplier: 0.01,
// onUpdate: (self) => print(self)
),

```

```

ModbusUint32Register(
name: "L2",
address: 23314,
type: ModbusElementType.holdingRegister,
multiplier: 0.01,
// onUpdate: (self) => print(self)
),

```

```

ModbusUint32Register(
name: "L3",
address: 23316,
type: ModbusElementType.holdingRegister,
multiplier: 0.01,
// onUpdate: (self) => print(self)
),

```

```

ModbusInt32Register(
name: "Active_Power_total",
address: 23322,
type: ModbusElementType.holdingRegister,
multiplier: 0.01,
// onUpdate: (self) => print(self)
),

```

```

ModbusInt32Register(
name: "kW1",
address: 23324,
type: ModbusElementType.holdingRegister,
multiplier: 0.01,
// onUpdate: (self) => print(self)
),

```

```

ModbusInt32Register(

```

```

name: "kW2",
address: 23326,
type: ModbusElementType.holdingRegister,
multiplier: 0.01,
// onUpdate: (self) => print(self)
),

```

```

ModbusUint32Register(
name: "kW3",
address: 23328,
type: ModbusElementType.holdingRegister,
multiplier: 0.01,
// onUpdate: (self) => print(self)
),

```

```

ModbusInt32Register(
name: "Reactive_Power_total",
address: 23330,
type: ModbusElementType.holdingRegister,
multiplier: 0.01,
// onUpdate: (self) => print(self)
),

```

```

ModbusInt32Register(
name: "kVAR1",
address: 23332,
type: ModbusElementType.holdingRegister,
multiplier: 0.01,
// onUpdate: (self) => print(self)
),
//

```

```

ModbusInt32Register(
name: "kVAR2",
address: 23334,
type: ModbusElementType.holdingRegister,
multiplier: 0.01,
// onUpdate: (self) => print(self)
),

```

```

ModbusInt32Register(
name: "kVAR3",
address: 23336,
type: ModbusElementType.holdingRegister,

```

```

multiplier: 0.01,
// onUpdate: (self) => print(self)
),
ModbusInt32Register(
name: "Apparent_Power_total",
address: 23338,
type: ModbusElementType.holdingRegister,
multiplier: 0.01,
// onUpdate: (self) => print(self)
),
ModbusInt32Register(
name: "kVA1",
address: 23340,
type: ModbusElementType.holdingRegister,
multiplier: 0.01,
// onUpdate: (self) => print(self)
),
ModbusInt32Register(
name: "kVA2",
address: 23342,
type: ModbusElementType.holdingRegister,
multiplier: 0.01,
// onUpdate: (self) => print(self)
),
ModbusInt32Register(
name: "kVA3",
address: 23344,
type: ModbusElementType.holdingRegister,
multiplier: 0.01,
// onUpdate: (self) => print(self)
),
ModbusUint16Register(
name: "Freq",
address: 23346,
type: ModbusElementType.holdingRegister,
multiplier: 0.01,
// onUpdate: (self) => print(self)
),
ModbusUint16Register(
name: "Power_Factor_total",
address: 23360,
type: ModbusElementType.holdingRegister,

```

```

multiplier: 0.001,
// onUpdate: (self) => print(self)
),

]
);

var m1mTHDVoltage_L1 = ModbusUint16Register(
name: "THDVoltage_L1",
address: 23808,
type: ModbusElementType.holdingRegister,
multiplier: 0.1,
onUpdate: (self) => print(self)
);

var m1mTHDVoltage_L2 = ModbusUint16Register(
name: "THDVoltage_L2",
address: 23936,
type: ModbusElementType.holdingRegister,
multiplier: 0.1,
// onUpdate: (self) => print(self)
);

var m1mTHDVoltage_L3 = ModbusUint16Register(
name: "THDVoltage_L3",
address: 24064,
type: ModbusElementType.holdingRegister,
multiplier: 0.1,
// onUpdate: (self) => print(self)
);

var m1mTHDCurrent_L1 = ModbusUint16Register(
name: "THDCurrent_L1",
address: 24576, // 24576
type: ModbusElementType.holdingRegister,
multiplier: 0.1,
// onUpdate: (self) => print(self)
);

var m1mTHDCurrent_L2 = ModbusUint16Register(
name: "THDCurrent_L2",
address: 24704,

```

```
type: ModbusElementType.holdingRegister,  
multiplier: 0.1,  
// onUpdate: (self) => print(self)  
);
```

```
var m1mTHDCurrent_L3 = ModbusUint16Register(  
name: "THDCurrent_L3",  
address: 24832,  
type: ModbusElementType.holdingRegister,  
multiplier: 0.1,  
//onUpdate: (self) => print(self)  
);
```

APÊNDICE O – Faixas de classificação de qualidade da tensão de fase

Tensão Nominal	Faixa Adequada	Faixa Precária	Faixa Crítica
110V	$101V \leq \text{Tensão} \leq 116V$	$(96V \leq \text{Tensão} < 101V)$ ou $(116V < \text{Tensão} \leq 117V)$	Tensão < 96V ou Tensão > 117V
115V	$106V \leq \text{Tensão} \leq 121V$	$(100V \leq \text{Tensão} < 106V)$ ou $(121V < \text{Tensão} \leq 122V)$	Tensão < 100V ou Tensão > 122V
120V	$110V \leq \text{Tensão} \leq 126V$	$(104V \leq \text{Tensão} < 110V)$ ou $(126V < \text{Tensão} \leq 127V)$	Tensão < 104V ou Tensão > 127V
127V	$117V \leq \text{Tensão} \leq 133V$	$(110V \leq \text{Tensão} < 117V)$ ou $(133V < \text{Tensão} \leq 135V)$	Tensão < 110V ou Tensão > 135V
220V	$202V \leq \text{Tensão} \leq 231V$	$(191V \leq \text{Tensão} < 202V)$ ou $(231V < \text{Tensão} \leq 233V)$	Tensão < 191V ou Tensão > 233V