



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

JOSAFÁ TAVARES DE MELO JÚNIOR

**MODELO DE BANCO DE DADOS PARA GERENCIAMENTO ENERGÉTICO
REMOTO EM MICRORREDES:
ESTUDO DE CASO DA MICRORREDE DO LABORATÓRIO DE
ARMAZENAMENTO E MOBILIDADE (LAM)**

Recife
2023

JOSAFÁ TAVARES DE MELO JÚNIOR

**MODELO DE BANCO DE DADOS PARA GERENCIAMENTO ENERGÉTICO
REMOTO EM MICRORREDES:
ESTUDO DE CASO DA MICRORREDE DO LABORATÓRIO DE
ARMAZENAMENTO E MOBILIDADE (LAM)**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Controle e Automação.

Orientador(a): Prof. Dr. Douglas Contente Pimentel Barbosa

Recife
2023

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Melo Júnior, Josafá Tavares de.

Modelo de banco de dados para gerenciamento energético remoto em microrredes: estudo de caso da microrrede do Laboratório de Armazenamento e Mobilidade (LAM) / Josafá Tavares de Melo Júnior. - Recife, 2023.

65 P: il., tab.

Orientador(a): Douglas Contente Pimentel Barbosa

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Engenharia de Controle e Automação - Bacharelado, 2023.

1. Microrredes. 2. Sistema de Armazenamento. 3. Banco de dados. 4. Visualização de Dados. 5. Inteligência de Negócios. I. Barbosa, Douglas Contente Pimentel. (Orientação). II. Título.

620 CDD (22.ed.)

JOSAFÁ TAVARES DE MELO JÚNIOR

**MODELO DE BANCO DE DADOS PARA GERENCIAMENTO ENERGÉTICO
REMOTO EM MICRORREDES:
ESTUDO DE CASO DA MICRORREDE DO LABORATÓRIO DE
ARMAZENAMENTO E MOBILIDADE (LAM)**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Controle e Automação.

Aprovado em: 26/09/2023.

BANCA EXAMINADORA

Prof. Dr. Douglas Contente Pimentel Barbosa (Orientador)
Universidade Federal de Pernambuco

Prof. Dr. José Filho da Costa Castro (Examinador interno)
Universidade Federal de Pernambuco

Prof. Dr. Pedro André Carvalho Rosas (Examinador interno)
Universidade Federal de Pernambuco

AGRADECIMENTOS

Agradeço a Deus por ter me concedido saúde, resiliência, sabedoria, fé e capacidade de superar todas as dificuldades que surgiram em meu caminho até chegar até aqui. Por meio da aplicação dos princípios da engenharia, que é fruto da sua obra, eu posso expressar minha gratidão a Deus de acordo com sua vontade, ao trabalhar em prol do bem da humanidade. Este trabalho que realizo é uma demonstração de minha gratidão por tudo que Ele tem feito em minha vida.

Agradeço também, em especial, a meus pais, Josafá e Lucimar, que me ajudaram e me deram suporte para conclusão do curso de graduação em todo o momento e sempre buscaram o meu melhor, e a minha esposa e filho, Mayse e Willian pelo apoio incondicional e companheirismo.

Agradeço também, minha irmã Juliana, minha tia Joseane e minhas avós, Cícera e Nice, e aos meus amigos Cauê, Coloia, Emerson e Giovani, por terem sido um grande suporte em todas as fases do meu curso de graduação, sempre me incentivando e me orientando em direção ao meu melhor desempenho.

Agradeço ao meu orientador professor Dr. Douglas, que sempre esteve solícito em me ajudar e se dispôs em me orientar no desenvolvimento deste trabalho. Agradeço a Deus pelos seus ensinamentos.

Agradeço também a meu coorientador e amigo Dr. Davidson, o qual esteve comigo nos anos finais da graduação e durante o desenvolvimento deste trabalho, me proporcionando oportunidades, momentos alegres e ensinamentos que levarei comigo para a vida. Obrigado por todo o tempo e esforço investidos em minha formação.

Agradeço a meus amigos que fiz durante este período de graduação em especial: Alliny, André, Gilmar, Paulo, Rodrigo companheiros de disciplinas; Amauri, Caio, Gabriel, Gustavo, Mateus, Ruan companheiros de laboratório.

Agradeço também ao professor Dr. Pedro Rosas pela estruturação do Laboratório de Armazenamento e Mobilidade, o qual me proporcionou a realização deste trabalho de conclusão de curso.

Agradeço a oportunidade de ter feito meu trabalho no Laboratório de Armazenamento e Mobilidade, financiado pela CPFL Energia via Programa de P&D da Agência Nacional de Energia Elétrica (ANEEL), com recursos obtidos no âmbito do projeto Chamada Estratégica da

ANEEL: PD-0397-3026/2017 - “Inserção de Sistema de Armazenamento em Configurações Múltiplas Para Suporte à Geração Eólica” e do projeto “PA3059 - Soluções de Apoio à Expansão da Infraestrutura de Recarga de Veículos Elétricos: Estações de Carregamento Integradas à Tecnologia de Baterias e Sistemas Fotovoltaicos (PV)”.

Agradeço ao Programa de P&D da ANEEL e CPFL Energia pelo apoio financeiro para aquisição dos equipamentos que permitiram a elaboração deste trabalho, em particular aos engenheiros Guilherme Rissi e Amanda Lopes Fernandes da CPFL Energia pelo apoio para concretização das aquisições dos equipamentos do laboratório.

Bem sei eu que tudo podes e nenhum dos teus
pensamentos serão impedidos. Jó 42: 2.

RESUMO

A crescente demanda por eficiência energética e a busca por fontes renováveis têm impulsionado o desenvolvimento de microrredes como soluções inteligentes para a geração e gerenciamento de energia. Nesse cenário, as microrredes se destacam ao integrar fontes renováveis e armazenamento energético, promovendo a produção sustentável e impulsionando a geração de energia limpa por meio da combinação de diversas fontes, como painéis solares, baterias e geradores, visando maximizar a eficiência e confiabilidade do fornecimento energético. No entanto, o gerenciamento eficaz dessas microrredes exige o tratamento de grandes volumes de dados provenientes de múltiplas fontes acarretando na problemática da aquisição, armazenamento e análise dessas informações de forma estruturada e facilmente acessível. Essa questão centraliza-se na ausência de um sistema integrado e eficaz para administrar esses dados, dificultando a tomada de decisões informadas e a identificação de oportunidades de melhoria. Essa lacuna ressalta a necessidade de um banco de dados estruturado e um sistema de visualização que possibilite o acesso fácil e a interpretação desses dados complexos. Para abordar esses desafios, este trabalho propõe o desenvolvimento de um sistema de gerenciamento de dados para microrredes, que se fundamenta em um banco de dados relacional de código aberto. O processo de definição dos objetivos do banco de dados e das informações a serem armazenadas é cuidadosamente discutido, levando em consideração a identificação dos componentes do laboratório responsáveis por fornecer os dados. Esse processo orienta a abordagem de modelagem dos dados e o desenvolvimento da estrutura necessária para o armazenamento. Além disso, o trabalho explora o processo de aquisição dos dados por meio da comunicação MQTT, apresentando exemplos de dados coletados e como eles foram estruturados para posterior armazenamento no banco de dados. O resultado final desse esforço é evidenciado na implementação de uma interface de monitoramento, realizada com a ferramenta Microsoft Power BI, que demonstra a capacidade de comunicação da interface tanto em ambiente local quanto na nuvem. Também são oferecidas diretrizes para pesquisas futuras na área de gerenciamento de dados em microrredes, consolidando assim um avanço significativo neste campo de estudo. Através desta abordagem, o trabalho propõe uma solução robusta para lidar com a complexidade do gerenciamento de dados em microrredes, contribuindo para a eficiência e sustentabilidade desse cenário energético em constante evolução.

Palavras-chave: Microrredes; Sistema de Armazenamento; Banco de dados; Visualização de Dados; Inteligência de Negócios.

ABSTRACT

The growing demand for energy efficiency and the search for renewable sources have driven the development of microgrids as smart solutions for energy generation and management. In this scenario, microgrids stand out by integrating renewable sources and energy storage, promoting sustainable production and boosting the generation of clean energy through the combination of different sources, such as solar panels, batteries and generators, aiming to maximize efficiency and reliability of supply. energetic. However, the effective management of these microgrids requires the treatment of large volumes of data from multiple sources, resulting in the problem of acquiring, storing and analyzing this information in a structured and easily accessible way. This issue centers on the absence of an integrated and effective system to manage this data, making it difficult to make informed decisions and identify opportunities for improvement. This gap underscores the need for a structured database and visualization system that enables easy access and interpretation of this complex data. To address these challenges, this work proposes the development of a data management system for microgrids, which is based on an open-source relational database. The process of defining the objectives of the database and the information to be stored is carefully discussed, taking into account the identification of the laboratory components responsible for providing the data. This process guides the approach to modeling the data and developing the necessary structure for storage. In addition, the work explores the data acquisition process through MQTT communication, presenting examples of collected data and how they were structured for later storage in the database. The end result of this effort is evidenced in the implementation of a monitoring interface, carried out with Microsoft Power BI, which demonstrates the interface's ability to communicate both locally and in the cloud. Guidelines are also offered for future research in the area of data management in microgrids, thus consolidating a significant advance in this field of study. Through this approach, the work proposes a robust solution to deal with the complexity of data management in microgrids, contributing to the efficiency and sustainability of this energy scenario in constant evolution.

Keywords: Microgrid; Management System; Database; Data Visualization; Business Intelligence.

LISTA DE ILUSTRAÇÕES

Figura 1 - Layout da microrrede do LAM.....	26
Figura 2 - Microrrede do LAM.....	27
Figura 3 - Quadro elétrico da Microrrede.....	27
Figura 4 - Quadro de Automação.....	28
Figura 5 - Cerbo GX.....	29
Figura 6 - Visão geral do protocolo MQTT.....	31
Figura 7 - Processo de publicação de mensagem MQTT.....	32
Figura 8 - Sequência de comunicação MQTT.....	33
Figura 9 - Exemplo modelo conceitual textual.....	35
Figura 10 - Exemplo modelo conceitual diagrama entidade-relacionamento (ER).....	36
Figura 11 - Exemplo de modelo lógico.....	37
Figura 12 - Exemplo modelo lógico preenchido da tabela de pedido.....	38
Figura 13 - Exemplo modelo físico na linguagem SQL para o PostgreSQL.....	39
Figura 14 - Modelo conceitual textual do banco de dados do LAM.....	40
Figura 15 - Modelo lógico do banco de dados do LAM.....	42
Figura 16 - Modelo físico na linguagem SQL para o PostgreSQL.....	43
Figura 17 - Ativação do protocolo MQTT no Cerbo-GX.....	47
Figura 18 - Automação para mensagem keepalive.....	48
Figura 19 - Conexão com o <i>Broker GX</i>	49
Figura 20 - Tópicos disponíveis.....	50
Figura 21 - Arquivo Json com os tópicos.....	50
Figura 22 - Comandos de conexão com o banco de dados na nuvem.....	52
Figura 23 - Conexão de fonte de dados ODBC no Power BI Desktop.....	54
Figura 24 - Cadeia de conexão para o banco de dados.....	54
Figura 25 - Seleção de tabelas para importação no Power BI.....	55
Figura 26 - <i>Dashboard</i> dos parâmetros elétricos do inversor <i>on-grid</i>	57
Figura 27 - <i>Dashboard</i> parâmetros elétricos em tempo real do equipamento Fronius.....	58
Figura 28 - <i>Dashboard</i> da bateria.....	59
Figura 29 - <i>Dashboard</i> de consumo das cargas.....	60
Figura 30 - Painel de controle de produção fotovoltaica dos controladores de carga.....	61
Figura 31 - Página de publicação online da <i>Dashboard</i> no Power BI.....	62

LISTA DE TABELAS

Tabela 1 - Códigos e respostas de retorno.....	31
Tabela 2 - Exemplo modelo físico detalhamento de colunas (campos) da tabela cliente.	38

LISTA DE ABREVIATURAS E SIGLAS

BD	Banco de Dados
API	<i>Application Programming Interface</i>
BESS	<i>Battery Energy Storage System</i>
BI	<i>Business Intelligence</i>
BMS	<i>Battery Management System</i>
CLP	Controlador Lógico Programável
ER	Entidade-Relacionamento
FV	Fotovoltaico
IoT	<i>Internet of Things</i>
LAM	Laboratório de Armazenamento e Mobilidade
MQTT	<i>Message Queuing Telemetry Transport</i>
MR	Microrrede
ODBC	<i>Open Database Connectivity</i>
QoS	<i>Quality of Service</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
SGE	Sistema de Gerenciamento de Energia
SGER- MR	Sistema de Gerenciamento de Energia em Microrredes
TLS	<i>Transport Layer Security</i>
UFPE	Universidade Federal de Pernambuco

SUMÁRIO

1	INTRODUÇÃO	15
1.1	OBJETIVOS.....	16
1.1.1	Geral	16
1.1.2	Específicos.....	16
1.2	MOTIVAÇÃO.....	17
1.3	ORGANIZAÇÃO DO TRABALHO	18
2	REVISÃO DO ESTADO DA ARTE –MODELOS DE BANCO DE DADOS. ..	19
2.1	BANCO DE DADOS	19
2.2	EMBASAMENTO PARA ESCOLHA DO MODELO DE BANCO DE DADOS ADOTADO NO TRABALHO	21
3	METODOLOGIA PARA INTEGRAÇÃO DO BANCO DE DADOS E A MICRORREDE	23
3.1	DEFINIÇÃO DOS OBJETIVOS DO BANCO DE DADOS	23
3.2	COMPONENTES DA MICRORREDE E IDENTIFICAÇÃO DA ORIGEM DOS DADOS 24	
3.2.1	Microrredes	25
3.2.2	Topologia da MR do LAM.....	25
3.2.3	Comunicação com o Cerbo GX.....	28
3.2.4	Protocolo e comunicação MQTT	30
3.2.5	Publicação e inscrição MQTT	32
4	DESENVOLVIMENTO DO BANCO DE DADOS	34
4.1	MODELAGEM DOS DADOS	34
4.1.1	Modelo Conceitual	34
4.1.1.1	<i>Modelo Entidade-Relacionamento</i>	35
4.1.2	Modelo Lógico	36
4.1.3	Modelo Físico.....	38
4.2	APLICAÇÕES DA MODELAGEM DE DADOS AOS ITENS DO LAM	39
4.3	FERRAMENTAS UTILIZADAS PARA DESENVOLVIMENTO DO TRABALHO 44	
4.3.1	Python.....	44
4.3.2	Visual Studio Code.....	44
4.3.3	PostgreSQL	45
4.3.4	Power BI.....	45
4.3.5	Azure	45
4.3.6	BR Modelo Web.....	46
4.3.7	Dbdiagram	46
4.4	IMPLEMENTAÇÃO DA INTEGRAÇÃO DO SGER-MR.....	46
4.4.1	Integração entre Python e MQTT.....	46
4.4.2	Inserção do banco de dados na nuvem	51

4.4.3	Configuração do Power BI.....	53
5	RESULTADOS DA <i>DASHBOARD</i> COM CONJUNTO DE DADOS DA MICRORREDE DO LAM.....	56
6	CONCLUSÕES E PROPOSTAS DE CONTINUIDADE	63

1 INTRODUÇÃO

O Brasil, com sua vasta extensão territorial equivalente a 8,5 milhões km², atrelada a abundância de recursos energéticos predominantemente renováveis, tem o desafio de implementar soluções que acompanhem o crescimento do mercado brasileiro de energia. Consciente de que o sistema elétrico de potência se divide em três partes essenciais (geração, transmissão e distribuição) e que a energia elétrica pode ser produzida de diversas formas, o Brasil busca aproveitar a abundância dos ventos, a alta incidência solar e a disponibilidade de recursos naturais. Esses fatores demandam abordagens inovadoras e sustentáveis, visando garantir o suprimento energético necessário para atender às demandas crescentes do país (INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA, 2021).

Nesse contexto, as microrredes desempenham um papel crucial ao operar na geração e distribuição de energia em níveis locais, oferecendo uma solução promissora para o país. Esses sistemas autônomos de energia podem operar de forma isolada ou em conjunto com a rede elétrica convencional, integrando fontes renováveis como energia eólica, energia solar e biomassa, juntamente com sistemas de armazenamento de energia, como baterias. Dessa forma, as microrredes impulsionam a geração de energia mais limpa e sustentável, reduzindo a dependência de fontes não renováveis.

Uma microrrede é uma rede elétrica de baixa ou média tensão que integra inteligentemente recursos de geração distribuída, armazenamento de energia e cargas elétricas. Ela tem a capacidade de se desconectar do sistema elétrico principal, por meio de um equipamento de manobra, formando uma rede autônoma capaz de fornecer energia sem interrupções para suas cargas. Essas microrredes são compostas por diferentes componentes, como um controlador responsável pelo equilíbrio de energia e potência, sistemas de armazenamento de energia, como baterias, geradores controláveis e cargas que consomem a energia gerada. Essas cargas podem ser priorizadas de acordo com a importância, garantindo um melhor equilíbrio entre oferta e demanda de energia (RIBEIRO, 2023).

O gerenciamento de energia em uma microrrede tem sido objetivo de muitas pesquisas e estudos, e diversas ações de controle têm sido utilizadas para lidar com a imprevisibilidade de algumas fontes de energia, como a carga e descarga de elementos de armazenamento, como baterias. Além disso, o monitoramento das condições climáticas locais é relevante para prever o desempenho das fontes renováveis, pois a partir da série histórica local podem ser obtidos informações meteorológicas (FERREIRA, FARDIN e RUEDA-MEDINA, 2018).

Uma característica relevante das microrredes é a sua capacidade de estabelecer comunicação com diversos equipamentos. Essa rede de comunicação e automação desempenha um papel fundamental no monitoramento e gerenciamento dos componentes e equipamentos da microrrede, assegurando sua proteção e eficiência operacional. Essa interação contínua entre os dispositivos resulta na geração de uma quantidade expressiva de dados. Para que essas informações sejam úteis e orientem as decisões adequadas, é necessário coletar, transformar e organizar esses dados de forma precisa, permitindo uma análise criteriosa e embasada.

A existência de um banco de dados (BD) é de extrema importância para as microrredes. Um BD possibilita a coleta sistemática e estruturada dos dados provenientes das microrredes, permitindo a sua posterior transformação e análise precisa. Ao centralizar e consolidar os dados, torna-se possível extrair *insights* relevantes para o monitoramento, gerenciamento e tomada de decisões eficazes em relação aos componentes e operação da microrrede. Além disso, um banco de dados bem estruturado facilita a identificação de padrões, tendências e possíveis melhorias na eficiência energética, segurança e desempenho da microrrede, contribuindo para sua otimização contínua e aproveitamento máximo dos recursos disponíveis.

1.1 Objetivos

Esta seção apresentará os objetivos geral e específicos deste trabalho.

1.1.1 Geral

O objetivo geral deste trabalho consiste em apresentar o desenvolvimento de um sistema de gerenciamento de dados, que utiliza um banco de dados relacional de código aberto para armazenar e gerenciar grandes quantidades de informações com acessos simultâneos e propostas para aplicações futuras.

1.1.2 Específicos

Este trabalho busca também os seguintes objetivos específicos:

- Realizar uma revisão do Estado da Arte – Modelo de banco de dados;

- Definir o objetivo do BD e o que ele deverá armazenar;
- Modelar os dados que serão armazenados e o desenvolvimento da estrutura do banco de dados;
- Demonstrar a inserção do Sistema de Gerenciamento de Banco de Dados (SGBD) em um serviço de *backup* em nuvem, bem como o acesso remoto ao mesmo;
- Desenvolver uma interface de monitoramento customizável que possa ser integrada ao banco de dados, que permitirá a criação de visualizações e relatórios personalizados para visualização dos dados;
- Validar as informações armazenadas;
- Propor estudos para continuidade deste trabalho.

1.2 Motivação

A microrrede do Laboratório de Armazenamento e Mobilidade (LAM) está em funcionamento, proporcionando um ambiente propício para pesquisa e o desenvolvimento de soluções inovadoras no campo do gerenciamento energético e testes de novas tecnologias aplicadas às microrredes. Nesse contexto, um banco de dados desempenha um papel fundamental na coleta e análise de dados gerados por essa microrrede.

Entre as principais motivações para a implantação de sistemas de gerenciamento de dados em microrredes podem ser destacadas: (1) organização e armazenamento eficiente dos dados; (2) a visualização e tomada de decisões embasadas em dados; (3) monitoramento através do histórico dos dados e detecção de anomalias; (4) integração com outros sistemas e dispositivos. A utilização de fontes renováveis de energia, como os painéis solares, implica necessidade de acompanhamentos constantes dessas fontes.

A análise dos dados armazenados no banco de dados também contribui para a identificação de tendências e a previsão de demanda futura. Com base nesses conceitos, a gestão da microrrede pode implementar estratégias para otimizar o uso da energia, estabelecer políticas de consumo de energia e planejar investimentos em expansão ou atualização da infraestrutura.

Além disso, o histórico de dados permite uma análise retrospectiva do desempenho da microrrede ao longo do tempo. Essas informações permitem otimizar a eficiência, garantir a confiabilidade, realizar manutenção preventiva, planejar estrategicamente e tomar decisões informadas para promover o uso sustentável e eficiente das energias nas microrredes.

1.3 Organização do Trabalho

O presente trabalho foi dividido em capítulos e sua organização segue uma sequência descrita posteriormente após a introdução:

- Capítulo 1: Foi apresentado o cenário atual da geração elétrica no Brasil e das MR (microrredes), bem como os objetivos pretendidos e a motivação deste trabalho.
- Capítulo 2: Será realizada uma revisão do estado da arte no que diz respeito aos modelos de banco de dados utilizados atualmente, explorando as características, vantagens e desvantagens de cada um, de modo a embasar a escolha do modelo adotado no projeto;
- Capítulo 3: Será discutido o processo de definição dos objetivos do BD e das informações a serem armazenadas, bem como a metodologia do estudo, descrevendo através da identificação dos respectivos componentes do laboratório responsáveis por fornecer os dados;
- Capítulo 4: Serão abordados a modelagem dos dados e o desenvolvimento da estrutura necessária para o armazenamento. Além disso, será descrito o processo de aquisição dos dados por meio da comunicação MQTT, apresentando exemplos de dados coletados e como eles foram estruturados para serem armazenados no banco de dados;
- Capítulo 5: Será apresentado o resultado da interface de monitoramento, utilizando a ferramenta da *Microsoft Power BI* (MICROSOFT, 2023), demonstrando assim a capacidade de comunicação da interface tanto em ambiente local quanto na nuvem. O mesmo capítulo irá expor os dados obtidos através da metodologia apresentada no capítulo 3, bem como a visualização dos resultados para validação;
- Capítulo 6: Apresenta as conclusões e possíveis diretrizes para realização de trabalhos futuros.

2 REVISÃO DO ESTADO DA ARTE – MODELOS DE BANCO DE DADOS.

Este capítulo fornece os fundamentos teóricos essenciais para compreender o contexto deste trabalho. Inicialmente, a Seção 2.1 aborda o sistema de banco de dados adotado neste projeto, fornecendo uma visão geral de suas funcionalidades e importância para a coleta, transformação e análise dos dados provenientes da microrrede. Em seguida, a Seção 2.2 explora o embasamento teórico utilizado para a escolha do modelo adotado no projeto, destacando as razões e critérios considerados na tomada de decisão.

2.1 Banco de dados

O termo banco de dados originou-se do termo inglês *Databanks*, no entanto, o termo foi substituído pela palavra *Databases* (Base de dados), devido à sua significação mais apropriada para uma finalidade específica (SILVA, 2005). Porém antes de se entender o que é banco de dados, é necessário entender a diferença entre informação e dado. De acordo com (SILVA, 2005), dados são fatos brutos, representações simbólicas de observações, medidas e descrições de entidades, eventos, fenômenos ou qualquer outra coisa que possa ser registrada. Por outro lado, informações são dados que passaram por processamento e organização significativa, resultando em contexto, conhecimento e compreensão de um determinado assunto ou situação. As informações agregam valor aos dados ao fornecerem compreensão, significado e relevância para os usuários, sendo capaz de orientar a tomada de decisões, apoiar ações e gerar conhecimento.

Um BD é uma coleção organizada de dados inter-relacionados, que são armazenados e acessados por meio de um sistema computacional. Esses dados representam informações sobre entidades, eventos, fenômenos ou qualquer outra coisa que possa ser registrada (SILVA, 2005). Em suma, é um sistema que permite o armazenamento, organização e gerenciamento de grandes volumes de dados, onde fornece uma estrutura para a coleta, edição, modificação entre os diferentes conjuntos de dados, o que facilita a obtenção de informações relevantes e a realização de análises detalhadas.

Existem diversos exemplos de bancos de dados utilizados em diferentes áreas e contextos, entre eles banco de dados de mídia social, que são usados em plataformas, como Facebook, Twitter e Instagram, possuem bancos de dados que armazenam informações de perfil de

usuário, postagens, conexões entre usuários, curtidas, comentários, entre outros. Esses bancos de dados são projetados para suportar grande volume de dados e alta taxa de acesso.

Portanto, é importante considerar as vantagens e desvantagens ao decidir utilizar um banco de dados, levando em conta as necessidades e características específicas do ambiente em que será aplicado. A seguir, cita-se algumas vantagens e desvantagens do uso de bancos de dados segundo (BEGG, 2010).

Vantagens:

- Armazenamento eficiente de grandes quantidades de informações;
- Acesso rápido e fácil aos dados armazenados;
- Compartilhamento de informações com outros usuários autorizados;
- Redução do tempo e dos custos necessários para armazenar e gerenciar informações;
- Facilidade para atualizar e manter os dados armazenados;
- Maior segurança e controle de acesso aos dados;
- Possibilidade de automatizar processos de negócios;
- Facilidade para geração de relatórios e análises.

Desvantagens:

- Necessidade de treinamento especializado para configurar, gerenciar e manter o banco de dados;
- Alto custo para a aquisição e manutenção de softwares e hardware necessários;
- Possibilidade de perda de dados em caso de falha no sistema;
- Possibilidade de problemas de segurança, como invasões ou roubo de dados;
- Possibilidade de dificuldades na integração com outros sistemas;
- Necessidade de planejamento cuidadoso e esforço significativo para projetar um banco de dados eficiente;
- Possibilidade de problemas de desempenho e escalabilidade em bancos de dados muito grandes ou complexos;
- Dependência de fornecedores de software e hardware.

Um banco de dados relacional é aquele que segue o modelo relacional, no qual as informações são organizadas em tabelas com colunas e linhas. Cada tabela tem uma chave

primária, que serve para identificar exclusivamente cada linha, e as tabelas são conectadas através de chaves estrangeiras para formar relações (ELMASRI e NAVATHE, 2010).

O BD relacional é projetado para armazenar dados estruturados e consistentes, o que o torna ideal para aplicações que requerem consistência e precisão, como sistemas financeiros e de gestão empresarial. Algumas tecnologias de banco de dados relacional incluem PostgreSQL, MySQL, Oracle e SQL Server (CONNOLLY, 2015).

Por outro lado, um banco de dados não relacional, também conhecido como NoSQL (Not Only SQL), é projetado para armazenar e gerenciar dados não estruturados e sem relação pré-definida (CONNOLLY, 2015).

É importante ressaltar que, ao longo dos anos, essa tecnologia tem evoluído e atualmente existem diversos modelos em uso. Nesse contexto, este trabalho de conclusão de curso procura focar em explorar e apresentar informações relevantes sobre os sistemas de banco de dados relacional e não relacional, abordando suas características, diferenças e aplicações.

2.2 Embasamento para escolha do modelo de banco de dados adotado no trabalho

A escolha entre um banco de dados relacional e não relacional deve levar em consideração as necessidades específicas da aplicação em questão, como: o tipo de dados que serão armazenados, a quantidade de dados, a complexidade das consultas e as necessidades de escalabilidade (NETO, 2017).

PostgreSQL é um gerenciador de BD relacional de objeto de código aberto que otimiza muito o trabalho de quem precisa administrar informações nesses níveis. Esta ferramenta é de fácil instalação e de uso prático, proporcionando uma série de vantagens, especialmente com o uso de extensões, sendo essencial na criação e no gerenciamento de aplicações, como sites e aplicativos (POSTGRESQL, 2023).

O PostgreSQL é compatível com relacional (SQL) e não relacional (JSON) e oferece funções SQL avançadas, incluindo chaves estrangeiras, subconsultas e gatilhos. O PostgreSQL também é altamente extensível, permitindo que você defina tipos de dados e gere funções personalizadas (POSTGRESQL, 2023).

Quando estão em funcionamento, as estruturas dessas aplicações, como: Aplicações de Internet das Coisas (IoT, do inglês *Internet of Things*), sistemas de gerenciamento de dados empresariais; aplicações Web, entre outros, precisam realizar consultas constantes ao banco de dados, para então carregar informações importantes. Paralelamente, usuários que fazem

manutenções também precisam ter um acesso rápido, seguro e facilitado a esses dados. Com o PostgreSQL, essa rotina se torna mais prática (POSTGRESQL, 2023).

No contexto de aplicações em funcionamento, é essencial que as consultas ao banco de dados sejam rápidas e eficientes, tanto para a carga de informações importantes, como para as atividades de manutenção. O PostgreSQL facilita essa rotina, oferecendo um acesso rápido, seguro e prático aos dados, contribuindo para o bom desempenho e funcionamento das aplicações.

Portanto, o uso do PostgreSQL justifica-se pela sua compatibilidade com modelos relacional e não relacional, suas funções avançadas de SQL, sua extensibilidade e sua capacidade de proporcionar um acesso rápido e eficiente aos dados, contribuindo para a eficácia e praticidade no desenvolvimento e gerenciamento de aplicações.

3 METODOLOGIA PARA INTEGRAÇÃO DO BANCO DE DADOS E A MICRORREDE

Neste capítulo, será discutido o processo de definição dos objetivos do banco de dados e das informações que serão armazenadas nele. Para que um BD atenda às necessidades de forma correta, é essencial que os objetivos e as informações a serem armazenadas sejam definidos previamente. Além disso, será abordada a importância de identificar a origem dos dados e os respectivos componentes do laboratório responsáveis por fornecer esses dados. Esse processo de definição é fundamental para a criação de um banco de dados que atenda às necessidades específicas de um determinado contexto de negócio ou aplicação.

3.1 Definição dos objetivos do banco de dados

A criação de um banco de dados para informações padronizadas relacionadas a microrredes em um laboratório de universidade é um trabalho de grande importância técnica. O desenvolvimento desse BD permitirá que os dados coletados durante os experimentos realizados no laboratório sejam armazenados de forma padronizada, possibilitando a análise e interpretação desses dados em busca de insights e informações importantes.

Além disso, a padronização dos dados armazenados em um banco de dados ajuda a garantir a precisão e a confiabilidade das informações coletadas. Isso é particularmente importante no caso de microrredes elétricas, que envolvem vários elementos e parâmetros complexos, como geração, armazenamento e distribuição de energia elétrica (GUERRERO, CHANDORKAR, *et al.*, 2013).

O banco de dados também pode ser utilizado para gerar relatórios e análises que ajudarão os pesquisadores a entender melhor o comportamento da MR em diferentes condições e cenários. Isso pode contribuir para o aprimoramento dos modelos e técnicas utilizados no desenvolvimento de microrredes mais eficientes e sustentáveis.

Criar um banco de dados em ferramenta gratuita para um laboratório de pesquisa do curso de engenharia de controle e automação pode oferecer diversas vantagens, tais como:

- Armazenamento e organização de informações: um banco de dados pode ser utilizado para armazenar informações coletadas de experimentos ou disponibilizadas para

análise. Isso torna mais fácil para os alunos e professores acessarem essas informações de forma organizada e estruturada;

- **Facilidade de acesso:** um banco de dados pode ser acessado de qualquer lugar a qualquer momento, desde que haja uma conexão com a internet. Isso significa que os alunos podem acessar materiais de estudo e projetos a partir de suas próprias casas ou em qualquer outro lugar com conexão à internet;
- **Aprendizado colaborativo:** um banco de dados compartilhado pode permitir que os alunos colaborem em projetos e trabalhos em grupo, facilitando a comunicação e o compartilhamento de informações;
- **Melhoria no desempenho:** o acesso a informações relevantes e atualizadas pode melhorar o desempenho dos alunos no curso, permitindo que eles se mantenham atualizados e aprimorem seu conhecimento;
- **Redução de custos:** um banco de dados gratuito pode reduzir custos para a universidade e para os alunos, tornando o acesso a materiais de ensino e aprendizado mais acessível e econômico.

Além dessas vantagens, é importante ressaltar que um banco de dados bem desenvolvido e gerenciado pode ser uma ferramenta fundamental para a inovação e desenvolvimento no campo da Engenharia Elétrica e da Engenharia de Controle e Automação, permitindo a coleta e armazenamento de dados em tempo real, análises de desempenho e aprimoramento de projetos. Em resumo, a criação de um BD pode trazer diversas vantagens, e facilidade no acesso e aprendizado colaborativo, contribuindo para a inovação e desenvolvimento na área.

3.2 Componentes da Microrrede e Identificação da origem dos dados

Essa seção abordará a importância da identificação e compreensão das fontes de dados e dos componentes do laboratório que geram e fornecem esses dados. É fundamental compreender a origem dos dados e suas características como o formato (imagem, texto, números) para uma gestão adequada e utilização eficaz dessas informações no contexto do laboratório.

3.2.1 *Microrredes*

As microrredes são compostas por recursos energéticos distribuídos e cargas elétricas que trabalham em conjunto como uma única rede autônoma, podendo operar tanto conectadas à rede de distribuição elétrica convencional quanto de forma autônoma, em uma operação conhecida como "ilhada", contribuindo significativamente para a mitigação de distúrbios transitórios e servindo de recurso para uma resposta mais rápida do sistema (MARTINS, 2022).

O desenvolvimento de MR tem sido apontada como uma alternativa promissora para garantir uma confiabilidade no fornecimento de energia, no entanto, a escassez de laboratórios especializados pode representar um obstáculo para a implantação (MARTINS, 2022).

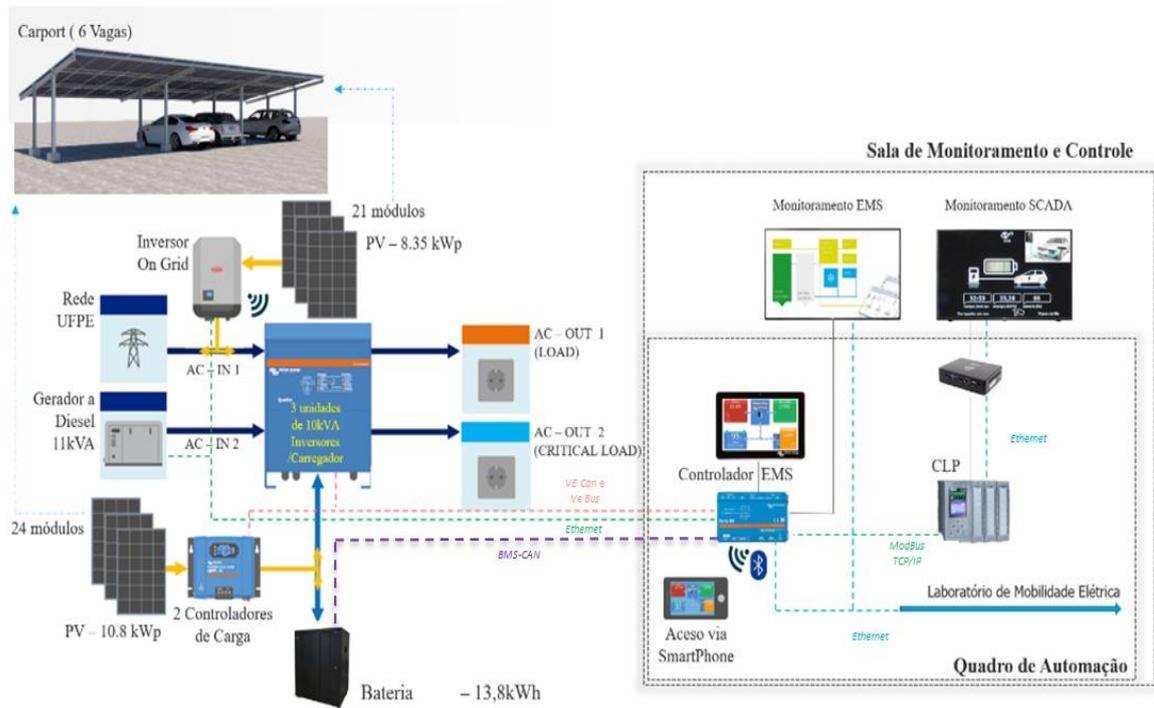
Todavia, visando sanar esse problema, o LAM da Universidade Federal de Pernambuco (UFPE) está equipado com uma microrrede híbrida com a capacidade de simular cenários de carregamento de veículos elétricos e aplicações operacionais de *Battery Energy Storage System* (BESS), além de possuir um sistema de monitoramento e supervisão de dados. O uso de bancos de dados é crucial para análise e otimização de desempenho desses sistemas, incluindo análise de operação de armazenamento de energia baseados em baterias de íons de lítio.

Dessa forma, torna-se necessário conhecer os componentes que formam a MR do LAM e que fornecerão as informações a serem armazenadas no banco de dados.

3.2.2 *Topologia da MR do LAM*

A topologia da microrrede do LAM é constituída por uma estrutura híbrida que permite a realização de diversos cenários em diferentes aplicações. Entre os sistemas disponíveis, destacam-se os sistemas fotovoltaicos (FV) autônomos com armazenamento, *back-up* de gerador a diesel, sistemas conectados à rede, entre outros. Para facilitar a compreensão da microrrede, o seu layout é apresentado na Figura 1.

Figura 1 - Layout da microrrede do LAM.



Fonte: (MARTINS, 2022).

No LAM, a microrrede é composta por diversos subsistemas e componentes, incluindo o inversor *on grid*, os inversores/carregadores, os controladores de carga, o sistema de armazenamento, *carport* (estacionamento coberto com placas solares instaladas na sua estrutura) e o gerador, conforme demonstrado no layout da Figura 1. É importante ressaltar que este layout representa de forma simplificada as conexões e possibilidades do sistema real, apresentado na Figura 2, o qual é integrado fisicamente no quadro elétrico, conforme evidenciado na Figura 3.

A Figura 2 possibilita demonstrar de forma visual o sistema real da Figura 1, onde estão destacados o inversor *on-grid* (1), os três inversores/carregadores (2), os dois controladores de carga (3), o sistema de armazenamento (4) e o gerador (5) (MARTINS, 2022).

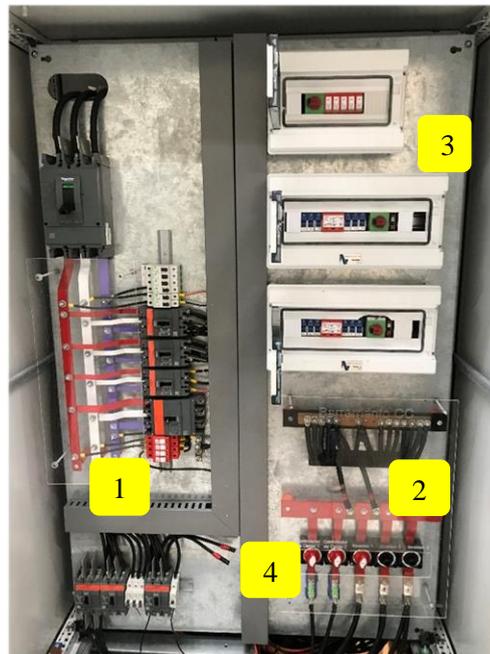
Figura 2 - Microrrede do LAM.



Fonte: Próprio Autor.

De outra forma, a Figura 3 evidencia o quadro elétrico em que os equipamentos são integrados fisicamente. No quadro portanto, estão dispostos os barramentos dos circuitos de corrente alternada (1) e de corrente contínua (2), *string box* (3) para conexão dos módulos FV ao inversor *on-grid* e aos controladores de carga, e chaves seccionadoras (4).

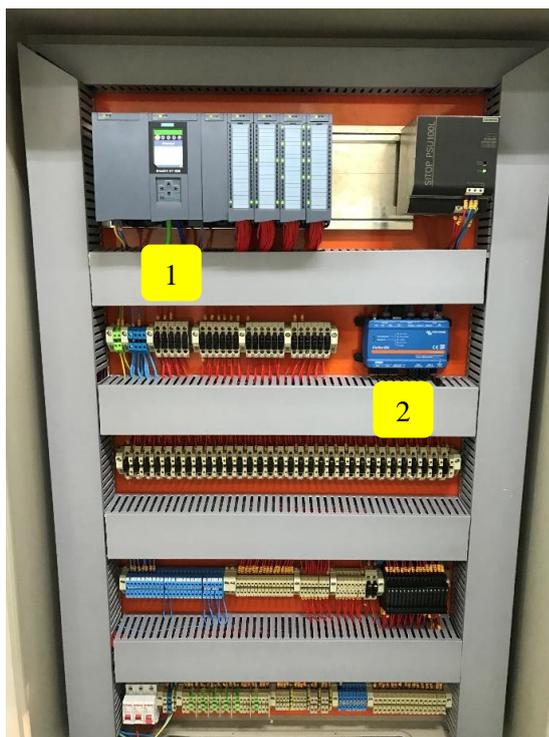
Figura 3 - Quadro elétrico da Microrrede.



Fonte: Próprio Autor.

Além do quadro elétrico, a microrrede do LAM conta com o quadro de automação, apresentado na Figura 4, que abriga o Controlador Lógico Programável (CLP) (1) e o sistema de monitoramento Cerbo GX (2), desenvolvido pela Victron Energy, responsável pela conectividade entre os equipamentos da microrrede.

Figura 4 - Quadro de Automação.



Fonte: Próprio Autor.

3.2.3 Comunicação com o Cerbo GX

O Cerbo GX representado na Figura 5, é um componente crucial do Sistema de Gerenciamento de Energia (SGE) da MR do LAM, desempenhando um papel essencial na monitoração, controle e otimização das atividades realizadas pelos equipamentos do sistema, incluindo inversores, baterias e carregadores. Embora possua capacidade de comunicação com esses equipamentos por meio de diferentes protocolos de comunicação, o protocolo MQTT foi o selecionado para o trabalho, devido à sua eficiência e confiabilidade na integração e processamento de dados em tempo real (HIVEMQ, 2020).

A capacidade de comunicação do Cerbo GX por meio do protocolo MQTT é fundamental para garantir uma troca de informações rápida e confiável entre os equipamentos, possibilitando

um gerenciamento mais inteligente e sustentável da energia elétrica. Com a ajuda do Cerbo GX, é possível obter informações valiosas sobre dados climáticos, consumo, geração solar, históricos de corrente, tensão e potência de cada equipamento, incluindo a bateria, através do *Battery Management System* (BMS).

Além disso, o Cerbo GX é responsável por fornecer informações em tempo real sobre o status do sistema e suas atividades por meio do Portal de Gestão Remota, disponibilizado pelo fornecedor. O *Dashboard* desse portal apresenta gráficos e tabelas que retratam a dinâmica do sistema, permitindo uma gestão mais inteligente e eficiente da energia elétrica.

Um dos equipamentos mais importantes do sistema é o inversor solar *on-grid*, que é responsável pela alimentação do circuito de iluminação do *carport*. Em conjunto com o Cerbo GX, esse equipamento é fundamental para garantir a eficiência energética da MR do LAM e proporcionar uma gestão mais sustentável da energia elétrica. Assim, o Cerbo GX é fundamental para garantir a operação eficiente do sistema de gerenciamento de energia, monitorando, controlando e otimizando o desempenho de todos os componentes presentes no sistema. Sua capacidade de integrar e processar dados de diferentes fontes torna o Cerbo GX um componente chave para a gestão inteligente e sustentável da energia elétrica (MARTINS, 2022).

Figura 5 - Cerbo GX.



Fonte: (VICTRON ENERGY, 2021).

3.2.4 Protocolo e comunicação MQTT

MQTT (do inglês, *Message Queue Telemetry Transport*) é um protocolo leve de mensagens de publicação/assinatura que é projetado para ser de fácil implementação em dispositivos (OASIS OPEN, 2023).

O protocolo MQTT é uma tecnologia de comunicação leve e eficiente que se tornou um padrão para a Internet das Coisas (IoT, do inglês *Internet of Things*). O MQTT permite que dispositivos se comuniquem com a internet e com outros dispositivos de maneira simples e confiável, usando assinatura de tópicos e mensagens com um *payload* (conteúdo) e um tópico que especifica o assunto da mensagem. Além disso, o MQTT possui recurso de Qualidade de Serviço (QoS, do inglês *Quality of Service*) que permite que os dispositivos garantam a entrega de mensagens, tornando-o uma opção ideal para ambientes com conexões instáveis ou de baixa largura de banda. Deste modo, O MQTT é amplamente utilizado em dispositivos IoT para troca de dados entre eles e a internet. A Figura 6, portanto, apresenta a visão geral do fluxo de comunicação do protocolo (HIVEMQ, 2020).

O protocolo MQTT é composto por três componentes principais: o cliente MQTT, o *broker* (negociador) MQTT e os tópicos MQTT. A conexão entre o cliente e o *broker* é baseada no protocolo TCP/IP e ocorre sempre entre o cliente e o *broker*, nunca diretamente entre clientes (ALFACOMP, 2020).

O cliente MQTT é o dispositivo que envia e recebe as mensagens do *broker* MQTT. Ele é responsável por se conectar ao *broker* e se inscrever em tópicos específicos para receber as mensagens pertinentes. O cliente MQTT também pode publicar mensagens em um tópico específico, que serão entregues a todos os clientes que estão inscritos nesse determinado tópico (ALFACOMP, 2020).

O *broker*, por sua vez, é o intermediário entre os clientes MQTT. Ele é responsável por receber mensagens publicadas pelos clientes e entregá-las aos clientes inscritos em tópicos correspondentes. Ele também é responsável por manter o controle de sessões persistentes, garantindo que as mensagens sejam entregues corretamente, mesmo se um cliente se desconectar temporariamente (ALFACOMP, 2020).

Os tópicos MQTT são o mecanismo de assinatura utilizado pelos clientes MQTT para receber mensagens relevantes. Cada tópico é identificado por uma *string* única que descreve o assunto da mensagem. Quando um cliente se inscreve em um tópico específico, ele receberá todas as mensagens publicadas neste tópico. Os tópicos MQTT permitem que os clientes se

comuniquem de forma seletiva e eficiente, recebendo apenas as mensagens relevantes para suas necessidades (ALFACOMP, 2020).

Figura 6 - Visão geral do protocolo MQTT.



Fonte: Adaptado de (ALFACOMP, 2020).

O protocolo TCP/IP é utilizado para estabelecer a conexão entre o cliente e o *broker* MQTT. O protocolo TCP/IP é responsável por estabelecer uma conexão confiável e garantir que as mensagens sejam entregues corretamente. A conexão TCP/IP é estabelecida quando o cliente MQTT envia uma mensagem CONNECT para o *broker* MQTT. Assim, o *broker* MQTT responde com uma mensagem CONNACK, indicando se a conexão foi estabelecida com sucesso. Uma vez que a conexão for estabelecida, o *broker* mantém a conexão aberta até que o cliente envie um comando de desconexão. A Tabela 1 apresenta os códigos de retorno disponíveis na mensagem CONNACK, que é enviada pelo *broker* em resposta à mensagem CONNECT enviada pelo cliente MQTT (HIVEMQ, 2020).

Tabela 1 - Códigos e respostas de retorno.

Código de retorno	Resposta do código de retorno
0	Conexão aceita
1	Conexão recusada, versão de protocolo não aceita
2	Conexão recusada, identificador rejeitado
3	Conexão recusada, servidor indisponível
4	Conexão recusada, mau <i>username</i> ou senha
5	Conexão recusada, não autorizado

Fonte: Adaptado de (HIVEMQ, 2020).

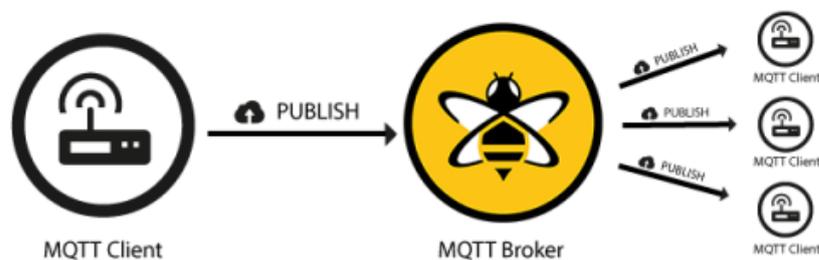
Os códigos de retorno disponíveis variam de 0 a 5, sendo que o código 0 indica que a conexão foi aceita pelo *broker* e os códigos de 1 a 5 indicam que a conexão foi recusada por algum motivo específico, como falta de autorização, nome de usuário ou senha inválidos, entre outros.

3.2.5 Publicação e inscrição MQTT

Uma das características mais importantes do protocolo MQTT é o seu modelo de publicação e inscrição, que permite uma comunicação entre dispositivos conectados em rede.

Ao publicar uma mensagem, o dispositivo cliente envia uma mensagem *PUBLISH* contendo um tópico, um nível de QoS, um flag de retenção, o *payload* e um identificador de pacote. O tópico é uma *string* hierárquica que permite a organização das mensagens em diferentes níveis de granularidade. O QoS determina a qualidade da entrega da mensagem e pode ser definido como: 0 (entrega única); 1 (entrega garantida pelo menos uma vez); 2 (entrega garantida exatamente uma vez). O flag de retenção indica se a mensagem deve ser armazenada como última mensagem válida do tópico. O *payload* é o conteúdo real da mensagem e pode ser de vários tipos, incluindo textos, imagens, dados binários e dados criptografados (A. MANOWSKA, 2022). O processo de publicação de mensagem MQTT pode ser visto na Figura 7.

Figura 7 - Processo de publicação de mensagem MQTT.

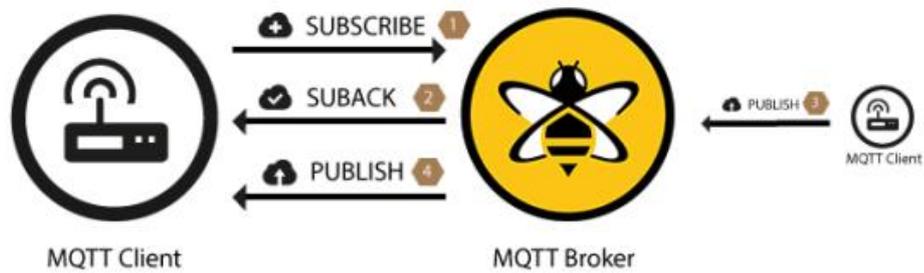


Fonte: (HIVEMQ, 2020).

De outra forma, os dispositivos interessados em receber as mensagens se inscrevem em tópicos específicos disponibilizados pelo *broker*, enviando uma mensagem *SUBSCRIBE* contendo um identificador de pacote e uma lista de tópicos e QoS desejados. O *broker* responde

com uma mensagem *SUBACK*, contendo o identificador de pacote e uma lista de códigos de retorno que indicam o resultado de cada inscrição. A partir desse momento, todas as mensagens publicadas no tópico em questão serão entregues aos dispositivos inscritos, de acordo com o nível de QoS definido na publicação (A. MANOWSKA, 2022). A Figura 8, portanto, apresenta a sequência de comunicação MQTT.

Figura 8 - Sequência de comunicação MQTT.



Fonte: (HIVEMQ, 2020).

4 DESENVOLVIMENTO DO BANCO DE DADOS

Neste capítulo, serão abordados dois temas importantes na criação de um banco de dados: a modelagem dos dados e o desenvolvimento da estrutura necessária para o armazenamento local e em conexão como a nuvem. Para que um BD seja eficiente, é fundamental que haja uma modelagem adequada dos dados a serem armazenados e que a estrutura seja desenvolvida de forma coerente com essa modelagem. Além disso, será descrito o processo de aquisição dos dados por meio da comunicação MQTT. Serão apresentados exemplos de dados coletados e como eles foram estruturados para serem armazenados no banco de dados.

A etapa de modelagem de BD foi realizada com o auxílio das ferramentas online descritas anteriormente como o BR Modelo Web e o DBdiagram. Para o desenvolvimento das aplicações, incluindo o BD do projeto e o código para comunicação, foi utilizado o software PostgreSQL e software Visual Studio Code.

4.1 Modelagem dos dados

A modelagem de dados é o processo de criação de um modelo de dados para a criação de um modelo de dados para que o software em questão armazene dados em um BD. É uma etapa fundamental na criação de um banco de dados eficiente, pois uma modelagem adequada permite uma melhor organização e acesso aos dados, além de garantir a integridade e a consistência dos mesmos. Esse processo é dividido em três etapas: modelo conceitual, modelo lógico e modelo físico, cada uma com suas próprias características e considerações. Além disso, é importante selecionar a abordagem mais adequada para o contexto em que o BD será utilizado, a fim de garantir a eficiência e a eficácia do sistema (SILVA, 2023).

4.1.1 *Modelo Conceitual*

É a etapa inicial da modelagem de BD, que tem como objetivo entender como o negócio realmente funciona e como as informações são relacionadas de uma forma abstrata, de mais alto nível, ou seja, que está próxima da linguagem humana, permitindo a criação de um modelo lógico e, posteriormente um modelo físico. É desenvolvido a partir dos requisitos do sistema, extraídos na fase de levantamento de requisitos. O modelo pode ser elaborado de forma textual

ou por meio do modelo ER, uma técnica que envolve a criação de diagramas que representam os objetos do sistema e as relações entre eles (SILVA, 2023).

O banco de dados é uma representação de elementos presentes no mundo real, frequentemente referida como um "mini mundo". Em outras palavras, é uma versão simplificada do mundo real, que pode ser armazenada e acessada de forma organizada através do banco de dados.

No modelo conceitual textual, a coleta de informações é feita através de textos, e as entidades e atributos são usados para representar objetos ou eventos do mundo real que serão registrados em uma tabela. A entidade é a própria tabela e o atributo é uma propriedade ou característica da entidade. Por exemplo, o cliente e o pedido são entidades que podem ser registradas em uma tabela. Os dados relacionados a essas entidades são armazenados em colunas específicas, como nome, endereço, telefone e e-mail, na tabela de cliente, conforme ilustrado na Figura 9 (SILVA, 2023).

Figura 9 - Exemplo modelo conceitual textual.

<p>1. Cliente Dados necessários: nome, endereço, telefone, e-mail.</p> <p>2. Pedido Dados necessários: número do pedido, data do pedido, código do cliente, código do vendedor.</p>

Fonte: Próprio Autor.

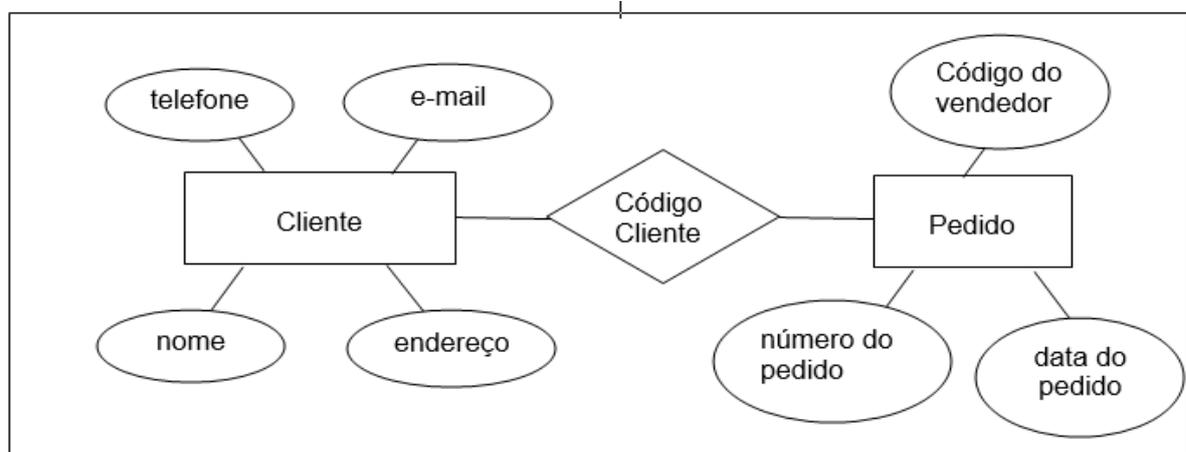
4.1.1.1 Modelo Entidade-Relacionamento

Modelo Entidade-Relacionamento é uma ferramenta de visual usada para representar e modelar dados, essa abordagem foi criada por Peter Chen em 1976, sendo considerada um padrão para a modelagem conceitual. atributos e relacionamentos. No modelo ER, as entidades são representadas como objetos e os relacionamentos entre elas são representados como linhas que conectam esses objetos. As entidades são apresentadas dentro de retângulos, que incluem o nome da entidade, enquanto que os atributos são listados dentro de balões, que são conectados ao retângulo correspondente. Os relacionamentos são representados por linhas que conectam os retângulos das entidades, e são apresentados dentro de um losango, como é apresentado na Figura 10 (SILVA, 2023).

Existe o conceito de cardinalidade, onde é importante em diagramas ER e indica quantas instâncias de uma entidade podem se relacionar com outra. Existem duas facetas da cardinalidade: a mínima e a máxima. A cardinalidade mínima é o número mínimo de instâncias de uma entidade que devem se relacionar com outra entidade. Já a cardinalidade máxima é o número máximo de instâncias que podem se relacionar, podendo ser 1 ou N. Existem três classificações possíveis de relacionamentos em diagramas ER (SILVA, 2023):

- 1x1: indica que uma instância de uma entidade está associada a apenas uma instância da outra entidade e vice-versa;
- 1xN: indica que uma instância de uma entidade pode estar associada a várias instâncias da outra entidade, mas cada instância da outra entidade está associada a apenas uma instância da primeira entidade.;
- NxN: indica que várias instâncias de uma entidade podem estar associadas a várias instâncias da outra entidade.

Figura 10 - Exemplo modelo conceitual diagrama ER.



Fonte: Próprio Autor.

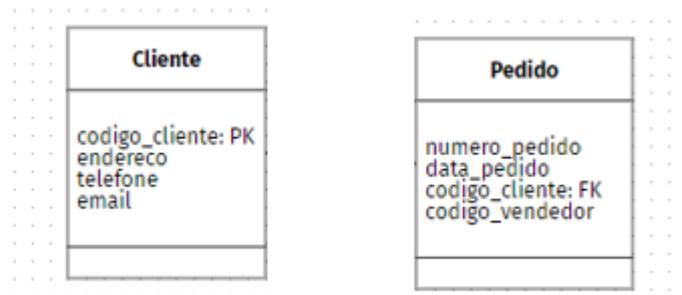
4.1.2 Modelo Lógico

O modelo lógico, descreve a organização das tabelas, colunas, chaves e relacionamentos, permitindo visualizar como as tabelas e colunas serão organizadas e relacionadas, independentemente da tecnologia específica de implementação, ou seja, o modelo lógico pode

ser utilizado tanto em bancos de dados relacionais, como MySQL, quanto em bancos de dados não-relacionais, como NoSQL (SILVA, 2023).

Após a estruturação do modelo conceitual em forma de diagrama, as entidades do modelo conceitual são mapeadas em tabelas no modelo lógico de banco de dados, nesse processo de mapeamento, cada entidade do modelo conceitual é transformada em uma tabela no modelo lógico, com cada atributo da entidade sendo mapeado em uma coluna na tabela correspondente. Dessa forma, o modelo conceitual, que é representado em termos de entidades e seus relacionamentos, é convertido em um modelo lógico, que é representado em termos de tabelas e suas colunas. A Figura 11 apresenta um exemplo de modelo lógico (SILVA, 2023).

Figura 11 - Exemplo de modelo lógico



Fonte: Próprio Autor

Desta forma, a Figura 12 apresenta a tabela após preenchimento dos dados. Onde, a tabela "cliente" representa uma entidade de negócio chamada "cliente", que tem atributos como nome, endereço, telefone e e-mail. A coluna "id" é a chave primária da tabela, que é usada para identificar unicamente cada registro da tabela. Já a tabela "pedido" representa uma entidade de negócio chamada "pedido", que tem atributos como número do pedido, data do pedido e código do vendedor. A coluna "cliente" é a chave estrangeira que faz referência a tabela "cliente". Essa chave estrangeira define um relacionamento entre a tabela "pedido" e "clientes", indicando que um registro na tabela "pedido" está relacionado a um registro de tabela "cliente".

Figura 12 - Exemplo modelo lógico preenchido da tabela de pedido.

Id [PK] integer	numero_pedido character varying (20)	data_pedido date	id_cliente integer	id_vendedor integer
1	PED001	2022-02-10	1	1
2	PED002	2022-02-15	1	2
3	PED003	2022-03-01	2	1
4	PED004	2022-03-05	3	3

Fonte: Próprio Autor

4.1.3 Modelo Físico

É uma representação técnica que define como os dados serão armazenados e organizados no SGBD. Essa etapa leva em conta a estrutura definida no modelo lógico e as necessidades de desempenho e eficiência da aplicação ou sistema que utilizará o banco de dados. A Linguagem de Consulta Estruturada (SQL) é a mais comum para a implementação do modelo físico e as informações são organizadas em uma estrutura tabular, onde as tabelas representam entidades e as colunas representam campos, definindo o tipo de dados que será armazenado. A Tabela 2 apresenta um exemplo de modelo físico (SILVA, 2023).

Tabela 2 - Exemplo modelo físico detalhamento de colunas (campos) da tabela cliente.

Nome do Campo	Tipo de dado	Tamanho do Campo
Id	Numérico	Dígitos
Nome	Alfanumérico	100 Caracteres
Endereço	Alfanumérico	200 Caracteres
Telefone	Alfanumérico	20 Caracteres
E-mail	Alfanumérico	100 Caracteres

Fonte: Próprio Autor

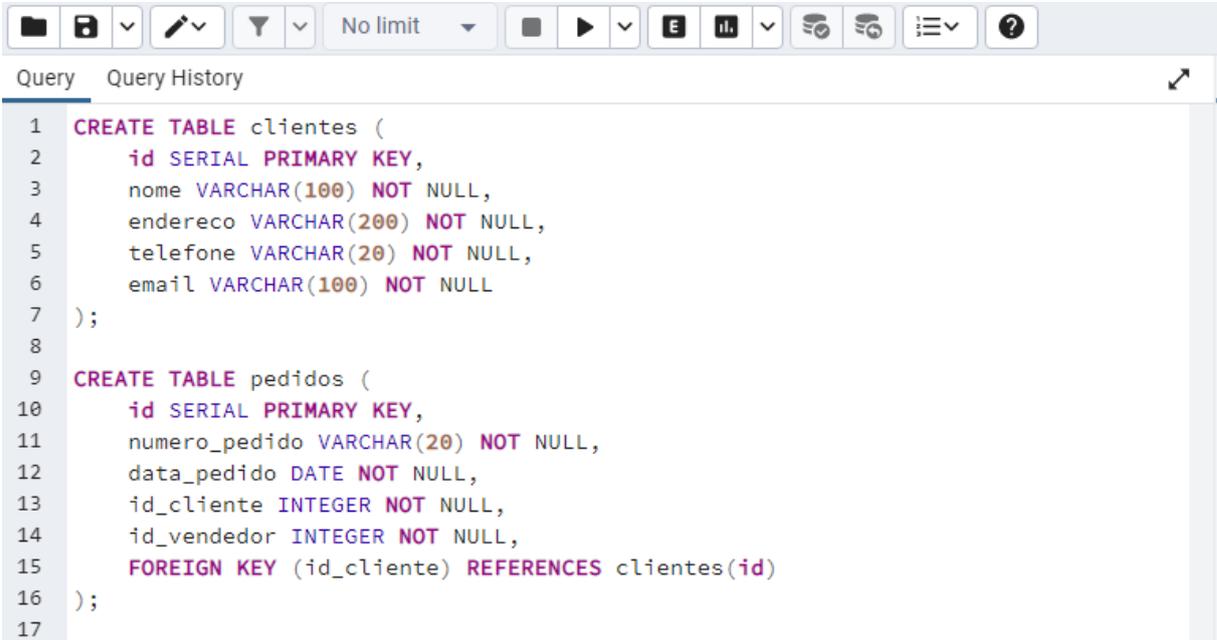
A Tabela 2 possui as seguintes indicações:

- Id: o campo numérico que armazena um identificador único para cada registro;
- Nome: campo alfanumérico com espaço para 100 caracteres, utilizando pra armazenar o nome do cliente;
- Endereço: campo alfanumérico com espaço para 200 caracteres, utilizado para armazenar o endereço do cliente;

- Telefone: campo alfanumérico com espaço para 20 caracteres, utilizado para armazenar o número de telefone do cliente;
- E-mail: campo alfanumérico com espaço para 100 caracteres, utilizado para armazenar o endereço de e-mail do cliente.

Para representar este modelo físico em uma linguagem SQL compatível com o PostgreSQL, utiliza-se a seguinte sintaxe apresentada na Figura 13:

Figura 13 - Exemplo modelo físico na linguagem SQL para o PostgreSQL.



```
1 CREATE TABLE clientes (  
2     id SERIAL PRIMARY KEY,  
3     nome VARCHAR(100) NOT NULL,  
4     endereco VARCHAR(200) NOT NULL,  
5     telefone VARCHAR(20) NOT NULL,  
6     email VARCHAR(100) NOT NULL  
7 );  
8  
9 CREATE TABLE pedidos (  
10    id SERIAL PRIMARY KEY,  
11    numero_pedido VARCHAR(20) NOT NULL,  
12    data_pedido DATE NOT NULL,  
13    id_cliente INTEGER NOT NULL,  
14    id_vendedor INTEGER NOT NULL,  
15    FOREIGN KEY (id_cliente) REFERENCES clientes(id)  
16 );  
17
```

Fonte: Próprio Autor

Este código cria as tabelas "clientes" e "pedido" com as colunas especificadas, definindo a coluna "id" como uma chave primária e as outras colunas como valores obrigatórios ("NOT NULL"). O tipo de dados "SERIAL" é utilizado para o campo "id" para gerar automaticamente valores únicos para cada registro inserido na tabela.

4.2 Aplicações da modelagem de dados aos itens do LAM

Essa seção apresenta a modelagem de dados para o LAM e seu resultado final, utilizando os conceitos de modelagem de dados e técnicas de desenvolvimento de banco de dados para

criar uma estrutura de dados eficiente e flexível, capaz de atender às necessidades específicas do projeto. O resultado final deste capítulo consistirá em um sistema de gerenciamento energético remoto que será robusto e confiável, onde permitirá a criação de um *Dashboard* apresentado no próximo capítulo, que mostrará aos usuários o monitoramento da produção e do consumo de energia elétrica em microrredes de forma eficiente e econômica.

É possível descrever uma representação conceitual utilizando informações coletadas em formato de texto, definindo as partes importantes do sistema, tais como suas entidades, atributos e relacionamentos, sem se preocupar com a implementação física do banco de dados. O objetivo principal é obter uma visão clara e completa dos dados que o sistema irá gerenciar para facilitar a criação de sua estrutura de armazenamento e gerenciamento posteriormente. A Figura 14 apresenta o modelo conceitual textual do BD do LAM.

Figura 14 - Modelo conceitual textual do banco de dados do LAM.

1. Fronius:

Dados necessários: Potência Total AC, Energia Total AC, Potência AC L1, Energia AC L1, Tensão AC L1, Corrente AC L1, Potência AC L2, Energia AC L2, Tensão AC L2, Corrente AC L2, Potência AC L3, Energia AC L3, Tensão AC L3, Corrente AC L3, Potência Injetada na rede L1, Potência Injetada na rede L2, Potência Injetada na rede L3, Data de criação.

2. Alarme 1 da bateria:

Dados necessários: Temperatura de alta da carga - BYD, Data de criação.

3. Alarme 2 da bateria:

Dados necessários: Temperatura de baixa da carga - BYD, Data de criação.

4. Alarme 3 da bateria:

Dados necessários: Temperatura de alta - BYD, Data de criação.

5. Alarme 4 da bateria:

Dados necessários: Temperatura de baixa - BYD, Data de criação.

6. Bateria -BYD:

Dados necessários: Potência da bateria- BYD, Tensão da bateria- BYD, Corrente da bateria- BYD, Data de criação.

7. Bateria -SOC:

Dados necessários: soc, Data de criação.

8. Carregador solar -01:

Dados necessários: Potência PV -Solarcharger1, Tensão PV -Solarcharger1, Corrente PV -Solarcharger1, Data de criação.

9. Carregador solar -02:

Dados necessários: Potência PV -Solarcharger1, Tensão PV -Solarcharger1, Corrente PV -Solarcharger1, Data de criação.

10. Quattro -01:

Dados necessários: Potência de saída total L1, Tensão de saída total L1, Corrente de saída total L1, Data de criação.

11. Sistema:

Dados necessários: Potência saída total, Data de criação.

12. Sistema Conexão- 01:

Dados necessários: Conexão IN1, Data de criação.

13. Sistema Conexão- 02:

Dados necessários: Conexão IN2, Data de criação.

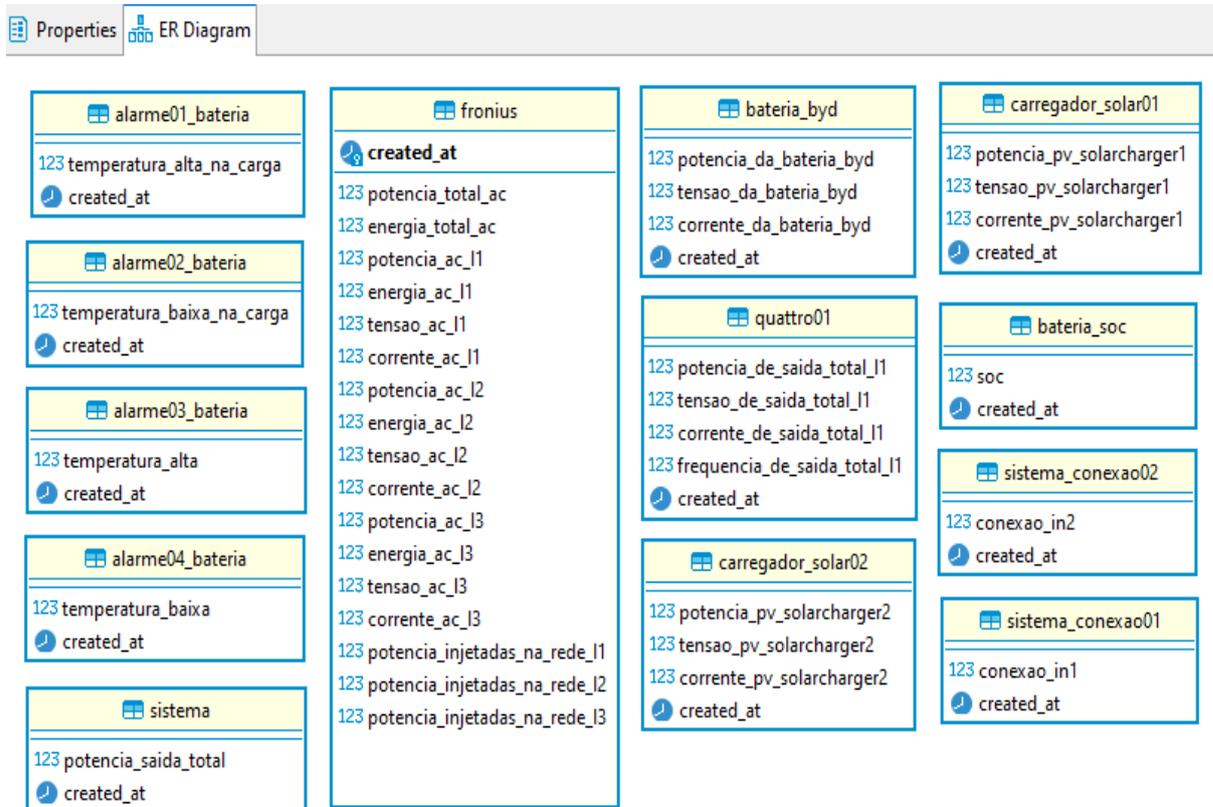
Fonte: Próprio Autor.

Após definir as entidades, atributos do sistema em formato de texto, o próximo passo é criar o modelo lógico do banco de dados. Isso envolve traduzir as informações coletadas anteriormente para um formato visual que represente de forma clara a estrutura do banco de dados. Portanto, são criadas as tabelas para cada entidade, onde as colunas representam os atributos e as linhas representam as instâncias dessas entidades.

Ao modelar o banco de dados para as entidades, notou-se que cada registro dessa entidade pode ser identificado de forma única pela sua data de criação, uma vez que essa informação é única e não se repete para diferentes registros. Portanto, decidiu-se que não seria necessário utilizar uma chave primária para essa entidade. Em vez disso, a coluna de data de criação será utilizada como um valor único e distintivo para identificar cada registro dessa entidade. Também foi determinado que, mesmo se tratando de um banco de dados relacional, as tabelas funcionariam de forma independente, sem a necessidade de relacionamentos entre elas.

É possível criar tabelas independentes sem relacionamentos explícitos entre elas. Os autores enfatizam que, embora seja comum usar relacionamentos entre tabelas através de chaves primárias e estrangeiras, não é uma exigência do modelo relacional e que a flexibilidade do modelo permite diferentes abordagens. A criação de tabelas independentes pode ser útil em casos em que é necessário manter um alto nível de autonomia entre as entidades, ou quando as relações entre elas são complexas e difíceis de modelar de maneira adequada. Deste modo, a Figura 15 apresenta o modelo lógicos do BD do LAM.

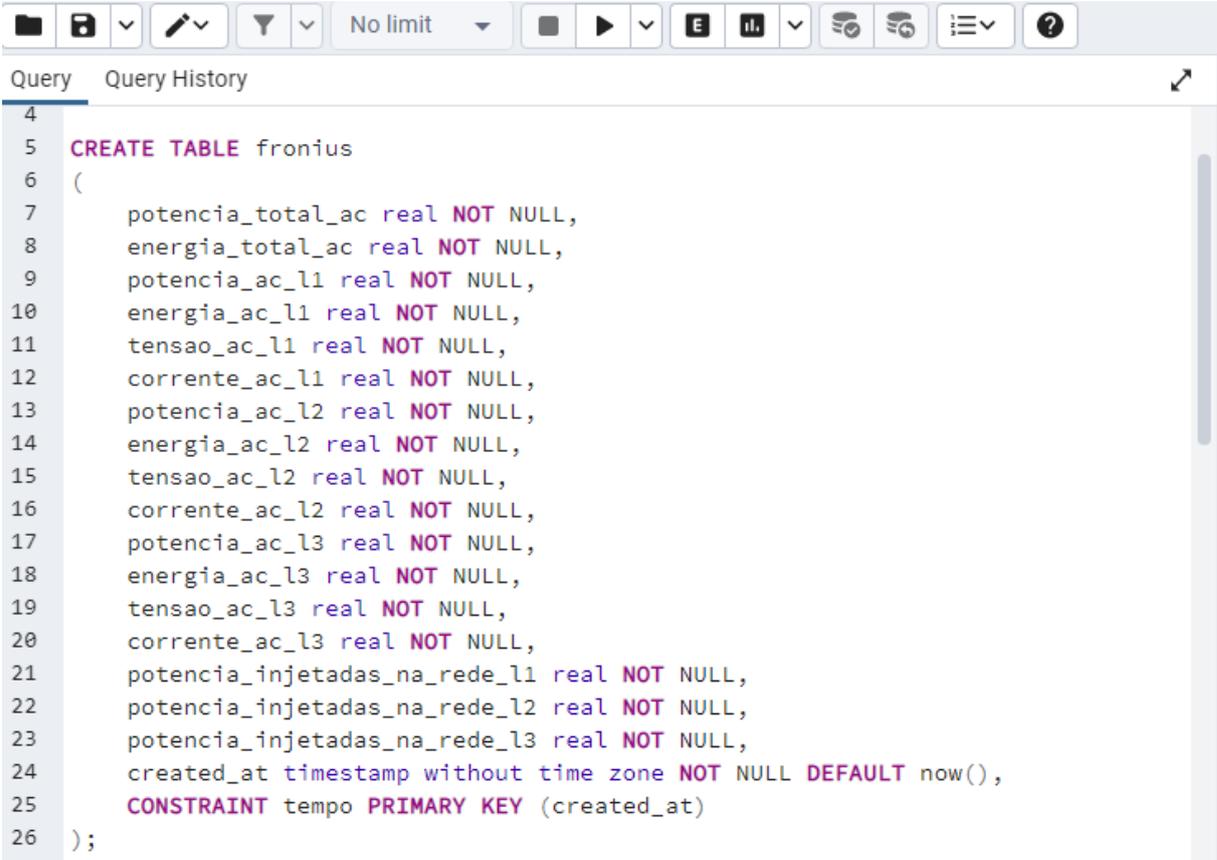
Figura 15 - Modelo lógico do banco de dados do LAM.



Fonte: Próprio Autor.

Após a definição do modelo lógico do banco de dados, é possível prosseguir com a criação do modelo físico utilizando a linguagem SQL apropriada para a plataforma, neste caso, o PostgreSQL. Esses comandos podem ser executados diretamente no PostgreSQL para criar as tabelas que foram definidas no modelo lógico do banco de dados. É possível executá-los em um cliente SQL, como o PgAdmin que é um software gratuito e de código aberto para administração de bancos de dados PostgreSQL, que fornece uma interface gráfica para gerenciar bancos de dados, tabelas, esquemas, usuários, permissões e outras funcionalidades relacionadas ao banco de dados. O PgAdmin é um software multiplataforma, o que significa que ele pode ser executado em diferentes sistemas operacionais, como Windows, Linux e macOS. O modelo físico é apresentado na Figura 16.

Figura 16 - Modelo físico na linguagem SQL para o PostgreSQL.



```

4
5 CREATE TABLE fronius
6 (
7     potencia_total_ac real NOT NULL,
8     energia_total_ac real NOT NULL,
9     potencia_ac_l1 real NOT NULL,
10    energia_ac_l1 real NOT NULL,
11    tensao_ac_l1 real NOT NULL,
12    corrente_ac_l1 real NOT NULL,
13    potencia_ac_l2 real NOT NULL,
14    energia_ac_l2 real NOT NULL,
15    tensao_ac_l2 real NOT NULL,
16    corrente_ac_l2 real NOT NULL,
17    potencia_ac_l3 real NOT NULL,
18    energia_ac_l3 real NOT NULL,
19    tensao_ac_l3 real NOT NULL,
20    corrente_ac_l3 real NOT NULL,
21    potencia_injetadas_na_rede_l1 real NOT NULL,
22    potencia_injetadas_na_rede_l2 real NOT NULL,
23    potencia_injetadas_na_rede_l3 real NOT NULL,
24    created_at timestamp without time zone NOT NULL DEFAULT now(),
25    CONSTRAINT tempo PRIMARY KEY (created_at)
26 );

```

Fonte: Próprio Autor.

Este código cria uma tabela chamada "Fronius" com as colunas especificadas, definindo a coluna "created_at" como uma coluna obrigatória ("NOT NULL") e com um valor padrão ("DEFAULT") definido como a data e hora atual ("now ()"). A coluna "created_at" também está sendo definida como a chave primária da tabela usando a restrição "CONSTRAINT". Todas as outras colunas especificadas também são obrigatórias ("NOT NULL"), o que significa que o valor deve ser fornecido ao inserir novos registros na tabela. O tipo de dados "real" no PostgreSQL é equivalente ao tipo "float" em outras linguagens de programação, representando um número de ponto flutuante de precisão simples, com 6 dígitos de precisão. Ou seja, é um tipo de dado numérico que pode representar números decimais com casas decimais.

Após a criação da tabela "Fronius" no banco de dados PostgreSQL usando a linguagem SQL apropriada, o mesmo processo foi repetido para as demais tabelas. Com o banco de dados devidamente configurado, será estabelecida a comunicação com os equipamentos para coleta os dados, o que será exposto no capítulo seguinte.

4.3 Ferramentas utilizadas para desenvolvimento do trabalho

A seção 3.3 do presente estudo aborda as ferramentas utilizadas ao longo da pesquisa, destacando a sua importância para o desenvolvimento e visualização dos dados. Dentre as ferramentas adotadas estão: a linguagem de programação Python, o Visual Studio Code (VS Code), o PostgreSQL, o Power BI e o Azure.

4.3.1 Python

Python é uma linguagem de programação de alto nível e interpretada, criada por Guido van Rossum em 1991 e amplamente usada em uma variedade de aplicações, desde a automação de tarefas simples até o desenvolvimento de aplicativos web, ciência de dados, inteligência artificial e aprendizado de máquina. Wes McKinley, em seu livro, afirma que:

"Python é uma linguagem de programação mais populares da ciência de dados devido à sua ampla gama de bibliotecas e ferramentas disponíveis para processamento e análise de dados" (MCKINNEY, 2018).

Para o desenvolvimento deste trabalho, foi escolhida essa linguagem, em sua versão 3.10, devida sua capacidade em atender as necessidades de aquisição, manipulação de dados, bem como integração com as ferramentas de banco de dados, visualização de dados e persistência em nuvem.

4.3.2 Visual Studio Code

O Visual Studio Code é um editor de código-fonte gratuito e multiplataforma, desenvolvido pela Microsoft e lançado em 2015. Esse editor, usado na versão 1.79.2, amplamente usado na comunidade de desenvolvimento de software foi tomado como escolha para o ambiente de desenvolvimento devido a aderência com as ferramentas já utilizadas e os ganhos de produtividade que esse editor proporciona.

4.3.3 PostgreSQL

PostgreSQL é um sistema de gerenciamento de banco de dados relacional de código aberto que utiliza linguagem SQL. Ele tem sua origem em 1986 como parte do projeto POSTGRE na universidade da Califórnia tendo mais 35 anos de desenvolvimento. Dentre suas características, destaca-se a possibilidade de utilização de diferentes tipos de dados, como: Primitivos (Inteiros, numéricos, *string*, booleano), estruturados (data/tempo, *array*, *range*), documento (JSON, XML), geométrico (ponto, linha, polígono) e customizados (POSTGRESQL, 2023).

4.3.4 Power BI

Power BI é uma plataforma unificada para *Business Intelligence* (BI) focada na visualização, modelagem e conexão de dados. O seu uso é fundamentado pela sua facilidade de implementação online e integração com a plataforma Azure. Com o Power BI, pode-se conectar facilmente a fontes de dados, visualizar e descobrir conteúdo importante e compartilhá-lo com todas as pessoas que quiser, gerando relatórios e *Dashboards* que mostram números, estatísticas, valores, listas e gráficos de maneira simples e intuitiva. Isso porque a ferramenta tem o propósito de facilitar a visualização de dados e promover um maior envolvimento do espectador em relação à maneira como essas informações são apresentadas (MICROSOFT, 2023).

4.3.5 Azure

A plataforma Microsoft Azure foi utilizada com o intuito de armazenar os dados obtidos e devido a sua integração com o Power BI. A Azure oferece uma ampla variedade de soluções, incluindo bancos de dados relacionais, como o SQL Server, e soluções NoSQL, como o Azure Cosmos DB, entre outras. Com essas opções, é possível gerenciar, armazenar e analisar dados de maneira eficiente, escalável e flexível, permitindo a adaptação da capacidade de armazenamento de dados de acordo com a necessidade. Além disso, a plataforma oferece recursos avançados de segurança, gerenciamento e monitoramento, garantindo total controle e visibilidade sobre os dados na nuvem. Adicionalmente, a plataforma oferece serviços de banco

de dados gerenciados, tais como o Azure Database for MySQL e o Azure Database for PostgreSQL. Esses serviços permitem a implantação rápida de bancos de dados MySQL e PostgreSQL na nuvem, sem a necessidade de gerenciamento de infraestrutura. (MICROSOFT, 2023).

4.3.6 BR Modelo Web

BR Modelo Web é uma ferramenta voltada para a modelagem de BD no formato Entidade-Relacionamento (ER). O mesmo foi utilizado para a elaboração do modelo conceitual e modelo lógico do banco de dados (BR MODELO WEB, 2023).

4.3.7 Dbdiagram

DBdiagram é uma ferramenta de design de diagrama de banco de dados relacional. Através desta ferramenta, o modelo físico será desenvolvido, permitindo a criação, edição e visualização de modelos de BD de maneira visual e intuitiva, por meio de uma sintaxe simples (HOLISTICS, 2023).

4.4 Implementação da integração do SGER-MR

Através da implementação do SGER-MR (Sistema de Gerenciamento de Energia em Microrredes), tem-se como objetivo apresentar uma plataforma que centralize e coordene as operações de visualização dos dados, além de descrever sua configuração e demonstrar um método para realizar o backup dos dados na nuvem. Esta seção irá abordar a plataforma de visualização de dados, detalhar sua configuração e também fornece uma abordagem para efetuar o backup dos dados na nuvem.

4.4.1 Integração entre Python e MQTT

Amplamente utilizada para implementação de aplicações de Internet das Coisas (IoT) e monitoramento de sensores. O Python é uma linguagem de programação popular e de alto nível,

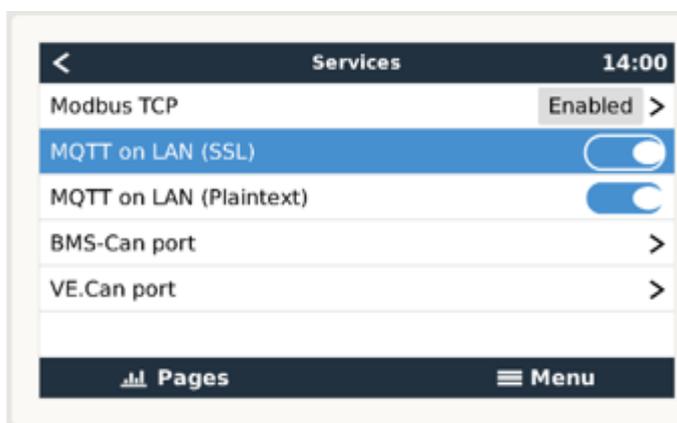
que possui diversas bibliotecas para facilitar a integração com protocolos de comunicação como o MQTT.

A biblioteca *paho-mqtt* é uma das mais utilizadas para a implementação de clientes MQTT em Python. Ela fornece uma interface simples e eficiente para publicar e subscrever tópicos MQTT, além de suportar diversos recursos avançados como QoS e TLS (*Transport Layer Security*). Essa biblioteca permite a criação de um cliente MQTT em Python, capaz de se conectar a um *broker* MQTT e troca mensagens com outros dispositivos conectados à rede (ECLIPSE FOUNDATION, 2023).

A criação do código-fonte em Python foi realizada por meio da utilização do ambiente de desenvolvimento integrado (IDE) Visual Studio Code. Para desenvolver um cliente MQTT em Python, é necessário definir as configurações de conexão com o *broker* MQTT, incluindo o host, a porta e as configurações de autenticação. No entanto, no caso em questão, foi dispensada a necessidade de autenticação de usuário e senha por meio da ativação e uso da configuração *Plaintext*.

O dispositivo Cerbo GX, que é parte integrante do sistema de Gerenciamento de Energia (EMS), possui um *broker* MQTT com configurações que permitem o envio de dados da instalação como mensagens com tópicos MQTT para a rede. É possível ativar o *broker* MQTT do Cerbo GX no menu configurações, conforme apresentado na Figura 17. É importante notar que a configuração *Plaintext* foi ativada e utilizada, o que não requer autenticação de usuário e senha e utiliza a porta 1883.

Figura 17 - Ativação do protocolo MQTT no Cerbo-GX.



Fonte: (OLIVEIRA, 2023).

Para que a comunicação seja feita de forma ativa com outros dispositivos por meio do protocolo MQTT, é necessário adicionar um Add-on *broker* MQTT, chamado Mosquitto

Broker. Após instalado, é realizada uma conexão entre os *brokers* para que o *Mosquitto Broker* colete os tópicos publicados pelo *Broker GX*, agindo apenas como “ponte”, passando os tópicos inscritos. Para garantir que a conexão permaneça ativa e estável, é criada uma automação através de um programa em Python, enviando regularmente a mensagem *keepalive*. Se a conexão for desativada, pode haver perda de dados ou problemas de comunicação entre os equipamentos do sistema de gerenciamento de energia, portanto, essa automação é fundamental para garantir o funcionamento adequado do sistema. A Figura 17 apresenta tal configuração.

Figura 18 - Automação para mensagem *keepalive*.

```
# Defina o tópico e a mensagem que deseja publicar
TOPIC = "R/102c6b9d30f3/system/0/Serial"
MESSAGE = "Hello"

async def publish_run():
    while True:
        publish.single(TOPIC, MESSAGE, hostname=HOST, port=PORT, client_id=CLIENT_ID)
        await asyncio.sleep(10)
```

Fonte: Próprio Autor.

Após a configuração das informações de conexão com o *broker* MQTT, o próximo passo para criar um cliente MQTT em Python é definir os tópicos MQTT aos quais o cliente irá se inscrever e/ou publicar mensagens. A biblioteca `paho.mqtt.client` fornece as funções "subscribe" e "publish" para se inscrever em tópicos e publicar mensagens em tópicos MQTT, respectivamente.

A função "subscribe" recebe como parâmetro o nome do tópico ao qual se deseja se inscrever. Já a função "publish" recebe como parâmetro o nome do tópico ao qual a mensagem será enviada e a própria mensagem a ser enviada. Com essas funções, é possível realizar a troca de mensagens em tempo real entre os dispositivos conectados à rede MQTT (ECLIPSE FOUNDATION, 2023).

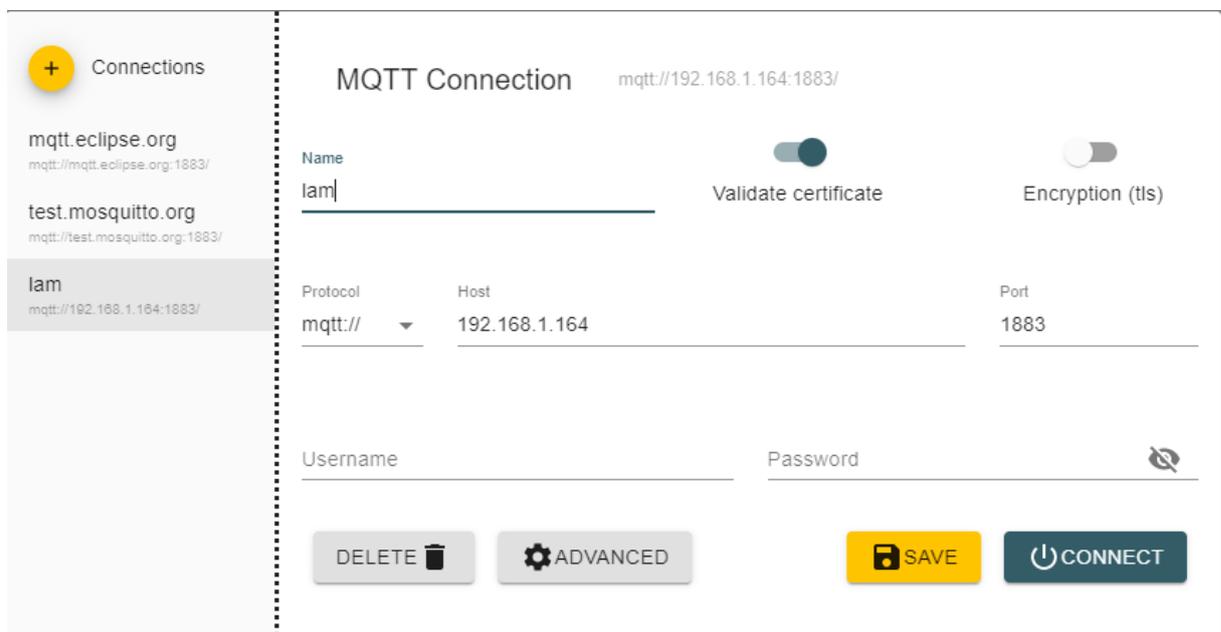
Após a inicialização e verificação da integração e do Add-on, utilizou-se o programa MQTT Explorer para visualizar os tópicos MQTT disponíveis para os dispositivos conectados ao *broker*, como mostram a Figura 19 e Figura 20. Esta etapa foi necessária devido à grande quantidade de tópicos disponíveis e publicados pelo Cerbo GX, permitindo navegar facilmente pelos tópicos de interesse, já estabelecidos no modelo conceitual através da Figura 14, de modo que cada dado da tabela requer um endereço específico chamado Tópico.

Desta forma, desenvolveu-se um arquivo no formato JSON (Figura 21) contendo uma lista de tópicos que são associados aos parâmetros específicos da tabela, permitindo uma organização clara e estrutura dos dados a serem recebidos pelo sistema. Essa abordagem contribui para a eficiência e facilidade de manutenção do sistema uma vez que os tópicos podem ser configurados e gerenciados de forma centralizado.

Ressalta-se que a utilização da linguagem Python permitiu que esses dados foram coletados e armazenados periodicamente (de cinco em cinco minutos) em um banco de dados, através da especificação de um intervalo de tempo de monitoramento.

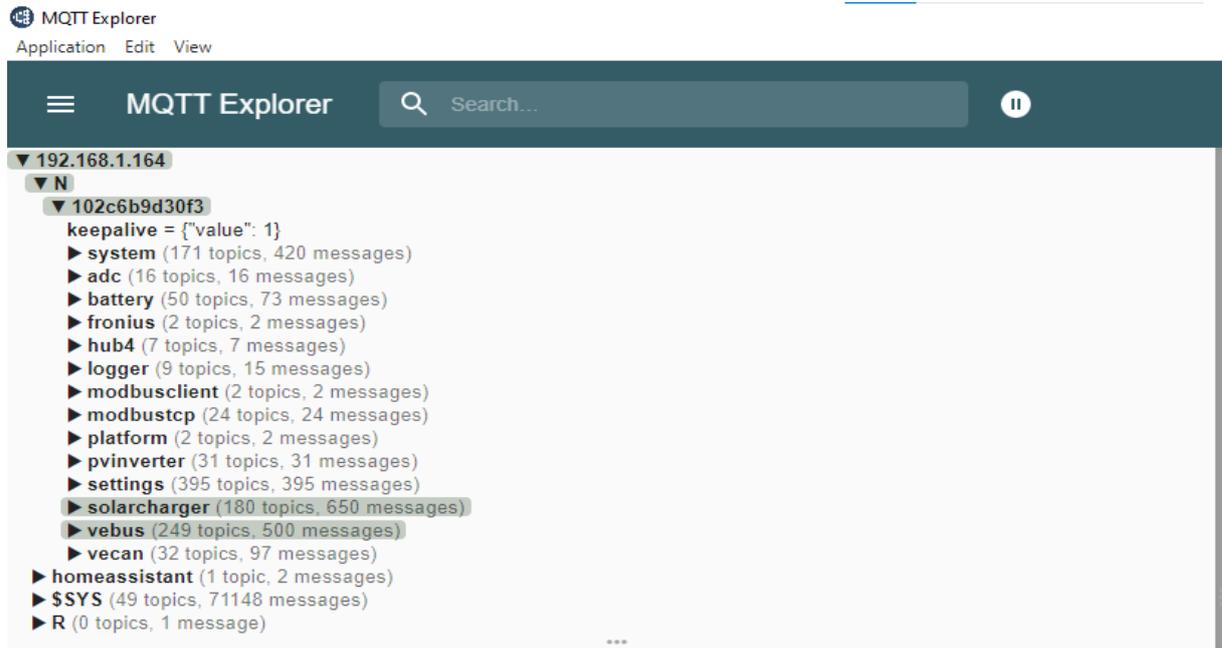
É importante mencionar que, caso não seja recebido nenhum dado dentro do intervalo de tempo estabelecido, o programa irá automaticamente adicionar o valor “0” naquele intervalo de tempo, identificando que nenhum dado foi gerado nesse período. Essas informações serão enviadas do *broker* MQTT para o banco de dados, permitindo o armazenamento e análise dos dados recebidos.

Figura 19 - Conexão com o *Broker GX*



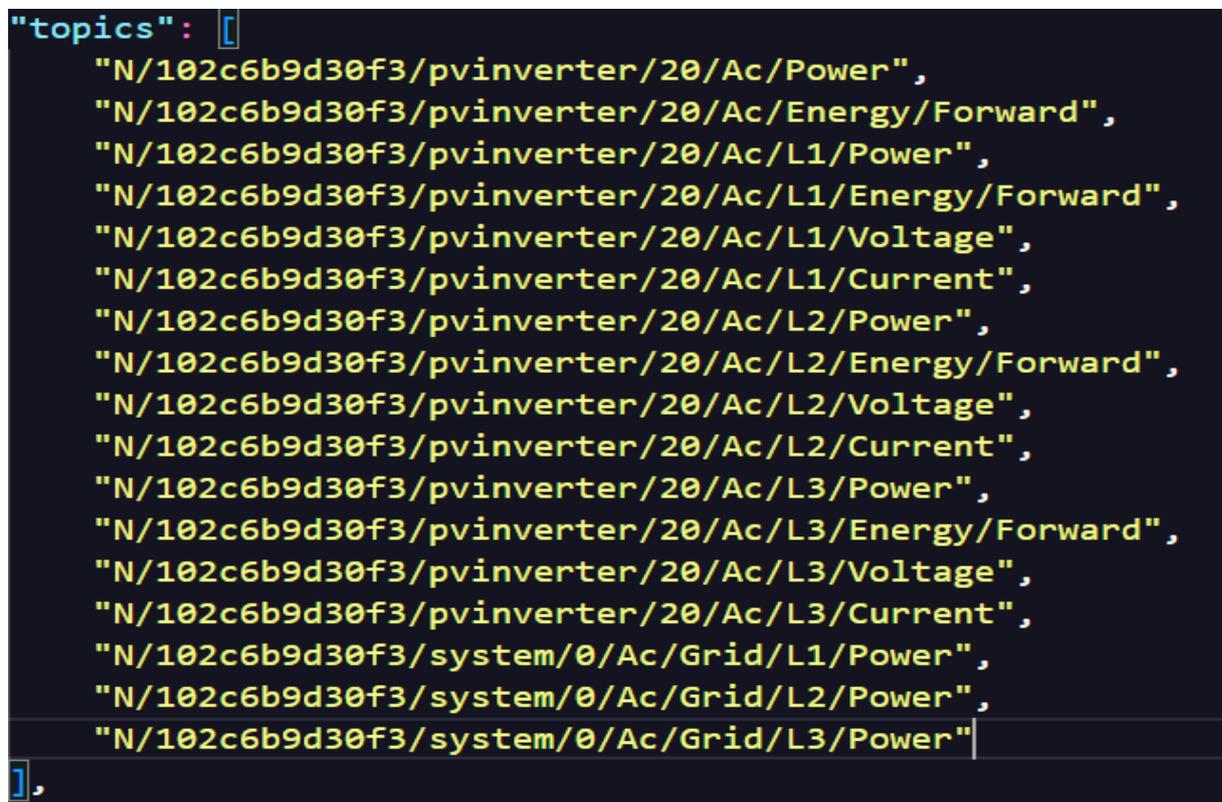
Fonte: Próprio Autor.

Figura 20 - Tópicos disponíveis



Fonte: Próprio Autor.

Figura 21 - Arquivo Json com os tópicos



Fonte: Próprio Autor.

Ao definir os tópicos MQTT e as funções correspondentes, o programa em Python pode interagir com o *broker* MQTT e trocar informações com outros dispositivos conectados à rede, permitindo a implementação de uma ampla variedade de aplicações de Internet das Coisas (IoT) e outras soluções de comunicação entre dispositivos.

4.4.2 Inserção do banco de dados na nuvem

A inserção do banco de dados na nuvem é importante dado que permite que as empresas tenham acesso a uma infraestrutura de TI altamente escalável e flexível, sem a necessidade de investir em hardware ou infraestrutura de TI própria. Adicionalmente, com a inserção do banco de dados na nuvem, as empresas podem reduzir seus custos operacionais.

A migração do banco de dados na nuvem pode oferecer maior escalabilidade, flexibilidade e redução de custos, podendo facilmente diminuir ou aumentar a sua capacidade de armazenamento de dados conforme a necessidade em um modelo de precificação flexíveis pagando apenas pelo que usa. Além disso, a infraestrutura altamente flexível da nuvem oferece aos usuários acesso fácil e gerenciamento de dados em qualquer lugar do mundo, a qualquer momento (ISMAIL, HAMZA e KOTB, 2018).

Para a criação do banco de dados PostgreSQL na plataforma Microsoft Azure, foram consultadas as informações detalhadas presentes na documentação oficial da Azure Database for PostgreSQL, disponível em (MICROSOFT, 2023). Esta documentação fornece um guia para criar um servidor e um banco de dados PostgreSQL usando o portal do Azure, além de oferecer informações importantes sobre segurança, escalabilidade e gerenciamento do banco de dados. Dessa forma, foi possível garantir uma implantação adequada e segura do banco de dados na nuvem.

Após a criação do banco de dados na plataforma Azure, é possível realizar o backup do banco de dados na máquina local onde o mesmo está localizado, utilizando ferramentas específicas para esse fim. Posteriormente, pode-se enviar o backup para o banco de dados na nuvem utilizando os comandos apropriados para o gerenciamento e transferência de dados, como o comando "pg_restore" no caso do PostgreSQL. Para se conectar ao banco de dados PostgreSQL hospedado na nuvem, foi utilizado o comando "psql" em um terminal. Esse comando permite a conexão com o banco de dados remoto e recebe diversos parâmetros, tais como:

- Host: o endereço do servidor onde o banco de dados está hospedado;

- Port: o número da porta onde o PostgreSQL está configurado para escutar as conexões, que, neste caso, é "5432", que é a porta padrão do PostgreSQL;
- Dbname: o nome do banco de dados ao qual você deseja se conectar;
- User: o nome do usuário que será utilizado para autenticar no banco de dados;
- Password: a senha correspondente ao nome de usuário informado;
- Sslmode: o modo de segurança da conexão, onde "require" significa que a conexão só será estabelecida se houver uma conexão segura SSL/TLS.

Abaixo, na Figura 22, é apresentado os comandos possíveis para se conectar ao banco de dados PostgreSQL na nuvem:

Figura 22 - Comandos de conexão com o banco de dados na nuvem.

```

URL de Conexão do PostgreSQL
postgres://labmicrorrede:{your_password}@lam-microrrede-db.postgres.database.azure.com/postgres?sslmode=require

psql
psql "host=lam-microrrede-db.postgres.database.azure.com port=5432 dbname={your_database} user=labmicrorrede password={your_password} sslmode=require"

JDBC
jdbc:postgresql://lam-microrrede-db.postgres.database.azure.com:5432/{your_database}?user=labmicrorrede&password={your_password}&sslmode=require

Node.js, Python, Ruby, PHP, C++ (libpq)
host=lam-microrrede-db.postgres.database.azure.com port=5432 dbname={your_database} user=labmicrorrede password={your_password} sslmode=require

ADO.NET
Server=lam-microrrede-db.postgres.database.azure.com;Database={your_database};Port=5432;User Id=labmicrorrede;Password={your_password};Ssl Mode=Require;

```

Fonte: (MICROSOFT, 2023).

Ao executar esse comando no terminal, é estabelecida uma conexão segura com o banco de dados remoto, permitindo que se execute comandos SQL e realize operações no banco de dados.

Para transferir um banco de dados local para a nuvem, é necessário primeiro criar um arquivo de backup do banco de dados local utilizando o comando “*pg_dump*”, que inclui informações como o nome do usuário, nome do servidor, porta do servidor, caminho e nome do arquivo de backup e nome do banco de dados local. Em seguida, o arquivo de backup pode ser restaurado na nuvem utilizando o comando “*pg_restore*”, que inclui informações como o nome do usuário, nome do servidor, porta do servidor, nome do banco de dados na nuvem e

caminho e nome do arquivo de *backup*. Para manter o banco de dados na nuvem atualizado, pode-se automatizar esse processo utilizando ferramentas como o Azure Data Factory, o qual permite criar *pipelines* de dados para mover e transformar dados entre diferentes fontes e destinos. É possível configurar um *pipeline* para executar o backup e a restauração do banco de dados em intervalos regulares, mantendo a nuvem atualizada com os dados mais recentes do banco de dados local.

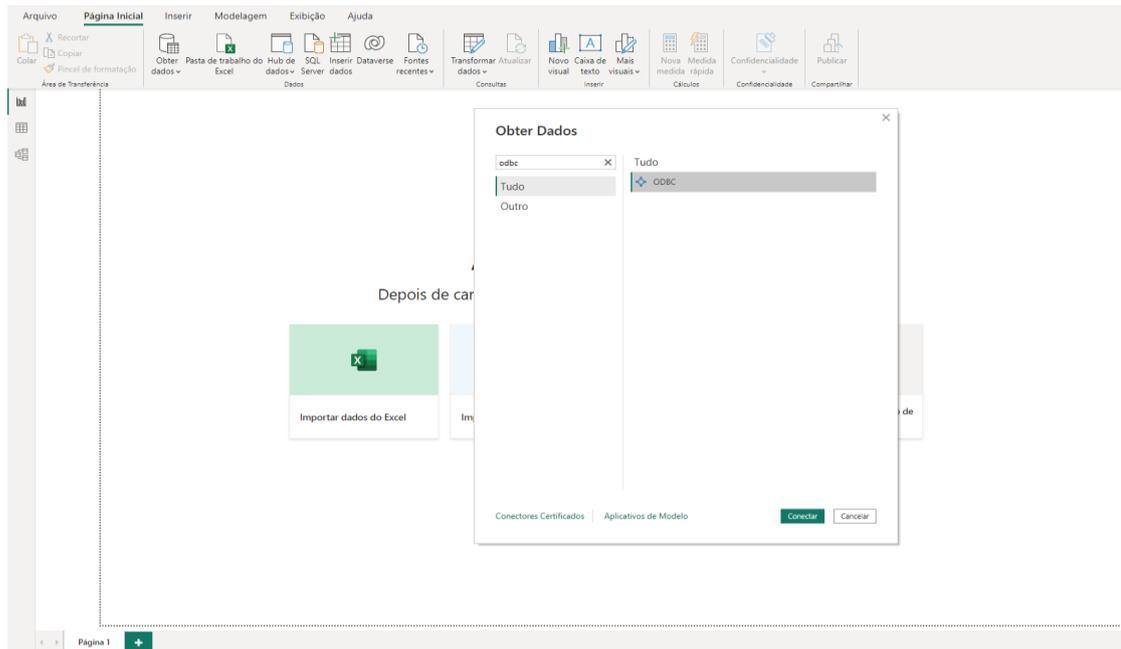
4.4.3 Configuração do Power BI

Para utilizar o Power BI foi necessário conectar-se a um banco de dados, transformar dados, criar um *Dashboard* e publicar seu relatório na nuvem. Nesse sentido, é preciso ter o Power BI Desktop instalado na máquina local e uma fonte de dados (banco de dados, planilha, etc.).

Desta forma, segue-se com a seleção do tipo de fonte de dado que se deseja conectar, como o ODBC (do inglês, *Open Database Connectivity*), o qual oferece uma maneira eficaz de estabelecer uma ponte entre o Power BI e diversas fontes de dados, facilitando a integração.

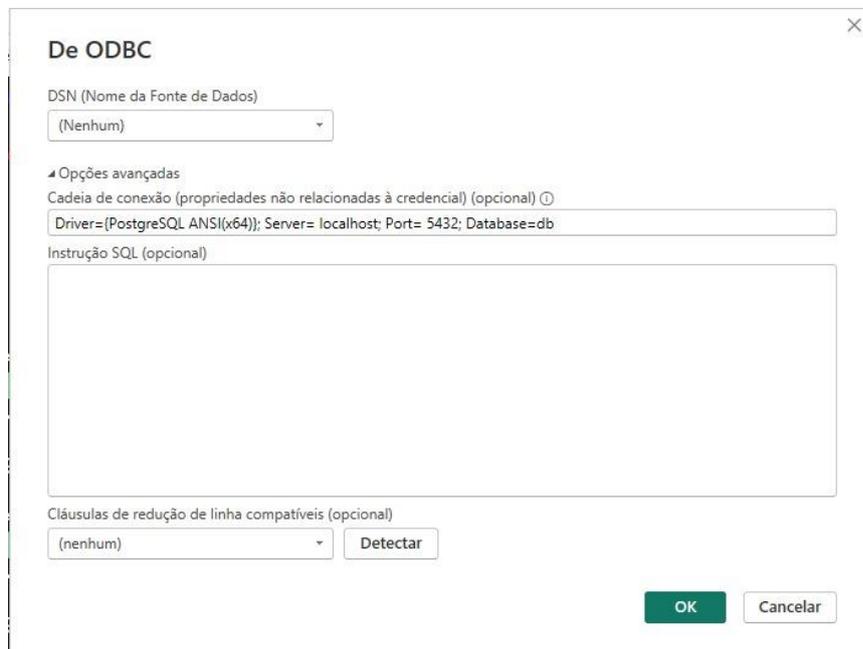
Em etapas posteriores, é necessário especificar as configurações de conexão, como Driver, servidor, porta e nome do banco de dados, como mostram a Figura 23 e Figura 24, respectivamente.

Figura 23 - Conexão de fonte de dados ODBC no Power BI Desktop.



Fonte: Próprio Autor.

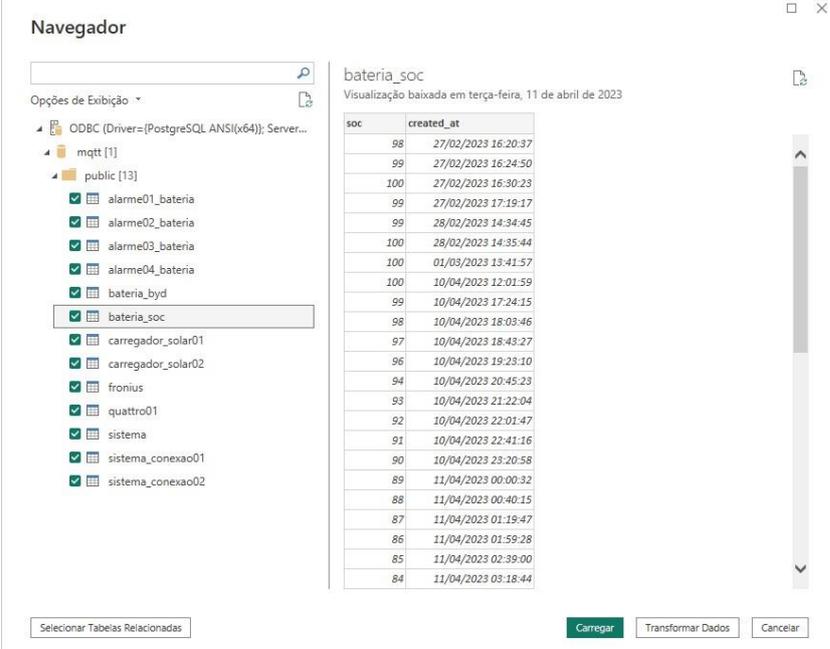
Figura 24 - Cadeia de conexão para o banco de dados.



Fonte: Próprio Autor.

A Figura 25 demonstra a escolha das tabelas ou exibições que se deseja importar para o Power BI.

Figura 25 - Seleção de tabelas para importação no Power BI.



The screenshot displays the Power BI Navigator interface. On the left, under 'Navegador', the 'public [13]' schema is expanded, and the 'bateria_soc' table is selected. The right pane shows a preview of the 'bateria_soc' table with the following data:

soc	created_at
98	27/02/2023 16:20:37
99	27/02/2023 16:24:50
100	27/02/2023 16:30:23
99	27/02/2023 17:19:17
99	28/02/2023 14:34:45
100	28/02/2023 14:35:44
100	01/03/2023 13:41:57
100	10/04/2023 12:01:59
99	10/04/2023 17:24:15
98	10/04/2023 18:03:46
97	10/04/2023 18:43:27
96	10/04/2023 19:23:10
94	10/04/2023 20:45:23
93	10/04/2023 21:22:04
92	10/04/2023 22:01:47
91	10/04/2023 22:41:16
90	10/04/2023 23:20:58
89	11/04/2023 00:00:32
88	11/04/2023 00:40:15
87	11/04/2023 01:19:47
86	11/04/2023 01:59:28
85	11/04/2023 02:39:00
84	11/04/2023 03:18:44

At the bottom of the window, there are buttons for 'Selecionar Tabelas Relacionadas', 'Carregar', 'Transformar Dados', and 'Cancelar'.

Fonte: Próprio Autor.

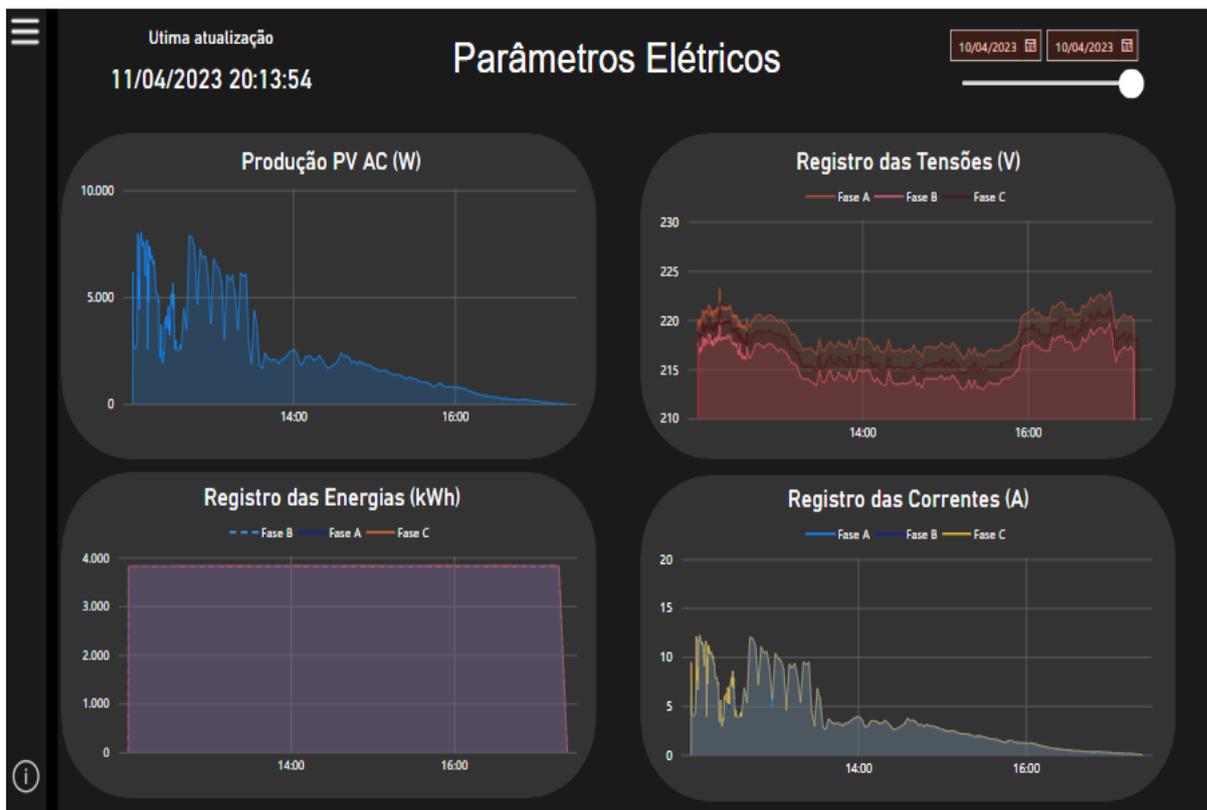
5 RESULTADOS DA *DASHBOARD* COM CONJUNTO DE DADOS DA MICRORREDE DO LAM

Neste capítulo, será apresentado um conjunto de *Dashboards* no Power BI que apresenta uma visualização detalhada dos parâmetros elétricos de cada componente da MR. É possível navegar pelas diferentes páginas no menu de navegação e selecionar o período de tempo desejado. Os *Dashboards* incluem gráficos de série temporal, gráficos de barras e indicadores, que permitem uma análise mais precisa dos valores em diferentes períodos de tempo e da última atualização do BD.

Assim, a Figura 26 representa um *Dashboard* no Power BI que contém gráficos de série temporal que mostram parâmetros elétricos do inversor *on-grid* Fronius. Na tela, é possível visualizar quatro gráficos distintos, a saber:

- Tensão AC Trifásica;
- Corrente AC trifásica;
- Produção PV AC (W);
- Registro das energias.

Cada um desses gráficos apresenta informações relevantes sobre o desempenho do inversor Fronius em diferentes áreas, permitindo o monitoramento e visualização dos históricos dos dados.

Figura 26 - *Dashboard* dos parâmetros elétricos do inversor *on-grid*.

Fonte: Próprio Autor.

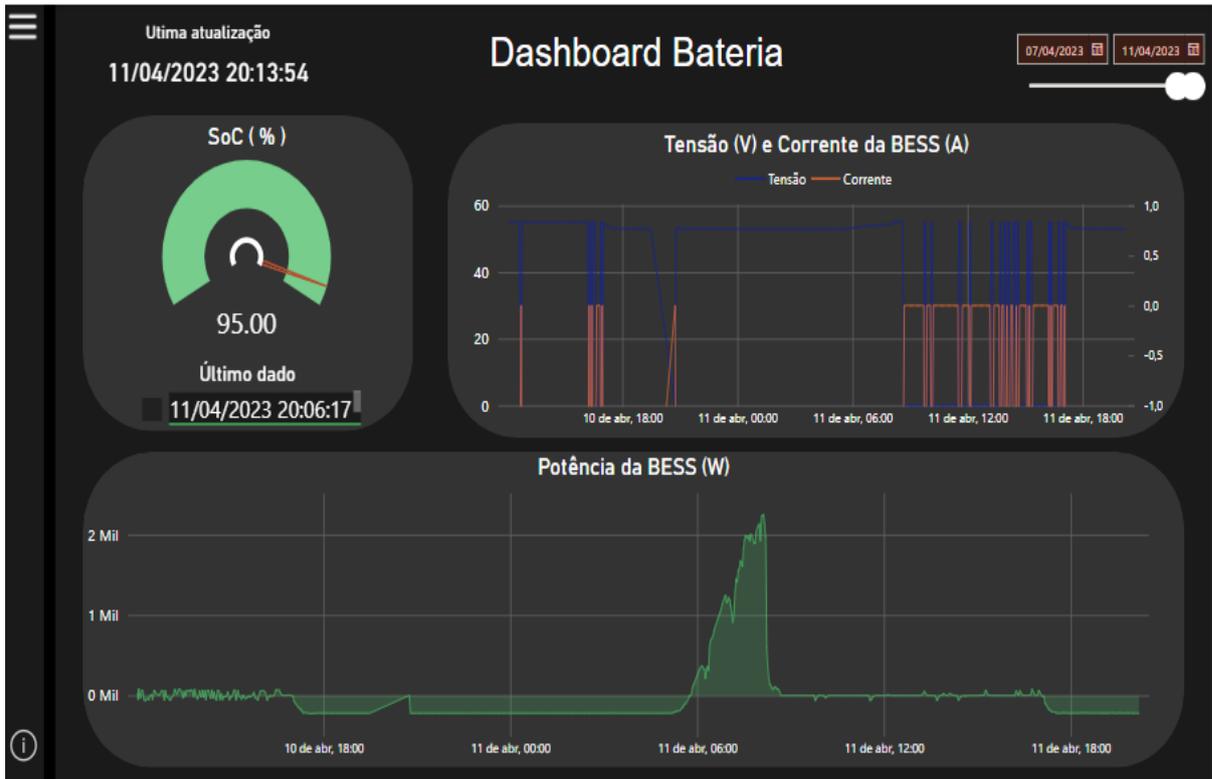
A Figura 27 em questão exibe um *Dashboard* contendo informações em tempo real dos parâmetros elétricos do inversor Fronius. É possível visualizar as Correntes, Tensão e Potência AC de cada fase, além da energia AC e a potência excedente injetada de cada fase. O *Dashboard* é alimentado por uma conexão ativa com o banco de dados, garantindo que os dados exibidos sejam sempre os mais recentes. Os parâmetros elétricos exibidos podem variar dentro de um intervalo definido, desde o valor mínimo até o máximo registrado até aquele momento, fornecendo informações precisas e atualizadas sobre o desempenho do equipamento na microrrede. Ao acionar o botão localizado no canto superior direito do *Dashboard*, é possível obter os dados mais recentes disponíveis no banco de dados, permitindo uma visualização ainda mais atualizada das informações apresentadas.

Figura 27 - *Dashboard* parâmetros elétricos em tempo real do equipamento Fronius.



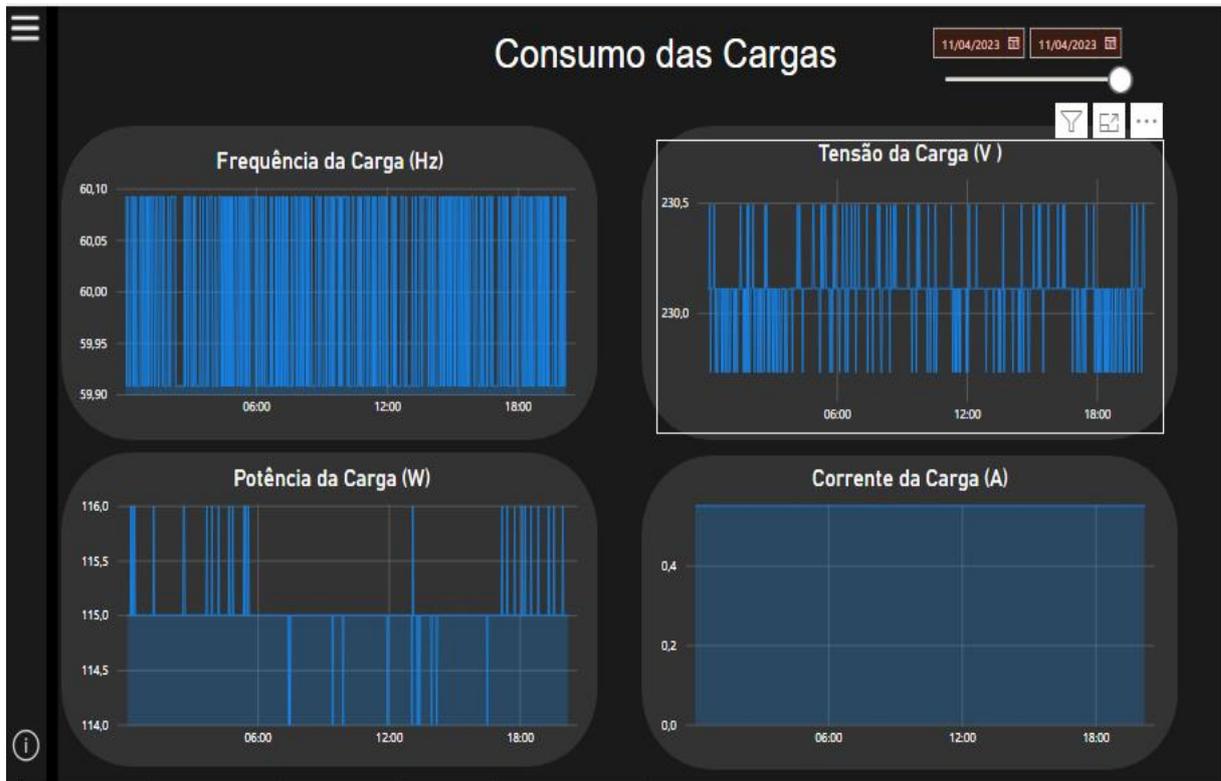
Fonte: Próprio Autor.

A Figura 28 apresenta o *Dashboard* da bateria, com parâmetros de potência, tensão e corrente, além do estado atual da carga da bateria, que varia de 0 a 100%. Cada parâmetro tem sua escala de tempo, enquanto o estado da bateria requer uma prevenção para visualizar seu estado atual. Os demais gráficos permitem a seleção do intervalo de tempo desejado por meio do segmento acima deles. Dessa forma, é possível sempre observar o estado atual da bateria e analisar o histórico dos outros parâmetros.

Figura 28 - *Dashboard* da bateria.

Fonte: Próprio Autor.

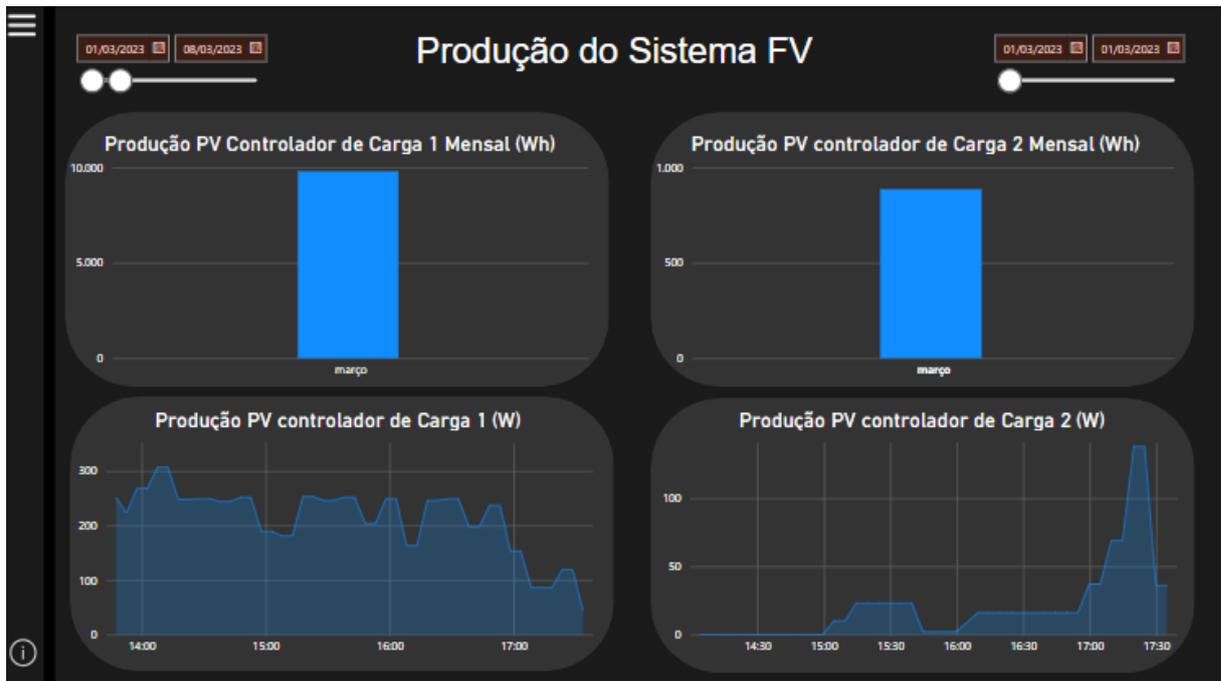
A Figura 29 apresenta um *Dashboard* com informações detalhadas sobre o consumo de carga, mostrando gráficos de frequência, corrente, tensão e potência da carga. Esses parâmetros elétricos são importantes para monitorar o consumo de energia e garantir o correto funcionamento do sistema elétrico da microrrede. Com tais informações, pode-se identificar possíveis falhas e ajustar o sistema para maximizar a eficiência energética. Além disso, esses dados são essenciais para a análise e otimização do consumo de energia, permitindo a identificação de padrões de uso e a redução de custos operacional.

Figura 29 - *Dashboard* de consumo das cargas.

Fonte: Próprio Autor.

A Figura 30 mostra um painel de controle da produção fotovoltaica dos controladores de carga 1 e 2. Ele é capaz de monitorar a eficiência de cada controlador individualmente, o que permite identificar possíveis problemas ou falhas no sistema e adotar medidas corretivas imediatas. Além disso, o painel exibe a produção de energia em Wh, permitindo acompanhar o total de energia gerada por cada controlador. Os gráficos mostram a produção de energia em W em relação ao tempo, possibilitando calcular o desempenho do sistema diariamente, semanalmente ou mensalmente. Esses dados podem ser usados para ajustar e melhorar o sistema.

Figura 30 - Painel de controle de produção fotovoltaica dos controladores de carga.



Fonte: Próprio Autor.

Após a finalização do processo de desenvolvimento dos *Dashboards*, é possível realizar a publicação da mesma na plataforma de nuvem da Microsoft, como apresentado na Figura 31. Isso implica na geração de um URL único, que pode ser acessado remotamente de qualquer lugar. Essa funcionalidade permite o compartilhamento da *Dashboard* com um público amplo, incluindo colaboradores, gestores, clientes e outros interessados, sem que haja a necessidade de compartilhar arquivos ou realizar a instalação de softwares específicos. Dessa forma, é possível garantir a disseminação de informações importantes de forma mais ágil e eficiente, contribuindo para a tomada de decisões mais informadas e para o sucesso do negócio.

Figura 31 - Página de publicação online da *Dashboard* no Power BI.

Fonte: Próprio Autor.

6 CONCLUSÕES E PROPOSTAS DE CONTINUIDADE

Com tudo que foi descrito e analisado conclui-se que o desenvolvimento de um modelo de banco de dados para gerenciamento energético remoto em microrredes é de extrema importância para a gestão eficiente e eficaz de energia elétrica, especialmente em um contexto em que as fontes de energia renováveis têm se tornado cada vez mais presentes.

Esse modelo permite o armazenamento e gerenciamento de grandes quantidades de informações com acessos simultâneos, além de fornecer propostas para aplicações futuras e contribuir para a adoção de tecnologias mais limpas, eficientes e confiáveis. Com esse modelo de banco de dados, é possível garantir a qualidade da energia fornecida, bem como tornar o processo de gerenciamento energético mais simples e prático.

Além disso, a criação desse banco de dados com armazenamento na nuvem permite que grandes quantidades de dados sejam armazenadas e recuperadas de forma segura e acessível, o que é fundamental para empresas e organizações que lidam com informações sensíveis e precisam compartilhá-las com seus colaboradores e clientes de forma rápida e eficiente.

Outra vantagem importante do banco de dados na nuvem é sua escalabilidade, permitindo que as empresas expandam seus bancos de dados à medida que crescem, sem a necessidade de investir em novos servidores e infraestrutura de TI. Isso torna a gestão de dados mais dinâmica e eficiente, facilitando o acesso e a manipulação de informações pelos usuários.

Como propostas de continuidade deste trabalho, cita-se:

- Integração de Algoritmos de Aprendizado de Máquina e Modelos Preditivos: Integração de algoritmos de aprendizado de máquina e modelos preditivos ao banco de dados desenvolvido. Essa expansão permitiria a criação de ferramentas capazes de prever padrões de demanda, produção e consumo de energia em microrredes. Por meio dessas previsões, seria possível otimizar a gestão energética, reduzindo desperdícios e aumentando a utilização de fontes de energia renováveis, promovendo assim uma abordagem mais sustentável e ecologicamente responsável;
- Criação de API (*Application Programming Interface*) para Acesso Programático: Possibilidade de acesso automatizado aos dados armazenados no banco;
- Desenvolvimento de Site para Visualização de Dados: capacidade de gerar gráficos e relatórios intuitivos, proporcionando uma visualização clara e compreensível das informações contidas no banco de dados, possibilitando a tomada de decisões mais assertivas e eficazes na gestão de energia em microrredes, contribuindo para o

desenvolvimento sustentável e para a transição energética para fontes limpas e renováveis.

REFERÊNCIAS

A. MANOWSKA, A. W. A. N. A. J. P. The Use of the MQTT Protocol in Measurement, Monitoring and Control Systems as Part of the Implementation of Energy Management Systems. **Electronics**, 12, Dezembro 2022. 17. Acesso em: Julho 2023.

ALFACOMP. MQTT – Conheça o protocolo padrão do IoT, 2020. Disponível em: <<https://alfacomp.net/2022/03/24/mqtt-conheca-o-protocolo-padrao-do-iot/>>. Acesso em: Julho 2023.

BEGG, Thomas C. E. C. **Banco de dados: Uma abordagem prática**.

BR MODELO WEB. BR Modelo Web. **BR Modelo Web**, 2023. Disponível em: <<https://docs.brmodeloweb.com/#/>>. Acesso em: Julho 2023.

CONNOLLY, T. M. . & B. C. E. **Database Systems: A Practical Approach to Design, Implementation, and Management**. 6ª. ed.

DATABASE, Microsoft. (. D.). A. S. Disponível em: <<https://azure.microsoft.com/pt-br/products/azure-sql/database/>>.

ECLIPSE FOUNDATION. Eclipse Paho. **Eclipse Foundation**, 2023. Disponível em: <<https://eclipse.dev/paho/index.php?page=clients/python/index.php>>. Acesso em: Março 2023.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Fundamentals os Database Systems**. 6ª. ed.

FERREIRA, Raisa T.; FARDIN, Jussara F.; RUEDA-MEDINA, Augusto C. **GERENCIAMENTO DE ENERGIA EM UMA MICRORREDE ISOLADA**. VII Congresso Brasileiro de Energia Solar. Gramado: Universidade Federal do Espírito Santo. 2018.

GUERRERO, Josep M. et al. Advanced Control Architectures for Intelligent Microgrids—Part I: Decentralized and Hierarchical Control. **IEEE Transactions on Industrial Electronics**, Abril 2013. 1254-1262. Acesso em: Abril 2023.

HIVEMQ. **MQTT & MQTT 5 Essential - A comprehensive overview of MQTT fact and features for beginners and experts alike**.

HOLISTICS. Draw Entity-Relationship Diagrams, Painlessly. **dbdiagram.io**, 2023. Disponível em: <<https://dbdiagram.io/home>>. Acesso em: Julho 2023.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. Brasil em síntese: Território e ambiente, 2021. Disponível em: <<https://cidades.ibge.gov.br/brasil/panorama>>. Acesso em: Março 2023.

ISMAIL, A. A.; HAMZA, H. S.; KOTB, A. M. **Performance Evaluation of Open Source IoT Platforms**. IEEE Global Conference on Internet of Things (GCIoT). Alexandria: [s.n.]. 2018. p. 1-5.

MARTINS, Maria C. D. G. N. **Implantação de Microrrede dotada de Sistema de Geração Solar Fotovoltaica e Sistema de Armazenamento de Energia**. Universidade Federal de Pernambuco. Recife, p. 77. 2022.

MCKINNEY, Wes. **Python para análise de dados: Tratamento de dados com Pandas, NumPy e IPython**.

MICROSOFT. Azure. **Azure**, 2023. Disponível em: <https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-azure/?&ef_id=_k_CjwKCAjwtuOIBhBREiwA7agf1gMDEWSMIQ-D5TCo6f6LtSw-f7Vig1aDL-cGXgzgzoXZENIcEIDI5hoC88AQAvD_BwE_k_&OCID=AIDcmmzmn0182_SEM__k_CjwKCAjwtuOIBhBREiwA7agf1gMDEWSMIQ-D5>. Acesso em: Julho 2023.

MICROSOFT. Azure Database for PostgreSQL, 2023. Disponível em: <<https://docs.microsoft.com/pt-br/azure/postgresql/quickstart-create-server-database-portal>>. Acesso em: Março 2023.

MICROSOFT. Microsoft Power BI. **Microsoft**, 2023. Disponível em: <https://powerbi.microsoft.com/pt-br/landing/free-account/?ef_id=_k_Cj0KCQjwzdOIBhCNARIsAPMwjbzT2S3CHPg_StBUvW-wCJ0a28cpu3cxU-iiXHjWzMSCbHRh5eV5Pi8aAiYtEALw_wcB_k_&OCID=AIDcmmk4cy2ahx_SEM__k_Cj0KCQjwzdOIBhCNARIsAPMwjbzT2S3CHPg_StBUvW-wCJ0a28cpu3cxU-iiXHjWz>. Acesso em: Julho 2023.

MICROSOFT. O que é o Power BI? **Power BI**, 2023. Disponível em: <https://powerbi.microsoft.com/pt-br/what-is-power-bi/?&ef_id=_k_CjwKCAjwтуOIBhBREiwA7agf1nWoJUqFLubDAhf_Q3BEHZ_1m2i9rqEvvzk5scCQ0d2658EPNRvYoRoCg5sQAvD_BwE_k_&OCID=AIDcmmk4cy2ahx_SEM__k_CjwKCAjwтуOIBhBREiwA7agf1nWoJUqFLubDAhf_Q3BEHZ_1m2i9rqEvvzk5scCQ0d265>. Acesso em: Julho 2023.

NETO, J. M. Banco de dados relacional vs. banco de dados não relacional: entenda as diferenças., 2017. Disponível em: <<https://blog.geekhunter.com.br/banco-de-dados-relacional-vs-banco-de-dados-nao-relacional/>>. Acesso em: Abril 2023.

OASIS OPEN. MQTT Version 5.0, 2023. Disponível em: <<https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html>>. Acesso em: Abril 2023.

OLIVEIRA, Mateus S. D. **Desenvolvimento de sistema de monitoramento e gerenciamento para microrredes residenciais dotadas de geração solar fotovoltaica e sistema de armazenamento de energia por bateria**. Universidade Federal de Pernambuco. Recife. 2023.

POSTGRESQL. PostgreSQL: The World's Most Advanced Open Source Relational Database. **PostgreSQL**, 2023. Disponível em: <<https://www.postgresql.org/>>. Acesso em: Julho 2023.

POSTGRESQL. Sobre o PostgreSQL, 2023. Disponível em: <<https://www.postgresql.org/about/>>. Acesso em: Julho 2023.

RIBEIRO, Mateus F. **Avaliação de Microrredes Isoladas com Geração de Energia**. Universidade Estadual Paulista. Ilha Solteira. 2023.

SILVA, Timóteo G. D. **Banco de Dados**.

SILVA, Valdemar W. S. E. F. S. C. D. **Bancos de Dados: Aprenda o que são, melhore seu conhecimento, construa os seus**.

VICTRON ENERGY. **Cerbo GX Manual**.

VICTRON ENERGY. Cerbo Gx. **Victron Energy**, 2023. Acesso em: Março 2023.