



**UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

Izidio Sousa de Carvalho

**SIS2GER E SIGAE2B FORMANDO UM ECOSISTEMA DE SOLUÇÕES
PARA GERENCIAMENTO DE INDICADORES ELÉTRICOS, AMBIENTAIS
E FINANCEIROS EM SISTEMAS DE GERAÇÃO FOTOVOLTAICA E DE
ARMAZENAMENTO DE ENERGIA**

DM: 17/2023

UFPA / ITEC / PPGEE
Campus Universitário Guamá
Belém – Pará - Brasil
2023



**UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

Izidio Sousa de Carvalho

**SIS2GER E SISGAE2B FORMANDO UM ECOSISTEMA DE SOLUÇÕES
PARA GERENCIAMENTO DE INDICADORES ELÉTRICOS, AMBIENTAIS
E FINANCEIROS EM SISTEMAS DE GERAÇÃO FOTOVOLTAICA E DE
ARMAZENAMENTO DE ENERGIA**

Dissertação apresentada no Programa de Pós-Graduação em Engenharia Elétrica do Instituto de Tecnologia como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica na área de Sistemas de Energia Elétrica.

Universidade Federal do Pará

Orientador: Prof. Dra. Maria Emília de Lima Tostes

UFPA / ITEC / PPGEE
Campus Universitário Guamá
Belém – Pará - Brasil
2023

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

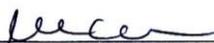
**“SIS2GER E SISGAE2B FORMANDO UM ECOSISTEMA DE SOLUÇÕES PARA
GERENCIAMENTO DE INDICADORES ELÉTRICOS, AMBIENTAIS E FINANCEIROS
EM SISTEMAS DE GERAÇÃO FOTOVOLTAICA E DE ARMAZENAMENTO DE
ENERGIA”**

AUTOR: **IZÍDIO SOUSA DE CARVALHO**

DISSERTAÇÃO DE MESTRADO SUBMETIDA À BANCA EXAMINADORA APROVADA PELO
COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA, SENDO
JULGADA ADEQUADA PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA
ELÉTRICA NA ÁREA DE SISTEMAS DE ENERGIA ELÉTRICA.

APROVADA EM: 02/06/2023

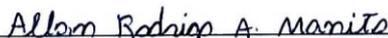
BANCA EXAMINADORA:



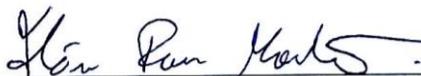
Prof.ª Dr.ª Maria Emília de Lima Tostes
(Orientadora – PPGEE/UFPA)



Prof. Dr. Ubiratan Holanda Bezerra
(Avaliador Interno – PPGEE/UFPA)



Prof. Dr. Allan Rodrigo Arrifano Manito
(Avaliador Externo ao Programa – FEED/UFPA)



Prof.ª Dr.ª Flávia Pessoa Monteiro
(Avaliadora Externa – UFOPA)

VISTO:

Prof. Dr. Diego Lisboa Cardoso

(Coordenador do PPGEE/ITEC/UFPA)

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBDSistema de Bibliotecas da Universidade Federal do Pará

Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)

C331s Carvalho, Izidio Sousa de.
SIS2GER E SISGAE2B FORMANDO UM ECOSSISTEMA DE SOLUÇÕES PARA GERENCIAMENTO DE INDICADORES ELÉTRICOS, AMBIENTAIS E FINANCEIROS EM SISTEMAS DE GERAÇÃO FOTOVOLTAICA E DE ARMAZENAMENTO DE ENERGIA /
Izidio Sousa de Carvalho. — 2023.
123 f. : il. color.

Orientador(a): Prof^a. Dra. Maria Emília de Lima Tostes
Dissertação (Mestrado) - Universidade Federal do Pará,
Instituto de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, Belém, 2023.

1. Ecosistema Computacional. 2. Monitoramento de Sistemas. 3. Sistemas de Geração de Energia Renováveis. 4. Sistema de Armazenamento de Energia.
I. Título.

CDD 621.310981

DEDICATÓRIA

Dedico este trabalho à minha família, minha esposa Cássia Paz, minha irmã Tatyane Cristina, meu pai Arnaldo Amaral e em especial à minha querida e amada mãe Genilde Gomes. Esta conquista é nossa.

AGRADECIMENTOS

A Deus, criador e salvador da minha vida, meu ponto de refúgio e fortaleza, a Ele toda a honra e glória.

A minha família que sempre esteve presente e apoiando durante a produção deste trabalho. Minha esposa e amiga Cassia Paz, por dividir essa jornada comigo e estar ao meu lado, minha mãe Genilde Gomes que sempre me incentivou aos estudos e a ser uma pessoa melhor. Amo vocês.

Ao CEAMAZON, que ao longo destes 7 anos foram parte fundamental da minha formação, não somente acadêmica, mas principalmente pessoal. Agradeço por estar rodeado por pessoas maravilhosas ao longo destes anos que englobam graduação e mestrado, me acolheram como parte dessa grande família. Em especial, a pessoa da Prof. Maria Emília que esteve comigo ao longo desta jornada e sempre com uma alegria contagiante, ao Prof. Ubiratan Bezerra que é uma fonte de inspiração, ao Thiago Mota que é um excelente professor pesquisador, à Flavia Monteiro como a minha primeira gerente de projetos e supervisor técnico, e aos demais amigos deste maravilhoso centro.

Aos queridos irmãos e amigos que fazem parte da minha jornada de vida, Ulisses Junior, Alex Duarte e Alan Bentes. Obrigado pelos conselhos, conversas e brincadeiras. Vocês tornam a vida mais leve.

A toda equipe do projeto SIMA que proporcionaram a implementação deste trabalho, em especial a Priscila Lobato e ao João Rodrigo pelas contribuições.

As instituições PPGE da UFPA e Norte Energia AS pelo apoio e fomento ao trabalho desenvolvido.

**“O trabalho duro vence o dom natural.”
Rock lee**

RESUMO

Atualmente existe um crescente aumento na demanda por energia elétrica em diversos países do mundo, esse aumento é impulsionado pela maior quantidade de equipamentos elétricos e eletrônicos conectados à rede elétrica em residências, bem como pela utilização de novos maquinários por parte da indústria. Dado o constante aumento na demanda de energia elétrica, também tem aumentado a geração de energia elétrica, principalmente, utilizando-se fontes renováveis. Acompanhando a tendência mundial, o Brasil vem passando por um processo de transição energética, utilizando fontes renováveis como solar e eólica, principalmente. E nesse contexto tem-se o presente trabalho, visando o desenvolvimento e implementação de softwares a formar um ecossistema computacional, contendo diversos tipos de sistemas conectados, sendo parte do Projeto de Pesquisa e Desenvolvimento (P&D) do Sistema Integrado de Mobilidade Elétrica Multimodal (SIMA), tendo como sua base o projeto piloto de uma cidade inteligente implementada na Cidade Universitária Professor José Silveira Netto da Universidade Federal do Pará (UFPA) para o estudo do impacto da implementação de uma infraestrutura desse porte, originado por meio da chamada nº 22/2018 da ANEEL e nomeado como cabeça de série dentro da Cadeia de Inovação da ANEEL. Com base nesse contexto, este trabalho propõem o desenvolvimento de dois softwares de gestão de sistemas de geração de energia renovável e sistemas de armazenamento de energia elétrica, uma vez que, por meio de um bom monitoramento e gerenciamento, é possível aumentar eficiência, segurança e qualidade da energia elétrica de um determinado sistema. Para desenvolvimento dos softwares foram utilizadas tecnologias que proporcionasse aos sistemas serem sistemas escaláveis e de fácil implementação, como o Docker, para a construção dos ambientes de desenvolvimento e produção; o framework Laravel, para a codificação dos softwares; e o Middleware DOJOT para o armazenamento dos dados de medição para cada sistema gerenciado.

Palavras-chave: Ecossistema Computacional, Monitoramento de Sistemas, Sistemas de Geração de Energia Renováveis, Sistema de Armazenamento de Energia.

ABSTRACT

Currently, there is a growing demand for electricity in several countries around the world, this increase is driven by the greater amount of electrical and electronic equipment connected to the electrical grid in homes, as well as the use of new machinery by industry. Given the constant increase in the demand for electrical energy, the generation of electrical energy has also increased, mainly using renewable sources. Following the world trend, Brazil has been going through an energy transition process, using renewable sources such as solar and wind, mainly. And in this context we have the present work, aiming at the development and implementation of software to form a computational ecosystem, containing several types of connected systems, being part of the Research and Development Project (R&D) of the Integrated System of Multimodal Electric Mobility (SIMA), based on the pilot project of a smart city implemented in Cidade Universitária Professor José Silveira Netto of the Federal University of Pará (UFPA) to study the impact of implementing an infrastructure of this size, originated through call nº 22/ 2018 by ANEEL and nominated as head of series within ANEEL's Innovation Chain. Based on this context, this work proposes the development of two management software for renewable energy generation systems and electrical energy storage systems, since, through good monitoring and management, it is possible to increase efficiency, safety and power quality of a given system. For the development of the software, technologies were used that provided the systems with scalable and easy-to-implement systems, such as Docker, for building the development and production environments; the Laravel framework, for coding the software; and the DOJOT Middleware for storing measurement data for each managed system.

Keywords: Computational Ecosystem, Monitoring Systems, Renewable Energy Generation Systems, Energy Storage System.

LISTA DE FIGURAS

Figura 1 - Sistema Integrado de Mobilidade Elétrica Multimodal (SIMA), projeto n 22/2018 da ANEEL.	13
Figura 2 - Modelo da arquitetura MVC.	15
Figura 3 - Serviços da DOJOT.	16
Figura 4 - Comparação entre a virtualização em máquinas virtuais e virtualização em containers.	17
Figura 5 - Visualização de métricas utilizando o Grafana.	19
Figura 6 - Oferta Interna de Energia por fonte.	22
Figura 7 - Setores que mais demandaram por energia no ano de 2021.	22
Figura 8 - Aumento do consumo de energia elétrica por setor.	23
Figura 9 - Casas atendidas pelo projeto Pisco de Luz.	25
Figura 10 - Impacto do uso de banco de baterias no pico de demanda.	27
Figura 11 - Exemplificação do GitFlow.	29
Figura 12 – XCP-ng (esquerda) e Proxmox (direita) configurados no Centro de Ciências Exatas e Tecnologia (CCET) da Universidade Federal do Estado do Rio de Janeiro.	30
Figura 13 – Cluster Proxmox.	30
Figura 14 - Funcionamento do Docker Daemon e sua infraestrutura.	32
Figura 15 - Exemplificação de uma imagem em Docker.	32
Figura 16 - Exemplificação de um trecho de arquivo Dockerfile.	34
Figura 17 - Exemplo de Registry com imagens Docker de versoes (tags) diferentes hospedadas no Registry do Gitlab.	35
Figura 18 – Utilização de uma imagem Docker e a execução de scripts em um Dockerfile para subir um container com o serviço configurado.	36
Figura 19 - Comparação entre aplicações em um servidor e em containers em Docker.	37
Figura 20 - Formas de armazenamento de dados em Docker.	38
Figura 21 – Docker Networks, rede bridge.	39
Figura 22 - Comparação na utilização de CPU.	39
Figura 23 - Comparação da utilização de memória RAM.	40
Figura 24 – Diagrama de funcionamento do servidor para o CI/CD implementado para o Departamento de Informática e Estatística da Universidade Federal de Santa Catarina.	41
Figura 25 - Diagrama de funcionamento dos tetes implementados Sistema Integrado de Gestão de Unidades de Alimentação e Nutrição.	42
Figura 26 - diagrama do software Softway4IoT.	43
Figura 27 - Arquitetura interna do middleware DOJOT.	44
Figura 28 - Componentes do Next-GISSA.	45
Figura 29 - Camada MVC do Laravel.	46
Figura 30 - Árvores de características de qualidade de softwares.	47
Figura 31 - Páginas principais da Área do cliente (esquerda) e área do Gestor (direita) das quadras do Guaranix.	51
Figura 32 - Tela de visualização do atendente.	52
Figura 33 - Tela de dashboard do Sistema de Gerenciamento do Viveiro de Mudas do IFMG-SJE.	53
Figura 34 – Histórico de coleta de dados do Sistema de inspeção predial para engenheiros e arquitetos.	54
Figura 35 - Servidor Power Edge T440 configurado no CEAMAZON.	56

Figura 36 – Arquitetura das aplicações desenvolvidas.	57
Figura 37 - Arquivo docker-compose.yml do banco de dados.....	58
Figura 38 - Arquivo docker-compose.yml do SIS2GER.....	59
Figura 39 - Arquivo de configuração do NGEX do do SIS2GER.....	60
Figura 40 - Arquivo Dockerfile.....	61
Figura 41 - Imagens do container do SIS2GER.	62
Figura 42 - Fluxo de comunicação com a DOJOT.	64
Figura 43 - Fluxo de comunicação com a DOJOT com cache.	64
Figura 44 - Modelo Entidade Relacionamento dos softwares desenvolvidos.	65
Figura 45 - MER com tabela usuario em destaque.	66
Figura 46 – MER com as tabelas alerta_fotovoltaico e alerta_armazenamento	67
Figura 47 - MER do sistema de notificação no banco de dados.	68
Figura 48 - Aquisição de dados.	69
Figura 49 - Integralização de dados a serem exibidos na tela de Home de forma síncrona.	70
Figura 50 - Integralização de dados a serem exibidos na tela de Home de forma assíncrona.	71
Figura 51 - Grafana, tela de monitoramento do host do ambiente.....	72
Figura 52 – Grafana, sistemas monitorados no servidor.	73
Figura 53 - Tela de Login.	75
Figura 54- Tela de Cadastro de Usuários.....	76
Figura 55 - E-mail enviado para confirmar ao novo usuário.	76
Figura 56 - Tela de Perfil de Usuário.	78
Figura 57 - Alerta de exclusão de dados do usuário na plataforma.....	78
Figura 58 - Tela de Home do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.	80
Figura 59 - Menu Lateral Esquerdo do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.	81
Figura 60 - Menu de aplicações do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER	82
Figura 61 - Tela de grandezas monitoradas do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.....	83
Figura 62 - Opção de seleção grandezas no SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.....	84
Figura 63 - Tela de Indicadores Elétricos do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.....	85
Figura 64 – Opção de seleção dos indicadores elétricos do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.	85
Figura 65 - Tela de Análise Financeira do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.....	86
Figura 66 - Tela de gerenciamento de dispositivos do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.....	87
Figura 67 - Tela de edição de dispositivos do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.....	88
Figura 68 - Tela de Monitoramento do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.	89

Figura 69 - Tela de Alertas do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.	90
Figura 70 - Tela de Logs do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.	91
Figura 71 - Tela de Configurações do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.	92
Figura 72 - Tela de Usuários do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.	93
Figura 73- Tela de Home SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.	94
Figura 74- Tela de Detalhes de Home do SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.	95
Figura 75 - Menu de ferramentas do SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.	96
Figura 76 - Menu de aplicações do SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.	97
Figura 77 - Tela de Grandezas do SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.	98
Figura 78 - Tela de Grandezas disponíveis no SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.	99
Figura 79 - Tela de indicadores do SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.	100
Figura 80 - Tela de Indicadores do SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.	100
Figura 81 - Tela de Análise Financeira do SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.	101
Figura 82 - Tela de Dispositivos com a opção de visualizar e criar dispositivos no SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.	102
Figura 83- Tela de Dispositivos na opção de editar um dispositivo SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.	103
Figura 84 - Tela de Alertas e Notificações do SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.	104
Figura 85 - Tela de Logs do SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.	105
Figura 86- Tela de Logs exibindo logs por um usuário no SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.	105
Figura 87 - Tela de Configurações do SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.	106
Figura 88 - Tela de usuários do SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.	107

LISTA DE TABELAS

Tabela 1 - Tabela de classificação de sistemas fotovoltaicos conforme NBR 11.704:2008.	24
Tabela 2 – Geração de energia elétrica em SFVCR rede.	26
Tabela 3 - Vulnerabilidades detectadas no cenário controlado.	50
Tabela 4 - Servidor de Implementação.	55

SUMÁRIO

Sumário

CAPÍTULO 1: INTRODUÇÃO	12
1.1 Considerações Gerais	12
1.1 Objetivo.....	19
1.1.1 Objetivo Geral.....	19
1.1.2 Objetivo Específico.....	20
1.2 Organização do Trabalho	20
CAPÍTULO 2: ESTADO DA ARTE	21
2 Introdução.....	21
2.1 Matriz Energética Brasileira	21
2.1.1 Sistemas Fotovoltaicos	23
2.1.2 Sistemas de Armazenamento	26
2.2 Tecnologias de Desenvolvimento.....	27
2.2.1 Virtualização	28
2.2.2 DOJOT.....	43
2.2.3 Framework Laravel.....	45
2.3 Conclusão.....	54
CAPÍTULO 3: METODOLOGIA	55
3 Introdução.....	55
3.1 Servidor de Implementação do Software	55
3.2 Desenvolvimento.....	56
3.2.1 Docker	56
3.3 Persistência de dados	62
3.3.1 DOJOT.....	63
3.3.2 Banco de Dados Relacional	65
3.4 Implementação de Processamento Assíncrono.....	69
3.4.1 Observabilidade	71
3.5 Conclusão.....	73
CAPÍTULO 4: IMPLEMENTAÇÃO DO SOFTWARE	74

4.1	Introdução.....	74
4.2	Telas dos sistemas desenvolvidos.....	74
4.2.1	Pontos comuns para os softwares.....	74
4.2.2	Sistema de Gestão da Geração de Energia Renovável – SIS2GER.....	79
4.2.3	Sistema de Gestão do Sistema de Armazenamento de Energia Elétrica – SISGAE2B.....	93
4.3	Conclusão.....	107
	CAPÍTULO 5: CONCLUSÃO.....	109
5.1.	Considerações finais.....	109
5.2.	Trabalhos Futuros.....	111
5.3.	Produções.....	112
•	Registro de Softwares.....	112
	REFERÊNCIAS BIBLIOGRÁFICAS.....	113

CAPÍTULO 1: INTRODUÇÃO

1 Introdução

1.1 Considerações Gerais

É possível constatar o aumento no consumo de energia elétrica em grande parte dos países ao redor do globo terrestre, dentre outros motivos, isto vem ocorrendo devido à alguns fatores como: as mudanças nos hábitos de consumo da população, o aumento populacional, o desenvolvimento de novas tecnologias e equipamentos eletrônicos, a melhoria de processos e o surgimento de novos serviços (DIAS, 2020) (COUTO et al., 2022).

Um dos grandes desafios tangentes do aumento do consumo se dá pela busca por utilização de fontes de geração renovável (SOARES; KATARINA, 2022), uma vez que a utilização de uma matriz composta por energias renováveis é um importante fator para a redução de custos de produção de energia, emissão de gases do efeito estufa, e dentro desta abordagem tem-se que a crescente adoção por energias renováveis pode se tornar a principal fonte de energia do mundo até 2040 (IEA, 2020).

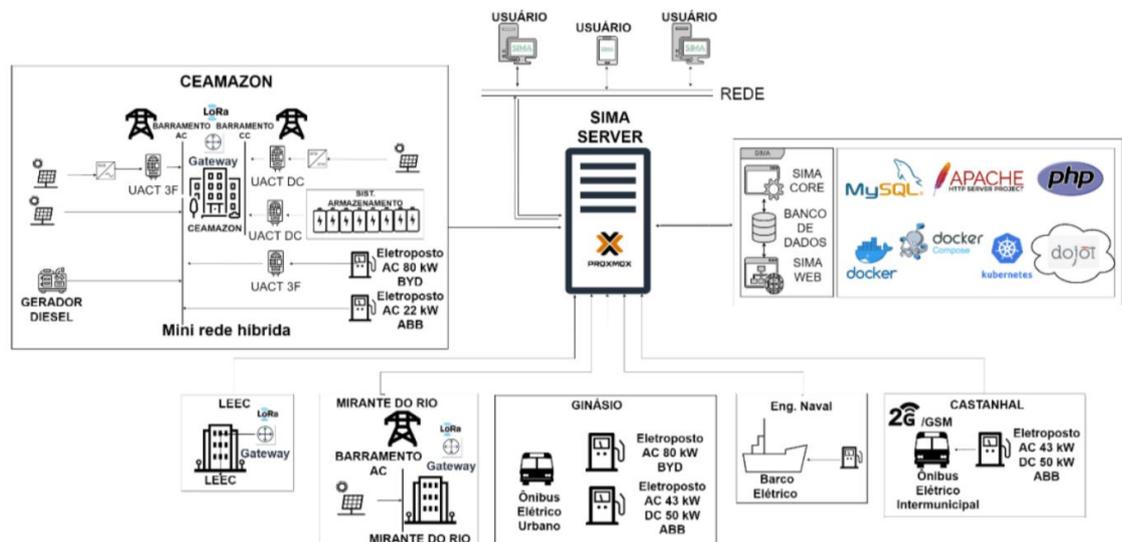
Logo, com a inserção de novas fontes de geração de energia elétrica tem-se a crescente necessidade do acompanhamento de aspectos de qualidade de energia elétrica, a redução do desperdício de energia elétrica, a melhor da utilização de recursos energéticos, bem como o acompanhamento de indicadores ambientais para sistemas de geração de energias por fontes renováveis, visando o acompanhamento do retorno ambiental obtido pela utilização de sistemas de geração de energia renováveis, alguns dos principais indicadores são o número de árvores cultivadas e o carbono evitado (Lucinda, 2015)(ALMEIDA, 2021).

Dentro desse contexto, este trabalho surge através do desenvolvido do Projeto de Pesquisa e Desenvolvimento (P&D) do Sistema Integrado de Mobilidade Elétrica Multimodal (SIMA). Tendo como sua base a implementação de diversas características de uma cidade inteligente por meio de um sistema multimodal servindo de laboratório a céu aberto dentro do campus da Cidade Universitária Professor José Silveira Netto da Universidade Federal do Pará (UFPA), contemplando assim as características da região de desenvolvimento (Região Amazônica) para o estudo do impacto da implementação de uma infraestrutura desse porte. O projeto foi originado

por meio de uma chamada no tema de mobilidade elétrica nº 22/2018 da ANEEL e nomeado como cabeça de série dentro da Cadeia de Inovação da ANEEL.

Entre os objetivos da chamada, se dá a implementação e monitoramento de dois sistemas de geração de energia fotovoltaica e um sistema de armazenamento de energia elétrica dispostos dentro do campus da UFPA. Logo, para tem-se a necessidade de desenvolver softwares capazes monitorar os sistemas propostos e realizar a correta implementação e gestão de uma estrutura complexa e com diversos tipos de sistemas conectados, a Figura 1 exibe a composição do projeto, o qual possui ônibus elétricos, barcos elétricos, sistemas de armazenamento de energia, sistemas de geração de energia fotovoltaica e eletropostos.

Figura 1 - Sistema Integrado de Mobilidade Elétrica Multimodal (SIMA), projeto n 22/2018 da ANEEL.



Fonte: (LOBATO, 2022)

Dentro do escopo deste trabalho, tem-se o desenvolvimento dos softwares de gestão de sistemas de geração de energia renováveis e os sistemas de armazenamentos que constam na chamada mencionada, e tem por objetivo prover um ecossistema de ferramentas para que seja possível do gestor realizar os estudos inerente a área de eficiência energética e qualidade de energia elétrica.

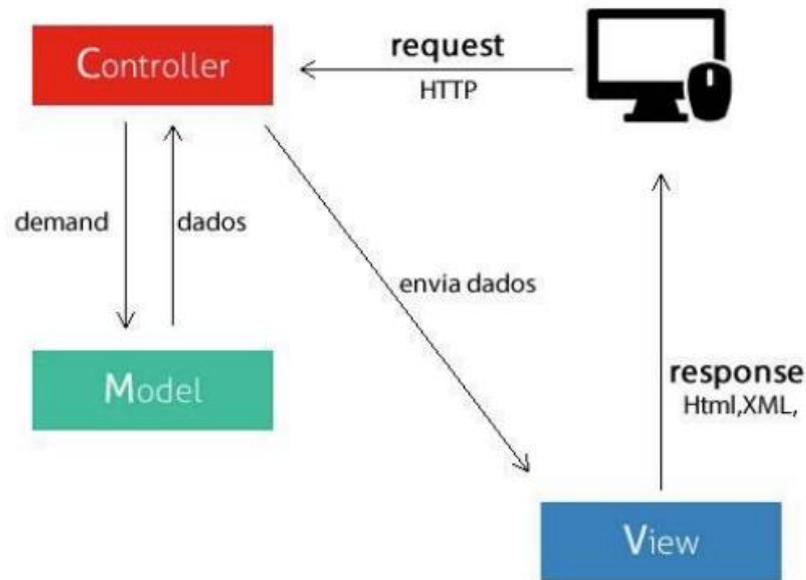
Dado ao exposto, tem-se o desenvolvimento e utilização de softwares como ferramenta capazes de auxiliar em tais fatores, dado a importância do monitoramento de sistemas de geração de energias renováveis em cada local de instalação da geração de forma distribuída e possibilitando a análises de desempenho (AGLIARDI; CHEPP; GASPARIN, 2021).

Nos pontos inerentes ao desenvolvimento e implementação, é uma boa prática utilizar frameworks para garantir a padronização a escalabilidade no desenvolvimento de sistemas (CHAVES; SILVA, 2008). Um framework possui como base os principais conceitos da Programação Orientada a Objetos (POO), sendo estes: a abstração de dados, o polimorfismo e a herança. Estes conceitos permitem a criação de *frameworks* com o objetivo da reutilização do código em diferentes partes do *software* por meio da criação de classes genéricas. Além destes, é importante destacar pontos como a facilidade de manutenção, a capacidade de extensão e a modularidade (CHAVES; SILVA, 2008).

O Laravel é um framework desenvolvido na linguagem de programação PHP com o foco no desenvolvimento web. Dentro do conceito de *design pattern*, este framework trabalha com o padrão MVC (*Model, View, Controller*), desenvolvido para facilitar a implementação de sistemas online de maneira a garantir aspectos como a segurança, alto desempenho, agilidade e com código limpo (PELIZZA; BERTOLINI; SILVEIRA, 2018).

O MVC utilizado pelo Laravel é um padrão de design de sistema que visa aumentar a modularidade do sistema para que as classes modeladas no *software* possam desempenhar papéis específicos em cada camada da aplicação (SILVA, 2018). O fluxo da informação pode ser visto na Figura 2.

Figura 2 - Modelo da arquitetura MVC.



Fonte: (SILVA, 2018).

Um aspecto inerente ao desenvolvimento de softwares escaláveis e de alto desempenho é a capacidade de lidar com um volume cada vez maior de dados. Entre as principais necessidades no desenvolvimento de aplicações, destacam-se a capacidade de suportar um alto tráfego de informações de usuários simultaneamente, utilizando um serviço e a capacidade de responder às interações dos usuários rapidamente, sendo uma das formas mitigar pontos tangente à essa problemática se dá pelo uso de banco de dados ideal para cada necessidade de software (COSTA et al., 2021).

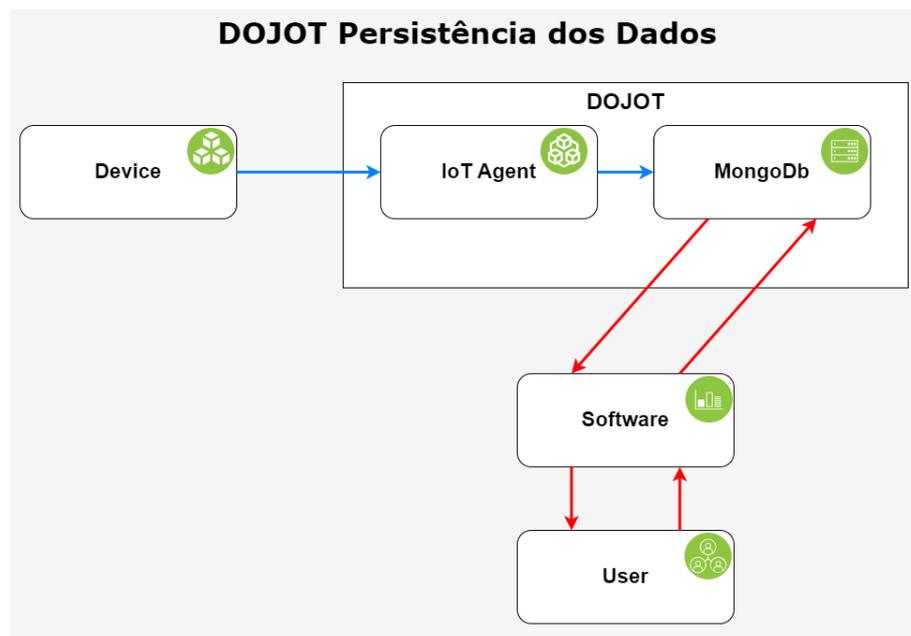
Um banco de dados ou Sistema Gerenciador de Banco de Dados (SGBD) é definido como uma coleção de dados em um armazenamento persistente capaz de manter um nível de organização dos dados armazenados e a possibilidade de obtenção de dados por meio de consultas. Como forma de garantir a integridade dos dados, é importante garantir as seguintes propriedades: Atomicidade, Consistência, Isolamento, Durabilidade (SCHUERZOSKI, 2019).

No que tange ao gerenciamento e persistência de um grande volume de dados, é possível utilizar outras ferramentas para atuarem como *middleware* entre aplicações que são responsáveis por salvar um grande volume de dados, tais como dispositivos de IoT (*Internet of Things* – Internet das Coisas) Essa abordagem permite que a comunicação entre os dados salvos do dispositivo e a utilização por parte do sistema possa ocorrer em uma camada de alto nível (GOMES, 2019).

Nesse cenário tem-se a DOJOT para servir de middleware devido a sua atuação entre os dispositivos de uma rede IoT e os softwares, que são responsáveis por implementar regras de negócio da aplicação, atuando como um *middleware* para sistemas que necessitam da leitura de um grande volume de dados (MORAES et al., 2021).

A DOJOT é um middleware capaz de armazenar os dados de telemetria, como as grandezas medidas e os indicadores calculados para diversos dispositivos IoT em um *template* configurado e utiliza o MongoDB, um banco de dados no paradigma NoSQL. Na Figura 3 é possível observar o MongoDB, os módulos de *IoT Agent* para conexão com os dispositivos *IoT* (DOJOT, 2023) e como os softwares desenvolvidos podem acessar as informações persistidas com os dados da solicitação feita pelo usuário.

Figura 3 - Serviços da DOJOT.

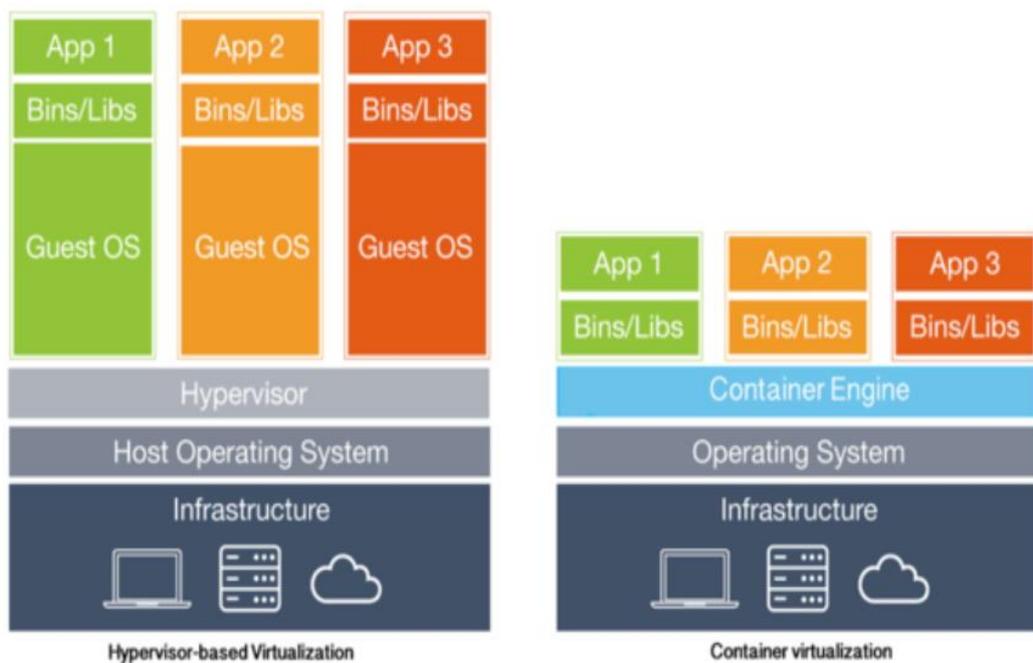


Fonte: O Autor.

Em relação à implementação de *softwares* em servidores de aplicações, estas podem ser instaladas de diferentes formas, porém a utilização de contêineres se destaca por ser uma tecnologia utilizada para o empacotamento de código, bibliotecas e dependências em um único objeto. Essas características permitem a capacidade de isolamento das aplicações, otimização, flexibilidade, poder de processamento, aumento da demanda e eficiência energética (SANTOS; ENDO; SILVA, 2019).

Os aspectos relacionados a utilização de contêineres em detrimento a outras tecnologias, como a virtualização em máquinas virtuais, se dão pela capacidade de utilizar apenas um sistema operacional para todas as aplicações sendo executadas, sem a necessidade de criar um sistema operacional para cada aplicação como acontece com a virtualização em máquinas virtuais por meio do Hypervisor, essa diferença torna a aplicação mais portátil e segura, a Figura 4 representa a comparação entre o uso de containers e máquinas virtuais (GARCIA; PEREIRA, 2019).

Figura 4 - Comparação entre a virtualização em máquinas virtuais e virtualização em containers.



Fonte: (GARCIA; PEREIRA, 2019).

Dada a instalação de um software em um servidor ou ambiente final, tem-se a importância de monitoramento e acompanhamento desse software. Este processo é conhecido como observabilidade, que tange ao aspecto de acompanhamento e gerenciamento dos diversos componentes de um software. O processo é crucial para garantir a estabilidade, o desempenho e a disponibilidade do software. O que permite realizar análises tanto relacionadas ao comportamento atual quanto realizar inferências a respeito do comportamento futuro do sistema (NUNES, 2019).

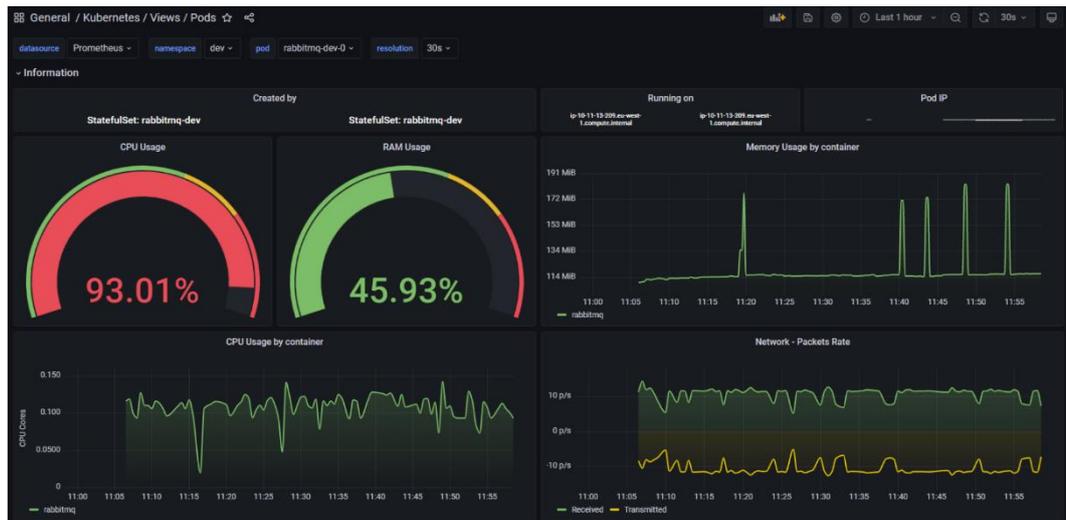
A observabilidade engloba diferentes níveis da implementação de uma aplicação, sendo capaz de acompanhar elementos de infraestrutura como rede;

processamento computacional; armazenamento dos dados; e o software em si, podendo ser usada para distribuir os recursos computacionais e otimizar custos (NUNES, 2019). A observabilidade pode ser compreendida em 4 funcionalidades: monitoramento, visualização/alerta, logs e rastreamento.

- Monitoramento: Faz menção a coleta de métricas de forma periódica, de diversos dados de telemetria da aplicação, tais como histogramas de latências, uso de recursos como a Memória ou CPU ao longo de um período ou a quantidade de ações processadas com sucesso ou erro pela aplicação.
- Visualização e Alertas: Com base nos dados coletados relacionados ao monitoramento, é possível implementar dashboards com os dados de telemetria. Os alertas fazem menção a capacidade de emitir algum comunicado, seja visual, sonoro, via e-mail etc., de quando algum componente do sistema apresenta anomalias.
- Logs: Os logs são as informações a respeito dos eventos que ocorreram no sistema, contendo o momento exato que ocorreu e as informações referentes ao evento.
- Rastreamento: Por meio de rastreamento, é possível entender a relação entre os eventos que ocorreram no sistema, como ele se propaga na aplicação e seus efeitos, tal como uma rotina mal programada que consome muitos recursos como memória ou CPU e pode fazer com que a aplicação não funcione corretamente.

O ambiente Docker permite a utilização de diversas imagens de ferramentas *open source* que estão presentes no repositório público do Docker, o *Docker Hub*. Estas imagens são capazes de auxiliar na implementação de rotinas de observabilidade, tais como o Grafana, Prometheus, cAdvisor (SILVA, 2019). A Figura 5 exemplifica a visualização de dashboard por meio do grafana em que é possível acompanhar métricas como uso de CPU e Memória RAM de uma aplicação (CARVALHO, 2022).

Figura 5 - Visualização de métricas utilizando o Grafana.



Fonte: (CARVALHO, 2022).

Dado o exposto até então, no cerne deste trabalho tem-se o desenvolvimento de softwares resilientes e capazes de comportar um alto volume de dados que advém da necessidade de monitoramento cada vez mais constantes do sistema elétrico de potência, podendo ser utilizado para monitorar a implementação de geração distribuída, para a tomada de ações de manutenção do sistema, eficiência energética e de qualidade de energia elétrica.

Esse trabalho faz parte do projeto de pesquisa e desenvolvimento do Sistema Integrado de Mobilidade Elétrica Multimodal (SIMA), implementado na Cidade Universitária Professor José Silveira Netto da Universidade Federal do Pará (UFPA), através de chamada no tema de mobilidade elétrica nº 22/2018 da ANEEL, o qual conta com o apoio da Norte Energia S.A.

1.1 Objetivo

1.1.1 Objetivo Geral

Uma vez que ocorre a inserção de novos equipamentos no sistema elétrico brasileiro em conjunto com a demanda por monitoramento, tem-se o objetivo de desenvolvimento de dois *softwares*, tanto para a realização da gestão da geração de energia renovável quanto para o gerenciamento do armazenamento de energia elétrica. Ambos os softwares devem ser capazes de realizar o acompanhamento de grandezas elétricas, indicadores elétricos e indicadores ambientais, para se obter

sistemas com características referentes tanto ao acompanhamento quanto ao gerenciamento dos sistemas. Sendo assim, o cerne deste trabalho se dá pelo desenvolvimento e implementação de softwares de alto desempenho nos servidores da UFPA para acompanhamento e gerenciamento de sistemas, servido como ferramentas para gestão de energia, análise de qualidade de energia elétrica, acompanhamento de grandezas elétricas, eficiência energética, indicadores ambientais e econômicos.

1.1.2 Objetivo Específico

Tendo como objetivo específico:

- a) Desenvolver o Sistema de Gestão da Geração de Energia Renovável (SIS2GER) para ser capaz de gerenciar múltiplos sistemas fotovoltaicos.
- b) Desenvolver o Sistema de Gestão do Sistema de Armazenamento de Energia Elétrica (SISGAE2B) para ser capaz de gerenciar múltiplos sistemas de baterias.

1.2 Organização do Trabalho

Este trabalho é dividido em capítulos que são:

O Capítulo 1 sendo o capítulo introdutório.

O Capítulo 2 é responsável por apresentar o estado da arte, contendo as referências utilizadas como base no desenvolvimento do trabalho.

O Capítulo 3 tem por objetivo explicar sobre a metodologia utilizada para a implementação e desenvolvimento do trabalho.

O Capítulo 4 fica a cargo de exibir os resultados alcançados com o desenvolvimento do sistema.

O Capítulo 5 é o responsável por concluir a apresentação do trabalho, destacando os seus principais pontos e mencionar possíveis linhas de pesquisa para continuidade e melhoria do trabalho exposto.

CAPÍTULO 2: ESTADO DA ARTE

2 Introdução

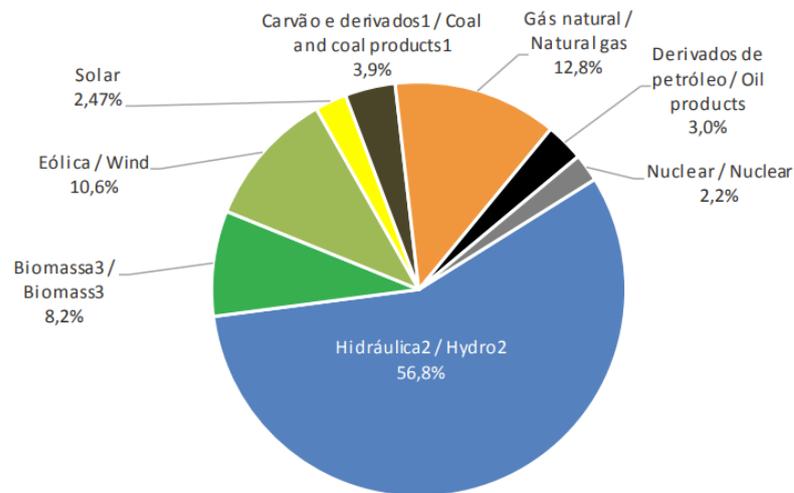
Este capítulo tem como objetivo apresentar estudos desenvolvidos e fundamentos utilizados no decorrer desta dissertação, que utilizou como base a geração fotovoltaica, sistemas de armazenamento de energia, frameworks de desenvolvimento e tecnologias necessárias para a implementação dos softwares de gestão de geração e armazenamento de energia.

2.1 Matriz Energética Brasileira

Segundo (PEREIRA, 2021) a matriz energética representa o somatório de recursos ou fontes empregados para atender as demandas por energia, o que inclui a eletricidade, o aquecimento, o transporte e quaisquer outras atividades que necessitam de energia para serem executadas.

De acordo com (BEN, 2022) o Sistema Elétrico Brasileiro (SEP) possui uma matriz elétrica constituída majoritariamente por fontes renováveis, sendo a fonte hídrica a energia mais utilizada e representando 56,8% da Oferta Interna por Energia (OIE), considerando as importações de energia. O somatório de todas as fontes renováveis corresponde a 78,07% da oferta interna de eletricidade no Brasil, sendo que os sistemas fotovoltaicos representam 2,47% da matriz energética brasileira, como pode ser visto na Figura 6.

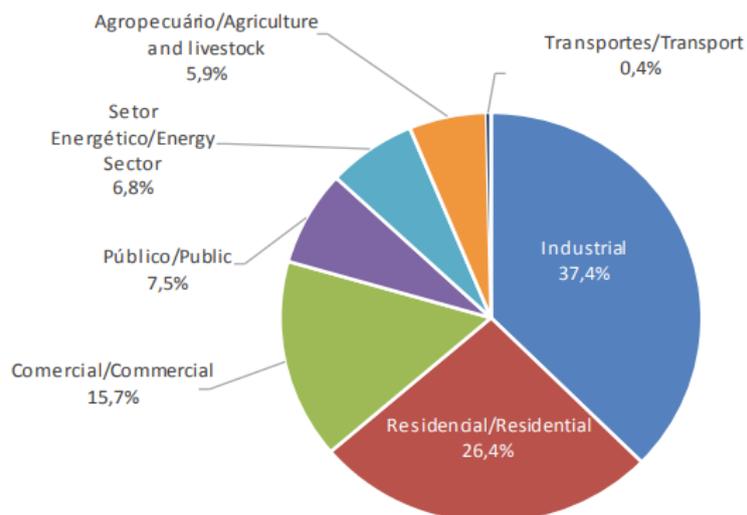
Figura 6 - Oferta Interna de Energia por fonte.



Fonte: (BEN, 2022)..

Segundo (BARDELIN, 2004) existe uma forte dependência brasileira de fontes de energia hidráulicas e apesar de ser uma fonte barata, ela está sujeita às condições climáticas das chuvas, de modo que em períodos de secas e estiagens se faz necessário utilizar outras fontes de energia. A Figura 7 exhibe os setores que mais demandaram por energia no ano de 2021.

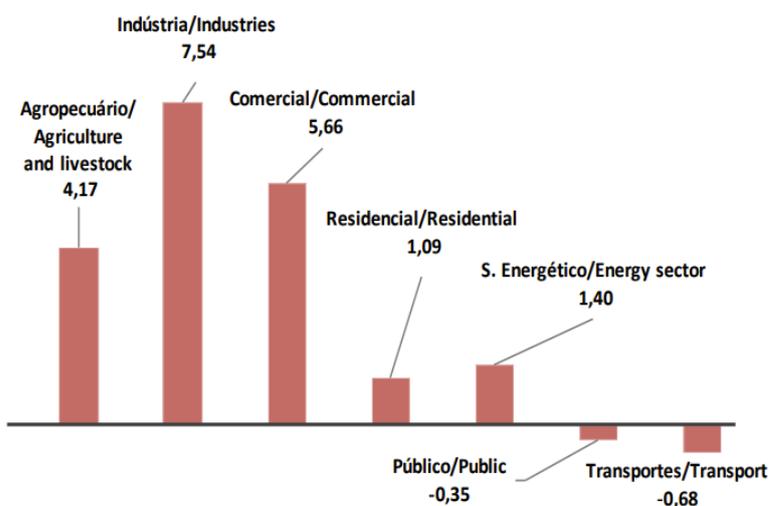
Figura 7 - Setores que mais demandaram por energia no ano de 2021.



Fonte: (BEN, 2022).

Segundo o balanço energético nacional 2022 feito pela Empresa de Pesquisa Energética (EPE), o Brasil teve um aumento do consumo final na ordem de 5,7% em comparação com 2021, atingindo um total de 570,8 TWh (BEN, 2022). Esse aumento se deve principalmente pelo aumento do consumo nos setores: Agropecuário (4,17%), Indústrias (7,54%), Comercial (5,66%), conforme pode ser visto na Figura 8.

Figura 8 - Aumento do consumo de energia elétrica por setor.



Fonte: (BEN, 2022).

De acordo com (COSTA; GARREFA, 2022) existe uma crescente expectativa do aumento por demanda de energia elétrica no Brasil, sendo necessário cada vez mais a implantação de sistemas fotovoltaicos em clientes residenciais, industriais e comerciais.

2.1.1 Sistemas Fotovoltaicos

Segundo (ALMEIDA, 2021) é possível classificar as instalações geradoras por duas características: Quanto ao tipo de interligação e quanto ao tipo de configuração. No tocante aos tipos de ligação, estes podem ser Sistemas isolados ou Sistemas conectados à rede elétrica, e em relação à configuração, estes podem ser sistemas puros ou sistemas híbridos, de acordo com a Norma Brasileira NBR 11704:2008, como pode ser visto na Tabela 1.

Tabela 1 - Tabela de classificação de sistemas fotovoltaicos conforme NBR 11.704:2008.

Classificação	Quanto à interligação	Sistemas Isolados	São aqueles que não possuem qualquer conexão com o sistema público de fornecimento de energia elétrica.
		Sistemas conectados à rede elétrica	São aqueles que efetivamente conectados ao sistema público de energia elétrica.
	Quanto à configuração	Sistemas puros	São aqueles que utilizam o gerador fotovoltaico como única fonte de energia elétrica.
		Sistemas Híbridos	São aqueles que resultam na associação do gerador fotovoltaico com outros tipos de geradores de energia elétrica.

Fonte: (ALMEIDA, 2021)

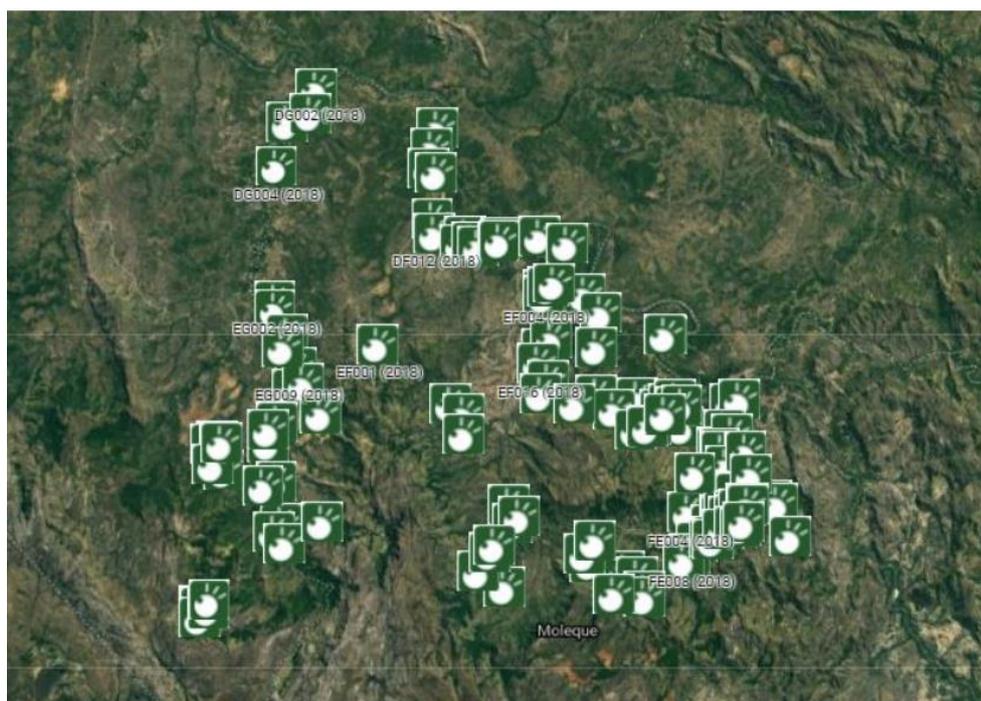
A) Sistemas fotovoltaicos isolados

Segundo (ALVES, 2019) os sistemas fotovoltaicos isolados são, geralmente, utilizados em locais de baixa viabilidade para a instalação da rede elétrica, por um conjunto de fatores econômicos, ambientais ou de acesso.

De acordo com (BRASIL, 2018) é comum nesse tipo de sistema a existência de sistema de armazenamento para garantir a continuidade do fornecimento energético em períodos que não é possível realizar a produção de energia por meio de geradores fotovoltaicos. Nesse modo, o gerador fotovoltaico fornece energia para a carga e para recarregar o sistema de armazenamento de energia, para que o sistema de armazenamento possa abastecer a carga quando não houver geração solar.

No trabalho desenvolvido por (MONTEIRO, 2020) foram analisados o uso de sistemas fotovoltaicos isolados em comunidades quilombolas remotas no cerrado, com o objetivo de analisar as viabilidades técnica, econômica e social da tecnologia de geração de energia solar isolada, proposta pelo projeto intitulado Pisco de Luz. A Figura 9 representa a geolocalização dos locais atendidos pelo projeto.

Figura 9 - Casas atendidas pelo projeto Pisco de Luz.



Fonte: (MONTEIRO, 2020).

B) Sistemas fotovoltaicos conectados à rede

Segundo (JUNIOR; DAS NEVES, 2022) os sistemas fotovoltaicos conectados à rede elétrica possuem como característica a ligação com a rede de distribuição das concessionárias de energia. De acordo a LEI Nº 14.300/2022 os sistemas fotovoltaicos conectados à rede podem injetar o excedente de energia gerado por meio do medidor bidirecional, o qual contabiliza a energia injetada na rede da concessionária elétrica para ser reutilizada em períodos que não tiver geração ou a demanda for maior do que a geração, por exemplo.

Dentro nesse contexto, o trabalho feito por (JUNIOR; DAS NEVES, 2022) teve como o objetivo a análise de 7 sistemas fotovoltaicos conectados à rede elétrica, instalados em residências na cidade Curitiba, conforme descrito na Tabela 2.

Tabela 2 – Geração de energia elétrica em SFVCR rede.

SFVCR	Potência (kWp)	Geração de Energia Elétrica (kWh/ano)
Sistema 1	4,02	5.020
Sistema 2	4,2	5.350
Sistema 3	4,2	4.850
Sistema 4	4,26	4.550
Sistema 5	7,2	7.640
Sistema 6	5,11	5.820
Sistema 7	4,1	4.630

Fonte: (JUNIOR; DAS NEVES, 2022)

2.1.2 Sistemas de Armazenamento

Segundo (BUENO; BRANDÃO, 2016), os Sistemas de Armazenamento de Energia (SAE) pode ser acoplado a sistemas de geração de energia elétrica renováveis e possuem a capacidade de agregar à energia elétrica uma maior disponibilidade, qualidade e confiabilidade. Para (COSTA; BORTONI, 2016) os SAE possuem um papel importante na unificação, distribuição e ampliação da capacidade dos sistemas de geração distribuída, com o uso das energias solar, eólica e outras renováveis.

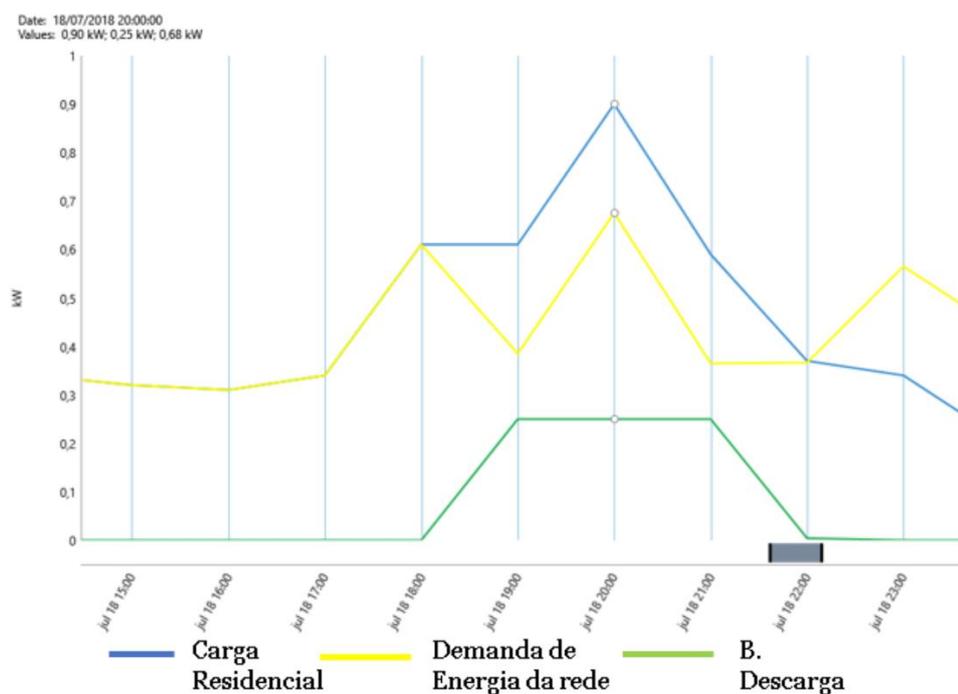
De acordo com (COSTA, 2021), durante o processo de armazenamento de energia elétrica, a energia armazenada é comumente convertida para outro tipo de energia, para que somente no momento da utilização ela possa ser convertida novamente em energia elétrica, a saber:

- Energia mecânica: Pode ser armazenada por bombagem hidroelétrica, ar comprimido ou volante de inércia, conhecido também como *Flywheels*.
- Eletroquímica: Pode ser armazenada por Baterias estacionárias (Íons de lítio, Ácido-chumbo, Ni-cd) ou baterias de fluxo.

- Elétrica: Pode ser armazenada por meio de Bobinas supercondutoras.
- Química: Pode ser armazenada por meio de pilhas combustíveis, chamadas de *Fuel cell*.
- Térmica: Pode ser armazenada por calor latente ou sensível.

Como uma de suas funcionalidades no uso de sistemas de armazenamento em conjunto com sistemas fotovoltaicos, (FARIAS, 2018) fez uma análise da utilização de sistemas de baterias para a diminuição do pico de demanda. Em um de seus resultados foi possível obter uma redução de 24% da demanda no período de ponta, como é exibido na Figura 10.

Figura 10 - Impacto do uso de banco de baterias no pico de demanda.



Fonte: (FARIAS, 2018).

2.2 Tecnologias de Desenvolvimento

Dentro deste tópico tem-se os referenciais teóricos a respeito do desenvolvimento de software nos temas de desenvolvimento de ambientes virtuais para implementação e difusão de software, utilização de middlewares para gerenciamento de dispositivos IoT, persistência em banco de dados, utilização de

frameworks para a codificação do software e a observabilidade para garantir o correto funcionamento.

2.2.1 Virtualização

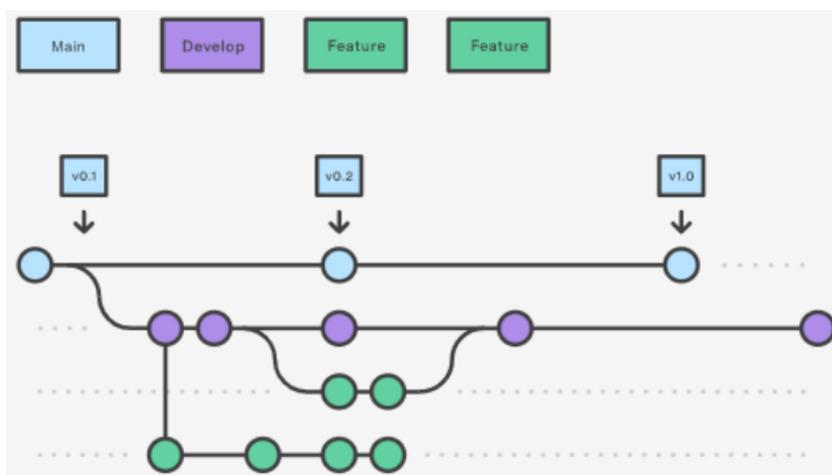
Segundo (BATISTA, 2022) o início do conceito de virtualização se deu a partir da criação da tecnologia de *time-sharing*, o que abriu portas para a possibilidade de utilização de um mesmo computador, de forma remota, por diversas pessoas via utilização de terminais. Assim, surgiu então a necessidade de proteger os recursos da máquina contra aplicações suscetíveis a falhas, tendo origem na década de 1960 com os *mainframes* da IBM. Dada as primeiras versões da computação em nuvem, o termo virtualização começou a ser mais utilizado.

Em (LIMA, 2021) o conceito de virtualização se dá ao uso indireto de recursos computacionais de hardware, como as unidades de processamento de dados (CPU), unidades de armazenamento (HD/SSD), unidades de entradas e saída de dados (I/O), entre outros. A utilização de ambientes virtuais vem a depender da necessidade da implementação, podendo ser utilizado para a abstração de hardware ou de sistemas operacionais.

Para (PAIVA, 2019), devido à facilidade de alocação de recursos computacionais em ambientes virtuais, é possível obter ganhos de escalabilidade e flexibilidade por meio do aumento dos recursos virtuais que foram previamente configurados (memória, processamento, armazenamento, rede) e, com isso, o melhor dimensionamento da carga de trabalho de cada aplicação, de modo que é possível montar uma infraestrutura resiliente para os diferentes tipos de ambientes, sejam eles ambientes de produção, de desenvolvimento ou de testes.

O conceito de isolamento de ambientes descrito por (PAIVA, 2019) pode ser visto por meio da Figura 11, a qual representa o *gitflow* que para (SILVA, 2022) é um processo de desenvolvimento baseado em *git* para o versionamento de código, no qual é possível observar algumas *branches*: *Main*, *Develop*, *Feature*. Cada *branch* é descrita como uma versão diferente do código principal do *software*, porém, podendo ser isoladas em diferentes ambientes para desenvolvimento, testes, produção, dentre outros.

Figura 11 - Exemplificação do GitFlow.



Fonte: (SILVA, 2022).

Para (BATISTA, 2022), dadas as necessidades expostas, surgiram os conceitos de virtualização e de máquinas virtuais. O processo de virtualização possui diversos benefícios, tais como: a segurança, o isolamento de desempenho, a facilidade de gerenciamento e a flexibilidade de execução em um ambiente personalizado.

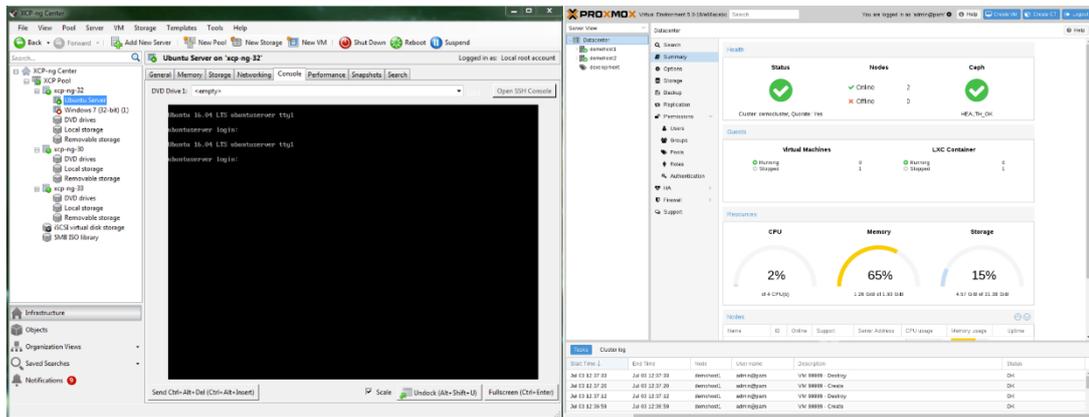
A) Máquinas Virtuais

Segundo (BATISTA, 2022) as máquinas virtuais são os sistemas hospedeiros do sistema principal. As máquinas hospedeiras possuem uma camada de software com a capacidade de realizar virtualização a nível de *hardware*, para que possa ser capaz de hospedar as aplicações nessa nova arquitetura. Para (JACINTO, 2022) os sistemas que estão em execução em uma máquina virtual desconhecem que estão sendo virtualizados, essa abordagem necessita de alto gasto de recursos computacionais, pois o sistema hospedeiro precisa servir não somente os seus próprios recursos operacionais, como também os recursos de cada máquina virtual residente, cada uma com o seu próprio sistema operacional.

Em (MORAIS, 2019) foi implementado uma virtualização por meio de máquinas virtuais utilizando tanto o software *XCP-ng* quanto o *Proxmox*, para uma análise comparativa entre as ferramentas no ambiente de datacenter do Centro de Ciências Exatas e Tecnologia (CCET) da Universidade Federal do Estado do Rio de Janeiro –

UNIRIO (Figura 12). O *Proxmox* foi o software mais indicado para a problemática pela facilidade de configuração das máquinas virtuais.

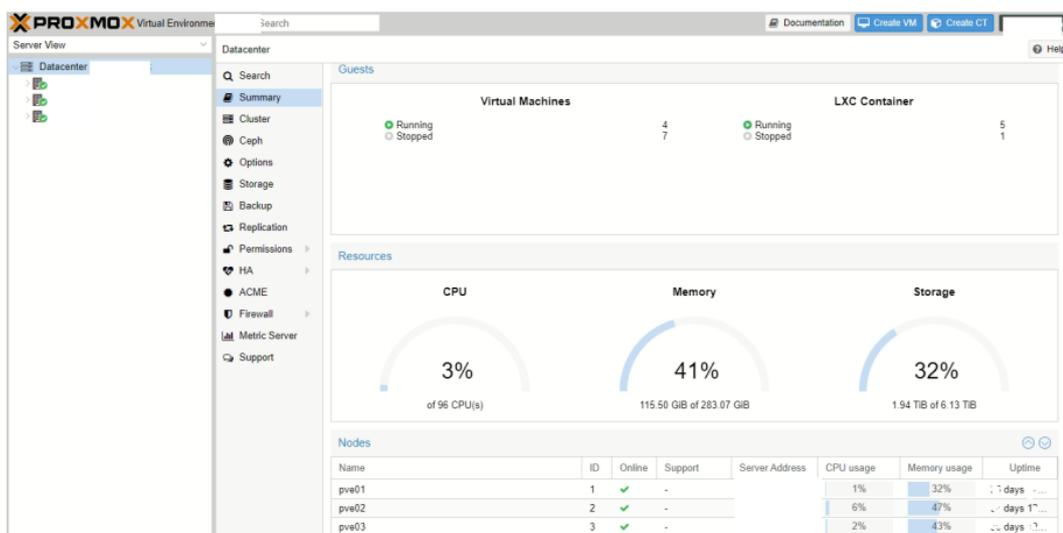
Figura 12 – XCP-ng (esquerda) e Proxmox (direita) configurados no Centro de Ciências Exatas e Tecnologia (CCET) da Universidade Federal do Estado do Rio de Janeiro.



Fonte: (MORAIS, 2019).

No trabalho feito por (CLOSS, 2021), a virtualização foi utilizada para a criação de máquinas virtuais (VM) para o Cartório de Mato Grosso, para atendimento a requisitos do provimento 74/2018 do Conselho Nacional de Justiça (CNJ), e o gerenciamento das VMs foi realizado pelo software *Proxmox*, como pode ser visto na Figura 13.

Figura 13 – Cluster Proxmox.



Fonte: (CLOSS, 2021).

B) Containerização

De acordo com (SILVA, 2017) a utilização de containers é conhecida como virtualização a nível de sistema operacional. A solução baseada em containers permite criar uma camada de virtualização mais leve quando comparada com máquinas virtuais, pois o kernel da máquina host permite que sejam executadas múltiplas instâncias isoladas compartilhando o mesmo kernel, mas garantindo que estas instancias estão isoladas umas das outras.

Segundo (SILVA, 2017) o Docker, foi lançado no dia 25 de março de 2013, é uma ferramenta utilizada para virtualização baseada em containers e desde então, é a plataforma de código aberto mais utilizada para containerização. E, para (BATISTA, 2022), o Docker é mais leve e oferece ganhos em portabilidade, agilidade, controle e isolamento. A tecnologia Docker é composta por diversos componentes para o seu funcionamento:

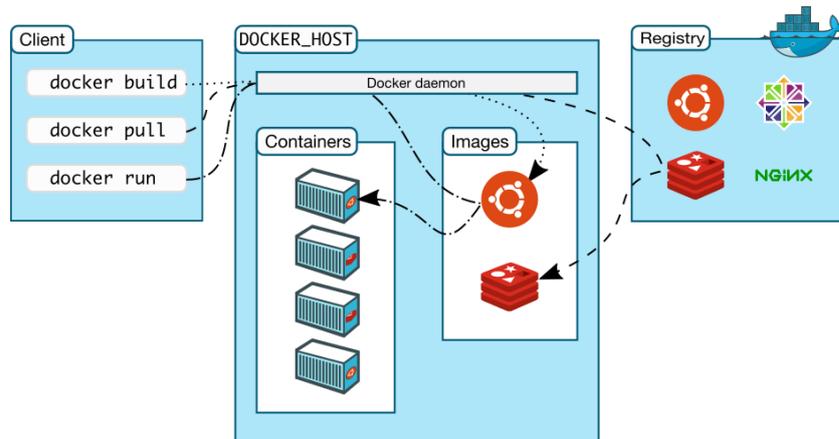
I. *Docker Engine*

Para (BATISTA, 2022), o *Docker Engine* é uma ferramenta cliente-servidor baseada na virtualização de containers, sendo utilizada para o empacotamento de forma leve e portátil, além de fornecer ferramentas para que os usuários possam interagir com os contêineres. Os serviços como o do *Docker daemon* ficam em execução na máquina hospedeira, criando uma camada para que seja possível construir e executar os containers, independente do sistema operacional desta máquina.

Para (BOSCO, 2019) o *Docker Engine* é responsável por ser um servidor em execução com os serviços necessários, para isso possui uma API (*Application Programming Interface*) responsável por processar as solicitações do *Docker Client*, como gerenciar imagens, containers, networks e volumes.

A Figura 14 representa o funcionamento do Docker Daemon e sua infraestrutura, na qual é capaz de receber comandos, gerenciar as imagens na máquina local e acessar o registry, para levantar os containers com as imagens necessárias.

Figura 14 - Funcionamento do Docker Daemon e sua infraestrutura.

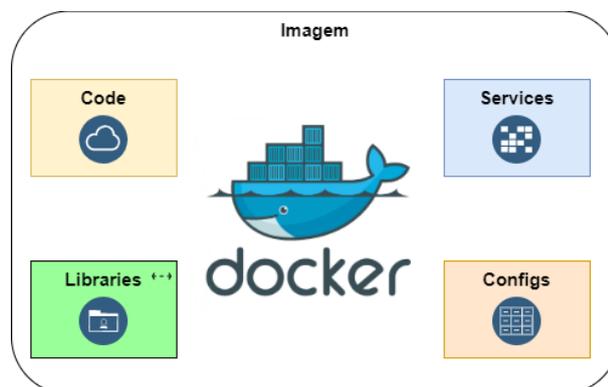


Fonte: (GARCIA; PEREIRA, 2019).

II. Docker Images

Para (MONTEIRO; ALMEIDA, 2017), uma imagem Docker é construída com a inserção de uma ou diversas camadas de dados em cima de uma camada utilizada como base. Logo, cada imagem pode ser vista como um *template*, base com as instruções adequadas para executar uma instância de um contêiner, de modo que cada nova imagem é criada a partir de uma outra imagem base. Cada uma dessas imagens é autocontida e autossuficiente para o seu propósito, com a capacidade de executar os requisitos de uma aplicação. A imagem 15 representa uma imagem Docker com o código, as bibliotecas, os serviços e as configurações.

Figura 15 - Exemplificação de uma imagem em Docker



Fonte: O Autor.

De acordo com (BATISTA, 2022), todas as imagens e as diversas camadas de containers possuem seu conteúdo endereçado por SHA256. Essa característica de endereçamento tem como objetivo principal, a melhoria da segurança, evitando colisões com os nomes dos containers, além de manter integridade dos dados.

III. Dockerfiles

Segundo (MAIA, 2020), o Dockerfile é um arquivo de texto que possui uma sequência de comandos para a criação de imagens Docker, além de possibilitar o incremento serviços com passos adicionais para criar uma imagem com mais funcionalidades.

Para (BATISTA, 2022) os arquivos *Dockerfiles* são scripts utilizados para a criação de uma nova imagem Docker passando por um conjunto de instruções e a passagem de parâmetros bem definidos. O seu uso se dá pela possibilidade de automatizar todo o processo da criação de novas imagens para que seja possível criar versões de um sistema, partindo da definição de uma imagem base e as instruções a ser executada no novo ambiente. A Figura 16 representa um arquivo de exemplo.

Figura 16 - Exemplificação de um trecho de arquivo Dockerfile.

```

FROM php:7.4-fpm

# Install system dependencies
RUN apt-get update && apt-get install -y \
    git \
    curl \
    libpng-dev \
    libonig-dev \
    libxml2-dev \
    zip \
    unzip

# nodejs
RUN apt-get update -yq \
    && apt-get install curl gnupg -yq \
    && curl -sL https://deb.nodesource.com/setup_12.x | bash \
    && apt-get install nodejs -yq

#Prevent: The SCRAM_SHA_256 authentication mechanism requires libmongoc built with ENABLE_SSL
RUN apt-get install -y libcurl4-openssl-dev pkg-config libssl-dev

# Install PHP extensions
RUN docker-php-ext-install pdo_mysql mbstring exif pcntl bcmath gd

# Install PHP extensions
RUN docker-php-ext-install pdo_mysql mbstring exif pcntl bcmath gd

# Get latest Composer
COPY --from=composer:latest /usr/bin/composer /usr/bin/composer
# Izidio Carvalho, 3 years ago * Configuration for Docker Compose

# Configure path
WORKDIR /var/www
COPY . /var/www

```

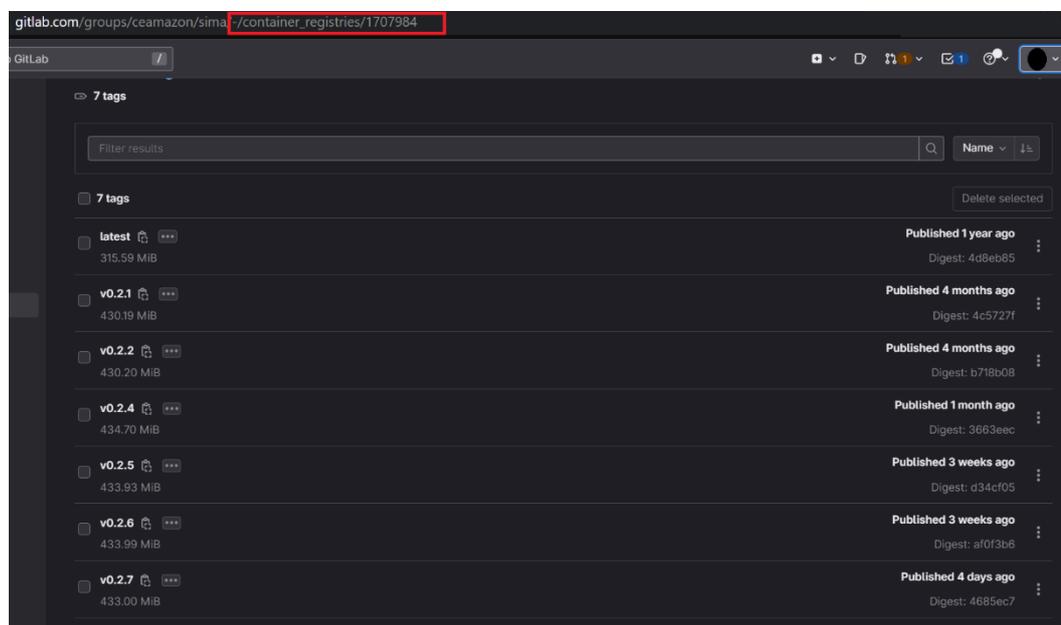
Fonte: O Autor.

IV. Docker registry

Para (BOSCO, 2019) os sistemas de registros de *Dockers Images* são serviços que atuam em rede conhecidos como *Docker Registry*, tendo como função fazer com que a construção de imagens vista no tópico anterior só ocorra uma vez, sendo possível usar a imagem construída e salva no *Registry*. É possível utilizar repositórios públicos como *Docker Hub* ou repositórios privados para manter a confidencialidade dos *softwares*.

Segundo (BOSCO, 2019) uma vez que as imagens são construídas por intermédio do *Docker daemon*, essas imagens somente estão construídas na máquina que executou o comando de *build* e ao invés de executar o mesmo comando em todas as máquinas de servidor que forem executar a imagem criada, é possível enviar essa imagem para um *Docker Registry* para que todos as máquinas possam utilizar esta imagem, não sendo necessário construir novamente. A Figura 17 representa um exemplo do registro de containers utilizado no *Gitlab*.

Figura 17 - Exemplo de Registry com imagens Docker de versoes (tags) diferentes hospedadas no Registry do Gitlab.

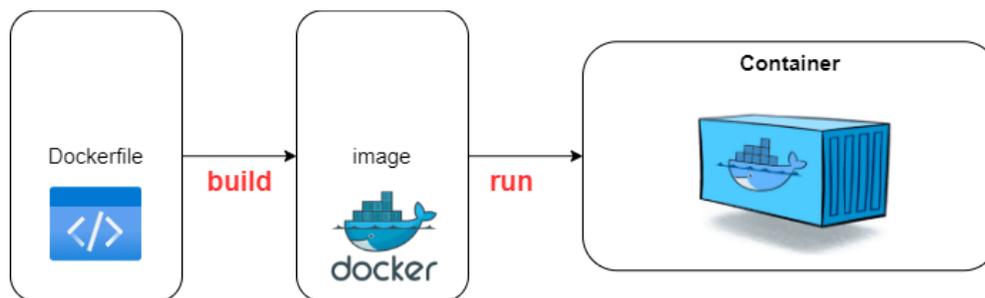


Fonte: O Autor.

V. Docker Containers

De acordo com (BOSCO, 2019), o container é uma instancia que está em execução com base em uma imagem. Para (BATISTA, 2022), a mudança de abordagem das máquinas virtuais para a utilização de contêineres em *Docker*, faz com que os contêineres sejam isolados sem a necessidade de emulação de um hardware na máquina hospedeira. Logo, o container *Docker* executa no *Docker Daemon*, passando as instruções necessárias para o sistema operacional da máquina hospedeira e eliminando a necessidade de um sistema os sistemas visitantes. A Figura 18 permite visualizar a utilização de uma imagem Docker e a execução de scripts em um *Dockerfile*, gerando um container com o serviço configurado.

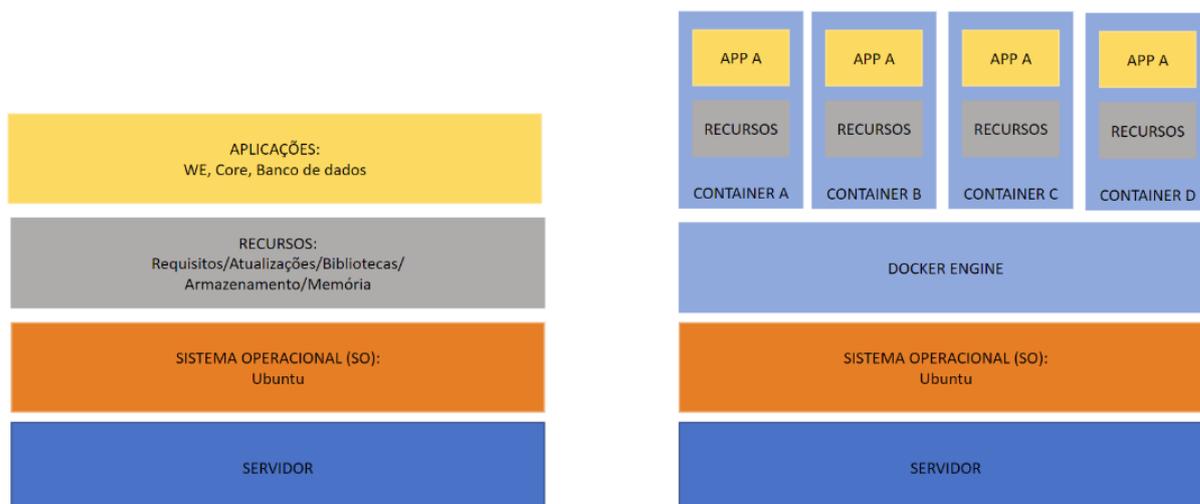
Figura 18 – Utilização de uma imagem Docker e a execução de scripts em um Dockerfile para subir um container com o serviço configurado.



Fonte: O Autor.

Para (CLEMENTE; SILVA, 2022), a utilização de containers é amplamente utilizada na migração de sistemas legados para micro serviços. Os sistemas legados também são chamados de monolitos quando todos os elementos para o seu funcionamento estão rodando no mesmo ambiente, sendo compostos, geralmente, por três elementos: backend, frontend e banco de dados. Esses sistemas possuem vantagens e desvantagens, tendo como uma de suas vantagens, a maior facilidade no desenvolvimento da aplicação; e como desvantagem, os softwares tendem a aumentar de complexidade durante o processo de desenvolvimento, o que torna a manutenção do código um processo complexo. A Figura 19, é exibida a comparação entre um sistema em que todas as aplicações estão em execução no sistema operacional do servidor e um conjunto de aplicações sendo executados em containers no mesmo servidor.

Figura 19 - Comparação entre aplicações em um servidor e em containers em Docker.



Fonte: O Autor.

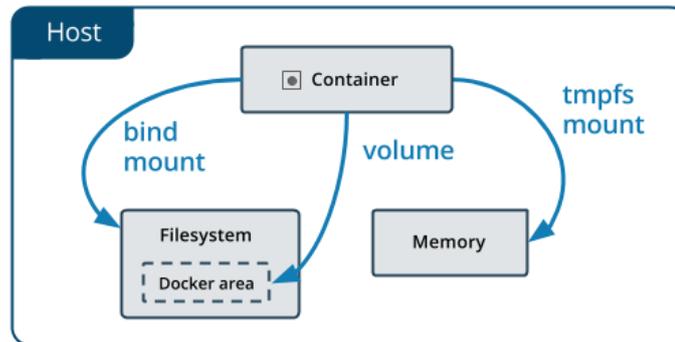
VI. Docker Volumes

Segundo (BOSCO, 2019), o volume do Docker é um espaço de armazenamento para salvar o conteúdo produzido ou recebido por um container, sejam estes arquivos de código, arquivos de configuração ou arquivos criados durante a execução do container. Estes arquivos são armazenados em uma camada reservada para escrita de dados, de modo que eles fiquem persistidos, porém, também existe a memória que não persiste informações, chamada de tmpfs (*temporary file system*).

De acordo com (ROCHA, 2022), o Docker possui duas formas de armazenar informações de forma persistida, o que significa que ao reiniciar um container as informações salvas podem ser recuperadas, tais como logs de auditoria e informações salvas no banco de dados, como: *volumes* e *bind mounts*.

A principal diferença entre as formas de persistência de dados, é que os volumes são montados e gerenciados pelo próprio Docker na máquina host em um endereço específico de memória (*var/lib/docker/volumes/* no linux), enquanto *bind mounts* podem ser gerenciados por outros usuários ou softwares existentes na máquina host e podem estar contidos em qualquer endereço existente no armazenamento. Na Figura 20, tem-se representa as 3 formas de armazenamento de dados mencionadas.

Figura 20 - Formas de armazenamento de dados em Docker.



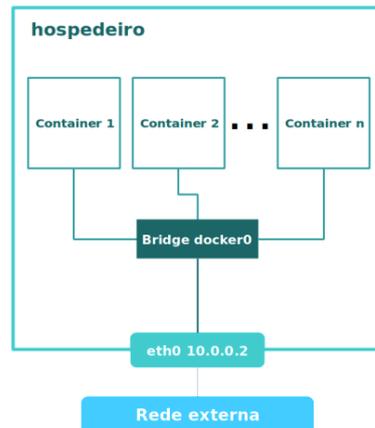
Fonte: (ROCHA, 2022).

VII. Docker Network

Para (ROCHA, 2022), uma das vantagens de se utilizar serviços em Docker é que estes podem conversar entre si em uma rede interna, uma vez que os serviços presentes no host podem se conectar de forma lógica.

Segundo (BARBOSA et al., 2019), trabalhar com redes em Docker faz com que a comunicação entre os containers aconteça de forma mais segura, pois o isolamento de rede entre os sistemas ocorre por meio do uso de *Namespaces* de rede, os quais atribuem endereços IP, tabelas de roteamento e dispositivos de rede únicos a cada container. A comunicação entre os containers ocorre por meio das suas respectivas interfaces de rede, sendo a comunicação com o hospedeiro por meio da placa de rede virtual *Virtual Ethernet bridge* criada pelo Docker. E, de acordo com (FERREIRA, 2018), essa rede no modo *bridge*, criada no hospedeiro é chamada de `docker0`, recebe um endereço IP privado e uma sub rede. A Figura 21 demonstra uma rede Docker.

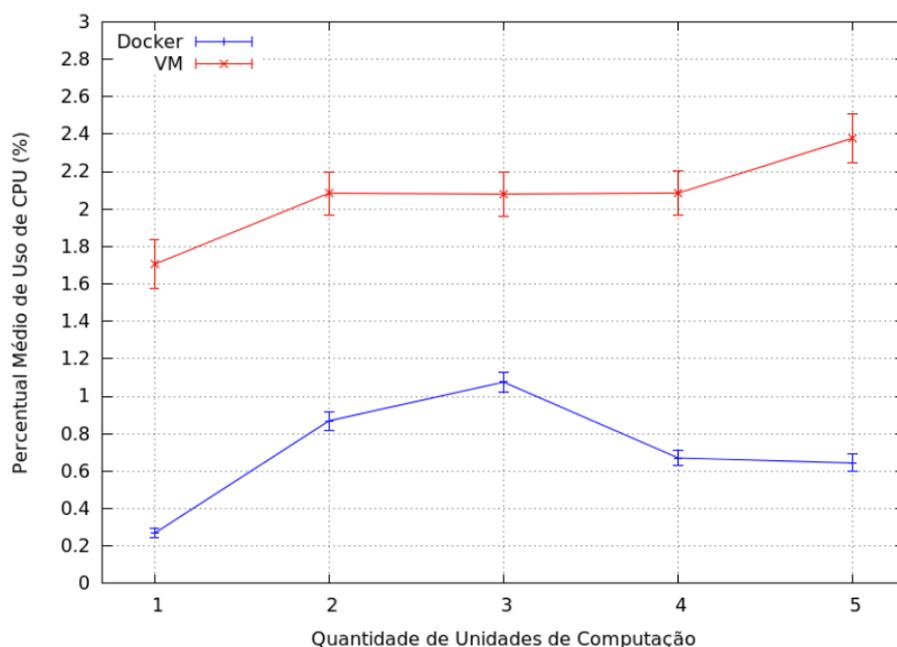
Figura 21 – Docker Networks, rede bridge.



Fonte: (FERREIRA, 2018).

No quesito da utilização dos recursos computacionais, como o CPU e memória RAM, é possível ver que a utilização de Containers possui o melhor uso dos recursos da máquina hospedeira em comparação ao ambiente virtualizado com hipervisor (VM), no trabalho feito por (SILVA, 2017) a medida em que novos serviços são inseridos o consumo de memória e CPU da máquina hospedeira é maior na utilização de VM por conta da necessidade de emulação do hardware. Foi utilizado o conceito de Unidades de computação para se referir ao número de máquinas virtuais ou containers utilizados, na Figura 22 pode ser visto a porcentagem do uso de CPU.

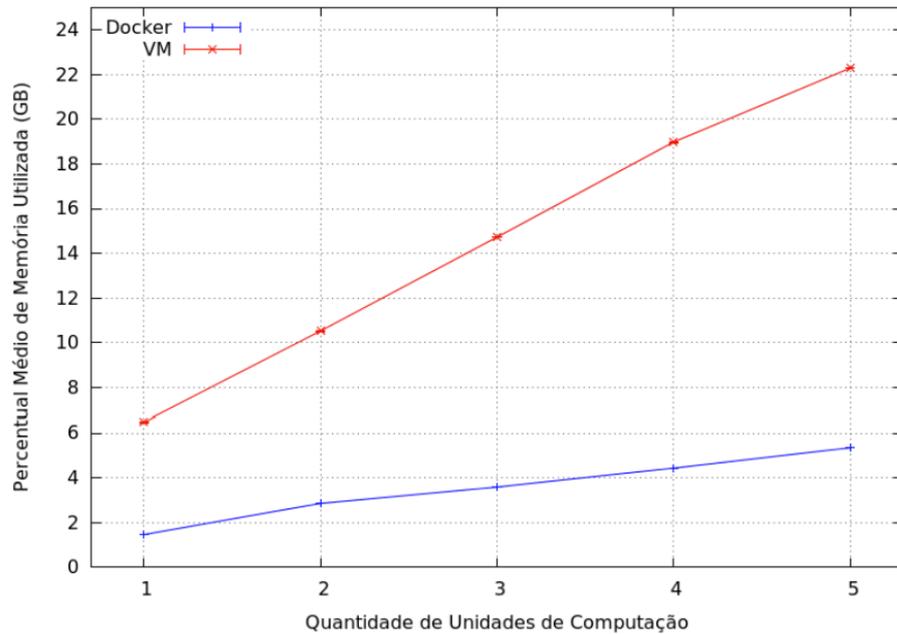
Figura 22 - Comparação na utilização de CPU.



Fonte: (SILVA, 2017)

Na Figura 23 pode ser visto a porcentagem na utilização de memória RAM entre as duas abordagens.

Figura 23 - Comparação da utilização de memória RAM.

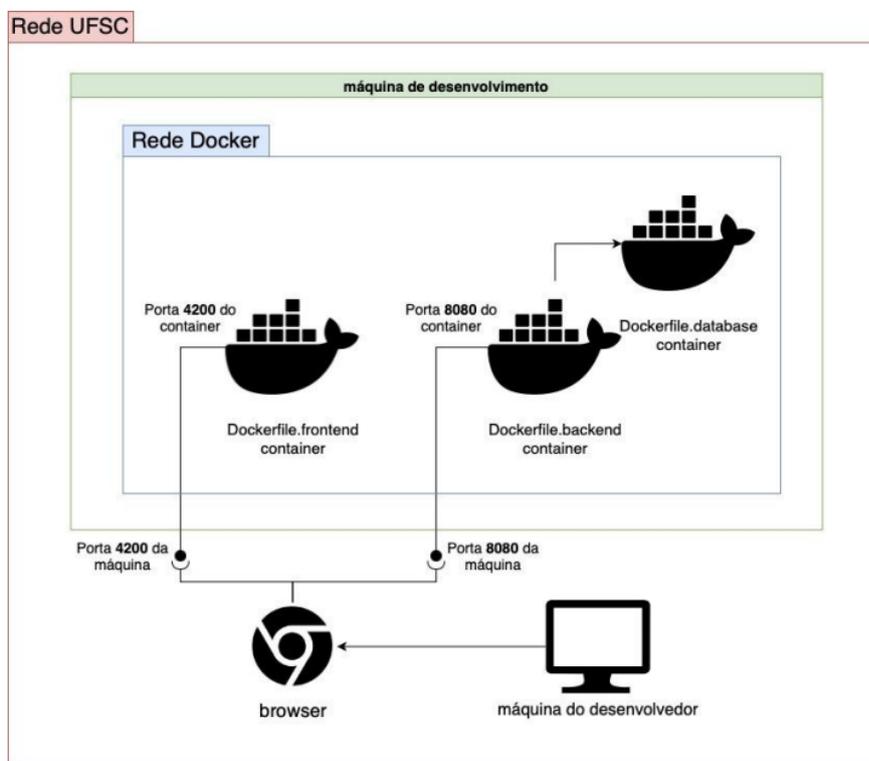


Fonte: (SILVA, 2017)

No trabalho desenvolvido por (ANDRADE, 2022), foram implementadas rotinas de Integração Contínua (*Continuous Integration – CI*) e de Entrega Contínua (*Continuous Deployment – CD*) em Docker, para automatizar os processos de entrega de software, integração e de testes para o Grupo de Pesquisas em Qualidade de Software (GQS) do Departamento de Informática e Estatística da Universidade Federal de Santa Catarina. A

Figura 24 representa o diagrama final do servidor em funcionamento com as rotinas de CI/CD implementadas.

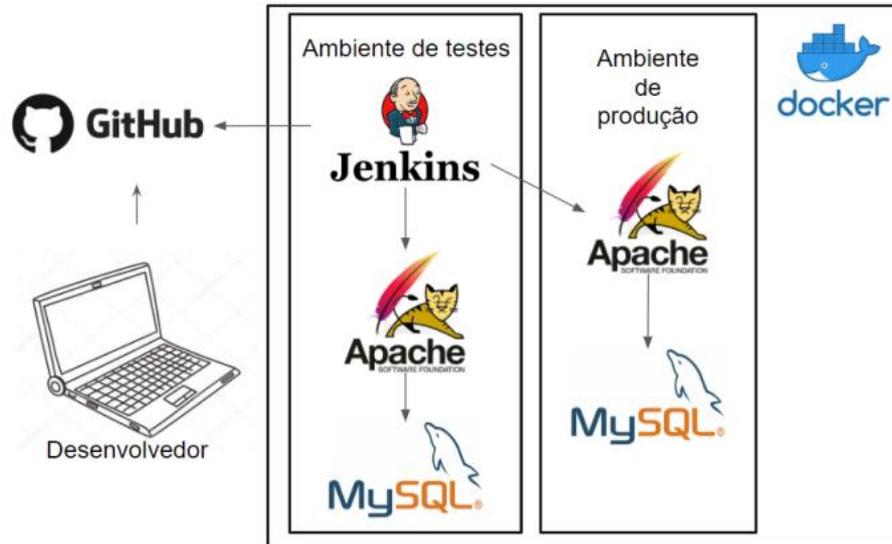
Figura 24 – Diagrama de funcionamento do servidor para o CI/CD implementado para o Departamento de Informática e Estatística da Universidade Federal de Santa Catarina.



Fonte: (ANDRADE, 2022).

Em (OLIVEIRA, 2019), foi implementado uma rotina de testes automatizados em Docker no Sistema Integrado de Gestão de Unidades de Alimentação e Nutrição (SIGUAN) para que o sistema pudesse continuar em desenvolvimento, pois novos módulos não estavam mais sendo desenvolvidos pela possibilidade de prejudicar parte funcional do *software*. A Figura 25 representa o diagrama da implementação.

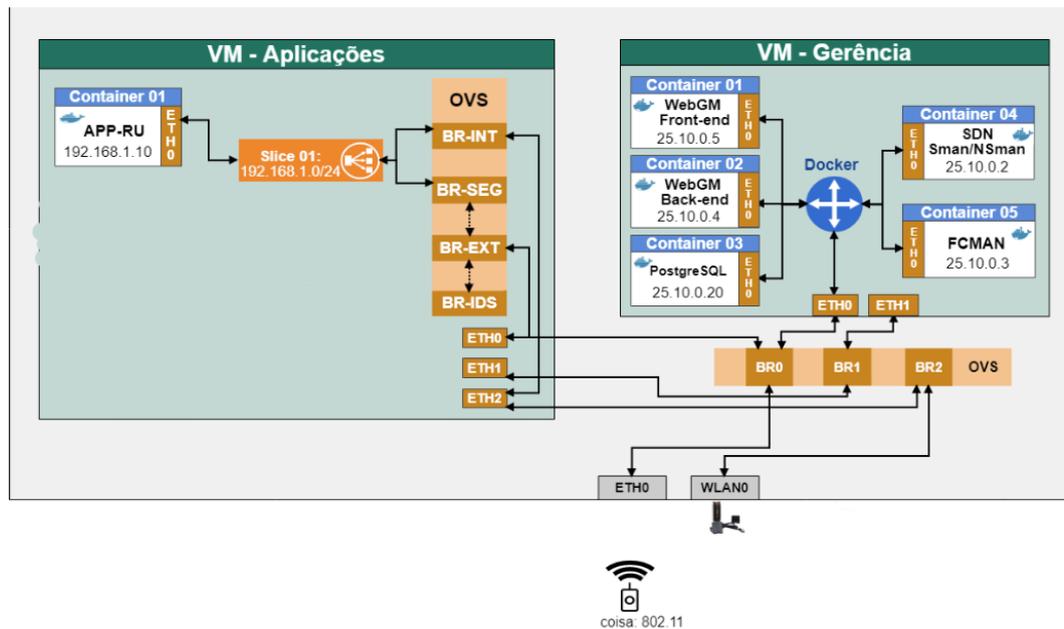
Figura 25 - Diagrama de funcionamento dos testes implementados Sistema Integrado de Gestão de Unidades de Alimentação e Nutrição.



Fonte: (OLIVEIRA, 2019).

No trabalho desenvolvido por (JÚNIOR *et al.*, 2019) foi implementado um Gateway, denominado Softway4IoT, em Docker para sistema de acompanhamento de filas para o Restaurante Universitário da UFG, Campus Samambaia. Foi realizada uma avaliação de performance para esse sistema e constatado que a perda de pacotes (dentre outras métricas) de dados foi de 1,5%. A Figura 26 representa o diagrama implementado.

Figura 26 - diagrama do software Softway4IoT.



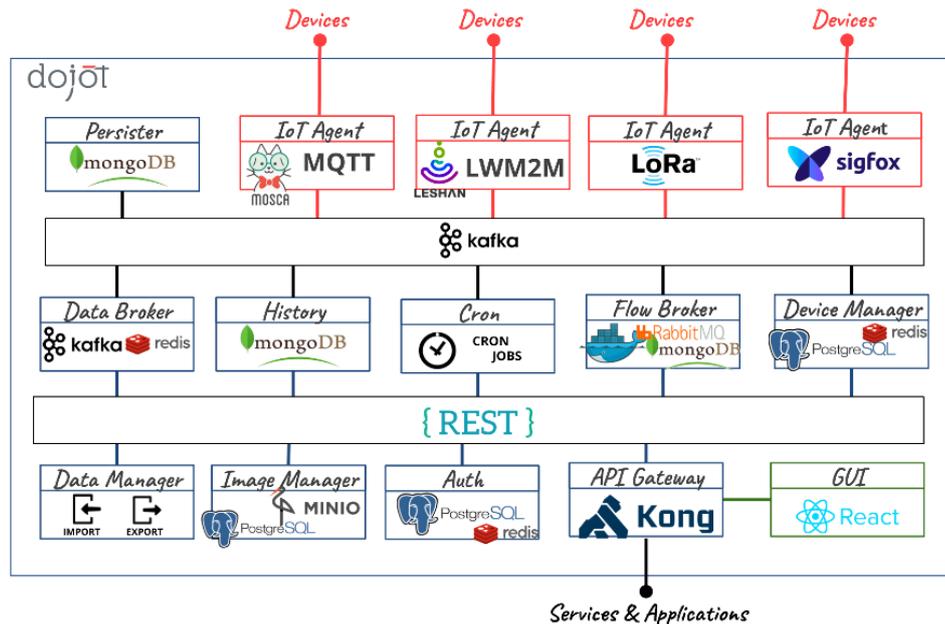
Fonte: (JÚNIOR et al., 2019).

2.2.2 DOJOT

De acordo com (SAMPAIO, 2018) a DOJOT é uma plataforma desenvolvida pelo CPQD para facilitar a implementação de arquiteturas IoT (*Internet of Things*) de modo que seja possível ter um *middleware* capaz de armazenar os dados temporais. E para (MORAES et al., 2021), é possível criar perfis de usuários, *templates*, dispositivos e fluxos de processamento de dados, em que os dispositivos configurados podem ser reais ou virtuais, além de receber diversos tipos de dados.

De acordo com (MEDEIROS, 2023), a *Dojot* é composta por diversos módulos internos organizados em camadas, de modo que existem módulos que são responsáveis pela aquisição de dados, como o *MQTT*, o *Lora*, o *Kafka* utilizado para o *Stream* de informação; e os módulos da camada de comunicação, tal como a interface gráfica e as APIs do sistema. A Figura 27 representa a arquitetura interna da DOJOT.

Figura 27 - Arquitetura interna do middleware DOJOT.



Fonte: (MEDEIROS, 2023).

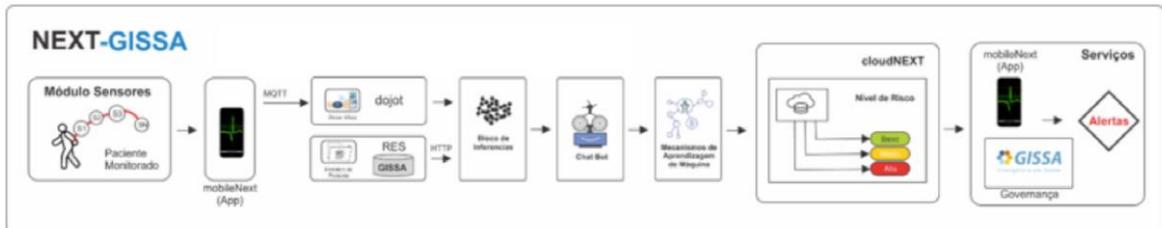
Para (MEDEIROS, 2023), a DOJOT possui a capacidade de comunicação com aplicações por meio das suas interfaces de comunicação, em que cada interface é compreendida por um protocolo. A plataforma possui uma gama de IoT's Agent, cuja missão principal é realizar a comunicação da DOJOT com as aplicações nas suas respectivas interfaces, isso inclui também a comunicação entre os dispositivos e a DOJOT por um canal seguro, como o uso de Lora ou MQTT.

Segundo (COLANTONI, 2022), a DOJOT possui dois componentes que trabalham em conjunto para o armazenamento e aquisição dos dados: *Persister* e o *History*. O módulo *Persister* é responsável pelo armazenamento de dados, sendo o armazenamento interno da DOJOT um banco de dados não relacional, *MonogDB*. Os dados que foram persistidos podem ser acessados por *requests* via API Rest, assim, esta API é fornecida pelo componente *History* ao qual permite consultar informações de dados armazenados, para um determinado dispositivo.

Em (JUNIOR et al., 2022), foi desenvolvido o Next-GISSA, um *software* IoT denominado Sistemas Inteligentes no trato do problema da Hipertensão Arterial. O trabalho propõe ampliar as funcionalidades de um *software* já existente, chamado de GISSA. O GISSA é uma plataforma que realiza coleta automática de um conjunto de dados e está presente em diversos municípios do Nordeste do Brasil. Nesse projeto,

são coletados dados referentes a pressão arterial, frequência cardíaca, oxigenação e temperatura, por meio de uma pulseira e enviados para a DOJOT, como pode ser visto na Figura 28.

Figura 28 - Componentes do Next-GISSA.

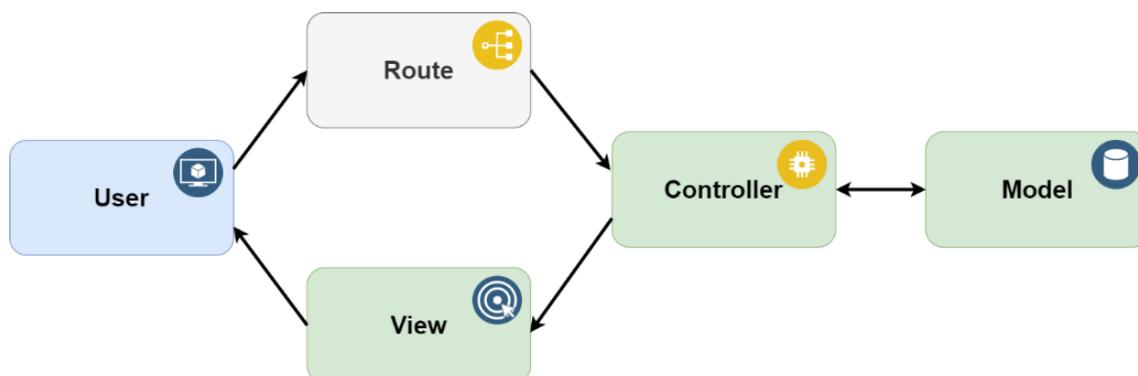


Fonte: (JUNIOR et al., 2022).

2.2.3 Framework Laravel

De acordo com (PELIZZA; BERTOLINI; SILVEIRA, 2018), o Laravel é um Framework que trabalha no modelo *MVC (Model-View-Controller)*, desenvolvido em PHP e utilizado para desenvolvimento de softwares escaláveis, com módulos que podem ser reutilizáveis, facilitando o desenvolvimento de sistemas complexos de forma ágil e segura. O MVC é responsável por entregar Frameworks: blocos de construção, funcionalidades e controle de fluxo entre páginas web, além de promover a separação da lógica de negócio. O Route é um componente do Laravel responsável por gerenciar os requests recebidos e encaminhar para o Controller quando necessário. A Figura 29 representa a camada MVC do Laravel.

Figura 29 - Camada MVC do Laravel.



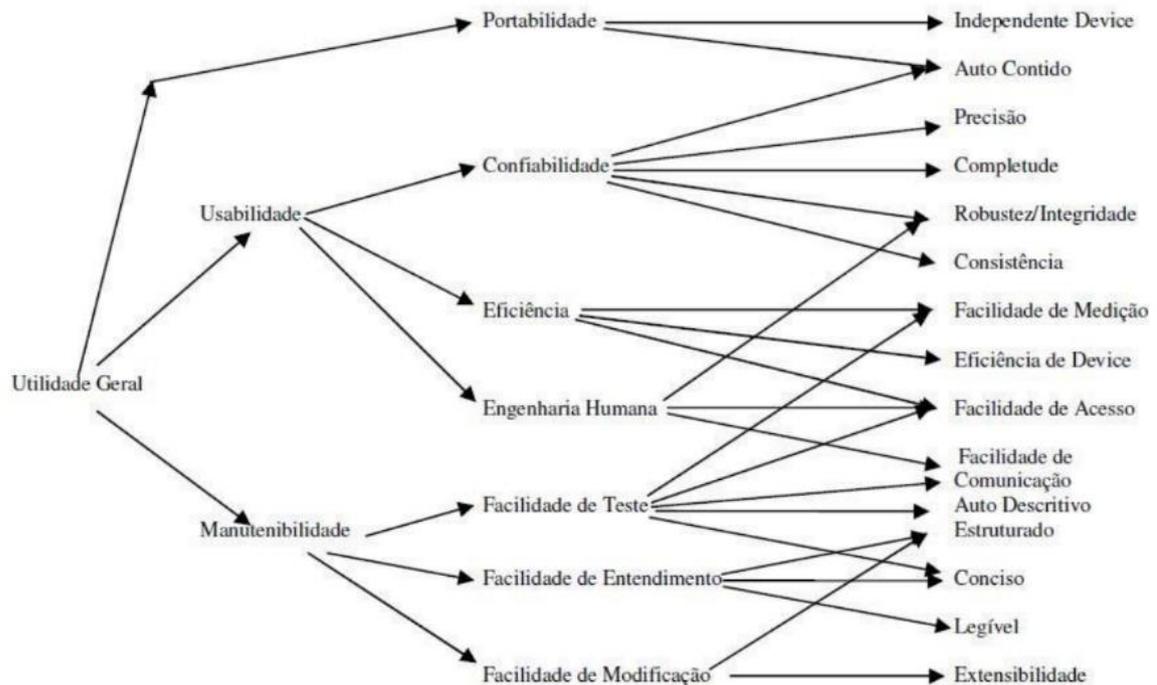
Fonte: O Autor.

Dado o modelo MVC, para (PELIZZA; BERTOLINI; SILVEIRA, 2018), tem-se:

- **Controller:** As requisições web trafegam sob os protocolos HTTP e HTTPS. A junção das informações de domínio do site, porta e o recurso acessado na URL é chamada de rota. O Laravel possui um sistema de mapeamento entre a rota solicitada e um controller mapeado, enviando o request recebido para o *controller*. O controller possui a função de validação e execução das regras de negócios.
- **Model:** Este módulo é responsável por modelar as entidades do banco de dados e, para tal, o Laravel usa o Eloquent como ferramenta de *ORM (Object Relational Mapping)*.
- **View:** Esta camada é responsável por receber as informações processadas pelo controller, tratar os dados recebidos para serem exibidos, montar o layout do frontend da aplicação e inserir os elementos visuais. A camada do modelo MVC é o local onde o usuário costuma interagir com a aplicação.

Para (SILVA, 2018), dentre as árvores de características de Qualidade de Software, é possível notar que a arquitetura que compreende o Laravel, possui os atributos necessários para ajudar na implementação de código confiável e escalável, como pode ser visto na Figura 30.

Figura 30 - Árvores de características de qualidade de softwares.



Fonte: (SILVA, 2018).

De acordo com (SILVA, 2018), o Laravel possui diversos componentes que são utilizados para facilitar o ciclo de vida de uma aplicação, partindo desde o desenvolvimento e criação de estruturas de banco de dados, até a implementação do sistema no servidor. Nesse contexto tem-se os componentes internos do Laravel.

A. Route

Segundo (FERREIRA, 2021), o Laravel possui um sistema de rotas para mapear um controller com uma requisição de acordo com os parâmetros recebidos, tal como os verbos do protocolo HTTP recebidos, podendo ser: GET, POST, PUT, PATCH, DELETE e OPTIONS. Essa rota redireciona a solicitação para o controller e retorna o que foi processado, podendo ser o acesso a uma informação ou a uma tela do sistema. Com o Laravel é possível inserir *middlewares* para atuarem em rotas, de modo que o controller não seja o responsável por fazer validação do token expirado, verificar a existência de usuário logado na aplicação, verificar condições antes de entrar em um fluxo, etc.

B. Artisan

Para (SANTOS; MARQUES, 2022), o artisan é a interface utilizada para linha de comando no Laravel, fornecendo uma gama de comandos para construir a aplicação, tais como a criação dos componentes vistos na camada de MVC, com a criação dos controllers; dos models e das views que pertencem ao sistema. Além disso a interface do artisan pode construir as classes de middleware e a camada de gerenciamento de acesso ao banco de dados, como os seeds e migrations. Seeds são utilizados como pontos de entrada de dados em uma aplicação em desenvolvimento, de modo que seja capaz de popular o banco de dados. Já as migrations são responsáveis por modelar a estrutura do banco de dados e criar uma base para que os dados possam ser persistidos.

C. Blade

De acordo com (SILVA, 2018), o Blade é um sistema de templates que o Laravel utiliza para abstrair e modelar informações para as páginas web em PHP na camada de view. O blade compila os arquivos que possuem a extensão *.blade.php* para a execução de forma rápida em cache, isso permanece até que aconteça alguma modificação no arquivo original. O blade também tem suporte para estruturas de programação que o HTML sozinho não suporta, tais como condicionais e laços de repetição. O blade auxilia na reutilização do código, por meio da criação de pequenos trechos de código, para serem utilizados em diversos pontos do sistema sem necessidade de duplicidade.

D. Eloquent ORM

De acordo com (PELIZZA; BERTOLINI; SILVEIRA, 2018), o Laravel possui a capacidade de realizar mapeamentos entre a aplicação e o banco de dados, para isso ele utiliza o Eloquent, um mapeador objeto-relacional (ORM) que facilita a integração com o banco de dados e a execução de queries, além de prover a melhora da segurança do sistema. Tendo em vista o Eloquent, cada tabela de banco de dados é mapeada em um objeto Model, o qual representa a tabela, podendo assim realizar todo o processo necessário de CRUD (create, read, update e delete).

Além de padrões de arquitetura o Framework Laravel dispõe de aspectos gerais de segurança, sendo este um ponto em comum durante o desenvolvimento de

aplicações web. A *Open Web Application Security Project* (OWASP) divulga vulnerabilidades críticas que podem ser exploradas e precisam de atenção durante o desenvolvimento de sistemas web, esses pontos são devidos à sua natureza de alta disponibilidade em aplicações voltadas para serviços web e costumam ser alvo para usuários maliciosos.

A OWASP é uma entidade online reconhecida internacionalmente, que tem como objetivo aumentar a segurança no desenvolvimento de softwares ao redor do mundo, de modo a tornar visível os pontos inerentes a segurança em softwares. A organização é responsável pela criação de diversos projetos focados em segurança, dentre eles tem-se o *OWASP Top Ten*, que é responsável por elencar uma lista com as 10 maiores vulnerabilidades que podem ocorrer em sistemas web.

Segundo a OWASP, existem regras de segurança a serem empregadas para o tratamento das principais vulnerabilidades no decorrer do desenvolvimento de softwares web e de acordo com o levantado pela *OWASP Top Ten*, tem-se as seguintes vulnerabilidades:

- A1 - Injeção: Falhas de Injeção ocorrem quando dados não confiáveis são inseridos nos campos de entrada como parte de um comando válido.
- A2 - Quebra de Autenticação: Atacantes podem assumir identidades de outros usuários devido ao comprometimento das senhas ou tokens de sessão.
- A3 - Exposição de Dados Sensíveis: Dados sensíveis como número de cartão de crédito e dados pessoais necessitam de segurança tanto no armazenamento como em trânsito.
- A4 - Entidades Externas de XML: Vulnerabilidade ocasionada devido a processadores XML antigos ou mal configurados que avaliam entidades externas em documentos XML.
- A5 - Quebra de Controle de Acessos: Atacantes podem ter acesso a dados ou funcionalidades não autorizadas devido à falta de implementação correta das restrições que cada usuário pode fazer no sistema.
- A6 - Configurações de Segurança Incorretas: Tem como objetivo as configurações definidas na aplicação e como elas podem trazer vulnerabilidades a aplicações mal definidas.

- A7 - Cross-Site Scripting: O atacante pode inserir código malicioso no navegador da vítima por falta de validação.
- A8 - Desserialização Insegura: Invasores podem realizar ataques de injeção, por repetição ou alteração de privilégios
- A9 - Utilização de Componentes Vulneráveis: Componentes externos como bibliotecas e módulos estão vulneráveis a ataques externos e são implementados na aplicação.
- A10 - Registo e Monitorização Insuficiente: Atacantes continuam a realizar ataques às aplicações devido a resposta a incidentes serem insuficientes ou até inexistentes.

No trabalho feito por (FERRAO; DE MACEDO; KREUTZ, 2018) foi feita a identificação dos frameworks mais utilizados para o desenvolvimento de aplicações Web baseada em PHP, a implementação das dez vulnerabilidades mais recorrentes em sistemas Web em cada um dos frameworks seleccionados em um ambiente controlado e a execução e avaliação de dez scanners de vulnerabilidades para analisar o impacto dos frameworks na segurança de sistemas Web. Foi constatado que o Laravel e o Symfony possuem as maiores coberturas entre as análises feitas, como pode ser visto na tabela abaixo.

Tabela 3 - Vulnerabilidades detectadas no cenário controlado.

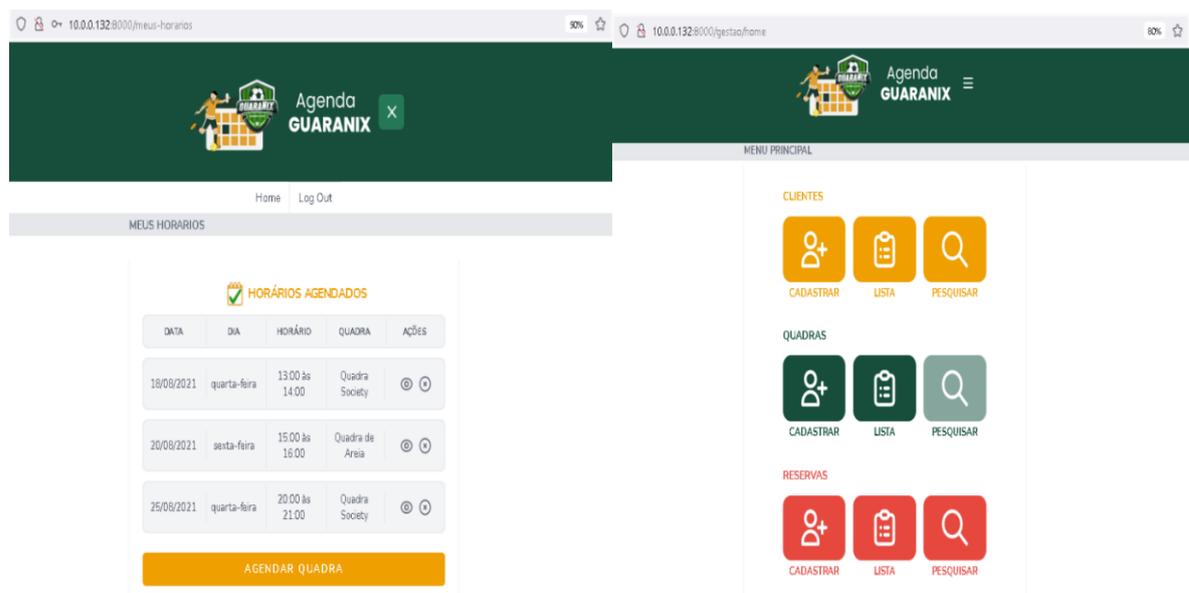
Framework	% Vuln.
Laravel	40
Symfony	40
Yii	50
Zend	50
Cakephp	60
Codeigniter	70
Phalcon	70

Fonte: (FERRAO; DE MACEDO; KREUTZ, 2018)

No trabalho feito por (FERREIRA, 2021), foi desenvolvida uma aplicação em Laravel para realizar agendamentos de horários para o uso das quadras do Guarani,

visando prover o gerenciamento das quadras, como agendamento, horários e clientes de acordo com as necessidades do gestor. Na figura 35 é possível ver as áreas de acesso do gestor e do cliente.

Figura 31 - Páginas principais da Área do cliente (esquerda) e área do Gestor (direita) das quadras do GuaraniX.



Fonte: (FERREIRA, 2021).

Em (SANTOS; MARQUES, 2022), é apresentada a implementação de um software desenvolvido no Framework Laravel, para registrar e guardar dados dos alunos de creches nos Centro de Educação Infantil da Prefeitura do município de Coxim-MS, podendo ser utilizado por todas as unidades ligadas a secretaria Municipal de Educação. A Figura 32 demonstra uma das telas do Atendente, o qual pode Cadastrar, Matricular e Editar os dados dos alunos.

Figura 32 - Tela de visualização do atendente.

ID	Nome	DataNasc	Mae	Pai	TelResponsavel	Ação	Ação	Matricular
1	adao Aluno	2018-11-13	mae adao	pai adao	1111111111111111	Editar	Remover	Matricular
2	eva aluna	2018-11-01	mae eva	pai eva	22222222222222	Editar	Remover	Matricular
3	yasmin aluna	2018-11-27	mae yasmin	pai yasmin	3332222222	Editar	Remover	Matricular
4	tayna	2018-11-26	mae tayna	pai tayna	99993333322	Editar	Remover	Matricular
5	leandro	2016-02-10	mae leandro	pai leandro	filho	Editar	Remover	Matricular
6	eder aluno 2	1972-03-06	mae eder	pai eder	88888888899	Editar	Remover	Matricular

Fonte: (FERREIRA, 2021).

O trabalho desenvolvido por (CAMPOS, 2020), tem como o objetivo a implementação de um sistema de gerenciamento de mudas. Para isso, tem-se o desenvolvimento do sistema denominado Sistema de Gerenciamento do Viveiro de Mudas do IFMG-SJE (SGVM IFMG-SJE), em que foram pautados os objetivos de compreender e extrair informações relevantes à produção e expedição de mudas no viveiro. Na figura 31 é possível ver a tela inicial, com um dashboard com dados quantitativos em relação ao viveiro.

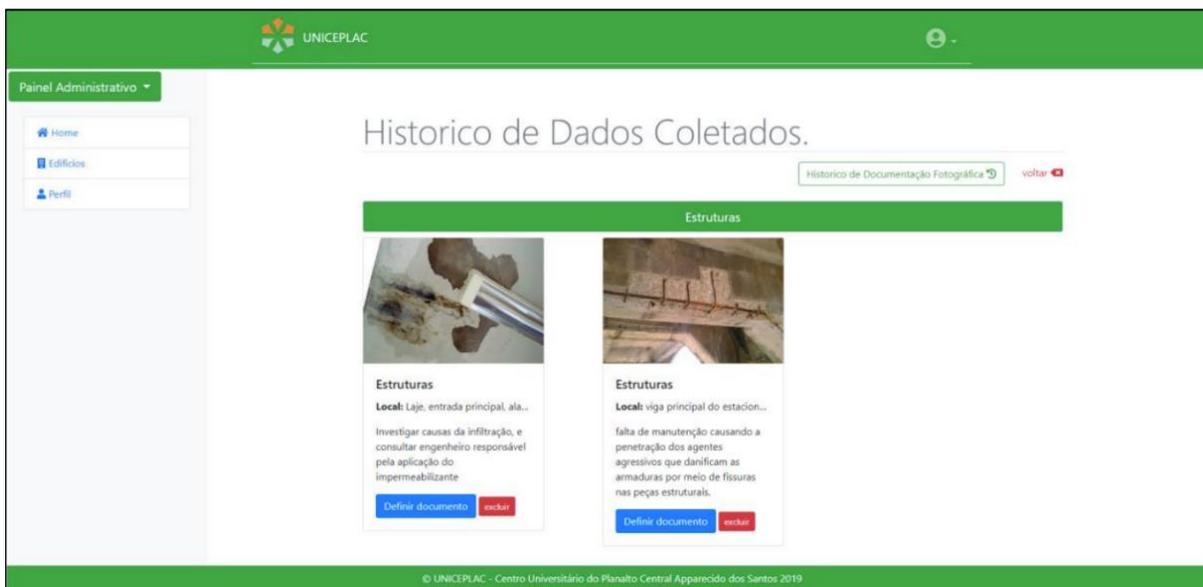
Figura 33 - Tela de dashboard do Sistema de Gerenciamento do Viveiro de Mudas do IFMG-SJE.



Fonte: (CAMPOS, 2020).

Em (NONATO; MENDES; MENDES, 2020), foi desenvolvida uma aplicação para auxiliar os engenheiros e arquitetos especializados em inspeções prediais, realizando a criação de uma lista de coleta dados de inspeções em campo de forma eletrônica, para otimizar a geração de laudo com base no relatório fotográfico gerado pelo sistema, além de gerar histórico dos edifícios e vistorias. A Figura 34 demonstra o histórico de coleta do sistema.

Figura 34 – Histórico de coleta de dados do Sistema de inspeção predial para engenheiros e arquitetos.



Fonte: (NONATO; MENDES; MENDES, 2020).

2.3 Conclusão

Neste capítulo foi discorrido a respeito do panorama geral da geração de energia elétrica brasileira e acerca das classificações de sistemas de geração de energia fotovoltaica e sistemas de armazenamento de energia. Dentro do capítulo tem-se discursões a respeito das tecnologias empregadas no trabalho, o uso do Docker para o gerenciamento de containers e como ele atua, facilitando o processo de implementação de sistemas. Também se tem a DOJOT atuando como middleware para as aplicações visando o armazenamento de dados de medições e a criação de uma interface de API para a aquisição dos dados, tem-se também o Laravel como um framework, capaz de auxiliar o desenvolvimento de código e aspectos de segurança e qualidade de software.

Como conclusão do capítulo, é possível compreender a necessidade de inserção, monitoramento e gerenciamento de fontes renováveis de geração energia elétrica no sistema elétrico brasileiro, visando uma visão geral da demanda e geração por energia elétrica. Também se tem o entendimento de como as tecnologias podem ser utilizadas para implementações de *softwares* para o sistema de gestão.

CAPÍTULO 3: METODOLOGIA

3 Introdução

Nesse capítulo será comentado sobre a metodologia de desenvolvimento para o uso das tecnologias destacadas no capítulo anterior e dos principais pontos de arquitetura de aplicação utilizados para a usabilidade deste trabalho.

3.1 Servidor de Implementação do Software

Os sistemas foram desenvolvidos na Universidade Federal do Pará, sendo os softwares aqui presentes implementados no Laboratório CEAMAZON (Centro de Excelência em Eficiência Energética da Amazônia), para tal, foi instalado um servidor na sala de servidores para servir às aplicações descritas, tendo as suas características de hardware descritas na tabela abaixo:

Tabela 4 - Servidor de Implementação.

Modelo	Power Edge T440
Processador	Intel Xeon Silver
Núcleos	20 Principais e 20 Secundários
Memória RAM	2 pentes de 32Gb (64Gb)
Controladora Raid	PERC H330
Memória ROM	5 HD's SATA 7.2k RPM de 1Tb (5Tb)
Potência	450 W
Placa de Rede	2 portas Ethernet de 1Gbit/s

Fonte: O Autor.

Dado os requisitos, a Figura 35 representa o servidor descrito.

Figura 35 - Servidor Power Edge T440 configurado no CEAMAZON.



Fonte: O Autor.

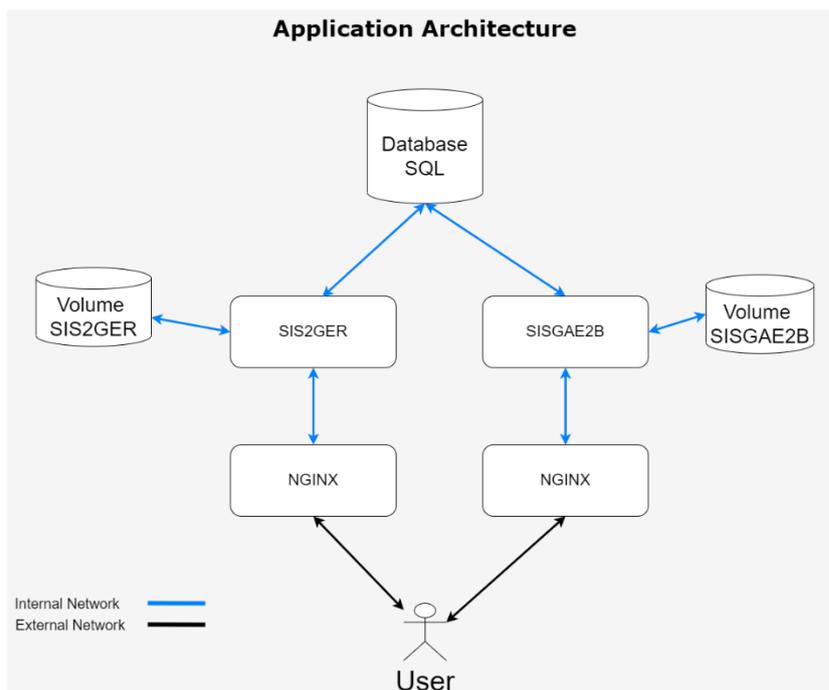
3.2 Desenvolvimento

A codificação dos sistemas foi desenvolvida em PHP utilizando o Laravel como framework principal, foram implementados os modelos representando o banco de dados dentro da aplicação para a criação das entidades ORM e foram criadas as rotas Web no arquivo de configurações de rotas, para que fosse possível adicionar um *Controller* responsável para cada um dos endpoints construído para a aplicação.

3.2.1 Docker

Uma vez que os sistemas foram desenvolvidos e implementados em Docker, sua estrutura pode ser definida por meio de arquivos de configurações para todos os containers levantados e para cada sistema foram criados arquivos de configurações. A Figura 36 representa a arquitetura criada para que as aplicações se comunicassem e fossem disponíveis para os usuários via Web.

Figura 36 – Arquitetura das aplicações desenvolvidas.



Fonte: O Autor.

Para a criação e gerenciamento dos containers foi utilizado o Docker Composer, sendo este configurado pelo arquivo *docker-compose.yml*, e cada software desenvolvido apresenta o seu arquivo de configuração. A função do Docker Composer é facilitar a criação das imagens e volumes para a persistência de dados; definir networks para a comunicação entre os containers; e passar parâmetros de configuração. A Figura 37 representa o *docker-compose.yml* do banco de dados, que define 2 serviços: o *sima-db* e o *sima-db-phpmyadmin*. O *sima-db* é definido na linha 3 do programa, sendo o serviço do banco de dados dos sistemas; e o *sima-db-phpmyadmin* é definido na linha 21, sendo o serviço para o gerenciamento via Web do banco de dados. As variáveis de ambientes estão inseridas nas linhas de 8 a 12 e na linha 30. Para as configurações do sistema (linha 34), tem-se a rede e as portas que serão utilizadas por cada serviço para a comunicação entre os containers. Já o volume para o armazenamento dos dados foi inserido na linha 38.

Figura 37 - Arquivo docker-compose.yml do banco de dados.

```
1  version: "3.7"
2  services:
3    db:
4      image: mysql:8
5      container_name: sima-db
6      restart: always
7      environment:
8        MYSQL_DATABASE: sima-db
9        MYSQL_ROOT_PASSWORD: #####
10       MYSQL_PASSWORD: #####
11       MYSQL_USER: #####
12       SERVICE_NAME: mysql
13     volumes:
14       - sima-datavolume:/var/lib/mysql
15       - ./docker-compose/mysql:/docker-entrypoint-initdb.d
16     ports:
17       - "3306:3306"
18     networks:
19       - network
20
21   phpmyadmin:
22     image: phpmyadmin/phpmyadmin:latest
23     container_name: sima-db-phpmyadmin
24     restart: always
25     links:
26       - db
27     ports:
28       - 3307:80
29     environment:
30       - PMA_ARBITRARY=1
31     networks:
32       - network
33
34   networks:
35     network:
36       name: sima_network
37
38   volumes:
39     sima-datavolume: {}
```

Fonte: O Autor.

De forma semelhante, também foram criados arquivos de configuração do Docker Compose para os softwares desenvolvidos. A Figura 38 representa o arquivo de configuração do SIS2GER, onde é possível verificar a definição dos serviços, volumes e networks.

Figura 38 - Arquivo docker-compose.yml do SIS2GER.

```

1  version: "3.7"
2  services:
3    app-sima-web-fotovoltaico:
4      image: registry.gitlab.com/ceamazon/sima/sima-web:v0.2.6
5      restart: always
6      container_name: sima-web-fotovoltaico
7      working_dir: /var/www/
8      command:
9        - php-fpm
10     volumes:
11       - sima_fotovoltaico_web_data:/var/www
12     environment:
13       - DB_HOST=sima-db
14       - DB_PORT=3306
15       - DB_DATABASE=#####
16       - DB_USERNAME=#####
17       - DB_PASSWORD=#####
18       - DOJOT_URL="http://10.107.1.102"
19       - DOJOT_USER=#####
20       - DOJOT_PASS=#####
21       - GGER_URL=http://10.107.1.110:8010
22       - GSAE_URL=http://10.107.1.110:8011
23       - SIAR_URL=http://10.107.1.110:8012
24       - PlanIoT_URL=http://10.107.1.110:8013
25       - GSAM_URL=http://10.107.1.110:8014
26       - SIMA_URL=http://10.107.1.110:8015
27       - GCEM_URL='#'
28       - EM_URL='#'
29       - MGEESM_URL='#'
30       - PLAY_STORE_URL=#####
31       - MAIL_MAILER=#####
32       - MAIL_HOST=#####
33       - MAIL_PORT=#####
34       - MAIL_USERNAME=#####
35       - MAIL_PASSWORD=#####
36       - MAIL_ENCRYPTION=#####
37       - MAIL_FROM_ADDRESS=#####
38       - APP_DEBUG=#####
39     nginx:
40       image: nginx:alpine
41       container_name: sima-web-fotovoltaico-nginx
42       restart: always
43       ports:
44         - 8010:80
45       volumes:
46         - sima_fotovoltaico_web_data:/var/www
47         - ./docker-compose/nginx:/etc/nginx/conf.d/
48
49     networks:
50       default:
51         external:
52           name: sima_network
53
54     volumes:
55       - sima_fotovoltaico_web_data: {}

```

Fonte: O Autor.

Dentro do *software* (Figura 38), foi utilizado o serviço do NGINX para prover acesso aos sistemas criados, de modo que este atua como *proxy* dentro da aplicação,

sendo responsável por encaminhar as requisições feitas para a aplicação para as portas correspondentes de cada uma. O arquivo de configuração do NGINX do SIS2GER (Figura 38), é responsável por encaminhar os requests para *app-sima-web-fotovoltaico* na porta 9000, como definido na linha 3 da Figura 38.

Figura 39 - Arquivo de configuração do NGEX do do SIS2GER.

```

1  server {
2      listen 80;
3      index index.php index.html;
4      error_log /var/log/nginx/error.log;
5      access_log /var/log/nginx/access.log;
6      root /var/www/public;
7      location ~ /\.php$ {
8          try_files $uri =404;
9          fastcgi_split_path_info ^(.+\.(php|\.+))$;
10         fastcgi_pass app-sima-web-fotovoltaico:9000;
11         fastcgi_index index.php;
12         include fastcgi_params;
13         fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
14         fastcgi_param PATH_INFO $fastcgi_path_info;
15     }
16     location / {
17         try_files $uri $uri/ /index.php?$query_string;
18         gzip_static on;
19     }
20 }

```

Fonte: O Autor.

Dado a descrição do processo de configuração das imagens dos sistemas para que essas possam ser gerenciadas pelo Docker Compose, a construção pode ocorrer por meio de um arquivo de configuração chamado *Dockerfile*, responsável por construir o ambiente necessário para a execução do sistema seguindo uma sequência de etapas definidas no arquivo, de modo que para os softwares a sequência de construção se dá em:

- a) Selecionar uma imagem PHP para ser o padrão utilizada no sistema.
- b) Instalar as dependências da versão do da imagem escolhida.
- c) Instalar o Composer do PHP na imagem;
- d) Instalar o NPM para na imagem;
- e) Copiar os arquivos da aplicação;
- f) Executar a instalação das dependências do *software* por meio do composer;
- g) Instalar as dependências do *software* por meio do NPM;
- h) Executar o NPM no modo produção para compilar todos os Assets e deixar os arquivos CSS e JS Mimificados.

A Figura 40 representa o arquivo Dockerfile de ambos os softwares.

Figura 40 - Arquivo Dockerfile.

```
FROM php:7.4-fpm

# Install system dependencies
RUN apt-get update && apt-get install -y \
    git curl libpng-dev libonig-dev libxml2-dev zip unzip
RUN apt-get update -yq \
    && apt-get install curl gnupg -yq \
    && curl -sL https://deb.nodesource.com/setup_12.x | bash \
    && apt-get install nodejs install libcurl4-openssl-dev pkg-config libssl-dev -yq

# Install PHP extensions
RUN docker-php-ext-install pdo_mysql mbstring exif pcntl bcmath gd

# Get latest Composer
COPY --from=composer:latest /usr/bin/composer /usr/bin/composer

# Configure path
WORKDIR /var/www
COPY . /var/www

# Create system user to run Composer and Artisan Commands
RUN useradd -G www-data,root -u $uid -d /home/$user $user

RUN apt-get install -y sudo
RUN adduser --disabled-password \
    --gecos '' docker
RUN adduser docker sudo
RUN adduser www-data sudo
RUN echo '%sudo ALL=(ALL) NOPASSWD:ALL' >> \
    /etc/sudoers

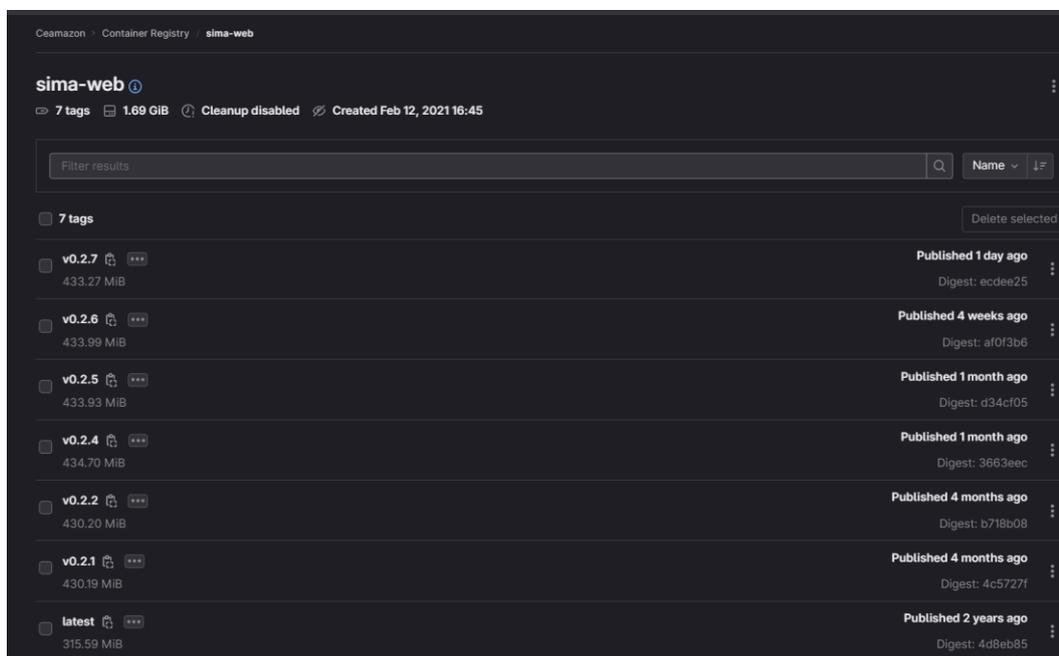
# now we can set USER to the
RUN sudo rm -rf /var/www/storage/logs
RUN sudo rm -rf /var/www/node_modules
RUN cd /var/www && sudo npm clean-install
RUN cd /var/www && sudo npm install --global cross-env
RUN cd /var/www && sudo composer install --no-dev
RUN cd /var/www && sudo composer dump-autoload

USER www-data
RUN sudo chown -R www-data:www-data /var/www
```

Fonte: O Autor.

Dado que a aplicação foi construída, a imagem gerada possui um número de versão, ou seja, uma tag. A tag gerada é utilizada para marcar as versões de código e das imagens container do software. Na sequência, a tag é enviada para o registry do gitlab, como pode ser visto na Figura 40.

Figura 41 - Imagens do container do SIS2GER.



Fonte: O Autor.

Sabendo que a nova imagem já está disponível no sistema, é possível atualizar o *software* de forma extremamente rápida (comumente chamado de *deploy*) e realizar a volta caso ocorra algum problema (comumente chamado de *rollback*).

3.3 Persistência de dados

A começar pela camada mais baixa das aplicações, os *softwares* desenvolvidos são aplicações Web que possuem a capacidade de se comunicar com os serviços necessários para a obtenção de dados inerentes ao seu funcionamento. Para que seja possível salvar dados da aplicação, tem-se a presença de um banco de dados responsável pela persistência de dados da aplicação. Serão mencionados aqui as formas de persistência dos dados dos *softwares*.

Como mencionado, os sistemas possuem 2 pontos principais de fonte de dados nos quais os *softwares* desenvolvidos podem se comunicar:

- DOJOT como Middleware: A DOJOT é utilizada na plataforma como um banco de dados para séries temporais, persistindo os dados em um banco

NoSQL, de modo em que os dispositivos podem ser configurados e os dados referentes a eles podem ser consumidos diretamente na DOJOT.

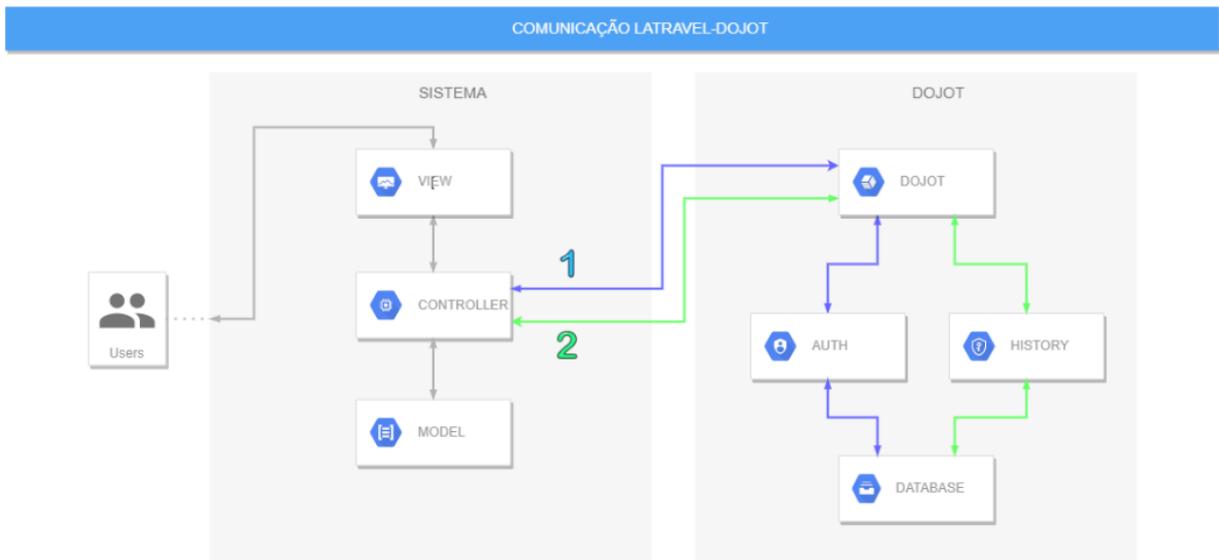
- b) Banco de Dados Relacional: O banco de dados SQL é utilizado para mapear as entidades internas, com o objetivo de ter maior controle em relação aos pontos configuráveis do *software*, como os grupos, usuários, dados dos medidores, permissões, logs etc.

3.3.1 DOJOT

Uma das formas de aquisição de dados na plataforma DOJOT ocorre por meio da API History, que é integrada na plataforma. A API necessita que seja passado como parâmetro um token de segurança JWT (JSON Web Token). Sendo assim, se faz necessário realizar uma requisição para o acesso do token, por meio de um usuário cadastrado no sistema (DOJOT), e após o recebimento do token, será realizado um novo acesso de consulta para aquisição de dados.

A comunicação é realizada no Controller da aplicação, o qual irá retornar um JSON que é mapeado para uma entidade do sistema. Na Figura 42 é apresentado o fluxo descrito no formato MVC, onde o fluxo 1 representa o processo de autenticação e recebimento do token; e o fluxo 2 representa a aquisição dos dados desejados, ambos na camada de Controller das aplicações.

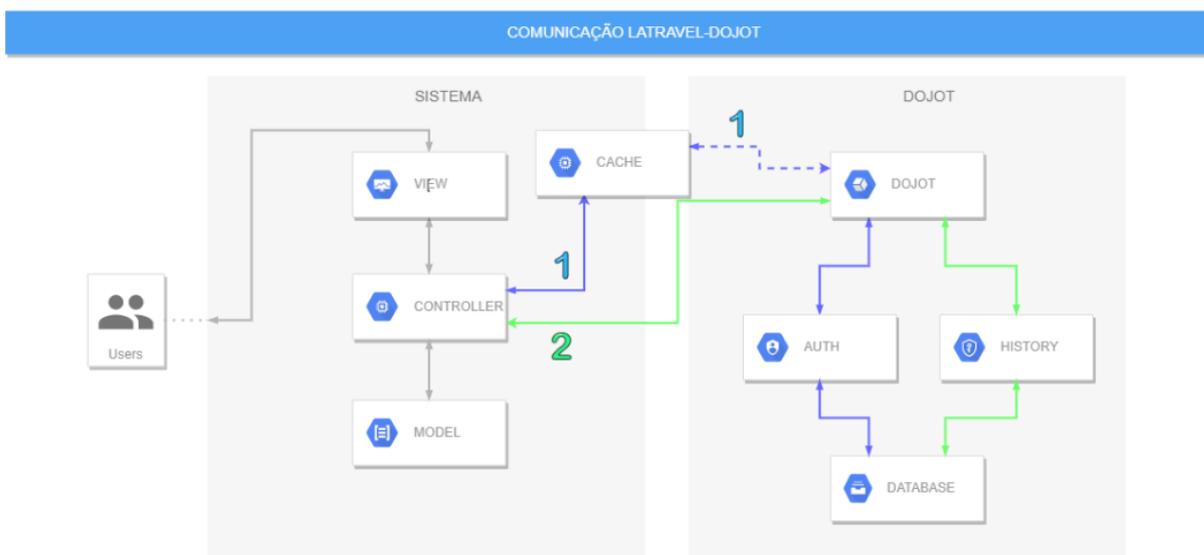
Figura 42 - Fluxo de comunicação com a DOJOT.



Fonte: O Autor.

Para evitar realizar 2 requests para a DOJOT, sempre que um usuário precisar ver algum gráfico ou informação na tela, foi implementado um sistema de cache em memória dentro da aplicação, sendo assim, antes do Controller acessar a DOJOT, ele verifica se já possui o token de autenticação, fazendo com que o sistema fique mais rápido. A Figura 43 representa o fluxo descrito.

Figura 43 - Fluxo de comunicação com a DOJOT com cache.

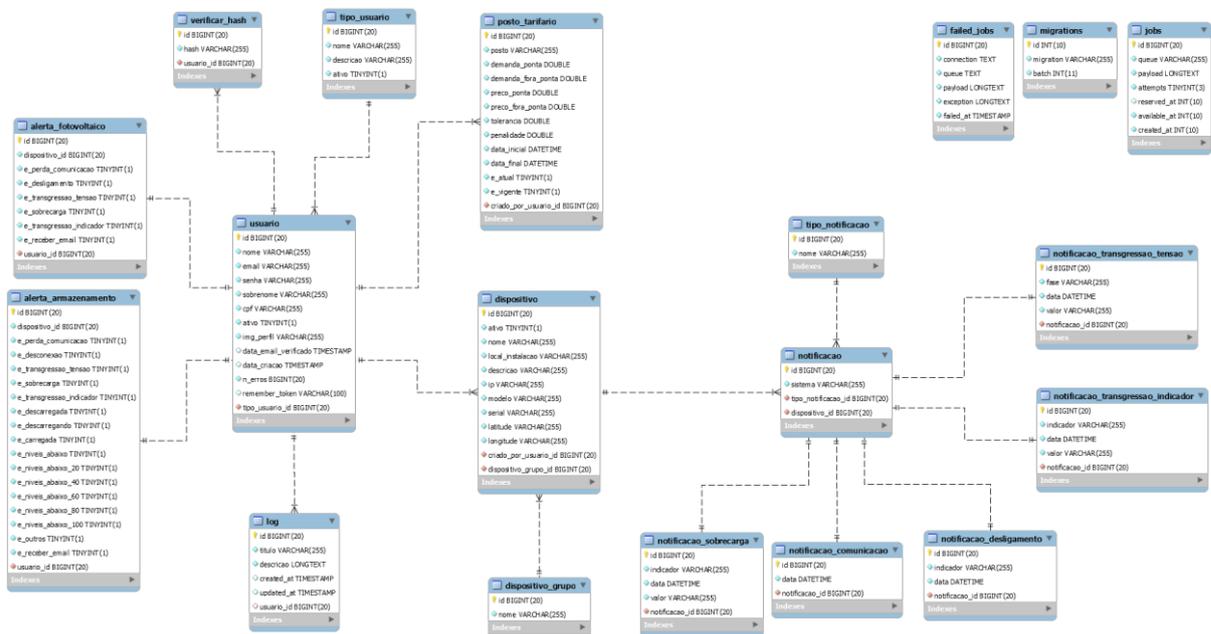


Fonte: O Autor.

3.3.2 Banco de Dados Relacional

Um banco de dados relacional é composto por entidades (tabelas) e relacionamentos entre as entidades (podendo ser 1:1, 1:N, N:1 ou N:N) que comumente são exibidas por meio de um MER (Modelo Entidade Relacionamento). Na Figura 44 será apresentada o MER com os principais relacionamentos do banco de dados desenvolvidos para os softwares.

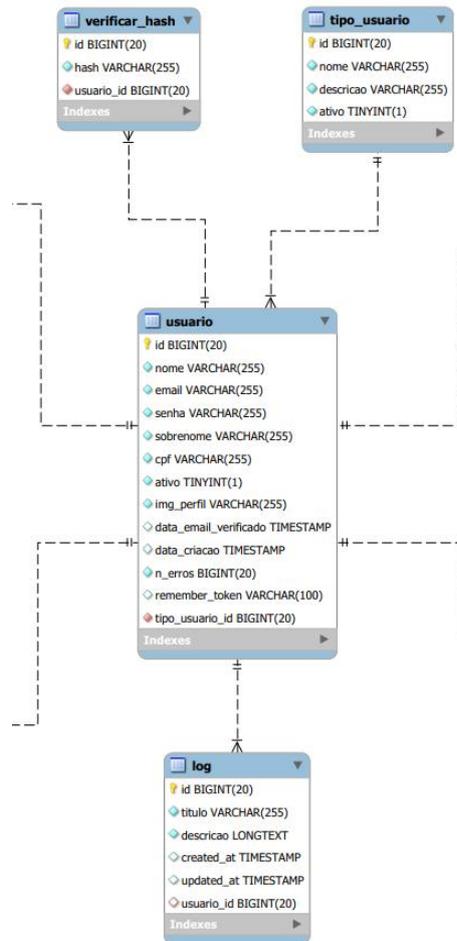
Figura 44 - Modelo Entidade Relacionamento dos softwares desenvolvidos.



Fonte: O Autor.

Com base no diagrama apresentado, é possível notar que existe um conjunto de tabelas dependentes da tabela *usuario*, sendo essa a tabela principal dos sistemas. A tabela *usuario* possui relacionamento com as tabelas: *verificar_hash*, *tipo_usuario* e *log*, sendo estas entidades usadas para o gerenciamento de usuário. A Figura 45 apresenta as tabelas citadas.

Figura 45 - MER com tabela usuario em destaque.



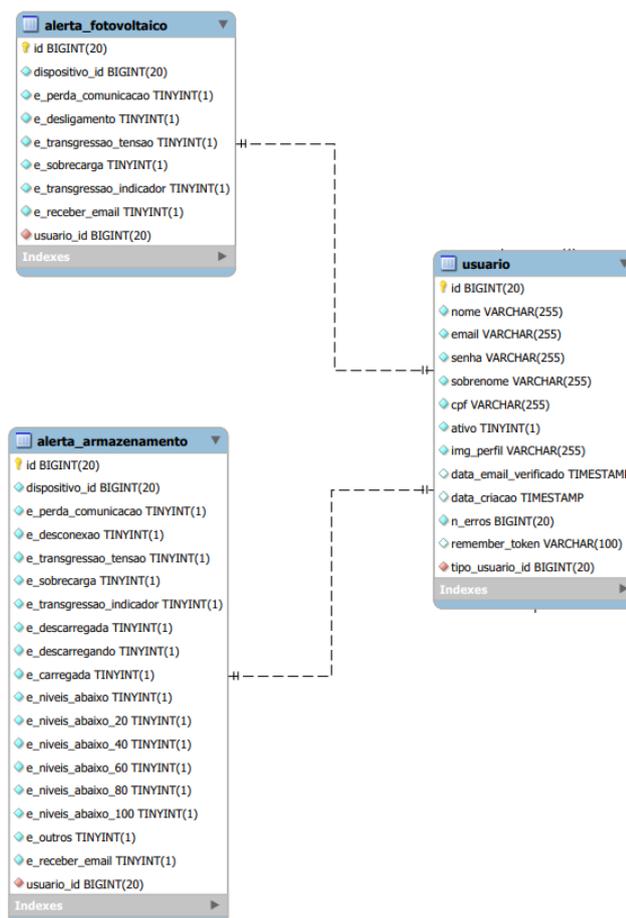
Fonte: O Autor.

- verificar_hash*: Utilizada para realizar autenticações de ações que ocorrem fora do sistema, por meio do uso de um conjunto de caracteres únicos (hash) para a validação de requisições. A validação ocorre sem a necessidade de o usuário conhecer a hash utilizada ou o seu mecanismo de funcionamento, garantindo maior segurança ao software. Como exemplo, tem-se a validação do e-mail e pedidos de solicitação de alteração de senha.
- tipo_usuario*: Esta tabela é responsável por criar diversos níveis de usuários e permissões de usuários, tais como: Administrador e Normal.
- log*: Esta tabela existe para auxiliar a manutenção da integridade dos dados no sistema, sendo responsável por armazenar as ações sensíveis dentro do software, como: tornar um usuário Administrador, deletar um

usuário, deletar um dispositivo, tarefas de criação e edição dentro do sistema. Também é utilizada para identificar se uma ação do software é proveniente de erro da aplicação ou de uma ação humana, para que possa ser corrigida com mais facilidade.

Ainda a respeito da tabela *usuario*, é visto que ela possui relação com as tabelas *alerta_fotovoltaiico* e *alerta_armazenamento*. Tais tabelas possuem a função de informar ao Core do sistema quais alertas cada usuário deseja receber por e-mail, sendo uma configuração específica para cada usuário. A Figura 46 representa a estrutura comentada.

Figura 46 – MER com as tabelas *alerta_fotovoltaiico* e *alerta_armazenamento*

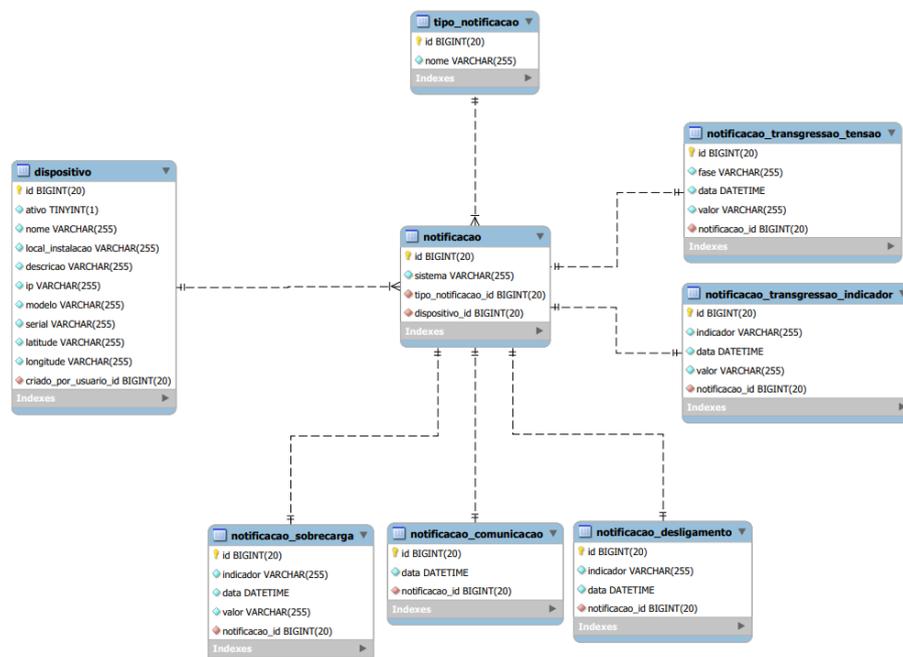


Fonte: O Autor.

Com base no MER, tem-se o relacionamento de “dispositivo” e “notificação”, em que a tabela “dispositivo” armazena dados de detalhes dos dispositivos cadastrados no sistema; e a tabela “notificação” armazena os dados referentes aos alertas e notificações salvas. A medida em que estes dispositivos estão em funcionamento, podem ocorrer diversos tipos de fenômenos como desligamento, sobrecarga, falta de comunicação etc. Sendo assim é de suma importância que estes dados sejam armazenados no banco de dados, com o objetivo de manter um histórico de ocorrência e de alertas enviados, e para tal, a tabela “dispositivo” possui um relacionamento de um para vários (1:N) com a tabela *notificacao*.

Dado que podem ocorrer diversos tipos de notificações, cada notificação possui seus próprios dados a serem salvos. Visando a otimização do banco de dados foi utilizado o conceito de generalização e especialização, em que a tabela notificação possui os atributos em comum a todas as notificações e cada tipo de notificação possui os seus atributos relevantes. A tabela “tipo_notificacao”, é responsável por informar a qual tabela uma notificação pertence, e o atributo “sistema”, informa se a notificação é proveniente do SIS2GER ou do SISGAE2B. A Figura 47 explicita a lógica descrita.

Figura 47 - MER do sistema de notificação no banco de dados.

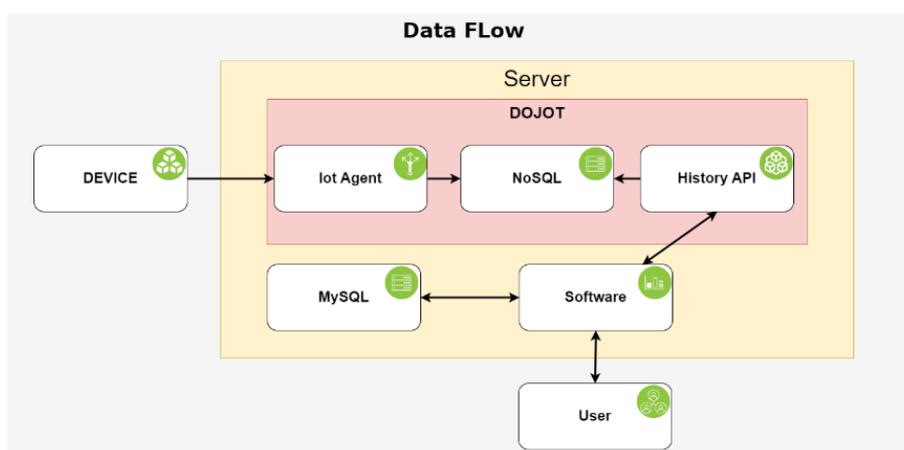


Fonte: O Autor.

Os softwares desenvolvidos possuem plena capacidade de gerenciar seus usuários e seus níveis de permissão, de modo que todos os sistemas podem definir um nível de acesso para o mesmo sistema. Em termos práticos, é possível que um usuário tenha perfil de administrador no software ao qual ele precisa gerenciar os dispositivos e usuários, bem como pode inserir os dados dos postos tarifários. Entretanto, em outros softwares, o usuário é apenas um usuário de perfil normal e sem capacidade de gerenciamento. Tal medida permite criar uma estrutura em que as permissões de cada usuário possam ser mais bem geridas e manter a segurança da aplicação.

A Figura 48 representa o diagrama final de armazenamento de dados por parte dos *softwares* desenvolvidos, utilizando tanto a DOJOT com um banco de dados não relacional (NoSQL) para os dados de telemetria, quanto um banco de dados relacional (SQL) para os dados referentes às aplicações desenvolvidas.

Figura 48 - Aquisição de dados.



Fonte: O Autor.

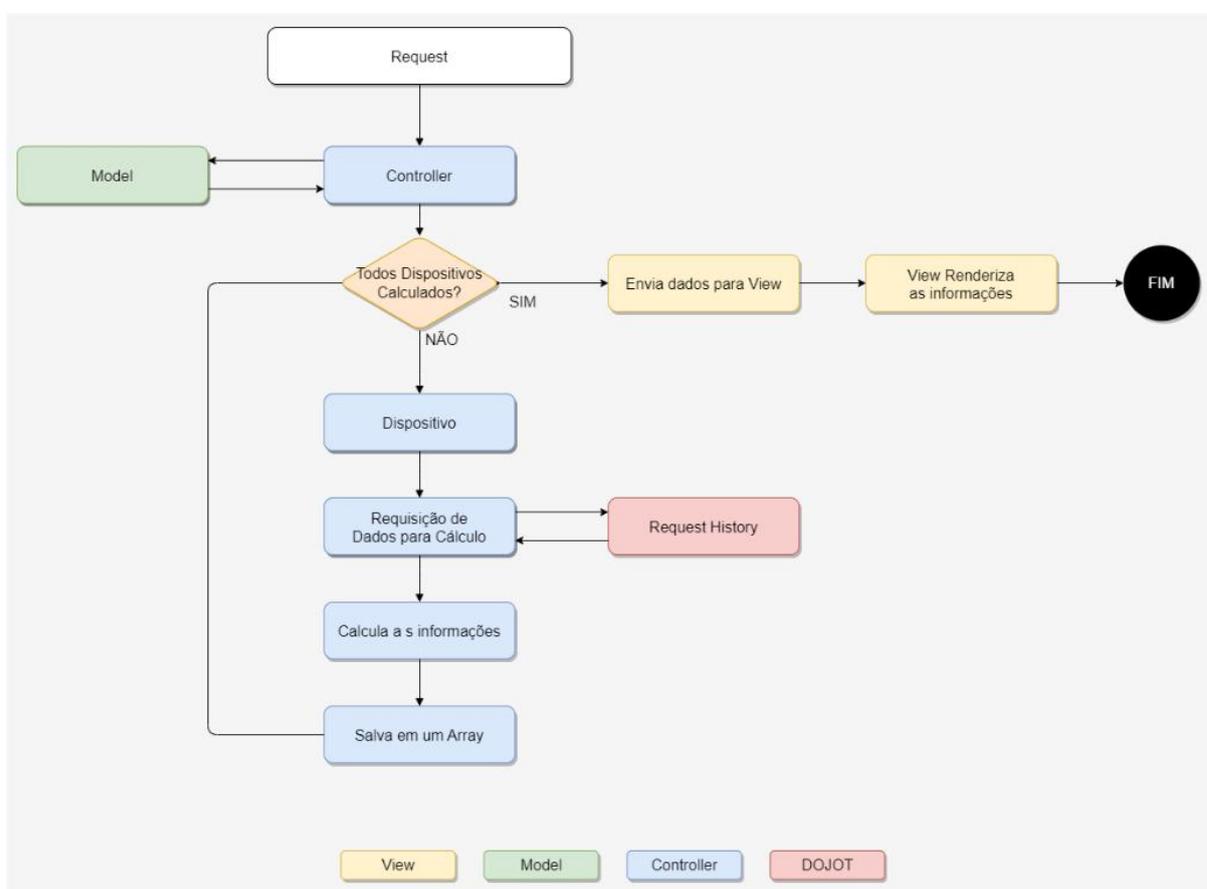
3.4 Implementação de Processamento Assíncrono

Em aplicações que utilizam IoT, em geral, o objetivo é ter dezenas de dispositivo operando de forma simultânea com dados salvos na DOJOT, juntamente com as informações referentes no banco de dados relacional. Para estes softwares, foram realizados testes de performance no sistema contendo diversos dispositivos para gerenciar e consolidar informações. Com o objetivo de verificar o comportamento do sistema, foram simulados testes contendo 128 dispositivos nos sistemas e após

isso foi calculado o tempo de resposta das funcionalidades nas páginas do software. Para a etapa de testes foram cadastrados os 128 dispositivos na DOJOT sem dados de medição, pois o objetivo foi determinar quanto tempo o software iria demorar para fazer o processo de comunicação com a DOJOT.

Com os testes realizados foi constatado que o tempo de resposta médio estava em torno de 5 minutos, considerado elevado demais. O tempo elevado se dá por conta do conjunto de cálculos para cada dispositivo, a ser exibido na tela ocorrer de forma serial antes dela ser exibida, como pode ser visto na Figura 49.

Figura 49 - Integralização de dados a serem exibidos na tela de Home de forma síncrona.

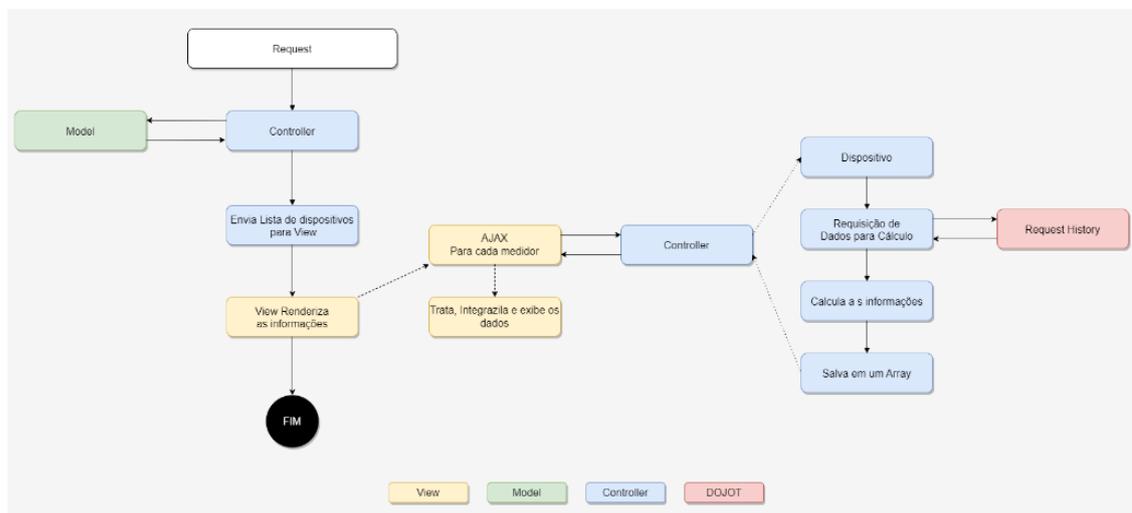


Fonte: O Autor.

Como solução, se fez necessário implementar modificações no fluxo do sistema que funcionava de forma síncrona consultando e calculando os dados necessários para cada um dos dispositivos. Como medida foi implementado um fluxo assíncrono capaz de fazer consultas *multithreading* em paralelo, fazendo com que o tempo de

resposta ficasse em 2 segundos na média. A Figura 50 representa o fluxo anterior da requisição.

Figura 50 - Integralização de dados a serem exibidos na tela de Home de forma assíncrona.



Fonte: O Autor.

3.4.1 Observabilidade

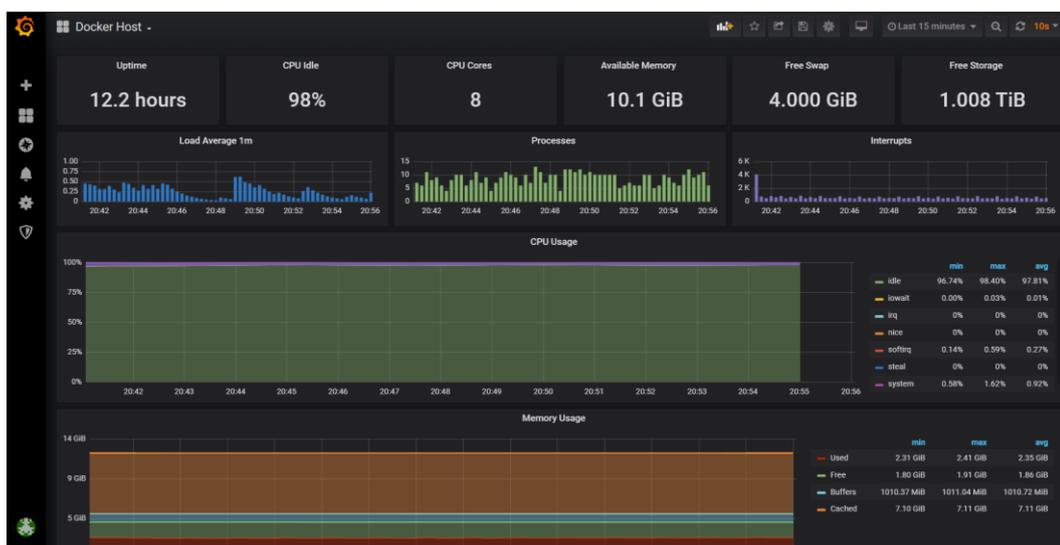
Uma das formas de garantir que um sistema implementado não possua problemas é por meio da observabilidade e coleta de métricas, tanto para garantir que o sistema não esteja gerando nenhum tipo de problema aos demais sistemas com quem ele se conecta, quanto para garantir que ele não está apresentando problemas tais como: estouro de memória, consumo excessivo de processamento de CPU ou alto consumo de outros recursos.

É possível utilizar diversos utilitários em Docker para realizar o serviço de monitoramento e coleta de métricas, pois esses dados ajudam a entender a saúde do sistema, garantindo que todos os softwares em execução possam operar adequadamente e que o servidor possua os recursos necessários, além de servir de estudo para a melhoria de forma contínua dos processos e tecnologias de desenvolvimento.

Dentro da suíte de serviços disponíveis no Docker Hub (sistema de registro gratuito e público do Docker), foram utilizados o cAdvisor (Container Advisor), responsável por monitorar as aplicações; o Node Exporter, como um serviço para o monitoramento do servidor; o Prometheus, utilizado para a persistência de dados de séries temporais; e o Grafana, responsável para a visualização dos dados

armazenados. Como resultado, é possível observar na Figura 51 um exemplo de informações relacionadas ao host do ambiente, sendo as mais importantes: Memória livre, CPU Disponível e Número de processos sendo executados.

Figura 51 - Grafana, tela de monitoramento do host do ambiente.



Fonte: O Autor.

Também é possível mensurar o comportamento dos serviços em execução por meio de Docker, como pode ser visto na Figura 52, onde podem ser vistos a porcentagem de CPU e a quantidade de memória utilizadas por cada serviço.

Figura 52 – Grafana, sistemas monitorados no servidor.



Fonte: O Autor.

3.5 Conclusão

Tem-se que foi abordado no presente capítulo pontos referentes ao desenvolvimento que foi realizada no trabalho, tendo como enfoque principal os tópicos acerca das tecnologias que foram utilizadas para a implementação dos sistemas, de modo que foi explanado sobre a arquitetura de serviços desenvolvida em Docker por meio do Docker Compose. Foi abordado sobre as comunicações dos softwares com a plataforma DOJOT, para aquisição dos dados de séries temporais, e o banco de dados da aplicação, para a aquisição de dados relacionados aos usuários e aplicações, bem como sobre o monitoramento do sistema por meio de gráficos do Grafana.

A metodologia demonstrada faz com que os *softwares* possam ser resilientes e ter ganhos de performance quando se faz necessário a utilização de diversas requisições, com isto, temos a implantação do sistema que será apresentado no capítulo 4.

CAPÍTULO 4: IMPLEMENTAÇÃO DO SOFTWARE

4.1 Introdução

Nesse capítulo serão abordados os resultados deste trabalho, de modo em que serão exibidas as telas que foram desenvolvidas e a função de cada tela nos softwares para gestão da geração de energia fotovoltaica e para o armazenamento de energia. Desta forma, será comentado sobre as formas de autenticação nos sistemas, os Menus disponíveis em cada um dos softwares, os pontos de segurança e a configuração que foram implementados.

4.2 Telas dos sistemas desenvolvidos

Durante o ciclo de desenvolvimento dos sistemas, foram adotados estilos de *design de software* na prototipação das telas que iriam compor os sistemas, tanto para garantir a identidade visual de cores e estilos entre os *softwares*, quanto para padronização das telas e maior facilidade por parte do usuário na utilização das ferramentas. Por esse motivo os sistemas possuem telas comuns na identidade visual, podendo também ser possível implementar separadamente cada aplicação de forma modular em diferentes locais, se for o desejado.

4.2.1 Pontos comuns para os softwares

Para respeitar o padrão do *software* desenvolvido e a tipologia gráfica de estilos, cores e fontes, foram criadas telas que estão disponíveis nos 2 *softwares* de forma idênticas, sendo assim, tem-se: Tela de Login, Tela de Cadastro e Tela de Perfil. Tendo em vista a similaridade, serão exibidas as telas do Sistema de Gestão do sistema de armazenamento de energia elétrica (SISGAE2B) para a presente seção.

Tela de Login

O login é a ação principal em termos de garantia de permissão de acesso aos recursos do sistema, sendo utilizado para facilitar o login e o cadastro no sistema. O visual foi desenvolvido para ser minimalista e intuitivo. Na tela de login é possível

verificar as opções de inserção de dados de acesso (e-mail e senha), opção de login usando o Gmail, opção de cadastro e opção de redefinir a senha. Essas informações podem ser vistas na Figura 53.

Figura 53 - Tela de Login.

A imagem mostra a tela de login do sistema SIMA. O cabeçalho verde contém o título "SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA (SIGAE2B)". O formulário centralizado, intitulado "FAZER LOGIN", apresenta o logo SIMA e campos para "Email" (com uma mensagem de erro "Formato inválido") e "senha". Abaixo dos campos, há um link "Esqueceu sua senha?", um botão verde "FAZER LOGIN", um botão branco "GOOGLE LOGIN" e um botão amarelo "CADASTRE-SE". O rodapé do formulário indica "Copyright © 2023 CEAMAZON | Equipe SIMA". À esquerda, um menu lateral verde contém ícones para perfil e home.

Fonte: O Autor.

Tela de Cadastro

Caso o usuário não possua um cadastro para fazer login no sistema, é possível solicitar acesso de duas formas: Clicando no botão de login utilizando o Google, o sistema identificará que não há usuários cadastrados no banco de dados, logo, o *software* coletará as informações disponíveis no serviço de autenticação do Google (Imagem de Perfil, e-mail e nome) e redirecionará para a tela de cadastro para concluir a solicitação; ou clicando no botão de “cadastra-se” na tela de Login e inserindo manualmente os dados solicitados.

O menu lateral esquerdo apresenta apenas 2 opções, a tela de Login e a tela de Sobre, sendo essas opções de acesso público. A Figura 54 representa a tela de cadastro de Usuários do sistema.

Figura 54- Tela de Cadastro de Usuários.

SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA (SISGAE2B)

CADASTRO

SIMA

Nome Sobrenome

Email CPF

Formato inválido.

..... Confirmar

As senhas não conferem

CADASTRAR-SE

Copyright © 2023
CEAMAZON | Equipe SIMA

Fonte: O Autor.

A segurança do sistema na tela de cadastro e login, foi desenvolvido para evitar que robôs possam utilizar o formulário para submeter solicitações de cadastros automáticas e obter acesso indevido. A seguir tem-se três funções de segurança desenvolvidas:

- O envio de um e-mail para o e-mail cadastrado, de modo que para o usuário ser considerado desbloqueado na plataforma, é preciso acessar sua caixa de e-mails e confirmar o recebimento do e-mail, como pode ser visto na Figura 55.

Figura 55 - E-mail enviado para confirmar ao novo usuário.



Fonte: O Autor.

- A confirmação por meio de um usuário ativo na plataforma e com permissões suficientes para conceder acesso. Logo, para que um usuário possa acessar o sistema, um usuário ativo e com acesso de administrador precisa autorizar esse novo usuário.
- A existência de logs para auditoria em ações críticas, como o log de concessão de permissão na plataforma e login, de modo que, quando existe a confirmação de um novo usuário nos softwares, é possível saber quem concedeu o acesso e quando foi realizado, para então revogar ambos os acessos, caso seja necessário.

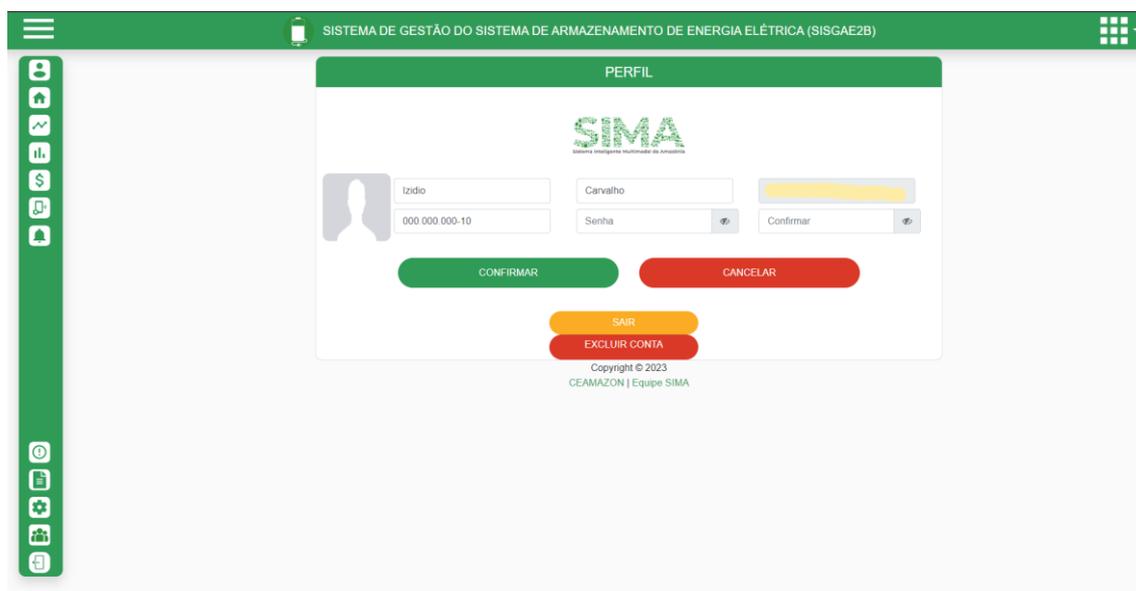
Após a realização da autenticação na plataforma é possível ter acesso ao recurso do *software*. Os recursos da plataforma possuem validações para visualização e acesso, tanto no *frontend*, para não exibir ferramentas que o usuário autenticado não pode acessar, quanto para o *backend*, para não processar requisições dado o nível de permissão de usuário.

Tela de Perfil

Uma vez que o usuário fez cadastro e posteriormente o login no sistema, é possível gerenciar as suas informações de cadastro, com exceção do e-mail, pois esse foi utilizado para confirmar a identidade do usuário e foi tido como válido no processo de cadastro. A

Figura 56 representa a tela de perfil.

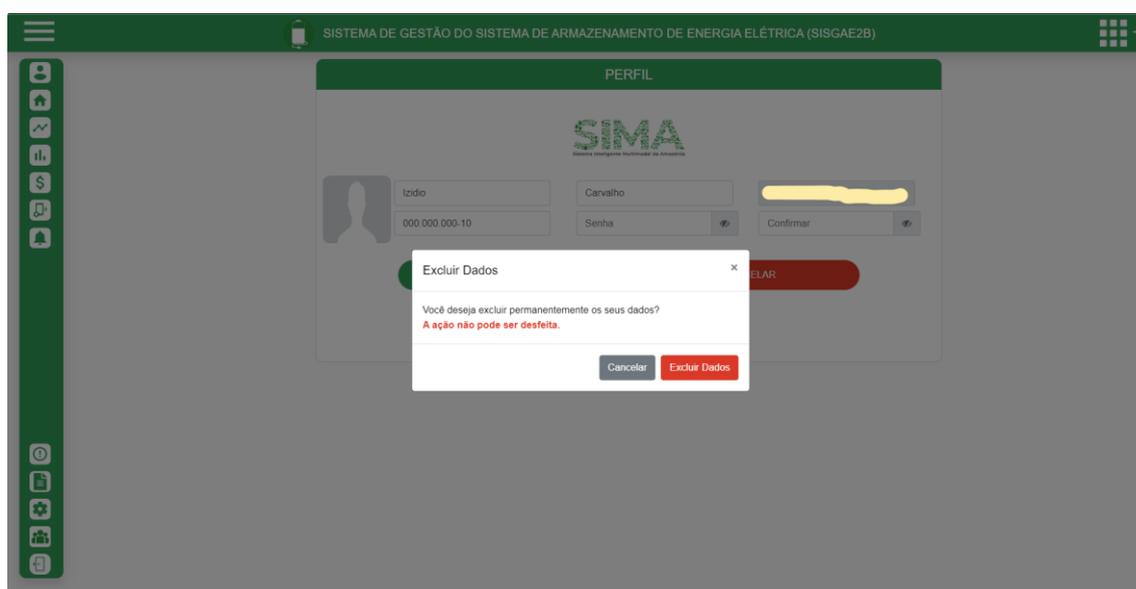
Figura 56 - Tela de Perfil de Usuário.



Fonte: O Autor.

Dentre as opções da tela de perfil, cabe destacar a opção de exclusão da conta. Uma vez que o usuário é o dono dos seus dados, segundo a LGPD (Lei Geral de Proteção de Dados), o usuário tem que ser capaz de solicitar o seu direito de esquecimento, em outra abordagem, o usuário tem que ser capaz de se deletar do sistema quando assim desejar. Para evitar que ações sejam tomadas por um clique acidental, a Figura 57 representa uma imagem com o alerta que o usuário receberá quando solicitar o seu direito de esquecimento.

Figura 57 - Alerta de exclusão de dados do usuário na plataforma.



Fonte: O Autor.

4.2.2 Sistema de Gestão da Geração de Energia Renovável – SIS2GER

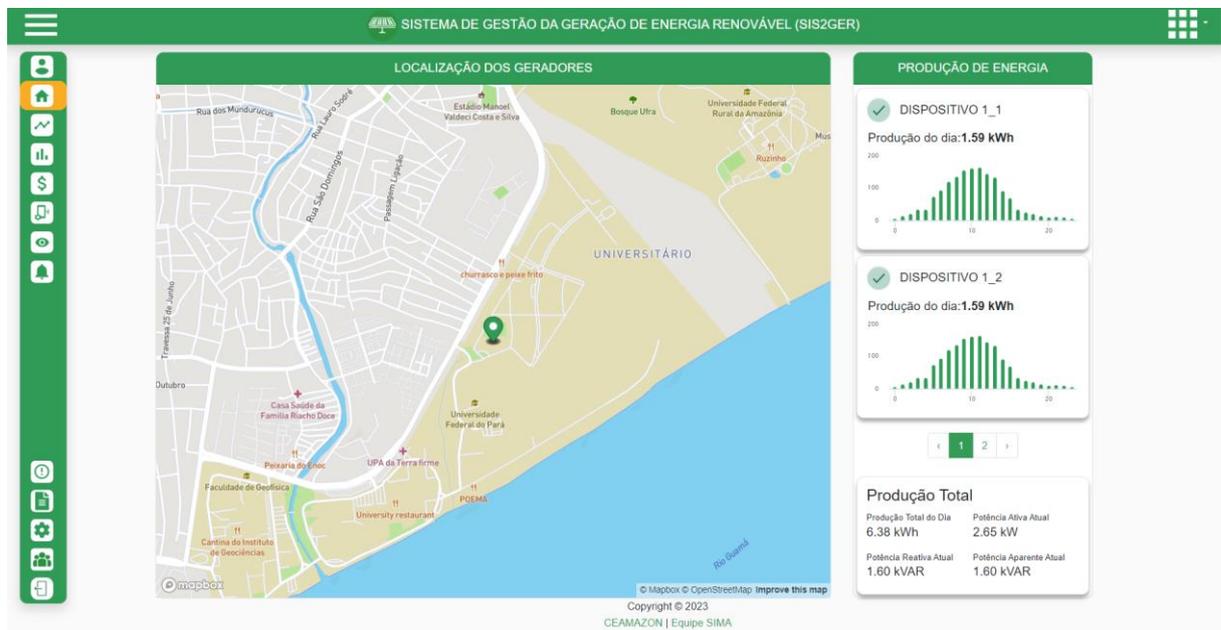
O SIS2GER possui como objetivo o gerenciamento de sistemas de geração de energia renovável, em especial os sistemas fotovoltaicos, de modo que seja possível realizar o cadastro de dispositivos tanto no banco de dados relacional do sistema, quanto na DOJOT, permitindo assim o seu completo gerenciamento. Nesse tópico tem-se as telas e funcionalidades do Sistema de Gestão da Geração de Energia Renovável.

Tela de Home

A tela de home é a primeira tela acessada pelo usuário após o login na aplicação, sendo responsável por exibir um panorama geral do estado dos sistemas fotovoltaicos gerenciados pelo *software*.

Para facilitar a identificação de cada unidade geradora de energia, é possível observar o mapa no centro da tela, com a geolocalização de cada um dos sistemas gerenciados. Na medida em que novos medidores sejam inseridos na rede, o mapa é capaz de ser readequado automaticamente para que todos fiquem visíveis. À direita do mapa é possível verificar a geração do dia por cada um dos dispositivos, bem como a quantidade de energia horária produzida. Na mesma tela, também é possível verificar o panorama geral da geração de todos os dispositivos no sistema (Figura 58).

Figura 58 - Tela de Home do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.



Fonte: O Autor.

Menus

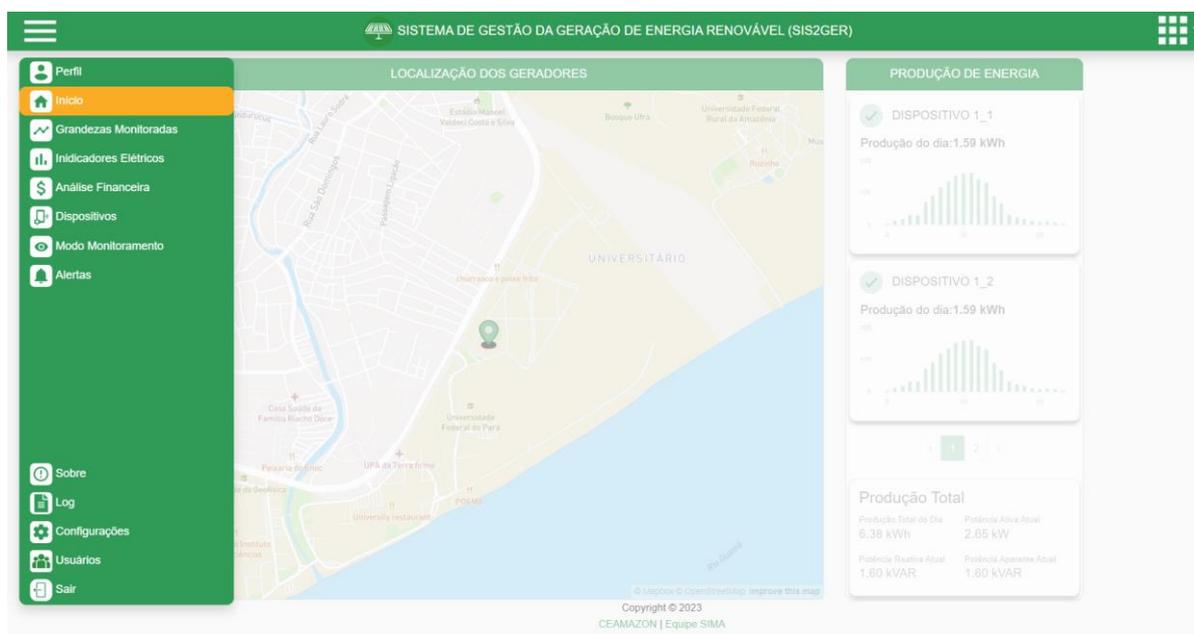
O *software* desenvolvido é constituído por dois Menus principais: o Menu Lateral e o Menu de Aplicações. O Menu Lateral fica disposto na parte esquerda do SIS2GER e as telas ficam disponíveis de acordo com o nível do usuário autenticado. A plataforma possui dois níveis de usuário, sendo o Usuário Normal, capaz de acessar o sistema e fazer análises, e o Usuário Administrador, capaz de fazer tudo que o normal faz e ainda gerenciar a plataforma. Dado o exposto, para o SIS2GER tem-se as seguintes telas no Menu Lateral esquerdo e seu nível de permissão:

- Perfil: Usuário Normal;
- Início: Usuário Normal;
- Grandezas Monitoradas: Usuário Normal;
- Indicadores Elétricos: Usuário Normal;
- Análise Financeira: Usuário Normal;

- Dispositivos: Usuário Administrador;
- Modo Monitoramento: Usuário Administrador;
- Sobre: Usuário Normal;
- Log: Usuário Administrador;
- Configurações: Usuário Administrador;
- Usuário: Usuário Administrador;

Uma vez que as informações a respeito do nível de permissão de cada opção do Menu Lateral foram dispostas, a Figura 59 representa o menu lateral esquerdo do sistema mencionado.

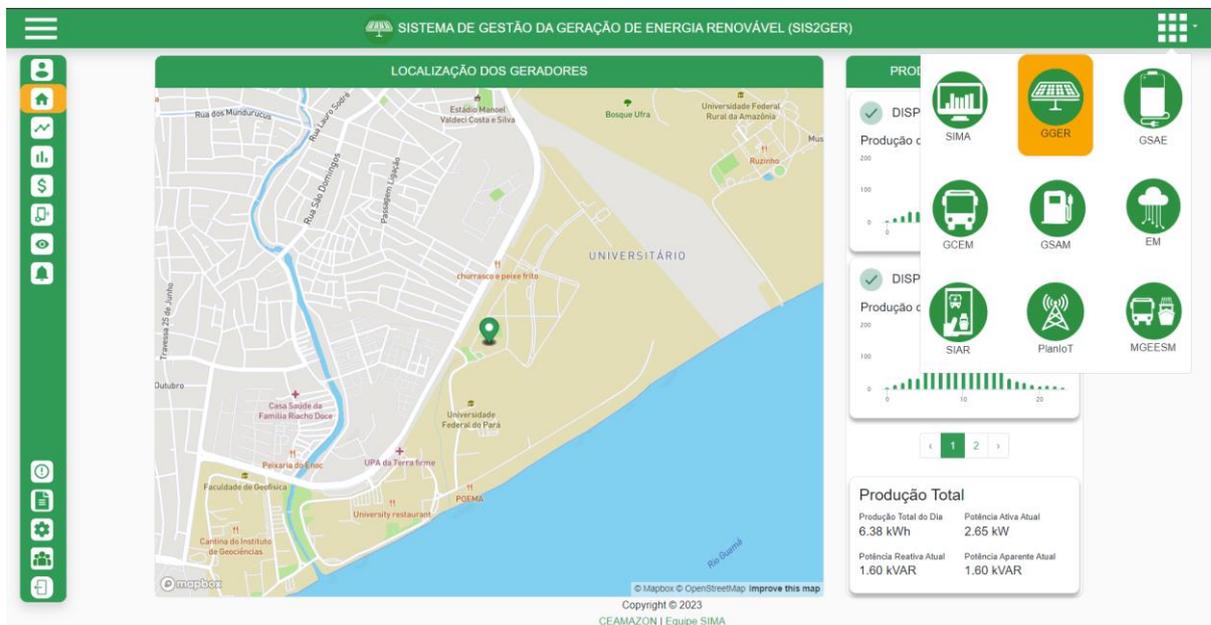
Figura 59 - Menu Lateral Esquerdo do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.



Fonte: O Autor.

Dado o que foi visto a respeito do Menu Lateral do sistema, tem-se também a possibilidade de inclusão de acesso para outras aplicações internas do *software*, de forma a facilitar a transição entre aplicações. Para tal, o sistema conta com um Menu de Aplicações no canto superior direito, de modo que ao clicar em algum ícone, o usuário é redirecionado em uma nova aba. A Figura 60 representa o Menu de Aplicações.

Figura 60 - Menu de aplicações do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER



Fonte: O Autor.

Tela de Grandezas

Como um dos principais objetivos do *software*, são apresentados o comportamento de grandezas elétricas provenientes dos sistemas fotovoltaicos monitorados. O SIS2GER possui três características em relação a visualização dos gráficos desenvolvidos:

- Gráficos nos dois Eixos: O sistema possui flexibilidade para exibir em um mesmo gráfico diversas grandezas, de acordo com as escolhas do usuário, de modo que o SIS2GER é capaz de agrupar as unidades selecionadas nos eixos Y (esquerdo e direito) do gráfico, permitindo que o usuário selecione até duas unidades diferentes (Ex: tensão e corrente) e diversas grandezas e medidores. Esta opção permite a criação de correlações e gráficos mais complexos para uma análise mais profunda do sistema de geração de energia.
- Integralização: Dentro do sistema é possível ter o acompanhamento pontual das grandezas na forma exata em que foram medidas (as medições que

estão salvas na DOJOT), entretanto, também é possível integralizar os valores das grandezas para apresentar as medições por dia, mês ou ano.

- c) Tipos de Gráficos: O sistema permite que o usuário possa modificar a forma com que as grandezas são exibidas, seja as grandezas pontuais ou integralizadas. Logo, se o usuário quiser ver um determinado gráfico com diversas grandezas em um gráfico de linhas ou em um gráfico de barras, será possível ajustar a exibição de acordo com o seu interesse.

Dado ao que foi comentado, a Figura 61 representa a tela de grandezas monitoradas.

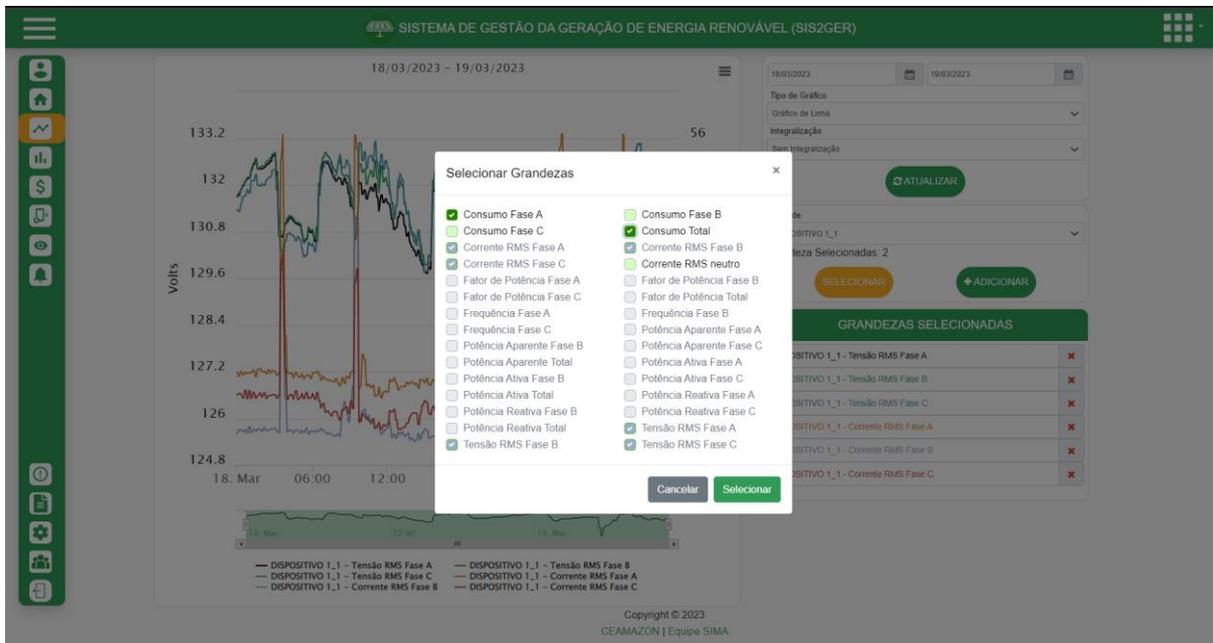
Figura 61 - Tela de grandezas monitoradas do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.



Fonte: O Autor.

Ainda a respeito da tela de Monitoramento de Grandezas, o sistema permite selecionar ao mesmo tempo as grandezas a serem monitoradas no gráfico. Com isso, é possível selecionar um determinado dispositivo fotovoltaico e adicionar todas as grandezas de análise necessárias em um único gráfico. A Figura 62 representa a seleção de grandezas.

Figura 62 - Opção de seleção grandezas no SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.



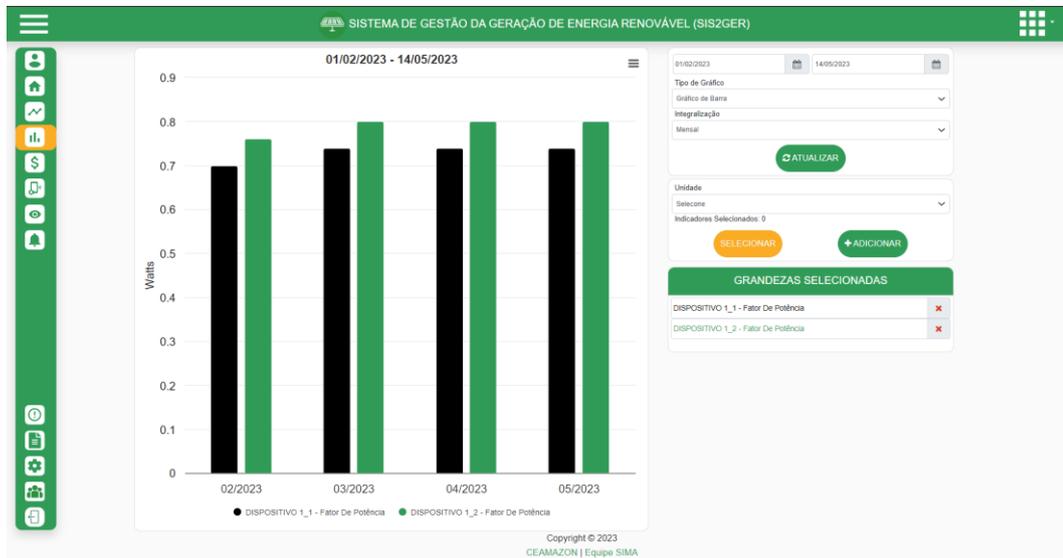
Fonte: O Autor.

Tela de Indicadores

De modo similar ao comportamento ocorrido na tela de Monitoramento de Grandezas, a tela de Indicadores Elétricos permite ao usuário plotar informações usando os dois eixos Y do gráfico, integralizar as grandezas selecionadas e exibir múltiplos indicadores.

Os indicadores elétricos são responsáveis por exibir indicadores de qualidade de energia e indicadores ambientais. Os valores destes indicadores não são medidos diretamente das instalações elétricas, são calculados dentro de um intervalo de 1 mês para cada Sistema Fotovoltaico. A Figura 63 representa a tela de indicadores elétricos.

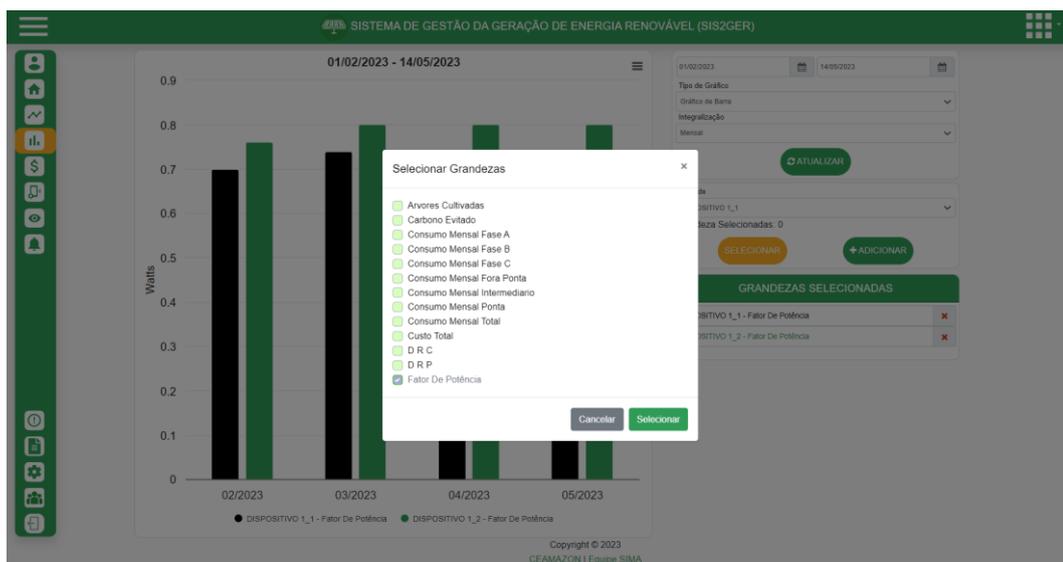
Figura 63 - Tela de Indicadores Elétricos do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.



Fonte: O Autor.

A tela de indicadores elétricos possui os indicadores calculados para cada dispositivo, conforme podem ser vistos na Figura 64.

Figura 64 – Opção de seleção dos indicadores elétricos do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.

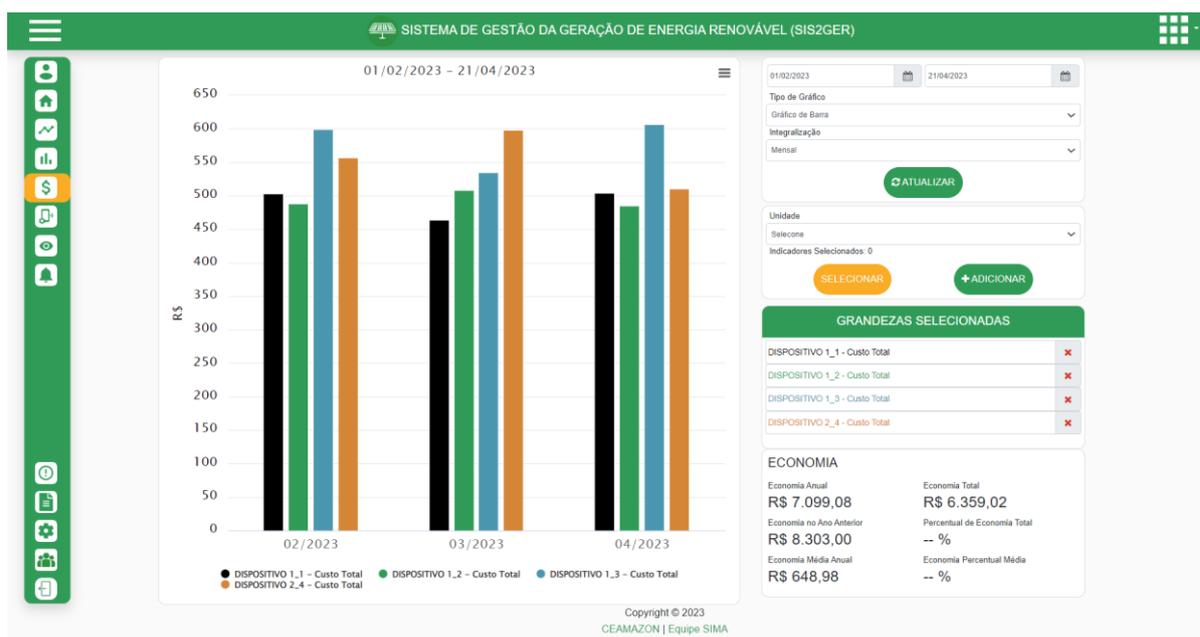


Fonte: O Autor.

Tela de Análise Financeira

A tela de Análise financeira visa exibir o indicador de Custo Total, sendo que os valores apresentados são calculados mensalmente e salvo na DOJOT, para que o *software* web possa consultar a informação para o usuário. O objetivo dessa tela é separar o Custo Total para facilitar a visualização deste indicador, além de exibir dados adicionais de Economia, tal como a Economia Anual, seja do ano corrente ou qualquer outro período selecionado. A Figura 65 representa a tela de Análise Financeira.

Figura 65 - Tela de Análise Financeira do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.



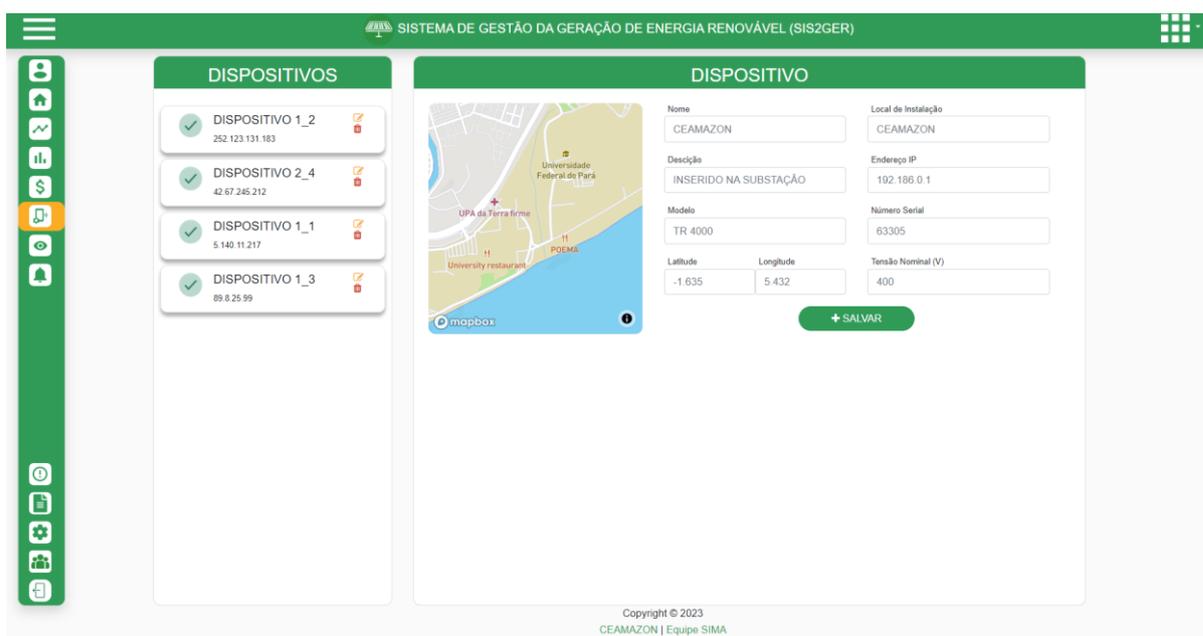
Fonte: O Autor.

Tela de Gerenciamento de Dispositivos

Para que seja possível exibir as informações gráficas, o sistema precisa conhecer os sistemas fotovoltaicos. O *software* possui uma tela dedicada ao gerenciamento de dispositivos e por meio desta é possível visualizar dados de

configuração, além de ver o status de funcionamento, ou seja, se está comunicando ou se houve alguma perda de comunicação. Por meio dessa tela, também é possível cadastrar um novo dispositivo, informando os seus dados e sua geolocalização para que possa ser exibido no mapa da tela inicial, conforme pode ser visto na Figura 66.

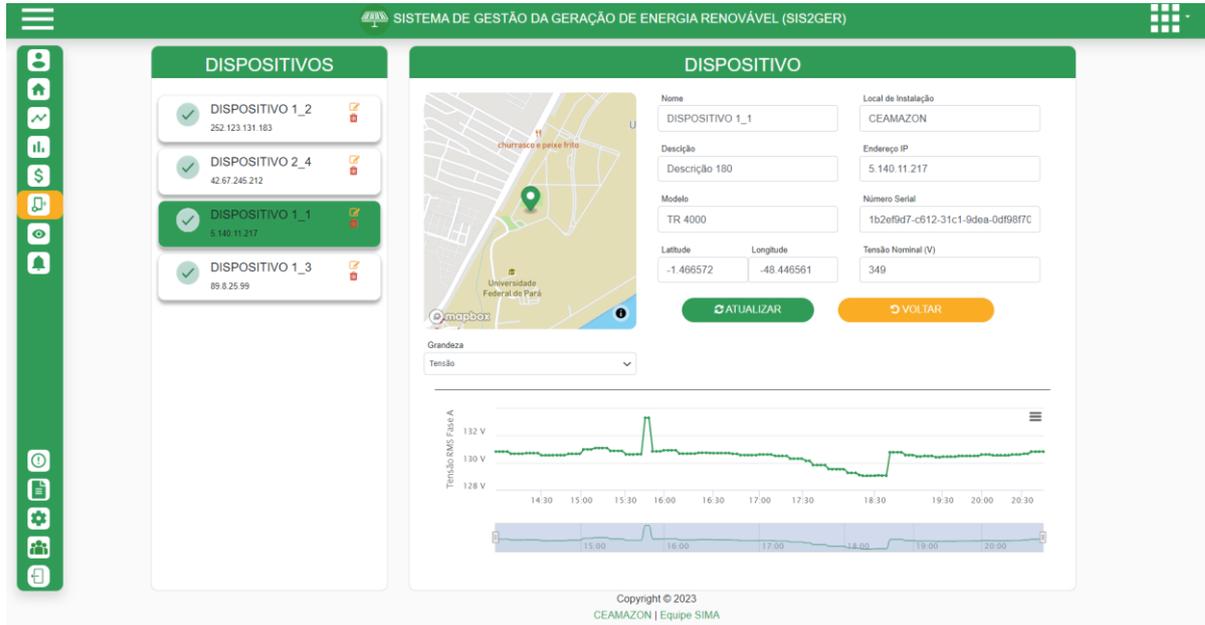
Figura 66 - Tela de gerenciamento de dispositivos do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.



Fonte: O Autor.

Além da opção de cadastrar, também é possível editar ou deletar os dispositivos com os devidos logs de auditoria, logo, é possível saber quem efetuou a alteração e o que foi mudado nas telas e configurações do medidor. Durante o processo de edição de um medidor, é possível conferir os valores de tensão e corrente, facilitando a análise da ocorrência, sem que seja preciso voltar para a tela de Monitoramento de Grandezas e realizar uma análise completa em busca de saber qual medidor está com problemas. A Figura 67 representa o exposto.

Figura 67 - Tela de edição de dispositivos do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.

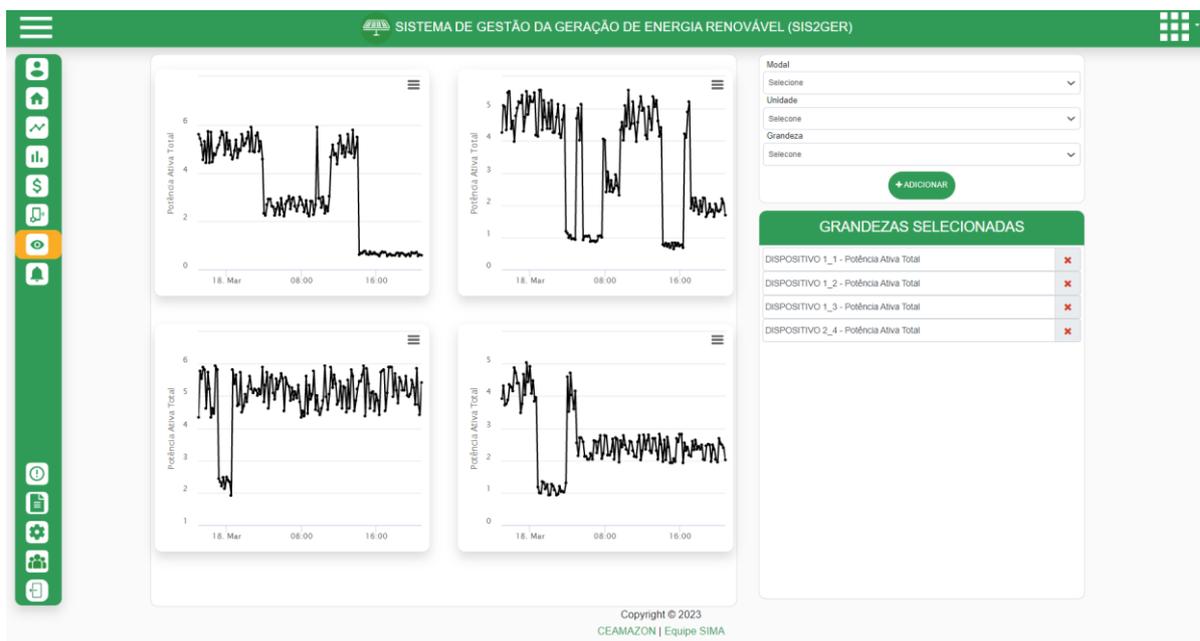


Fonte: O Autor.

Modo monitoramento

A tela de monitoramento é utilizada para o acompanhamento constante das medições do sistema. Os gráficos são atualizados automaticamente de forma simultânea a cada 10 minutos, podendo ser inserido até 6 gráficos de grandezas para o acompanhamento em tempo real. A Figura 68 representa a descrição.

Figura 68 - Tela de Monitoramento do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.



Fonte: O Autor.

Tela de Gerenciamento de Alertas

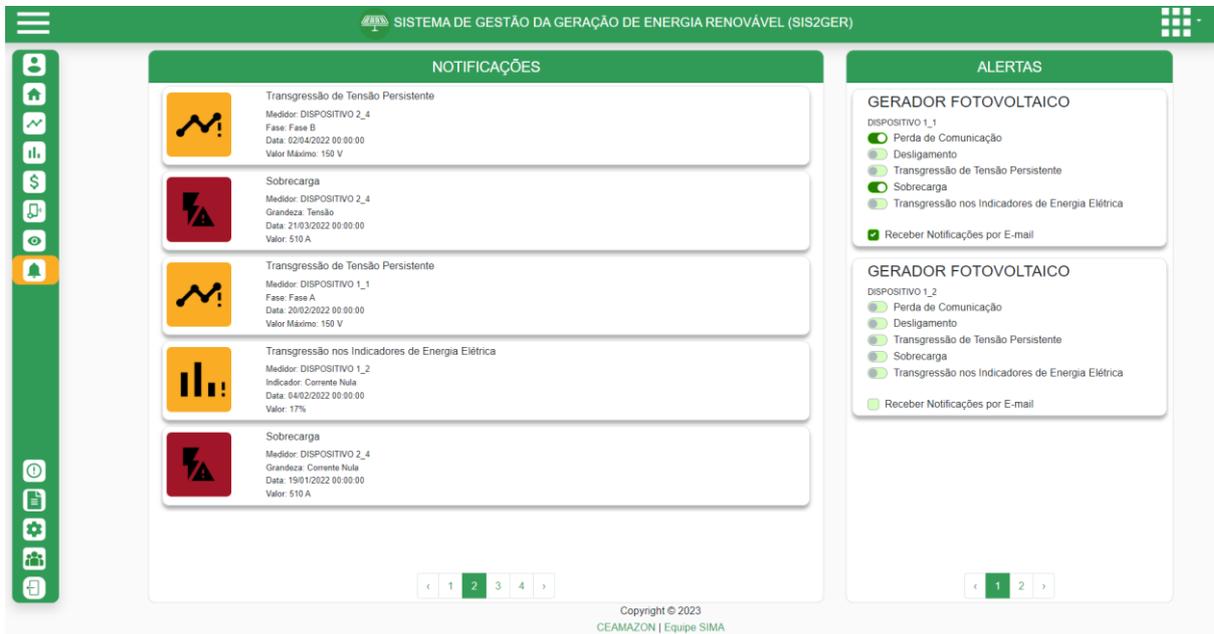
Além das possibilidades de monitoramento das grandezas de forma ativa por meio de uma tela, também é possível visualizar um conjunto de alertas pré-estabelecidos, por meio da tela de Alertas.

O sistema possui a funcionalidade de envio de alertas quando o usuário se cadastra para recebê-los via e-mail, sendo possível escolher os tipos de alertas existentes e para quais sistemas fotovoltaicos o alerta será configurado. Essa funcionalidade permite a existência de pessoas dedicadas para a manutenção e acompanhamento em cada um dos sistemas fotovoltaicos, para que somente esses recebam os alertas. Dado o exposto, os tipos de alertas disponíveis são:

- Perda de comunicação;
- Desligamento;
- Transgressão de Tensão Persistente;
- Sobrecarga;
- Transgressão nos Indicadores de Energia Elétrica.

Essa tela pode ser vista na Figura 69.

Figura 69 - Tela de Alertas do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL–SIS2GER.



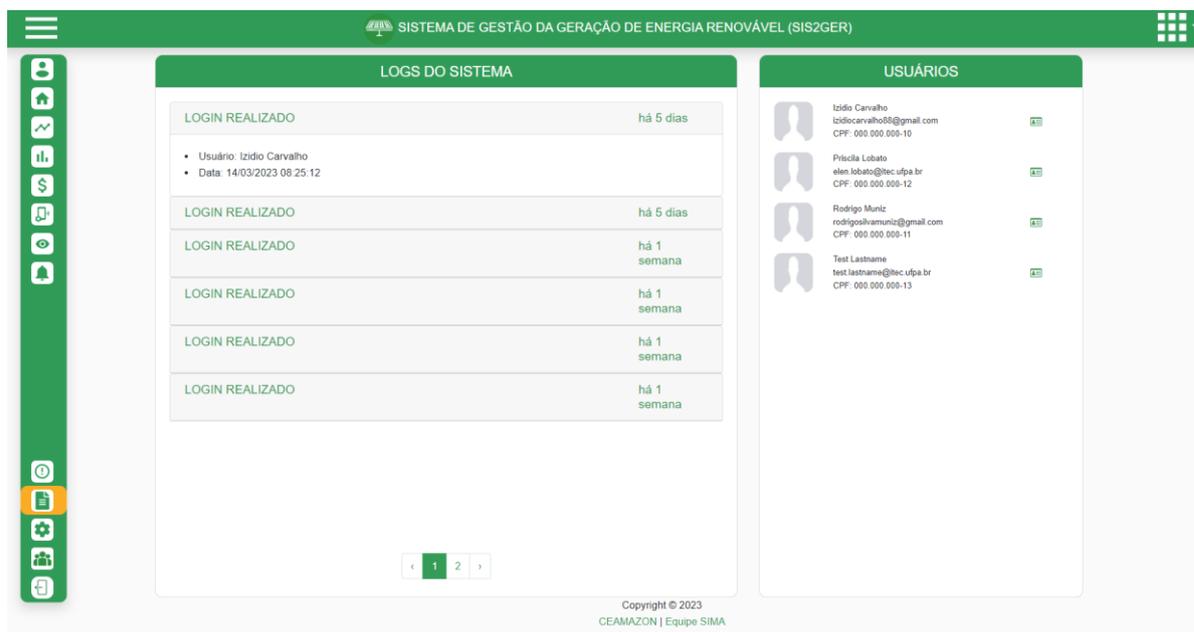
Fonte: O Autor.

Tela de Gerenciamento de Logs

Além de pontos de segurança relacionados à ataques externos, também existem medidas para evitar ataques internos ou atividades danosas que podem vir a ocorrer e, dentre estas, a principal forma de proteção é a utilização de logs de auditoria em diversas rotas da aplicação, fazendo com que seja possível conhecer todos os usuários que acessam o sistema, quando acessam e editam as configurações de atributos realizadas na plataforma, medidores e usuários.

A tela de Logs possui a responsabilidade de ser o ponto central das informações de auditoria do *software*, de modo que é possível saber todo o histórico de mudanças no sistema, podendo, inclusive, filtrar por um usuário específico, conforme pode ser visto na Figura 70.

Figura 70 - Tela de Logs do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL–SIS2GER.



Fonte: O Autor.

Tela de Configurações

As configurações do sistema possuem como característica a capacidade de auxiliar o gestor no acompanhamento de indicadores econômicos. Para que possa ser possível calcular os dados econômicos, fez-se necessário implementar uma tela para configurar os postos tarifários do sistema, o que nada mais são períodos (ou janelas) em que o sistema irá levar em consideração as informações do intervalo descrito para calcular os dados. A Figura 71 demonstra a tela descrita.

Figura 71 - Tela de Configurações do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL– SIS2GER.

SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA FOTOVOLTAICA (SGGEF)

CONFIGURAÇÕES

POSTOS TARIFÁRIOS

Posto: Conventional

Demanda Ponta (kW): 5000

Demanda Fora Ponta (kW): 200

Preço da Demanda Ponta (R\$/kW): 2.00

Preço da Demanda Ponta (R\$/kW): 2.00

Preço do Consumo Ponta (R\$/Wh): 2.00

Preço do Consumo Fora Ponta (R\$/Wh): 2.00

Tolerância de Ultrapassagem (%): 0.00%

Penalização de Ultrapassagem (%): 0.00%

14/03/2023

14/03/2023

Atualmente Vigente
 Valor Padrão

Histórico

\$ 01/01/2019 - 31/07/2019	<input checked="" type="checkbox"/>	<input type="checkbox"/>
\$ 01/01/2019 - 31/07/2019	<input checked="" type="checkbox"/>	<input type="checkbox"/>
\$ 01/08/2019 - 31/12/2019	<input checked="" type="checkbox"/>	<input type="checkbox"/>
\$ 01/08/2019 - 31/12/2019	<input checked="" type="checkbox"/>	<input type="checkbox"/>
\$ 01/01/2020 - 31/07/2020	<input checked="" type="checkbox"/>	<input type="checkbox"/>
\$ 01/01/2020 - 31/07/2020	<input checked="" type="checkbox"/>	<input type="checkbox"/>
\$ 01/08/2020 - 31/12/2020	<input checked="" type="checkbox"/>	<input type="checkbox"/>
\$ 01/08/2020 - 31/12/2020	<input checked="" type="checkbox"/>	<input type="checkbox"/>

+ SALVAR

Copyright © 2023
CEAMAZON | Equipe SIMA

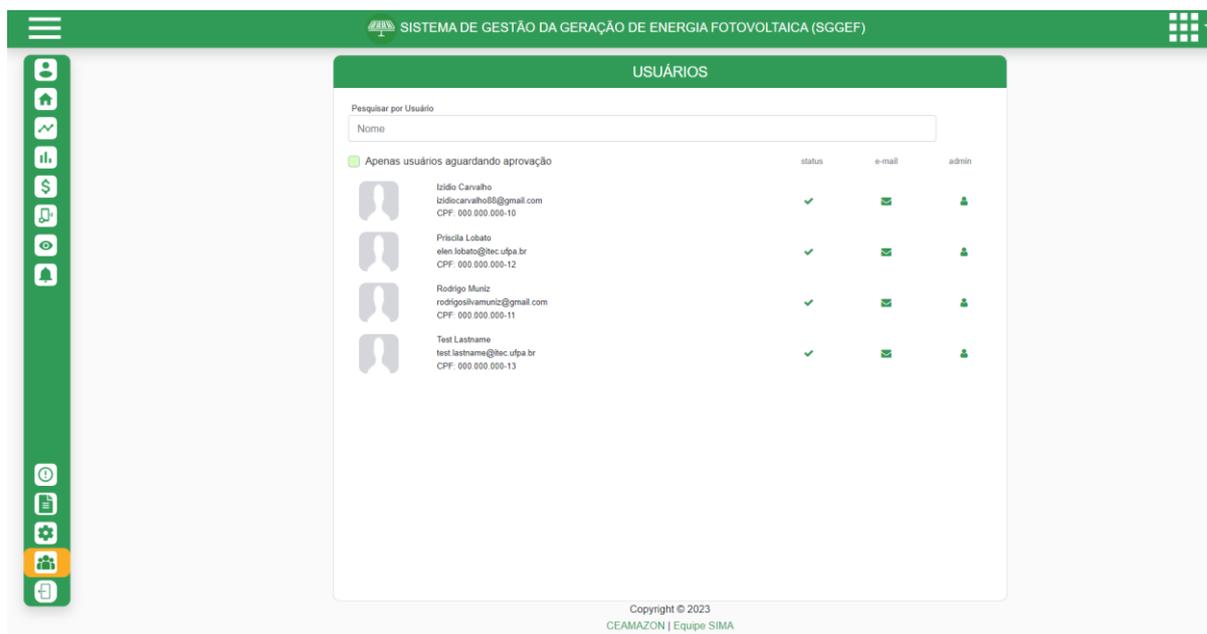
Fonte: O Autor.

Tela de Gerenciamento de Usuários

Para que seja possível considerar um usuário do sistema como Ativo, é preciso concluir dois passos importantes além do cadastro, sendo: a confirmação do e-mail cadastrado e o recebimento da confirmação de um usuário Administrador ativo no sistema. O cumprimento de apenas um dos requisitos na plataforma não habilita o usuário a fazer login.

Uma vez que o usuário com perfil de administrador teve acesso à SIS2GER, ele pode ver o menu de Usuários da plataforma e filtrar os usuários tanto por nome, quanto por usuários que estão aguardando a aprovação. Além disto, também é possível bloquear e desbloquear o usuário, mudar o perfil de um usuário de Administrador para Normal e o inverso. Esses pontos podem ser vistos na Figura 72.

Figura 72 - Tela de Usuários do SISTEMA DE GESTÃO DA GERAÇÃO DE ENERGIA RENOVÁVEL–SIS2GER.



Fonte: O Autor.

4.2.3 Sistema de Gestão do Sistema de Armazenamento de Energia Elétrica – SISGAE2B

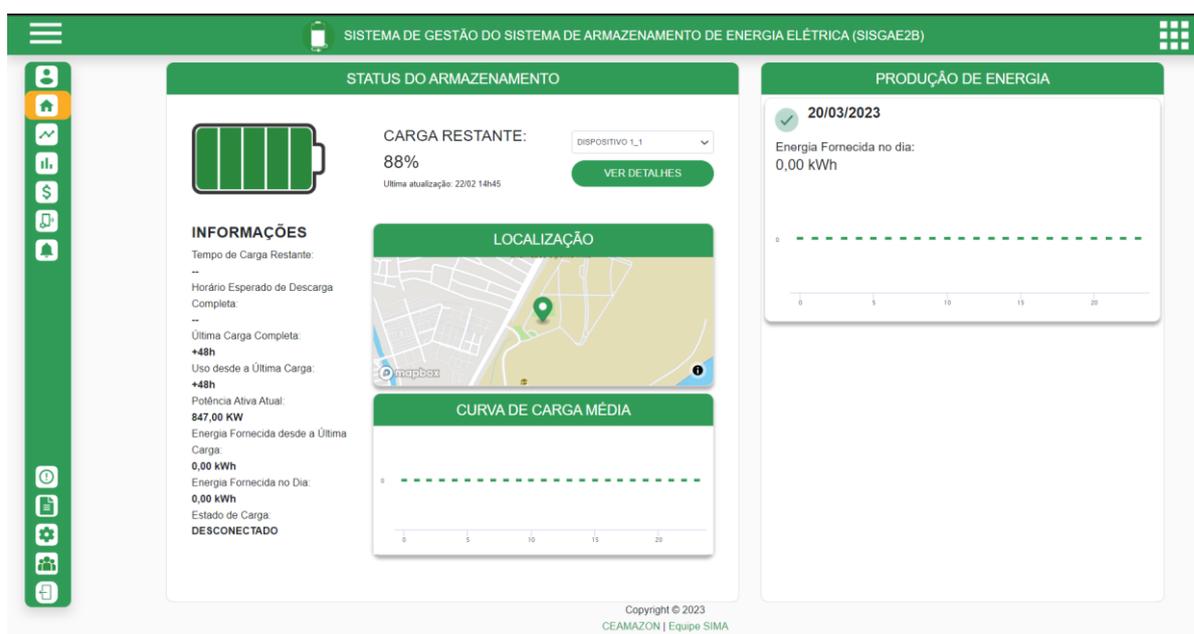
O SISGAE2B é responsável por gerenciar os sistemas de armazenamento de energia elétrica, de modo que seja possível cadastrar e realizar o papel de gestão com o acompanhando de grandezas e indicadores.

Tela de Home

Tendo como objetivo o monitoramento de sistemas de armazenamento, a tela de Home do SISGAE2B foi desenvolvida com função de monitorar diariamente os sistemas de armazenamento. Por meio da tela de Home é possível verificar a carga de energia fornecida por cada um dos sistemas de armazenamento ao longo dia, onde tem-se a parte de Produção de Energia (lado direito da tela). Também é possível verificar as demais características individuais de um sistema de armazenamento, onde tem-se o Status do Armazenamento (centro da tela). Caso exista mais de um sistema de armazenamento sendo gerenciado pelo SISGAE2B o mapa descrito como Localização exibirá a geolocalização de todos os

dispositivos existente e o gráfico de barras, chamado de Curva de Carga Média, exibirá o somatório de cada uma das cargas ao longo do dia. A Figura 73 representa a tela descrita.

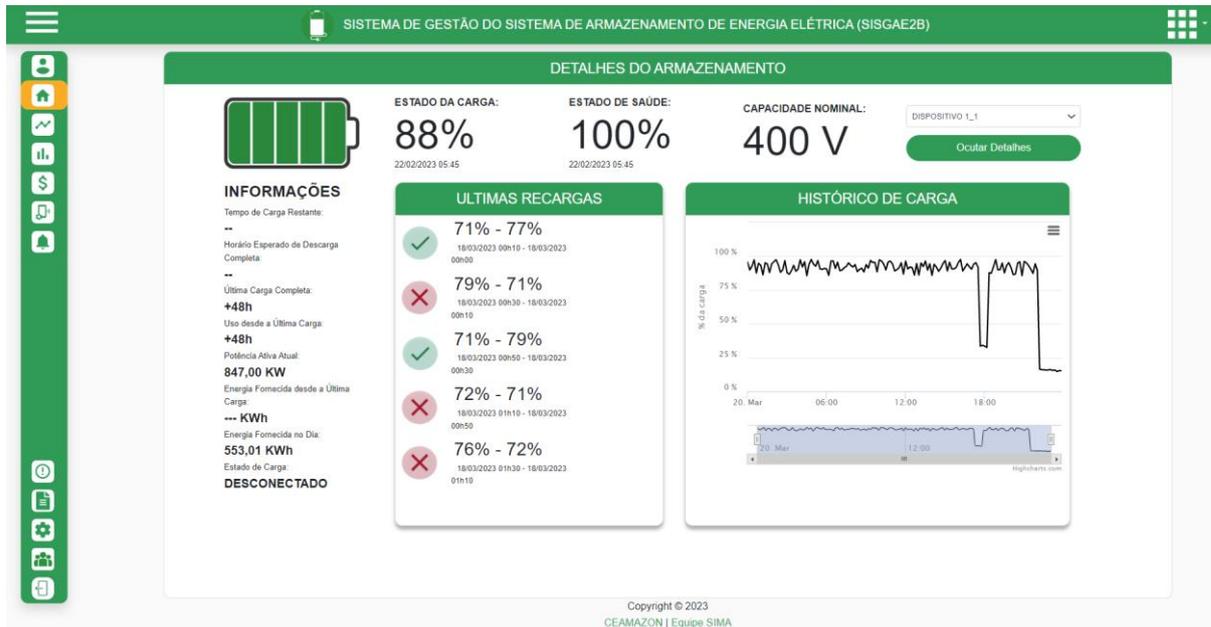
Figura 73- Tela de Home SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.



Fonte: O Autor.

A tela Home do SISGEE2B possui uma opção específica para visualizar os detalhes de sistema de armazenamento selecionado, devido a necessidade de acompanhamento da carga restante e os ciclos de carga da bateria. Desse modo, as informações de localização e curva de carga são removidas da tela e o sistema passa a exibir os últimos intervalos de cargas e de descargas do sistema. Também é possível visualizar o histórico da porcentagem da bateria ponto a ponto, possibilitando acompanhar os intervalos de carga e descarga. A Figura 74 representa essa tela.

Figura 74- Tela de Detalhes de Home do SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.



Fonte: O Autor.

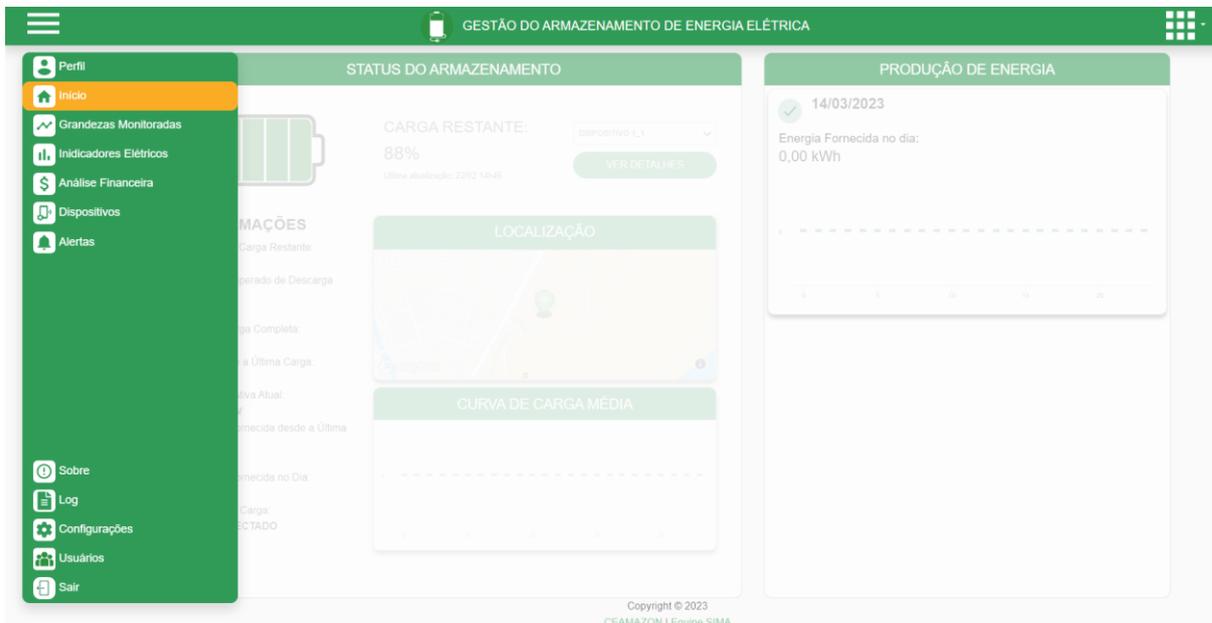
Menus

O software desenvolvido possui dois Menus principais: o Menu Lateral e o Menu de Aplicações. O Menu lateral fica na parte esquerda do *software*, possibilitando o acesso às telas do *software*, que só será permitido, de acordo com o nível de permissão do usuário autenticado na plataforma, sendo elas:

- Tela de Perfil: Usuário Normal;
- Tela de Início: Usuário Normal;
- Tela de Detalhes: Usuário Normal;
- Tela de Grandezas Monitoradas: Usuário Normal;
- Tela de Indicadores Elétricos: Usuário Normal;
- Tela de Análise Financeira: Usuário Normal;
- Tela de Dispositivos: Usuário Administrador;
- Tela de Alertas: Usuário Administrador;
- Tela de Sobre: Usuário Normal;
- Tela de Log: Usuário Administrador;
- Tela de Configurações: Usuário Administrador;

- Tela de Usuários: Usuário Administrador;
- Dado as telas apresentadas, o menu pode ser visto na Figura 75.

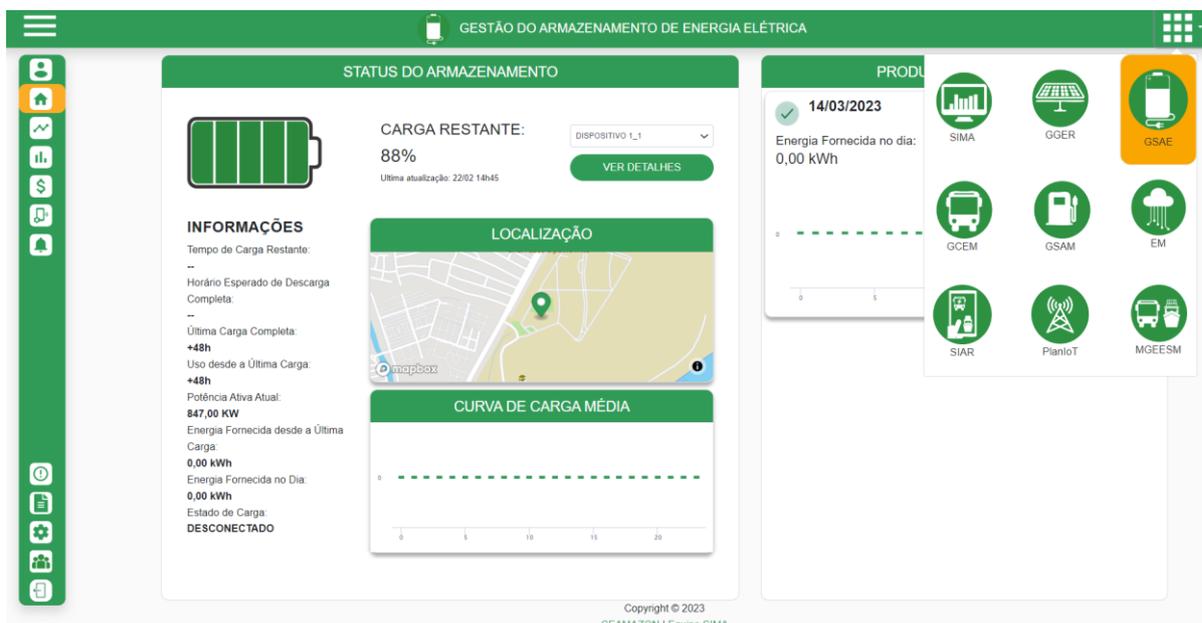
Figura 75 - Menu de ferramentas do SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.



Fonte: O Autor.

O Menu de Aplicações, se resume na possibilidade de transitar entre sistemas de modo que cada opção de *software* possa ser habilitada ou desabilitada via docker-compose. A Figura 76 representa o Menu de Aplicações descrito.

Figura 76 - Menu de aplicações do SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.



Fonte: O Autor.

Tela de Grandezas

A tela de Grandezas Monitoradas do SISGAE2B é responsável por exibir as informações (em séries temporais) referentes às medidas pontuais salvas na DOJOT. O *software* possui capacidade de visualizar as informações por diferentes formas, integralizando os dados em grandezas por dia, mês e ano.

Outra funcionalidade dessa tela é a possibilidade de mudar o tipo de gráfico, podendo ser possível exibir dados tanto no formato de linha, quanto no formato em gráfico de barras. Além disto, é possível utilizar ambos os eixos Y do gráfico para selecionar diversas grandezas, desde que sejam de até 2 tipos diferentes. Então o sistema consegue agrupar os dados para exibir e correlacionar o maior número de grandezas possível, como pode ser visto na Figura 77.

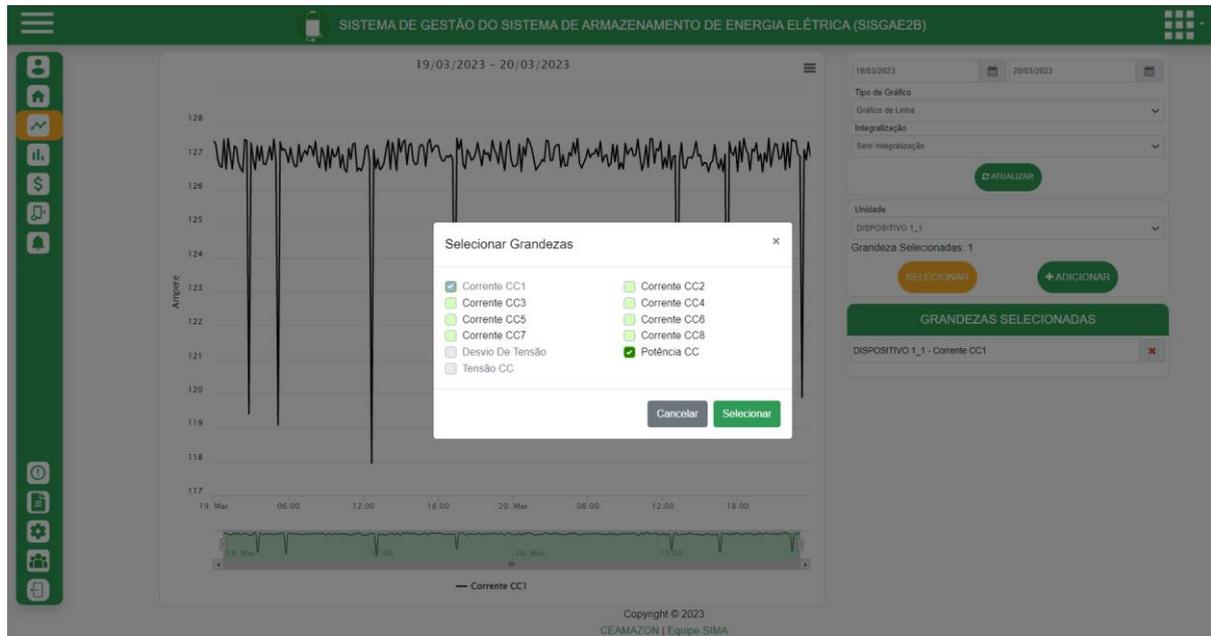
Figura 77 - Tela de Grandezas do SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.



Fonte: O Autor.

Ainda a respeito da tela de grandezas, é possível verificar os gráficos disponíveis para o acompanhamento na Figura 78, em que os gráficos de corrente descritos como Corrente CC 1, Corrente CC 2 até Corrente CC 8, são os pontos de conexão das *stringboxes* do sistema de armazenamento, esses dados são salvos na DOJOT e o sistema carrega de forma automática todas as *stringboxes* existentes.

Figura 78 - Tela de Grandezas disponíveis no SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.

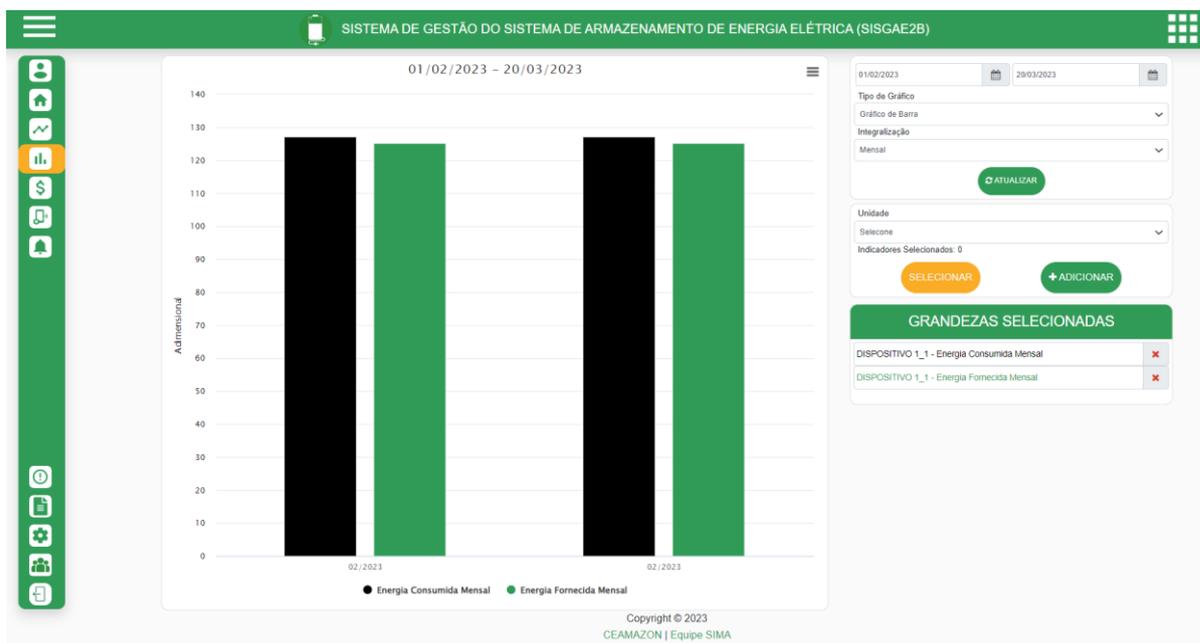


Fonte: O Autor.

Tela de Indicadores

De forma análoga à tela de grandezas, a tela de Indicadores Elétricos é capaz de auxiliar o gestor com diferentes ferramentas de visualização para fazer uma análise mais assertiva. Entretanto, a tela de indicadores se diferencia por exibir informações de indicadores de energia elétrica calculados com base nas grandezas. A Figura 79 representa a tela exposta.

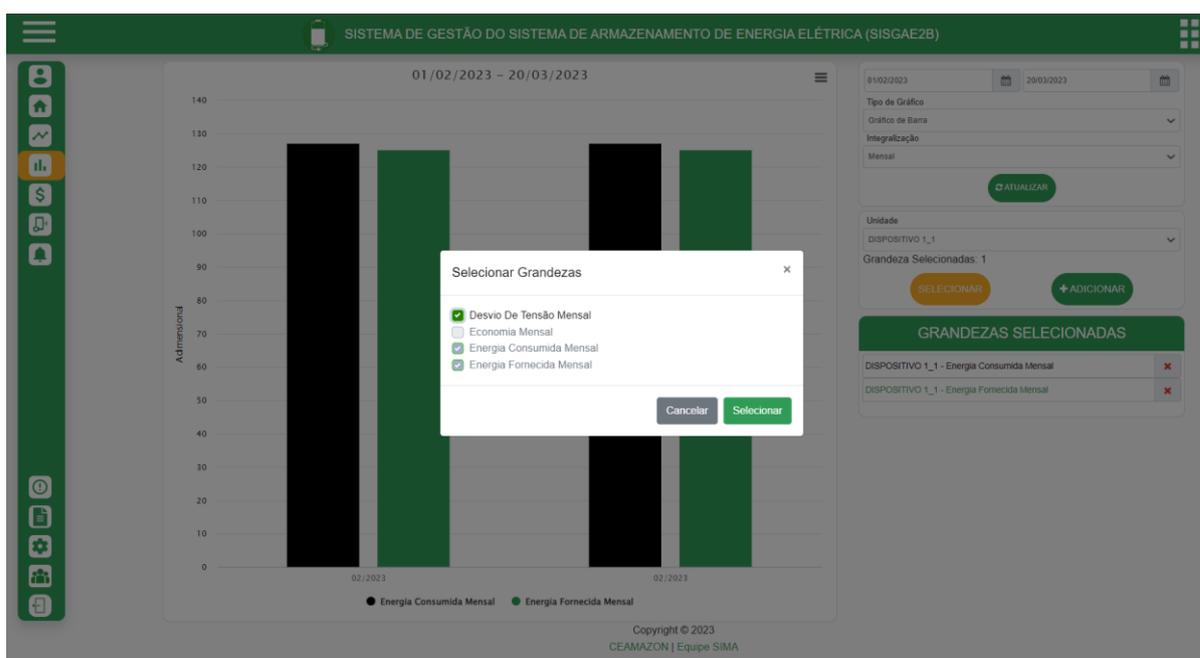
Figura 79 - Tela de indicadores do SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.



Fonte: O Autor.

A Figura 80 representa os gráficos que estão disponíveis:

Figura 80 - Tela de Indicadores do SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.

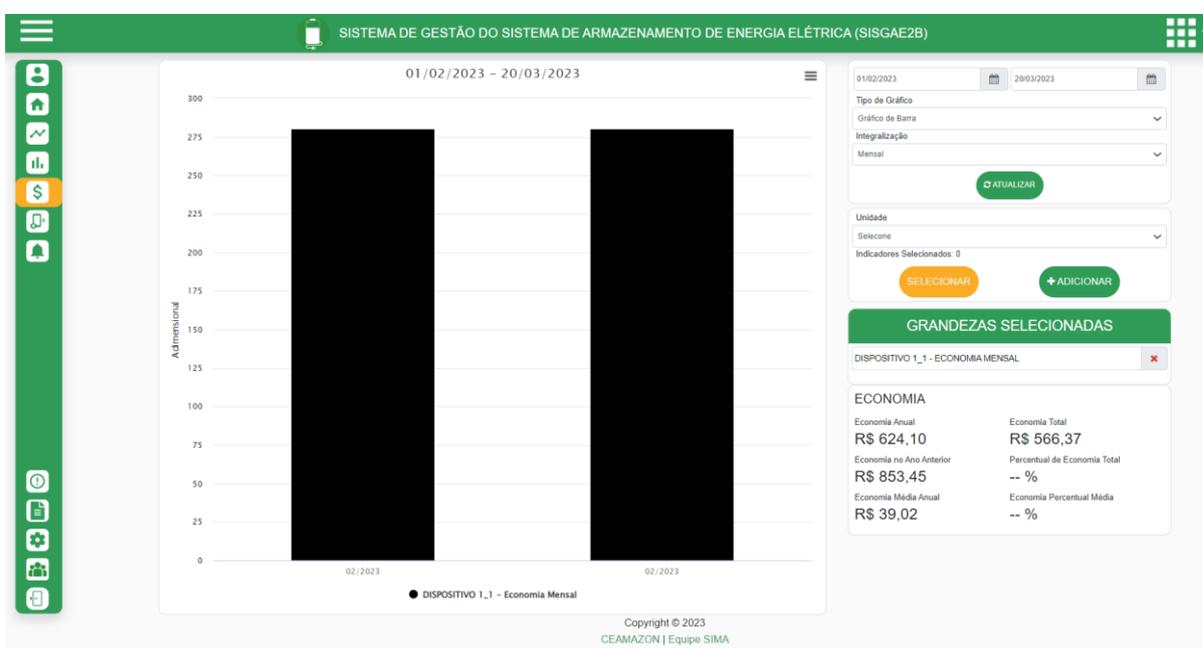


Fonte: O Autor.

Tela de Análise Financeira

Com o objetivo de facilitar a visualização de dados financeiros, esse indicador foi separado dos demais indicadores e foram calculados dados específicos voltados para análise financeira de um determinado período, como pode ser visto na Figura 81 abaixo.

Figura 81 - Tela de Análise Financeira do SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.



Fonte: O Autor.

Tela de Gerenciamento de Dispositivos

É responsável por realizar o gerenciamento dos sistemas de armazenamento no SISGAE2B, possibilitando visualizar, inserir, atualizar e até excluir sistemas os de armazenamento de energia. A Figura 82 demonstra a tela de visualização dos sistemas de armazenamento de energia no lado esquerdo e à direita da tela a opção de criar um novo sistema de armazenamento.

Figura 82 - Tela de Dispositivos com a opção de visualizar e criar dispositivos no SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.

The screenshot displays the SISGAE2B web application interface. On the left is a vertical green sidebar with various icons. The main content area is split into two panels. The left panel, titled 'DISPOSITIVOS', shows a list of devices, with the first one being 'DISPOSITIVO 1_1' with IP '91.190.146.102'. The right panel, titled 'DISPOSITIVO', shows a form for editing or creating a device. The form includes a map of the location, a 'mapbox' logo, and several input fields: 'Nome' (CEAMAZON), 'Local de Instalação' (CEAMAZON), 'Descrição' (INSERIDO NA SUBESTAÇÃO), 'Endereço IP' (192.186.0.1), 'Modelo' (TR 4000), 'Número Serial' (63305), 'Latitude' (-1.635), 'Longitude' (5.432), and 'Capacidade Nominal (V)' (400). A green '+ SALVAR' button is at the bottom right of the form. At the bottom of the page, there is a copyright notice: 'Copyright © 2023 CEAMAZON | Equipe SIMA'.

Fonte: O Autor.

Ainda no processo de gerenciamento dos sistemas de Armazenamento, ao clicar no ícone de lápis em um dispositivo no SISGAE2B, é possível editar o dispositivo, bem como ver o seu estado atual de tensão ou corrente. Isso se torna útil quando é identificado que um sistema está inoperante e é preciso ver o seu último ponto de medição, sem que seja necessário voltar para a tela de grandezas, conforme pode ser visto na Figura 83.

Figura 83- Tela de Dispositivos na opção de editar um dispositivo SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.

The screenshot displays the 'DISPOSITIVO' (Device) editing screen in the SISGAE2B system. The interface includes a sidebar with navigation icons, a top header with the system name, and a main content area. The main content area is divided into two panels: 'DISPOSITIVOS' (Devices) on the left and 'DISPOSITIVO' (Device) on the right. The 'DISPOSITIVO' panel contains a map showing the device location, a form with fields for Name, Description, Model, Latitude, Longitude, Local de Instalação, Endereço IP, Número Serial, and Capacidade Nominal (V). Below the form are 'ATUALIZAR' (Update) and 'VOLTAR' (Back) buttons. At the bottom of the panel is a line graph showing voltage (Tensão) over time, with a data point highlighted for 'Quarta, 22/02/2023 12h05' with a value of 53.31V.

Fonte: O Autor.

Tela de Gerenciamento de Alertas

No contexto do gerenciamento dos dispositivos de armazenamento, tem-se que na tela de Alertas o SISG2B pode ser configurado como ferramenta para a criação de um conjunto de alertas, de modo que cada usuário pode ser responsável por acompanhar determinados sistemas de armazenamento com os seguintes alertas disponíveis:

- Perda de comunicação;
- Desconexão;
- Transgressão de Tensão Persistente;
- Sobrecarga;
- Transgressão nos Indicadores de Energia Elétrica.
- Descarregada.
- Descarregando.
- Carregado.
- Nível de tensão abaixo de: 100%, 80%, 60%, 40%, 20%.
- Outros erros.

Estes alertas podem ser vistos na Figura 84.

Figura 84 - Tela de Alertas e Notificações do SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.

The screenshot displays the 'SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA (SISGAE2B)' interface. On the left is a vertical navigation menu with icons for home, user, charts, settings, and notifications. The main content area is split into two panels:

- NOTIFICAÇÕES:** A list of five notifications, each with a battery icon and a date:
 - SOBRECARGA:** Unidade: DISPOSITIVO 1,1; Grandeza: DRC; Data: 28/11/2022 00:00; Valor: 510 A.
 - BATERIA CARREGADA:** Unidade: DISPOSITIVO 1,1; Data: 03/09/2022 00:00.
 - NÍVEL DE CARGA ABAIXO DE 40%:** Unidade: DISPOSITIVO 1,1; Data: 27/08/2022 00:00.
 - NÍVEL DE CARGA ABAIXO DE 20%:** Unidade: DISPOSITIVO 1,1; Data: 17/06/2022 00:00.
 - CARREGANDO:** Unidade: DISPOSITIVO 1,1; Data: 02/06/2022 00:00.
- ALERTAS:** A section titled 'SISTEMA DE ARMAZENAMENTO' for 'DISPOSITIVO 1,1' containing a list of alert types with status indicators:
 - Perda de Comunicação
 - Desconexão
 - Transgressão de Tensão Persistente
 - Sobrecarga
 - Transgressão nos Indicadores de Energia Elétrica
 - Descarregada
 - Carregando
 - Carregada
 - Nível de Carga Abaixo de: 20%, 40%, 60%, 80%, 100%
 - Outros Erros
 A checkbox 'Receber Notificações por E-mail' is checked.

At the bottom of the interface, there is a pagination control showing '1 2 3' and a copyright notice: 'Copyright © 2023 CEAMAZON | Equipe SIMA'.

Fonte: O Autor.

Tela de Gerenciamento de Logs

A tela de logs do sistema surge como uma forma de manter a confiabilidade e a auditoria do que está sendo realizado dentro do sistema, de modo que seja possível resolver problemas indesejados no que tangem às configurações na plataforma, bloqueio de usuários que não possuem mais vínculo ou necessidade de acesso na plataforma, bem como se resguardar contra qualquer ação indevida. Essas informações ficam persistidas e podem ser acessadas a qualquer momento, conforme apresentado na Figura 85.

Figura 85 - Tela de Logs do SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.

The screenshot displays the SISGAE2B system interface. The top navigation bar is green and contains the system name: "SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA (SISGAE2B)". On the left, there is a vertical sidebar with various icons, including a home icon, a bar chart, a dollar sign, a document, and a gear. The main content area is divided into two panels. The left panel, titled "LOGS DO SISTEMA", shows a list of login events. The right panel, titled "USUÁRIOS", shows a list of users with their profile pictures, names, emails, and CPF numbers.

LOGS DO SISTEMA	
LOGIN REALIZADO	há 9 horas
<ul style="list-style-type: none"> Usuário: Izidio Carvalho Data: 19/03/2023 21:44:59 	
LOGIN REALIZADO	há 5 dias
LOGIN REALIZADO	há 6 dias
LOGIN REALIZADO	há 1 semana
LOGIN REALIZADO	há 1 semana
LOGIN REALIZADO	há 1 semana

USUÁRIOS	
	Izidio Carvalho izidiocarvalho88@gmail.com CPF: 000.000.000-10
	Priscila Lobato elen.lobato@tec.ufpa.br CPF: 000.000.000-12
	Rodrigo Muniz rodrigomuniz@gmail.com CPF: 000.000.000-11
	Test Lastname test.lastname@tec.ufpa.br CPF: 000.000.000-13

Copyright © 2023
CEAMAZON | Equipe SIMA

Fonte: O Autor.

Da mesma forma como é possível visualizar as informações gerais dos logs da plataforma, também é possível acessar logs para um usuário em específico, como pode ser visto na Figura 86.

Figura 86- Tela de Logs exibindo logs por um usuário no SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.

The screenshot displays the SISGAE2B system interface, similar to Figure 85, but with the logs filtered by a specific user. The top navigation bar and sidebar are the same. The main content area is divided into two panels. The left panel, titled "LOGS DO SISTEMA", shows a list of login events for the selected user. The right panel, titled "USUÁRIOS", shows a list of users, with the selected user highlighted in green.

LOGS DO SISTEMA	
LOGIN REALIZADO	há 10 horas
<ul style="list-style-type: none"> Usuário: Izidio Carvalho Data: 19/03/2023 21:44:59 	
LOGIN REALIZADO	há 5 dias
LOGIN REALIZADO	há 6 dias
LOGIN REALIZADO	há 1 semana
LOGIN REALIZADO	há 1 semana
LOGIN REALIZADO	há 1 semana

USUÁRIOS	
	Izidio Carvalho izidiocarvalho88@gmail.com CPF: 000.000.000-10
	Priscila Lobato elen.lobato@tec.ufpa.br CPF: 000.000.000-12
	Rodrigo Muniz rodrigomuniz@gmail.com CPF: 000.000.000-11
	Test Lastname test.lastname@tec.ufpa.br CPF: 000.000.000-13

Copyright © 2023
CEAMAZON | Equipe SIMA

Fonte: O Autor.

Tela de Configurações

A tela de configurações do sistema permite gerenciar os dados referentes aos postos tarifários, para que sejam realizados os cálculos financeiros do software, de modo que é possível definir um conjunto de postos para os intervalos de interesse que poderão ser utilizados como indicadores (Figura 87).

Figura 87 - Tela de Configurações do SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.

The screenshot shows the 'CONFIGURAÇÕES' (Configurations) screen for 'POSTOS TARIFÁRIOS' (Tariff Rates). The interface includes a sidebar with navigation icons and a main content area with input fields and a history table.

POSTOS TARIFÁRIOS

Posto: Convencional

Demanda Ponta (kW): 5000

Demanda Fora Ponta (kW): 200

Preço da Demanda Ponta (R\$/kW): 2,00

Preço da Demanda Ponta (R\$/kW): 2,00

Preço do Consumo Ponta (R\$/kWh): 2,00

Preço do Consumo Fora Ponta (R\$/kWh): 2,00

Tolerância de Ultrapassagem (%): 3,00

Penalização de Ultrapassagem (%): 3,00

20/03/2023 - 20/03/2023

Atualmente Vigente
 Valor Padrão

Histórico

\$ 01/01/2020 - 31/07/2020	<input checked="" type="checkbox"/>	<input type="checkbox"/>
\$ 01/01/2020 - 31/07/2020	<input checked="" type="checkbox"/>	<input type="checkbox"/>
\$ 01/08/2020 - 31/12/2020	<input checked="" type="checkbox"/>	<input type="checkbox"/>
\$ 01/08/2020 - 31/12/2020	<input checked="" type="checkbox"/>	<input type="checkbox"/>
\$ 01/01/2021 - 31/07/2021	<input checked="" type="checkbox"/>	<input type="checkbox"/>
\$ 01/01/2021 - 31/07/2021	<input checked="" type="checkbox"/>	<input type="checkbox"/>
\$ 01/08/2021 - 31/12/2021	<input checked="" type="checkbox"/>	<input type="checkbox"/>
\$ 01/08/2021 - 31/12/2021	<input checked="" type="checkbox"/>	<input type="checkbox"/>

+ SALVAR

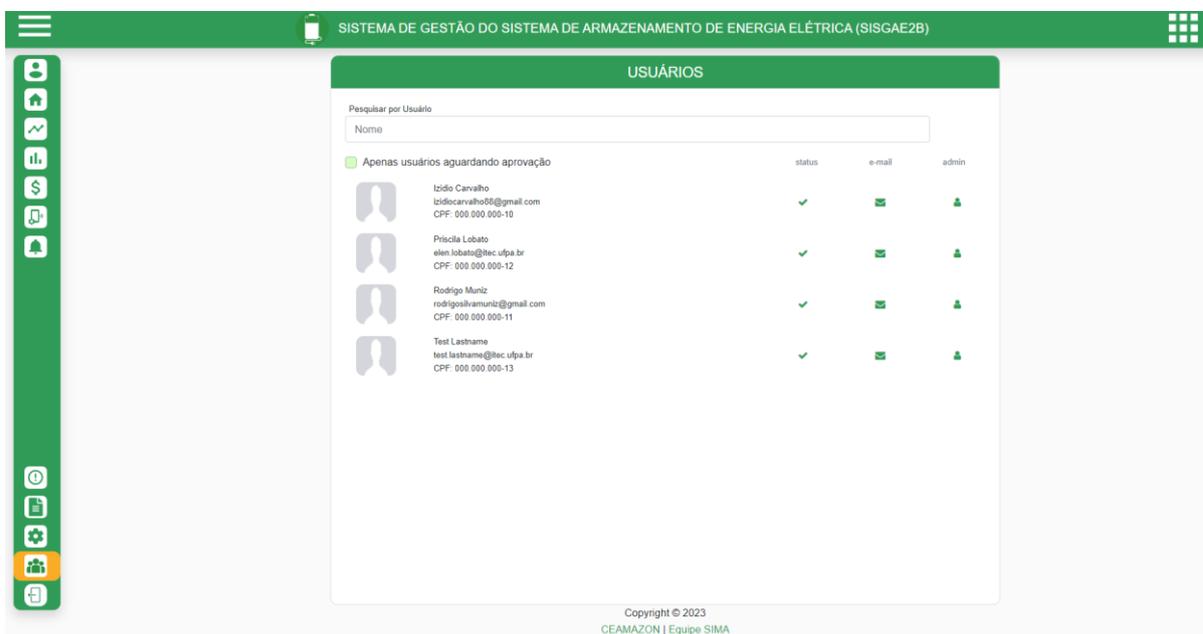
Copyright © 2023
CEAMAZON | Equipe SIMA

Fonte: O Autor.

Tela de Gerenciamento de Usuários

Por meio da tela de gerenciamento de usuários é possível filtrar os usuários do sistema, verificar o status do seu e-mail, habilitar e desabilitar usuários, modificar o perfil de um usuário entre usuário Administrador e usuário Normal. Esses detalhes ocorrem por meio do clique nos respectivos ícones ao lado dos dados do usuário, como pode ser visto na Figura 88.

Figura 88 - Tela de usuários do SISTEMA DE GESTÃO DO SISTEMA DE ARMAZENAMENTO DE ENERGIA ELÉTRICA – SISGAE2B.



Fonte: O Autor.

4.3 Conclusão

No decorrer do capítulo foram apresentados os pontos a respeito dos resultados obtidos por meio da implementação dos *softwares* de gerenciamento da geração e de armazenamento de energia, de modo a ser possível englobar os pontos de análise da qualidade de energia elétrica por meio de grandezas e indicadores existentes em ambos os softwares.

Os *softwares* foram desenvolvidos contendo um módulo de cadastro e autenticação de usuários e uma tela inicial voltada para o entendimento geral dos sistemas analisados, visando o acompanhamento de dados de produção de energia para os sistemas fotovoltaicos e do ciclo de carga e descarga para os sistemas de armazenamento de energia elétrica; também ficou disposto no presente capítulo as telas referentes aos gráficos de grandezas e indicadores elétricos que foram implementados, estes possuem a finalidade de auxiliar a gestão na tomada de decisão; mediante as telas de monitoramento, alertas e logs que foram desenvolvidas é possível ter o acompanhamento do funcionamento dos softwares; e dado as telas de gerenciamento de medidores, configurações e

gerenciamento de usuários, é possível efetuar os principais gerenciamentos para configurar o *software*.

CAPÍTULO 5: CONCLUSÃO

Este capítulo é organizado em duas seções, em que na seção 5.1 estão descritas as considerações finais a respeito do trabalho desenvolvido, tendo como base os objetivos gerais e específicos definidos para o trabalho em questão. Já na seção 5.2 são apresentadas opções de trabalhos futuros, visando a continuidade deste trabalho.

5.1. Considerações finais

A base do trabalho se dá pela crescente utilização de sistemas de geração de energia renovável no Brasil e no mundo, em especial a geração fotovoltaica, decorrente do aumento do consumo e da necessidade de diminuição do uso de energia proveniente de fontes não renováveis. Devido a intermitência durante a geração de sistemas fotovoltaicos, vem sendo discutido a utilização de sistemas de armazenamento, principalmente baterias, capazes de suprir a demanda de energia.

Nesse contexto surge o trabalho em questão, visando desenvolver *softwares* capazes de melhorar o processo de gestão de sistemas geradores de energia renovável por meio do monitoramento de grandezas e de indicadores elétricos provenientes destes sistemas. Para este trabalho, buscou-se a implementação de dois sistemas, um voltado para o gerenciamento de sistemas de geração de energia fotovoltaica e outro para o gerenciamento de sistemas de armazenamento de energia elétrica, de modo em que ambos os softwares possam ser instalados, configurados e gerenciados de forma individual.

O trabalho em questão visou ter como base um conjunto de tecnologias voltadas para uma implementação segura e robusta de um *software*, que garanta um bom funcionamento dos sistemas criados, bem como a interação com o gestor que irá utilizar as ferramentas de análises gráficas, permitindo que seja possível uma gama de gráficos para análises de qualidade de energia e controle financeira.

Sendo assim, por meio dos resultados evidenciados e tendo como base o objetivo geral, é visto que foi possível realizar a implementação dos sistemas desenvolvidos nos servidores da UFPA, com os *softwares* necessários para atender as atuais necessidades de acompanhamento e de gerenciamento de

sistemas de geração de energia elétrica e sistemas de armazenamento de energia elétrica.

Dentre os objetivos específicos para este trabalho, foi realizado o desenvolvimento e implementação de *softwares* voltados para o gerenciamento de sistemas, sendo estes o Sistema de Gestão da Geração de Energia Renovável (SIS2GER) e o Sistema de Gestão do Sistema de Armazenamento de Energia Elétrica (SISGAE2B), de modo que ambos os *softwares* possuem sistemas de configurações voltadas para um conjunto de especificidades de forma independente entre si, podendo configurar cada sistema de forma independente.

Dentre as opções de gerenciamento existente nos *softwares* é possível destacar: o gerenciamento de usuários, o qual permite que *softwares* possam ter a existência de usuários com níveis de permissão diferentes para cada *software*; tem-se também o gerenciamento de alertas, capaz de enviar alertas específicos com base em um usuário especificado, nos sistemas selecionados para o acompanhamento e nas opções configuradas para cada sistema; bem como a coleta de logs, os quais são voltados para auditoria as ações que ocorrem dentro dos *softwares*, visando uma maior segurança para os gestores e garantindo que seja possível desfazer qualquer ação realizada no sistema.

No que tange ao acompanhamento grandezas elétrica, o sistema é capaz de exibir os dados de telemetria, estes dados são referentes aos dispositivos salvos no *middleware* DOJOT em intervalos de 10 min de acordo com o PRODIST, também é possível constatar a utilização de dois eixos para a visualização dos gráficos implementando, melhorando a visualização no sistema.

Em relação aos indicadores elétricos, verificou-se a possibilidade de visualização utilizando gráficos para análises dos dados que são armazenados na DOJOT de forma mensal ou anuais sem limite de intervalo de busca, o que auxilia o trabalho da gestão por meio do acompanhamento dos dados indicadores do sistema em diversos períodos.

Desta forma, o trabalho apresentado é capaz de fornecer as ferramentas necessárias para o monitoramento e facilitar a tomada de decisão por parte de gestores através de diversos indicadores de qualidade de energia elétrica e a vasta possibilidade de visualização.

Tem-se também que o desenvolvimento dos sistemas apresentados neste trabalho permite que seja expandida a gama de ferramentas presentes no projeto SIMA (Sistema Integrado de Mobilidade Elétrica Multimodal), ao qual este trabalho possui vínculo, sendo possível acrescentar à estrutura do projeto novas opções de análise para a construção de uma cidade inteligente em que os diversos sistemas e dispositivos estão conectados e possuem uma massiva coleta de dados de telemetria para que seja possível o acompanhamento em tempo real.

5.2. Trabalhos Futuros

Como trabalhos futuros, tem-se a necessidade de um mundo cada vez mais conectado e com mais ferramentas de análises, além da utilização de novas tecnologias voltadas para a melhoria de sistemas, por conta disso tem-se alguns pontos levantados como melhorias:

- Tem-se a importância da existência agregar informações principais de outros softwares desenvolvidos em um único local, de modo que este software possa se comportar como um hub de softwares e ser capaz de concentrar uma visão geral de todos os sistemas desenvolvidos, enquanto os sistemas em si possam ser mais especializados nos seus respectivos softwares.
- A comunicação entre softwares é algo cada vez mais comum em um ambiente conectado e construído com base em micros serviços, para tal, quanto melhor a comunicação e menor o tempo para acesso em outros sistemas, mais rápido os softwares irão processar os dados, para tal, é importante começar o processo de validação do uso de gRPC (Google Remote Procedure Call).
- Uma vez que existam diversas grandezas e indicadores em uma base de dados, um ponto de melhoria futura se dá pela criação de relatórios de qualidade de forma automática e enviada via e-mail para que os gestores possam acompanhar o andamento mensal da plataforma.
- Dentro da rotina de desenvolvimento dos softwares do projeto, tem-se a criação de novas rotinas automáticas, como o CI/CD (*Continuous Integration / Continuous Deployment*) que são capazes de melhorar os tempos de entrega de sistemas para o ambiente final.

- No que tange ao desenvolvimento, tem-se a necessidade de implementação de testes unitários e testes de integração que possam ser utilizados quando existe uma gama de projetos sendo desenvolvidas em paralelo, com isso evitando que problemas possam acontecer em uma fase final de desenvolvimento.

5.3. Produções

- **Registro de Softwares**

Dentro do trabalho exposto, foram realizados dois registros de software pelo Instituto Nacional da Propriedade Industrial (INPI) de softwares de aplicações, um para cada Software desenvolvido dentro do projeto:

1. TOSTES, Emília; BEZERRA, Ubiratan; MUNIZ, João; **CARVALHO, Izidio**; LOBATO, Elen; ALBUQUERQUE, Bruno; PAIXÃO, Ulisses; MORAES, Jean. Sistema de Gestão de Geração de Energia Elétrica Fotovoltaica. Titular: Norte Energia S.A; Universidade Federal do Pará; BYD Energy do Brasil LTDA; ABB Eletrificação LTDA. Procurador: José Carlos Vaz e Dias. BR no BR512021003140-7. Pedido: 20 dez. 2021.
2. TOSTES, Emília; BEZERRA, Ubiratan; MUNIZ, João; **CARVALHO, Izidio**; LOBATO, Elen; ANDRADE, Vinícius; DUARTE, Ana; LIMA, Áthila; MORAES, Wuanda. Sistema de Gestão de Armazenamento de Energia Elétrica em Banco de Baterias. Titular: Norte Energia S.A; Universidade Federal do Pará; BYD Energy do Brasil LTDA; ABB Eletrificação LTDA. Procurador: José Carlos Vaz e Dias. BR no BR512021003139-3. Pedido: 20 dez. 2021.

REFERÊNCIAS BIBLIOGRÁFICAS

AGLIARDI, GUNTHER; CHEPP, ELLEN DAVID; GASPARIN, FABIANO PERIN. Monitoramento de sistemas fotovoltaicos utilizando diferentes base de dados de radiação solar e temperatura. **Simpósio sobre Sistemas Sustentáveis (6.: 2021:[On-line]). Anais: volume 4: sustentabilidade. Toledo, PR: GFM, 2021., 2021.**

ALMEIDA, ANA PAULA ROCHA. **Avaliação técnico-econômica da geração de energia fotovoltaica em unidades geradoras na região Oeste do Paraná.** Dissertação. Universidade Estadual do Oeste do Paraná - UNIOESTE, Toledo, 2021.

ALVES, MARLIANA DE OLIVEIRA LAGE. **Energia solar: estudo da geração de energia elétrica através dos sistemas fotovoltaicos on-grid e off-grid.** Monografia. Universidade Federal de Ouro Preto. Ouro Preto, 2019.

ANDRADE, LUCAS ESTEVÃO DE. **Implantação de um Processo de Integração e Deployment Contínuos em um Laboratório de Pesquisa.** Trabalho de Conclusão de Curso. Universidade Federal de Santa Catarina, Florianópolis, 2022.

ANDRADE, VINICIUS; CARVALHO, ULISSES; MUÑOZ TABORA, JONATHAN; TOSTES, MARIA; BEZERRA, UBIRATAN; MATOS, EDSON; DE ALBUQUERQUE, BRUNO; MOREIRA RODRIGUES, CARLOS; TAKEDA, FELIPE; CÉLIA, CARMINDA. **Modelagem e simulação de cenários da operabilidade de uma mini rede híbrida com geração fotovoltaico-diesel, armazenamento de energia conectada à rede elétrica.** Anais. XII Congresso Brasileiro de Planejamento Energético. Foz do Iguaçu, Brasil: abr. 2020.

ANDRE, FRANCISCO MENDES MICAEL. **Dimensionamento de Sistemas Fotovoltaicos com Baterias Ligados à Rede.** Departamento De Engenharia Eletrotécnica E De Computadores. Lisboa/PT. 2022.

BALBINO, NATALUZO DA SILVA. **Conflito de uso entre abastecimento público e irrigação associado à crise hídrica na Bacia do Alto Descoberto, Distrito Federal.** Dissertação. Universidade de Brasília. Distrito Federal, Brasília. 2020.

BALIEIRO, R. **Livro: ESTRUTURA DE DADOS.** 1ª edição SESES ed. Rio de Janeiro: SESES, 2015. v. 1

BARBOSA, NILTON FREITAS; VERGARA, GUILHERME FAY; DA SILVA, DANIEL ALVES; TABOSA, FÁBIO G FERREIRA; PRACIANO, FLÁVIO GARCIA; NZE, GEORGES DANIEL AMVAME. Preservação de documentos arquivísticos digitais: Um estudo inicial de segurança e vulnerabilidades em Docker Container para criação de serviços de RDC-ARQS. **Conferências IADIS Ibero-Americanas WWW/Internet e Computação Aplicada,** 2019.

BARDELIN, CESAR ENDRIGO ALVES. **Os efeitos do racionamento de energia elétrica ocorrido no Brasil em 2001 e 2002 com ênfase no consumo de energia elétrica.** Tese. Universidade de São Paulo, São Paulo. 2004.

BATISTA, HITALLO WILLIAM DE MEDEIROS. **Uma ferramenta para simulação e monitoramento de comportamentos anômalos em contêineres Docker**. Trabalho de Conclusão de Curso. Universidade Federal do Rio Grande do Norte, RN, Caicó. 2022.

BEN. **Balanco Energético Nacional**. 2022. Disponível em: <<http://www.epe.gov.br>>.

BOSCO, JOÃO PEDRO CAYRES. **Docker como ferramenta para construção de ambiente em nuvem**. Monografia. Faculdade De Tecnologia De Americana. São Paulo, Americana. 2019.

BRASIL, CRISTIANE FIGUEIRA. **Análise técnico-econômica e de eficiência dos principais tipos de baterias utilizadas em sistemas fotovoltaicos isolados**. Trabalho de Conclusão de Curso. Universidade do Estado do Amazonas. Amazonas, Manaus. 2018.

BUENO, A. F. M.; BRANDÃO, C. A. L. **Visão geral de tecnologia e mercado para os sistemas de armazenamento de energia elétrica no Brasil**. Associação Brasileira de Armazenamento e Qualidade de Energia, 2016.

CAMPOS, FELIPPE MELO; ARAÚJO, DANIELLY NORBERTO; TOLEDO, OLGA MORAES; SANTO FERNANDES, LUCAS DO ESPÍRITO; BORBA, ANA TEREZA ANDRADE. **Tecnologias e Aplicações de Sistemas de Armazenamento de Energia para Suporte à Integração de Fontes Renováveis no Brasil**. Congresso. Congresso Brasileiro de Energia Solar-CBENS. 2022.

CAMPOS, LUIZ HENRIQUE BICALHO. **Desenvolvimento de um Sistema de Gestão para o Viveiro de Mudas Do IFMG-SJE**. Trabalho de Conclusão de Curso. Instituto Federal De Ciência E Tecnologia De Minas Gerais. Campus São João Evangelista. IFMG, São João Evangelista 2020.

CARVALHO, ANTÔNIO PEDRO DOS SANTOS. **Observabilidade e telemetria em arquiteturas de micro-serviços**. Dissertação. Faculdade de Engenharia da Universidade do Porto. Porto-PT. 2022.

CARVALHO, HAELTON; SÁ, JOINER.; FARIAS, FABRICIO. **Implantação de uma Arquitetura de Software para Monitoramento de Dados Ambientais em um Cenário de Smart Campus**. Anais. Anais do XII Workshop de Computação Aplicada à Gestão do Meio Ambiente e Recursos Naturais. 2021.

CHAVES, ALINE MARTINS; SILVA, GABRIEL. **Proposta de uma arquitetura de software e funcionalidades para implementação de um ambiente integrado de desenvolvimento para a linguagem PHP**. I Jornada Científica e VI FIPA do CEFET Bambuí, v. 31, p. 32–36, 2008.

CLEMENTE, ANDERSON ALMEIDA DE SOUZA; SILVA, EVALDO DE OLIVEIRA DA. **Migração de sistemas Monolíticos para Microsserviços**. Caderno de Estudos em Sistemas de Informação, v. 7, n. 2, 2022.

CLOSS, THALES ALEXSANDER. **Virtualização em ambiente corporativo com ferramentas open-source**. Trabalho de Conclusão de Curso. Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso (IFMT). Mato Grosso. Cuiabá. 2021.

COLANTONI, MARCUS ARTIAGA. **Habilitando previsões de desastres em cenários de IoT com transferência de aprendizado**. Dissertação. Universidade Federal de Uberlândia, Uberlândia. 2022.

COSTA, CARRIJO COSTA.; GARREFA, FERNANDO. Ensaio sobre a geração distribuída de energia fotovoltaica: um modelo sustentável para a matriz energética brasileira. **Revista Científica ANAP Brasil**, v. 15, n. 1984–3240, 2022.

COSTA, LUCÉLIO MANUEL DA. **Análise Técnico-Económica de Sistema de Alimentação de Posto de Carregamento de Veículos Elétricos com recurso a Energia Fotovoltaica e Sistema de Armazenamento de Energia Elétrica**. Departamento De Engenharia Eletrotécnica. PT, Coimbra. 2021.

COSTA, KAIQUE JUVENCIO; SANTOS, KALVIN VINICIUS VASCONCELLOS DOS; ROJAS, ROGER THENCERÁ; DUARTE JUNIOR, WENDEL SERGIO. **Estudo Comparativo de Bancos de Dados NoSQL Distribuídos**. Faculdade de Computação e Informática. Trabalho de Conclusão de Curso. Universidade Presbiteriana Mackenzie. 2021.

COSTA, YÁSCARA F. F. COSTA E SILVA; BORTONI, SILVA EDSON C. **Sistemas de armazenamento de energia elétrica em redes inteligentes: Características, oportunidades e barreiras**. REVISTA BRASILEIRA DE ENERGIA. Vol. 22. Nº 1. 1º Sem. 2016.

COUTO, GABRIEL AUGUSTO SILVA; DE ALBUQUERQUE, BRUNO SANTANA; DE LIMA TOSTES, MARIA EMILIA; CARVALHO, CARMINDA CÉLIA MOURA DE MOURA; BEZERRA, UBIRATAN HOLANDA. **Análise e Soluções para a Qualidade da Rede de Energia Elétrica da UFPA**. Revista Contemporânea, v. 2, n. 5, p. 578–596, 2022.

DE MORAIS, ANDERSON MELO; CALLOU, GUSTAVO RAU DE ALMEIDA; LINS, FERNANDO ANTONIO AIRES. **Simulação e Avaliação de Desempenho de uma Rede Blockchain Utilizando Containers Docker**. Cadernos do IME-Série Informática, v. 44, p. 73–87, 2020.

DE SOUZA, JAIR SILVA; GALHARDO, MARCOS ANDRÉ BARROS. **Medição e indicadores de qualidade de energia elétrica em nanorrede de distribuição C.C**. Anais. Congresso Brasileiro de Energia Solar-CBENS. 2022.

DIAS, RICARDO VIEIRA DE AZEVEDO. **Sistema de Gestão Energética: estudo de caso no centro de convergência da Universidade Federal de Ouro Preto**. Mestrado. Universidade Federal de Ouro Preto. Minas Gerais. Ouro Preto. 2020.

DOJOT, A. **DOJOT - Architecture**. **DOJOT - Architecture**, 2023. Disponível em: <<https://dojotdocs.readthedocs.io/en/latest/architecture.html#architecture>>. Acesso em: 6 abr. 2023

FARIAS, HÉRICLES EDUARDO OLIVEIRA. **Análise técnica do impacto do uso de baterias em sistemas conectados à rede aplicado em diferentes classes de consumo**. Trabalho de Conclusão de Curso. Universidade Federal de Santa Maria (UFSM-RS). Santa Maria, RS. 2018.

FERRAO, ISADORA GARCIA; DE MACEDO, DOUGLAS DJ; KREUTZ, DIEGO. **Investigação o do impacto de frameworks de desenvolvimento de software na segurança de sistemas web**. 16a Escola Regional de Redes de Computadores (ERRC), 2018.

FERREIRA, BRENDA SOTERO. **Framework Laravel: um estudo de caso full stack development**. Universidade Federal De Ouro Preto. Monografia. Minas Gerais. Ouro Preto .2021.

FERREIRA, JARDEL GONÇALVES. **Beaver View: uma ferramenta interativa para o gerenciamento de infraestrutura virtual Docker baseada em SDN**. Universidade Federal do Ceará (UFC). Trabalho de Conclusão de Curso. 2018.

GARCIA, WALKER; PEREIRA, FAGNER COIN. **DOCKER – Containers não são VM's**. Seminário. Seminário de Tecnologia Gestão e Educação, v. 1, n. 2, p. 5–10, 2019.

GOMES, RAFAEL LOPES. **Middleware IoT para Monitoramento Ambiental**. Computação Brasil, n. 40, p. 37–40, 2019.

IEA. **International Energy Outlook 2020**. Disponível em: <<https://www.eia.gov/outlooks/ieo/>>. Acesso em: 10 maio de 2023.

JACINTO, ULISSES DOS SANTOS. **Blindagem de Ambientes Virtuais Baseados em Contêineres: Docker Hardening**. Monografia. Universidade Federal do Maranhão. Maranhão, São Luís. 2022.

JÚNIOR, DIVINO ALVES FERREIRA; CABRAL, JOÃO PAULO OLIVEIRA; MACEDO, CIRO; FILHO, TÉRCIO A. DOS SANTOS; CARDOSO, KLEBER VIEIRA; OLIVEIRA JR, ANTONIO C. **Implantação e Avaliação de um Protótipo para Filas Inteligentes utilizando um Dispositivo IoT Wi-Fi e um Gateway IoT Definido por Software**. Anais. Anais da VII Escola Regional de Informática de Goiás. 2019.

JUNIOR, FRANCISCO; QUINTINO, JOYCE; FREITAS, RENATO; MONTEIRO, ODORICO; CRISTINA, IVANA; OLIVEIRA, MAURO. **NextGISSA, um Sistema Inteligente para Monitoramento e Predição de Hipertensão Arterial em uma Plataforma de Saúde Digital**. Anais. Anais Estendidos do XXVIII Simpósio Brasileiro de Sistemas Multimídia e Web. 2022.

JUNIOR, JAIR URBANETZ; DAS NEVES, CARLOS EDUARDO TEIXEIRA. **Análise Comparativa de Sistemas Fotovoltaicos conectados a Rede Instalados em Curitiba com Diferentes Topologias**. Anais. Congresso Brasileiro de Energia Solar-CBENS. 2022.

JÚNIOR, ULISSES CARVALHO PAIXÃO. **Metodologia para Desenvolvimento e Gestão de Comunidades de Energia, com Estudo de Caso na Universidade Federal do Pará**. Universidade Federal do Pará. Tese. Belém, PA. 2022.

LIMA, DAMIANO ALVES DE. **Virtualização de sistema de detecção de ataque de negação de serviço**. Trabalho de Conclusão de Curso. Universidade Federal do Rio Grande do Norte. Rio Grande do Norte, Natal. 2021.

LOBATO, ELEN PRISCILA. **Implantação de um middleware IoT escalável para aplicações de Mobilidade Elétrica Multimodal**. Dissertação de Mestrado. Universidade Federal do Pará. Pará, Belém. 2022.

MAIA, HERICK MACIEL. **Análise de uso da tecnologia de software docker aplicando containerização na computação em nuvem**. Trabalho de Conclusão de Curso. Universidade do Extremo Sul Catarinense, UNESC. Santa Catarina, Criciúma. 2020.

MAIA, RIAN SARDINHA. **Energia solar: o desenvolvimento de um novo mercado**. Trabalho de Conclusão de Curso. Universidade Federal do Rio de Janeiro. Rio de Janeiro. 2018.

MEDEIRO, RAFAEL BARRETO DE. **Impactos Da Geração Distribuída Fotovoltaica Na Qualidade De Energia Em Rede De Distribuição De Energia Elétrica**. Trabalho de Conclusão de Curso. Universidade Federal do Ceará. Fortaleza. 2014.

MEDEIROS, ÁTILA DE SOUZA. **Monitoramento e medição de água em um campus inteligente com plataformas de IoT**. Dissertação. Instituto Federal de Educação, Ciência e Tecnologia da Paraíba Campus João Pessoa. Paraíba, João Pessoa .2023.

MONTEIRO, LUCIANO DE AGUIAR; ALMEIDA, WASHINGTON HENRIQUE CARVALHO. **Cluster de Alta Disponibilidade com Docker Swarm**. III Escola Regional de Informática do Piauí. Livro Anais - Artigos e Minicursos, v. 1, p. 279–295, 2017.

MONTEIRO, THAÍS SOARES. **Sistemas fotovoltaicos isolados em comunidades quilombolas remotas no cerrado: caso Pisco de Luz e a sustentabilidade**. Trabalho de Conclusão de Curso. Universidade de Brasília. Distrito Federal, Brasília.2020.

MORAES, JEAN; LOBATO, ELEN; ROSÁRIO, DENIS; BEZERRA, UBIRATAN; CERQUEIRA, EDUARDO; TOSTES, MARIA; ANTLOGA, ANDRÉIA. **Implementação de um cluster Kubernetes com a plataforma Dojot para Aplicações de Internet das Coisas**. Anais do XLVIII Seminário Integrado de Software e Hardware. 2021.

MORAIS, MICHEL AUGUSTO QUEIROZ. **Aplicação e comparação de ambientes de virtualização utilizando softwares livres ProxMox e XCP-ng**. Trabalho de Conclusão de Curso. Universidade Federal do Estado do Rio de Janeiro. RJ, Rio de Janeiro. 2019.

NETO, JOSÉ CLEANTO F ARRAIS; ARAÚJO, FRANCISCA LUZIA N; NETO, MAURCIO MOREIRA; PAILLARD, GABRIEL A L; MOREIRA, LEONARDO O. **Um Estudo Exploratório entre Banco de Dados NoSQL e Armazenamento de Dados em Blockchain**. Revista Sistemas e Mdias Digitais (RSMD), v. 6, n. 1, 2021.

NONATO, GABRIEL SOARES; MENDES, LUCAS LOPES; MENDES, MATHEUS LIMA. **Sistema de inspeção predial para engenheiros e arquitetos**. Trabalho de Conclusão de Curso. Distrito Federal. Brasília. 2020.

NUNES, JACYANA SUASSUNA. **Implantando a observabilidade do sistema SigSaúde através da utilização de service mesh**. Universidade Federal do Rio Grande do Norte. Dissertação. Rio Grande do Norte, Natal, 2019.

OLIVEIRA, FÁBIO HENRIQUE FERREIRA DE. **Automação do processo de implantação e testes de um sistema WEB utilizando Docker**. Trabalho de Conclusão de Curso. Universidade Federal do Rio Grande do Norte. RN, Macaíba. 2019.

PAIVA, DANILO ALMEIDA. **Virtualização da infraestrutura: sistematização e pesquisas correntes**. Monografia. Universidade Federal de Minas Gerais. Distrito Federal, Brasília 2019.

PELIZZA, ANGÉLICA CAETANE; BERTOLINI, CRISTIANO; SILVEIRA, SIDNEI RENATO. **Um estudo sobre técnicas de teste de software no Framework Laravel**. Revista Eletrônica de Sistemas de Informação e Gestão Tecnológica, v. 9, n. 3, 2018.

PEREIRA, NARLON XAVIER. **Desafios e perspectivas da energia solar fotovoltaica no Brasil: Geração Distribuída vs Geração Centralizada**. Dissertação. Universidade Estadual Paulista (Unesp), Sorocaba. 2019.

PRODIST, Procedimentos de Distribuição. **Módulo 8– Qualidade da Energia Elétrica**. Agência Nacional de Energia Elétrica–ANEEL, 2018.

ROCHA, GUSTAVO DOS SANTOS. **Docker Swarm e Kubernetes: avaliando benefícios em ambientes virtuais**. Trabalho de Conclusão de curso. Universidade Federal Fluminense. Rio de Janeiro, Niterói. 2022.

SAMPAIO, LUAN ARAUJO. **Avaliando a eficiência energética de uma conexão com a internet através do gprs em um cenário IoT: um estudo de caso com o SIM800I e o middleware DOJOT**. Trabalho de Conclusão de Curso. Universidade Federal do Ceará. Ceará, Quixadá. 2018.

SANTOS, BRENA; ENDO, PATRICIA TAKAKO; SILVA, FRANCISCO AIRTON. **Uma avaliação de desempenho de contêineres docker executando diferentes sgbds relacionais**. Anais do XVIII Workshop em Desempenho de Sistemas Computacionais e de Comunicação. 2019.

SILVA, FLÁVIO HENRIQUE ROCHA E. **Avaliação de Desempenho de Contêineres Docker para Aplicações do Supremo Tribunal Federal**. Monografia. Universidade de Brasília. Brasília, 2017.

SILVA, FRANCISCO AIRTON; MARQUES, EDER SOARES. **Desenvolvimento de uma aplicação Web para os Centros de Educação Infantil do município de Coxim-MS utilizando o Framework Laravel**. EDITORA CIENTÍFICA DIGITAL. TECNOLOGIAS EDUCACIONAIS: METODOLOGIAS, TÉCNICAS E AMBIENTES EM PESQUISA. v. 1p. 75–93. 2022.

SANTOS, KELDA APARECIDA GODÓI. **Impactos Do Aumento Da Entrada Da Energia Solar Fotovoltaica Na Rede Elétrica Distribuída: Implicações Técnicas Do Aumento Da Geração Própria De Energia**. Dissertação. Universidade Federal de Itajubá. Minas Gerais, Itajubá. 2016.

SCHUERZOSKI, M. R. **Estudo e aplicação de tuning em sistema gerenciador de banco de dados orientado a documentos**. Ponta Grossa: Universidade Tecnológica Federal do Paraná, 2019.

SHAYANI, RAFAEL AMARAL; OLIVEIRA, MARCO AURÉLIO GONÇALVES DE; CAMARGO, IVAN MARQUES DE TOLEDO. **Fontes alternativas de energia: Conceitos básicos sobre energia solar**. Congresso Brasileiro de Planejamento Energético, 2006.

SILVA, ANTÔNIO EUGÊNIO DOMINGUES. **Sistema de ensino remoto de educação física para crianças através da gamificação**. Trabalho de conclusão de curso. Universidade Estadual Paulista Júlio de Mesquita Filho. SP, Bauru. 2022.

SILVA, LEONARDO WEBSTER RIBEIRO. **Estudo dos benefícios da utilização do Laravel Framework na manutenibilidade de software**. Monografia. INSTITUTO DOCTUM DE EDUCAÇÃO E TECNOLOGIA FACULDADES INTEGRADAS DE CARATINGA CIÊNCIA DA COMPUTAÇÃO. Minas Gerais, Caratinga. 2018.

SILVA, TAINAH EMMANUELE. **Um processo para administração de serviços em contêiners Docker**. Trabalho de Conclusão de Curso. Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande. Campina Grande. 2019.

SILVA, Walafi Ferreira da. **Utilizando virtualização baseada em containers para criação de laboratórios práticos de disciplinas na área de TI**. 2017. 44 f. Trabalho de Conclusão de Curso. Universidade Federal do Ceará, Campus Quixadá, Quixadá, CE, 2017.

SOARES, M.; KATARINA, DE M. C. H. **O segmento de distribuição de energia elétrica no Brasil: uma avaliação das crises hídricas enfrentadas em 2001 e 2021**. Conjecturas, v. 22, n. 2, p. 307–321, 2022.