



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

EDINHO DO NASCIMENTO DA SILVA

**Sistema Web de Suporte A Mobilidade
Multimodal em Smart Campus Usando
Algoritmos Baseados em Inteligência Artificial e
Análise Estatística**

DM: 26/2023

Belém-PA

2023

EDINHO DO NASCIMENTO DA SILVA

**Sistema Web de Suporte A Mobilidade
Multimodal em Smart Campus Usando
Algoritmos Baseados em Inteligência Artificial e
Análise Estatística**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, da Universidade Federal do Pará, como requisito para a obtenção de título de Mestre em Engenharia Elétrica.

Orientadora: Prof^a. Dr^a. Jasmine Priscyla Leite de Araújo

Belém-PA

2023

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a)
autor(a)

S586s Silva, Edinho do Nascimento da.
Um Sistema Web De Suporte A Mobilidade Multimodal
Em Smart Campus Usando Algoritmos Baseados Em
Inteligência Artificial E Análise Estatística / Edinho do
Nascimento da Silva. — 2019.
85 f. : il. color.

Orientador(a): Prof^a. Dra. Jasmine Priscyla Leite de
Araújo
Dissertação (Mestrado) - Universidade Federal do Pará,
Instituto de Tecnologia, Programa de Pós-Graduação em
Engenharia Elétrica, Belém, 2019.

1. smart campus. 2. estatísticas. 3. rede neural. 4.
transporte público multimodal. 5. modal elétrico. I. Título.

CDD 006.3

EDINHO DO NASCIMENTO DA SILVA

Sistema Web de Suporte A Mobilidade Multimodal em Smart Campus Usando Algoritmos Baseados em Inteligência Artificial e Análise Estatística

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, da Universidade Federal do Pará, como requisito para a obtenção de título de Mestre em Engenharia Elétrica.

Aprovada em: 16/08/2023.

Prof^a. Dr^a. Jasmine Priscyla Leite de Araújo
Orientador - PPGEE/UFPA

Prof. Dr. Fabricio José Brito Barros
Membro interno - PPGEE/UFPA

Prof. Dr. Fabrício de Souza Farias
Membro externo - PPCA/UFPA

Visto:

Prof. Dr. Diego Lisboa Cardoso
Coordenador do PPGEE/UFPA

Agradecimentos

Agradeço primeiramente a Deus por ter me concedido saúde e sabedoria para lidar com todos desafios ao longo dessa jornada. A Professora Jasmine por me aceitar como orientando no programa de pós graduação e por toda orientação, ajuda, compreensão e apoio durante o programa. A todos os integrantes do Laboratório de Computação de Telecomunicações que me receberam ano passado e ajudaram na transição vida ao vir do interior para morar em Belém.

*Nossa maior fraqueza está em desistir.
O caminho mais certo de vencer é tentar mais uma vez.*

Thomas Edison

Abstract

Urban mobility in a large city largely depends on public transport, which plays a vital role in facilitating the flow of citizens between different areas where a variety of goods and services, such as health and physical activities, are available. According to the UN, it is expected that more than 60% of the world's population will live in urban areas by 2050, which increases the need to make the entire transport system more sustainable and energy efficient. For this, the transition from modes powered by fossil fuels to options powered by electricity is a watershed that will require the implementation of new infrastructure to provide electricity to future modes. Such installation of energy points, called points of electric recharge of electric modes, is a technical, economic and energetic challenge. To discuss this current and relevant topic, this paper presents the results obtained from the Amazon Multimodal Intelligent System Project (SIMA), developed in partnership between the company Norte Energia and the Federal University of Pará (UFPA), focusing on the implementation at UFPA's Smart Campus in Belém. In possession of the implementation of the structure, this branch of the project, made use of the infrastructure project for the development of a software-based solution that serves as support in decision-making on the management of the energy efficiency of an intelligent environment, that is, a Smart Campus. In view of the above, this work has as its main focus the design and development of an energy efficiency management system that converts the data generated by electric modes in order to be useful in the management of the energy efficiency of the entire multimodal system. For this, several algorithms were implemented in the software, namely: linear regression, general regression neural network and moving average. From this work it is possible to conclude that the tool can be useful to help modal managers to achieve reductions in energy consumption, consequently, reducing operating costs and increasing equipment longevity, which will also impact capital costs.

Key-words: Smart campus, statistics, neural network, multimodal public transport, electric modal.

Resumo

A mobilidade urbana de uma grande cidade depende em grande parte do transporte público, que desempenha um papel vital ao facilitar o fluxo de cidadãos entre diferentes áreas onde uma variedade de bens e serviços, como saúde e atividades físicas, estão disponíveis. Segundo a ONU, espera-se que mais de 60% da população mundial resida em áreas urbanas até 2050, o que aumenta a necessidade de tornar todo o sistema de transporte mais sustentável e eficiente energeticamente. Para isso, a transição de modais movidos a combustíveis fósseis para opções alimentadas por energia elétrica é um divisor de águas que irá requerer a implantação de nova infraestrutura para provimento de energia elétrica aos futuros modais. Tal instalação de pontos de energia, denominados de pontos de recarga elétrica de modais elétricos é um desafio técnico, econômico e energético. Para discutir sobre esse tema atual e relevante, este trabalho apresenta os resultados obtidos a partir do Projeto de Sistema Inteligente Multimodal da Amazônia (SIMA), desenvolvido em parceria entre a empresa Norte Energia e a Universidade Federal do Pará (UFPA), com foco na implementação do *Smart Campus* da UFPA, em Belém do Pará. De posse da implantação da estrutura, essa ramificação do projeto, fez uso do projeto de infraestrutura para o desenvolvimento de uma solução baseada em software que sirva de suporte na tomada de decisões sobre a gestão da eficiência energética de um ambiente inteligente, isto é, um *Smart Campus*. Diante do exposto, este trabalho tem como foco principal projetar e desenvolver um sistema de gestão da eficiência energética que convertam os dados gerados por modais elétricos com o intuito de serem úteis na gestão da eficiência energética de todo o sistema multimodal. Para isso diversos algoritmos foram implementados no software, sendo eles: regressão linear, rede neural de regressão geral e média móvel. A partir deste trabalho é possível concluir que a ferramenta pode ser útil para auxiliar gestores dos modais a alcançarem reduções no consumo de energia, por consequência, reduzindo custos de operação e aumentando a longevidade de equipamentos, o que também impactará em custos de capital.

Palavras-chave: Smart campus, estatísticas, rede neural, transporte público multimodal, modal elétrico.

Lista de ilustrações

| | |
|----------------------------------------------------------------------------------------------------------------------------|----|
| Figura 1 – Conectividade do conceito de Internet das Coisas | 21 |
| Figura 2 – <i>Internet of Things</i> (IoT) aplicado a <i>Smart Home</i> | 22 |
| Figura 3 – IoT em <i>Smart Cities</i> | 23 |
| Figura 4 – Aplicação de conceito de <i>Smart Campus</i> | 25 |
| Figura 5 – Exemplo de sistema de segurança em um <i>Smart Campus</i> | 26 |
| Figura 6 – Arquitetura da Dojot | 27 |
| Figura 7 – Arquitetura <i>Apache Kafka</i> | 28 |
| Figura 8 – Modelo da Arquitetura multi-inquilino. | 29 |
| Figura 9 – Arquitetura do <i>Persister</i> | 30 |
| Figura 10 – Arquitetura do <i>History</i> | 31 |
| Figura 11 – Tela de <i>login</i> da plataforma Dojot. | 32 |
| Figura 12 – Tela de criação de <i>Template</i> | 33 |
| Figura 13 – Tela de criação do <i>Device</i> | 34 |
| Figura 14 – Exemplos de Entidade no modelo Diagrama Entidade-Relacionamento (DER). | 35 |
| Figura 15 – Exemplos de relacionamentos no modelo DER. | 36 |
| Figura 16 – Exemplos de relacionamentos e cardinalidades no modelo DER. | 37 |
| Figura 17 – Visão geral do modelo lógico do banco de dados | 48 |
| Figura 18 – Tabelas que são armazenadas informações dos campi da Universidade Federal do Pará (UFPA) | 49 |
| Figura 19 – Tabelas que são armazenadas os pontos de geolocalização de cada campus da UFPA e os seus mapeamentos | 50 |
| Figura 20 – Tabelas de usuário e gamificação | 51 |
| Figura 21 – Tabelas de favorito | 52 |
| Figura 22 – Tabelas de equipamentos | 53 |
| Figura 23 – Tela de <i>Login</i> | 55 |
| Figura 24 – Tela de Cadastro de novo usuário | 55 |
| Figura 25 – Recebimento de <i>email</i> de confirmação | 56 |
| Figura 26 – Tela de modais para usuários que não realizaram <i>login</i> | 56 |
| Figura 27 – Tela de recuperação de senha (a) e recebimento do <i>email</i> (b) | 57 |
| Figura 28 – Tela principal | 58 |
| Figura 29 – Alocação dos modais ativos (a) e Informações em tempo real (b) | 58 |
| Figura 30 – Definindo os pontos inicial e final para a busca de rota | 59 |
| Figura 31 – Tipos de percurso disponíveis sendo: ônibus (a), a pé (b) e barco (c) | 60 |
| Figura 32 – Tela de rotas favoritas | 61 |
| Figura 33 – Tela de estatística Geral | 62 |
| Figura 34 – Tela de média móvel | 62 |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 35 – Tela de regressão linear | 63 |
| Figura 36 – Tela da GRNN | 64 |
| Figura 37 – Tela de gerenciamento de usuário | 65 |
| Figura 38 – Tela de gerenciamento de perfil | 66 |
| Figura 39 – Tela de alterar a senha | 66 |
| Figura 40 – Tela de alterar dados do perfil | 67 |
| Figura 41 – Vista aérea do campus da UFPA-Belém destacando os campi Básico em amarelo, Profissional em laranja e o Betina em verde | 68 |
| Figura 42 – Ônibus Circular em funcionamento | 69 |
| Figura 43 – Percurso do Ônibus Circular | 70 |
| Figura 44 – Parque de geração fotovoltaica no Centro de Excelência em Eficiência Energética da Amazônia (CEAMAZON) | 71 |
| Figura 45 – Eletroposto localizado no Ginásio da UFPA | 71 |
| Figura 46 – Ônibus Urbano e Rodoviário | 72 |
| Figura 47 – Esquema de captura e envio de dados do ônibus Circular | 73 |
| Figura 48 – Esquema de captura e envio de dados do ônibus rodoviário | 73 |
| Figura 49 – Device criado na Dojot | 75 |
| Figura 50 – Cenário de média (estatísticas gerais) | 77 |
| Figura 51 – Cenário de Média Móvel | 78 |
| Figura 52 – Cenário de regressão linear | 79 |

Lista de tabelas

| | |
|-----------------------------------------------------------|----|
| Tabela 1 – Exemplo de requisitos funcionais | 38 |
| Tabela 2 – Exemplo de requisitos não funcionais | 38 |
| Tabela 3 – Dispositivos presentes no Modal 01 | 45 |
| Tabela 4 – Requisitos Funcionais | 45 |
| Tabela 5 – Requisitos Não Funcionais | 45 |
| Tabela 6 – Criação de variáveis do simulador | 74 |

Lista de abreviaturas e siglas

| | |
|-----------------|-----------------------------------------------------------|
| ANB | <i>Adaptive Naïve Bayes</i> |
| API | <i>Application Programming Interface</i> |
| ATC | <i>American Tower Corporation</i> |
| ATR | Ativação do Temporizador Regressivo |
| CEAMAZON | Centro de Excelência em Eficiência Energética da Amazônia |
| CO2 | Dióxido de Carbono |
| CoAP | <i>Constrained Application Protocol</i> |
| CPQD | Centro de Pesquisa e Desenvolvimento em Telecomunicações |
| CS | <i>Compressive Sending</i> |
| DER | Diagrama Entidade-Relacionamento |
| DM | <i>Data Mining</i> |
| DMC | Desativação do Modo Colaborativo |
| EMA | <i>Exponential Moving Average</i> |
| ER | Entidade-Relacionamento |
| FV's | Frutas e Vegetais |
| GPS | <i>Global Positioning System</i> |
| GRNN | <i>General Regression Neural Network</i> |
| GSM | <i>Global System for Mobile Communications 2G</i> |
| HTTP | <i>HyperText Transfer Protocol</i> |
| IA | Inteligência Artificial |
| ID | Número de Identificação |
| IoT | <i>Internet of Things</i> |
| JSON | <i>JavaScript Object Notation</i> |
| KNN | <i>K-Nearest Neighbors</i> |
| LoRa | <i>Long Range</i> |
| LPWANs | <i>Low Power Wide Area Networks</i> |
| LWM2M | <i>Lightweight M2M</i> |
| M2M | <i>Machine to Machine</i> |

| | |
|-----------------|--------------------------------------------------------------------|
| MM | Média Móvel |
| MQTT | <i>Message Queuing Telemetry Transport</i> |
| ONU | Organização das Nações Unidas |
| RNA | Rede Neural Artificial |
| SGBD | Sistema de Gerenciamento de Banco de Dados |
| LPWANs | <i>Low Power Wide Area Networks</i> |
| SIMA | Sistema Inteligente Multimodal da Amazônia |
| SISPGEER | Sistema de Planejamento da Gestão da Eficiência Energética e Rotas |
| SMA | <i>Simple Moving Average</i> |
| TI | Tecnologia de Informação |
| UFPA | Universidade Federal do Pará |
| UML | <i>Unified Modelling Language</i> |
| Wi-Fi | <i>Wireless Fidelity</i> |
| XGBoost | <i>Extreme Gradient Boosting</i> |

Sumário

| | | |
|----------|---------------------------------------------------------------|-----------|
| 1 | Introdução | 16 |
| 1.1 | Contextualização | 16 |
| 1.2 | Motivação e Justificativa | 18 |
| 1.3 | Objetivos | 19 |
| 1.3.1 | Geral | 19 |
| 1.3.2 | Específicos | 20 |
| 1.4 | Estrutura do Trabalho | 20 |
| 2 | Referencial Teórico | 21 |
| 2.1 | Introdução | 21 |
| 2.2 | Internet das Coisas | 21 |
| 2.3 | <i>Smart Campus</i> | 24 |
| 2.4 | Banco de Dados em Tempo Real Dojot | 26 |
| 2.4.1 | Introdução | 26 |
| 2.4.2 | Arquitetura da Dojot | 26 |
| 2.4.2.1 | <i>Kafka</i> | 27 |
| 2.4.2.2 | <i>IoT Agent</i> | 29 |
| 2.4.2.3 | <i>Persiste/History</i> | 29 |
| 2.4.2.4 | <i>Kong</i> | 31 |
| 2.4.3 | Ambiente Dojot | 31 |
| 2.4.3.1 | <i>Login</i> | 31 |
| 2.4.3.2 | <i>Templates</i> | 32 |
| 2.4.3.3 | <i>Devices</i> | 33 |
| 2.5 | Banco de Dados Relacional | 34 |
| 2.6 | Requisitos | 37 |
| 2.7 | Média Móvel | 38 |
| 2.8 | Regressão Linear | 40 |
| 2.9 | Rede Neural de Regressão Geral | 42 |
| 3 | Metodologia e Ferramentas Utilizadas | 44 |
| 3.1 | Introdução | 44 |
| 3.2 | Metodologia | 44 |
| 3.3 | Ferramentas Utilizadas | 46 |
| 3.3.1 | Ferramentas <i>Backend</i> , Banco de Dados e <i>Frontend</i> | 46 |
| 3.3.1.1 | <i>Backend</i> | 46 |
| 3.3.2 | Banco de Dados Relacional | 47 |
| 3.3.3 | Visualização Para O Usuário | 47 |
| 3.4 | Modelagem Do Banco de Dados | 47 |

| | | |
|----------|--------------------------------------------------------------------------------------|-----------|
| 3.4.1 | Campus | 48 |
| 3.4.2 | Criação De Pontos De Interesse No Campus e Rotas | 49 |
| 3.4.3 | Usuário | 51 |
| 3.4.4 | Favoritos | 51 |
| 3.4.5 | Equipamento | 52 |
| 4 | Sistema De Planejamento Da Gestão Da Eficiência Energética e Rotas (SISPGEER) | 54 |
| 4.1 | Login | 54 |
| 4.2 | Tela Principal | 57 |
| 4.3 | Estatísticas | 61 |
| 4.4 | Gerenciamento de Usuários | 64 |
| 4.5 | Gerenciamento de Perfil | 65 |
| 4.6 | Implantação da Solução | 67 |
| 4.6.1 | Projeto SIMA | 70 |
| 4.7 | Cenários de Teste | 74 |
| 5 | Resultados | 77 |
| 5.1 | Cenário 1 - Média | 77 |
| 5.2 | Cenário 2 - Média Móvel | 78 |
| 5.3 | Cenário 3 - Regressão Linear | 78 |
| 6 | Conclusão | 80 |
| 6.1 | Trabalhos Futuros | 80 |
| | Referências | 81 |

1 Introdução

1.1 Contextualização

O transporte público é umas das áreas vitais na mobilidade urbana de uma grande cidade, pois garante o fluxo de cidadãos entre áreas da cidade que oferecem diferentes tipos de bens e diversos tipos de serviços como saúde, atividades físicas entre outros [Saif e MM-Torok 2018]. Um dos principais propósitos do transporte público é minimizar congestionamentos no trânsito oferecendo um serviço de baixo custo e democrático. Por essa razão, conforme a população aumenta esse serviço se torna cada vez mais essencial.

De acordo com o relatório da Organização das Nações Unidas (ONU), mais de 60% da população mundial irá residir em áreas urbanas até 2050, fazendo com que todo o sistema de transporte tenha que se tornar mais sustentável e eficiente [Saraiva e ONU 2022]. A respeito da sustentabilidade, atualmente um dos principais focos trata da troca dos modais que funcionam a partir de combustíveis fósseis para modais elétricos, que por sua vez, podem oferecer o mesmo serviço, mas diminuindo a poluição do ar, poluição sonora, e doenças causadas pelos gases gerados na combustão [Saif e MM-Torok 2018]. Porém trocar modais movidos a combustíveis fósseis para elétricos não é uma tarefa fácil, pois precisaria de uma nova estrutura envolvendo a geração de energia, pontos de recarga e novos veículos [Foley, Degirmenci e Yigitcanlar 2020], o que pode assustar o operador com os custos iniciais, porém podem ser compensados nos custos de operação como manutenção da frota, custo por quilômetro rodado e o custo da aquisição da matriz energética, quando comparado aos combustíveis fósseis [Say et al. 2023].

Porém aplicar um projeto em grandes cidades é uma tarefa complexa, por essa razão os espaços das universidades se tornam um cenário ideal para a implantação de protótipos, em pequena escala, que possam representar miniaturas de cenário reais, podendo ser coletados dados como: temperatura do ambiente, clima, vegetação, dinâmica populacional, infraestruturas de transporte [Negreiros et al. 2020].

Dentre os projetos que tratam do planejamento do modo em que usamos o transporte público, destacamos o Projeto Sistema Inteligente Multimodal da Amazônia (SIMA), desenvolvido em parceria entre a empresa Norte Energia e a UFPA, com a ativa implementação do Campus da UFPA em Belém. O projeto tem como objetivo desenvolver um modelo de implantação de mobilidade elétrica em cenário amazônico, o qual poderá ser replicado em outros cenários. Para isso, foi criado um parque de geração fotovoltaica e foi adotado o uso de dois modais elétricos, sendo dois ônibus e um barco [LOBATO 2022].

Para monitorar o funcionamento dos modais, foram instalados diversos sensores que capturam dados de uso dos modais como por exemplo: temperatura, localização, quantidade

de passageiros, isto é, transformando a UFPA em um *Smart Campus*. Com a modernização do ambiente, se tornando um *Smart Campus*, surge o desafio de fazer um sistema que possibilite o uso dos dados gerados e provenientes dos modais elétricos, sendo antes, tais dados negligenciados e descartados. Embora descartados, os dados podem ser ferramentas importantes para a tomada de decisão, e por consequência, para a garantia de uma gestão da eficiência energética de um ambiente controlado por sensores. Além disso, surge o desafio de se adotar no sistema a aplicação de algoritmos de inteligência artificial e aprendizagem de máquina, os quais podem servir como técnicas capazes de auxiliar na manutenção da qualidade do serviço ofertado para a população. Também, podendo servir como ferramentas que otimizem, por exemplo, o tempo de vida de equipamentos sob monitoramento.

No que diz respeito a inteligência artificial e aprendizagem de máquina, diversos trabalhos têm apresentado soluções que aplicam tais algoritmos com o propósito de dar suporte na tomada de decisões.

Como no trabalho de [Afreem e Bajwa 2021](#), onde propuseram um sistema de monitoramento e notificação inteligente em tempo real em locais de armazenamentos de Frutas e Vegetais (FV's), utilizando IoT para avaliar diferentes variáveis e analisar parâmetros na qualidade do ambiente de armazenamento do pós ciclo de colheitas de frutas e vegetais. Para isto, utilizaram sensores como: temperatura, umidade, luminosidade e detecção de Dióxido de Carbono (CO₂); para a coleta de dados conectados a um módulo Arduino. A pesquisa também se preocupou em fazer uma análise de comparação com soluções existentes para o monitoramento de ambientes visando estocagem de FV's.

Neste trabalho ainda foi desenvolvido um aplicativo *Android* para notificar o usuário das variações dos parâmetros do local monitorado. Para isso, os autores implementaram uma Rede Neural Artificial (RNA) e puderam classificar o ambiente em três classes, ou seja, bom, insatisfatório ou alarmante. Segundo os autores, foram obtidos resultados com 99% de precisão usando o modelo de rede neural de propagação direta, enquanto os modelos existentes, como *Compressive Sending* (CS), *Adaptive Naïve Bayes* (ANB), *Extreme Gradient Boosting* (XGBoost), *Data Mining* (DM) retornaram as respectivas precisões, 95,60%, 87,50%, 93,59%, 90% [[Afreem e Bajwa 2021](#)]. Embora os autores tenham obtidos resultados positivos, não foi possível identificar um uso de uma plataforma robusta para compor toda a parte sistêmica de *hardware* e *software*, nem ao menos se trabalhou com uma topologia virtualizada, *containers*, ou com a utilização de serviços como Dojot.

Em [Melo et al. 2022](#), foi apresentada a proposta uma solução baseada em Inteligência Artificial (IA) para classificação de um ambiente de *smart home*, monitorando variáveis ambientais através de sensores, como: luminosidade, temperatura e umidade. Para tanto foi implementado uma rede IoT para para um monitoramento em tempo real de um quarto e uma sala de estar de uma residência. Com os dados coletados, foi passado para as etapas de seleção, pré-processamento, transformação e apresentação ao algoritmo de IA baseado em árvore de

decisão. Segundo os autores, os resultados indicam que é possível usar dados coletados de uma rede IoT para monitorar a qualidade de determinado ambiente por meio da aplicação de IA.

Como contribuição principal, foi obtido uma eficácia da IA em conjunto com a árvore de decisão na previsão de condições ambientais em residências, a partir dos acertos alcançados nos dados coletados [Melo et al. 2022]. Neste trabalho é explorado uma proposta para ambientes residenciais e a utilização da plataforma Dojot instalado no *Docker Compose*, utilizando apenas um dispositivo IoT, no entanto propõe a instalação da Dojot em ambiente orquestrado pelo *Kubernetes*, e o cenário de estudo é em um campus universitário, levando em consideração sete dispositivos IoT com diversos sensores coletando diferentes parâmetros ambientais.

E no trabalho de Sa et al. 2019, são abordadas soluções baseadas em *software* e IA que visam auxiliar a população brasileira em sua rotina de deslocamento no trânsito. O estudo apresenta uma estratégia de monitoramento colaborativo online de rotas de ônibus, que não requer autorização de empresas privadas ou do governo local. Esta abordagem envolve a aplicação do algoritmo *K-Nearest Neighbors* (KNN) e a criação de dois novos algoritmos, denominados Ativação do Temporizador Regressivo (ATR) e Desativação do Modo Colaborativo (DMC). Esses algoritmos foram implementados no aplicativo "Olha o Ônibus" e submetidos a cinco testes com o objetivo de avaliar a eficácia do sistema.

Os resultados demonstraram que os algoritmos desenvolvidos conseguiram assegurar a confiabilidade das informações compartilhadas entre os usuários com uma precisão de 100% nos testes relacionados à verificação do cumprimento da rota e à detecção de períodos de inatividade dentro da rota, isso se deve ao fato de que o algoritmo KNN é capaz de corrigir todas as imprecisões decorrentes dos dados do *Global Positioning System* (GPS) [Sa et al. 2019]. Os autores obtiveram resultados satisfatórios e eficazes em sua pesquisa com base nessa abordagem. A infraestrutura de *hardware* foi avaliada usando tecnologia GPS em dispositivos móveis de usuários finais, coletando as rotas por meio de um aplicativo *Android*. As coordenadas foram enviadas para um servidor de banco de dados na nuvem e, em seguida, ajustadas e recuperadas por meio dos algoritmos propostos. Porém, os autores não apresentaram ou adotaram uma arquitetura específica de *software*, com servidor dedicado e utilização de aplicações em containers como Dojot para monitoramento, coleta e tratamento dos dados em dispositivos IoT em campo.

1.2 Motivação e Justificativa

A mobilidade urbana é um dos principais desafios enfrentados pelas cidades modernas. O aumento constante do número de veículos tem causado impactos negativos, tais como congestionamentos, poluição atmosférica e atrasos nas viagens. Nesse contexto, o sistema *web* de apoio ao Projeto SIMA se apresenta como uma alternativa capaz de integrar os modais elétricos de maneira inteligente. Ao fornecer informações em tempo real sobre a localização dos modais e a previsão de horários de chegada, o sistema busca viabilizar uma mobilidade mais eficiente e

fluida para os cidadãos, tornando o transporte público uma opção atraente e confiável.

Além disso, a crescente preocupação com a sustentabilidade ambiental impulsiona a busca por alternativas mais ecológicas e sustentáveis no transporte urbano. Os modais elétricos representam uma opção promissora nesse sentido, visto que contribuem para a redução das emissões de carbono e, conseqüentemente, para a preservação do meio ambiente. Ao unir esses modais em um sistema integrado e oferecer funcionalidades que otimizam sua operação, o sistema *web* de apoio tem como objetivo direto a promoção da sustentabilidade ambiental nas cidades, incentivando a adoção de meios de transporte mais limpos e ecológicos.

Outro fator determinante é o avanço das tecnologias de informação e comunicação, aliado à disponibilidade de grandes volumes de dados. Essas inovações possibilitam o desenvolvimento de soluções inteligentes para os desafios urbanos. A motivação por trás do sistema *web* de apoio ao Projeto SIMA é justamente aproveitar essas tecnologias e técnicas de análise de dados para fornecer informações precisas e valiosas. Por meio da análise estatística dos dados enviados pelos modais elétricos ao banco de dados Dojot, o sistema oferece *insights* relevantes tanto para os usuários dos modais elétricos quanto para os gestores do projeto. Essas informações podem ser utilizadas para melhorar a eficiência operacional dos modais, planejar manutenções com antecedência e embasar decisões estratégicas que otimizem o funcionamento do sistema como um todo.

Em síntese, a justificativa do projeto baseia-se em proporcionar uma solução abrangente e eficiente para os desafios da mobilidade urbana contemporânea. A integração dos modais elétricos por meio do sistema *web* de apoio visa aprimorar a experiência do usuário, promover a sustentabilidade ambiental e aproveitar os avanços tecnológicos para oferecer um sistema de transporte público moderno, inteligente e alinhado às necessidades das cidades do futuro. Ao enfrentar esses desafios de forma estratégica e inovadora, o projeto busca contribuir significativamente para a construção de cidades mais sustentáveis, conectadas e eficientes.

1.3 Objetivos

1.3.1 Geral

A implantação de um *Smart Campus* se dá pelo planejamento e configuração de uma rede IoT fim a fim, isto é, do dispositivo no campo até o servidor que armazena os dados coletados. Com isso, torna-se possível a transmissão e armazenamento de um grande volume de dados telemétricos oriundos de sensores presentes nos dispositivos implantados no campo, por exemplo em modais elétricos que auxiliam na mobilidade urbana. Tal implantação permite a persistência de dados que anteriormente eram negligenciados. Tais dados podem ser úteis nas tomadas de decisão de gestão da eficiência energética, por isso se faz necessário o desenvolvimento de soluções baseadas em *software* capazes de estratificar conhecimentos a partir do que é coletado.

Diante deste contexto, este trabalho tem como objetivo geral projetar e implementar um sistema *web* capaz de auxiliar no planejamento da gestão da eficiência energética utilizando diferentes algoritmos de inteligência artificial e os dados coletados a partir de determinada rede IoT implantada em um cenário de *Smart Campus*.

1.3.2 Específicos

Propor uma solução baseada em *software* compreensiva para gerar dados estatísticos a partir das informações coletadas pelos sensores localizados nos veículos elétricos.

Apresentar um *software* que use inteligência artificial e aprendizado de máquina para ajudar a tomada de decisões da gestão da eficiência energética. Ao exibir de forma gráfica a resposta do processamento dos algoritmos.

1.4 Estrutura do Trabalho

Este trabalho está sendo dividido no seguinte modo: O Capítulo 2 será dedicado ao referencial teórico dos principais temas utilizados, a fim de abordar conceitos imprescindíveis ao tema proposto.

No Capítulo 3 será destacado toda a abordagem metodológica utilizada para se chegar ao cumprimento do objetivo principal e dos objetivos específicos deste trabalho.

No Capítulo 4 será dado o enfoque ao desenvolvimento do sistema *web*, detalhando as suas principais funções e exibindo as telas disponíveis para o usuário. Além disso, é dado destaque na implantação do projeto e cenários de teste.

O Capítulo 5 dará destaque aos resultados obtidos em relação ao uso do sistema, sendo seguido do Capítulo 6, onde são apresentadas as conclusões e perspectivas de pesquisas futuras.

Por fim, será detalhado as referências utilizadas para a produção desta escrita.

2 Referencial Teórico

2.1 Introdução

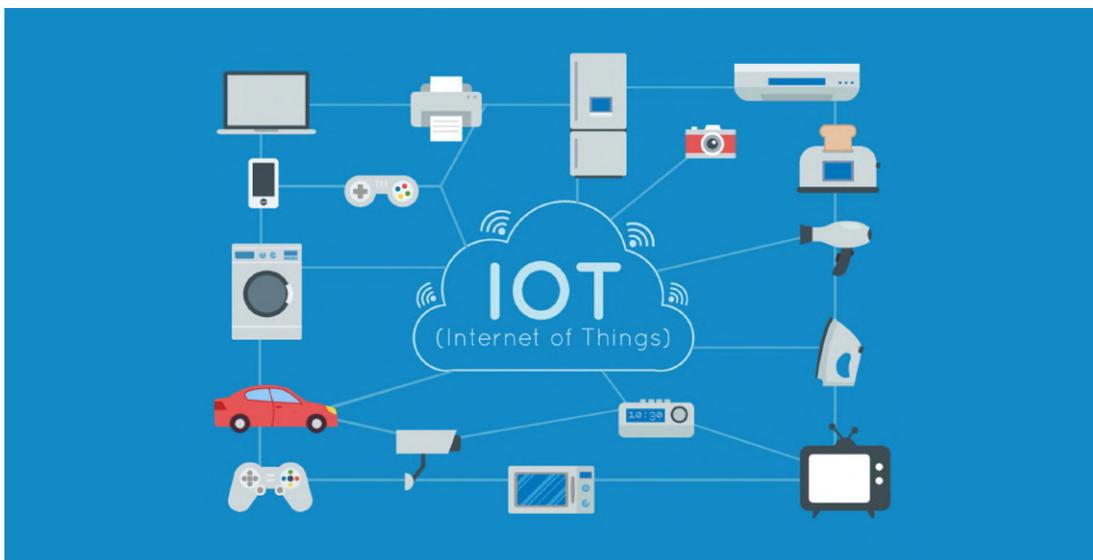
Neste capítulo, serão abordados os principais temas que auxiliaram no desenvolvimento da metodologia proposta, servindo como uma sólida base teórica para o pleno entendimento do mesmo. Além disso, apresentaremos uma descrição das tecnologias e métricas de desempenho empregadas ao longo deste trabalho de pesquisa. A análise dessas ferramentas e medidas de avaliação contribuirão para uma compreensão aprofundada dos resultados obtidos, permitindo conclusões sólidas sobre a eficácia e eficiência do trabalho.

2.2 Internet das Coisas

A Internet das Coisas (IoT) é um conceito que se refere à conexão de dispositivos físicos à Internet, permitindo que eles troquem dados e realizem tarefas de forma autônoma. Esses dispositivos podem ser desde objetos simples, como eletrodomésticos e sensores, até máquinas complexas, como veículos e equipamentos industriais [Khaled et al. 2018].

A ideia fundamental por trás da IoT é habilitar objetos do cotidiano para se comunicarem e interagirem entre si e com usuários, criando um ecossistema de dispositivos inteligentes interconectados, ilustrado na Figura 1. Esses dispositivos estão equipados com sensores, processadores e possuem conectividade com a rede, o que lhes permite coletar, compartilhar e analisar dados em tempo real [Plageras et al. 2018].

Figura 1 – Conectividade do conceito de Internet das Coisas



Fonte: [FirstTech 2023].

A IoT possui uma ampla gama de aplicações em diversos setores, incluindo transporte/mobilidade urbana, automação residencial, saúde, agricultura, indústria e cidades e campus inteligentes. Por exemplo, no setor de transporte/mobilidade urbana, a IoT tem um papel fundamental na criação de veículos autônomos e na melhoria da eficiência logística. Para isso, sensores e dispositivos IoT são integrados aos automóveis, que por sua vez, podem coletar dados sobre o desempenho dos veículos, condições de tráfego e hábitos dos motoristas, deste modo, possibilitando a condução mais segura e a implementação de soluções de transporte inteligentes [Muñoz et al. 2018].

Em outro nicho, na automação residencial, com a implantação da IoT é possível controlar remotamente a iluminação, temperatura, segurança e eletrodomésticos por meio de dispositivos conectados à *Internet*, tais como *smartphones* ou assistentes virtuais [Anas, Singh e Kamarudin 2022, Melo et al. 2022, Plageras et al. 2018]. A Figura 2 ilustra um cenário de *Smart Home* sendo acessado a partir de um aplicativo para *smartphone*.

Figura 2 – IoT aplicado a *Smart Home*



Fonte: [SYSTEMS 2023]

Além disso, a IoT desempenha um papel fundamental no desenvolvimento de cidades inteligentes, isto é, quando sensores e dispositivos IoT são integrados à infraestrutura urbana para monitorar o tráfego, o consumo de energia, a qualidade do ar e a segurança pública, exibido na Figura 3. Com essas informações, é possível otimizar o uso de recursos, melhorar o planejamento urbano e oferecer serviços [Fraga-Lamas et al. 2019].

Figura 3 – IoT em Smart Cities



Fonte: [FATECNOLOGIA 2019]

A infraestrutura de uma rede IoT é fundamental para a operação e comunicação eficiente de dispositivos interconectados. Ela consiste em vários componentes-chave que permitem a coleta, o processamento e o compartilhamento de dados entre os dispositivos IoT. Esses componentes-chave abrangem diversos conceitos como: sensores, *gateway*, plataformas, aplicações [Domínguez-Bolaño et al. 2022]. Esses conceitos serão detalhados a seguir.

Sensores ou *endnodes* são dispositivos físicos que capturam dados ou ações do mundo real como: humidade, temperatura, movimento, proximidade, nível de líquidos, corrente, tensão. São dispositivos pequenos, possuem recursos limitados e produzidos para uma atividade específica [Domínguez-Bolaño et al. 2022].

Gateway é um dispositivo físico que realiza a intermediação de comunicação entre os *endnodes* e a plataforma IoT, podendo suportar diversos tipos de comunicação como redes sem fio (*Wireless Fidelity* (Wi-Fi), *Bluetooth*, *Long Range* (LoRa)), redes de comunicação móvel (4g, 5g), *ethernet*. Além de suportar diversos tipos de dados e de camadas de aplicação. O *gateway* possui uma área de cobertura onde um ou mais *endnodes* possam ser conectados, extraindo seus dados e enviando a uma plataforma IoT [Domínguez-Bolaño et al. 2022].

Plataforma IoT é um sistema que funciona como um *hub* para o sistema IoT. Na plataforma fica como responsabilidade as tarefas de coleta, armazenamento e processamento dos dados vindos dos *endnodes*; fornecendo funcionalidades como gerenciamento dos *endnodes*, segurança, análise de dados e a habilitação do uso por aplicações. A plataforma IoT pode ser em servidores na nuvem, local ou uma combinação entre local-nuvm [Domínguez-Bolaño et al. 2022].

Aplicações são *softwares* externos que utilizam os dados nas plataformas para realizar

tarefas específicas de leitura ou controle de atuadores, por exemplo: uma interface gráfica que exibe os dados de um sensor de temperatura e a sua evolução conforme o tempo, permitir que o usuário possa ligar/desligar alguns dispositivos usando como base os dados armazenados, emitir alertas como: fumaça em um ambiente e inundação ao ultrapassar algum limite estabelecido [Domínguez-Bolaño et al. 2022].

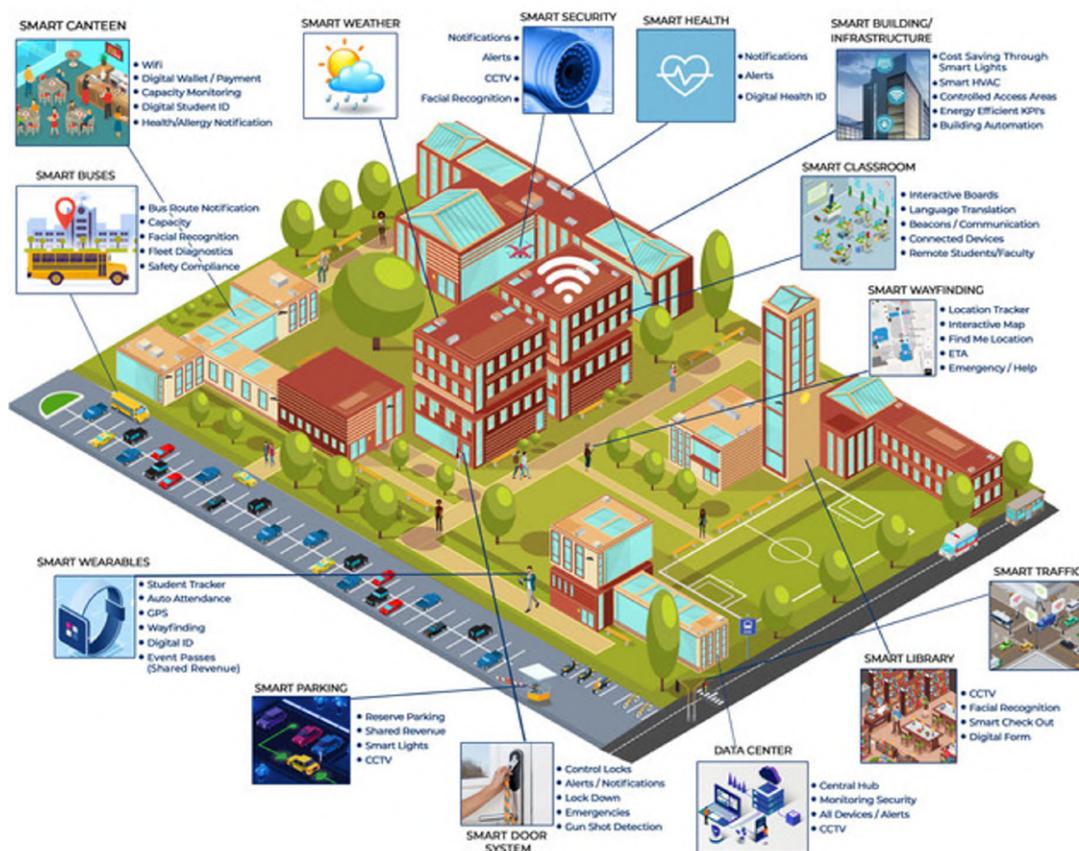
Em resumo, a infraestrutura de uma rede IoT é composta por uma série de elementos interconectados que trabalham juntos para possibilitar a coleta, o processamento e o uso de dados gerados por dispositivos interconectados. Cada componente desempenha um papel fundamental na operação eficiente da rede IoT.

2.3 *Smart Campus*

Um *Smart Campus* é um ambiente universitário ou educacional que utiliza tecnologias avançadas para melhorar a eficiência operacional, o desempenho acadêmico e a qualidade de vida dos estudantes, professores e funcionários.

Ao integrar infraestrutura física composta por sistemas de Tecnologia de Informação (TI) e dispositivos inteligentes, um *Smart Campus* oferece aos seus usuários um ambiente conectado, seguro e sustentável, impulsionado pela rede IoT e pela capacidade de análise e processamento de dados gerados [Fraga-Lamas et al. 2019].

No coração de um *Smart Campus* estão os sensores e dispositivos IoT, os quais são implantados em toda a área para coletar informações em tempo real. Esses dispositivos podem incluir sensores de temperatura, umidade, iluminação, presença, qualidade do ar, ruído, consumo de energia, entre outros. De acordo com Fraga-Lamas et al. 2019, tais dados são coletados e analisados para fornecer *insights* valiosos sobre o funcionamento do campus e apoio à tomada de decisões. Um exemplo de aplicação de *Smart Campus* é ilustrado na Figura 4.

Figura 4 – Aplicação de conceito de *Smart Campus*

Fonte: [CLOVITY 2020].

Um aspecto importante de um *Smart Campus* é a infraestrutura de rede robusta e de alta velocidade, que permite a comunicação e a troca de dados entre os dispositivos conectados. Isso pode envolver a instalação de redes sem fio de alta densidade, como Wi-Fi e redes de área ampla de baixo consumo de energia (*Low Power Wide Area Networks (LPWANs)*), para cobrir toda a extensão do campus [Fraga-Lamas et al. 2019].

Deste modo, um *Smart Campus* oferece benefícios significativos para todos os envolvidos na comunidade educacional, por exemplo, para estudantes essa implantação pode significar uma experiência única de aprendizado técnico, a partir da obtenção de informações e dados antes descartados.

Além disso, o *Smart Campus* pode impactar na segurança no campus, podendo essa reforçada com a implementação de sistemas de câmera de vigilância inteligentes e de novas tecnologias de controle de acesso. Câmeras de segurança, sensores de movimento e análise de vídeo podem ajudar a monitorar áreas críticas, identificar comportamentos suspeitos e responder rapidamente a incidentes de segurança, mostrado na Figura 5. Com isso, os estudantes também podem se beneficiar de recursos de segurança pessoal, como aplicativos de segurança que podem enviar alertas em caso de emergência.

Figura 5 – Exemplo de sistema de segurança em um *Smart Campus*

Fonte: [ARDTECHS 2020]

2.4 Banco de Dados em Tempo Real Dojot

2.4.1 Introdução

A plataforma Dojot foi desenvolvida pelo Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPQD) com o lançamento oficial realizado no evento *IoT Latin America* em 2017.

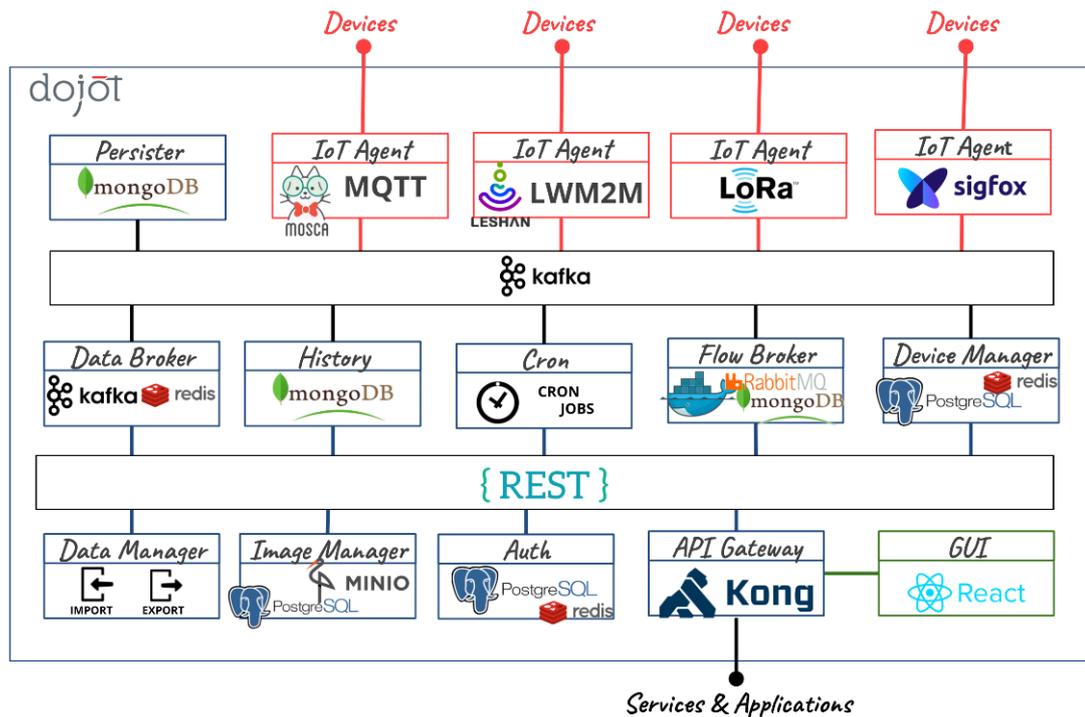
A função da Dojot é acelerar o desenvolvimento de tecnologias aplicadas a IoT adaptadas às necessidades brasileiras em diferentes áreas como por exemplo: *Smart Cities*, *Smart Campus*, Agronegócio, entre outros. E para incentivar a adoção por parte de empresas públicas e privadas, *startups* e instituições de ensino, a plataforma foi desenvolvida utilizando ferramentas *open source* e com código aberto [CPQD 2020].

2.4.2 Arquitetura da Dojot

Esta subseção apresenta os principais elementos da arquitetura Dojot que foram utilizados neste trabalho, sendo este conteúdo baseado na documentação do ecossistema [DOCS 2020].

A arquitetura do Dojot está ilustrada na Figura 6 e utiliza o padrão de microsserviços, ou seja, o *software* possui pequenos serviços que são independentes, sendo a comunicação entre eles feita por meio de Interfaces de Programação de Aplicação (API, do inglês *Application Programming Interface*) específicas, assim facilitando a escalabilidade, agilizando o desenvolvimento de aplicativos e facilitando a implementação de novos recursos [AmazonMicroservicos 2020].

Figura 6 – Arquitetura da Dojot

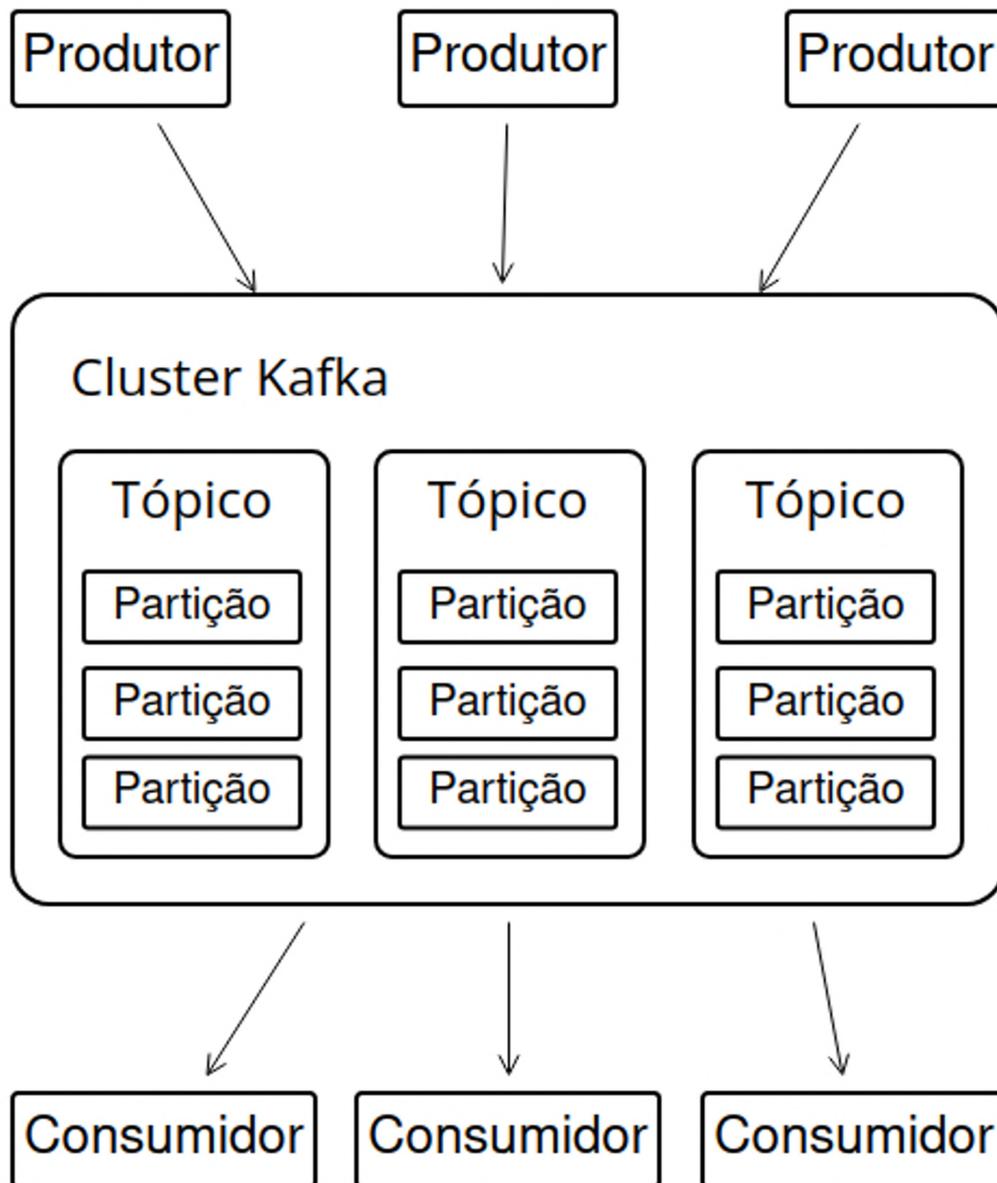


Fonte: [DOCS 2020]

Para este trabalho, serão explicados alguns desses microsserviços como: *Kafka*, *IoT Agent*, *Persister/History* e *Kong*.

2.4.2.1 *Kafka*

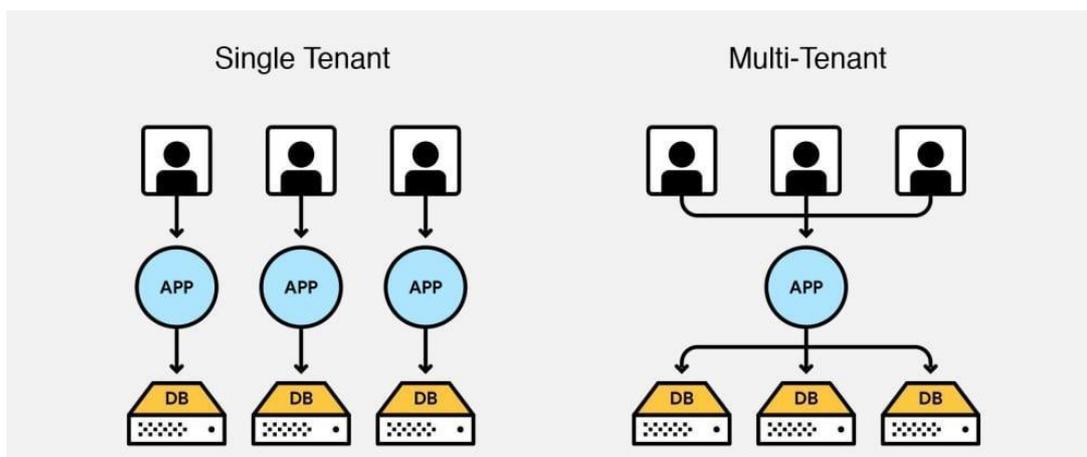
Apache Kafka é uma plataforma distribuidora de mensagens que pode ser utilizada em aplicações que possuam transmissão de dados (*streaming*), permitindo uma comunicação pouco acoplada, ou seja, quem envia uma mensagem pelo *Kafka* (produtor) não sabe quais são os componentes que irão recebê-la, assim como o componente que irá consumir (consumidor) a mensagem não sabe quem foi que mandou [Kafka 2020], como ilustrado na Figura 7.

Figura 7 – Arquitetura *Apache Kafka*.

Fonte: [Kafka 2020].

Deste modo, garantindo o isolamento dos dados de usuário e das aplicações, o que viabiliza a arquitetura de multi-inquilino (do inglês, *multi-tenant*), isto é, uma arquitetura de *software* no qual uma instância de *software* ou servidor é capaz de atender a vários usuários ou clientes, mas com bancos de dados separados para cada um, como ilustrado na Figura 8. Tal funcionalidade permite a segregação dos dados entre os usuários, assim como, o gerenciamento e personalização de acordo com a viabilidade de cada usuário sem alterar o código fonte, assim maximizando a eficiência e reduzindo o custo para implementação de atualizações ou manutenções, pois as edições são aplicadas a todos os usuários e geralmente gerenciadas pela empresa responsável [LFTECHNOLOGY 2019].

Figura 8 – Modelo da Arquitetura multi-inquilino.



Fonte: [DEV.TO 2020].

2.4.2.2 IoT Agent

É como um “*driver*” responsável pela comunicação de dispositivos externos como um Arduino¹, sensores variados, entre outros com a Dojot, possibilitando que a plataforma tenha um conjunto de dispositivos, sendo cada um especializado em um protocolo como: MQTT/JSON², CoAP/LWM2M³, LoRa/ATC⁴, Sigfox/WDN⁵ e/ou HTTP/JSON⁶.

2.4.2.3 Persister/History

O *Persister* tem como objetivo receber os dados e eventos dos dispositivos e fazer o cadastro dos mesmos no banco de dados MongoDB, como ilustrado na Figura 9. Sua arquitetura é baseada nos dispositivos físicos que enviam os dados para a plataforma Dojot através do *IoT Agent*. Os dados são recebidos pelo *Kafka* que redireciona para o *Persister*, em seguida, o *Persister* faz a comunicação com o MongoDB para a realização do cadastro no banco.

¹ Arduino é uma plataforma programável de prototipagem eletrônica de placa única e *hardware* livre, usando o microcontrolador Atmel ou ARM com suporte de entrada/saída embutido tanto digital quanto analógico, sendo útil para projetos de automação, robótica e sensoriamento remoto [Arduino 2023].

² O *Message Queuing Telemetry Transport* (MQTT) é um protocolo popular para transmissão de dados IoT criado em 1999, inicialmente para dispositivos com restrição de *hardware* e redes de alta latência e/ou baixa banda, no setor de óleo e gás. Sua arquitetura baseada em *Machine to Machine* (M2M) é leve, segura, confiável, escalável e eficiente. Hoje, especificado para aplicações de mensageiria, suporta vários formatos de dados, como texto, binários, arquivos XML ou JSON [Nunes 2018, AWS 2023].

³ O *Constrained Application Protocol* (CoAP) foi projetado para permitir a troca de mensagens diretamente entre dispositivos interconectados por redes com baixas taxas de transferência. O modelo de comunicação *Lightweight M2M* (LWM2M) é baseado em métodos CoAP como *GET*, *PUT*, *POST* e *DELETE* sobre UDP ou SMS como camada de transporte. [Sztajnberg, Macedo e Stutzel 2018, Solutions 2023]

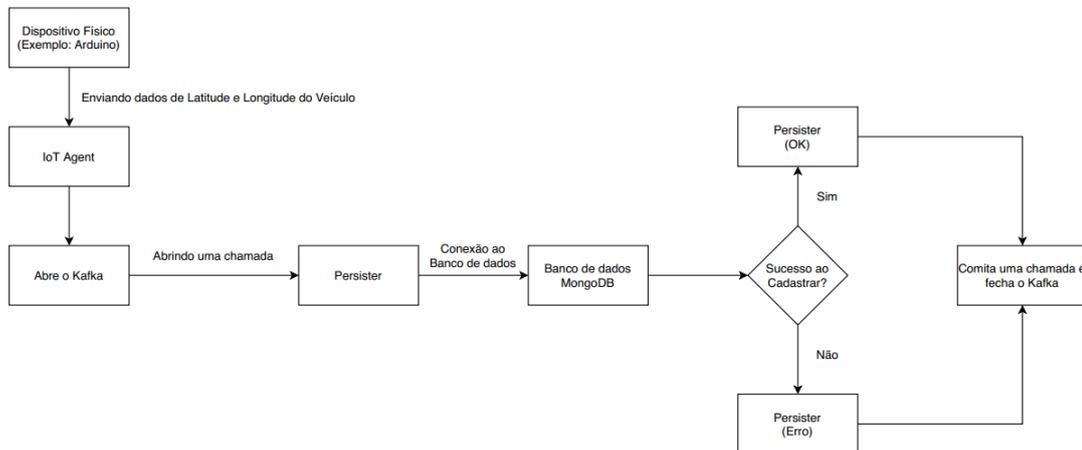
⁴ LoRa é um protocolo de comunicação de baixo consumo de energia e longo alcance para dispositivos de radio-frequência. *American Tower Corporation* (ATC) é fabricante de dispositivos de rede neutras para infraestruturas compartilhadas de aplicações LoRa [Castro 2022]

⁵ O Sigfox é um protocolo *wireless* para aplicações IoT que opera com baixas potências e atua em bandas não licenciadas. A WDN é a empresa responsável pela rede. [Blog 2023]

⁶ *HyperText Transfer Protocol* (HTTP) é o protocolo cliente-servidor tradicionalmente usado para requisição de páginas na *web*, enquanto *JavaScript Object Notation* (JSON) é uma especificação que define uma estrutura de dados para troca de informações entre *hosts*, aplicações etc

Se o banco de dados cadastrar as informações com sucesso é retornado uma mensagem de OK para o *Persister* e o *Kafka* finaliza a comunicação. Por fim, sempre que um novo dado for enviado por um dispositivo, todo o processo é realizado novamente [DOCS 2020].

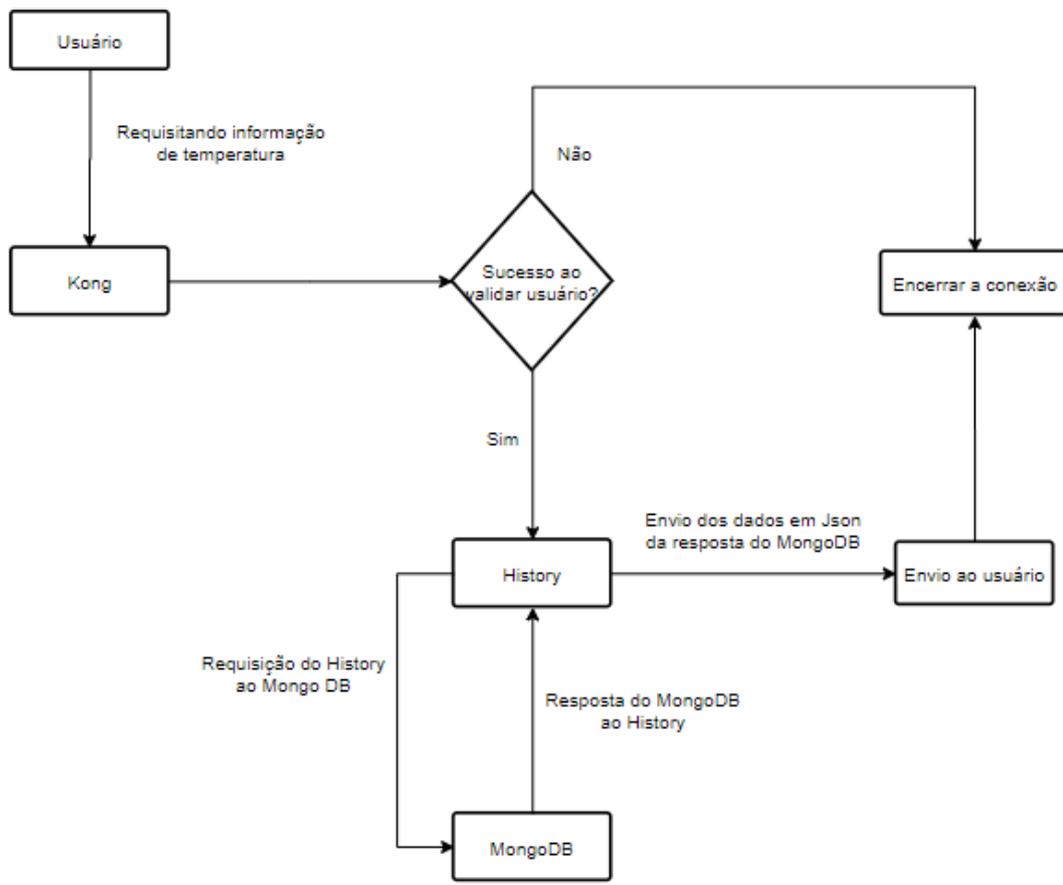
Figura 9 – Arquitetura do *Persister*.



Fonte: Elaborado pelo autor.

Sobre o *History*, este é o componente que recupera os dados salvos no MongoDB para serem usados em outras aplicações ou por usuários. Os dados são recuperados e enviados em formato JSON para o requerente.

A Figura 10 detalha a arquitetura no qual um usuário envia o pedido de um dado, no caso um dado de temperatura, esse pedido vai para a API Kong que verifica se o usuário é válido e autoriza o pedido enviando para o *History*. Em seguida, o *History* envia para o MongoDB uma requisição para recuperar os dados que o usuário está solicitando, assim, o banco de dados manda para o *History* os documentos pedidos e o *History* processa esse documento recebido transformando-o no formato JSON para posterior envio de volta para o usuário [DOCS 2020].

Figura 10 – Arquitetura do *History*.

Fonte: Elaborado pelo autor.

2.4.2.4 Kong

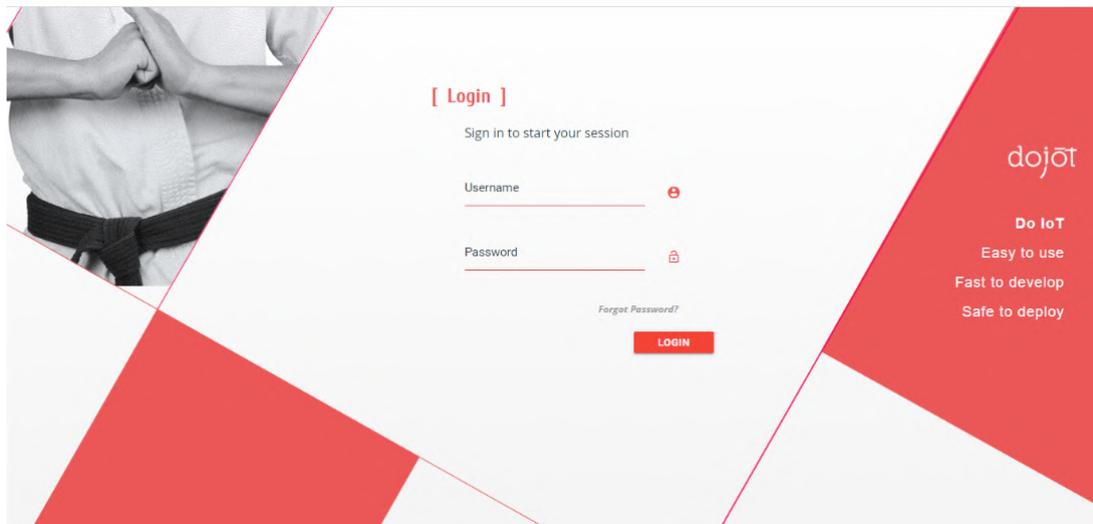
Kong é uma API utilizada na Dojot como uma fronteira entre aplicações externas e as aplicações e funções plataforma, assim tendo apenas um ponto de acesso. Além disso, o Kong facilita as regras de chamadas da API como de limitação de tráfego e de controle de acesso, também é responsável pelo gerenciamento, autenticação e autorização dos usuários na plataforma.

2.4.3 Ambiente Dojot

Neste tópico será destacado a parte visual da plataforma do Dojot com as funcionalidades que foram usadas neste trabalho.

2.4.3.1 Login

A primeira tela da Plataforma Dojot é a tela de *login*, exibida na Figura 11. Na tela de *login* é obrigatório ter uma conta criada para poder acessar as funções, neste caso, a plataforma possui uma conta criada automaticamente de administração que pode ser acessada com *login/senha admin*.

Figura 11 – Tela de *login* da plataforma Dojot.

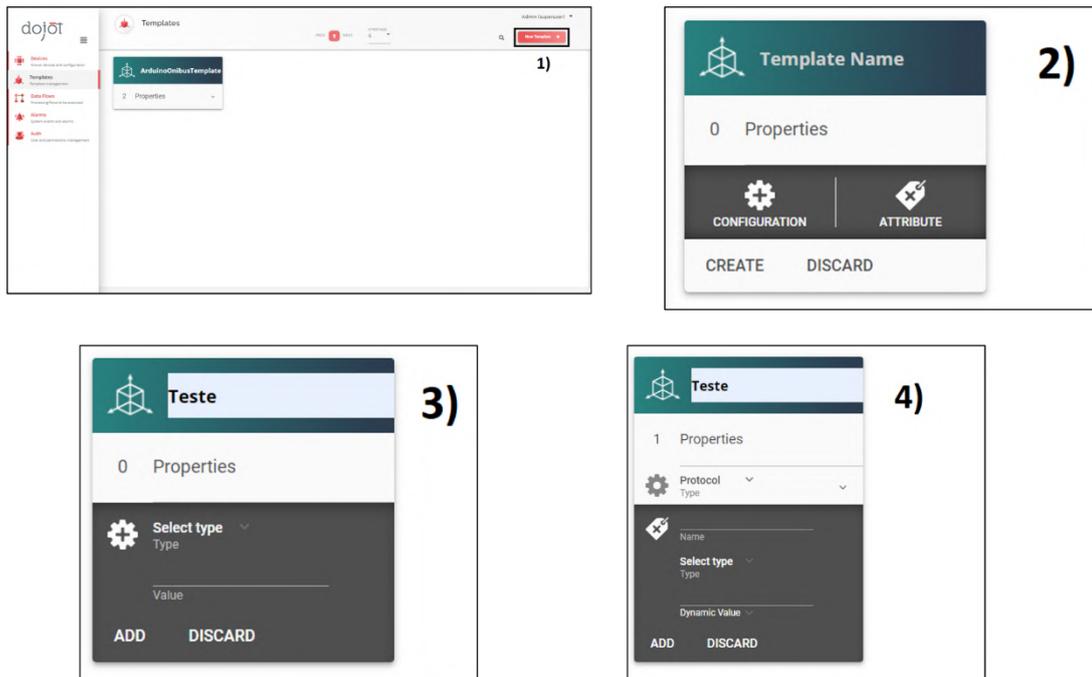
Fonte: Elaborado pelo autor.

Após o *login* realizado com sucesso, o usuário será redirecionado para a *dashboard* da plataforma sendo a primeira tela a de *Devices*, mas possuindo também outras funções de *Templates*, *Data Flows*, *Alarms* e *Auth*, todas dispostas do lado superior esquerdo.

2.4.3.2 *Templates*

Na seção de *templates* fica o gerenciamento de todos os *templates* da plataforma Dojot, e cada *template* é um modelo que padroniza os dados e atributos que serão recebidos na Dojot pelos dispositivos em campo. Esse modelo é usado como base para a parte de *Devices* da plataforma, fazendo com que dispositivos iguais dispostos no campo possam adotar o mesmo *template* base para o recebimento dos dados desses dispositivos, fazendo a conexão das informações que serão enviadas por eles para a plataforma Dojot.

Na Figura 12 é exemplificado a criação de um *template*, onde em 1) é mostrado os *templates* ativos com as suas propriedades; em 2) são mostrados as opções do *template* sendo elas *Template Name* para dar um nome de identificação, *Configuration* para as configurações básicas e *Attribute* para a definição dos atributos e os botões para criar e de descartar o novo *template*; em 3) na opção de *Configuration* é possível definir o seu tipo e valor como *protocol*, *topic*, *translator* e *device timeout*, por exemplo se selecionar *protocol*, o campo de valor terá a opção MQTT para selecionar; em 4) na opção de *Attribute* o nome e tipo de dado pode ser definido, podendo ser: *Geo*, *Float*, *Integer*, *String* ou *Boolean*; por fim, seu valor pode ser: *Dynamic Value*, *Static value* ou *Actuator*.

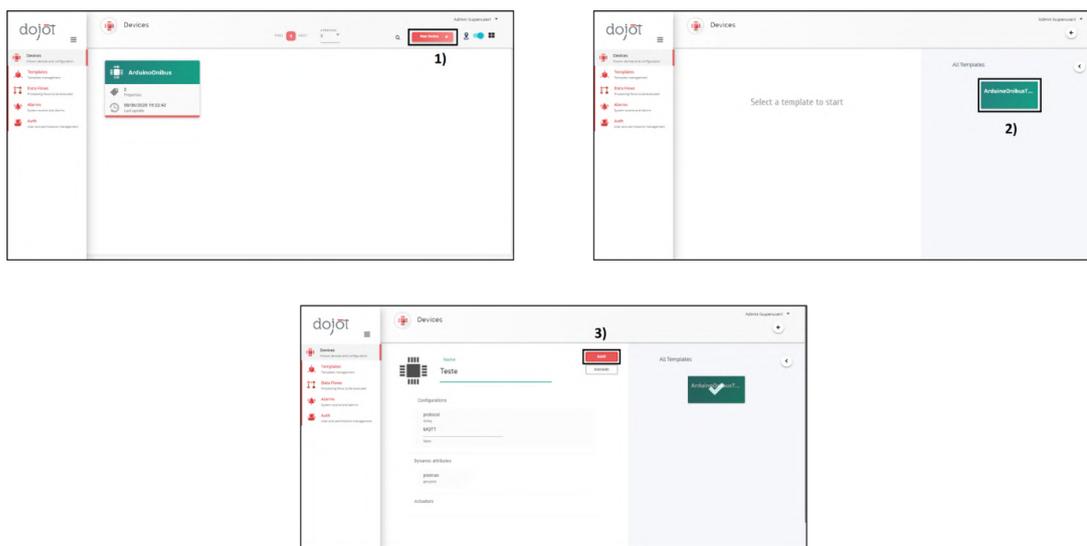
Figura 12 – Tela de criação de *Template*.

Fonte: Elaborado pelo autor.

2.4.3.3 *Devices*

Esta área é responsável pelo gerenciamento dos dispositivos físicos que estão em campo e que fazem parte do projeto, onde cada *Device* criado na plataforma Dojot corresponde a um único dispositivo físico disposto no campo, onde é atribuído a um *template* informando quais os dados que são enviados pelos dispositivos e cada *Device* possui um identificador dentro da plataforma para poder acessar essas informações. Para acessar esse identificador, basta clicar em um *device* já existente e procurar o campo *device id*.

Na Figura 13 é exemplificado a criação de um *device*, onde em 1) é mostrado os *devices* ativos e a opção para criar um novo *device*; em 2) é mostrado uma página para a escolha de um *template* e em 3) é onde é dado o nome deste dispositivo, suas configurações e valores são definidos pelo *template* escolhido, finalizando assim a sua criação.

Figura 13 – Tela de criação do Device.

Fonte: Elaborado pelo autor.

2.5 Banco de Dados Relacional

De acordo com [Heuser 2009](#), o banco de dados é um conjunto de integração e compartilhamento de dados que visa atender a uma comunidade de usuários, ou seja, podendo ser acessada por vários sistemas os quais fazem requisição de dados, evitando assim uma redundância de armazenamento de dados entre os sistemas.

O conceito de banco de dados relacional foi estabelecido na segunda metade do século 20 e com o avanço na área da computação, a partir da década de 1970 foi dado início ao surgimento do Sistema de Gerenciamento de Banco de Dados (SGBD), que por sua vez, são *softwares* onde são implementadas funções para a definição, alteração, recuperação e exclusão de dados presentes em determinado banco de dados. Um dos principais objetivos de um SGBD é facilitar a programação e criação de aplicações de diferentes tipos de banco de dados.

Existem diversos tipos de banco de dados, dentre eles destacamos o banco de dados relacional como o tipo de banco de dados que organiza e apresenta os dados em tabelas e a partir das relações entre as tabelas. Cada tabela é um conjunto não ordenado de linhas, também conhecido como tuplas ou registros.

Para a recuperação de uma informação em uma tabela é necessário realizar uma requisição ao banco de dados, e caso não seja explicitamente declarado qual o tipo de ordenação das tuplas, por exemplo, ordenando por idade do menor valor para o maior valor, o retorno das informações é feito de acordo com o Número de Identificação (ID) da chave primária da tabela consultada. Cada tupla da tabela possui uma série de campos denominados de atributos, sendo os tipos de atributos definidos durante o projeto de modelagem do banco de dados [[Heuser 2009](#)].

A primeira etapa para a elaboração de um banco de dados relacional refere-se à criação

de um modelo de representação da estrutura do banco de dados, sendo este responsável em descrever de maneira geral quais serão os tipos de informações que esse banco armazenará. O banco de dados relacional pode ser representado em diferentes formas, sendo elas a modelagem conceitual, modelagem lógica e modelagem física.

O modelo conceitual desenvolve uma proposta de solução de forma independente às restrições presentes no SGBD a ser adotado. Para isso, este modelo se baseia na notação do diagrama de classes da Linguagem de Modelagem Unificada (UML, do inglês *Unified Modelling Language*) [Pressman 2011].

O modelo conceitual mais conhecido é o modelo de Entidade-Relacionamento (ER), que foi criado por Peter Chen em 1976. Este modelo possui uma representação gráfica denominada de DER, que é composta por entidade, relacionamento e cardinalidade. A entidade é o conceito base do modelo ER, sendo a modelagem abstrata de objetos ou coisas da realidade que serão utilizadas no banco de dados. Para armazenar as informações, uma entidade, no DER, possui como representação gráfica um retângulo em que contém o nome da entidade [Heuser 2009], como ilustrado na Figura 14.

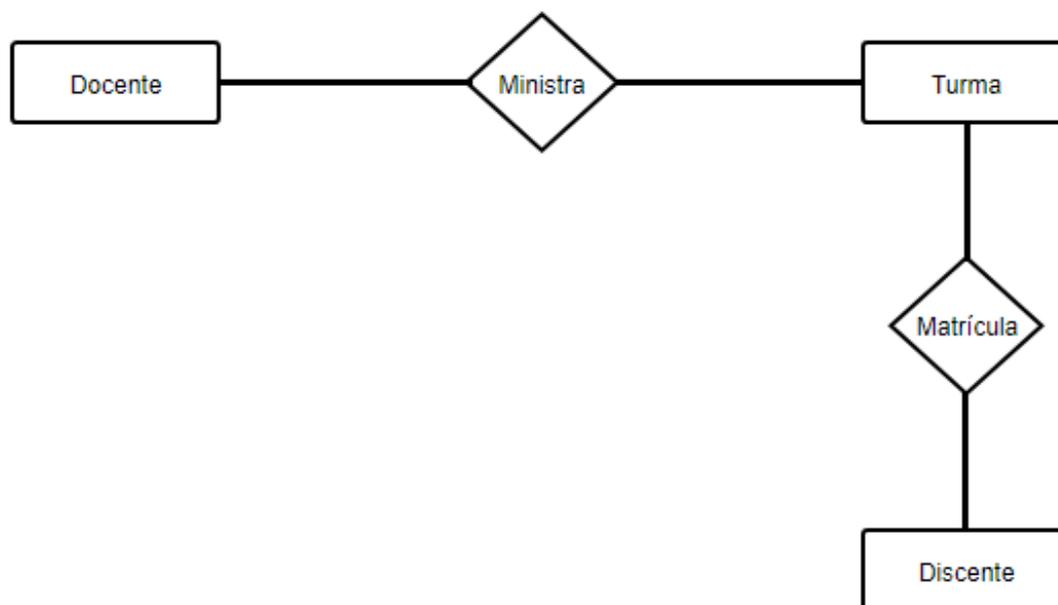
Figura 14 – Exemplos de Entidade no modelo DER.



Fonte: Elaborado pelo autor.

O relacionamento fornece detalhes sobre uma função, e sobre a ocorrência ou associação entre duas entidades. No DER, o relacionamento é representado por um losango que contém o nome do relacionamento ligando as entidades que fazem parte desse relacionamento através de linhas [Heuser 2009], como ilustrado na Figura 15.

Figura 15 – Exemplos de relacionamentos no modelo DER.



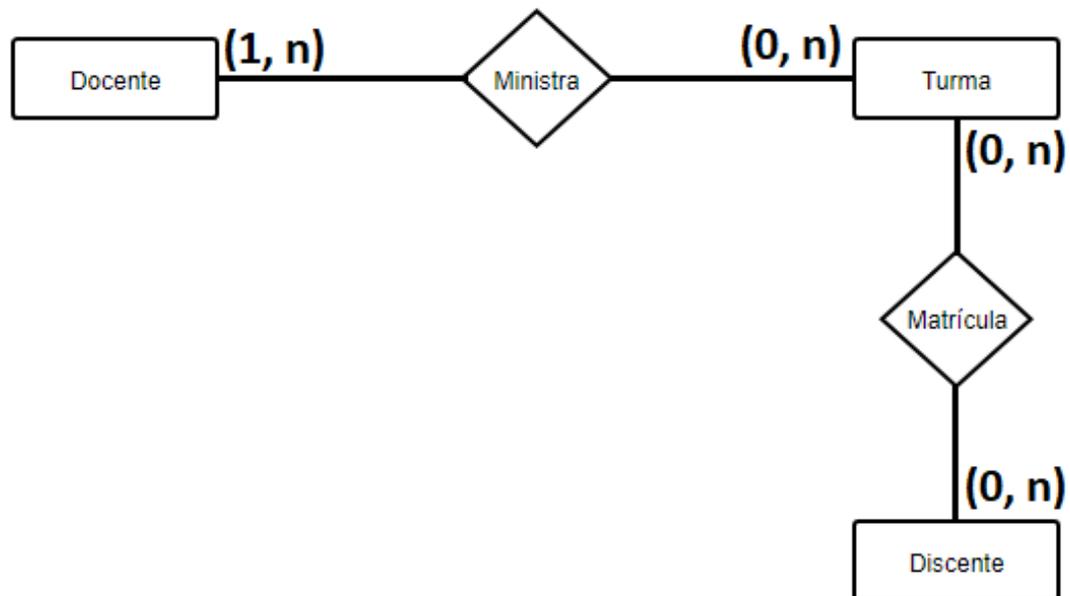
Fonte: Elaborado pelo autor.

Dentro da ligação de uma entidade e seu relacionamento é onde está estabelecida a cardinalidade, que é a quantidade de ocorrências de uma entidade associada a uma quantidade de ocorrências de um relacionamento, podendo ter cardinalidades máxima e mínima.

A cardinalidade máxima representa uma quantidade máxima de ocorrências entre as entidades com os relacionamentos, sendo representada com dois tipos: cardinalidade para 1 ou para muitos representado pela letra n [Heuser 2009]. Já a cardinalidade mínima é uma quantidade de ocorrências entre as entidades com os relacionamentos com quem há interação, que pode ter apenas os valores 0 ou 1.

Quando o valor é 1 indica uma "associação obrigatória", fazendo com que a associação entre uma entidade e o relacionamento seja feita de maneira obrigatória a cada ocorrência em questão. Já quando o valor é 0 indica uma "associação opcional", indicando que a relação entre uma entidade e o relacionamento não é feita de maneira obrigatória a cada ocorrência [Heuser 2009]. No DER, a cardinalidade é representada por números que indicam a quantidade de ocorrências, e que são posicionados ao lado das entidades e relacionamentos. Na Figura 16 é ilustrado um exemplo do uso das cardinalidades, a sua leitura é dada no sentido contrário à da entidade em questão, por exemplo: Um docente pode ministrar em nenhuma ou várias turmas e uma turma pode ser ministrada por 1 ou vários docentes.

Figura 16 – Exemplos de relacionamentos e cardinalidades no modelo DER.



Fonte: Elaborado pelo autor.

O segundo modelo é o modelo lógico, que por sua vez é dependente do tipo de SGBD, pois neles é atribuído o detalhamento de como serão as tabelas, seus atributos e relações entre as tabelas de acordo com a sintaxe do SGBD em questão. A partir das modelagens lógicas é possível definir o modelo físico do banco de dados, o qual possui as definições e sintaxe para a implementação física das tabelas, processos e funcionalidades [Heuser 2009].

2.6 Requisitos

A engenharia de requisitos compreende de um conjunto de atividades especializadas que através da interação entre os participantes do sistema, têm como objetivo identificar os requisitos a fim de alcançar uma compreensão e classificação ótimas de todas as necessidades do projeto. Assim, a engenharia de requisitos proporciona um processo de modelagem de *software* que contempla uma perspectiva do projeto. Isso viabiliza a análise prévia do trabalho que será executado, permitindo antecipar as necessidades inerentes ao projeto, bem como as funcionalidades que devem ser incorporadas para garantir o êxito do empreendimento [Pressman 2011].

Os requisitos de um sistema de *software* podem ser categorizados como funcionais e não funcionais. De maneira geral, os requisitos funcionais englobam as funcionalidades e os serviços que o sistema executará, ou seja, descrevem o comportamento do programa de acordo com as necessidades dos usuários. Para atender a essas operações, é fundamental levar em consideração vários fatores relativos ao sistema e ao seu desenvolvimento, incluindo o tipo de *software* e as especificações fornecidas pelos usuários [Sommerville 2011].

A fim de garantir a efetiva execução das funcionalidades estabelecidas, é crucial que os requisitos funcionais sejam definidos de maneira precisa, especialmente durante as fases iniciais ou avançadas do desenvolvimento do projeto. Isso tem por objetivo evitar custos extras e atrasos na entrega [Sommerville 2011]. A Tabela 1 ilustra três exemplos de requisitos funcionais pertencentes a um sistema de gerenciamento de estoque.

Tabela 1 – Exemplo de requisitos funcionais

| Requisitos Funcionais |
|----------------------------------------------------------------------------------------------|
| 1 – O sistema deve dedicar um espaço para a busca dos produtos cadastrados pelos clientes. |
| 2 – O sistema deve organizar os itens marcados para compra em um carrinho de compras |
| 3 – O sistema deve suportar a compra de produtos isolados ou a partir do carrinho de compras |

Fonte: Elaborado pelo autor.

Por outro lado, os requisitos não funcionais são caracterizados por não possuírem uma conexão direta com as funcionalidades do sistema, ou seja, não estão diretamente vinculados às operações específicas ou às suas características individuais. Em vez disso, estão associados a fatores decorrentes do sistema como um todo e alguns desses requisitos podem impor limitações ao desenvolvimento do sistema, podendo em alguns casos, torná-lo inviável ou prejudicar a implementação de outras funcionalidades no projeto. De modo geral, os requisitos não funcionais dizem respeito às condições que o sistema deve atender [Sommerville 2011]. A Tabela 2 ilustra três exemplos de requisitos não funcionais sobre um sistema de gerenciamento de estoque.

Tabela 2 – Exemplo de requisitos não funcionais

| Requisitos Não Funcionais |
|------------------------------------------------------------------------------------------------------------|
| 1 – O sistema deve proteger as senhas dos usuários através de criptografia |
| 2 – O sistema deve possuir alta disponibilidade para compras em regime de 24/7 |
| 3 – O sistema deve funcionar em navegadores <i>Mozilla Firefox</i> , <i>Opera</i> e <i>Google Chrome</i> . |

Fonte: Elaborado pelo autor.

Em última análise, os requisitos devem especificar de maneira clara o que deve ser desenvolvido, conferindo a eles um papel crucial no sucesso do *software*, já que possibilitam a estimativa de prazos de entrega, execução e testes, ao mesmo tempo em que viabilizam um maior controle durante a manutenção do *software* após a conclusão.

2.7 Média Móvel

A Média Móvel (do inglês *Moving Average*) é um método estatístico para suavizar flutuações de curto prazo nos dados, proporcionando uma visão mais clara das tendências de longo prazo. Para isso, a partir de uma série de dados em diferentes períodos de tempo, como por exemplo preços de ativos financeiros, o objetivo do método é calcular uma média geral em

cima das médias de valores anteriores em um intervalo contínuo, atualizando conforme novos dados se tornam disponíveis [Droke 2001].

Existem diferentes tipos de médias móveis, sendo as mais comuns a média móvel simples (SMA, do inglês *Simple Moving Average*) e a média móvel exponencial (EMA, do inglês *Exponential Moving Average*). A média móvel simples é calculada pela somatória dos valores dados em um determinado período pela divisão da quantidade de pontos de dados nesse período, dado pela equação abaixo. A Equação (1) apresenta o modelo:

$$SMA = \frac{\sum_{i=1}^n v_i}{n} \quad (2.1)$$

onde n é a quantidade do intervalo total e v_i é o valor dado dentro do período [Droke 2001].

Para exemplificar, temos um cenário com um intervalo de 10 minutos de coleta do nível de bateria de um determinado modal elétrico, tal cenário envolve somar os níveis de bateria capturados pelo sensor nos últimos 10 minutos e dividir pelo número de amostras n obtidas durante o intervalo de tempo estabelecido.

Já a média móvel exponencial atribui um peso maior aos dados mais recentes, tornando-se mais sensível às mudanças. Nesse caso, é calculada usando uma fórmula que leva em consideração os níveis de bateria anteriores e a taxa de suavização escolhida. A fórmula é iterativa, atribuindo mais peso aos níveis de bateria mais recentes e ajustando-se gradualmente à medida que novos dados são adicionados [Droke 2001]. A Equação (2) apresenta o modelo:

$$EMA = (V_{atual} - EMA_{anterior}) + \frac{2}{n+1} + EMA_{anterior} \quad (2.2)$$

Onde v é o valor do período atual, n é a quantidade de intervalo total e $EMA_{anterior}$ sendo o valor anterior da média móvel exponencial [Droke 2001].

A escolha do período de tempo para o cálculo da média móvel depende do horizonte de tempo que se deseja analisar. Os períodos mais comuns incluem 50, 100 e 200 dias para médias móveis de longo prazo, enquanto períodos menores, como 10 ou 20 dias, são usados para médias móveis de curto prazo. É importante notar que não existe um período "correto" ou "melhor" para todos os casos, e a escolha dependerá do contexto e da estratégia do analista, com isso um sistema que implementa tal método deve permitir ao usuário variar os parâmetros de tempo que servem de entrada para os modelos apresentados nas Equação (1) e (2) [Droke 2001].

Em termos de aplicação prática para um sistema de gestão multimodal, as médias móveis permitem a aplicação de análise de dados capazes de suavizar variações de comportamento em curto prazo, tornando mais fácil identificar as tendências subjacentes e com isso prever, por exemplo, o momento ideal para recarga de um modal ou a falha de determinado sensor instalado para monitorar o ambiente.

É importante ressaltar que a média móvel é um indicador retrospectivo, ou seja, é baseada em dados históricos. Ela pode ajudar a identificar tendências passadas e atuais, mas não pode prever o futuro com certeza [Droke 2001]. Portanto, é recomendado utilizar a média móvel em conjunto com outras ferramentas de análise técnica e considerar outros fatores, como indicadores econômicos, notícias e eventos do mercado, antes de tomar decisões de gestão da eficiência energética.

2.8 Regressão Linear

A Regressão Linear (do inglês *Linear Regression*) é uma técnica estatística usada para modelar e analisar a relação entre uma variável dependente e uma ou mais variáveis independentes. É uma das técnicas mais básicas e amplamente utilizadas na análise de dados e na previsão de resultados [Seber e Lee 2003, Su, Yan e Tsai 2012].

O objetivo da regressão linear é encontrar a melhor linha reta que descreve a relação entre as variáveis. Essa linha é chamada de "linha de regressão" e é determinada de acordo com o princípio dos mínimos quadrados, que busca minimizar a soma dos quadrados das diferenças entre os valores observados e os valores previstos pela linha [Seber e Lee 2003, Su, Yan e Tsai 2012].

A regressão linear simples é usada quando há apenas uma variável independente, enquanto a regressão linear múltipla é usada quando há duas ou mais variáveis independentes. Vamos nos concentrar na regressão linear simples para uma melhor compreensão inicial.

A equação da regressão linear simples pode ser expressa como:

$$Y = X\beta + \varepsilon \quad (2.3)$$

Onde Y é a variável dependente que estamos tentando prever, X é a variável independente que usamos para fazer a previsão, o valor ε é o coeficiente linear (intercepto) da linha de regressão, β é o coeficiente angular (inclinação) da linha de regressão [Seber e Lee 2003, Su, Yan e Tsai 2012].

O coeficiente linear ε representa o valor de Y quando X é igual a zero. Ele indica onde a linha de regressão intercepta o eixo vertical. O coeficiente angular β indica a inclinação da linha de regressão, ou seja, a taxa de mudança de Y em relação a X .

Para encontrar os valores dos coeficientes ε e β , é necessário realizar uma análise estatística utilizando os dados observados. A técnica mais comum é conhecida como "Método dos Mínimos Quadrados", que busca minimizar a soma dos quadrados das diferenças entre os valores observados e os valores previstos pela linha de regressão [Seber e Lee 2003, Su, Yan e Tsai 2012].

Em muitos casos, ε é um erro Gaussiano com média zero e variância que pode ser obtida da base de dados. Para obtenção do β , temos a Equação (4):

$$Y = X\beta \quad (2.4)$$

Onde se aplica a matriz transposta de X' em ambos os lados da igualdade, resultando na Equação (5):

$$X'Y = X'X\beta \quad (2.5)$$

Em seguida, se existir a identidade da matriz $(X'X)$ existe via a matriz inversa, logo temos a Equação (6):

$$(X'Y)^{-1}X'Y = (X'Y)^{-1}X'X\beta \quad (2.6)$$

Por fim, a partir da equação (6) é possível obter o valor de β isolado e representado na Equação (7):

$$\beta = (X'Y)^{-1}X'X \quad (2.7)$$

Com isso, é possível utilizar os atributos de entrada e classe de saída para a realização da predição. Uma vez que os coeficientes são estimados, podemos usar a equação da regressão linear para prever valores futuros de Y com base em valores conhecidos de X . Essas previsões podem ser úteis para entender a relação entre as variáveis e para fazer estimativas ou projeções sobre o comportamento futuro [Seber e Lee 2003, Su, Yan e Tsai 2012].

No entanto, é importante destacar que a regressão linear possui certas suposições subjacentes que devem ser atendidas para que os resultados sejam válidos. Essas suposições incluem a linearidade da relação entre as variáveis, a independência dos erros, a normalidade dos resíduos (diferença entre os valores observados e previstos) e a igualdade da variância dos resíduos ao longo da linha de regressão [Seber e Lee 2003, Su, Yan e Tsai 2012].

A regressão linear tem uma ampla gama de aplicações em diferentes campos, para esse trabalho sendo útil para prever: a variação de parâmetros ambientais, tais como, temperatura e umidade, assim como, para prever o número de usuários no transporte e o nível de bateria e tempo de carregamento da bateria a partir do seu consumo energético e velocidade de recarga, respectivamente.

2.9 Rede Neural de Regressão Geral

A Rede Neural de Regressão Geral (GRNN, do inglês *General Regression Neural Network*), é um tipo de rede neural utilizada para resolver problemas de regressão, ou seja, para prever valores numéricos contínuos. A GRNN é uma abordagem versátil que pode ser aplicada em várias áreas, como previsão de séries temporais, modelagem estatística e análise de dados [Specht et al. 1991].

A GRNN é caracterizada pela sua capacidade de aprendizado rápido e pela sua capacidade de lidar com problemas complexos de regressão. Ela é conhecida por sua simplicidade e eficiência [Specht et al. 1991], tornando-se uma escolha popular para muitos aplicativos.

Em relação a sua estrutura básica, a GRNN consiste em quatro camadas principais, sendo elas: Camada de entrada, camada de padrões, camada de soma e a camada de saída [Specht et al. 1991]. Os detalhes de cada camada serão destacados a seguir.

Camada de entrada: Responsável em recebe as variáveis independentes ou os atributos que serão usados como entrada para o modelo. Cada nó nessa camada está associado a um atributo específico [Specht et al. 1991].

Camada de padrões: Responsável em conter os nós correspondentes a cada padrão de treinamento, que são os exemplos utilizados para treinar o modelo. Cada nó nessa camada armazena um padrão específico e é conectado a todos os nós da camada de entrada [Specht et al. 1991].

Camada de soma: Responsável em computar a distância Euclidiana entre os valores de entrada e os padrões de treinamento. Essa distância mede a similaridade entre o padrão e os dados de entrada. Quanto mais próximo o valor de entrada estiver do padrão, maior será a similaridade [Specht et al. 1991].

Camada de saída: Responsável em computar os cálculos ponderados das saídas dos nós na camada de padrões. Os pesos são determinados pela similaridade calculada na camada de soma. A saída final é a combinação linear das saídas dos nós da camada de padrões, ponderada pela similaridade [Specht et al. 1991].

Durante o treinamento da GRNN, os padrões de treinamento são apresentados à rede, e os pesos são ajustados para minimizar o erro entre as saídas previstas e os valores reais correspondentes. O processo de minimização dos pesos usualmente utiliza os algoritmos *back-propagation* [Specht et al. 1991], Algoritmo híbrido com atualização dos pesos a partir de algoritmo genético e enxame de partículas [Agarkar, Hajare e Bawane 2016], entre outros.

Uma das principais características da GRNN é a capacidade de generalização. Uma vez treinada, a GRNN pode fazer previsões para novos dados, sem a necessidade de um novo treinamento completo. Ela é capaz de generalizar a partir dos padrões de treinamento e aplicar esse conhecimento para estimar os valores correspondentes para dados não vistos anteriormente.

A GRNN também apresenta outras vantagens, como a rapidez na fase de treinamento, a capacidade de lidar com dados ruidosos e a resistência a *overfitting*, ou seja, ajuste excessivo aos dados de treinamento.

Além disso, a GRNN não requer o conhecimento prévio sobre a estrutura subjacente dos dados. Ela é capaz de aprender automaticamente as relações entre as variáveis de entrada e saída, tornando-a uma ferramenta flexível e poderosa para lidar com problemas complexos de regressão.

3 Metodologia e Ferramentas Utilizadas

3.1 Introdução

Este capítulo apresenta os passos realizados para a compreensão do sistema e posterior implementação. É válido reforçar que este *software* visa atender um cenário de *Smart Campus* localizado na UFPA campus Belém.

3.2 Metodologia

Esse capítulo apresenta a metodologia adotada para a construção do *software* de gestão da eficiência energética. Após a conclusão do projeto de gestão multimodal [SILVA 2021], iniciaram-se conversas definir os próximos passos de trabalho. Com o tempo, a ideia de desenvolver um *software* que unisse o projeto de gestão multimodal com a adição de utilizar os dados enviados pelos modais para a plataforma Dojot e extrair informações dessa base de dados ganhou força e tornou-se o objetivo principal. Ao longo do processo de desenvolvimento, foi determinada a execução de quatro etapas principais, sendo elas: dispositivos Dojot e sensores, definição dos requisitos, programação, validação.

Na primeira etapa, foram realizadas reuniões com a equipe de engenharia do projeto com objetivo de compreender os elementos/sensores instalados nos modais. Para isso, foram realizadas reuniões quinzenais com a equipe do CEAMAZON e CPQD. Durante a reunião, foram discutidos como os dados coletados pelos sensores deveriam ser armazenados na Dojot para posterior coleta e utilização por uma ferramenta de gestão da eficiência energética. Para a instalação da plataforma Dojot foi utilizado um servidor Linux localizado nas dependências do CEAMAZON. O servidor conta com a configuração de processador Intel Xeon 4210, memória RAM de 64GB e disco rígido de 5TB. Para persistência dos dados no banco, foi configurada o armazenamento dos dados coletados por um período de 100 anos [LOBATO 2022].

Com a plataforma selecionada e configurada, foi planejado e implementado na plataforma Dojot os dispositivos virtuais, os quais representam os dispositivos físicos e seus respectivos sensores. Para este trabalho, foram implementados 1 dispositivo físico, possuindo um dispositivo Arduino, três módulos, um sensor. Para exemplificar, a Tabela 3 apresenta os dispositivos implementados para o modal 01, que representa um ônibus elétrico.

Tabela 3 – Dispositivos presentes no Modal 01

| ID | Dispositivos | Referência |
|----|----------------------------------------------|------------------|
| 1 | Dispositivo Arduino Mega | (CARVALHO, 2023) |
| 2 | Módulo de comunicação LoRa integrado com GPS | (CARVALHO, 2023) |
| 3 | Sensor de temperatura DHT-11 | (CARVALHO, 2023) |
| 4 | Dispositivo de contagem de pessoas CFP100 | (CARVALHO, 2023) |

Fonte: Elaborado pelo autor.

O dispositivo do modal envia os dados para um dos três *gateways* RAK 7249, localizados nos prédios Mirante do Rio, Laboratório de Engenharia Elétrica e Computação e CEAMAZON. Com o recebimento, os *gateways* enviando os dados para serem persistidos na Dojot [CARVALHO 2023].

Após a etapa de entendimento da Dojot e dos sensores envolvidos, deu-se início ao levantamento dos requisitos funcionais e não funcionais. Esse levantamento é fundamental para o alinhamento da programação do *software* e a modelagem do banco de dados. Na Tabela 4 será destacado os requisitos funcionais do *software*.

Tabela 4 – Requisitos Funcionais

| RF# | Descrição |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RF1 | O <i>software</i> deve permitir que o usuário se cadastre no sistema através do email e senha |
| RF2 | O <i>software</i> deve permitir que o usuário possa recuperar a sua senha em caso de perda |
| RF3 | O <i>software</i> deve permitir que o usuário possa fazer <i>login</i> e <i>logout</i> do sistema |
| RF4 | O <i>software</i> deve permitir que o usuário possa mudar suas informações de cadastro |
| RF5 | O <i>software</i> deve permitir que um usuário que tenha realizado <i>login</i> possa visualizar a posição do ônibus assim como buscar rotas entre dois pontos dentro do <i>Smart Campus</i> da UFPA |
| RF6 | O <i>software</i> deve permitir que apenas um usuário administrador possa acessar as estatísticas e gerenciamento de usuários. |
| RF7 | O <i>software</i> deve permitir que um usuário que não tenha realizado o <i>login</i> , possa visualizar a posição do ônibus. |
| RF8 | O <i>software</i> deve permitir que o retorno das informações de estatísticas possua respostas em forma de gráfico. |

Fonte: Elaborado pelo autor.

Com os requisitos funcionais definidos, o próximo passo foi a conclusão dos requisitos não-funcionais. Exibidos na Tabela 5.

Tabela 5 – Requisitos Não Funcionais

| RF# | Descrição |
|-----|----------------------------------------------------------------------------------------------------------------------------------|
| RF1 | O <i>software</i> deve ser compatível com os navegadores Opera, Mozilla Firefox, Google Chrome e Microsoft Edge. |
| RF2 | O <i>software</i> deve possuir mecanismos de segurança para a senha do usuário, controle de acesso e acessos externos indevidos. |
| RF3 | O <i>software</i> deve possuir interface responsiva para adaptar a qualquer tamanho de tela de visualização. |
| RF4 | O <i>software</i> deve possuir alta disponibilidade em regime 24/7. |
| RF5 | O <i>software</i> deve executar as tarefas desejadas de maneira rápida e estável. |

Fonte: Elaborado pelo autor.

Após a definição dos requisitos, deu-se início à programação do sistema *web*, começando pela montagem do banco de dados e do *backend*. O banco de dados passou por modificações até alcançar a versão atual, destacada na seção 3.4, enquanto o *backend* foi programado para comunicar-se com o banco de dados, o *frontend* e para o desenvolvimento dos algoritmos.

O *frontend* teve o desenvolvimento das cores, *templates* e telas do *software*. As ferramentas utilizadas para a escrita do *backend* e *frontend* serão descritas na seção 3.3.

O próximo passo foi a criação dos algoritmos estatísticos no *backend* do sistema *web*. Inicialmente, foi implementado a média, variância e desvio padrão, seguidos pela regressão linear e GRNN. Durante o desenvolvimento, constatou-se que o processamento desses algoritmos ocorria de maneira estável. Por fim, avançou-se para a parte de *frontend*, começando com uma interface mais simples para fins de validação e, ao longo do tempo, foram implementadas funcionalidades mais completas, como gráficos e animações até chegar no estado atual.

Para validar a implementação, foi desenvolvido um simulador em Python 3, utilizando as variáveis que serão destacadas na seção 4.7. A função deste simulador foi a criação e o envio para a Dojot de uma base de dados para validar a codificação dos algoritmos propostos. Observou-se que mesmo em cenários com grande quantidade de dados e processamento, não foram encontrados problemas de desempenho em relação aos códigos e ao computador de teste, um Acer modelo ES1-572 com Intel Core i3 de sexta geração e 4GB de memória RAM.

No último ciclo de implementação, integrou-se o sistema de gestão multimodal [SILVA 2021] e o sistema de *login* e gerenciamento de usuários, atendendo a todos os requisitos propostos e finalizando a atual fase de desenvolvimento.

3.3 Ferramentas Utilizadas

Para o desenvolvimento foi utilizado diversos meios como como linguagens de programação para a parte do servidor e do usuário, banco de dados relacional para armazenar as informações correspondentes do usuário, modais disponíveis, rotas mapeadas, rotas favoritas etc; banco de dados em tempo real Dojot no qual armazena as informações coletadas dos modais disponíveis no *Smart Campus*, podendo ser consultadas pelo sistema *web*. Para a simulação do comportamento dos modais, foi desenvolvido um simulador utilizando a linguagem Python 3.

3.3.1 Ferramentas *Backend*, Banco de Dados e *Frontend*

3.3.1.1 *Backend*

O *backend* é composto do processamento bruto do sistema, recebimento e envio das requisições do usuário, banco de dados e segurança. No qual um computador fica responsável por realizar essas tarefas, podendo funcionar de maneira local no computador do usuário ou de maneira remota com o computador podendo estar em qualquer lugar do planeta [Challapalli et al. 2021].

A programação do *backend* foi realizada utilizando a linguagem de programação PHP na versão 8, onde foi desenvolvido o recebimento e envio das requisições do usuário através do formato JSON, a comunicação com o banco de dados relacional e o banco de dados em tempo

real, a implementação dos algoritmos de Média Móvel, Regressão Linear e GRNN, a consulta e processamento dos dados coletados dos modais como a sua localização, velocidade média, distância do modais relacionados ao seus pontos de ônibus para os ônibus elétricos e ponto de cais para o barco elétrico. Para a implementação foi utilizado a plataforma de escrita de código VS Code.

3.3.2 Banco de Dados Relacional

O bando de dados utilizado foi o MariaBD, possui código aberto e é um projeto ramificado do banco de dados *MYSQL*, desenvolvido pelos mesmos criadores do *MYSQL*, possuindo alta compatibilidade entre os projetos [[MARIADB 2023](#)].

A utilização desse banco de dados foi relacionada a armazenamento do controle do usuário como *login* através de *email* e senha e seus níveis de acesso como: usuário comum e administrador; da geolocalização dos prédios existentes, rotas entre os prédios e entre os prédios e ponto de ônibus próximos; assim como os *ids* dos modais existentes para a consulta de suas informações no banco de dados em tempo real Dojot. A modelagem foi realizada no programa *Mysql Workbench* e o gerenciamento das tabelas e dados foi utilizado a aplicação *web* local *PhpMyAdmin*.

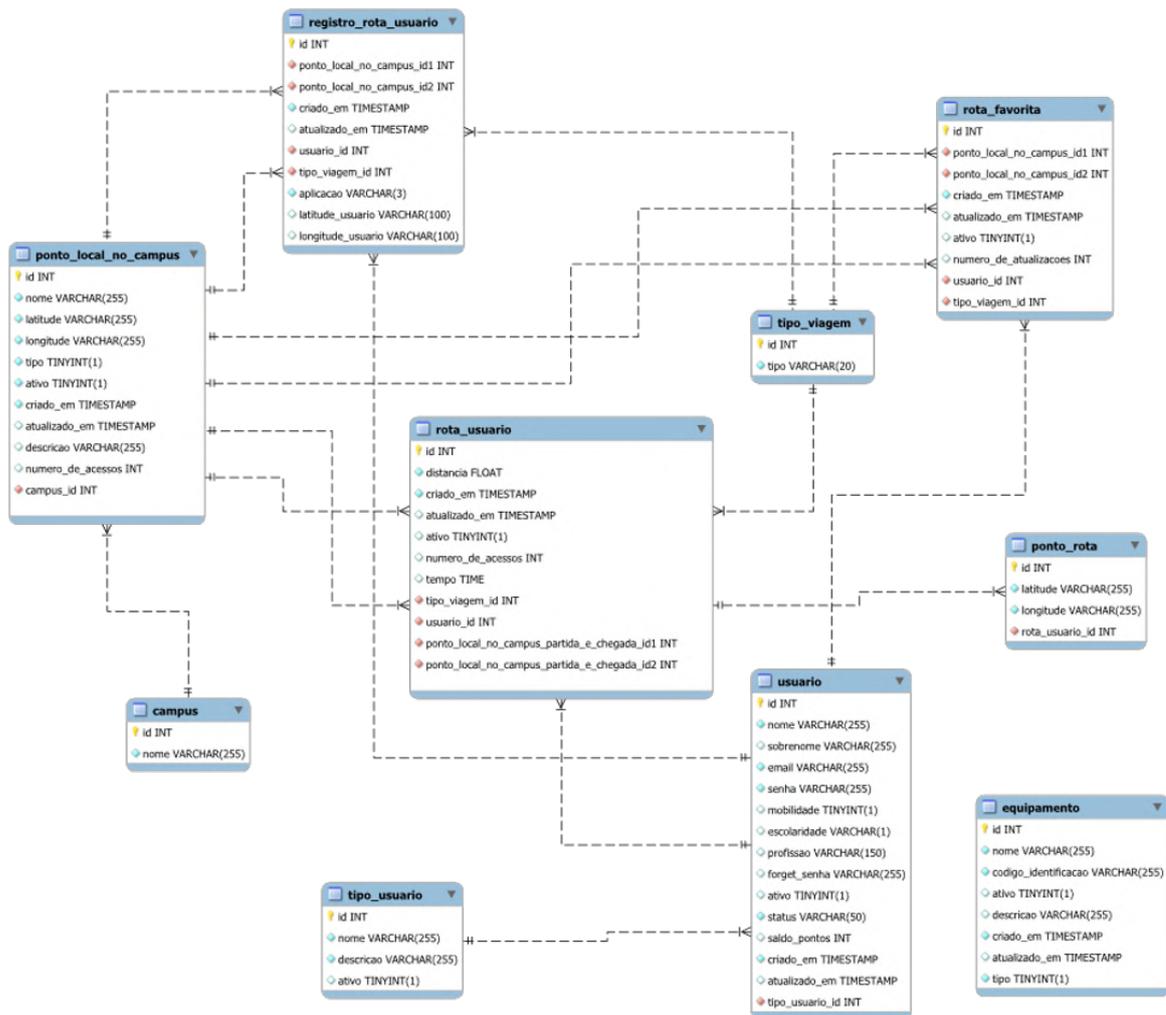
3.3.3 Visualização Para O Usuário

Com os dados sendo processados na parte do servidor, é necessário enviar a resposta de volta para os usuários e apresentar na tela de maneira legível. Para este fim, foi desenvolvido um *template* utilizando o *framework Bootstrap 4*, no qual são implementadas as linguagens de marcação HTML e CSS, produzindo *templates* responsivos a tela de visualização do usuário. Para implementar requisições ao *backend* e expandir as ações do usuário com o sistema *web*, foi utilizado a biblioteca *JQuery*, desenvolvida em *JavaScript*. A escrita do *frontend* também foi realizada na plataforma VS Code.

3.4 Modelagem Do Banco de Dados

Com base nas reuniões realizadas, foi desenvolvido a modelagem do banco de dados relacional para armazenamento dos dados coletados que será posteriormente usado pelo sistema *web* assim como já preparar o banco para novas aplicações. Na Figura 17 é apresentado uma visão geral do banco de dados em forma de entidade-relacionamento com suas tabelas, atributos, relacionamentos e cardinalidade

Figura 17 – Visão geral do modelo lógico do banco de dados



Fonte: Elaborado pelo autor.

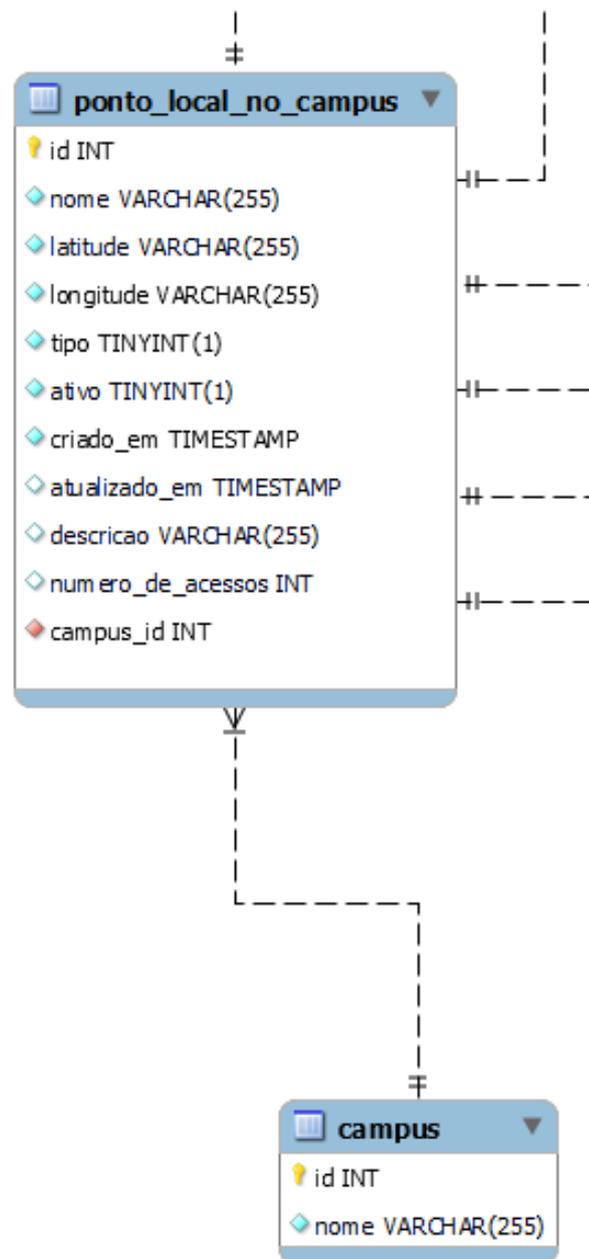
O banco de dados é composto por 10 tabelas, no qual são armazenadas como dados de geolocalização de prédios, pontos de ônibus, cais e rota, assim como informações de usuários que podem usar o sistema e seus níveis de acesso. Nos subtópicos seguintes, será discutido as tabelas utilizadas por este trabalho do banco de dados proposto acima.

3.4.1 Campus

Na Figura 18 são apresentadas as tabelas *campus*, *ponto_local_no_campus*. Na tabela *campus* ficam salvos os *campi* da UFPA que são: Campus Básico, Campus Profissional e Campus Betina, possuindo cada campus um identificador único.

Na tabela *ponto_local_no_campus* são armazenados os pontos de geolocalização de cada campus assim como uma identificação de referência à qual campus pertence.

Figura 18 – Tabelas que são armazenadas informações dos campi da UFPA



Fonte: Elaborado pelo autor.

3.4.2 Criação De Pontos De Interesse No Campus e Rotas

Na Figura 19 são apresentadas as tabelas **ponto_local_no_campus**, **registro_rota_usuario**, **rota_usuario**, **tipo_viagem** e **ponto_rota**. A tabela **ponto_local_no_campus** é responsável por armazenar os pontos de parada de ônibus e prédios. Nesta tabela são armazenadas a geolocalização de cada ponto, o nome para identificação de cada ponto, o tipo de ponto, sendo 0 (zero) para ponto de ônibus e 1 (um) para prédio, a quantidade de vezes que o ponto foi acessado e a identificação de referência a qual campus esse ponto pertence.

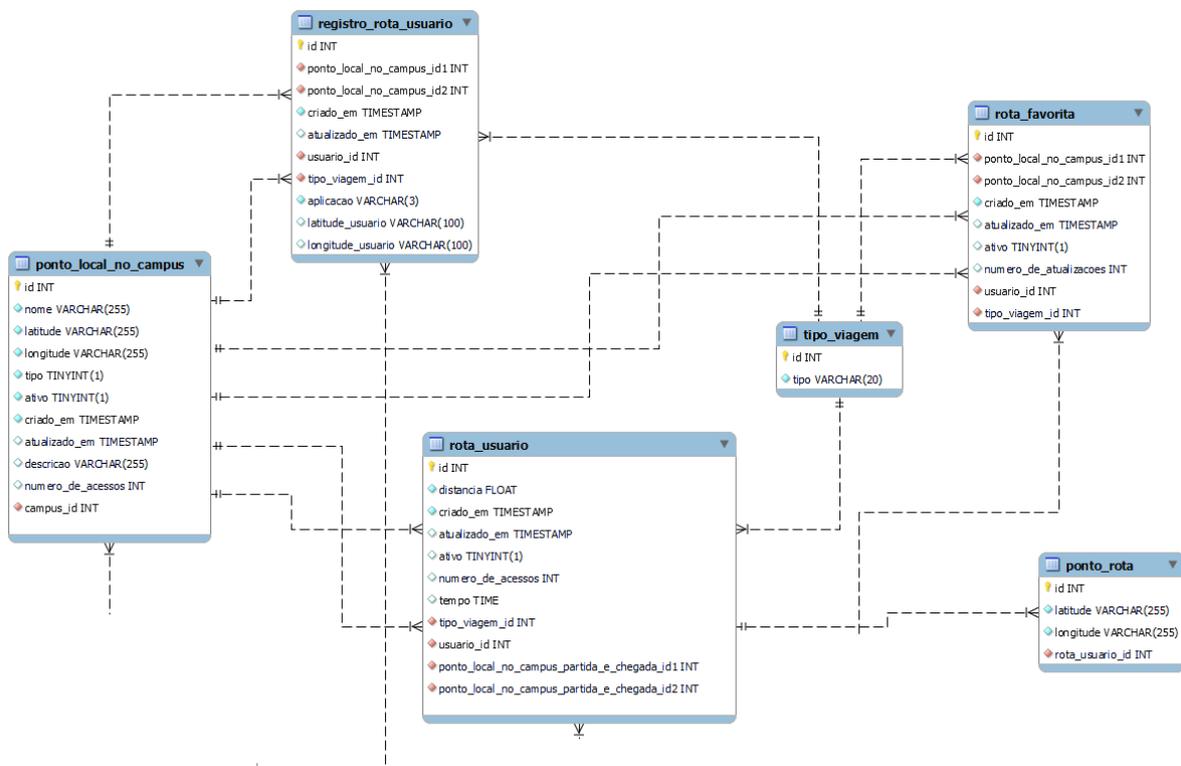
A tabela **rota_usuario** é responsável por armazenar as combinações possíveis de rotas

que devem estar disponíveis para o usuário final, cujo os pontos inicial e final são identificadores advindos da tabela ponto_local_no_campus, as combinações de rotas podem ser dos tipos: prédio → ponto de ônibus, prédio → prédio e prédio → cais, ou seja, armazenando as possíveis rotas que o usuário poderá percorrer entre 2 (dois) pontos mapeados, sendo armazenadas as informações sobre a distância total dessa rota, tempo médio de percurso (levando como base uma velocidade média de caminhada de 4 km/h), a quantidade de acessos que a rota teve, assim como o seu tipo de viagem e o usuário que a criou.

A tabela ponto_rota é responsável por armazenar todos os pontos de geolocalização de cada rota criada, ou seja, cada ponto de latitude e longitude da rota é um registro nessa tabela e, cada registro é atribuído ao identificador da rota criada na rota_usuario.

Na tabela tipo_viagem fica as informações sobre quais os tipos podem ser usados para percorrer um trajeto, sendo atualmente: ônibus, a pé e de barco, complementando a tabela rota_usuario e a rota_usuario_colaborada. Por fim na tabela registro_rota_usuario são armazenadas informações de qual aplicação o usuário utilizou para criar uma rota, como a aplicações web ou o aplicativo.

Figura 19 – Tabelas que são armazenadas os pontos de geolocalização de cada campus da UFPA e os seus mapeamentos

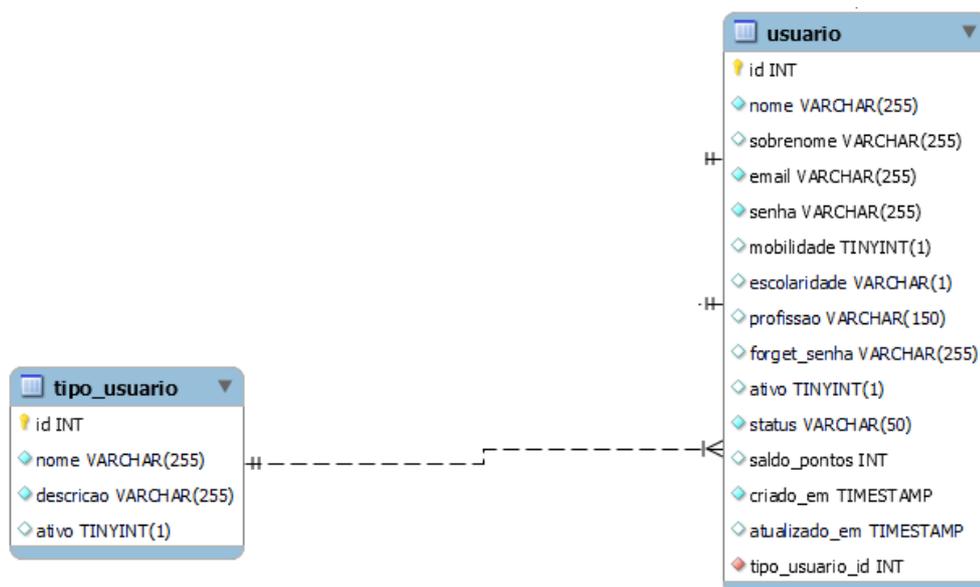


Fonte: Elaborado pelo autor.

3.4.3 Usuário

Na Figura 20 são apresentadas as tabelas de usuario e tipo_usuario. Na tabela usuario são armazenadas as informações de cadastro de todos os usuários que utilizam o sistema. Para diferenciação, os usuários são separados por níveis de acesso, por exemplo, um usuário comum tem apenas os seguintes acessos: consumo de dados já existentes no banco, gerência de seus favoritos e de seus *feedbacks*, enquanto que, um usuário administrador além de possuir as funções de um usuário comum, também tem acesso ao *software web* que é responsável pelo gerenciamento do sistema; esses tipos de usuários são definidos na tabela tipo_usuario.

Figura 20 – Tabelas de usuário e gamificação

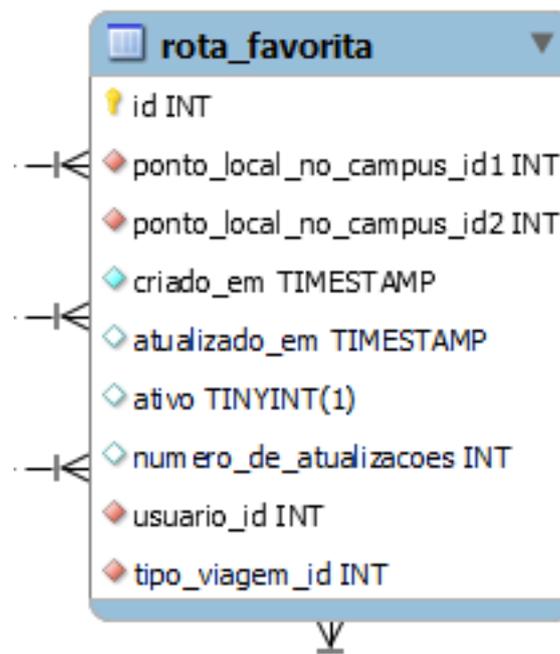


Fonte: Elaborado pelo autor.

3.4.4 Favoritos

A Figura 21 exibe a tabela rota_favorita, na qual ficam salvas as rotas que determinado usuário mais usa, podendo ser salvas automaticamente por meio de Sistemas Especialistas ou pelo usuário fazendo manualmente.

Figura 21 – Tabelas de favorito

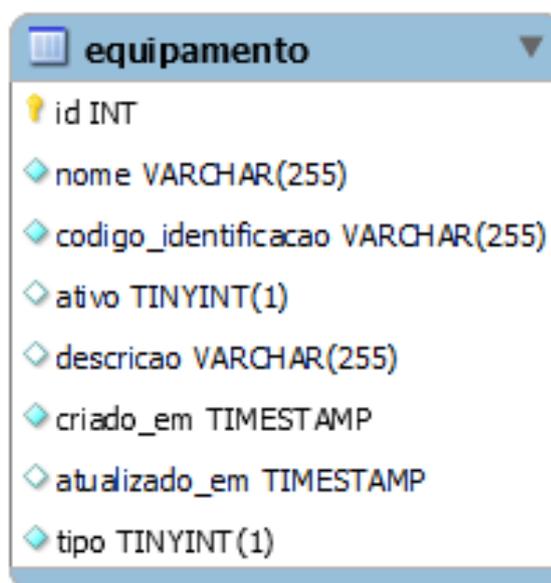


Fonte: Elaborado pelo autor.

3.4.5 Equipamento

Por fim, na Figura 22, é exibida a tabela equipamento, onde são gerenciadas as informações dos *devices* disponíveis na Dojot, permitindo um uso mais eficiente e controle da consulta de informações a partir das aplicações. Essa tabela foi criada pois a Dojot possui uma limitação de não retornar os *devices* e seus identificadores que estão disponíveis na plataforma, permitindo apenas o uso direto de suas as informações.

Figura 22 – Tabelas de equipamentos



The image shows a screenshot of a database table definition for a table named 'equipamento'. The table is displayed in a window with a blue header bar containing the table name and a dropdown arrow. Below the header, the table's columns and their data types are listed, each preceded by a diamond-shaped icon. The columns are: 'id' (INT, primary key), 'nome' (VARCHAR(255)), 'codigo_identificacao' (VARCHAR(255)), 'ativo' (TINYINT(1)), 'descricao' (VARCHAR(255)), 'criado_em' (TIMESTAMP), 'atualizado_em' (TIMESTAMP), and 'tipo' (TINYINT(1)).

| Column Name | Data Type |
|----------------------|--------------|
| id | INT |
| nome | VARCHAR(255) |
| codigo_identificacao | VARCHAR(255) |
| ativo | TINYINT(1) |
| descricao | VARCHAR(255) |
| criado_em | TIMESTAMP |
| atualizado_em | TIMESTAMP |
| tipo | TINYINT(1) |

Fonte: Elaborado pelo autor.

4 Sistema De Planejamento Da Gestão Da Eficiência Energética e Rotas (SISPGEER)

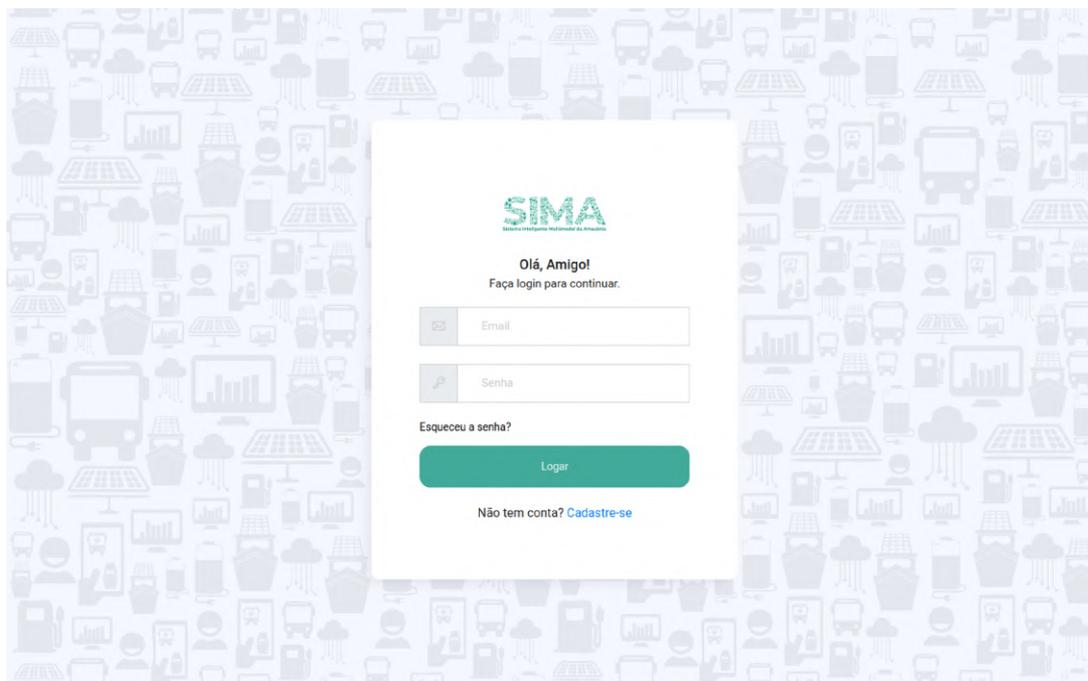
O Sistema de Planejamento da Gestão da Eficiência Energética e Rotas (SISPGEER) é um *software* que realiza consultas nos dados enviados pelos sensores para a nuvem. Para a realização da tarefa de gestão, o sistema permite aos seus usuários a utilização de diversas técnicas para tomada de decisão, sendo elas: Regressão Linear, GRNN, Média Móvel.

Tal sistema também permite aos usuários acompanhar em tempo real a geolocalização dos modais a partir de um painel *web* e permite a realização de consultas e planejamento de viagens entre pontos do *Smart Campus*.

Com o sistema proposto, o usuário pode gerar relatórios para verificação do desempenho de cada modal. Também, o sistema permite a verificação de qual o melhor período possível agendar o envio do veículo para o eletroposto afetando a menor quantidade de passageiros, qual a maior faixa de tempo de passageiros embarcados durante um dia, verificar se consumo de bateria está aumentando e predizer quando será o melhor período para a troca. Nas subseções seguintes, será detalhado as funções do sistema.

4.1 Login

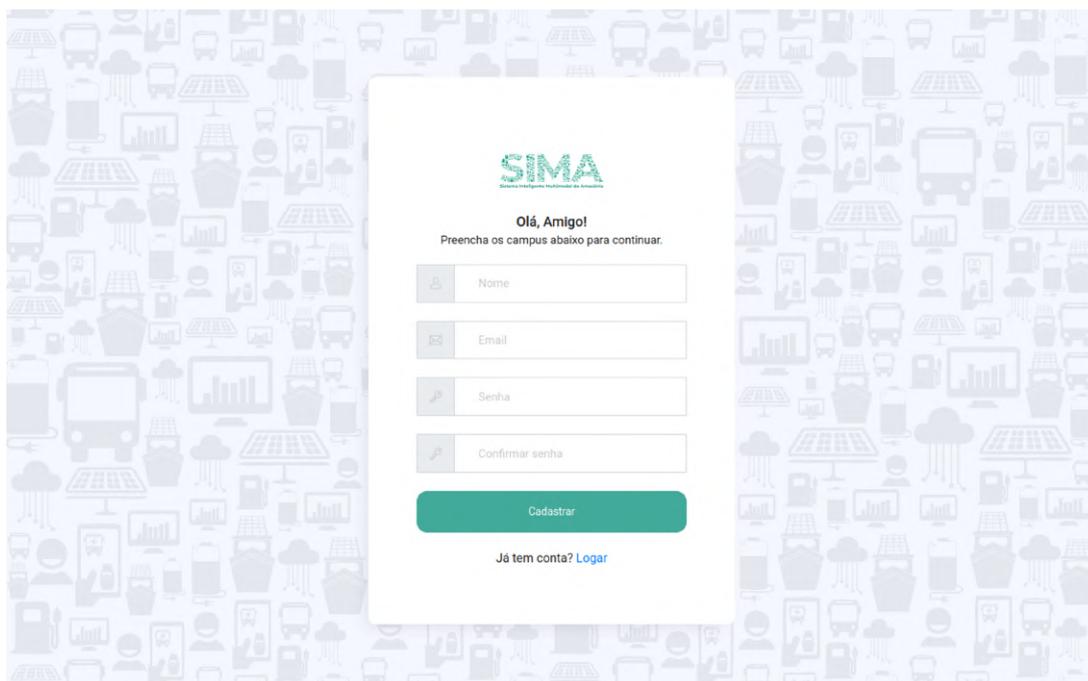
Para acessar o sistema, é preciso ter um *email* e senha válidos e realizar o *login* através da tela de *Login* no qual é a primeira a ser visualizada, caso o usuário esquecer a senha de uso, é possível recuperar no campo “Esqueceu a sua senha?” conforme a Figura 23.

Figura 23 – Tela de Login

A tela de login do sistema SIMA apresenta o logo da instituição no topo. Abaixo, há uma saudação amigável e uma instrução para fazer login. O formulário contém campos para 'Email' e 'Senha', um link para 'Esqueceu a senha?' e um botão verde para 'Logar'. Na base, há um link para 'Cadastre-se' para quem não possui uma conta.

Fonte: Elaborado pelo autor.

Caso não possua um cadastro, é possível criá-lo ao clicar no texto “cadastre-se” destacado na figura anterior, assim será exibido uma nova tela onde pedirá para inserir informações de nome de usuário, *email*, senha e para confirmar a senha digitada anteriormente conforme mostrado na Figura 24.

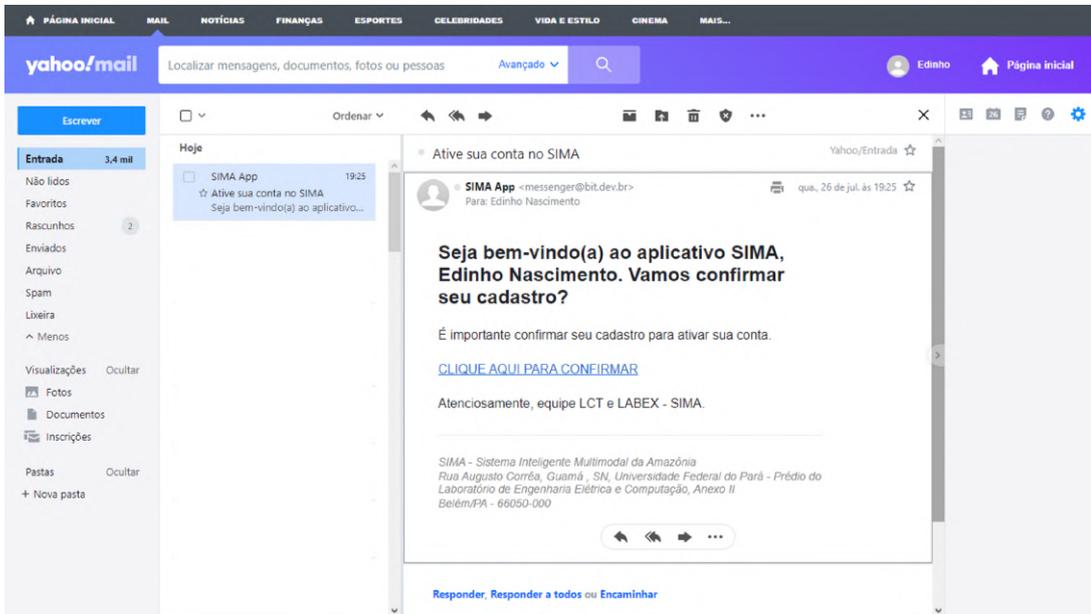
Figura 24 – Tela de Cadastro de novo usuário

A tela de cadastro do sistema SIMA solicita a criação de uma nova conta. O formulário inclui campos para 'Nome', 'Email', 'Senha' e 'Confirmar senha', além de um botão verde para 'Cadastrar'. Um link para 'Logar' é exibido na base para quem já possui uma conta.

Fonte: Elaborado pelo autor.

Ao clicar em “cadastrar”, será enviado um *email* de confirmação na caixa de entrada do que foi colocado no respectivo campo e assim que for confirmado, conforme a Figura 25, assim será possível fazer o *login* como destacado anteriormente na Figura 23.

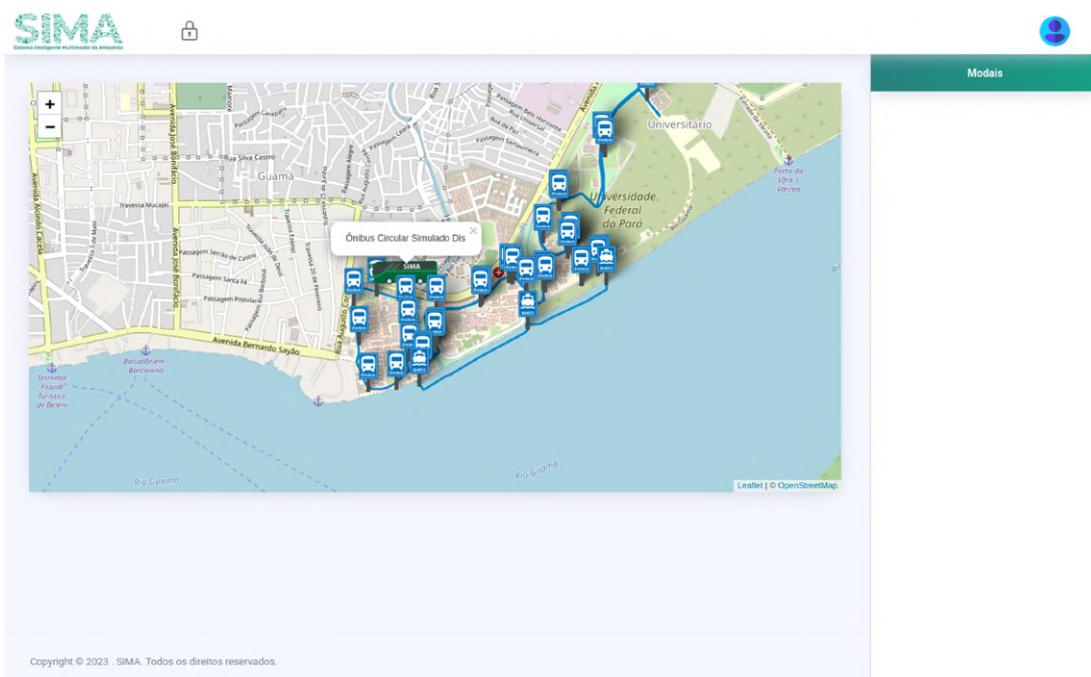
Figura 25 – Recebimento de *email* de confirmação



Fonte: Elaborado pelo autor.

Na Figura 26 é exibido uma tela de visualização dos modais caso o usuário não tenha realizado o *login*, porém o mesmo não poderá utilizar as outras funções do sistema *web*.

Figura 26 – Tela de modais para usuários que não realizaram *login*



Fonte: Elaborado pelo autor.

Conforme descrito na Figura 23, se o usuário esquecer a sua senha, é possível fazer o processo de recuperação de senha ao clicar em “Esqueceu a Senha?”, onde será mostrada uma tela onde o usuário colocará o seu *email* cadastrado no sistema, onde na caixa de entrada será enviado um *email* de recuperação de senha com um *link* para colocar uma nova senha. Mostrado Na Figura 27 (a) e Figura 27 (b).

Figura 27 – Tela de recuperação de senha (a) e recebimento do *email* (b)

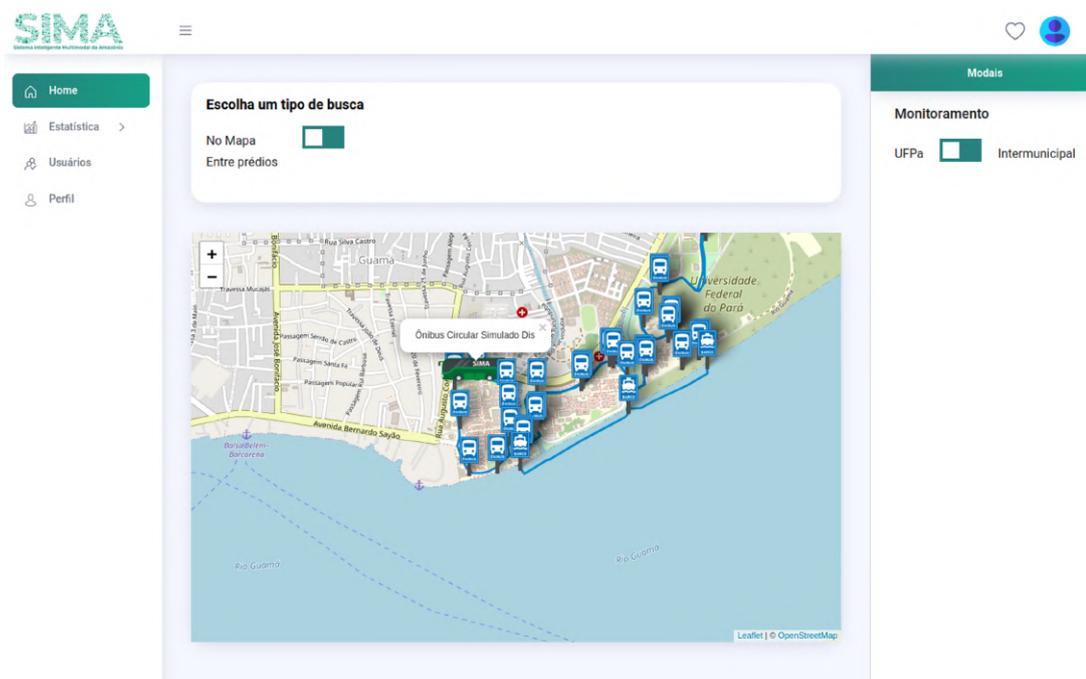


Fonte: Elaborado pelo autor.

4.2 Tela Principal

Ao realizar com sucesso o *login*, irá ser exibida a tela principal do sistema ilustrada na Figura 28, onde no centro é exibido os pontos de ônibus existentes e um marcador representando a posição atualizada do ônibus e barco em tempo real durante os seus respectivos trajetos. Na parte esquerda ficam localizadas as páginas do sistema, onde estamos é a página *home*, possuindo também a página focada nas estatísticas, no gerenciamento de usuários e no perfil do usuário que realizou o *login*.

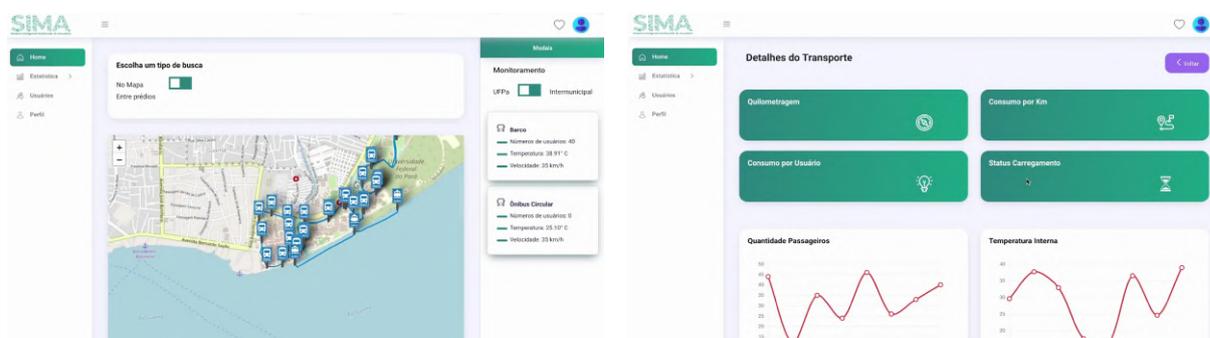
Figura 28 – Tela principal



Fonte: Elaborado pelo autor.

E na parte direita da Figura 29 (a) ficam alocadas as informações do modais em tempo real, podendo ser alterado qual o espaço a ser monitorado: os veículos presentes no interior da UFPA ou o Ônibus Intermunicipal. Ao clicar em um dos modais informados nesse campo, será mostrado uma nova tela onde dará mais informações sobre o estado do modal através de gráficos mostrando variações de um determinado atributo conforme o tempo e informações diretas como quilometragem, consumo e o status de carregamento da bateria, como na Figura 29 (b). Para o usuário comum, o acesso se restringirá a página *home* e suas funções, assim como a página de “perfil”. Já para o nível de administrador, além das funções de usuário comum, são liberadas as páginas de “estatísticas” e de “gerenciamento de usuários”.

Figura 29 – Alocação dos modais ativos (a) e Informações em tempo real (b)



(a) Exibição de modais ativos

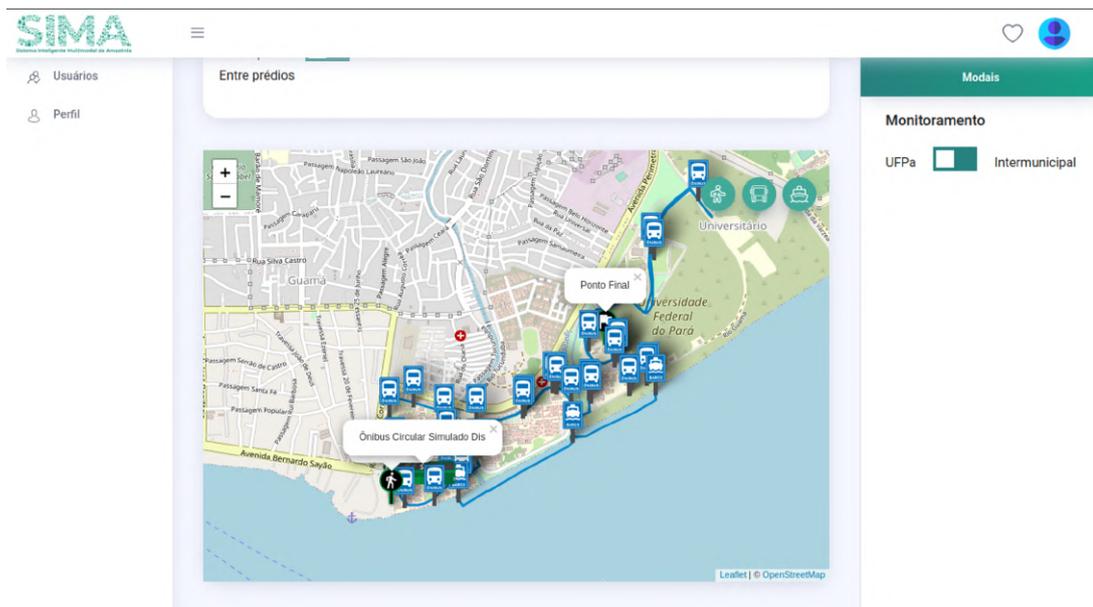
(b) Informações em tempo real

Fonte: Elaborado pelo autor.

Voltando a parte central da página *home*, no mapa são representadas duas funções: de

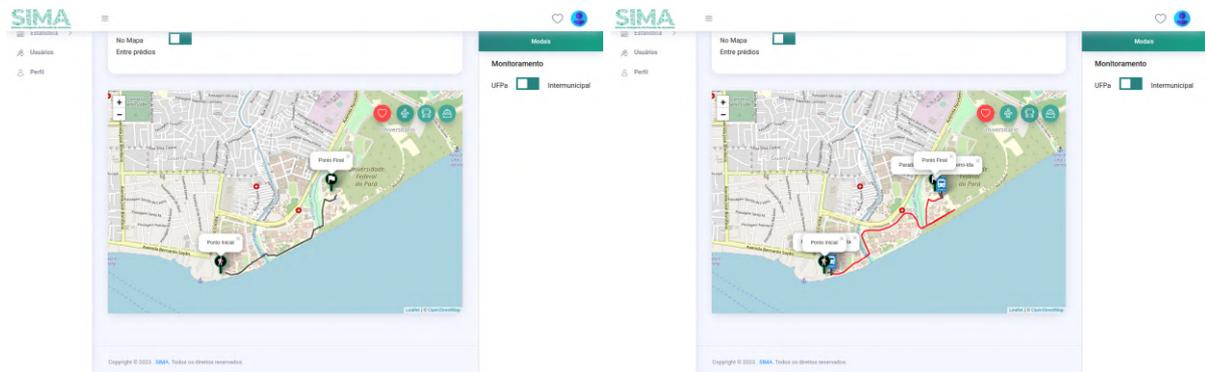
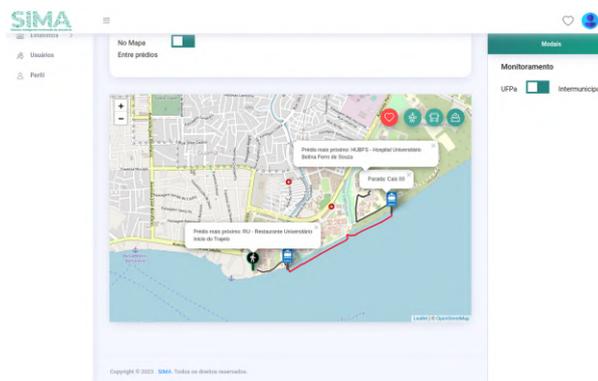
exibir a posição do ônibus e de recomendar um trajeto entre dois pontos. Para recomendar um trajeto, é preciso especificar na parte destacada qual o tipo de busca a ser utilizada, sendo uma busca de dois pontos arbitrários ao clicar em cima do mapa ou de selecionar dois prédios específicos a partir de uma lista criada com o mapeamento dos prédios existentes na UFPA [SILVA 2021]. Neste exemplo na Figura 30, foi utilizado a busca entre dois pontos no mapa.

Figura 30 – Definindo os pontos inicial e final para a busca de rota



Fonte: Elaborado pelo autor.

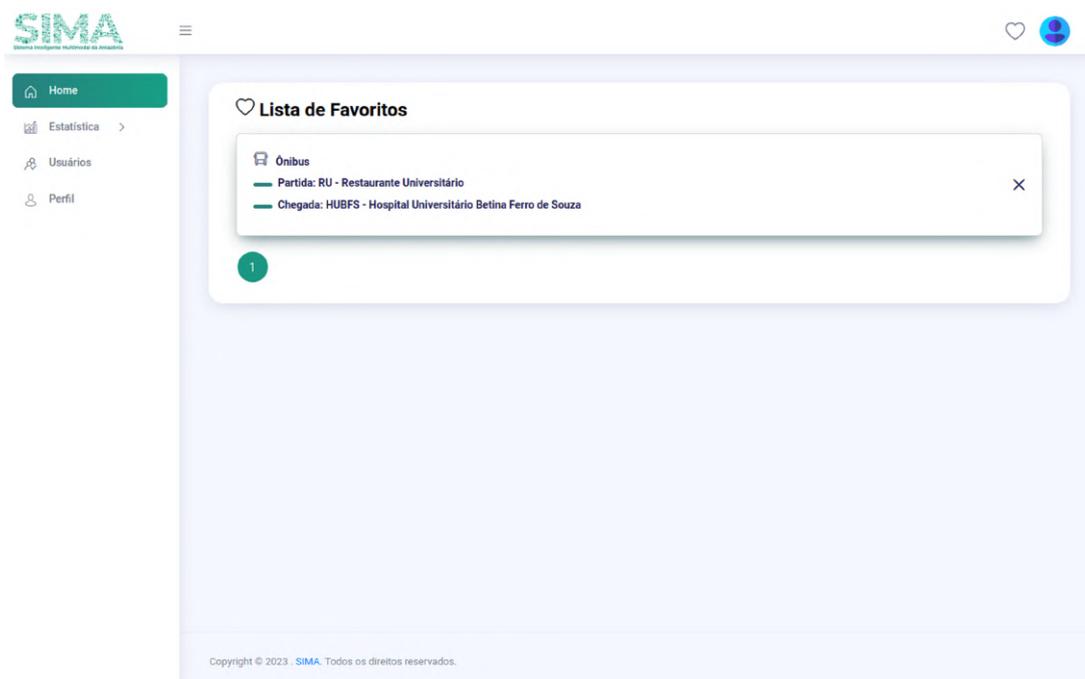
representam o tipo de percurso que está disponível sendo: trajeto a pé na Figura 31 (a), percurso de ônibus na Figura 31 (b) e de barco na Figura 31 (c).

Figura 31 – Tipos de percurso disponíveis sendo: ônibus (a), a pé (b) e barco (c)**(a) Percurso de ônibus****(b) Percurso a pé****(c) Percurso de barco**

Fonte: Elaborado pelo autor.

No lado esquerdo do ícone de trajetos a pé, mostrado na Figura 31, possui um novo ícone para adicionar o tipo de trajeto e os pontos selecionados a uma lista de favoritos do usuário, criando assim um acesso rápido do usuário para a rota preferida. E para a exibição das rotas favoritas é preciso clicar no ícone de coração no canto superior direito, ao lado da foto do usuário, onde será exibido em uma nova tela onde é mostrado o trajeto e o tipo da rota que foi salva, exibido na Figura 32.

Figura 32 – Tela de rotas favoritas



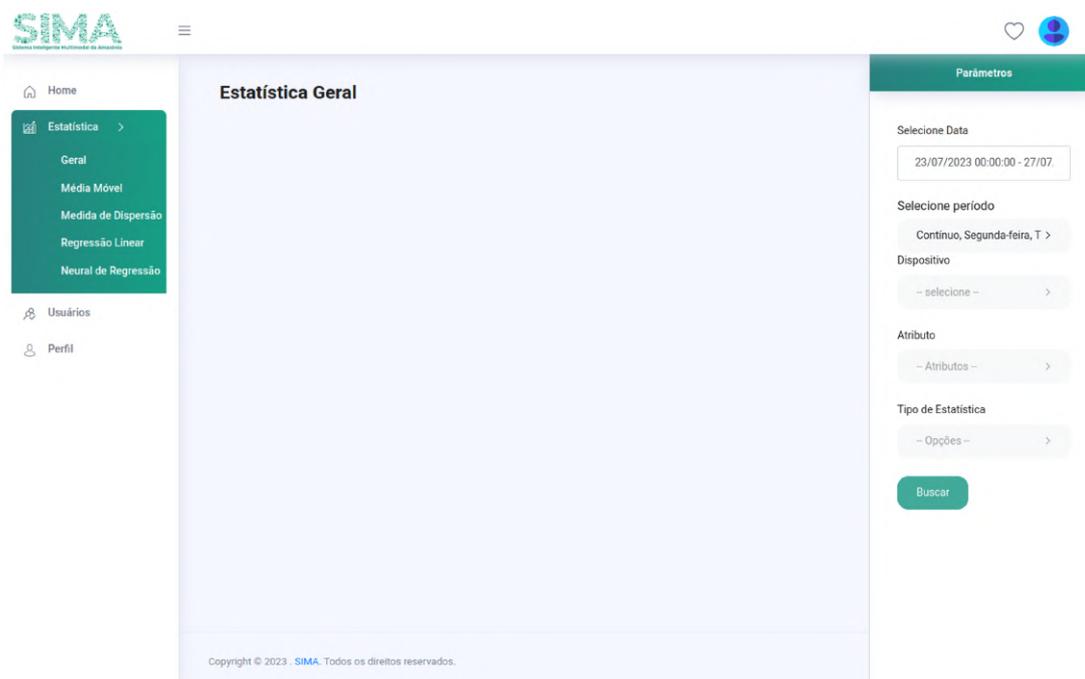
Fonte: Elaborado pelo autor.

4.3 Estatísticas

Na parte de Estatísticas é onde ficam localizadas as técnicas implementadas para se extrair informações através de todo o conjunto ou de uma parcela dos dados armazenados na Dojot. Essa parte se divide em Estatística geral, Média Móvel, Regressão Linear e GRNN.

A Estatística Geral é a primeira parte do bloco de Estatísticas e nela são implementadas as funções de média, variância e desvio padrão. Para o seu uso é preciso selecionar quais atributos que serão monitorados e qual o período de tempo sendo eles: período contínuo, semanal e mensal; assim como de quais dias da semana os dados serão exibidos. Conforme ilustrado na Figura 33.

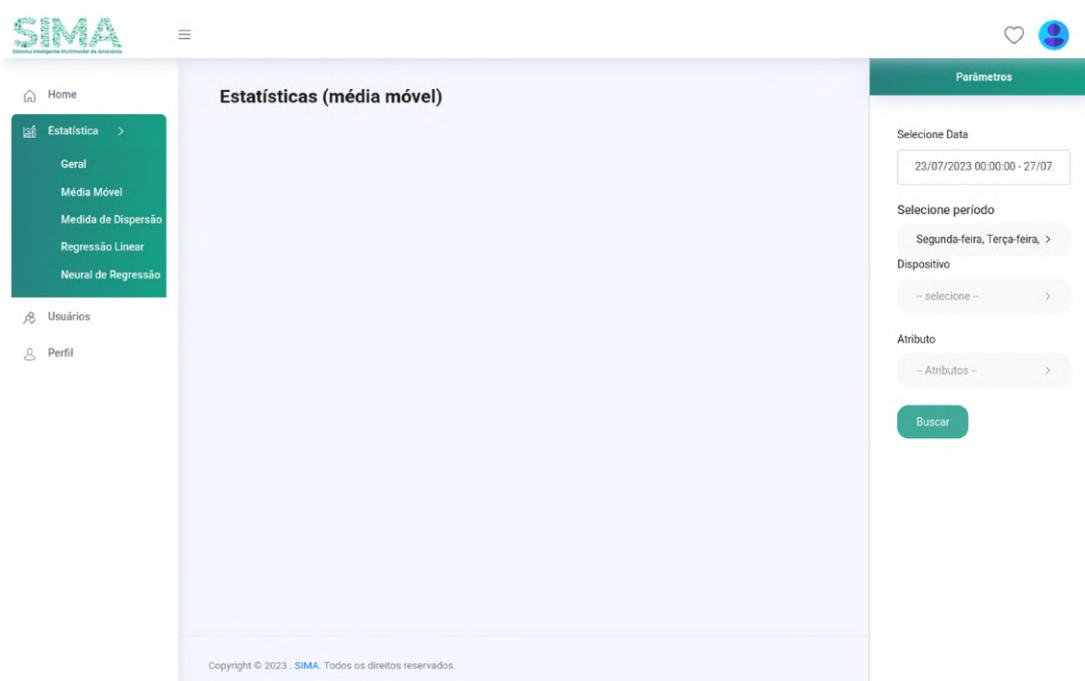
Figura 33 – Tela de estatística Geral



Fonte: Elaborado pelo autor.

O Método de Média Móvel (MM) segue os mesmos critérios para o seu uso no qual foi destacado na parte Estatística Geral, acrescentando a parte de como esses dados podem ser divididos: se for do dia corrente ou de um dia específico, poderão ser particionados em minutos ou por hora, caso for de um período maior que um dia a divisão será realizada por hora e se for maior que um período semanal os dados da MM serão por dia, mostrado na Figura 34.

Figura 34 – Tela de média móvel



Fonte: Elaborado pelo autor.

Para a Regressão Linear, além dos atributos para servirem de entrada, é preciso selecionar qual vai ser o atributo a ser comparado, podendo ser mais de um atributo para essa comparação. Além do gráfico dado ao término do processamento da Regressão, é possível prever o comportamento futuro ao serem passados novos valores dos atributos de entrada, pois se tem os valores de α e β calculados, exibido na Figura 35.

Figura 35 – Tela de regressão linear

The screenshot shows the SIMA web application interface for Linear Regression. The main content area displays the equation $y = \sum x_i \beta_i$. The right sidebar, titled "Parâmetros", contains the following fields:

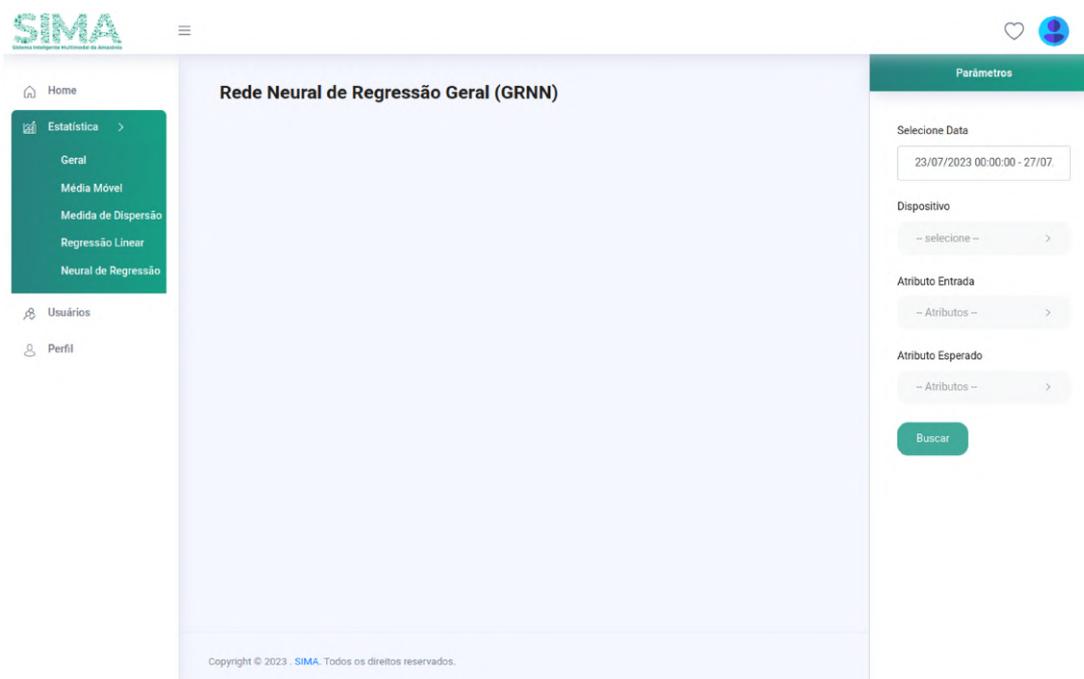
- Seleção Data: 23/07/2023 00:00:00 - 27/07
- Seleção período: Segunda-feira, Terça-feira, >
- Dispositivo: - seleccione - >
- Atributo Entrada: - Atributos - >
- Atributo Esperado: - Atributos - >

A "Buscar" button is located at the bottom of the sidebar. The footer of the page reads "Copyright © 2023 SIMA. Todos os direitos reservados."

Fonte: Elaborado pelo autor.

E para a GRNN os critérios são parecidos com a Regressão Linear, onde o usuário seleciona os atributos a serem usados como entrada da GRNN e o atributo esperado, exibido na Figura 36.

Figura 36 – Tela da GRNN

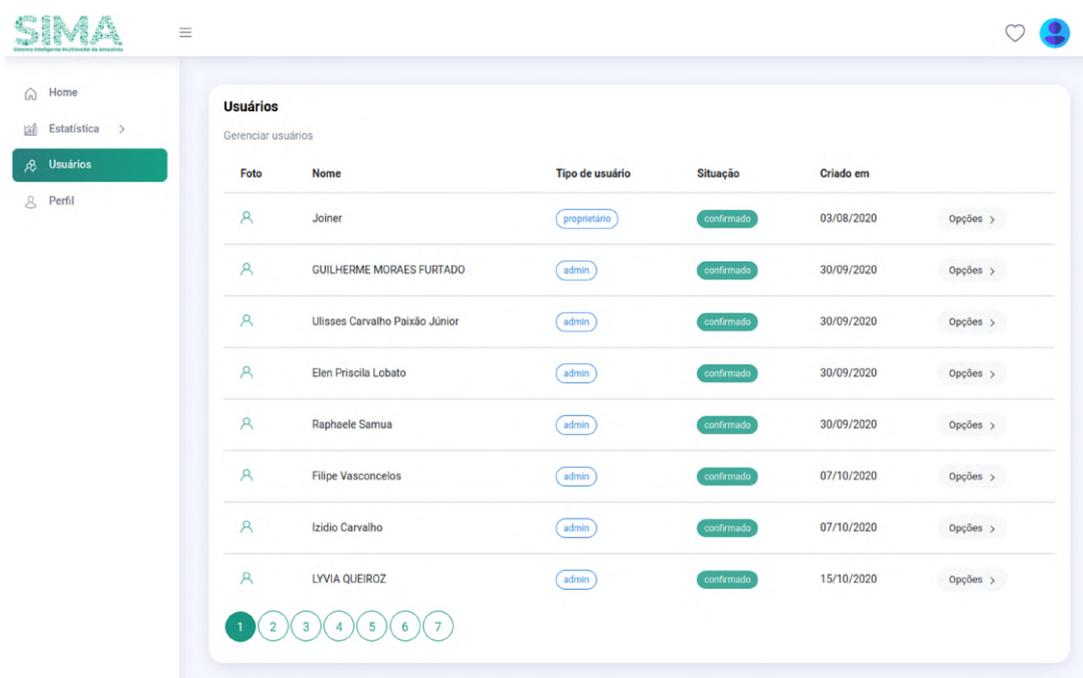


Fonte: Elaborado pelo autor.

4.4 Gerenciamento de Usuários

A próxima parte relacionada do sistema é o Gerenciamento de Usuários, onde se localiza todo o controle relacionado aos usuários cadastrados, conforme mostrado na Figura 37, se a conta foi devidamente confirmada através do *email* de confirmação, a data de criação e o nível de privilégio atual. Se o nível da conta for de administrador ou proprietário, é possível elevar uma conta de usuário comum para administrador e alterar o status da conta como: bloquear o acesso, excluir conta etc.

Figura 37 – Tela de gerenciamento de usuário



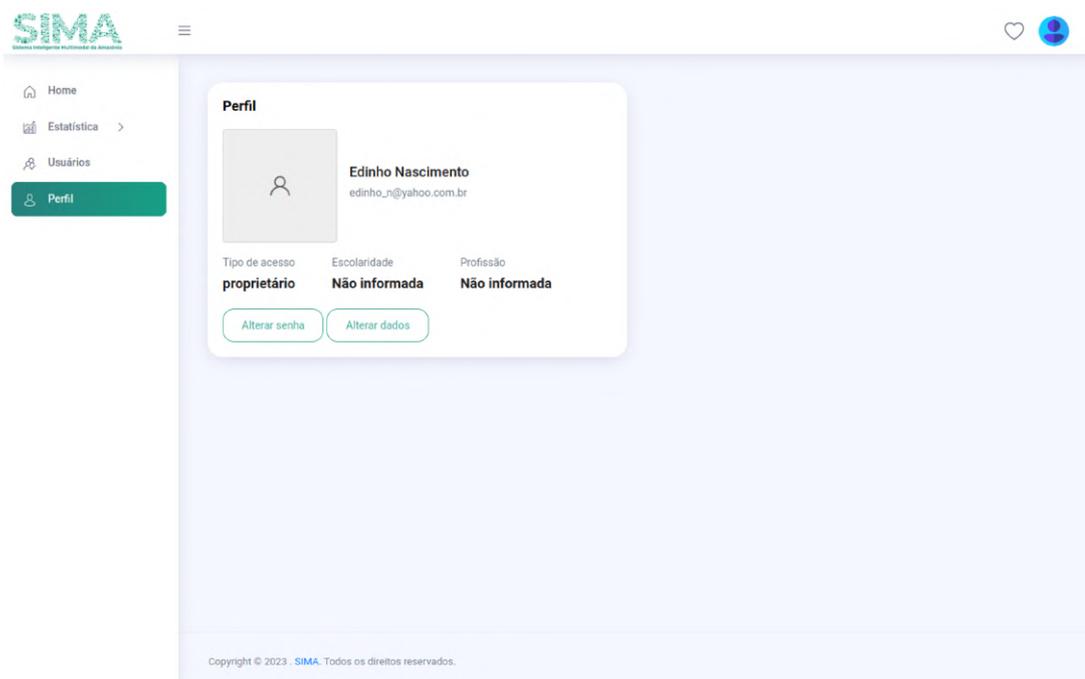
| Foto | Nome | Tipo de usuário | Situação | Criado em | |
|------|--------------------------------|-----------------|------------|------------|----------|
| | Joiner | proprietário | confirmado | 03/08/2020 | Opções > |
| | GUILHERME MORAES FURTADO | admin | confirmado | 30/09/2020 | Opções > |
| | Ulisses Carvalho Paixão Júnior | admin | confirmado | 30/09/2020 | Opções > |
| | Elen Priscila Lobato | admin | confirmado | 30/09/2020 | Opções > |
| | Rapshele Samua | admin | confirmado | 30/09/2020 | Opções > |
| | Filipe Vasconcelos | admin | confirmado | 07/10/2020 | Opções > |
| | Izidio Carvalho | admin | confirmado | 07/10/2020 | Opções > |
| | LYVIA QUEIROZ | admin | confirmado | 15/10/2020 | Opções > |

Fonte: Elaborado pelo autor.

4.5 Gerenciamento de Perfil

E a última parte do sistema é o gerenciamento do seu próprio perfil onde nele é possível visualizar as informações atuais do usuário como a foto, escolaridade, tipo de acesso e profissão. E possuindo dois botões abaixo, no qual possui função para alterar a senha e para atualizar as informações de cadastro. Conforme a Figura 38.

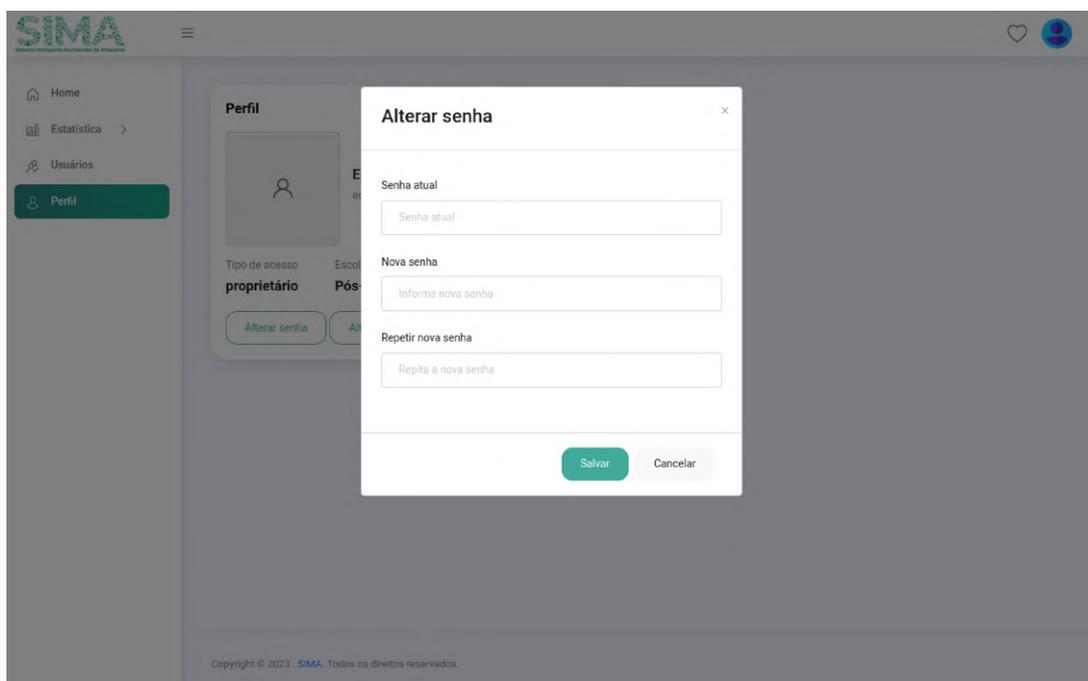
Figura 38 – Tela de gerenciamento de perfil



Fonte: Elaborado pelo autor.

Ao clicar no botão de Alterar Senha, é exibido uma tela onde o usuário coloca a senha antiga e a nova senha para a qual deseja alterar, ilustrada na Figura 39.

Figura 39 – Tela de alterar a senha



Fonte: Elaborado pelo autor.

e ao clicar em no botão de Alterar Dados, será mostrado a tela com os dados atuais do usuário e seus respectivos campos. Exibida na Figura 40.

Figura 40 – Tela de alterar dados do perfil

The screenshot displays the 'Perfil' (Profile) editing interface in the SIMA system. On the left, a sidebar menu includes 'Home', 'Estatística', 'Usuários', and 'Perfil'. The main area shows a profile card with a placeholder for a profile picture and a 'Tipo de acesso' (Access type) dropdown set to 'proprietário'. Below this is an 'Alterar senha' (Change password) button. The central form contains the following fields: 'Email' (edinho_n@yahoo.com.br), 'Escolaridade' (Pós-Graduação), 'Profissão' (Estudante), and a question 'Possui mobilidade reduzida?' with radio buttons for 'Sim' and 'Não' (selected). There are also sections for 'Foto atual' and 'Nova foto perfil' with camera icons. At the bottom, there are 'Salvar' and 'Cancelar' buttons. The footer shows 'Copyright © 2023. SIMA. Todos os direitos reservados.'

Fonte: Elaborado pelo autor.

4.6 Implantação da Solução

A Universidade Federal do Pará foi criada a partir da Lei Federal nº 3.191, sancionada pelo Presidente Juscelino Kubitschek com publicação no Diário Oficial da União no dia 02 de julho de 1957 [BRASIL 1957]. Localizada as margens do Rio Guamá na cidade de Belém capital do Estado do Pará, possui uma população universitária de 61 mil estudantes [UFPA 2023], uma área estimada em 450.000 e sendo dividida em 3 campi sendo: Básico, Profissional e Saúde; ilustrada na Figura 41 com o campi Básico destacado na cor amarela, o Profissional na cor laranja e o Saúde na cor verde. No dia 28 de dezembro de 2007 o nome do campus foi alterado para Cidade Universitária Profº José da Silva Ferreira Neto [UFPA 2018].

Figura 41 – Vista aérea do campus da UFPA-Belém destacando os campi Básico em amarelo, Profissional em laranja e o Betina em verde



Fonte: [Maps 2023].

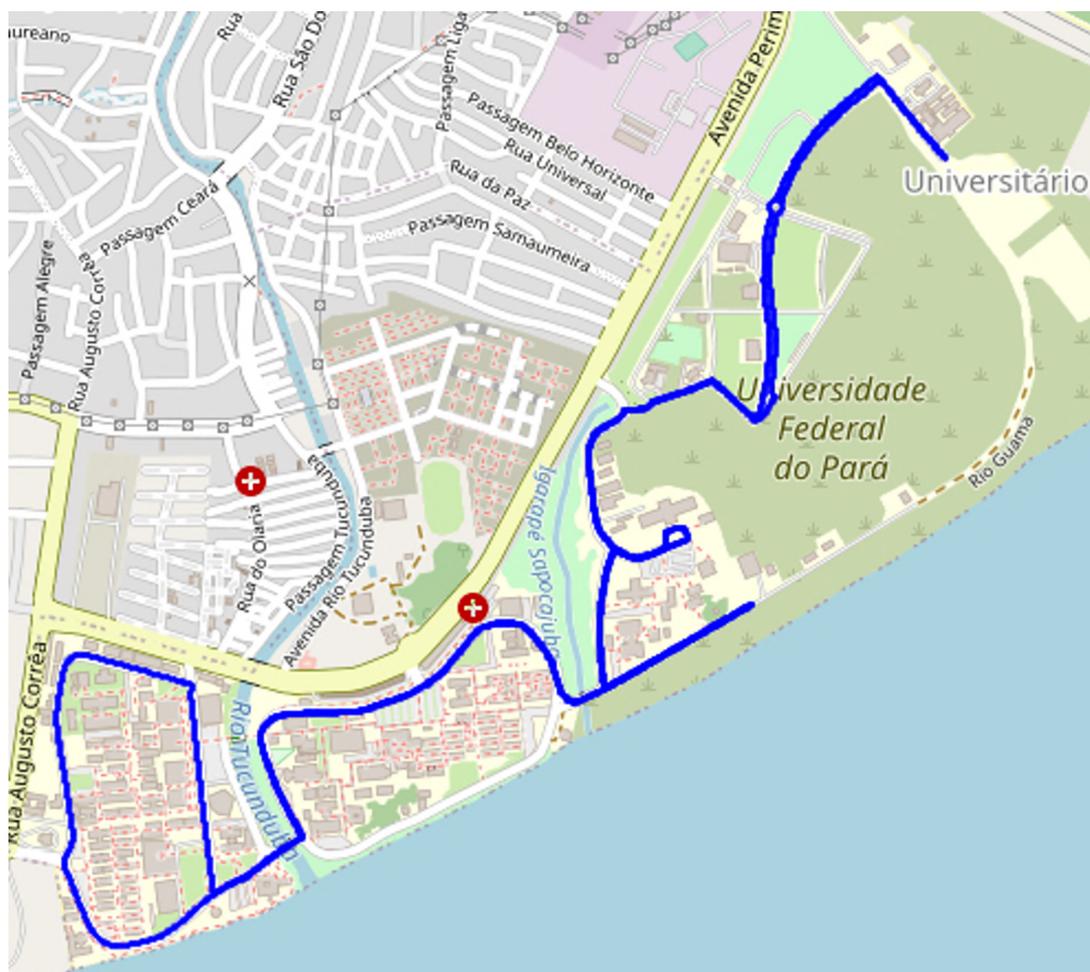
Dentro da área do campus existe a linha interna de ônibus denominada Circular que conecta os três campi levando estudantes e servidores. Essa linha é composta de 1 ônibus e 1 micro-ônibus com o uso variando dependendo do período acadêmico, no período de férias onde o campus possui pouca movimentação é utilizado o micro-ônibus e no período letivo de aulas é utilizado o ônibus. O ônibus sendo o modelo Mascarello Gran Via 2011 com chassi Mercedes OF1722 M exibido na figura 42 (a) e o micro-ônibus sendo chassi e carroceria Marcopolo Volare W9 na Figura 42 (b).

Figura 42 – Ônibus Circular em funcionamento**(a) Modelo de Ônibus Circular presente na UFPA****(b) Modelo de Micro-ônibus Circular presente na UFPA**

Fonte: [BRASIL 2013, BRASIL 2023].

O trajeto total da linha possui quase 11 km de extensão conectando os três campi simultaneamente conforme a figura 43, a velocidade permitida dos veículos é limitada a 30 km/h [UFPA 2017].

Figura 43 – Percurso do Ônibus Circular



Fonte: Elaborado pelo autor.

4.6.1 Projeto SIMA

Com as características descritas acima, foi criado o Projeto SIMA, uma parceria da empresa Norte Energia e a UFPA, visando propor uma solução sustentável e eficiente energeticamente ao propor soluções baseadas em mobilidade elétrica para solucionar desafios relacionados à mobilidade urbana e redução da queima de combustíveis fósseis. A aplicação prática do projeto pode ser considerada para o atendimento de centros urbanos que buscam por soluções integradas, menos poluentes e sustentáveis.

Visando validar o projeto, um *Smart Campus* foi implantado na UFPA-Belém, onde é proposto a criação de uma rede de Internet das coisas e a adoção de uma cadeia de transporte do tipo multimodal com natureza elétrica, isto é, formada por meios de transporte do tipo elétricos.

A implantação na UFPA-Belém contou com a instalação de um parque de geração fotovoltaica localizado no campus Saúde no espaço CEAMAZON conforme a Figura 44.

Figura 44 – Parque de geração fotovoltaica no CEAMAZON

Fonte: [G1 2022].

Com a energia sendo gerada no parque, é realizada a distribuição através de dois pontos de eletropostos, estando eles localizados no campus Básico da UFPA próximo ao Ginásio de Esportes e o segundo no Campus Saúde da UFPA no CEAMAZON, na Figura 45 é mostrado o eletroposto localizado no Ginásio de Esportes, sendo usados para alimentar veículos elétricos do Projeto SIMA assim como de quem frequentam as dependências da Universidade.

Figura 45 – Eletroposto localizado no Ginásio da UFPA

Fonte: [UFPA 2020].

Por fim, o projeto também contou com a aquisição de 2 (dois) ônibus de tipo urbano e rodoviário, os quais são utilizados para realizar o transporte dos frequentadores do *Smart*

Campus, sendo um ônibus Rodoviário Marcopolo Viaggio 1050 com chassi BYD D9A 20.410 de piso alto com autonomia de 250 km [BYD 2023], um ônibus urbano Marcopolo Torino-E com chassi BYD D9W 20.410 de piso baixo e autonomia de 240 km denominado Circular Elétrico [Silva et al. 2023] e um barco estilo catamarã que está em fase de desenvolvimento, na Figura 46 (a) é mostrado o ônibus rodoviário e na Figura 46 (b) o ônibus urbano, ambos os veículos já presentes na UFPA.

Figura 46 – Ônibus Urbano e Rodoviário



(a) Modelo de Ônibus rodoviário elétrico



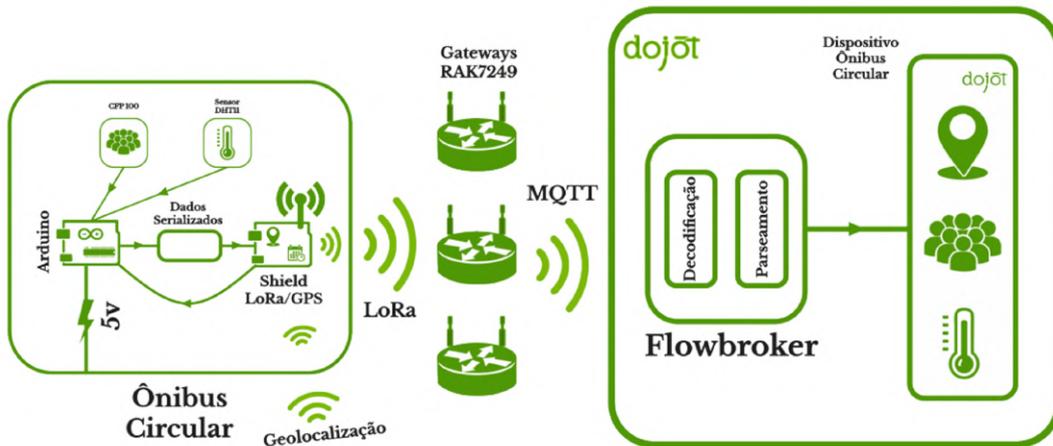
(b) Modelo de Ônibus urbano elétrico

Fonte: [UFPA 2020].

Na parte interna da frota, serão instalados diversos conjuntos *endnodes*, dispositivos que podem capturar dados como geolocalização, temperatura interna e externa dos veículos, corrente e tensão da bateria assim como sua porcentagem de carga e temperatura. Atualmente os dois ônibus possuem *endnodes* que capturam a sua geolocalização, temperatura interna e contagem de passageiros [CARVALHO 2023].

Os dados coletados serão enviados para o banco de dados em tempo real Dojot onde serão persistidos para futuro uso pelo sistema *web*, com o tempo de envio para a Dojot tendo um intervalo de 5 segundos. O envio ocorrerá de duas formas dependendo do veículo. O ônibus urbano circular elétrico e o barco enviam seus dados através de uma rede sem fio dentro do espaço da UFPA constituída da tecnologia LoRa na frequência de 915Mhz, conectando os *endnodes* da frota a um dos três *gateways* disponíveis no Campus localizados nos prédios: Mirante do Rio, Laboratório de Engenharia Elétrica e Computação, CEAMAZON [CARVALHO 2023]. Na Figura 47 é exibido a arquitetura de captura e envio de dados do ônibus Circular.

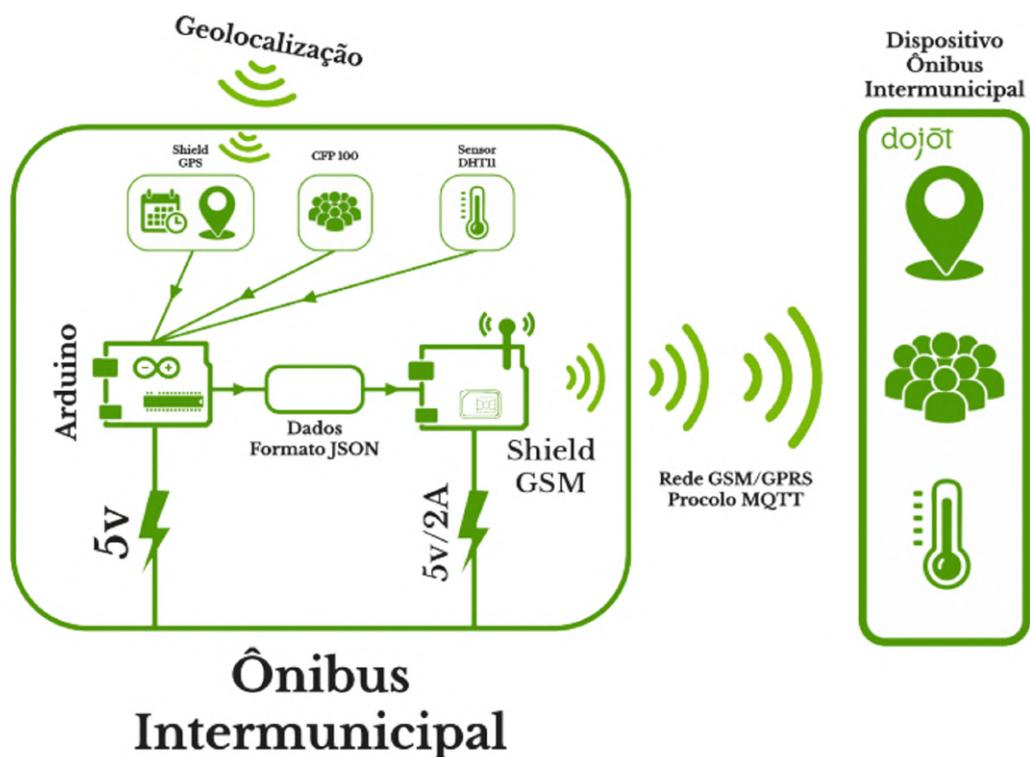
Figura 47 – Esquema de captura e envio de dados do ônibus Circular



Fonte: [CARVALHO 2023].

O ônibus rodoviário por fazer o trajeto entre as cidades de Belém e Castanhal, foi utilizado a tecnologia *Global System for Mobile Communications 2G* (GSM) com um sistema de reinicialização caso perca conexão com a rede [CARVALHO 2023], sua arquitetura é exibida na Figura 48.

Figura 48 – Esquema de captura e envio de dados do ônibus rodoviário



Fonte: [CARVALHO 2023].

Porém, alguns dados não enviados como autonomia, quilometragem, corrente e tensão da bateria. Para isto, foi desenvolvido um simulador utilizando Python 3 no qual busca exemplificar

os veículos em seu percurso de trabalho, com o foco sendo os veículos rodando dentro da UFPA, sendo o Circular Elétrico e o barco e enviando esses dados para a Dojot. O simulador será detalhado na subseção a seguir.

Com os dados persistidos, o sistema *web* irá realizar a consulta baseado no dia corrente, por semana, mês ou por um período especificado pelo usuário, no qual os dados serão agrupados nos respectivos períodos para poder utilizar os algoritmos de regressão linear, GRNN, etc. no qual será exibido para o usuário as respostas desses algoritmos de forma gráfica.

O sistema *web* também terá função de exibir a localização dos veículos em tempo real no mapa, assim como dentro do campus da UFPA Belém indicar a melhor rota entre dois pontos quaisquer do campus e levando em consideração se o percurso será realizado de barco, a pé ou de ônibus, dando uma indicação de tempo que o veículo chegará no seu ponto de parada correspondente.

Na subseção a seguir, serão detalhados os cenários de teste para a validação ferramentas utilizadas para a criação do sistema *web*.

4.7 Cenários de Teste

Como dito anteriormente, foi desenvolvido um simulador em Python 3 que gera dados de temperatura, geolocalização, quilometragem, entre outros. Os dados gerados são utilizados para criação da base de dados utilizada pelo sistema de gestão de eficiência energética.

O sistema é responsável consumir e utilizar os dados armazenados durante a realização dos cálculos estatísticos a fim de demonstrar o funcionamento do *software*. Os valores de algumas variáveis são gerados de forma aleatória enquanto outras se baseiam em cálculos e valores de variáveis dependentes. Para exemplificar, a Tabela 6 apresenta os atributos que contém os valores gerados.

Tabela 6 – Criação de variáveis do simulador

| Variável | Geração |
|---------------------------|----------------------------|
| Localização | Vetor de posições mapeadas |
| Tensão | 380v \pm 5% [BYD 2023] |
| Corrente | Aleatória |
| Temperatura Da Bateria | Aleatória (30°-70°) |
| Autonomia | 250 km [SILVA 2021] |
| Quilometragem | 0 |
| Porcentagem De Carga | 100% |
| Temperatura Interna | Aleatória (25°-30°) |
| Temperatura Externa | Aleatória (30°-35°) |
| Quantidade de Passageiros | Aleatória (0-51) |

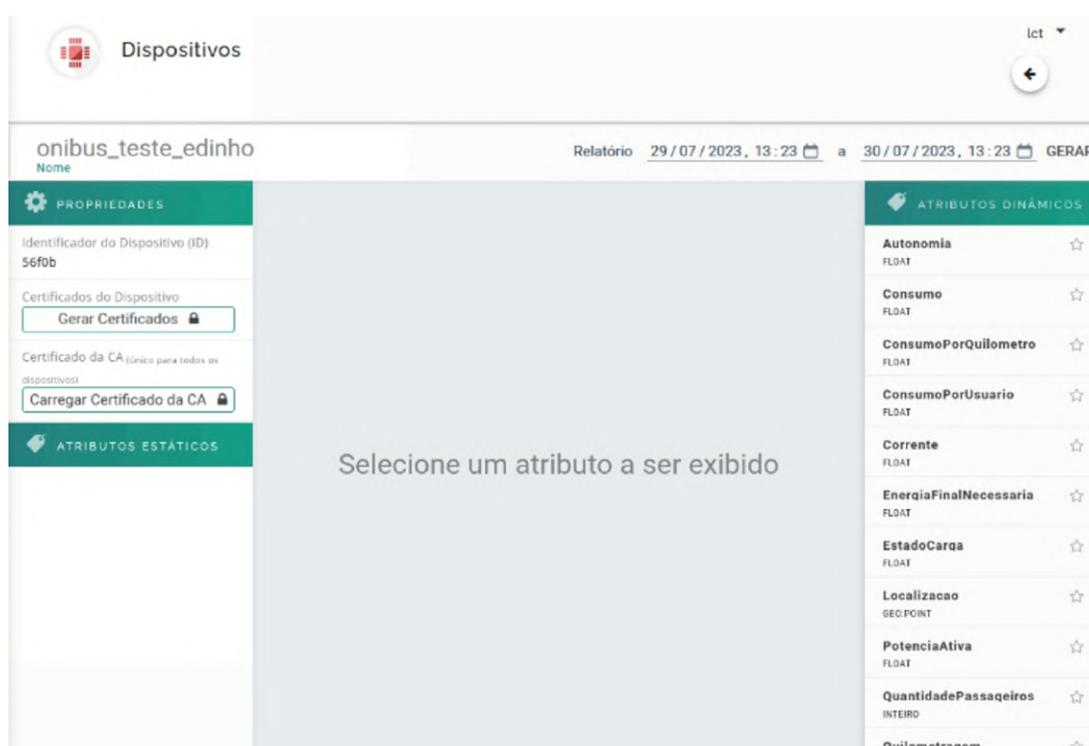
Fonte: Elaborado pelo autor.

Todas as variáveis tem seus valores atualizados a cada 5 segundos, que ocorre quando o índice do vetor que armazena as geolocalizações do ônibus é atualizado, porém as variáveis não

aleatórias como autonomia, quilometragem e porcentagem da carga, atualizam seus valores a partir de algumas relações, por exemplo: é calculado a distância em metros em relação do índice do vetor de localização atual com o índice anterior, a autonomia é diminuída de acordo com esse valor enquanto a quilometragem é aumentada, já a porcentagem de carga é diminuída de acordo com uma estimativa baseada na relação de quantos por cento de bateria é consumida a cada metro. E a quantidade de passageiros é atualizada quando o ônibus chega nos pontos de ônibus, onde é realizado a geração aleatória de quantas pessoas entram e saem do ônibus, com a diferença resultando nos passageiros dentro do modal.

Para o recebimento dos dados provenientes do simulador em Python 3, foi criado na Dojot o *device* `onibus_teste`, no qual para acesso a Dojot definiu o `device_id` como “56f0b”, onde nele ficam armazenados os dados enviados do simulador e ficam disponíveis para consulta pelo sistema *web*. Exibido na Figura 49.

Figura 49 – Device criado na Dojot



Fonte: Elaborado pelo autor.

Para testar os algoritmos implementados, foram utilizados os dados do dia 10 de julho até 21 de julho de 2023, onde no primeiro cenário esses dados serão utilizados para o cálculo de média e desvio padrão, no qual serão analisadas o retorno dado de forma gráfica na forma de gráfico de barra.

No segundo cenário será utilizado o cálculo de média móvel no qual o retorno será em forma de gráfico de barra com um gráfico de linha sobreposto indicando os valores da média móvel.

O terceiro cenário onde será utilizado a Regressão Linear, onde serão dois gráficos de linha onde o primeiro é exibido a relação das variáveis dependentes com valores previsto e esperado; e o segundo gráfico sendo exibido a média de erro.

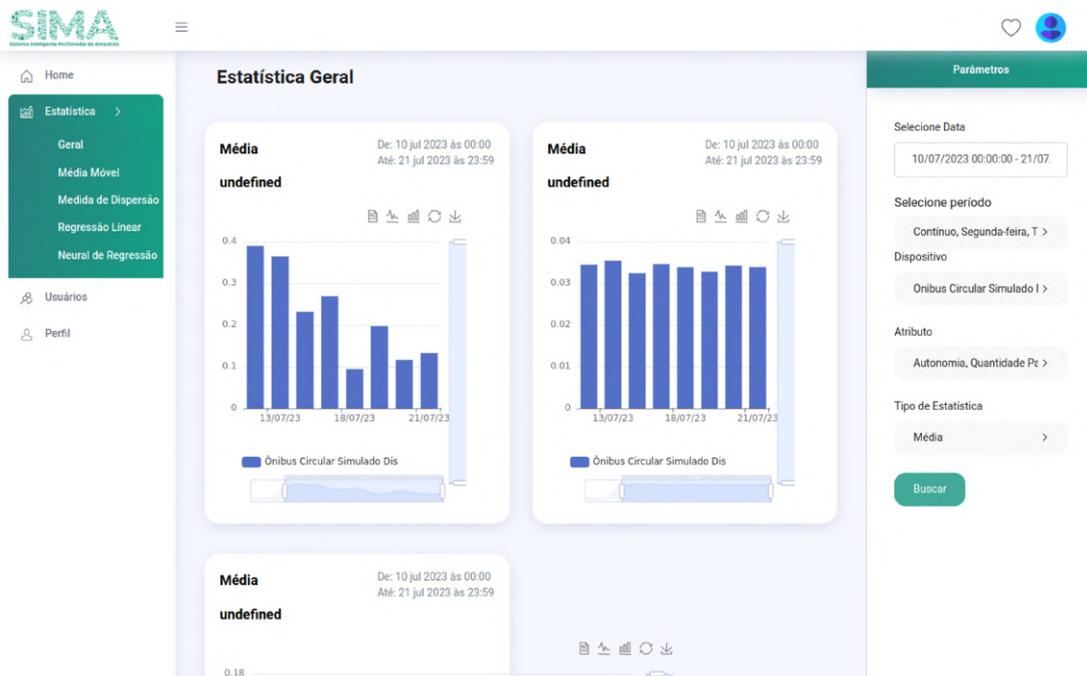
5 Resultados

Dados os detalhes do sistema *web* e do cenário para os testes nessa seção será exibido alguns dos resultados obtidos através dos algoritmos de média, média móvel, e regressão linear.

5.1 Cenário 1 - Média

Para os resultados deste cenário foi utilizado os dados enviados do dia 10 a 21 de julho, dados esses enviados pelo simulador destacado na seção anterior durante o horário comercial da UFPA, onde na Figura 50, o sistema retorna ao usuário a estatística geral representado pela estatística de média onde foram usadas as variáveis autonomia e quantidade de passageiros, com essas informações vindos do *endnode* localizado dentro do ônibus com a exibição sendo os dados divididos por dia. A partir desses dados é possível estimar algumas informações como: evolução da temperatura em determinado dia, média de passageiros durante a semana, horários de pico, quilometragem percorrida entre as semanas, evolução do uso dos modais pelos passageiros, entre outros.

Figura 50 – Cenário de média (estatísticas gerais)



Fonte: Elaborado pelo autor.

5.2 Cenário 2 - Média Móvel

Para a média móvel foi utilizado o mesmo período de captura e dados utilizados no cenário 1, a Figura 51 representa forma gráfica de como se comporta a média móvel no sistema, onde resultasse no gráfico de barras com as médias obtidas e sobreposto a elas representasse a linha da média móvel, sendo os dados divididos por dia. Foi utilizada as variáveis Autonomia, Estado da carga, Quilometragem e Quantidade de Passageiros. Como a média móvel suaviza os dados, é possível ver qual é o valor da tendência obtida a partir dos dados enviados durante o mês, por exemplo: se o modal está perdendo autonomia, verificar os valores de corrente, tensão e a temperatura da bateria durante os meses de uso, podendo indicar algum problema que requeira a sua substituição. Assim como, variações de temperaturas interna e externa, o interesse pelo serviço ao comparar a quantidade de passageiros durante as semanas de atividade dos modais, a evolução do tempo de carregamento das baterias, a taxa de entrada e saída de passageiros nos pontos de ônibus em relação aos pontos de parada.

Figura 51 – Cenário de Média Móvel



Fonte: Elaborado pelo autor.

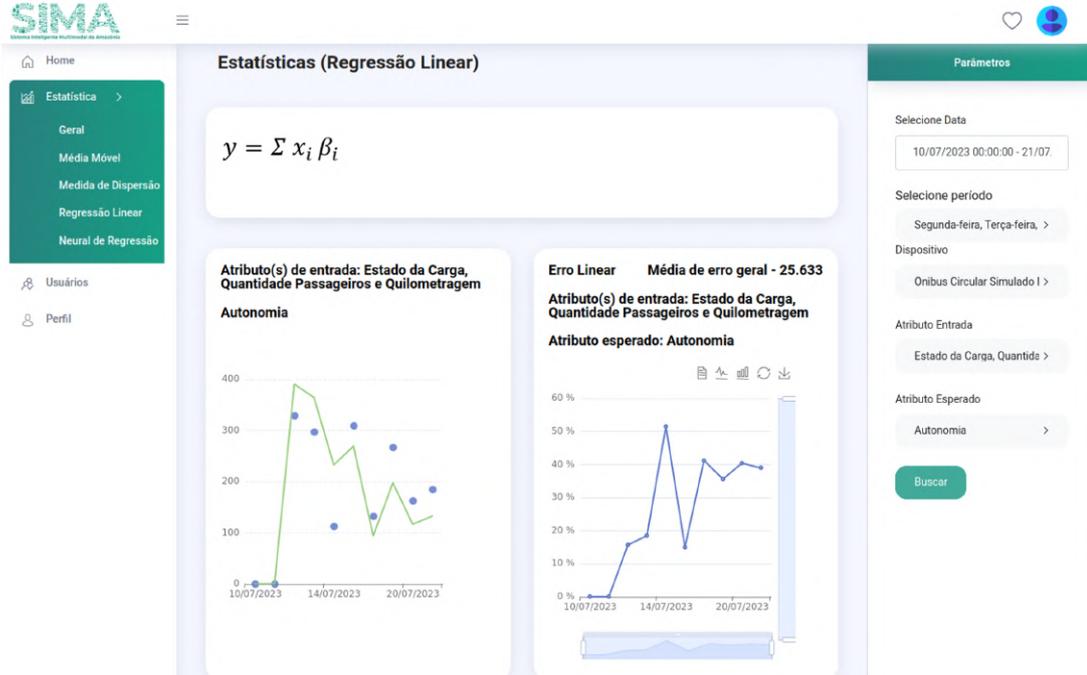
5.3 Cenário 3 - Regressão Linear

No último cenário para os resultados da regressão linear foram utilizados os dados descritos nas subseções anteriores, na regressão precisa de no mínimo duas variáveis para o seu uso que são: as variáveis independentes que no sistema é chamada de Atributo de entrada podendo ser uma variável ou mais e a variável dependente que no caso é chamada de Atributo Esperado, podendo escolher apenas uma. Para a Figura 52 como Atributo de Entrada foi utilizado as variáveis Estado da Carga, Quantidade de Passageiros e Quilometragem, no Atributo Esperado foi utilizado a Autonomia. Onde no gráfico é dividido a regressão por dia no lado esquerdo e a média do erro geral no lado direito.

A técnica de regressão linear pode ser usada em uma análise como precisa, por exemplo analisar valores de tensão, corrente, autonomia e temperatura em comparação com o estado da

carga da bateria, verificar se precisa ou não ser trocada; a predição da carga da bateria em relação a quilometragem e autonomia.

Figura 52 – Cenário de regressão linear



Fonte: Elaborado pelo autor.

6 Conclusão

Este trabalho apresentou o desenvolvimento do sistema *web* de suporte na tomada de decisões para gestão de eficiência energética, unindo dados dos modais elétricos, exibindo sua localização em tempo real, retornando a menor rota entre dois pontos, e prevendo o horário de chegada em determinado ponto selecionado no mapa.

Este sistema foi desenvolvido usando como base a implementação do *Smart Campus* da UFPA – Campus Belém, porém com as devidas adaptações, pode ser usado em outros projetos envolvendo *Smart Campus* ou *Smart Cities*.

A partir dos resultados, é possível concluir que a solução baseada em *software* é capaz de gerar dados estatísticos a partir das informações coletadas pelos sensores localizados nos veículos elétricos. Além disso, é possível concluir que dados de sensores IoT capturados em ambiente real podem ser ferramentas úteis para a tomada de decisão baseada em inteligência artificial e aprendizagem de máquina.

Os resultados também apresentaram os módulos estatísticos propostos no trabalho. Com base nos dados enviados para o banco de dados Dojot, foram realizados testes de validação dos algoritmos, sendo apresentados resultados obtidos com: regressão linear, média móvel, etc. Por essa razão, é possível concluir que algoritmos de inteligência artificial e aprendizagem de máquina podem ser ferramentas úteis para a gestão da eficiência energética a partir de dados coletados de uma rede IoT.

6.1 Trabalhos Futuros

Para o futuro deste trabalho, espera-se anexar a solução na versão final do projeto SIMA, passando a ser possível a utilização de dados reais coletados no *Smart Campus* da UFPA para a tomada de decisão sobre a gestão da eficiência energética em tempo real.

Finalizar a rede neural de regressão geral e sua implementação ao sistema *web*, ficando disponível a sua utilização para os usuários.

Além disso, de forma técnica, espera-se implementar mais soluções baseadas em inteligência artificial para incrementar o sistema de tomada de decisões, como por exemplo, implantando algoritmo genético e outras técnicas bioinspiradas.

Referências

- AFREEN, H.; BAJWA, I. S. An iot-based real-time intelligent monitoring and notification system of cold storage. **IEEE Access**, IEEE, v. 9, p. 38236–38253, 2021. Citado na página 17.
- AGARKAR, P.; HAJARE, P.; BAWANE, N. Optimization of generalized regression neural networks using pso and ga for non-performer particles. In: IEEE. **2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)**. [S.l.], 2016. p. 103–107. Citado na página 42.
- AMAZONMICROSERVICOS. **Microserviços**. 2020. <<https://aws.amazon.com/pt/microservices/>>. [Accessed 24-jul-2023]. Citado na página 26.
- ANAS, S. A. B.; SINGH, R. S. S.; KAMARUDIN, N. A. B. Designing an iot agriculture monitoring system for improving farmer’s acceptance of using iot technology. **Engineering, Technology & Applied Science Research**, v. 12, n. 1, p. 8157–8163, 2022. Citado na página 22.
- ARDTECHS. **Solution for Campus Environments**. 2020. <<https://www.ardtechs.com/smart-systems-solutions/smart-cities-and-public-safety/smart-campus/>>. [Accessed 24-jul-2023]. Citado na página 26.
- ARDUINO. **Arduino**. 2023. <<https://www.arduino.cc/reference/pt/>>. [Accessed 24-jul-2023]. Citado na página 29.
- AWS, A. **MQTT**. 2023. <<https://aws.amazon.com/pt/what-is/mqtt/>>. [Accessed 24-jul-2023]. Citado na página 29.
- BLOG, N. **Rede Sigfox: o que é e onde aplicar?** 2023. <<https://www.novida.com.br/blog/sigfox/>>. [Accessed on 24-jul-2023]. Citado na página 29.
- BRASIL. **Lei 3.191, de 2 de agosto de 1957**. 1957. Citado na página 67.
- BRASIL, O. **Ônibus Brasil - circular menor**. 2013. <<https://onibusbrasil.com/guilhermejsousa/1678766/>>. [Accessed 24-jul-2023]. Citado na página 69.
- BRASIL, O. **Ônibus Brasil - circular maior**. 2023. <<https://onibusbrasil.com/chacal/11053876?context=company>>. [Accessed 24-jul-2023]. Citado na página 69.
- BYD. **BYD D9A 20.410 - Padron piso alto**. 2023. <<https://www.byd.com.br/chassi-byd-d9a-20-410/>>. [Accessed 24-jul-2023]. Citado 2 vezes nas páginas 72 e 74.
- CARVALHO, J. A. R. **Proposição De Hardware E Softwares Para Transmissão De Dados Via Lora E Gsm: Estudo De Caso No Smart Campus Da Ufpa**. Tese (Master’s thesis) — Universidade Federal do Pará, 2023. Citado 3 vezes nas páginas 45, 72 e 73.
- CASTRO, G. de. **LoRaWAN - Conceitos Básicos - Tutoriais - RoboCore**. 2022. <<https://www.robocore.net/tutoriais/lorawan-conceitos-basicos/>>. [Accessed on 24-jul-2023]. Citado na página 29.

CHALLAPALLI, S. S. N. et al. Web development and performance comparison of web development technologies in node. js and python. In: IEEE. **2021 International Conference on Technological Advancements and Innovations (ICTAI)**. [S.l.], 2021. p. 303–307. Citado na página 46.

CLOVITY. **Smart School & Virtual Students: A Clovity, Qualcomm, & Microsoft Collaboration**. 2020. <<https://www.clovity.com/smart-campus-proposition>>. [Accessed 24-jul-2023]. Citado na página 25.

CPQD. **CPQD lança nova versão da plataforma aberta dojot**. 2020. <<https://www.cpqd.com.br/release/cpqd-lanca-nova-versao-da-plataforma-aberta-dojot/>>. [Accessed 24-jul-2023]. Citado na página 26.

DEV.TO. **Cloud: Multi-Tenant Architecture and its Issues**. 2020. <<https://dev.to/sciencebae/multi-tenant-architecture-and-it-s-issues-h06>>. [Accessed 24-jul-2023]. Citado na página 29.

DOCS, D. **Arquitetura Dojot**. 2020. <https://dojotdocs.readthedocs.io/pt_BR/stable/architecture.html>. [Accessed 24-jul-2023]. Citado 3 vezes nas páginas 26, 27 e 30.

DOMÍNGUEZ-BOLAÑO, T. et al. An overview of iot architectures, technologies, and existing open-source projects. **Internet of Things**, Elsevier, p. 100626, 2022. Citado 2 vezes nas páginas 23 e 24.

DROKE, C. **Moving averages simplified**. [S.l.]: Marketplace Books USA, 2001. v. 96. Citado 2 vezes nas páginas 39 e 40.

FATECNOLOGIA. **Os impactos da Internet das Coisas no setor de transportes**. 2019. <<https://www.fatecjales.edu.br/publicacoes/fatecnologia/557-os-impactos-da-internet-das-coisas-no-setor-de-transportes>>. [Accessed 24-jul-2023]. Citado na página 23.

FIRSTTECH. **Internet das coisas no setor de serviços**. 2023. <<https://first-tech.com/conteudo/internet-das-coisas-no-setor-de-servicos/>>. [Accessed 29-jul-2023]. Citado na página 21.

FOLEY, B.; DEGIRMENCI, K.; YIGITCANLAR, T. Factors affecting electric vehicle uptake: Insights from a descriptive analysis in australia. **Urban Science**, MDPI, v. 4, n. 4, p. 57, 2020. Citado na página 16.

FRAGA-LAMAS, P. et al. Design and experimental validation of a lorawan fog computing based architecture for iot enabled smart campus applications. **Sensors**, MDPI, v. 19, n. 15, p. 3287, 2019. Citado 3 vezes nas páginas 22, 24 e 25.

G1. **Onibus eletricos comecam a circular em campi da UFPA**. 2022. <<https://g1.globo.com/pa/para/noticia/2022/04/13/onibus-eletricos-comecam-a-circular-em-campi-da-ufpa.ghtml>>. [Accessed 24-jul-2023]. Citado na página 71.

HEUSER, C. A. **Projeto de banco de dados**. [S.l.]: Bookman Editora, 2009. Citado 4 vezes nas páginas 34, 35, 36 e 37.

KAFKA. **Apache Kafka**. 2020. <https://www.wikiwand.com/en/Apache_Kafka>. [Accessed 24-jul-2023]. Citado 2 vezes nas páginas 27 e 28.

KHALED, A. E. et al. Iot-ddl-device description language for the 't' in iot. **IEEE access**, IEEE, v. 6, p. 24048–24063, 2018. Citado na página 21.

- LFTECHNOLOGY. **Getting Started with a Multi-tenant Application on Node.js**. 2019. <<https://blog.lftechnology.com/designing-a-secure-and-scalable-multi-tenant-application-on-node-js-15ae13dda778>>. [Accessed 24-jul-2023]. Citado na página 28.
- LOBATO, E. P. D. S. **Implantação de um middleware IoT escalável para aplicações de Mobilidade Elétrica Multimodal**. Tese (Master's thesis) — Universidade Federal do Pará, 2022. Citado 2 vezes nas páginas 16 e 44.
- MAPS, G. **Mapa da Universidade Federal do Pará**. 2023. <<https://goo.gl/maps/VtYQEsh1G9vZrFGW7>>. [Accessed 24-jul-2023]. Citado na página 68.
- MARIADB. **Maria DB**. 2023. <<https://mariadb.org>>. [Accessed 24-jul-2023]. Citado na página 47.
- MELO, L. A. et al. Avaliação da qualidade ambiental de um ambiente de smart home utilizando internet of things e inteligência artificial. In: SBC. **Anais do XIII Workshop de Computação Aplicada à Gestão do Meio Ambiente e Recursos Naturais**. [S.l.], 2022. p. 41–50. Citado 3 vezes nas páginas 17, 18 e 22.
- MUÑOZ, R. et al. Integration of iot, transport sdn, and edge/cloud computing for dynamic distribution of iot analytics and efficient use of network resources. **Journal of Lightweight Technology**, IEEE, v. 36, n. 7, p. 1420–1428, 2018. Citado na página 22.
- NEGREIROS, I. et al. Smart campus® as a living lab on sustainability indicators monitoring. In: IEEE. **2020 IEEE International Smart Cities Conference (ISC2)**. [S.l.], 2020. p. 1–5. Citado na página 16.
- NUNES, D. **Protocolos de mensageria: AMQP, MQTT e STOMP - dnune's Blog**. 2018. <<https://dnunesblog.wordpress.com/2018/07/06/protocolos-de-menssageria-amqp-mqtt-e-stomp/>>. [Accessed on 24-jul-2023]. Citado na página 29.
- PLAGERAS, A. P. et al. Efficient iot-based sensor big data collection–processing and analysis in smart buildings. **Future Generation Computer Systems**, Elsevier, v. 82, p. 349–357, 2018. Citado 2 vezes nas páginas 21 e 22.
- PRESSMAN, R. S. **Engenharia de Software: uma abordagem profissional**. [S.l.]: Amgh Editora, 2011. Citado 2 vezes nas páginas 35 e 37.
- SA, J. et al. Online monitoring of buses information using knn, atr and dmc algorithms. **IEEE Latin America Transactions**, IEEE, v. 17, n. 04, p. 564–572, 2019. Citado na página 18.
- SAIF, M.-Z.; MM-TOROK, A. **Public transport accessibility: A literature review** *Periodica Polytechnica Transportation Engineering* 47 (1): 36-43. 2018. Citado na página 16.
- SARAIVA, A.; ONU. **ONU-Habitat: população mundial será 68% urbana até 2050**. 2022. <<https://brasil.un.org/pt-br/188520-onu-habitat-popula%C3%A7%C3%A3o-mundial-ser%C3%A1-68-urbana-at%C3%A9-2050#:~:text=Relat%C3%B3rio%20Mundial%20das%20Cidades%202022,de%20pessoas%20anualmente%20at%C3%A9%202050.>> [Accessed 24-jul-2023]. Citado na página 16.
- SAY, K. et al. The economics of public transport electrification: A case study from victoria, australia. **Energy Economics**, Elsevier, v. 120, p. 106599, 2023. Citado na página 16.

SEBER, G. A.; LEE, A. J. **Linear regression analysis**. [S.l.]: John Wiley & Sons, 2003. v. 330. Citado 2 vezes nas páginas 40 e 41.

SILVA, E. N. **Uma Solução De Software Back End Para Gestão E Consulta De Dados Em Bancos De Dados Relacional E Em Tempo Real: Um Caso De Estudo No Contexto Da Mobilidade Urbana**. Tese (Monografia) — Universidade Federal do Pará, 2021. Citado 4 vezes nas páginas 44, 46, 59 e 74.

SILVA, M. d. S. P. et al. Mobilidade elétrica na universidade federal do pará, construindo uma cartilha para os usuários dos modais rodoviário e fluvial. **Peer Review**, v. 5, n. 4, p. 1–19, 2023. Citado na página 72.

SOLUTIONS, T. I. **Visão geral do LWM2M**. 2023. <<https://docs.devicewise.com/Content/GettingStarted/Overview-of-LWM2M.htm>>. [Accessed on 24-jul-2023]. Citado na página 29.

SOMMERVILLE, I. **Engenharia de Software**. [S.l.]: Pearson Prentice Hall, 2011. Citado 2 vezes nas páginas 37 e 38.

SPECHT, D. F. et al. A general regression neural network. **IEEE transactions on neural networks**, Citeseer, v. 2, n. 6, p. 568–576, 1991. Citado na página 42.

SU, X.; YAN, X.; TSAI, C.-L. Linear regression. **Wiley Interdisciplinary Reviews: Computational Statistics**, Wiley Online Library, v. 4, n. 3, p. 275–294, 2012. Citado 2 vezes nas páginas 40 e 41.

SYSTEMS, A. S. **Smart Home Automation**. 2023. <<https://armsecuritysystems.com/residential/smart-home-automation/>>. [Accessed 24-jul-2023]. Citado na página 22.

SZTAJNBERG, A.; MACEDO, R. da S.; STUTZEL, M. Protocolos de aplicação para a internet das coisas: conceitos e aspectos práticos. **Sociedade Brasileira de Computação**, 2018. Citado na página 29.

UFPA. **UFPA em números**. 2023. <<https://ufpanumeros.ufpa.br>>. [Accessed 24-jul-2023]. Citado na página 67.

UFPA, P. **Circular "encurta" distâncias na UFPA e transporta diariamente milhares de passageiros**. 2017. <<https://ascom.ufpa.br/index.php/banco-de-pautas/68-servicos-da-ufpa/967>>. [Accessed 24-jul-2023]. Citado na página 69.

UFPA, P. **No dia 13 de agosto de 1968, portanto há exatos 50 anos, a UFPA inaugura o Conjunto Universitário Pioneiro, atual Campus Guamá**. 2018. <<https://ufpa.br/index.php/ultimas-noticias2/8794>>. [Accessed 24-jul-2023]. Citado na página 67.

UFPA, P. **Projeto de mobilidade elétrica iniciado na UFPA completa um ano**. 2020. <<https://www.ufpa.br/index.php/ultimas-noticias2/12205>>. [Accessed 24-jul-2023]. Citado na página 71.

UFPA, P. **Sistema Inteligente Multimodal da Amazônia recebe ônibus elétrico**. 2020. <<https://portal.ufpa.br/index.php/ultimas-noticias2/11907>>. [Accessed 24-jul-2023]. Citado na página 72.