INSTITUTO FEDERAL DO ESPÍRITO SANTO
CAMPUS CACHOEIRO DE ITAPEMIRIM
CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS DE INFORMAÇÃO

RENAN GOMES POGGIAN

PLATAFORMA PARA MONITORAMENTO E GESTÃO INTEGRADA DE UM SMART CAMPUS

Cachoeiro de Itapemirim
2024

RENAN GOMES POGGIAN

PLATAFORMA PARA MONITORAMENTO E GESTÃO INTEGRADA DE UM SMART CAMPUS

Trabalho de Conclusão de Curso apresentado à Coordenadoria do Curso de Sistemas de Informação do Instituto Federal do Espírito Santo, Campus Cachoeiro de Itapemirim, como requisito parcial para a obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Lucas Poubel Timm do Carmo

Cachoeiro de Itapemirim

2024

(Biblioteca do Campus Cachoeiro de Itapemirim)

P746p Poggian, Renan Gomes.

Plataforma para monitoramento e gestão integrada de um smart campus / Renan Gomes Poggian. - 2024.
51 f. : il. ; 30 cm.

Orientador: Lucas Poubel Timm do Carmo

TCC (Graduação) Instituto Federal do Espírito Santo, Campus Cachoeiro de Itapemirim, Sistemas de Informação, 2024.

1. Internet das coisas. 2. Redes de computador - Protocolos. 3. Educação Administração. I. Carmo, Lucas Poubel Timm do . II.Título III. Instituto Federal do Espírito Santo.

CDD: 004.678

Bibliotecário/a: Renata Lorencini Rizzi CRB6-ES nº 085

MINISTÉRIO DA EDUCAÇÃO
INSTITUTO FEDERAL DO ESPÍRITO SANTO
CAI - COORDENADORIA DO CURSO DE BACHARELADO EM SISTEMAS DE INFORMACAO

FOLHA DE APROVAÇÃO-TCC Nº 5 / 2024 - CAI-CCSI (11.02.18.01.08.02.13)

Nº do Protocolo: 23151.003505/2024-83

Cachoeiro De Itapemirim-ES, 09 de dezembro de 2024.

RENAN GOMES POGGIAN
PLATAFORMA PARA MONITORAMENTO E GESTÃO INTEGRADA DE UM SMART CAMPUS
Trabalho de Conclusão de Curso apresentado à Coordenadoria do Curso de Sistemas de Informação do Instituto Federal do Espírito Santo, Campus Cachoeiro de Itapemirim, como requisito para a obtenção do título de Bacharel em Sistemas de Informação. Orientador: Prof. Dr. Lucas Poubel Timm do Carmo

Aprovado em 09 de dezembro de 2024
COMISSÃO EXAMINADORA
Dr. Lucas Poubel Timm do Carmo Instituto Federal Do Espírito Santo Orientador
Dr. Igor Henrique Beloti Pizetta Instituto Federal Do Espírito Santo
Dr. Rafael Silva Guimarães Instituto Federal Do Espírito Santo

(Assinado digitalmente em 10/12/2024 13:40)
IGOR HENRIQUE BELOTI PIZETTA
PROFESSOR DO ENSINO BASICO TECNICO E TECNOLOGICO
CAI-CCTE (11.02.18.01.08.02.09)
Matrícula: 1026928

(Assinado digitalmente em 09/12/2024 15:06)
LUCAS POUBEL TIMM DO CARMO
PROFESSOR DO ENSINO BASICO TECNICO E TECNOLOGICO
CAI-CCSI (11.02.18.01.08.02.13)
Matrícula: 2417426

(Assinado digitalmente em 09/12/2024 15:09)

RAFAEL SILVA GUIMARAES
PROFESSOR DO ENSINO BASICO TECNICO E TECNOLOGICO
CAI-CCSI (11.02.18.01.08.02.13)
Matrícula: 1919203

Visualize o documento original em **https://sipac.ifes.edu.br/public/documentos/index.jsp** informando seu número: **5**, ano: **2024**, tipo: **FOLHA DE APROVAÇÃO-TCC**, data de emissão: **09/12/2024** e o código de verificação: **c4dfe687ae**

DECLARAÇÃO DO AUTOR

Declaro, para fins de pesquisa acadêmica, didática e técnico-científica, que este Trabalho de Conclusão de Curso pode ser parcialmente utilizado, desde que se faça referência à fonte e ao autor.

Cachoeiro de Itapemirim, 09 de Dezembro de 2024.

Renan Gomes Poggian

Dedico esse trabalho primeiramente a Deus e a todos que de alguma forma contribuíram para que o mesmo fosse realizado.

AGRADECIMENTOS

A realização deste trabalho só foi possível graças ao apoio e incentivo de muitas pessoas, às quais sou profundamente grato.

Agradeço, em primeiro lugar, à minha família, por todo o amor, paciência e motivação inabalável que me deram ao longo dessa jornada. Vocês são meu alicerce e minha inspiração.

Aos meus amigos, que compartilharam comigo momentos de alegria e descontração, mas também me ofereceram apoio nos períodos de maior dificuldade. Obrigado por estarem sempre por perto.

Expresso minha gratidão aos professores e orientadores que contribuíram com sua orientação, ensinamentos e estímulo ao pensamento crítico, especialmente ao professor Lucas Poubel, cuja dedicação e conhecimento foram fundamentais para o sucesso deste trabalho.

Por fim, sou grato àqueles que, direta ou indiretamente, contribuíram para que este momento se tornasse realidade. Cada palavra de incentivo, cada gesto de apoio, e cada conversa construtiva tiveram um impacto significativo ao longo dessa caminhada.

A todos, o meu mais sincero obrigado.

"Não importa o quão devagar você vá, desde que você não pare."
Confúcio

RESUMO

Com o avanço da Internet das Coisas (IoT), os campi universitários têm adotado o conceito de Smart Campus, visando a integração tecnológica para otimizar operações, promover sustentabilidade e melhorar a experiência dos usuários. Este trabalho apresenta o desenvolvimento de uma plataforma de gestão integrada para dispositivos IoT em um Smart Campus. A plataforma foi projetada para oferecer recursos de monitoramento e controle de dispositivos em tempo real, empregando protocolos de comunicação como MQTT e uma abordagem de banco de dados poliglota com PostgreSQL e MongoDB. Os resultados demonstraram a eficácia da plataforma em integrar dispositivos heterogêneos e fornecer suporte à tomada de decisões baseada em dados. Este estudo contribui para a construção de ambientes acadêmicos mais inteligentes e sustentáveis, além de abrir caminho para melhorias futuras, como integração com novos protocolos e implementação de algoritmos de aprendizado de máquina.

Palavras-chave: Smart Campus, Internet das Coisas, MQTT, Gestão Integrada, Sustentabilidade.

ABSTRACT

With the advancement of the Internet of Things (IoT), university campuses have adopted the concept of Smart Campus to optimize operations, promote sustainability, and enhance user experience. This study presents the development of an integrated management platform for IoT devices in a Smart Campus. The platform was designed to provide real-time device monitoring and control capabilities, employing communication protocols like MQTT and a polyglot database approach with PostgreSQL and MongoDB. The results demonstrated the platform's effectiveness in integrating heterogeneous devices and supporting data-driven decision-making. This study contributes to building smarter and more sustainable academic environments and paves the way for future improvements, such as integration with new protocols and implementation of machine learning algorithms.

Keywords: Smart Campus, Internet of Things, MQTT, Integrated Management, Sustainability.

LISTA DE FIGURAS

LISTA DE SIGLAS

ACID - Atomicidade, Consistência, Isolamento e Durabilidade

API - Application Programming Interface

CRUD - Create, Read, Update, Delete

DRF - Django Rest Framework

HTTP - HyperText Transfer Protocol

IA - Inteligência Artificial

IoT - Internet of Things (Internet das Coisas)

JSON - JavaScript Object Notation

MQTT - Message Queuing Telemetry Transport

QoS - Quality of Service

RA - Realidade Aumentada

SGBD - Sistema de Gerenciamento de Banco de Dados

SQL - Structured Query Language

SSL - Secure Sockets Layer

TLS - Transport Layer Security

Wi-Fi - Wireless Fidelity

XML - Extensible Markup Language

SUMÁRIO

1 INTRODUÇÃO

Com o avanço exponencial da Internet das Coisas (IoT), a integração de dispositivos inteligentes em ambientes urbanos, como campi universitarios, tem se tornado uma realidade inovadora e estrategica (CAVUS et al., 2022). Um conceito que vem sendo amplamente adotado em universidades é o Smart Campus.

Segundo Polin et al. (2023), o conceito de smart campus é uma evolução das instituições de ensino superior, impulsionada pela transformação digital. Os campi inteligentes são descritos como réplicas em miniatura de cidades inteligentes, atuando como laboratórios vivos para a pesquisa, desenvolvimento e adoção de novas tecnologias. Estes integram funções tradicionais de ensino e pesquisa com avançadas infraestruturas tecnológicas, abrangendo domínios como sociedade, economia, ambiente e governança, realçando a importância do uso de tecnologia digital e big data.

Os campi universitários estão adotando esse conceito cada vez mais, visando eficiência operacional, melhor experiência estudantil e sustentabilidade ambiental (VALKS MONIQUE H. ARKESTEIJN; HEIJER, 2021). Segundo Dong et al. (2020), um smart campus não seria possível sem a inovação nas tecnologias, e as principais que estão presentes nesse cenário são a Computação em Nuvem, a Realidade Aumentada (RA), a Inteligência Artificial (IA) e a IoT. Por meio da IoT, esse modelo de campus conecta dispositivos para monitorar e controlar aspectos como energia, resíduos, acesso e segurança, otimizando recursos e facilitando a tomada de decisões (ABUARQOUB et al., 2017).

Dentro do contexto de um Smart Campus, pode haver uma ampla variedade de dispositivos e sensores (hardware) e formas de comunicação entre eles (protocolos) que são empregados em diferentes soluções (JAVED et al., 2020). Essa diversidade pode tornar complexa a integração e a interoperabilidade entre dispositivos IoT nesse tipo de cenário (FERREIRA et al., 2022).

A questão da interoperabilidade no contexto dos dispositivos IoT vai além de simples desafios tecnológicos, impactando diretamente a capacidade de inovação e a eficácia

das soluções em um Smart Campus. Cada dispositivo IoT opera sob um conjunto específico de padrões e especificações técnicas, que nem sempre são compatíveis com outros sistemas em uso, criando barreiras significativas para a integração plena, onde a falta de um framework comum pode impedir a comunicação fluida entre dispositivos de diferentes domínios ou fabricantes (AIELLO, 2022). Além disso, a interoperabilidade efetiva é crucial para a coleta e análise de dados em larga escala, permitindo que decisões baseadas em dados sejam tomadas de maneira mais rápida e informada (ABDEL-BASSET et al., 2018). Sem ela, o potencial para automatização completa e gestão eficiente de recursos fica comprometido, restringindo o aproveitamento de todas as vantagens que um ambiente inteligente pode oferecer.

Além de resolver os desafios de interoperabilidade e integração, esta plataforma busca simplificar o desenvolvimento de soluções IoT ao fornecer uma estrutura robusta para gestão e monitoramento de dispositivos. Com isso, pesquisadores podem concentrar seus esforços no design, fabricação e otimização dos dispositivos, sem precisar dedicar tempo e recursos significativos à criação de sistemas de gerenciamento. Essa separação de responsabilidades permite que soluções específicas e altamente especializadas sejam desenvolvidas mais rapidamente, enquanto a plataforma central garante o gerenciamento integrado e o monitoramento eficaz dos dispositivos em operação.

1.1 OBJETIVOS

Desenvolver uma plataforma de gestão integrada de dispositivos IoT por meio de padrões de comunicação que facilitem o monitoramento das soluções de Smart Campus.

Os objetivos específicos deste trabalho são:

- Desenvolver um painel de monitoramento e gestão integrada que facilite o gerenciamento e a tomada de decisão relativa ao funcionamento dos dispositivos integrados.

- Criar um padrão de comunicação para que dispositivos IoT consigam integrar ao painel utilizando o protocolo MQTT.

- Validar esse padrão de comunicação por meio da integração de diferentes tipos de dispositivos IoT.

2 REFERENCIAL TEÓRICO

2.1 SMART CAMPUS

O conceito de Smart Campus representa uma evolução na gestão e operação de instituições de ensino superior, impulsionada pela integração de tecnologias avançadas. De acordo com Muhamad et al. (2017), a ideia básica de um Smart Campus é um esforço para integrar um conjunto de tecnologias inteligentes avançadas pela Universidade para otimizar operações, melhorar a eficiência energética, e promover experiências enriquecedoras para estudantes e demais membros da comunidade acadêmica.

Um dos principais objetivos de um Smart Campus é a melhoria da experiência do usuário. Através da implementação de soluções tecnológicas, como aplicativos móveis e sistemas de navegação indoor, é possível proporcionar aos alunos, professores e funcionários uma experiência mais personalizada e conveniente no campus (AL-FUQAHA et al., 2015). Outra área de foco crítica é o aumento da sustentabilidade do campus. A implementação de sensores e sistemas de automação permite o monitoramento dos recursos hídricos e energéticos em tempo real. Isso pode resultar em uma redução significativa no consumo de energia e na pegada de carbono da instituição (DONG et al., 2020).

Estratégias robustas de segurança cibernética e políticas de privacidade bem definidas são cruciais para proteger as informações sensíveis e garantir a confiança da comunidade acadêmica. Como menciona Dong et al. (2020), a comunicação de rede de um Smart Campus pode facilmente ser insegura pois o sistema do campus geralmente é aberto, por isso é fácil para que intrusos tenha acesso a dados e realize ataques. Portanto, é importante um aumento da redundância dos dados e que se tenha um bom sistema de proteção contra ataques de negação de serviço. Em resumo, um Smart Campus representa uma evolução significativa na gestão e operação de instituições de ensino superior, aproveitando a tecnologia para melhorar a eficiência, a experiência do usuário e a sustentabilidade ambiental.

2.2 INTERNET DAS COISAS

A Internet das Coisas (IoT) representa uma revolução tecnológica que tem transformado a maneira como interagimos com o ambiente ao nosso redor. Segundo Gubbi et al. (2013), a IoT é definida como a interconexão de objetos físicos, veículos, prédios e outros itens incorporados com sensores, software e conectividade para coletar e trocar dados. Esses objetos, também conhecidos como "coisas", são capazes de coletar e transmitir informações em tempo real, proporcionando uma visão mais detalhada e contextual do mundo físico.

A IoT oferece um vasto leque de aplicações em diversos setores, desde saúde e agricultura até cidades inteligentes e ambientes industriais. Segundo Atzori, Iera e Morabito (2010), a IoT tem o potencial de criar um impacto significativo na eficiência operacional, na qualidade de vida e na inovação de serviços em muitos setores econômicos. Essa capacidade de transformação é especialmente evidente em contextos urbanos, onde a integração de dispositivos inteligentes pode levar à criação de ambientes mais eficientes e sustentáveis.

Para viabilizar a comunicação entre os dispositivos, a IoT depende de uma infraestrutura de rede robusta. Isso inclui a utilização de tecnologias como Wi-Fi, Bluetooth, Zigbee, LoRaWAN, entre outras, para estabelecer conexões confiáveis e de baixa latência. Conforme apontado por Borgia (2014), a escolha da tecnologia de comunicação é crucial para determinar a eficácia e a escalabilidade de um sistema IoT.

2.3 ARQUITETURA DE SISTEMAS IOT

Segundo Al-Fuqaha et al. (2015), a IoT terá a capacidade de conectar bilhões de objetos heterogêneos através da internet. Isso gera uma necessidade crucial de uma arquitetura flexível, estruturada em camadas. Até o momento, apesar do aumento no número de propostas de arquiteturas, ainda não houve uma convergência em direção a um modelo de referência (KRčO; POKRIć; CARREZ, 2014). Contudo, diversos projetos estão em andamento, visando criar uma arquitetura comum ao analisar as demandas tanto dos pesquisadores quanto da indústria.

Dentre os modelos propostos, o modelo básico é a Arquitetura de 3 camadas, composta pelas camadas de aplicação, rede e percepção. No entanto, recentemente, foram propostos outros modelos que adicionam níveis adicionais de abstração à arquitetura da IoT, tendo o modelo de 5 camadas como o mais aceitado (AL-FUQAHA et al., 2015). A seguir uma breve apresentação do modelo 5 camadas:

(a) Camada de Objetos (também chamada de camada de Percepção)

Neste nível estão os aparelhos responsáveis por coletar e processar informações. Sensores e atuadores desempenham variadas funções, como a obtenção de localização, vibração, temperatura, peso, movimento, umidade, aceleração, entre outros (RAMESH; REDDY; REDDY, 2021; AL-FUQAHA et al., 2015). Para configurar objetos diversos, é necessário o uso de mecanismos padronizados "plug and play"nessa camada (NAVANI; JAIN; NEHRA, 2017).

(b) Camada de Abstração de Objetos

Segundo Navani, Jain e Nehra (2017), na camada de Abstração de Objetos, os dados originados na camada de Dispositivos são encaminhados de maneira segura para a camada de Gerenciamento de Serviços. Para essa transferência, são empregadas várias tecnologias, incluindo 3G, GSM, UMTS, Wi-Fi, Bluetooth de Baixa Energia, infravermelho, ZigBee, entre outras (AL-FUQAHA et al., 2015). Além disso, nesta camada, são realizadas funções como o gerenciamento de dados e a execução de processos de computação em nuvem.

(c) Camada de Gerenciamento de Serviços

Sua função principal é agilizar o processamento de informações, suportar tomadas de decisões e controlar o emparelhamento de informações para tarefas pertinentes (RAMESH; REDDY; REDDY, 2021). Essencialmente, essa camada possibilita que os desenvolvedores de aplicativos para IoT lidem com uma variedade de objetos sem se preocupar com uma plataforma de hardware específica.

(d) Camada de Aplicação

A camada de aplicação fornece aos clientes instalações inteligentes de alta qualidade de acordo com a pré-solicitação deles. Por exemplo, a camada de aplicação pode fornecer medidas de temperatura e umidade do ar para o cliente interessado nos dados relevantes (AL-FUQAHA et al., 2015). Segundo Wu et al. (2010), esta camada desempenha um papel importante em impulsionar a Internet das Coisas para um desenvolvimento em larga escala, devido a esta camada fornecer todos os tipos de aplicativos para cada setor.

(e) Camada de Negócios

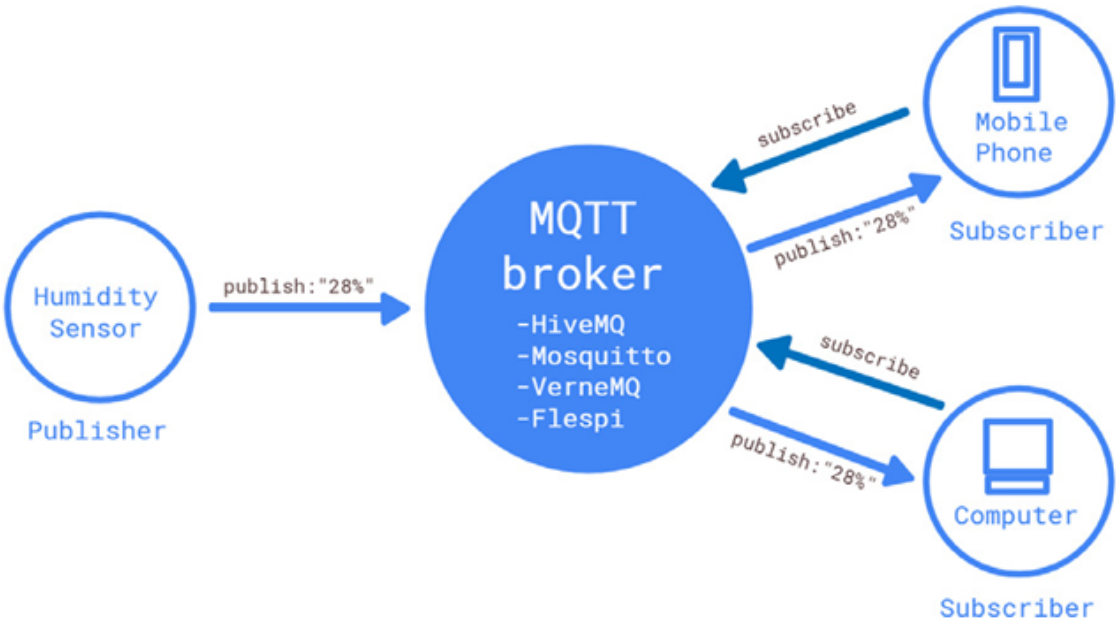
A camada de negócios oferece uma representação do modelo de negócios e dos dados recebidos da camada de aplicação. Esta camada gerencia as atividades e serviços gerais do sistema IoT (KHAN et al., 2012). Também suporta processos de tomada de decisão com base na análise de Big Data. Além disso, o monitoramento e gerenciamento das quatro camadas subjacentes são realizados nesta camada (AL-FUQAHA et al., 2015).

2.4 COMUNICAÇÃO MQTT

A IoT tem se consolidado como um dos pilares para a construção de ambientes inteligentes, como os Smart Campi, onde a integração e a comunicação eficiente entre dispositivos são essenciais. Neste contexto, o Protocolo MQTT (Message Queuing Telemetry Transport) se destaca pela sua capacidade de oferecer uma comunicação leve e eficaz, suportando assim a conectividade entre uma ampla gama de dispositivos em redes com diferentes características de desempenho e confiabilidade. Este protocolo baseia-se em um modelo de publicação/assinatura, permitindo que dispositivos com recursos limitados comuniquem-se de maneira eficiente, transmitindo dados de maneira confiável mesmo em redes com alta latência ou limitações de largura de banda (SONI; MAKWANA, 2017; KEGENBEKOV; SAPAROVA, 2022).

Além disso, a flexibilidade do MQTT em termos de configuração de tópicos e qualidade de serviço (QoS) permite adaptar o fluxo de mensagens às necessidades específicas

Figura 1 – Modelo de Publicação/Assinatura MQTT.



Fonte: (MILEVA et al., 2021)

de cada aplicação, garantindo assim que os dados críticos sejam entregues com a prioridade e a confiabilidade necessárias (KEGENBEKOV; SAPAROVA, 2022). A implementação de tais tecnologias em um Smart Campus não apenas otimiza a gestão de recursos, mas também enriquece a experiência dos usuários, ao facilitar o acesso a informações em tempo real e melhorar a segurança e a sustentabilidade do ambiente.

No âmbito educacional, a aplicação do MQTT estende-se ao controle e à comunicação com dispositivos restritos de recursos, evidenciando a importância deste protocolo na promoção de uma aprendizagem mais interativa e tecnologicamente integrada (PRADA et al., 2016). A adoção do MQTT em tais cenários destaca a relevância da escolha estratégica de tecnologias de comunicação na implementação de soluções IoT, onde a eficiência e a adaptabilidade são cruciais para o sucesso do projeto.

2.4.1 Qualidade de serviço (Quality of Service, QoS)

O protocolo MQTT também se destaca pela flexibilidade oferecida através de seus níveis de QoS, que definem a confiabilidade na entrega de mensagens entre cliente e broker. Segundo Detti, Funari e Blefari-Melazzi (2020), existem três níveis de QoS:

QoS 0 - At most once: A mensagem é entregue no máximo uma vez, sem garantias de

que o receptor irá recebê-la. Este nível é o mais rápido e utiliza menos recursos, sendo adequado para aplicações onde a perda de algumas mensagens não é crítica.

QoS 1 - At least once: A mensagem é entregue pelo menos uma vez. O broker armazena a mensagem até que receba uma confirmação do receptor, garantindo maior confiabilidade, mas podendo ocasionar mensagens duplicadas.

QoS 2 - Exactly once: A mensagem é entregue exatamente uma vez. Este é o nível mais confiável e também o mais caro em termos de recursos, envolvendo um handshake adicional entre o broker e o cliente para evitar duplicações. É ideal para aplicações críticas, onde mensagens duplicadas ou perdidas não são aceitáveis.

2.4.2 Persistents Session e Clean Sessions

O protocolo MQTT oferece dois modos principais de gerenciamento de sessão: Persistent Session e Clean Session. A escolha entre esses modos impacta significativamente a confiabilidade da comunicação e a gestão de recursos, especialmente em sistemas IoT com dispositivos frequentemente desconectados ou operando em condições de rede instáveis. No modo Clean Session, o cliente solicita ao broker que descarte todas as informações relacionadas à sessão assim que a conexão for encerrada, incluindo tópicos de inscrição, mensagens não entregues e outros estados associados à conexão (VELINOV et al., 2019). Este modo é apropriado para aplicações que demandam começar sempre com um estado limpo, sem mensagens ou inscrições armazenadas, ou que são de natureza transacional, onde as mensagens trocadas não precisam ser armazenadas para entregas futuras. Ele também é útil em dispositivos com recursos limitados, que desejam minimizar o uso de memória no broker. Contudo, em casos de desconexões frequentes, a Clean Session apresenta limitações, pois todas as informações da sessão são perdidas, sendo necessário recriá-las manualmente (VELINOV et al., 2019).

Por outro lado, no modo Persistent Session, o broker mantém o estado do cliente mesmo após uma desconexão (MILEVA et al., 2021). Nesse cenário, informações como tópicos de inscrição do cliente, mensagens não entregues (dependendo do nível de QoS configurado) e dados de estado associados à sessão são armazenadas,

tornando este modo particularmente útil em dispositivos IoT móveis ou em áreas com cobertura de rede instável. Ele é também indicado para aplicações críticas que exigem entrega confiável de mensagens mesmo após falhas temporárias de conexão e para cenários onde há necessidade de evitar reinscrições frequentes em tópicos, reduzindo a sobrecarga operacional.

A decisão entre Clean Session e Persistent Session deve considerar os requisitos específicos da aplicação IoT. Por exemplo, para sensores de telemetria que reportam dados continuamente, como temperatura, o uso de Clean Session pode ser mais eficiente, uma vez que não há necessidade de manter o estado entre as conexões. Já em sistemas críticos, como monitoramento de segurança ou controle de acesso, Persistent Session é mais adequada, pois evita a perda de mensagens importantes em situações de falha de conexão (VELINOV et al., 2019). Dessa forma, o protocolo MQTT oferece flexibilidade para atender a uma ampla gama de requisitos de comunicação, equilibrando desempenho e confiabilidade.

2.5 TRABALHOS CORRELATOS

O conceito de Smart Campus é frequentemente associado à aplicação de tecnologias avançadas para otimizar a experiência acadêmica e promover a sustentabilidade. Trabalhos correlatos ao tema deste TCC variam de abordagens teóricas a soluções práticas, incluindo áreas além do contexto educacional.

Domínguez-Bolaño et al. (2024) desenvolveram um sistema IoT para monitoramento e controle de dispositivos heterogêneos em campi universitários, abordando desafios como interoperabilidade e escalabilidade. Embora semelhante ao presente trabalho no foco na integração de dispositivos, o estudo utiliza ferramentas conhecidas como InfluxDB e Grafana. Em contrapartida, este TCC adota uma abordagem customizada, baseada no protocolo MQTT e em bancos de dados poliglota, o que permite uma integração mais robusta e flexível.

Deshmukh et al. (2021) propuseram o Data Spine, um sistema para promover a interoperabilidade em ecossistemas IoT heterogêneos, especialmente no setor de manufatura. O Data Spine utiliza uma arquitetura federada que possibilita a criação

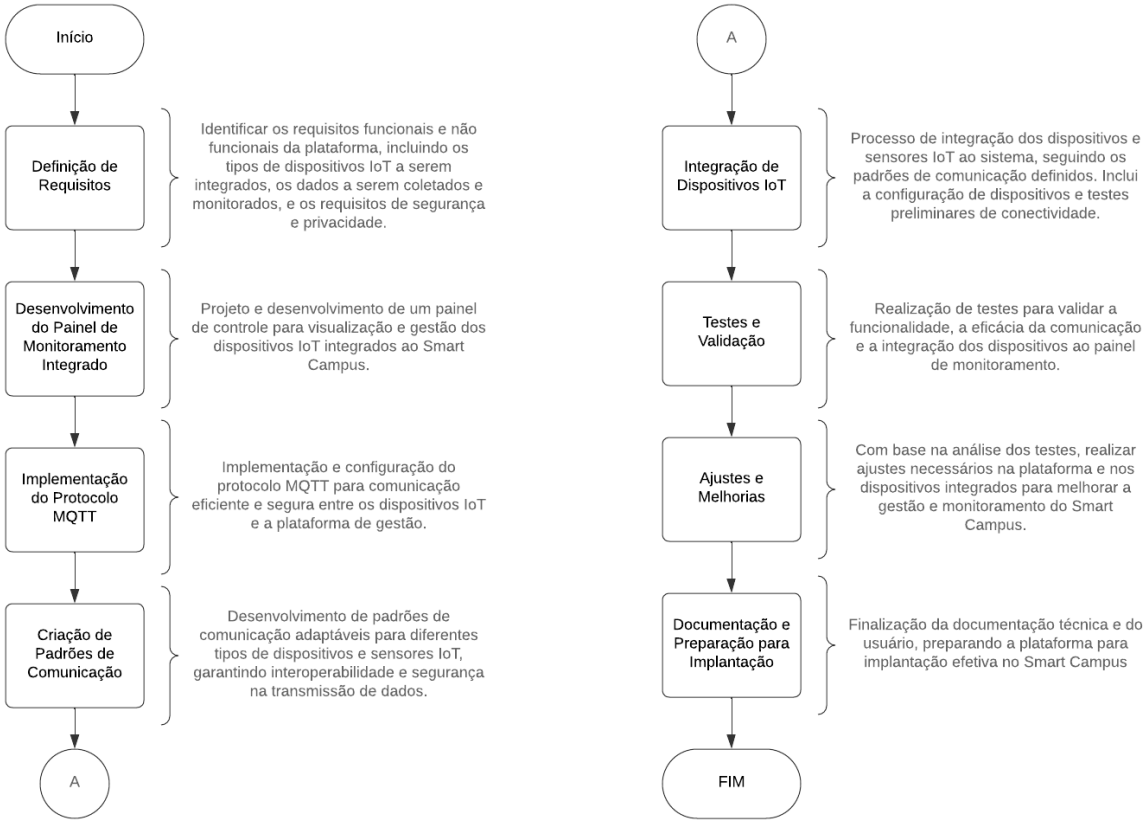
de aplicações compostas e promove a integração entre plataformas de diferentes fornecedores, enfrentando problemas de fragmentação e protocolos incompatíveis. Apesar de não ter como foco o Smart Campus, o Data Spine é semelhante a este TCC em termos de integração, mas difere ao focar em múltiplas plataformas IoT e usar um ambiente de desenvolvimento de baixo código. Já o presente trabalho foca em soluções voltadas especificamente para Smart Campus, integrando dispositivos IoT por meio de um backend especializado que suporta gerenciamento e monitoramento em tempo real.

Esses trabalhos demonstram como o tema de gestão integrada em IoT é abordado de formas diversas, permitindo posicionar o presente TCC como uma contribuição prática que avança o estado da arte, focando em desafios específicos de interoperabilidade e gestão em Smart Campus.

3 MATERIAL E MÉTODOS

A metodologia deste trabalho foi estruturada para desenvolver e implementar uma plataforma de gestão integrada de dispositivos IoT, com foco no monitoramento e na gestão de um Smart Campus. A abordagem metodológica adotada envolve várias etapas, descritas na figura a seguir:

Figura 2 – Fluxograma Processos da Metodologia



Fonte: O autor

3.1 ANÁLISE DE REQUISITOS

A análise de requisitos é uma etapa fundamental no desenvolvimento de qualquer sistema. Nesta fase, foram identificados os requisitos funcionais e não funcionais para a plataforma de gestão integrada. Isso incluiu a identificação dos tipos de dispositivos IoT que seriam utilizados, os dados a serem monitorados e as funcionalidades esperadas da plataforma, como o painel de monitoramento e a integração de diferentes padrões de comunicação.

3.2 DESENVOLVIMENTO DA PLATAFORMA

Com base nos requisitos levantados, foi desenvolvida uma plataforma de gestão integrada utilizando tecnologias web e frameworks adequados para suportar a comunicação em tempo real com os dispositivos IoT. A plataforma foi projetada para ser modular e escalável, permitindo a fácil adição de novos dispositivos e sensores.

3.2.1 O Frontend

Para desenvolver a interface gráfica da plataforma optamos pelo React, uma biblioteca Javascript popular para construção de interfaces interativas e dinâmicas. O React oferece diversas vantagens para o desenvolvimento de aplicações web. A componentização, por exemplo, permite a construção de componentes reutilizáveis, facilitando a manutenção e a expansão do código. Cada componente encapsula sua lógica e apresentação, permitindo um desenvolvimento modular e organizado.

Além disso, o React é muito utilizado pelos desenvolvedores, sendo possível encontrar diversos componentes e ferramentas disponibilizados pela comunidade. A comunicação entre o frontend em React e o backend em Django Rest Framework foi realizada por meio de requisições HTTP utilizando a biblioteca Axios. Por exemplo, para obter uma lista de dispositivos, foi configurada uma chamada à API utilizando Axios, que faz uma requisição GET à rota correspondente e retorna os dados para serem utilizados no componente React. Este processo garante uma integração eficiente entre o frontend e o backend, permitindo a atualização dinâmica dos dados exibidos na interface do usuário.

3.2.2 O Backend

O backend da plataforma é responsável por gerenciar a lógica de negócios, acesso aos dados e comunicação entre os componentes da aplicação. Para este projeto, escolhemos o Django Rest Framework (DRF) como a base para o desenvolvimento do backend devido à sua robustez, facilidade de uso e integração eficiente com o Django, um dos frameworks web mais populares em Python.

3.2.2.1 Escolha do Django Rest Framework

O Django Rest Framework (DRF) é conhecido por facilitar a criação de APIs RESTful de forma organizada e eficiente, o que foi fundamental para implementar funcionalidades como o cadastro de dispositivos IoT, o envio de comandos e a comunicação em tempo real com o frontend.

Além disso, o DRF possui um sistema integrado de autenticação e controle de permissões, que foi usado para garantir que apenas usuários autorizados pudessem acessar ou modificar os dados da plataforma. Isso foi crucial para proteger informações sensíveis, como os tópicos MQTT dos dispositivos e as métricas coletadas em tempo real. Outra vantagem importante foi a serialização de dados, que tornou mais fácil transformar as informações do banco de dados em respostas para o frontend, mantendo tudo consistente e seguro.

O DRF também foi escolhido porque tem uma ótima documentação e uma comunidade muito ativa. Isso facilitou bastante na hora de resolver problemas durante o desenvolvimento e também deixou o projeto preparado para futuras melhorias.

3.2.2.2 Arquitetura do Backend

A arquitetura escolhida para o backend, ilustrada na Figura 3, é a padrão quando se trabalha com o framework Django. Ela utiliza uma estrutura modular que permitiu organizar o código de forma clara e facilitar futuras expansões. Os modelos foram utilizados para definir a estrutura de dados, mapeando entidades como usuários, dispositivos IoT e projetos no banco de dados relacional PostgreSQL. Cada modelo foi planejado para garantir a integridade dos dados e suportar as operações necessárias ao funcionamento da plataforma.

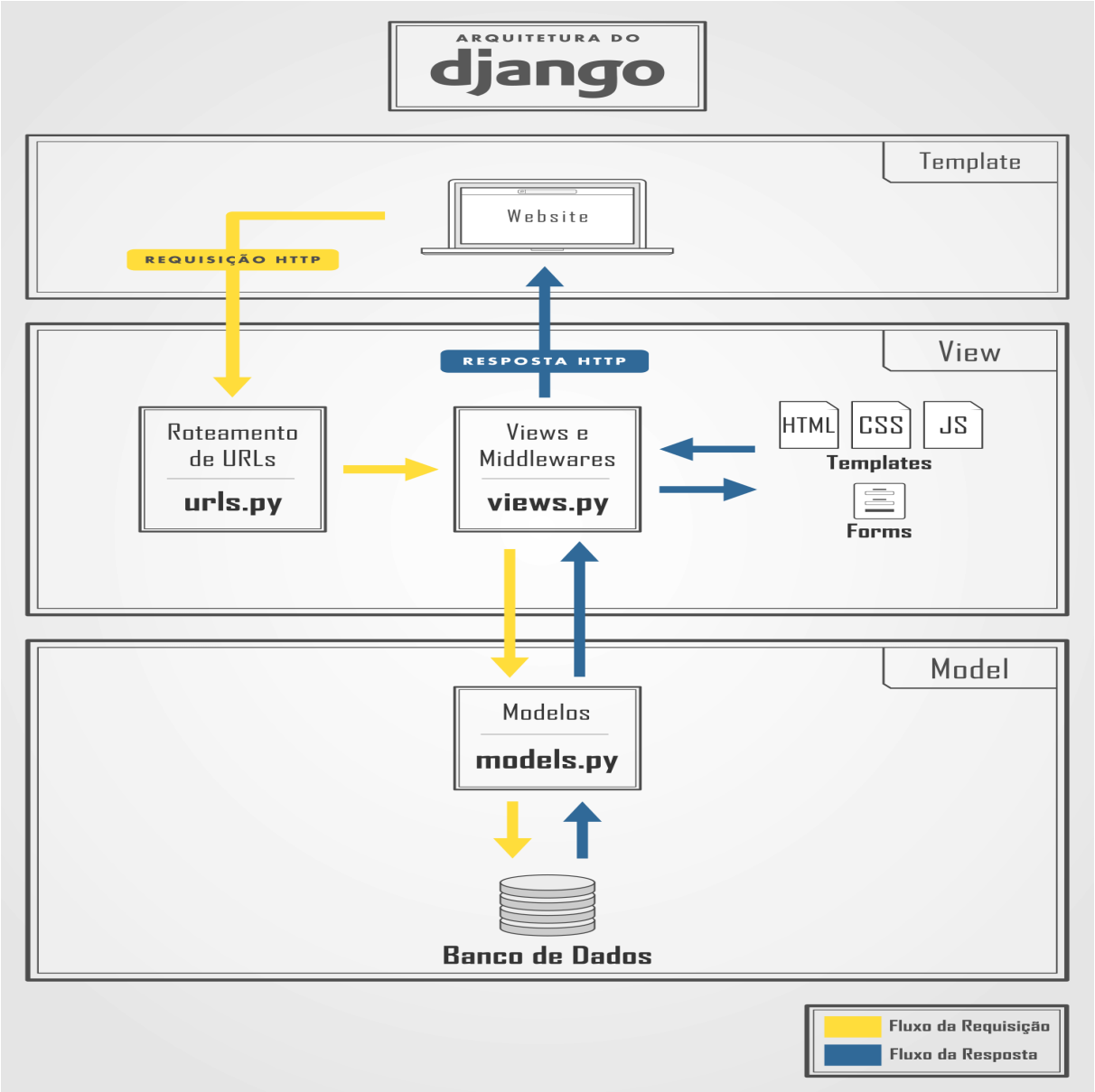
As views e viewsets foram responsáveis por processar as requisições enviadas pelo frontend e aplicar a lógica de negócios de maneira eficiente. Por exemplo, as views gerenciam ações como cadastro de dispositivos e envio de comandos MQTT, garantindo que cada requisição seja processada de forma segura e rápida. O uso de serializers foi indispensável para converter os dados do banco em respostas JSON compreensíveis para o frontend, além de permitir a validação das informações recebidas antes de salvar

no banco.

Em relação à segurança, apenas usuários autenticados podem acessar os recursos, e as permissões configuradas garantem que cada operação seja realizada por quem tem autorização.

Outro ponto importante foi a configuração das URLs e rotas, que organizam os end-points da API e permitem que as funcionalidades sejam acessada pelo frontend de forma intuitiva e consistente. O roteador do DRF simplificou essa tarefa, especialmente ao lidar com os viewsets, que agrupam as operações CRUD de maneira otimizada.

Figura 3 – Arquitetura do Django



Fonte: https://pythonacademy.com.br/

3.2.3 O banco de dados

A escolha do banco de dados é uma etapa muito importante pois ele é responsável pelo armazenamento, recuperação e gerenciamento dos dados essenciais para o funcionamento da plataforma. Para este projeto, foram escolhidos dois sistemas de banco de dados complementares: PostgreSQL, um banco relacional, e MongoDB, um banco NoSQL baseado em documentos.

3.2.3.1 PostgreSQL

O PostgreSQL é um sistema de gerenciamento de banco de dados (SGBD) relacional. O paradigma relacional organiza os dados em tabelas, permitindo uma estruturação lógica clara e coerente. Cada tabela pode ser relacionada a outras através de chaves primárias e estrangeiras, facilitando a manutenção da integridade referencial.

Uma das características positivas dos bancos relacionais, presente também no PostgreSQL, é a presença de um mecanismos robustos para garantir a integridade dos dados. As transações ACID (Atomicidade, Consistência, Isolamento e Durabilidade), asseguram que todas as operações em uma transação sejam concluídas com sucesso ou revertidas em caso de falha (MAKRIS et al., 2021).

Outro benefício do banco relacional é o suporte à linguagem SQL (Structured Query Language), que possui uma sintaxe declarativa que permite expressar consultas complexas de maneira concisa e intuitiva, facilitando a extração e manipulação de dados. Ademais, a linguagem SQL é extensível, permitindo a criação de procedimentos armazenados, gatilhos e funções customizadas, o que adiciona flexibilidade e poder ao gerenciamento de dados. Além disso, o SQL possui uma ampla adoção pela comunidade, com uma documentação abundante e uma grande variedade de ferramentas de desenvolvimento e administração (MAKRIS et al., 2021).

3.2.3.2 MongoDB

Além do PostgreSQL, utilizado para armazenar dados estruturados relacionados aos usuários, dispositivos e operações da plataforma, o MongoDB foi escolhido para gerenciar os dados não estruturados enviados pelos dispositivos IoT. Essa combinação de bancos de dados, conhecida como abordagem poliglota, foi essencial para atender

às diferentes necessidades de armazenamento e processamento da plataforma (KHINE; WANG, 2019).

O MongoDB foi selecionado devido à sua capacidade de lidar com grandes volumes de dados não estruturados e com estruturas variáveis, características comuns no contexto de dispositivos IoT. O modelo de documentos do MongoDB, que elimina a necessidade de esquemas fixos, permitiu que dados com diferentes formatos fossem armazenados e processados sem a necessidade de reconfigurações frequentes no banco de dados (ABOUDOUMAT et al., 2024).

Outra razão para a escolha deste banco de dados foi sua escalabilidade horizontal, que permite distribuir dados entre múltiplos servidores. Isso garante um desempenho estável mesmo em cenários onde o número de dispositivos conectados e o volume de dados aumentam significativamente. Além disso, a linguagem de consulta baseada em JSON do MongoDB simplificou o processo de extração e análise de informações diretamente dos documentos, tornando possível processar dados complexos de maneira eficiente e integrada à plataforma.

Na plataforma desenvolvida, o MongoDB foi utilizado para armazenar os dados enviados pelos dispositivos através dos tópicos MQTT. Cada dispositivo publica suas informações em tópicos específicos, que são processados pelo backend e registrados no banco. Essa configuração permitiu não apenas o armazenamento eficiente, mas também a rápida recuperação de dados para o painel de monitoramento, onde as métricas de cada dispositivo são exibidas.

3.3 IMPLEMENTAÇÃO DOS PROTOCOLOS DE COMUNICAÇÃO

No contexto deste projeto, a comunicação entre os dispositivos IoT e a plataforma é fundamental para o monitoramento em tempo real e a coleta de dados de telemetria. Para garantir uma comunicação eficiente e escalável, foi adotado o protocolo MQTT, amplamente utilizado em sistemas de Internet das Coisas devido à sua leveza, eficiência e capacidade de operar em redes com largura de banda limitada ou intermitente.

Para implementar o protocolo, foi escolhido o Mosquitto como broker, um servidor

central que recebe e encaminha mensagens entre publicadores e assinantes. O Mosquitto é uma implementação de código aberto do MQTT, amplamente utilizada por sua robustez, baixo consumo de recursos e facilidade de integração com sistemas de diferentes escalas. Como broker, o Mosquitto é responsável por gerenciar as mensagens que são publicadas pelos dispositivos, distribuindo-as para os clientes que estão inscritos nos tópicos correspondentes.

A escolha do Mosquitto como broker foi motivada por suas características que atendem diretamente às necessidades do projeto. Um dos principais fatores foi seu desempenho e escalabilidade, já que o Mosquitto é projetado para suportar uma grande quantidade de conexões simultâneas, o que o torna ideal para sistemas que envolvem diversos dispositivos IoT, como no caso desta plataforma. Além disso, sua facilidade de integração, com compatibilidade com diversas linguagens de programação e plataformas, simplificou a comunicação entre o backend e os dispositivos conectados. Outro aspecto fundamental foi o suporte oferecido pelo Mosquitto para autenticação e criptografia TLS/SSL, o que garantiu a segurança das comunicações entre os dispositivos e o broker. Essas características tornaram o Mosquitto uma escolha confiável e eficiente para gerenciar o fluxo de mensagens no projeto, atendendo tanto aos requisitos técnicos quanto às demandas de segurança.

3.3.1 Arquitetura de Comunicação

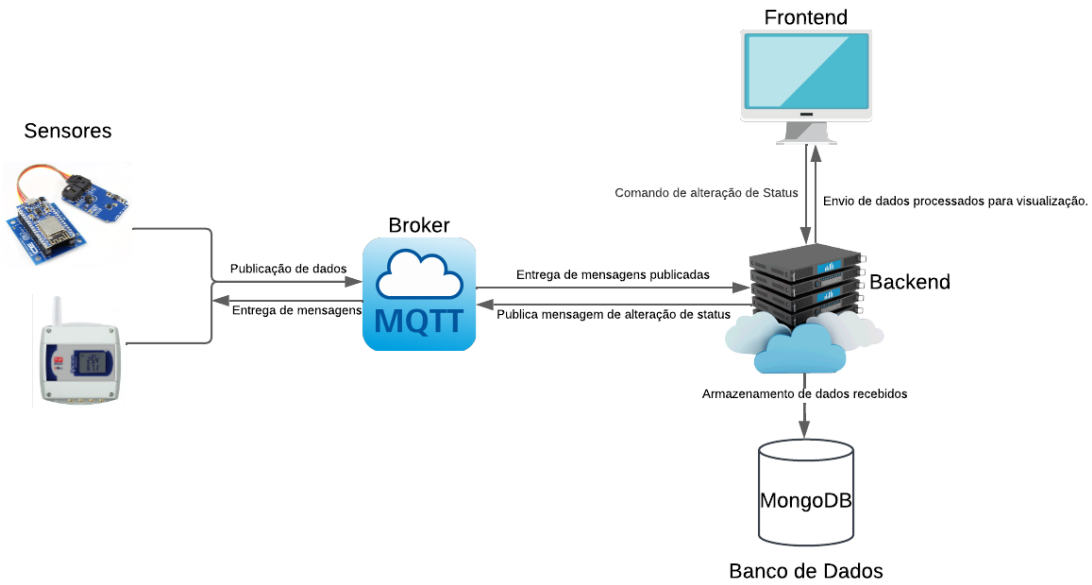
A arquitetura de comunicação da plataforma foi desenvolvida com base no modelo publicador/assinante, que é uma das principais características do protocolo MQTT. Nesse modelo, os dispositivos IoT publicam seus dados de telemetria, como temperatura e umidade, em tópicos específicos. Por exemplo, cada aparelho possui tópicos exclusivos, como "dispositivo/id/dados", onde as informações são transmitidas. Essas mensagens publicadas pelos dispositivos são recebidas pelo broker, que irá distribuir as informações para os clientes que estão inscritos nesses tópicos. Essa abordagem garante que os dados sejam entregues mesmo em casos onde o cliente está temporariamente offline, já que o broker pode armazenar as mensagens e enviá-las assim que o cliente se reconectar, dependendo do nível de QoS configurado.

O backend da plataforma, implementado com Django, opera como um cliente MQTT

inscrito nos tópicos relevantes, recebendo as informações publicadas pelos dispositivos em tempo real. Esses dados são processados e armazenados no MongoDB para análise e visualização no painel da plataforma. A comunicação também inclui o envio de mensagens do backend para os dispositivos, em casos onde comandos precisam ser emitidos, como o acionamento de dispositivos ou a alteração de estados. Nesse caso, o backend publica mensagens em tópicos específicos monitorados pelos dispositivos, como "dispositivo/id/status", permitindo uma comunicação bidirecional eficiente e segura.

Essa arquitetura foi projetada para ser simples, escalável e eficiente, atendendo aos requisitos de um sistema IoT em um Smart Campus. A escolha do modelo publish/-subscribe, aliada ao uso do Mosquitto como broker, garantiu uma integração robusta entre os dispositivos e a plataforma, possibilitando a troca de informações em tempo real com alta confiabilidade. Dessa forma, a arquitetura de comunicação, representada na Figura 4, desempenhou um papel central no sucesso do projeto, fornecendo uma base sólida para a operação e expansão futura da plataforma.

Figura 4 – Diagrama da Arquitetura de Comunicação



Fonte: O autor

3.4 TESTES E VALIDAÇÃO

A validação da plataforma foi realizada utilizando dispositivos reais em um ambiente simulado, com foco em verificar a funcionalidade de integração e comunicação. O dispositivo utilizado foi um sensor de corrente conectado ao sistema por meio do protocolo MQTT. Durante os testes, o sensor publicava dados em tópicos específicos, que eram processados pela plataforma e exibidos no painel de monitoramento.

A métrica principal considerada foi a verificação da entrega e tratamento correto das mensagens, garantindo que os dados enviados pelo sensor fossem recebidos e interpretados corretamente pelo backend. Dessa forma, o objetivo foi assegurar a funcionalidade básica da integração entre os dispositivos e a plataforma.

Quanto aos testes de desempenho, não foram realizados devido ao foco deste trabalho ser a validação funcional. No entanto, a realização de testes de carga e desempenho foi apontada como um objetivo para trabalhos futuros, visando avaliar a escalabilidade e a robustez da plataforma em cenários de uso real com um número maior de dispositivos conectados.

3.4.1 Testes Funcionais do Painel

Os testes funcionais do painel foram realizados para verificar o correto funcionamento das interfaces e das ações disponíveis para o usuário. Durante os testes, foram simulados diferentes cenários de uso, contemplando as principais funcionalidades do sistema:

(a) Sistemas de cadastro, login e convite para projetos:

Validação das funcionalidades de cadastro e login na plataforma, bem como a criação de novos projetos e o envio de convite por email para membros do projeto.

(b) Cadastro e configuração de dispositivos IoT:

Adição de novos dispositivos à plataforma, associando informações como categoria, localização, informações a serem enviadas à plataforma, por quais tópicos e o tipo

de exibição no dashboard.

(c) Dashboard com as métricas de dispositivos

Exibição dos dados enviados pelos dispositivos IoT no painel, atualizados dinamicamente por meio da integração com o Mosquitto.

(d) Controle de dispositivos:

Envio de comandos para dispositivos, como ligar/desligar atuadores, por meio do painel de controle.

Cada funcionalidade foi avaliada com base nos critérios de aceitação definidos na análise de requisitos. Todos os cenários obtiveram sucesso, confirmando a usabilidade e eficiência do painel.

3.4.2 Testes de Integração com Dispositivos IoT

A integração dos dispositivos IoT com a plataforma foi testada por meio de simulações utilizando dispositivos reais e emuladores. O objetivo foi validar a comunicação entre os dispositivos e o broker MQTT (Mosquitto), bem como o processamento dos dados pelo backend da plataforma.

Procedimentos realizados:

1. Publicação de mensagens por dispositivos nos tópicos MQTT correspondentes, com monitoramento pelo backend.

2. Subscrição do backend nos tópicos indicados durante o cadastro de cada dispositivo para receber atualizações em tempo real.

3. Testes de comandos de alteração de status enviados pelo painel para dispositivos, verificando sua execução.

Os resultados confirmaram a confiabilidade da integração e a compatibilidade dos dispositivos com a plataforma.

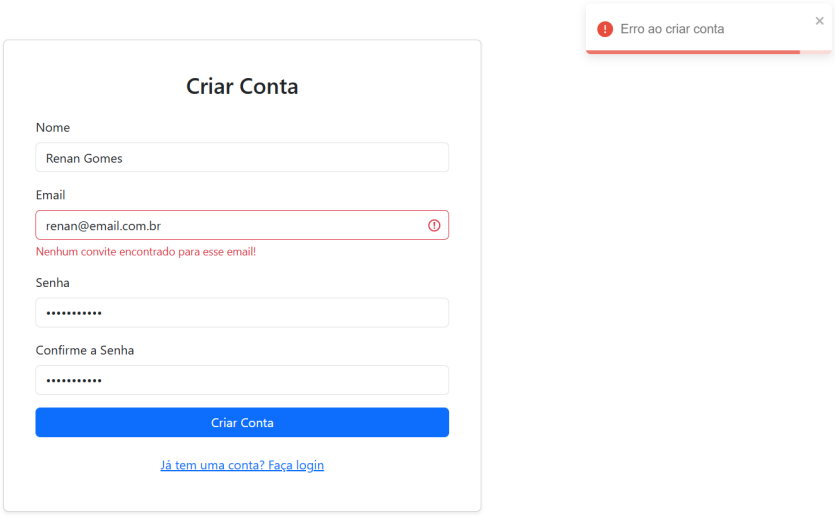
4 RESULTADOS

Os resultados obtidos a partir do desenvolvimento da plataforma de gestão integrada de dispositivos IoT para o Smart Campus demonstram a funcionalidade, a usabilidade e a eficiência do sistema em um ambiente simulado. A seguir, são apresentados os principais componentes da solução, com capturas de tela ilustrando o funcionamento da interface e as interações com os dispositivos IoT.

4.1 TELAS DE CADASTRO E LOGIN

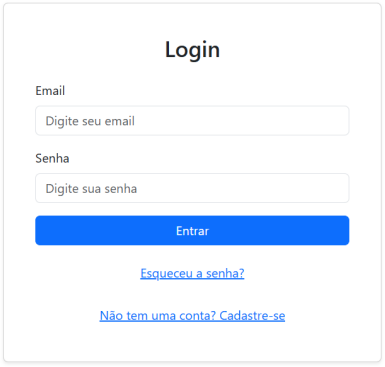
O sistema conta com um mecanismo de autenticação que permite o acesso apenas para usuários previamente convidados. A tela inicial oferece as opções de cadastro e login (figura 6). Durante o cadastro, o sistema verifica se o usuário possui um convite pendente, garantindo controle sobre quem pode acessar a plataforma, conforme exibido na figura 5.

Figura 5 – Tela de cadastro de usuário com validação de convite.



Fonte: O autor

Figura 6 – Tela de Login



Fonte: O autor

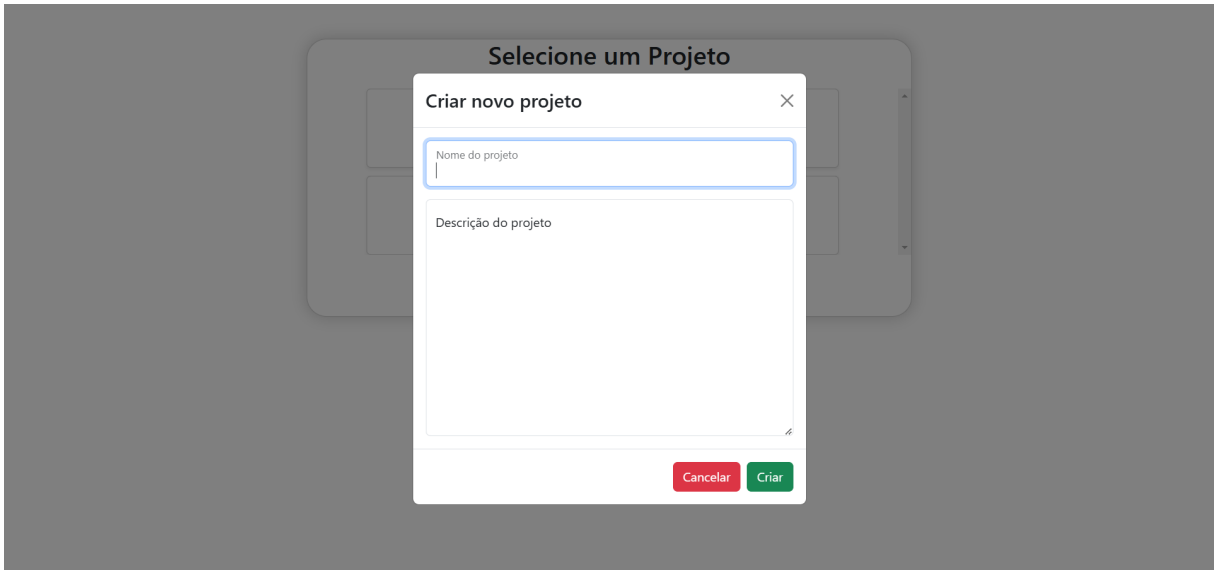
4.2 SELEÇÃO E CADASTRO DE PROJETOS

Após o login, o usuário é direcionado para a tela de seleção de projeto (Figura 7). Caso o usuário seja membro de vários projetos, ele pode escolher qual deseja acessar no momento. Além disso, é possível criar novos projetos diretamente por essa interface (Figura 8).

Figura 7 – Tela de seleção de projetos.

Fonte: O autor

Figura 8 – Tela de criação de um novo projeto.

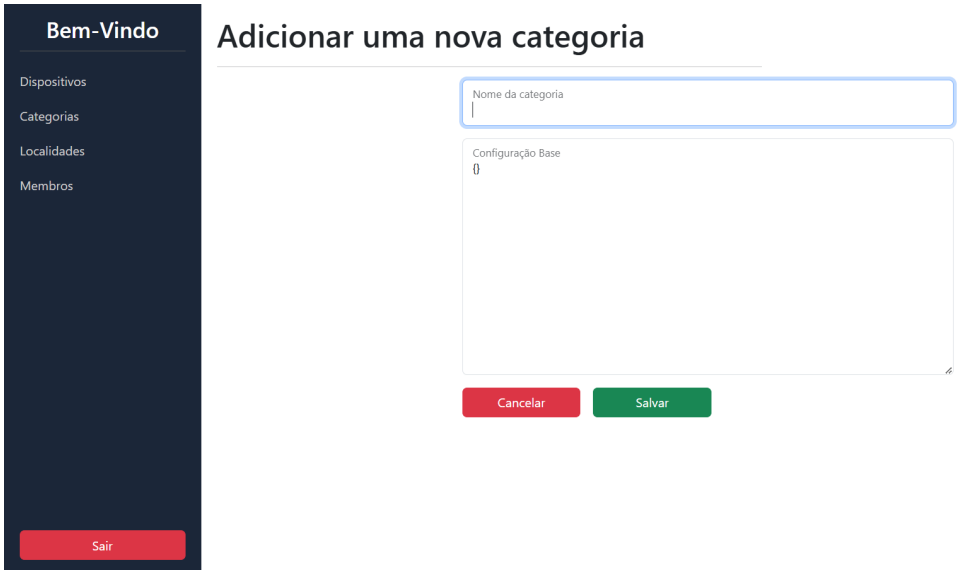


Fonte: O autor

4.3 CRUD DE CATEGORIAS

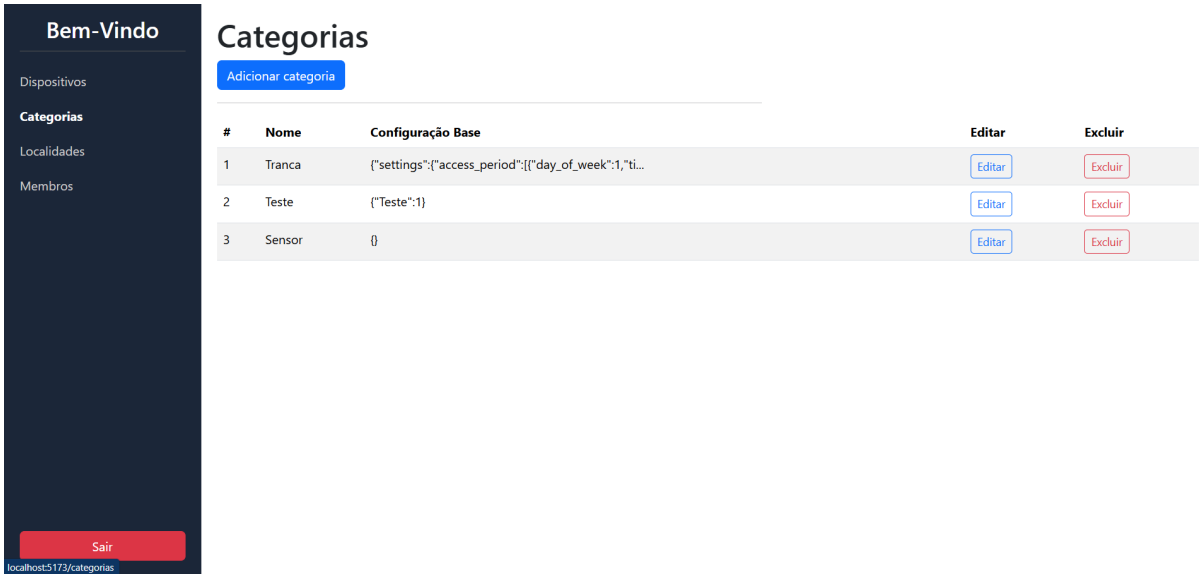
A funcionalidade de cadastro de categorias permite ao usuário definir tipos de dispositivos, como "sensor"ou "tranca", e associar uma configuração base em formato JSON. Essa configuração base é utilizada para facilitar o cadastro de dispositivos, pois é pré-carregada quando uma categoria específica é selecionada (Conforme será demonstrado na Figura 14).

Figura 9 – Tela de cadastro de categorias.



Fonte: O autor

Figura 10 – Tela de listagem de categorias com opções de edição e exclusão.

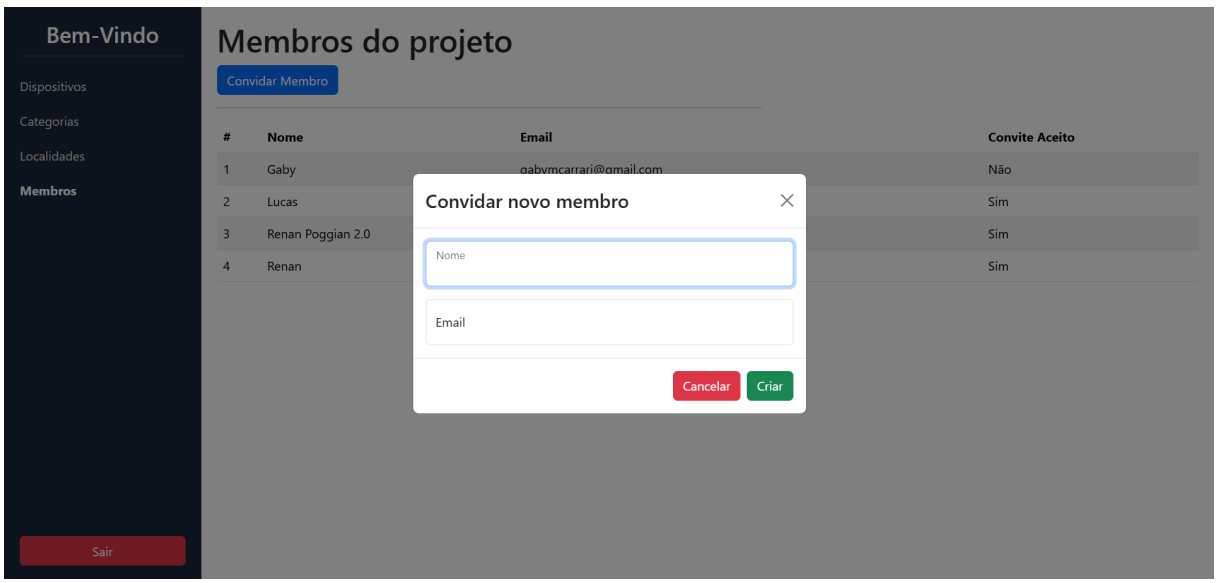


Fonte: O autor

4.4 TELA DE CONVITE DE MEMBROS

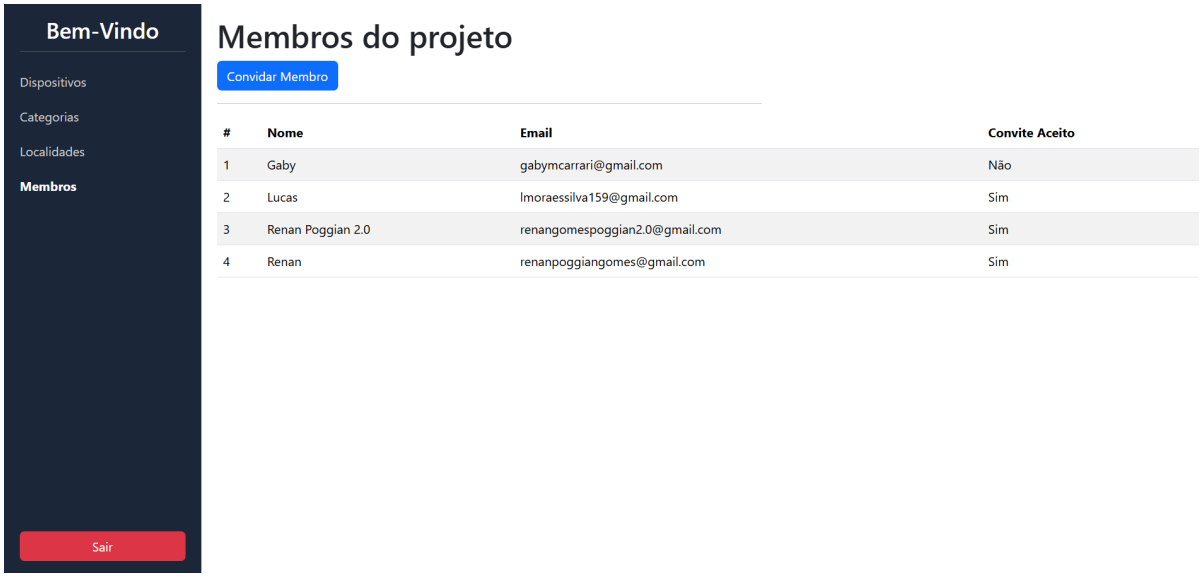
Os usuários membros de um projeto podem convidar novos membros a partir do menu "Membros". A interface permite que o convite seja enviado para um endereço de e-mail, com a validação posterior do cadastro.

Figura 11 – Tela de convite de membros para um projeto.



Fonte: O autor

Figura 12 – Tela de listagem de membros e convidados.



Fonte: O autor

4.5 CADASTRO E CONFIGURAÇÃO DE DISPOSITIVOS

O sistema oferece funcionalidades completas para o gerenciamento de dispositivos, incluindo cadastro, listagem, edição e exclusão. O cadastro de dispositivos na plataforma foi estruturado em um fluxo de quatro etapas, demonstrado nas figuras, projetado para equilibrar simplicidade e flexibilidade, permitindo que o usuário configure dispositivos de acordo com suas necessidades específicas:

(a) Etapa 1 - Identificação e Classificação (Figura 13)

O usuário insere o nome do dispositivo, define sua localização e escolhe uma categoria predefinida.

(b) Etapa 2 - Configuração (Figura 14)

Baseado na categoria selecionada, um arquivo JSON pré-carregado é disponibilizado com configurações padrão. O usuário pode optar por mantê-lo como está ou editá-lo conforme necessário, ajustando as configurações para atender requisitos específicos do dispositivo.

(c) Etapa 3 - Controle de Estado (Figura 15)

Caso o dispositivo opere com controle de estados, nesta etapa o usuário define:

- O nome do estado.
- O tipo de estado.
- Os valores associados ao estado

(d) Etapa 4 - Definição de Métricas (Figura 16)

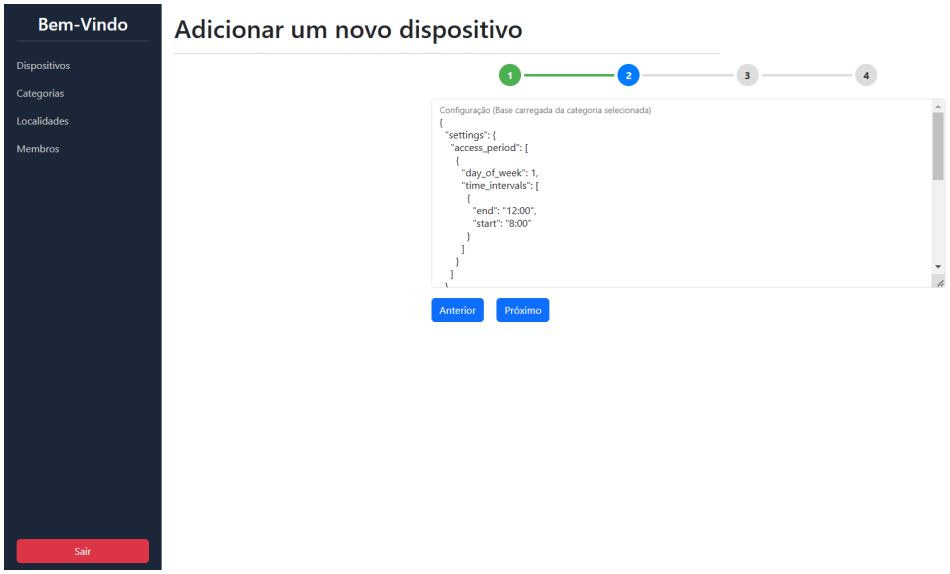
O usuário especifica a quantidade de campos que o dispositivo reportará à plataforma. Para cada campo, ele define:

- O nome.
- O tipo de dado.
- O tópico MQTT pelo qual o dado será enviado.
- O tipo de visualização no painel.

Figura 13 – Tela de cadastro de dispositivo - Etapa 1.

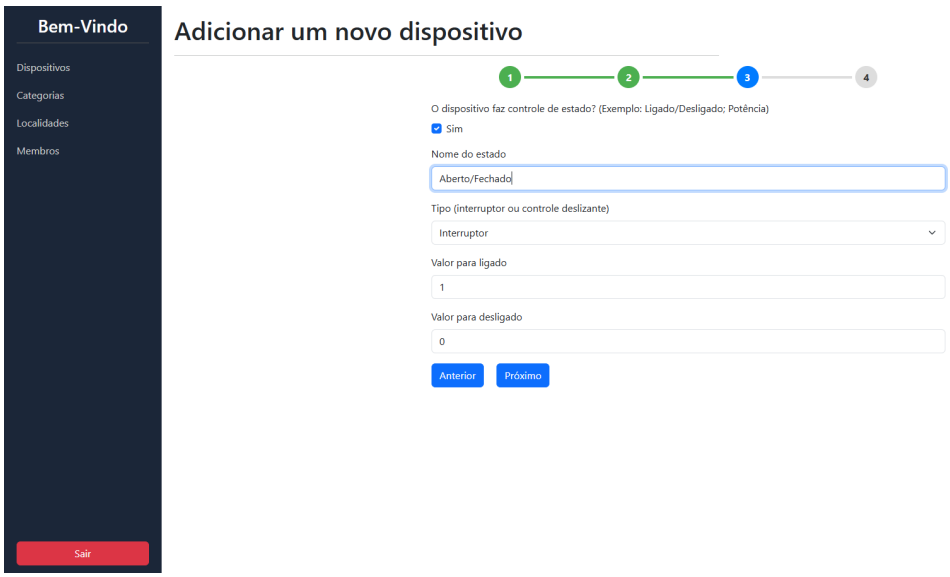
Fonte: O autor

Figura 14 – Tela de cadastro de dispositivo - Etapa 2.



Fonte: O autor

Figura 15 – Tela de cadastro de dispositivo - Etapa 3.



Fonte: O autor

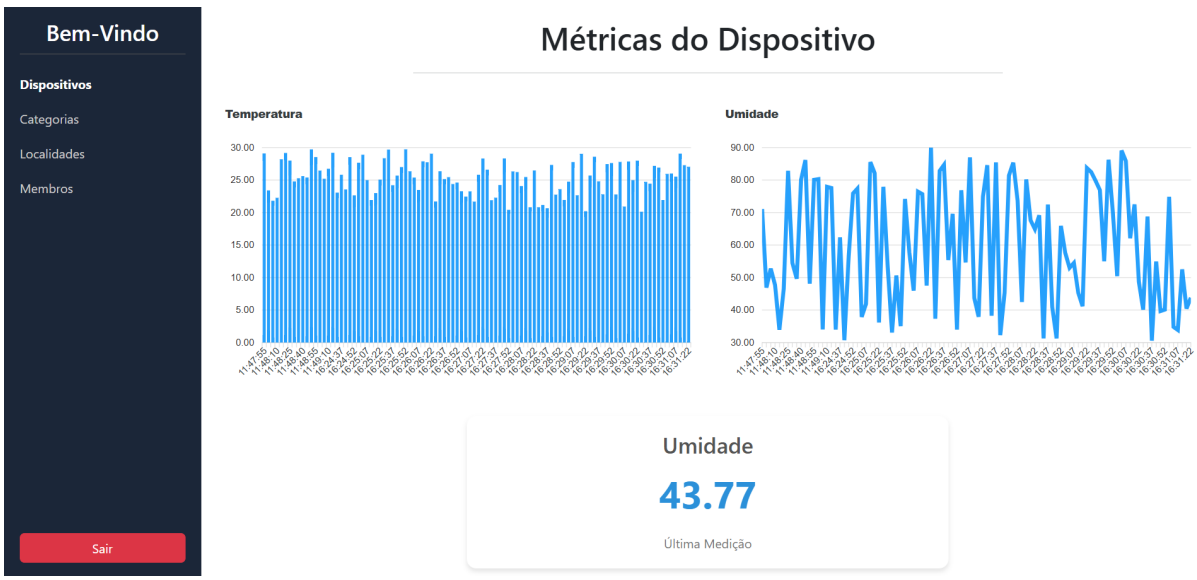
Figura 16 – Tela de cadastro de dispositivo - Etapa 4.

Fonte: O autor

4.6 MONITORAMENTO E VISUALIZAÇÃO DE MÉTRICAS

Os dispositivos configurados são integrados à plataforma, que oferece recursos de monitoramento, exibindo as métricas em tempo real conforme definido na etapa de configuração. Cada métrica pode ser apresentada de acordo com o tipo de visualização selecionado. A flexibilidade da plataforma garante que tanto dispositivos simples quanto complexos possam ser monitorados de forma eficiente, suportando diferentes cenários de uso e necessidades do usuário.

Figura 17 – Tela de dashboard com métricas de um dispositivo.



Fonte: O autor

5 CONCLUSÃO

Este trabalho apresentou o desenvolvimento de um sistema completo para o gerenciamento de dispositivos IoT, abrangendo funcionalidades como cadastro, configuração, monitoramento e visualização de métricas. O sistema foi projetado para equilibrar simplicidade de uso com flexibilidade de personalização, oferecendo uma plataforma robusta para atender às necessidades específicas de diferentes dispositivos e usuários.

Ao longo do desenvolvimento, foram priorizados aspectos como a criação de um fluxo intuitivo para o cadastro de dispositivos, permitindo que o usuário defina parâmetros customizados, como estados, tópicos de comunicação e tipos de visualização. Adicionalmente, foi implementado um sistema de autenticação e controle de membros que garante maior segurança e flexibilidade no gerenciamento de equipes e projetos.

5.1 LIMITAÇÕES ENCONTRADAS

Embora o sistema ofereça uma base para o gerenciamento de dispositivos, algumas limitações foram identificadas durante seu desenvolvimento. Primeiramente, testes de desempenho e segurança mais aprofundados não foram realizados, o que pode limitar a escalabilidade do sistema em ambientes com alta demanda. Além disso, a integração com protocolos de comunicação IoT mais avançados, como LoRaWAN ou ZigBee, ainda não foi implementada, restringindo o uso do sistema a dispositivos compatíveis com WiFi.

5.2 TRABALHOS FUTUROS

Com base nas limitações identificadas durante o desenvolvimento deste projeto, há diversas oportunidades para aprimoramentos e expansões futuras. Um dos principais pontos é a necessidade de realizar testes mais profundos de desempenho e de segurança da plataforma, pois apesar de ter sido pensado em escalabilidade e segurança durante o desenvolvimento, o foco nos testes e validação para este trabalho foi nas funcionalidades da plataforma e na integração eficaz dos dispositivos. Para a escalabilidade, seria interessante realizar testes de carga que avaliem o comportamento

do sistema em cenários com um grande número de dispositivos conectados. Esses testes poderiam indicar a necessidade de implementar melhorias como balanceamento de carga ou escalabilidade horizontal, garantindo um funcionamento mais robusto em ambientes de alta demanda.

Já na área de segurança, um ponto de melhoria seria a integração de métodos de autenticação multifatorial e encriptação ponta a ponta na comunicação entre dispositivos e o servidor, o que poderia oferecer maior proteção contra acessos não autorizados e ataques cibernéticos, reforçando a confiabilidade da solução. Além disso, a integração com protocolos de comunicação adicionais, como LoRaWAN e ZigBee, ampliaria a versatilidade do sistema, permitindo a conexão com uma gama ainda maior de dispositivos IoT e tornando a plataforma mais adaptável a diferentes cenários.

Outra área de grande potencial é a aplicação de algoritmos de aprendizado de máquina para a análise de dados enviados. Esses algoritmos poderiam identificar padrões ou anomalias nos dados de telemetria, facilitando a tomada de decisão e permitindo a geração de alertas em tempo real. Além disso, a inclusão de funcionalidades para geração de relatórios históricos com base nos dados armazenados também seria uma melhoria significativa, oferecendo aos usuários ferramentas avançadas para análise e visualização de informações relevantes ao longo do tempo.

Por fim, um estudo de caso em uma instituição de ensino que esteja implementando soluções de IoT poderia ser conduzido para demonstrar o impacto prático da plataforma desenvolvida. Embora o projeto já tenha sido validado com dispositivos reais, porém em um ambiente simulado, aplicar a solução em um cenário real permitiria avaliar seu desempenho sob condições operacionais autênticas e destacar seus benefícios em um Smart Campus. Além disso, um estudo de caso traria a oportunidade de identificar possíveis ajustes que possam tornar a plataforma ainda mais eficiente e alinhada às demandas específicas de uma instituição de ensino. Dessa forma, o estudo serviria para consolidar o trabalho desenvolvido e reforçar sua aplicabilidade no contexto de gestão integrada de dispositivos IoT.

5.3 CONSIDERAÇÕES FINAIS

O trabalho desenvolvido demonstra a viabilidade de um sistema robusto e flexível para o gerenciamento de dispositivos IoT, oferecendo uma base sólida para futuras melhorias e expansões. Apesar das limitações, o sistema cumpre seu objetivo de fornecer aos usuários uma ferramenta prática e funcional para configurar e monitorar seus dispositivos.

Com a continuidade das melhorias sugeridas, este projeto tem o potencial de se tornar uma solução amplamente aplicável em qualquer Smart Campus. O trabalho realizado marca um importante passo inicial nessa direção, pavimentando o caminho para avanços tecnológicos e novas oportunidades de pesquisa.

REFERÊNCIAS

ABDEL-BASSET, M. et al. Internet of things in smart education environment: Supportive framework in the decision-making process. *Concurrency and Computation: Practice and Experience*, v. 31, p. e4515, 05 2018.

ABOUDOUMAT, E. et al. Performance comparison between sql and nosql in terms of use with big data. *International Science and Technology Journal*, v. 34, p. 1–19, 04 2024.

ABUARQOUB, A. et al. A survey on internet of things enabled smart campus applications. In: *Proceedings of the International Conference on Future Networks and Distributed Systems*. New York, NY, USA: Association for Computing Machinery, 2017, (ICFNDS '17). ISBN 9781450348447. Disponível em: <https://doi.org/10.1145/3102304.3109810>.

AIELLO, M. Iot architectures: from data to smart systems. *Frontiers in the Internet of Things*, v. 1, 2022. ISSN 2813-3110. Disponível em: <https://www.frontiersin.org/articles/10.3389/friot.2022.959268>.

AL-FUQAHA, A. et al. Internet of things: A survey on enabling technologies, protocols and applications. In: *IEEE Communications Surveys & Tutorials*. [S.l.: s.n.], 2015. v. 17, p. Fourthquarter 2015.

ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. In: *Computer Networks*. [S.l.: s.n.], 2010. p. 2787–2805.

BORGIA, E. The internet of things vision: Key features, applications and open issues. In: *Computer Communications*. [S.l.: s.n.], 2014. v. 54.

CAVUS, N. et al. Internet of things and its applications to smart campus: A systematic literature review. In: *International Journal of Interactive Mobile Technologies (iJIM)*. [s.n.], 2022. v. 16, n. 23, p. pp. 17–35. Disponível em: <https://online-journals.org/index.php/i-jim/article/view/36215>.

DESHMUKH, R. A. et al. Data spine: A federated interoperability enabler for heterogeneous iot platform ecosystems. *Sensors*, v. 21, n. 12, 2021. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/21/12/4010>.

DETTI, A.; FUNARI, L.; BLEFARI-MELAZZI, N. Sub-linear scalability of mqtt clusters in topic-based publish-subscribe applications. *IEEE Transactions on Network and Service Management*, v. 17, n. 3, p. 1954–1968, 2020.

DOMíNGUEZ-BOLAñO, T. et al. An iot system for a smart campus: Challenges and solutions illustrated over several real-world use cases. *Internet of Things*, v. 25, p. 101099, 2024. ISSN 2542-6605. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2542660524000416>.

DONG, Z. Y. et al. Smart campus: definition, framework, technologies, and services. In: *IET Smart Cities*. [s.n.], 2020. v. 2, n. 1, p. 43–54. Disponível em: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-smc.2019.0072>.

FERREIRA, D. et al. Planning and optimization of software-defined and virtualized iot gateway deployment for smart campuses. In: . [s.n.], 2022. v. 22, n. 13. Disponível em: <https://www.mdpi.com/1424-8220/22/13/4710>.

GUBBI, J. et al. Internet of things (iot): A vision, architectural elements, and future directions. In: *Future Generation Computer Systems*. [S.l.: s.n.], 2013. v. 29, n. 7, p. 1645–1660.

JAVED, A. et al. Scalable iot platform for heterogeneous devices in smart environments. In: *IEEE Access*. [S.l.: s.n.], 2020. v. 8, p. 211973–211985.

KEGENBEKOV, Z.; SAPAROVA, A. Using the mqtt protocol to transmit vehicle telemetry data. In: . [S.l.: s.n.], 2022. v. 61. ISSN 23521465.

KHAN, R. et al. Future internet: The internet of things architecture, possible applications and key challenges. In: . [S.l.: s.n.], 2012. p. 257–260. ISBN 978-1-4673-4946-8.

KHINE, P. P.; WANG, Z. A review of polyglot persistence in the big data world. *Inf.*, v. 10, p. 141, 2019. Disponível em: <https://api.semanticscholar.org/CorpusID:146058780>.

KRčO, S.; POKRIć, B.; CARREZ, F. Designing iot architecture(s): A european perspective. In: *2014 IEEE World Forum on Internet of Things (WF-IoT)*. [S.l.: s.n.], 2014. p. 79–84.

MAKRIS, A. et al. Mongodb vs postgresql: A comparative study on performance aspects. *GeoInformatica*, p. 25, 04 2021.

MILEVA, A. et al. Comprehensive analysis of mqtt 5.0 susceptibility to network covert channels. *Computers & Security*, v. 104, p. 102207, 2021. ISSN 0167-4048. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167404821000316>.

MUHAMAD, W. et al. Smart campus features, technologies, and applications: A systematic literature review. In: *2017 International Conference on Information Technology Systems and Innovation (ICITSI)*. [S.l.: s.n.], 2017. p. 384–391.

NAVANI, D.; JAIN, S.; NEHRA, M. The internet of things (iot): A study of architectural elements. In: . [S.l.: s.n.], 2017. p. 473–478.

POLIN, K. et al. The making of smart campus: A review and conceptual framework. *Buildings*, v. 13, n. 4, 2023. ISSN 2075-5309. Disponível em: <https://www.mdpi.com/2075-5309/13/4/891>.

PRADA, M. A. et al. Communication with resource-constrained devices through mqtt for control education. In: . [S.l.: s.n.], 2016. v. 49. ISSN 24058963.

RAMESH, P.; REDDY, S. C. R.; REDDY, D. P. B. Architecture, protocols, layers and elements of iot. In: . [s.n.], 2021. Disponível em: <https://api.semanticscholar.org/CorpusID:243831664>.

SONI, D.; MAKWANA, A. A survey on mqtt: A protocol of internet of things(iot). In: . [S.l.: s.n.], 2017.

VALKS MONIQUE H. ARKESTEIJN, A. K. B.; HEIJER, A. C. den. Towards a smart campus: supporting campus decisions with internet of things applications. In: . Routledge, 2021. v. 49, n. 1, p. 1–20. Disponível em: <https://doi.org/10.1080/09613218.2020.1784702>.

VELINOV, A. et al. Covert channels in the mqtt-based internet of things. *IEEE Access*, v. 7, p. 161899–161915, 2019.

WU, M. et al. Research on the architecture of internet of things. In: *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*. [S.l.]: IEEE, 2010. v. 5, p. V5–484. ISBN 1424465397.