Pós-Graduação em Ciência da Computação

Gabriel Vanderlei de Oliveira

EDGEWIDGETS: PLATAFORMA IOT DUAL-PROTOCOL PARA MONITORAMENTO AMBIENTAL COM CAPACIDADES DE EDGE COMPUTING

Dissertação de Mestrado

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE
2025

Universidade Federal de Pernambuco
Centro de Informática
Pós-graduação em Ciência da Computação

Gabriel Vanderlei de Oliveira

EDGEWIDGETS: PLATAFORMA IOT DUAL-PROTOCOL PARA MONITORAMENTO AMBIENTAL COM CAPACIDADES DE EDGE COMPUTING

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: *Prof. Dr. Jamilson Ramalho Dantas*
Co-Orientador: *Prof. Dr. Gilmar Gonçalves de Brito*

RECIFE
2025

.Catalogação de Publicação na Fonte. UFPE - Biblioteca Central

Oliveira, Gabriel Vanderlei de.
Edgewidgets: plataforma IOT dual-protocol para monitoramento ambiental com capacidades de Edge Computing / Gabriel Vanderlei de Oliveira. - Recife, 2025.
84f.: il.

Dissertação (Mestrado)- Universidade Federal de Pernambuco, Centro de Informática, Programa de Pós-Graduação em Ciência da Computação, 2025.
Orientação: Jamilson Ramalho Dantas.
Coorientação: Gilmar Gonçalvez de Brito.

1. Internet das coisas; 2. EdgeWidgets; 3. Protocolos de comunicação; 4. MQTT; 5. HTTP; 6. Edge Computing. I. Dantas, Jamilson Ramalho. II. Brito, Gilmar Gonçalvez de. III. Título.

UFPE-Biblioteca Central

Gabriel Vanderlei de Oliveira

“EdgeWidgets: Plataforma IoT Dual-Protocol para Monitoramento Ambiental com Capacidades de Edge Computing”

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação. Área de Concentração: Redes de Computadores e Sistemas Distribuídos.

Aprovado em: 29/07/2025.

BANCA EXAMINADORA

_____

Prof. Dr. Nelson Souto Rosa
Centro de Informática / UFPE

_____

Prof. Dr. Jean Carlos Teixeira de Araújo
Departamento de Ciência Computação /UFAPE

_____

Prof. Dr. Jamilson Ramalho Dantas
Centro de Informática / UFPE
(**orientador**)

Dedico este trabalho às pessoas que foram pilares fundamentais durante minha trajetória no mestrado.

Agradecimentos

Ao meu orientador, Prof. Dr. Jamilson Ramalho Dantas, por sua orientação precisa e apoio constante durante todo o desenvolvimento deste trabalho. Sua expertise e visão foram fundamentais para o sucesso desta pesquisa.

Ao meu co-orientador, Prof. Dr. Gilmar Gonçalves de Brito, agradeço por todo o conhecimento e ensinamentos. Seus projetos foram muito importantes para o desenvolvimento deste trabalho.

À minha mãe e à minha tia, Neusa e Ana, minha eterna gratidão. O carinho, a sabedoria e o apoio incondicional de vocês me acompanharam desde os primeiros passos e continuam sendo a base de tudo que construo. Cada conquista é, também, de vocês.

A Amanda, minha noiva e parceira de vida, agradeço por estar ao meu lado em todos os momentos. Seu amor me fortaleceu, sua paciência me acolheu e sua presença me motivou a seguir, mesmo nos dias mais difíceis. Nada disso seria o mesmo sem você.

Aos mestres que cruzaram meu caminho acadêmico, muito obrigado por abrirem portas para o conhecimento e me incentivarem a ir além. Cada ensinamento deixou marcas importantes nesta jornada.

Aos amigos que compartilham comigo essa caminhada, agradeço pelas conversas, pelas risadas e pelo apoio sincero. Ter vocês por perto tornou tudo mais leve e significativo.

E, finalmente, agradeço a mim mesmo. Por ter persistido, por ter acreditado, por não ter recuado diante dos obstáculos. Este trabalho representa não apenas uma etapa acadêmica concluída, mas também uma prova pessoal de superação e crescimento.

Você não pode conectar os pontos olhando para frente;
você só pode conectá-los olhando para trás.
—STEVE JOBS

Resumo

Tecnologias baseadas em Internet das Coisas estão se tornando cada vez mais acessíveis, enquanto a necessidade de monitoramento ambiental constante aumenta, ampliando a gama de soluções necessárias para o apoio decisório urbano e industrial. Visando esse contexto, esta dissertação aborda o desenvolvimento da plataforma EdgeWidgets, envolvendo as motivações, arquitetura, estruturas de testes, validação e metodologia aplicada. A plataforma implementa recursos comuns para aplicações de gerenciamento IoT, como listagem de dispositivos, widgets e dashboards, com atenção especial aos aspectos de escalabilidade e configurações de infraestrutura. O foco da utilização concentra-se em um cenário de medição de estabilidade de encostas, replicando sensores equivalentes aos encontrados neste ambiente e simulando a comunicação pela rede. O estudo investiga possíveis estratégias de controle de risco utilizando a própria infraestrutura de rede como base. Foram elaborados dois estudos de caso utilizando-se dos processos relacionados tanto ao nível ambiental quanto à rede. O primeiro estudo apresenta questões vinculadas a deslizamentos, seus principais fatores e como as funções de rede 5G podem auxiliar na análise de riscos de eventos climáticos. O segundo estudo de caso foca na utilização da plataforma para receber informações de sensores de inclinação, analisando o desempenho de métricas relevantes como latência e delay em protocolos distintos (MQTT e HTTP). A investigação contribui com soluções que interagem com formatos já existentes de monitoramento ambiental, além de sugestões e análises sobre possíveis novas adoções no campo de estudo.

Palavras-chave: Internet das Coisas, EdgeWidgets, Protocolos de Comunicação, MQTT, HTTP, Edge Computing, Monitoramento Ambiental, Deslizamentos, Sensoriamento de Inclinação, Sistemas Distribuídos, RSSI, Redes 5G

Abstract

Internet of Things-based technologies are becoming increasingly accessible, while the need for constant environmental monitoring increases, expanding the range of solutions required for urban and industrial decision-making. In this context, this dissertation addresses the development of the EdgeWidgets platform, encompassing its motivations, architecture, testing frameworks, validation, and applied methodology. The platform implements common features for IoT management applications, such as device listing, widgets, and dashboards, with special attention to scalability aspects and infrastructure configurations. The usage focus concentrates on a slope stability measurement scenario, replicating sensors equivalent to those found in this environment and simulating network communication. The study investigates possible risk control strategies using the network infrastructure itself as a foundation. Two case studies were developed using processes related to both environmental and network levels. The first study presents issues linked to landslides, their main factors, and how 5G network functions can assist in climate event risk analysis. The second case study focuses on using the platform to receive information from inclination sensors, analyzing the performance of relevant metrics such as latency and delay in different protocols (MQTT and HTTP). The investigation contributes solutions that interact with existing environmental monitoring formats, in addition to suggestions and analyses regarding possible new adoptions in the field of study.

Keywords: Internet of Things, EdgeWidgets, Communication Protocols, MQTT, HTTP, Edge Computing, Environmental Monitoring, Landslides, Inclination Sensing, Distributed Systems, RSSI, 5G Networks

Lista de Figuras

Lista de Tabelas

List of Algorithms

Lista de Acrônimos

RSVP *Resource Reservation Protocol* 37

SaaS *Software as a Service* 36

SHALSTAB *Shallow Landsliding Stability Model* 30

SDN *Software-Defined Networks* 17

SNR *Signal-to-Noise Ratio* 17

SMTRAGE Sistema de Monitoramento em Tempo Real para Aplicação Geotécnica em Encostas 31

TCP *Transmission Control Protocol* 33

UDP *User Datagram Protocol* 33

Sumário

1 Introdução

1.1 Contexto

Os deslizamentos de encostas representam um dos fenômenos naturais mais devastadores globalmente, constituindo um grave problema que afeta milhões de pessoas anualmente e causa perdas econômicas e sociais significativas. Segundo análises recentes do Fórum Econômico Mundial, eventos climáticos extremos e desastres relacionados ao clima resultaram em perdas econômicas de aproximadamente 1,5 trilhão de dólares na década até 2019, com deslizamentos entre os eventos mais devastadores (World Economic Forum, 2023). Além das consequências econômicas, esses eventos têm profundas implicações sociais, incluindo perda de vidas humanas, deslocamento populacional e destruição de comunidades inteiras.

No Brasil, a situação é particularmente crítica, com cerca de 1.942 municípios afetados por riscos de deslizamentos (Brasil. Ministério do Desenvolvimento Regional, 2023). Esses eventos não apenas ameaçam vidas, mas também comprometem a estabilidade social e econômica de regiões inteiras. A região metropolitana do Recife exemplifica essa vulnerabilidade, onde a combinação de urbanização densa, assentamentos irregulares e chuvas tropicais intensas coloca milhares de vidas em risco anualmente durante a estação chuvosa (BANDEIRA; COUTINHO, 2015). Os eventos de abril-maio de 2024 no Rio Grande do Sul, caracterizados como um desastre hidrológico excepcional que resultou em dezenas de mortes e centenas de desabrigados devido a inundações e deslizamentos provocados por chuvas intensas, destacam a urgência de soluções de monitoramento eficazes e a necessidade de integração de tecnologias avançadas de sensoriamento remoto para mapeamento em tempo real de áreas afetadas (COLLISCHONN et al., 2024). O monitoramento constitui uma ferramenta essencial para auxiliar na tomada de decisões preventivas e corretivas, sendo fundamental não apenas para deslizamentos, mas também para outros riscos urbanos e ambientais, como monitoramento de barragens e gestão de resíduos de mineração.

Na análise de estabilidade de encostas, diferentes tipos de sensores desempenham papéis complementares cruciais. A medição da umidade do solo destaca-se como um dos mecanismos de sensoreamento mais utilizados na gestão de riscos referentes a deslizamentos de encostas, sendo

crucial para identificar o aumento da umidade, um dos principais fatores de risco, geralmente impulsionado por chuvas intensas (INTRIERI et al., 2012). Os inclinômetros fornecem dados precisos sobre movimentação e deformação do solo, enquanto os pluviômetros monitoram a precipitação, permitindo correlacionar eventos de chuva com mudanças na estabilidade das encostas. Juntos, esses dispositivos fornecem dados vitais para avaliar de forma abrangente a estabilidade das encostas.

A integração de tecnologias de *Internet of Things* (IoT) emerge como uma abordagem promissora para enfrentar esses desafios. A combinação desses sensores, integrados a tecnologias relativas ao conceito de IoT, visualização 3D e Indústria 4.0, possibilita a geração de alertas em tempo real, melhorando significativamente a capacidade de auxiliar a aplicação de medidas corretivas ou preventivas de forma mais eficaz. Sistemas de monitoramento tradicionais muitas vezes carecem da flexibilidade, escalabilidade e capacidade de resposta em tempo real necessárias para lidar com a natureza dinâmica dos deslizamentos de encostas. Além disso, muitas soluções existentes são limitadas por conectividade inadequada, especialmente em áreas remotas ou com infraestrutura de rede instável.

1.2 Justificativa

As plataformas existentes de IoT tais como ThingsBoard (2024), MathWorks (2024) ou SiteWhere (2024) permitem integração entre disposivos embarcados com interfaces disponíveis online. Porém o seu uso requer conhecimento técnico e funcionalidades como processamento em borda e emissão de alertas muitas vezes são complexos de configurar. No entanto, cenários de monitoramento ambiental frequentemente requerem esse tipo de abordagem. Com estruturas para rapidamente adicionar novos dispositivos, poder acessar seus dados por meio de visualizações online mediadas por autênticação e emissão de alertas.

O monitoramento urbano, por sua vez, também conta com o acesso a informações ainda não totalmente exploradas por serem incipientes ou recentes em muitos locais. A estrutura de rede 5G, por exemplo, permite a análise inteligente dos dados através de seu núcleo (BARMPOUNAKIS et al., 2020), utilizando funções como *Multi-access Edge Computing* (MEC) e *Software-Defined Networks* (SDN). Estas tecnologias possibilitam o ajuste das rotas de rede de forma distribuída e automática através da análise da variação de métricas de qualidade de sinal como *Received Signal Strength Indicator* (RSSI) e *Signal-to-Noise Ratio* (SNR).

Essas informações, já disponíveis nas redes 5G por conta da sua estrutura de processamento distribuído, podem ser usadas como complemento para a modelagem urbana. Os fatores que afetam o sinal de comunicação também podem afetar as condições ambientais locais como aumento da incidência de chuvas, por exemplo, mas apresentando um retorno mensurável muito rápido e com infraestrutura de sensoreamento já configurada. Um efeito colateral da rede, que pode ser utilizado para uso em ações preventivas, mediante parcerias.

Embora existam plataformas IoT consolidadas, essas soluções têm caráter mais genérico e

exigem configurações complexas, o que limita sua adoção em cenários críticos de monitoramento ambiental. Além disso, muitas dessas plataformas não oferecem fluxos otimizados para a rápida implantação em campo, dificultando sua utilização por órgãos públicos, pesquisadores e comunidades que necessitam de respostas imediatas diante de riscos como deslizamentos de encostas.

Nesse sentido, o diferencial deste trabalho está em propor uma plataforma com fluxos projetados especificamente para monitoramento ambiental com foco em monitoramento de deslizamento de encostas. Permitindo a integração com múltiplos protocolos de comunicação. Inicialmente, o foco recai sobre HTTP, por ser amplamente utilizado, facilitar a integração com sistemas legados e permitir maior interatividade com aplicações externas, e sobre MQTT, protocolo consolidado em soluções IoT pela leveza e eficiência em cenários de conectividade limitada. Essa abordagem visa preencher a lacuna existente entre plataformas genéricas e as necessidades reais de monitoramento em campo, oferecendo um ferramental mais direto, acessível e orientado ao contexto geotécnico.

1.3 Objetivos

Objetivo Geral

Desenvolver uma plataforma IoT, denominada EdgeWidgets, projetada especificamente para monitoramento ambiental, que combine sensoriamento permitindo a coleta de dados em tempo real de sensores para monitoramento.

Objetivos Específicos

- Analisar abordagens com tecnologias emergentes como as funcionalidades do *Fifth Generation Mobile Network Technology* (5G) para monitoramento ambiental.
- Implementar um sistema de sensoriamento simulando o cenário de monitoramento ambiental em encostas, como estudo de caso base. Se utilizando de sensores e controladores equivalentes ao utilizados em cenários de integração IoT.
- Desenvolver uma arquitetura de comunicação com suporte a múltiplos protocolos visando atender diferentes cenários de integração.
- Aplicar metodologia de medição de latência end-to-end diretamente no dispositivo edge, permitindo a comparação empírica de desempenho.

1.4 Estrutura do Trabalho

A dissertação está dividida em 6 capítulos, os quais serão brevemente destacados a seguir.

O **Capítulo 1** apresenta a introdução ao trabalho, contextualizando o problema de pesquisa, definindo objetivos e justificativas, além de delinear a estrutura geral da dissertação e abordar os trabalhos relacionados sobre fundamentos em monitoramento geotécnico, análise comparativa de protocolos IoT, sistemas de monitoramento baseados em eventos geotécnicos, estratégias de monitoramento sistemático e análise das contribuições existentes na área.

O **Capítulo 2** apresenta a fundamentação teórica necessária, cobrindo Internet das Coisas (IoT) e sua evolução, protocolos de comunicação para sistemas IoT (*Hypertext Transfer Protocol* (HTTP) e *Message Queuing Telemetry Transport* (MQTT)), tecnologias *Micro-Electro-Mechanical Systems* (MEMS) e sensoriamento de precisão, computação edge e processamento distribuído, qualidade de serviço e métricas de performance, segurança em sistemas IoT distribuídos, e monitoramento geotécnico com suas aplicações e técnicas.

O **Capítulo 3** descreve a metodologia de pesquisa adotada, incluindo a metodologia geral do trabalho, abordagens específicas para análise de aplicações baseadas em funções da rede, e metodologia para elaboração de ferramental de integração IoT.

O **Capítulo 4** trata do desenvolvimento do ferramental proposto, apresentando análise de processo, definição da arquitetura do sistema, desenvolvimento de hardware com ESP32 e BNO055, prototipagem e implementação de software para edge computing com estratégias de comunicação adaptativa.

O **Capítulo 5** apresenta os dois estudos de caso realizados com base nas análises elaboradas para este trabalho.

O **Capítulo 6** apresenta as conclusões do trabalho, consolidando os resultados obtidos nos estudos de caso, discutindo limitações identificadas e propondo direções para trabalhos futuros, incluindo perspectivas para expansão multissensorial e integração com tecnologias emergentes de visualização e análise de dados.

1.5 Trabalhos Relacionados

Jara Ochoa et al. (2023) conduziu análise comparativa de consumo de energia entre protocolos MQTT e HTTP especificamente para plataformas IoT em monitoramento de cadeia fria de longo prazo. A metodologia experimental implementou avaliação de consumo energético, latência de comunicação e confiabilidade de transmissão em condições laboratoriais controladas. Os resultados estabeleceram correlações quantitativas entre métricas de desempenho energético e características específicas de cada protocolo, determinando cenários ótimos de aplicação para sistemas IoT de monitoramento remoto. A experimentação no protocolo MQTT com *Quality of Service* (QoS) 0 e 1 mostra que o protocolo MQTT economiza mais energia que o protocolo HTTP (6.03% e 8.33%, respectivamente) em execuções de longo prazo. As técnicas de análise permitiram classificação e seleção do protocolo mais eficiente entre opções disponíveis, detectando diferenças mensuráveis em consumo de energia durante operações prolongadas.

SHARMA et al. (2023) desenvolveram um framework experimental baseado em aprendizado em conjunto para detecção, monitoramento, predição e alerta inteligente de deslizamentos em ambiente IoT-nuvem. A solução implementa sistema bifásico operando em cenários pré-deslizamento e pós-deslizamento. Durante o período pré-deslizamento, o sistema de monitoramento contínuo coleta dados de múltiplos sensores e, ao atingir limites críticos pré-estabelecidos, envia alertas automatizados para o departamento de Obras Públicas, acionando protocolos de resposta de emergência para reduzir tempo de resposta e potencialmente salvar vidas. No período pós-deslizamento, sensores ultrassônicos e módulos WiFi transmitem dados para placas Arduino que armazenam e enviam informações para a plataforma ThingSpeak, permitindo análise de risco retrospectiva e predição de condições futuras. O sistema utiliza técnicas avançadas de ensemble learning para correlacionar dados de múltiplos sensores com análise preditiva, identificando padrões complexos que precedem eventos de deslizamento. A limitação principal desta abordagem é a dependência de processamento centralizado em nuvem.

RAWAT; BARTHWAL (2024) propuseram o LANDSLIDE MONITOR, um sistema de monitoramento de deslizamentos em tempo real que aborda limitações de sistemas tradicionais como resposta lenta a eventos críticos e dependência de conectividade constante. O sistema de monitoramento em tempo real baseia-se em três tipos de parâmetros: hidrológicos, meteorológicos e geográficos. Estes parâmetros são utilizados para coletar e armazenar informações em tempo real em uma plataforma IoT cloud. A arquitetura implementa monitoramento contínuo multi-sensor, análise automatizada utilizando redes neurais para predição de locais de risco, e comunicação direta com autoridades responsáveis através de canais redundantes. O sistema incorpora algoritmos de machine learning para correlacionar dados históricos com condições atuais, melhorando precisão de predições através de aprendizado adaptativo. A solução demonstrou eficácia em cenários reais de teste, sendo testada e avaliada nas colinas Varunavat do distrito de Uttarkashi em Uttarakhand, Índia. Entretanto, o sistema baseia-se em processamento centralizado tradicional, não explorando benefícios de adaptação dinâmica entre múltiplos protocolos de comunicação.

LIU et al. (2023) desenvolveram uma abordagem para predição de deslizamentos baseada em sistemas de alerta precoce de baixo custo e sustentáveis utilizando tecnologias IoT. Sob chuva forte, o *Landslide Early Warning Systems* (LEWS) é considerado um método eficaz para fornecer avisos oportunos, mas os sistemas analisados anteriormente apresentam deficiências como alto custo e alto consumo de energia. A pesquisa enfatizou requisitos de custo-benefício e sustentabilidade ambiental para viabilizar aplicação em larga escala em regiões com recursos limitados. O sistema integra Internet das Coisas (IoT) e uma plataforma integrada alimentada por energia solar para aquisição de dados, transmissão de dados e análise de dados. O sistema implementou múltiplos pontos de monitoramento distribuídos que criam perfis comportamentais de encostas e identificam padrões temporais de movimentação de solo. A análise de dados utilizou algoritmos de predição baseados em séries temporais para antecipação de eventos críticos, demonstrando eficácia na detecção precoce de condições de instabilidade.

PECORARO; CALVELLO; PICIULLO (2019) investigaram estratégias de monitoramento para sistemas locais de alerta precoce de deslizamentos, estabelecendo framework conceitual para integração de diferentes tecnologias de sensoriamento. O principal objetivo deste estudo é a descrição e análise das estratégias de monitoramento implementadas dentro de sistemas locais de alerta precoce de deslizamentos *Local Landslide Early Warning Systems* (Lo-LEWS). A pesquisa analisou abordagens de monitoramento local versus remoto, identificando vantagens e limitações de cada estratégia em diferentes contextos geográficos e socioeconômicos. O autor destaca que para 28 dos 29 Lo-LEWS revisados, a principal variável usada para emitir um aviso é a velocidade do deslocamento. Os autores propuseram uma metodologia para seleção de estratégias de monitoramento baseada em características específicas do local, recursos disponíveis e requisitos de tempo de resposta. O trabalho estabeleceu critérios para avaliação de eficácia de sistemas de alerta precoce, incluindo métricas como precisão, tempo de resposta e cobertura geográfica.

LAU et al. (2023) elaboraram um sistema de monitoramento de deslizamentos induzidos por chuva em Songmao e Lushan, Taiwan, utilizando IoT e sistemas de monitoramento baseados em big data. Este estudo apresenta um novo sistema de monitoramento de deslizamentos, que é projetado para ser custo-efetivo, robusto, flexível e escalável, visando monitoramento de longo prazo e tempo real e implantação em larga escala em ambientes hostis e remotos. A pesquisa implementou rede de sensores distribuídos que coletam dados meteorológicos e geotécnicos em tempo real, integrando informações de múltiplas fontes para análise preditiva. Este sistema de monitoramento sem fio consiste no nó sensor híbrido (incluindo o receptor GNSS e acelerômetro MEMS), a antena GNSS, o sistema de alimentação e o centro de dados com uma infraestrutura de big-data. Utilizando técnicas de big data para processar grandes volumes de dados históricos e atuais, identificando padrões complexos que precedem eventos de deslizamento tais como processamento distribuído e utilização de GPS de precisão (com bibliotecas como RTKLIB para refino).

LIU et al. (2024) propuseram predição dinâmica de expectativa de vida de deslizamentos utilizando sistema ensemble que incorpora modelos de predição clássicos com machine learning. Com o desenvolvimento do sistema de monitoramento de deslizamentos, muitas tentativas foram feitas para prever o tempo de falha do deslizamento utilizando dados de monitoramento de deslocamentos. A abordagem combina técnicas tradicionais de análise geotécnica com algoritmos avançados de aprendizado de máquina para melhorar precisão de predições temporais. Um sistema ensemble é proposto para integrar os modelos clássicos para alcançar maior qualidade de predição. O sistema implementa análise contínua de degradação de estabilidade de encostas, fornecendo estimativas probabilísticas de tempo até falha estrutural. Um "índice de descrédito"é proposto para avaliar a qualidade de predição dos métodos sob predição dinâmica.

1.6 Análise Comparativa e Contribuições

A análise dos trabalhos relacionados revela lacunas significativas que motivam as contribuições desta pesquisa. A Tabela 1.1 apresenta comparação sistemática entre as principais características dos trabalhos analisados.

Tabela 1.1: Comparação Expandida dos Trabalhos Relacionados

ID	Mon. Geotéc.	Edge Comp.	Prot. Múlt.	Amb. Real	ML/AI
Jara Ochoa et al. (2023)			✓	✓	
SHARMA et al. (2023)	✓				✓
RAWAT; BARTHWAL (2024)	✓			✓	✓
LIU et al. (2023)	✓			✓	
PECORARO; CALVELLO; PICIULLO (2019)	✓			✓	
LAU et al. (2023)	✓			✓	✓
LIU et al. (2024)	✓				✓
Este Trabalho	✓	✓	✓	✓	

A análise aplicada verificou a abordagem dos trabalhos em questão com relação a sua plicabilidade em Monitoramento Geotécnico (foco em questões relacionada a riscos de deslizamento), Edge Compute (específicamente visando a utilização de capacidades edge como diferencial). Além de Protocolos Múltipos como foco nos resultados obtidos, uso em Ambiente Real (se a solução foi elaborada para uso em cenários reais com implementação em sensores in-loco ou através de outros mecanismos) e também uso de Machine Learning / Artificial Inteligence na solução proposta como diferencial.

Com base na análise dos trabalhos relacionados, foi possível perceber que a maioria dos trabalhos utiliza processamento centralizado, não explorando benefícios de computação edge diretamente para redução de latência e melhoria de confiabilidade. Poucos trabalhos conduziram análise experimental comparando desempenho de protocolos, especificamente para o contexto de monitoramento geotécnico e que processamento centralizado é predominante na maioria das soluções, não aproveitando capacidades de edge computing para análise local e autonomia operacional.

1.7 Considerações Finais

A revisão sistemática da literatura evidencia uma evolução consistente nas tecnologias de monitoramento de deslizamentos, com avanços significativos na integração de sensores IoT, algoritmos de machine learning e sistemas de comunicação. Os trabalhos analisados demonstram a progressão desde sensores multiparamétricos básicos até sistemas complexos de big data e ensemble learning.

Observa-se uma tendência crescente na utilização de tecnologias IoT para monitoramento geotécnico, com trabalhos recentes incorporando análise preditiva avançada e sistemas de alerta automatizados. Entretanto, persiste uma lacuna significativa na exploração de arquiteturas dual-protocol e processamento edge, áreas onde esta pesquisa contribui de forma original.

Esta pesquisa contribui para o estado da arte através do desenvolvimento de uma metodologia que integra capacidades de edge computing, protocolos múltiplos adaptativos e validação experimental controlada, preenchendo lacunas identificadas nos trabalhos relacionados e avançando o conhecimento em sistemas IoT para monitoramento geotécnico resiliente. A abordagem proposta estabelece fundamentos para uma nova geração de sistemas de monitoramento que combinam precisão sensorial, eficiência de comunicação e autonomia operacional.

2

Fundamentação Teórica

Este capítulo apresenta a fundamentação utilizada para o desenvolvimento do sistema EdgeWidgets e a aplicação nos estudos de caso abordados no trabalho em questão.

2.1 Metodologia Geral

O framework metodológico utilizado fundamenta-se nos estudos apresentados por MACIEL (2023), compreendendo seis etapas principais: Entendimento do Sistema, Coleta Preliminar de Dados, *Exploratory Data Analysis* (EDA), Coleta de Dados, Análise dos Dados e Apresentação de Resultados. A Figura 2.1 apresenta uma visão geral do fluxo metodológico.

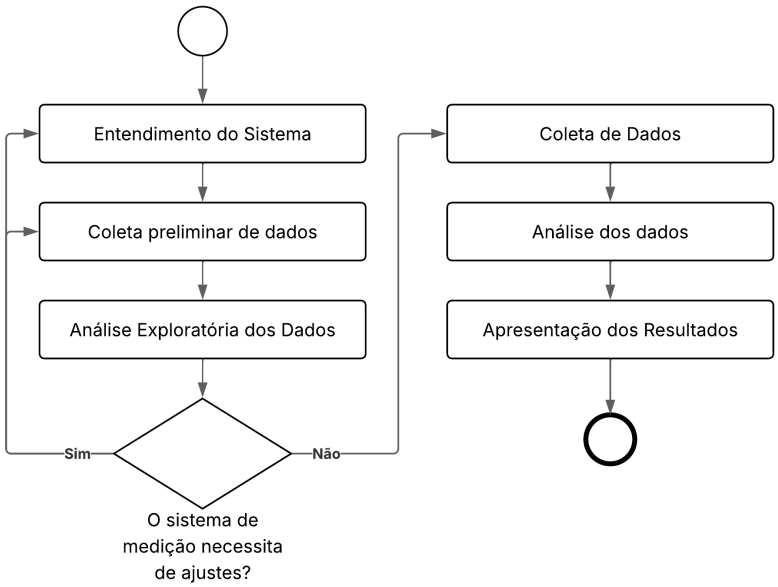

Figura 2.1: Visão Geral da Metodologia baseado em MACIEL (2023)

O Entendimento do Sistema envolve a definição dos componentes, entradas e saídas do sistema, estabelecendo os serviços que serão analisados através de representações das cargas de trabalho esperadas. O objetivo é identificar os fatores que influenciam a performance de

cada serviço, determinando o que deve ser mensurado, onde a medição será aplicada, como será executada, por quem e com qual frequência, estruturando assim os dados necessários para coleta. A Coleta Preliminar de Dados depende da utilização de mecanismos de medição disponíveis para obtenção de dados, definindo período, frequência e tamanho das amostras. Inclui a realização de backup dos dados coletados.

Acerca da Análise Exploratória dos Dados, a EDA objetiva compreender os dados obtidos através do cálculo de estatísticas da amostra para visualizações básicas relacionadas ao tipo e conjunto de dados. Quando o conjunto não está categorizado, busca-se evidências que suportem a análise por meio de distribuição probabilística através de testes de Goodness of Fit. Verifica-se a necessidade de ajustes e geram-se relatórios descrevendo o fitting de distribuição. A exploração aborda a utilização de visualizações disponíveis conforme estruturado na Figura 2.2.

Essa etapa permite verificar se existem erros no processo de medição e auxiliar na seleção do modelo além de relações entre as variáveis. Havendo a variação do tipo ideal de visualização conforme o tipo de dado (Categórico ou Quantitativo). Para o tipo Quantitativo deve ser verificado o tipo de gráfico (Resumo ou visualização de todos os pontos), objetivo (Visualização da distribuição ou Visualização da Tendência central e Dispersão). Para o tipo Categórico deve ser verificado o objetivo (Comparação entre classes ou a identificação de classes importantes).

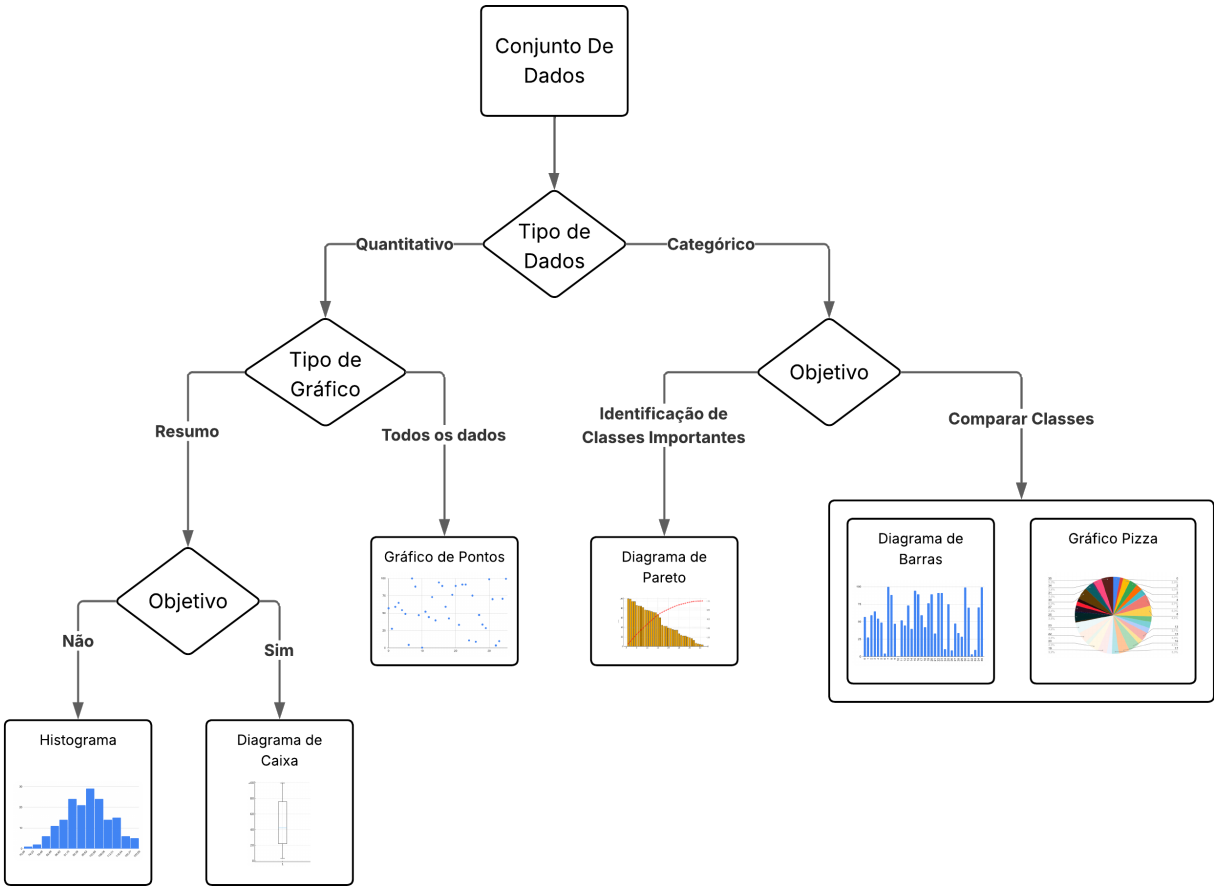

Figura 2.2: Tipos de diagramas e quando utilizar. Adaptado de MACIEL (2023)

Para cada serviço se define a etapa de Coleta de Dados, organiza-se o sistema de medição, realizando a coleta em conjunto com a obtenção das métricas em análise. Ao final, executa-se o backup dos dados. Utilizam-se ferramentas estatísticas para processar os conjuntos de dados coletados e aplicar a Análise de Dados logo em seguida. Podendo haver filtragem de informações para leitura apenas do relevante para a análise. Esta etapa inclui o armazenamento das estatísticas, geração de gráficos e backup dos resultados. O fluxo de análise dos dados pode ser visto na Figura 2.3.

Visando compreender a relevância estatística da amostra, calculando o intervalo de confiança esperado com base na comparação entre duas distribuições (de modo a evitar que o resultado obtido seja resultante de algo aleatório ou não relacionado com a alteração de aspectos do sistema). Se baseia em primeiramente entender a distribuição dos dados. Se for uma distribuição normal - através da análise da normalidade dos dados - é necessário verificar se o desvio padrão é conhecido para o sistema. Caso seja, deve ser usado o valor crítico (Z) - Com base nos valores críticos para distribuições normais - para obter o intervalo de confiança.

Caso contrário, deve ser utilizada a distribuição t de Student - com valores críticos da distribuição conhecidos - com base no desvio padrão da amostra.

Caso não seja uma distribuição normal, será necessário definir uma distribuição que melhor se enquadre. Nesse caso é necessário verificar se os dados se enquadram em algum conjunto de dados com distribuição conhecida (por meio de métodos como as transformações de Box-Cox e Yeo Johnson), uma vez transformado podemos aplicar os valores críticos com base em distribuição normal ou distribuição t de Student.

Caso não seja possível a transformação. Precisamos verificar se alguma distribuição se enquadra. Havendo uma expressão conhecida para o intervalo de confiança, deve ser utilizado. Em caso contrário, utilizar bootstrap semi-paramétrico. Não sendo possível definir uma distribuição normal nem transformar em uma distribuição normal ou expressão que possa analisar a distribuição teórica que represente os dados. Então, pode ser utilizado o Teorema de Limite Central e o uso de bootstrap não paramétrico. A normalidade da amostra por sua vez pode ser verificada através de métodos de Goodness of Fit (Bondade de ajuste). Como Probability-to-Probability Plot e o método Kolmogorov-Smirnov (KS test - utilizado nesse trabalho).

O método consiste em verificar se dois conjuntos de dados diferem de forma significativa. Comparando uma distribuição empírica e outra teórica com base em duas hipóteses (H_0 e H_1): As distribuições se aproximam e as distribuições são distintas. Dada a amostra de análise e uma distribuição teórica, deve-se obter a Função de Distribuição Cumulativa Empírica da amostra (CDF empírica) e a Função de Distribuição Cumulativa Teórica (CDF teórica). Calcular a maior diferença entre as distribuições e comparar com a maior diferença crítica para o grau de confiança desejado. Se a maior diferença calculada for maior que a crítica, então a hipótese H_0 (hipótese nula) é rejeitada pois a diferença é significativa. Caso contrário, podemos afirmar com o grau de confiança proposto a normalidade do conjunto.

Para os conjuntos em análise nesse trabalho, o desvio padrão dos sistemas não será dado como conhecido, sendo assim utilizando a distribuição t de Student e a função teórica de comparação é a distribuição Gaussiana. Dado dois conjuntos de dados que possam ser comparáveis em forma de teste pareado (medição de mesma grandeza como latência, delay, entre outros) o fluxo completo de validação será o seguinte:

1. Verificar a normalidade com teste KS

a. Consideramos as seguintes hipóteses:

- H_0 (Hipótese Nula): O conjunto possui normalidade comparável à função teórica definida dentro do intervalo de confiança
- H_1 (Hipótese Alternativa): O conjunto não possui normalidade comparável à função teórica definida dentro do intervalo de confiança

b. Obter o CDF da amostra (CDF Empírica) e da distribuição teórica através das fórmulas:

CDF Empírica:

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{x_i \leq x} \tag{2.1}$$

CDF Teórica (Normal):

$$F(x) = \Phi\left(\frac{x-\mu}{\sigma}\right) \tag{2.2}$$

onde Φ é a função de distribuição normal padrão, μ é a média da amostra e σ é o desvio padrão da amostra.

c. Calcular a diferença entre as distribuições através da estatística de Kolmogorov-Smirnov:

$$D = \max_x |F_n(x) - F(x)| \tag{2.3}$$

Comparar com a diferença crítica para grau de confiança de 95%:

$$D_\alpha = \frac{1{,}36}{\sqrt{n}} \tag{2.4}$$

d. Se a maior diferença calculada (D) for superior à diferença crítica (D_α), então a hipótese nula é rejeitada. Caso contrário, podemos prosseguir para a verificação da relevância estatística assumindo normalidade dos dados.

2. Verificar o enquadramento nos valores críticos da distribuição t de Student

a. Consideramos as seguintes hipóteses:

- H_0 (Hipótese Nula): Não há diferença estatística significativa entre os dois conjuntos ($\mu_1 = \mu_2$)

- H_1 (Hipótese Alternativa): Há diferença estatística significativa entre os dois conjuntos ($\mu_1 \neq \mu_2$)

b. Calcular as diferenças pareadas e suas estatísticas:

Diferenças pareadas:

$$d_i = x_{1i} - x_{2i}, \quad i = 1, 2, \ldots, n \tag{2.5}$$

Média das diferenças:

$$\bar{d} = \frac{1}{n} \sum_{i=1}^{n} d_i \tag{2.6}$$

Desvio padrão das diferenças:

$$s_d = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (d_i - \bar{d})^2} \tag{2.7}$$

c. Calcular a estatística t pareada:

$$t = \frac{\bar{d}}{s_d / \sqrt{n}} \tag{2.8}$$

com graus de liberdade $df = n - 1$.

d. Para o teste bilateral (bicaudal), compara-se o valor absoluto da estatística t calculada com o valor crítico $t_{\alpha/2,df}$ para $\alpha = 0{,}05$. O termo "bilateral"refere-se ao fato de que estamos testando se há diferença significativa em qualquer direção (maior ou menor), não apenas em uma direção específica. Portanto, a região de rejeição está dividida igualmente entre as duas caudas da distribuição, com $\alpha/2 = 0{,}025$ em cada cauda.

Critério de decisão:

- Se $|t| > t_{\alpha/2,df}$: Rejeita-se H_0 (há diferença estatística significativa)
- Se $|t| \leq t_{\alpha/2,df}$: Não se rejeita H_0 (não há evidência de diferença estatística significativa)

e. Calcular o intervalo de confiança para a diferença média:

$$IC_{95\%} = \bar{d} \pm t_{\alpha/2,df} \times \frac{s_d}{\sqrt{n}} \tag{2.9}$$

O procedimento algorítmico completo para execução desta análise estatística está detalhado no Apêndice A.

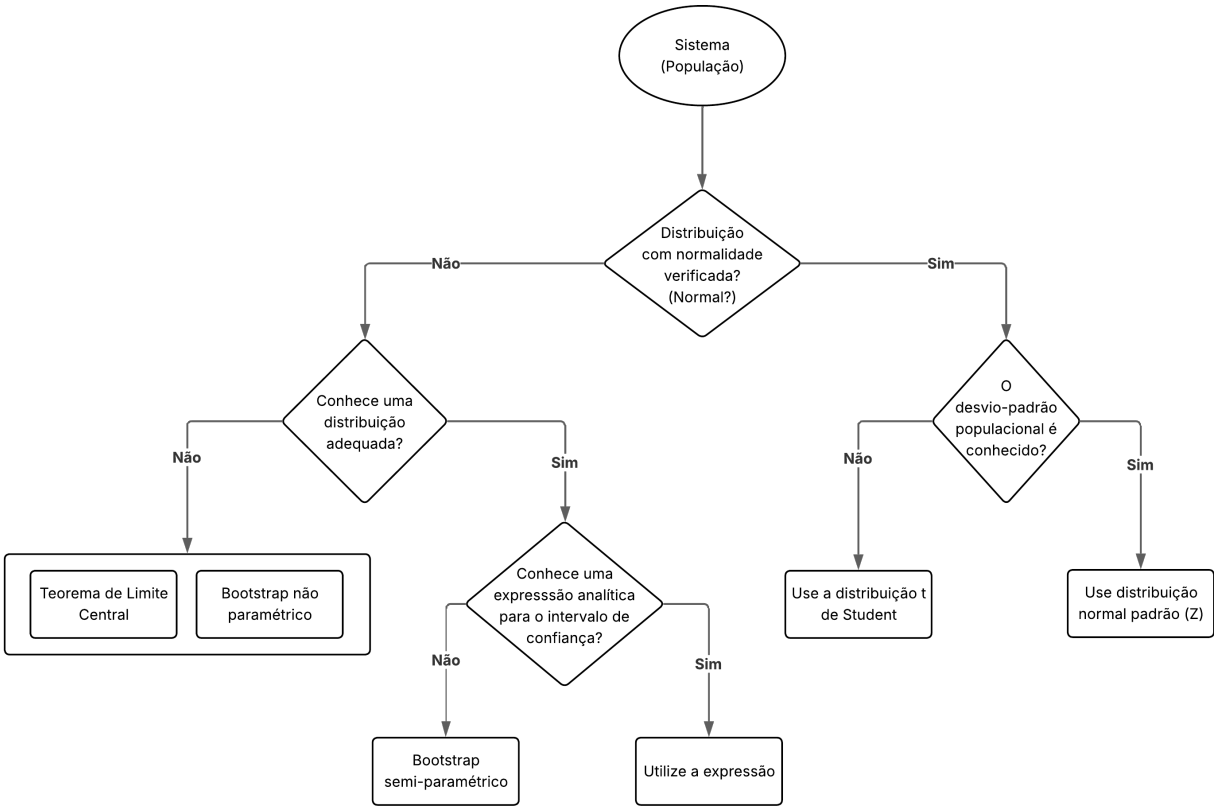

Figura 2.3: Metodologia de Análise de Dados. Adaptado de MACIEL (2023)

Havendo todas as informações disponíveis e processadas, ocorre a Apresentação dos Resultados. Que depende dos objetivos e da audiência relacionada, podendo resultar em relatórios ou apresentações que destacam os resultados obtidos.

2.2 Monitoramento Geotécnico

O monitoramento geotécnico de deslizamentos baseia-se na compreensão dos mecanismos físicos que governam estabilidade de encostas (CRUDEN; VARNES, 1996). Deslizamentos são fenômenos complexos influenciados por múltiplos fatores: geologia, geomorfologia, hidrologia, e atividades antrópicas. PECORARO; CALVELLO; PICIULLO (2019) apresentam análise de estratégias de monitoramento para *sistemas locais de alerta precoce*, identificando que eficácia do monitoramento depende da compreensão dos mecanismos de ruptura específicos e seleção de parâmetros apropriados. Diferentes tipos de deslizamentos exigem mecanismos de monitoramento distintos (INTRIERI et al., 2012). O estudo apresenta informações sobre Tipo de Deslizamento (deslizamento de destroços, avalanche de rochas ou deslizamento rochoso), Causa do Deslizamento (chuva e derretimento de neve), Mecanismos de Monitoramento (GPS, Inclinômetro Tilt, Câmera ou Piezoelétricos), Critério de Alerta (emissão acústica, medição de velocidade, pluviometria, pluviometria acumulada e aceleração), Modos de Notificação (SMS, chamada telefônica, página web, notificações automáticas), entidades que devem receber

o alerta (residentes, pesquisadores, centro de prevenção de alertas) e ente decisório (Defesa Civil, Governo, Autoridades Rodoviárias) - evidenciando as principais partes relacionadas ao gerenciamento de risco de deslizamento em cenários práticos.

Instrumentação geotécnica tradicional desenvolveu-se ao longo de décadas para fornecer medições de alta precisão de parâmetros críticos (DUNNICLIFF, 1988), os Inclinômetros de Furo são definido como padrão-ouro para medição de deslocamento subsuperficial. Sistemas tradicionais apresentam limitações para monitoramento contínuo em tempo real. Inclinômetros convencionais requerem acesso manual para leituras (limitando frequência de aquisição). Custos elevados de instrumentação e instalação restringem densidade de monitoramento - potencialmente resultando em cobertura inadequada. Sistemas IoT modernos oferecem vantagens complementares que abordam limitações de instrumentação tradicional (ATZORI; IERA; MORABITO, 2010). Sensores *MEMS de inclinação* (embora com precisão moderada) oferecem monitoramento contínuo, transmissão automática de dados, e custos reduzidos permitindo deployment em densidade superior (RAMESH; KUMAR, 2016).

Combinação de sistemas tradicionais (para calibração) com redes densas de sensores IoT (para monitoramento contínuo) permitem obter informações complementares para validação dos resultados. Literatura recente comentada no capítulo de Trabalhos Relacionados (RAWAT; BARTHWAL, 2024; LIU et al., 2023) demonstra evolução em sistemas IoT aplicados a monitoramento de deslizamentos, com o desenvolvimento de frameworks experimentais baseados em *ensemble learning* para detecção, monitoramento, predição e alerta de deslizamentos em ambiente IoT-cloud - melhorias na precisão de predição via combinação de múltiplos algoritmos de *machine learning* (SHARMA et al., 2023). Além de abordagens baseadas em Random Forest, Gradient Boosting e Redes Neurais Artificiais, visando estimar risco de deslizamento com base em dados de sensores (acelerômetros, medição de umidade e sensores ultrassônicos) armazenados na nuvem (HONG et al., 2015).

BRITO (2013) analisa fundamentos para monitoramento de deslizamentos através do desenvolvimento de um *"Modelo de monitoramento de deslizamento de encostas por meio de sensor multiparamétrico"*. Esta pesquisa conduziu análise abrangente acerca de modelos de estabilidade existentes, citando como referência para análise de risco de deslizamento o modelo *Shallow Landsliding Stability Model* (SHALSTAB) (DIETRICH; MONTGOMERY, 1998) - relevância corroborada por revisões literárias sobre modelos de risco de deslizamento como KÖNIG; KUX; CORSI (2022) -, e identificou que as chuvas constituem o principal agente físico na deflagração de escorregamentos no Brasil. O autor investigou diferentes tipos de sensores para monitoramento geotécnico, incluindo strain gages (extensômetros), sensores de umidade, giroscópios, acelerômetros e sensores piezoelétricos. A contribuição metodológica incluiu o desenvolvimento de um sensor multiparamétrico integrado que combinava sensor ultrasônico, sensor de umidade, acelerômetro e extensômetro em uma única unidade. O trabalho realizou simulações controladas de deslizamento utilizando solo coletado de localidades com risco real,

estabelecendo protocolos experimentais que influenciaram pesquisas subsequentes na área.

Dando continuidade aos fundamentos estabelecidos, (BRITO, 2023) desenvolveu o Sistema de Monitoramento em Tempo Real para Aplicação Geotécnica em Encostas (SMTRAGE), representando evolução significativa na aplicação de tecnologias IoT para monitoramento geotécnico. Esta pesquisa abordou limitações identificadas em trabalhos anteriores através da integração de tecnologias de transmissão de dados *General Packet Radio Service* (GPRS) e *Long Range* (LoRa) para comunicação em ambientes remotos. O sistema implementou métodos aprimorados de análise pluviométrica para correlacionar precipitação com eventos de instabilidade, desenvolveu técnicas avançadas de medição de umidade incluindo métodos gravimétricos, reflectometria e micro-ondas, e integrou medição de deslocamento através de inclinômetros e acelerômetros de alta precisão. A arquitetura utilizou microcontroladores ESP32 com comunicação LoRa para transmissão de dados de longo alcance, e o sistema foi validado experimentalmente na Região Metropolitana de Recife. Embora represente avanço significativo na aplicação de IoT para monitoramento geotécnico, o trabalho utilizou arquitetura de comunicação baseada em protocolo único, diferindo da abordagem dual-protocol proposta nesta pesquisa.

2.3 Internet das Coisas

O conceito de Internet das Coisas foi mencionado pela primeira vez por Kevin Ashton em 1999 (ASHTON, 2009), sendo apresentado formalmente em 2005 pela *International Telecommunication Union* (ITU) (International Telecommunication Union, 2005), como descrito em ATLAM et al. (2017).

A *Internet das Coisas* é um paradigma computacional onde objetos físicos cotidianos ganham capacidades de sensoriamento, computação, comunicação e controle, permitindo integração em sistemas distribuídos (ATZORI; IERA; MORABITO, 2010). O conceito vai além de simples conectividade - os objetos físicos interagem autonomamente com o ambiente e entre si, criando infraestrutura de informação global dinâmica. Middlewares permitem a interação entre aplicações e as camadas de sensoriamento (representando propriedades como objetos).

ATZORI; IERA; MORABITO (2010) identificam três paradigmas que convergem na IoT: o paradigma *orientado a coisas* (focado nos objetos e suas capacidades), o paradigma *orientado à internet* (protocolos e infraestrutura de rede), e o paradigma *orientado à semântica* (representação e processamento de conhecimento). Esta convergência resulta em sistemas que combinam mundo físico e digital, aplicáveis em diversos contextos como Saúde, Logística e Indústria, demandando abordagens para padronização e segurança.

A heterogeneidade dos sistemas IoT demanda taxonomias para classificação. PERERA et al. (2014) propõe taxonomia baseada em capacidade de processamento de contexto, com as camadas de Aquisição de Contexto (Obtenção de informações por meio de dispositivos),

Modelagem de Contexto (Armazenamento das informações coletadas), Raciocíonio de Contexto (Aplicação de técnicas para refino e fusão de dados) e Distribuição de Contexto (distribuição via API, publicações e mecanismos de análise). Os autores apresentam técnicas para 'Raciocínio' sobre contexto: Aprendizagem Supervisionada (quando existe grande quantidade de dados e os retornos esperados são facilmente identificáveis), Aprendizagem Não Supervisionada (não necessita conjunto de treinamento, útil para identificação de comportamentos), Regras (definido manualmente, permite relacionar condições exatas para controle de eventos), Lógica difusa (definido manualmente, para eventos que necessitam análises contextuais), Ontologia (depende da estruturação do conhecimento, mas permite análises complexas) e Probabilidade (modelagem probabilística para unificar informações de diferentes fontes, aproveitando probabilidades conhecidas).

As arquiteturas de sistemas IoT evoluíram de modelos simples de três camadas para estruturas hierárquicas que refletem a natureza distribuída destes sistemas (AL-FUQAHA et al., 2015). A arquitetura de referência clássica compreende três camadas fundamentais que fornecem separação de responsabilidades e facilitam modularidade.

A *camada de percepção* (base da hierarquia) engloba os dispositivos físicos que fazem interface com o mundo real. Inclui sensores para aquisição de dados, atuadores para modificação do ambiente físico, dispositivos de identificação (*Radio Frequency Identification* (RFID) e *Near Field Communication* (NFC)), e elementos de processamento local básico. A diversidade tecnológica reflete a variedade de fenômenos físicos monitorados e controlados.

A *camada de rede* assume responsabilidade de conectividade e transmissão de dados entre dispositivos e sistemas superiores. Acomoda múltiplas tecnologias de comunicação: redes de curto alcance (*Bluetooth* e *ZigBee*) até conectividade de longa distância (redes celulares e satelitais). A heterogeneidade de protocolos introduz desafios de interoperabilidade e gestão de qualidade de serviço.

A *camada de aplicação* fornece interfaces e serviços para usuários finais: dashboards de visualização, *Application Programming Interface* (API) de integração com sistemas corporativos, e aplicações mobile para acesso remoto. Esta camada traduz dados brutos em informações acionáveis, apresentando-as de forma intuitiva.

O protocolo HTTP, encontrou aplicação extensiva em sistemas IoT pela sua simplicidade, maturidade tecnológica e compatibilidade universal. O HTTP, pode ser estruturado através da organização REST, havendo requisições baseadas em recursos e formatos bem definidos, permitindo maior escalabilidade de interações entre componentes, generalidade de interfaces, deployment independente de componentes, e uso de componentes intermediários para reduzir latência, reforçar segurança e encapsular sistemas legados (FIELDING; TAYLOR, 2002). A natureza textual e semântica bem definida do HTTP facilitam desenvolvimento e depuração de aplicações IoT - características valiosas em prototipagem rápida.

O modelo *request-response* do HTTP oferece semântica clara para operações *Create, Read, Update, Delete* (CRUD) em recursos (RICHARDSON; RUBY, 2008). Isso permite implementação direta de APIs para controle e monitoramento de dispositivos, onde sensores podem ser representados como recursos acessíveis via URLs específicas (GUINARD et al., 2011). Exemplo: leitura de sensor de temperatura via requisição GET para '/sensors/temperature/01', configuração de parâmetros usando PUT ou POST.

A natureza *stateless* do HTTP simplifica implementação de servidores IoT. Define que cada requisição do cliente ao servidor deve conter toda a informação necessária para compreender a requisição, sem depender de contexto armazenado no servidor - característica que facilita escalabilidade horizontal. *Load balancers* distribuem requisições arbitrariamente entre servidores sem considerar afinidade de sessão.

ISMAIL; HAMZA; KOTB (2018) compararam APIs HTTP *Representational State Transfer* (REST) com MQTT em diferentes condições de carga (latência, throughput e utilização de recursos). HTTP apresentou overhead computacional superior em cenários de alta frequência de transmissão (processamento de headers textuais e estabelecimento repetitivo de conexões). THANGAVEL et al. (2014) demonstraram em experimentos que MQTT apresenta eficiência significativamente superior ao HTTP em termos de largura de banda para cenários de publicação frequente de mensagens pequenas, com ganhos que podem atingir uma ordem de magnitude.

Versões recentes do protocolo trazem melhorias para IoT. *HTTP/2* implementa *multiplexing* de streams - múltiplas requisições simultâneas sobre uma única conexão *Transmission Control Protocol* (TCP), reduzindo latência de estabelecimento de conexão (BELSHE; PEON; THOMSON, 2015). O mecanismo de *server push* permite que servidores enviem recursos antecipadamente (útil para distribuição de configurações ou atualizações de firmware). *HTTP/3* (baseado no protocolo *Quick UDP Internet Connections* (QUIC) sobre *User Datagram Protocol* (UDP)) oferece reduções de latência através da eliminação do handshake TCP e implementação de correção de erro integrada (LANGLEY et al., 2017). Relevante para monitoramento ambiental em locais remotos onde conectividade instável e alta latência são desafios (BISHOP, 2022).

O MQTT emergiu como protocolo predominante para IoT pelo design para ambientes com recursos limitados e conectividade instável (HUNKELER; TRUONG; STANFORD-CLARK, 2008). A arquitetura *publish-subscribe* do MQTT oferece desacoplamento temporal e espacial entre produtores e consumidores de dados. EUGSTER et al. (2003) explicam que no paradigma publish-subscribe os publicadores e assinantes são desacoplados no tempo (não precisam estar ativos simultaneamente), no espaço (não se conhecem) e na sincronização (produção e consumo de eventos podem ser assíncronos). *Publishers* publicam mensagens em tópicos hierárquicos sem conhecimento dos consumidores, enquanto *subscribers* expressam interesse em tópicos específicos. O *broker* MQTT atua como intermediário inteligente, gerenciando

subscrições, roteamento e persistência (MISHRA; KERTESZ, 2020; BANKS; GUPTA, 2014).

O sistema hierárquico de tópicos MQTT permite organização lógica e filtragem granular de dados (YOKOTANI; SASAKI, 2016). Tópicos estruturados hierarquicamente usando separadores de barra (/), permitindo subscrições específicas e genéricas via *wildcards*. Exemplo: sistema de monitoramento organiza dados como ‘site/location/sensor_type/sensor_id‘ - subscrições específicas (‘site/A/temperature/01‘) ou genéricas (‘site/A/+/+‘ para todos os sensores de um local).

O sistema de *qualidade de serviço* (QoS) é uma das características importantes do MQTT para monitoramento crítico (STANFORD-CLARK; TRUONG, 2013). *QoS 0* (At Most Once): entrega *best-effort* sem acknowledgment (dados não-críticos com alta frequência). *QoS 1* (At Least Once): garante pelo menos uma entrega via acknowledgments (balanceia confiabilidade e performance). *QoS 2* (Exactly Once): implementa entrega garantida sem duplicação via handshake de quatro etapas (comandos críticos (SINGH et al., 2015)).

A tecnologia MEMS permite integração de componentes mecânicos miniaturizados com circuitos eletrônicos em escala microscópica (SIMOES et al., 2023; YAZDI; AYAZI; NAJAFI, 1998). Inicialmente usados em sistemas de segurança automotivos, possuem características como resistência a vibrações, baixo consumo energético e baixos custos de produção.

Os princípios físicos que governam sensores MEMS baseiam-se na conversão de fenômenos mecânicos em sinais elétricos via efeitos físicos bem compreendidos (BEEBY et al., 2004). O *efeito piezoresistivo* forma a base de muitos sensores MEMS - deformação mecânica resulta em mudança proporcional da resistência elétrica (BARLIAN et al., 2009). O *princípio capacitivo* é outra abordagem fundamental - mudanças na distância entre placas condutoras resultam em variações de capacitância (AMARASINGHE et al., 2007). Sensores capacitivos oferecem vantagens: estabilidade térmica, baixo consumo energético e alta resolução (características essenciais para monitoramento contínuo em aplicações alimentadas por bateria).

O design estrutural de acelerômetros capacitivos utiliza configurações de placas paralelas onde a massa prova móvel forma capacitores variáveis com eletrodos fixos (LEMKIN; BOSER, 1999). A Aceleração aplicada causa deslocamento da massa prova proporcional à força inercial - aumento de capacitância em um lado e diminuição no lado oposto. Esta configuração diferencial oferece vantagens: rejeição de interferências e linearidade de resposta (WU et al., 2002).

O condicionamento de sinal on-chip diferencia sensores MEMS modernos de implementações discretas (YAZDI; AYAZI; NAJAFI, 1998). Circuitos integrados incluem: amplificação de baixo ruído, filtragem anti-aliasing, conversão analógico-digital de alta resolução, e compensação de temperatura. Técnicas avançadas (*oversampling* com modulação *delta-sigma* e filtragem digital) permitem resolução efetiva superior à resolução nominal do conversor (SAUKOSKI; AALTONEN; HALONEN, 2007). A calibração de fábrica reduz variações de processo e offset

inicial. Algoritmos de auto-calibração contínua compensam deriva de longo prazo. Sensores modernos incorporam referências internas estáveis e circuitos de teste para verificação durante operação.

Giroscópios MEMS operam baseados no *efeito Coriolis* - onde um objeto em movimento em sistema rotativo experimenta força perpendicular à velocidade do objeto e ao eixo de rotação. PASSARO et al. (2017) explicam que a força de Coriolis é proporcional ao produto vetorial da velocidade angular e da velocidade linear, permitindo detecção de rotação através da medição de forças secundárias induzidas por este efeito. Este princípio tem sido explorado em giroscópios MEMS desde os anos 1990. A implementação típica usa estruturas vibrantes em *modo ressonante* - massa oscila continuamente em frequência predeterminada (ACAR; SHKEL, 2009). Quando o sensor experimenta rotação angular, o efeito Coriolis induz movimento secundário perpendicular ao eixo de vibração primária (amplitude proporcional à velocidade angular aplicada).

A combinação de dados de múltiplos sensores via algoritmos de *fusão* permite estimação robusta de orientação tridimensional, superando limitações individuais (SABATINI, 2006). Acelerômetros: informação confiável sobre orientação gravitacional em condições estáticas, mas susceptíveis a acelerações lineares externas. Giroscópios: resposta dinâmica para mudanças rápidas de orientação, mas sofrem de deriva integral em baixas frequências (WOODMAN, 2007).

O *filtro de Kalman* representa a abordagem teoricamente ótima para fusão de sensores. KALMAN (1960) desenvolveu um algoritmo recursivo de processamento de dados que combina todas as medições disponíveis com conhecimento prévio sobre o sistema para produzir estimativas ótimas das variáveis de interesse, minimizando o erro estatístico. Para sistemas lineares com ruído gaussiano, oferece estimação de mínimo erro quadrático médio. O filtro estima estado atual e incerteza associada, permitindo pesagem ótima entre predições e observações (WELCH; BISHOP et al., 1995).

Para sistemas não-lineares (típicos de aplicações de orientação), o filtro de Kalman estendido (*Extended Kalman Filter* (EKF)) lineariza localmente a dinâmica do sistema via expansão de Taylor de primeira ordem (WAN; VAN DER MERWE, 2000). Embora não mantenha optimalidade teórica, o EKF oferece performance prática e complexidade computacional gerenciável. Algoritmos de filtros complementares oferecem alternativa computacionalmente eficiente explorando características complementares dos sensores no domínio da frequência (MAHONY; HAMEL; PFLIMLIN, 2008). Estes algoritmos implementam filtragem passa-altas para dados de giroscópio e passa-baixas para acelerômetro - combinando as saídas para obter estimação de orientação que mantém estabilidade de longo prazo do acelerômetro com resposta dinâmica do giroscópio.

2.4 Computação Edge e Processamento Distribuído

A evolução dos paradigmas computacionais reflete mudanças nas necessidades de processamento, características das aplicações e limitações tecnológicas. SATYANARAYANAN (2017) argumenta que o surgimento da computação edge é consequência direta da convergência entre computação móvel e Internet das Coisas - mudança paradigmática que representa transformação fundamental na arquitetura de sistemas distribuídos. O modelo de *computação centralizada* em mainframes (dominante nas décadas de 1960-1980) concentrava todo processamento em sistemas de grande porte com terminais passivos. A emergência de computadores pessoais (década de 1980) introduziu o paradigma *client-server*, distribuindo processamento entre clientes com capacidades computacionais e servidores especializados. Esta distribuição permitiu melhor utilização de recursos e maior flexibilidade arquitetural, mas introduziu complexidades de coordenação e sincronização. O *grid computing* estendeu processamento distribuído para redes de computadores heterogêneos geograficamente dispersos (FOSTER; KESSELMAN; TUECKE, 2001). Este paradigma possibilitou criação de supercomputadores virtuais via agregação de recursos distribuídos, mas enfrentou desafios de latência de rede e heterogeneidade. A *computação em nuvem* ofereceu recursos computacionais como serviços.

ARMBRUST et al. (2010) definem que a computação em nuvem refere-se tanto às aplicações entregues como serviços pela Internet quanto ao hardware e software de sistemas nos datacenters que fornecem esses serviços. Os autores observam que o aspecto de computação utilitária (capacidade de pagar conforme o uso) não é novo, mas a escala e acessibilidade alcançadas pela nuvem representam avanço significativo. *Cloud computing* democratizou acesso a recursos computacionais massivos e introduziu modelos de serviço flexíveis como *Infrastructure as a Service* (IaaS), *Platform as a Service* (PaaS) e *Software as a Service* (SaaS). No entanto, centralizar informações na nuvem, apesar de permitir grande escalabilidade, cria dependência com conectividade constante e pode aumentar a latência das comunicações. Sendo esse um problema para sistemas críticos ou que necessitem de operação em ambientes com conectividade restrita ou inexistente.

Nesse cenário emerge o Edge Compute, que consiste na utilização da borda da rede e seus dispositivos para computação ao invés de delegar esse processo totalmente para servidores em nuvem (SHI et al., 2016)). Para haver comunicação com as redes em cenários de Internet das Coisas, é necessário haver unidades computacionais na ponta para conectividade e processamento. Microcontroladores são formas estabelecidas de realizar essa integração. Para esse trabalho em específico o *ESP32 C3 SuperMini* foi escolhido, corresponde a um microcontrolador com processador *RISC-V* de 32 bits com conectividade Wi-Fi nativa, recursos de gestão energética (Espressif Systems, 2021). Recursos de *gestão energética* permitem operação otimizada para aplicações alimentadas por bateria (BANSAL; KUMAR, 2020). Como o modo *deep sleep* que consome apenas 5µA mantendo apenas Real-Time Clock ativo ou o modo *light sleep* com

consumo de 130µA mantendo RAM ativa, possibilitando despertar rápido para processamento de eventos críticos.

2.5 Qualidade de Serviço e Métricas de Performance

A *qualidade de serviço* em sistemas IoT apresenta desafios adicionais com relação a modelos tradicionais de QoS. AL-FUQAHA et al. (2015) enfatizam que QoS em ambientes IoT é mais complexo que em redes tradicionais devido à heterogeneidade de dispositivos, topologia dinâmica, restrições severas de recursos e requisitos diversos de aplicações. Os autores argumentam que uma abordagem única de QoS não é adequada para ambientes IoT. A heterogeneidade dos dispositivos, variabilidade de condições de rede, e requisitos diversos demandam abordagens de QoS adaptáveis.

Existem diversos modelos de QoS para diferentes cenários como os modelos *Best Effort* e *Integrated Services* (IntServ). Sendo o primeiro a abordagem mais simples (pacotes tratados igualmente sem garantias de desempenho), sendo amplamente utilizado pela simplicidade e baixo overhead. Enquanto o segundo oferece garantias de QoS via reserva explícita de recursos end-to-end, usando protocolo *Resource Reservation Protocol* (RSVP) para sinalização (BRADEN; CLARK; SHENKER, 1994). Para ambientes IoT emergem modelos híbridos, o *Class-based QoS* agrupa dispositivos com requisitos similares e *Context-aware QoS* adapta dinamicamente políticas baseadas em informações contextuais como localização, energia disponível e prioridade da aplicação.

Avaliação de performance em sistemas IoT requer métricas específicas, em especial para essa dissertação a *Latência end-to-end*, *Jitter*, *Confiabilidade de entrega* e *Disponibilidade do sistema*. A *Latência end-to-end* engloba tempo de transmissão de rede, atrasos de aquisição de dados, processamento em dispositivos edge, transmissão através de múltiplos saltos e processamento em servidores remotos (STANKOVIC, 2014). O *Jitter* corresponde a variação estatística da latência, sendo relevante em aplicações de monitoramento sísmico ou de vibração, onde irregularidades temporais podem mascarar eventos importantes ou gerar falsos positivos (MILLS, 1991). A *Confiabilidade de entrega* equivale ao percentual de mensagens entregues com sucesso dentro de janela temporal) é uma métrica relevante para sistemas de monitoramento de segurança (CHACHULSKI et al., 2007). Deve considerar perdas de rede, falhas de dispositivos, interrupções de energia, e condições ambientais adversas. A *Disponibilidade do sistema*, tradicionalmente medida como percentual de tempo operacional requer redefinição para sistemas IoT onde operação parcial pode ser aceitável (PATTERSON et al., 2002). Desses mencionados, a mais analisada em próximos capítulos será a Latência End-to-End, no entanto os demais são muito relevantes para o entendimento das condições de operação e atividade de sistemas.

Sistemas IoT apresentam *superfície de ataque* expandida e vulnerabilidades únicas devido

a quantidade de pontos de entrada e middlewares. WURM et al. (2016) analisam que dispositivos IoT apresentam superfície de ataque expandida com vulnerabilidades únicas que decorrem de sua natureza distribuída, diversidade de implementações e restrições severas de recursos. Os autores alertam que muitos dispositivos IoT de consumo e industriais carecem até mesmo de proteções de segurança básicas. Heterogeneidade tecnológica resulta em implementações de segurança inconsistentes.

ROMAN; ZHOU; LOPEZ (2013) apresentam taxonomia de ameaças para IoT distribuída, identificando ataques que exploram a natureza distribuída. Ataques de *negação de serviço distribuído* (DDoS) representam ameaça crescente (ZARGAR; JOSHI; TIPPER, 2013). SICARI et al. (2015) destacam que escala massiva de deployments IoT torna impraticável gestão manual de segurança, demandando soluções automatizadas. O design de arquiteturas de segurança para IoT deve equilibrar proteção robusta com limitações práticas de recursos (JING et al., 2014).

Questões como *Autenticação* compõem desafio crítico onde métodos tradicionais baseados em certificados digitais podem ser computacionalmente proibitivos (GRANJAL; MONTEIRO; SILVA, 2015). *Public Key Infrastructure* (PKI) oferece segurança robusta via certificados digitais hierárquicos, mas demanda recursos computacionais. Algoritmos de *Elliptic Curve Cryptography* (ECC) proporcionam segurança equivalente ao RSA com chaves menores, reduzindo requisitos computacionais (HANKERSON; MENEZES; VANSTONE, 2006).

2.6 Considerações Finais

Esta fundamentação teórica estabelece base para compreender desafios e oportunidades no desenvolvimento de sistemas IoT para monitoramento de deslizamentos. A convergência de múltiplas tecnologias maduras cria condições favoráveis para implementação de sistemas mais eficazes que soluções atuais.

Os fundamentos apresentados informam decisões arquiteturais da plataforma EdgeWidgets - estabelecendo critérios técnicos para seleção de tecnologias, design de experimentos, e métricas de avaliação. A base teórica sustenta desenvolvimento de contribuições que avançam o estado da arte em sistemas IoT para monitoramento ambiental, fornecendo direcionamento para etapas subsequentes de desenvolvimento e validação experimental.

3

Metodologia

Este capítulo apresenta a metodologia utilizada para fundamentar os estudos de caso elaborados, detalhando cada etapa vinculada ao processo metodológico. São abordadas tanto a metodologia geral quanto as específicas derivadas para aplicação nos estudos de caso. A metodologia em ambos estudos de caso se baseia na definição geral explorada na fundamentação teórica.

3.1 Estudo de Caso 1: Análise de Aplicações Baseadas em Funções da Rede

Para análise de aplicações de monitoramento ambiental baseadas em funções de rede, foi estruturado um conjunto específico de etapas, constituindo uma variação da metodologia geral. A Figura 3.1 apresenta o fluxo metodológico específico para esta abordagem, destacando a integração entre eventos ambientais, torres celulares, funções de rede e sistemas de processamento de dados.

A metodologia ilustrada na Figura 3.1 demonstra o fluxo sequencial desde o entendimento do sistema, passando pela coleta preliminar de dados (que envolve histórico de eventos ambientais, histórico esperado da rede e definição de pontos críticos), análise exploratória dos dados, e as etapas subsequentes de coleta, análise e apresentação dos resultados. O diagrama evidencia a especificidade desta abordagem ao incorporar elementos específicos das funções de rede 5G no processo metodológico.

Entendimento do Sistema: A metodologia foca na modelagem de risco ambiental baseada nas funções de rede. O conceito fundamenta-se na premissa de que processos ambientais relacionam-se com causas vinculadas a grandezas físicas que também afetam a rede local. Este efeito é mensurável através das funções de rede em estruturas modernas como o 5G, possibilitando a análise da relação entre dados da rede e processos ambientais. A análise dessas informações depende de estrutura de simulação 5G, controle de rede equivalente em nível de testbed ou uso de modelagens previamente verificadas.

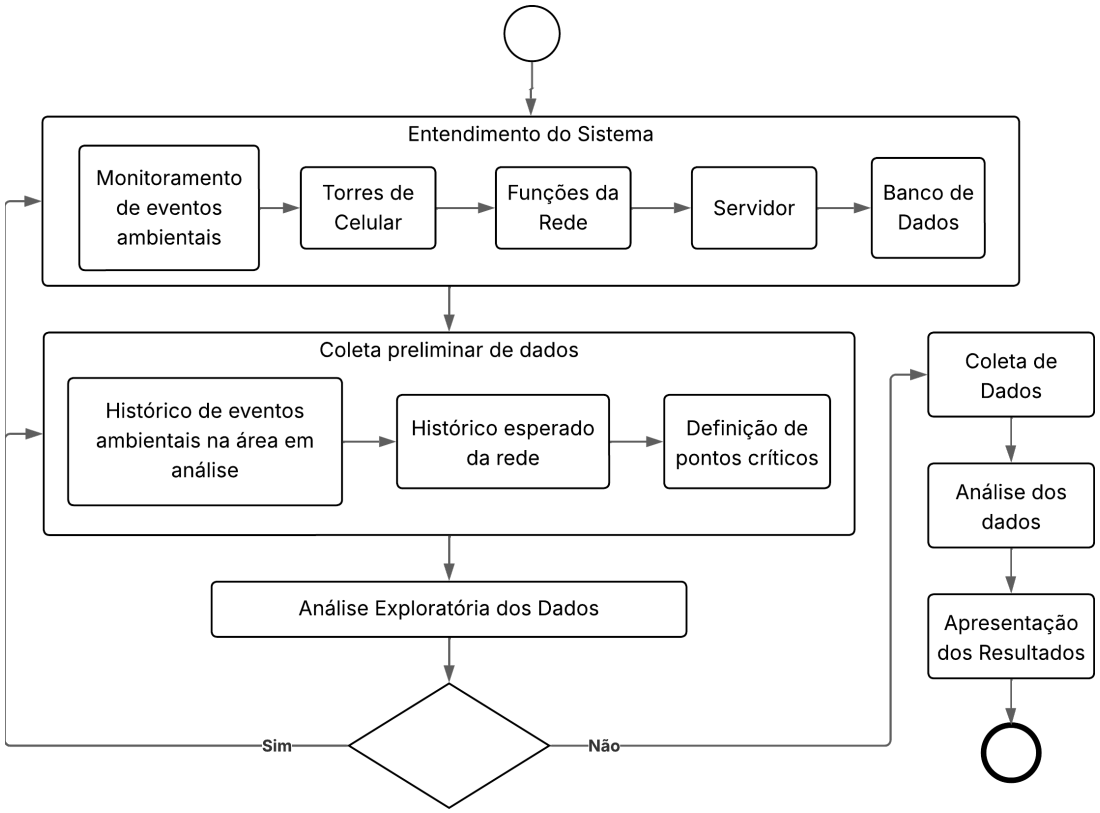

Figura 3.1: Metodologia para Análise de Aplicações Baseadas em Funções da Rede

O sistema é formado por aspectos reais que permitem análise das informações no cenário prático: o evento ambiental, torres de celular que recebem dados, funções de rede processadas em servidores sob controle da operadora e servidor próprio responsável por receber dados coletados, comparar com o modelo de risco utilizado e gerar alertas.

Monitoramento de eventos ambientais: O acompanhamento de processos como deslizamentos de terra ou inundações normalmente associa-se a variações físicas significativas, tais como chuvas prolongadas (aumento de pluviometria) ou alteração da inclinação do solo. Em casos envolvendo aumento de pluviometria, a qualidade dos sinais próximos será afetada.

Torres de Celular: O sinal da rede é emitido primariamente através de torres próximas, podendo envolver múltiplas unidades.

Funções da Rede: Em redes recentes como a 5G, existem funções específicas como SDN e MEC que permitem obter programaticamente informações do estado da rede, possibilitando a coleta de dados como RSSI e SNR.

Servidor: Através das Funções de Rede é possível comparar os valores obtidos com o histórico anterior da mesma localidade, baseando-se nas informações de localização de torres e valores de referência obtidos em modelos. Podem ser utilizadas bases de dados públicas ou coleta direta para geração das análises.

Banco de Dados: Responsável pelo armazenamento de informações relacionadas a históricos relevantes na modelagem de risco e dados coletados in loco.

Coleta Preliminar de Dados: Composta pela geração do modelo de risco e configuração

de alertas. O modelo é gerado através da análise do histórico de eventos ambientais, correlação dos elementos desse histórico e comparação com modelos existentes, relacionando o histórico de uma grandeza climática específica com uma que exista na rede. Dessa forma, a própria rede pode ser utilizada indiretamente como mecanismo de sensoriamento. Por exemplo, a utilização do histórico de pluviometria de uma localidade com a modelagem de RSSI por pluviometria, vinculando a informação com o RSSI obtido através da rede em determinada torre. A pluviometria relaciona-se com o risco de eventos como inundações e deslizamentos, possibilitando correlacionar grandezas distintas, considerando diferenças intrínsecas como a volatilidade do RSSI, mas permitindo visão geral do nível de risco sem medição direta da pluviometria na localização.

Histórico de eventos ambientais na área em análise: Para uma área definida onde ocorrerá o monitoramento (exemplo: intervalo de coordenadas que compõe um país ou região), o histórico compõe o conjunto de eventos organizados por data, horário e localização em coordenadas. O evento ambiental consiste no sinistro em análise (inundação, deslizamentos, terremotos). Havendo correlações diretas com bases de dados disponíveis, como aumento do nível de pluviometria, é necessário obter as informações como histórico de evento ambiental correlato.

Histórico esperado da rede: Uma vez localizados os eventos ambientais para análise, é necessário compreender a estrutura da rede existente, obtendo o registro das torres presentes na localidade para determinar a variação do sinal esperada com base na variação climática (relação RSSI e Pluviometria baseada em registro pluviométrico prévio).

Definição de pontos críticos: A emissão de alertas deve ocorrer quando os níveis da métrica original espelhada na métrica da rede estejam dentro do intervalo especificado para ocorrência do evento de sinistro em análise. Por exemplo, quando os níveis pluviométricos (possivelmente correlacionáveis com o RSSI do sinal) atingem patamares onde ocorrem deslizamentos. Nesse cenário, alertas podem ser identificados como pertinentes dentro de determinado intervalo (a ser definido no desenvolvimento, como utilizando percentil 10, 25 para níveis alto e médio).

Para as análises indicadas, além do sistema de integração com o mecanismo real (que compõe trabalho de implementação e integração com a operadora da rede), existe a obtenção de informações de históricos que podem ser obtidas de fontes públicas confiáveis e documentadas.

Análise Exploratória dos Dados: Os dados obtidos devem ser analisados para permitir a estruturação das correlações e análises, relacionando os eventos ambientais com outros históricos disponíveis e modelagens. Para essa análise especificamente será verificado a correlação entre os valores de dois modelos distintos: um modelo baseado em dados de RSSI (Received Signal Strength Indicator) obtidos via protocolo MQTT e outro modelo baseado em dados diretos de pluviometria. Considerando que o RSSI constitui uma medição indireta da pluviometria neste

contexto experimental, espera-se encontrar correlação significativa entre os modelos. Para essa análise será aplicada a correlação de Pearson e Mean Squared Error (MSE). O Coeficiente de Correlação de Pearson é calculado pela fórmula:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

onde r é o coeficiente de correlação de Pearson (-1 $\leq$ r $\leq$ 1), x_i e y_i são os valores individuais das variáveis, $\bar{x}$ e $\bar{y}$ são as médias das variáveis x e y, e n é o número de observações. O Mean Squared Error (MSE) é determinado por:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

onde y_i são os valores observados, $\hat{y}_i$ são os valores preditos pelo modelo, e n é o número de observações.

A interpretação do coeficiente de Pearson considera que valores próximos a +1 indicam correlação positiva forte, onde uma variável aumenta proporcionalmente à outra. Valores próximos a zero sugerem ausência de correlação linear entre as variáveis, enquanto valores próximos a -1 representam correlação negativa forte, onde uma variável diminui quando a outra aumenta. Considera-se correlação forte quando $|r| > 0,7$, moderada quando $0,3 < |r| < 0,7$, e fraca quando $|r| < 0,3$.

O MSE próximo a zero indica alta precisão do modelo experimental, enquanto valores elevados revelam baixa precisão com grandes discrepâncias entre valores observados e preditos. A avaliação do MSE deve considerar a escala dos dados e permitir comparação entre os modelos RSSI e pluviométrico.

Na aplicação desta análise exploratória, espera-se que a correlação entre o modelo RSSI-MQTT e o modelo pluviométrico apresente coeficiente de Pearson elevado ($|r| > 0,7$) combinado com baixo MSE, validando a hipótese de que as medições RSSI constituem um proxy adequado para a pluviometria. Correlação fraca ou MSE elevado indicariam limitações na capacidade do RSSI de representar indiretamente os dados pluviométricos, sugerindo necessidade de calibração dos sensores ou revisão da metodologia de coleta via MQTT.

3.2 Estudo de Caso 2: Elaboração de Ferramental de Integração IoT

Para o ferramental proposto para Integração IoT, foi estruturado um conjunto específico de etapas, constituindo uma variação da metodologia geral. A Figura 3.2 apresenta a visão geral desta metodologia, demonstrando o fluxo desde o entendimento do sistema baseado em sensores IoT até a apresentação dos resultados.

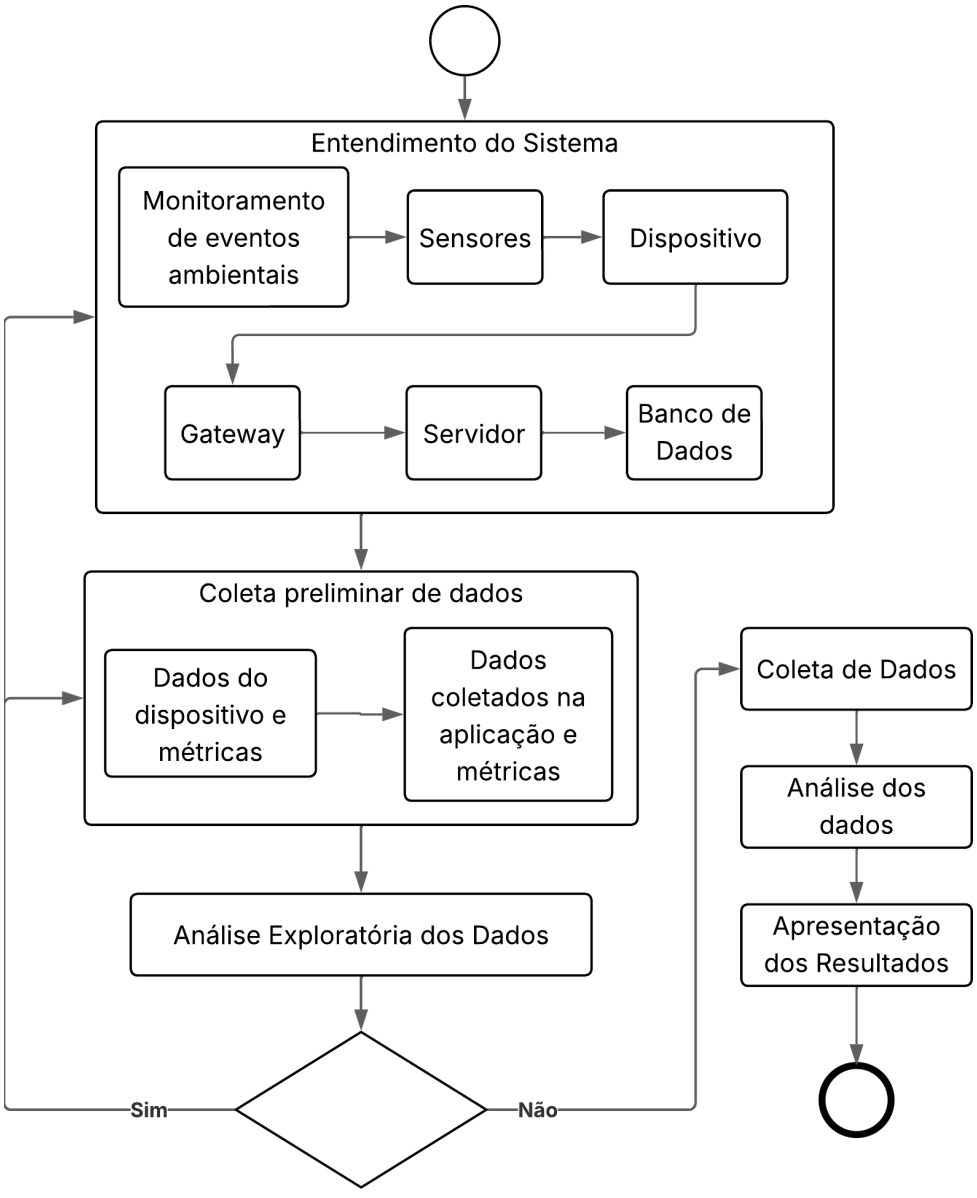

Figura 3.2: Visão Geral da Metodologia para Elaboração de Ferramental de Integração IoT

A metodologia apresentada na Figura 3.2 destaca a especificidade da abordagem IoT ao incluir na etapa de entendimento do sistema componentes específicos como sensores, dispositivos e gateway. A coleta preliminar de dados é focada na obtenção de dados do dispositivo e métricas da aplicação, diferenciando-se das outras metodologias por priorizar a medição direta através de sensores especializados.

Entendimento do Sistema: A metodologia foca em um sistema de integração IoT para monitoramento ambiental. Existe um processo ambiental em análise que altera determinada métrica mensurável localmente em um ponto de monitoramento. O dispositivo envia informações para a plataforma que deve receber através do protocolo utilizado, validar a autenticidade da requisição e armazenar os dados em banco de dados. Além disso, deve permitir a visualização das informações em interface online.

Monitoramento de eventos ambientais: Para que o acompanhamento ocorra, é necessário haver um processo periódico ou com risco de alteração que estará sendo monitorado (exemplo: deslizamento de terra, inundações, níveis de gases).

Sensores: O processo está vinculado a uma grandeza física mensurável (exemplo: nível pluviométrico, inclinação de encostas, umidade do solo, partículas por milhão de CO2).

Dispositivo: Os dados sensoreados são recebidos em uma unidade de processamento e armazenados, havendo o envio periódico por meio da rede e do protocolo definidos.

Gateway: A rede e protocolo dependem de diversos fatores, tanto da estrutura do dispositivo quanto do servidor e condições locais. Pode ser via Wi-Fi ou redes celulares, com o protocolo de comunicação podendo ser diversos entre os disponíveis (MQTT, HTTP, *Constrained Application Protocol* (CoAP), *Google Remote Procedure Call* (gRPC)).

Servidor: Processa os dados enviados pelo dispositivo e disponibiliza a interface da aplicação para interação com os dados recebidos. Possui os subcomponentes de *Recebimento de Dados* (responsável por receber as informações dos dispositivos por meio dos protocolos disponíveis), *Processamento da Plataforma* (responsável por ações em segundo plano e área administrativa) e distribuição da parte visual.

Banco de Dados: Responsável pelo armazenamento das informações do sistema.

Coleta Preliminar de Dados: Obtenção de informações do sistema por meio dos sistemas disponíveis, incluindo registros do dispositivo de envio de informações e também do software.

Análise Exploratória dos Dados: Os dados obtidos pelo ferramental são analisados para obter os principais resultados preliminares, validando o sistema de medição (via software e diretamente na borda) e alinhando métricas como latência. Mais detalhes vão ser abordados em 2.1.

3.3 Considerações Finais

As etapas subsequentes de ambas metodologias abordadas compreendem a coleta de dados, que uma vez configurado adequadamente o fluxo de medição, envolve a coleta dos dados para análise com definição de intervalos e uso de sistemas práticos, seguida pela análise de dados, que seguirá a metodologia detalhada na seção 2.1, e finalmente a apresentação dos resultados do trabalho em formato de artigo e relatório. Cada uma dessas etapas será detalhadamente descrita nas seções subsequentes.

As duas metodologias apresentadas neste capítulo compartilham a estrutura fundamental de seis etapas do framework proposto por MACIEL (2023), porém cada uma adapta-se às especificidades de seu domínio de aplicação. A metodologia geral oferece flexibilidade para análise de sistemas diversos, enquanto as metodologias específicas para funções de rede e integração IoT fornecem direcionamentos técnicos mais precisos para suas respectivas áreas.

A metodologia para análise de aplicações baseadas em funções da rede destaca-se no uso indireto de infraestrutura existente para monitoramento ambiental, aproveitando a correlação entre fenômenos ambientais e qualidade de sinal. Por sua vez, a metodologia para ferramental de integração IoT privilegia a medição direta através de sensores específicos, oferecendo maior precisão na coleta de dados, porém com maior complexidade de implementação e manutenção.

A escolha entre as abordagens metodológicas deve considerar fatores como disponibilidade de infraestrutura, precisão requerida, custos de implementação e escalabilidade do sistema. O conjunto de metodologias propostas oferece alternativas complementares para diferentes cenários de monitoramento ambiental, contribuindo para a flexibilidade e aplicabilidade dos estudos desenvolvidos.

A análise estatística proposta, fundamentada nos testes de normalidade de Kolmogorov-Smirnov e t-Student pareado, fornece base para validação da significância dos resultados obtidos. O procedimento algorítmico detalhado no Apêndice A permite a automatização desses procedimentos, garantindo rigor metodológico e reprodutibilidade dos experimentos. A implementação computacional completa está disponível no Apêndice B, incluindo exemplos práticos de aplicação da metodologia em diferentes cenários.

4

Desenvolvimento do Ferramental Proposto

Este capítulo apresenta o desenvolvimento e implementação da plataforma EdgeWidgets, seguindo uma metodologia estruturada que compreende análise de processos AS-IS/TO-BE, definição arquitetural, prototipagem, aplicação em estudos de caso e avaliação de resultados. O ferramental engloba uma pilha tecnológica completa, desde hardware embarcado até interfaces web modernas, passando por serviços de backend escaláveis e arquitetura de comunicação com duplo protocolo.

A plataforma EdgeWidgets foi projetada como uma solução modular e extensível para monitoramento ambiental, implementando uma arquitetura hierárquica que organiza o sistema em projetos, pontos de monitoramento, dispositivos, widgets e dashboards. Esta abordagem facilita manutenção, permite escalabilidade independente de componentes e habilita a evolução gradual do sistema.

4.1 Análise de Processo

Atualmente existem diversas plataformas de monitoramento IoT disponíveis. Tais como ThingsBoard, Sitewhere, Ubidots. Essas plataformas oferecem mecanismos distintos de configuração para sistemas. Normalmente focando em configuração de devices, com capacidades de configurações avançadas como firmware, dashboards, definição de regras, widgets entre muitas outras capacidades. O modelo foi estruturado visando apresentar uma estrutura mais simples para monitoramento ambiental: projetos, pontos de monitoramento, dispositivos, widgets e dashboards.

Os requisitos funcionais definidos contemplam operação dual-protocol com capacidade simultânea de comunicação HTTP e MQTT. O processamento edge avançado permite computação local para análise de desempenho e otimização de comunicação.

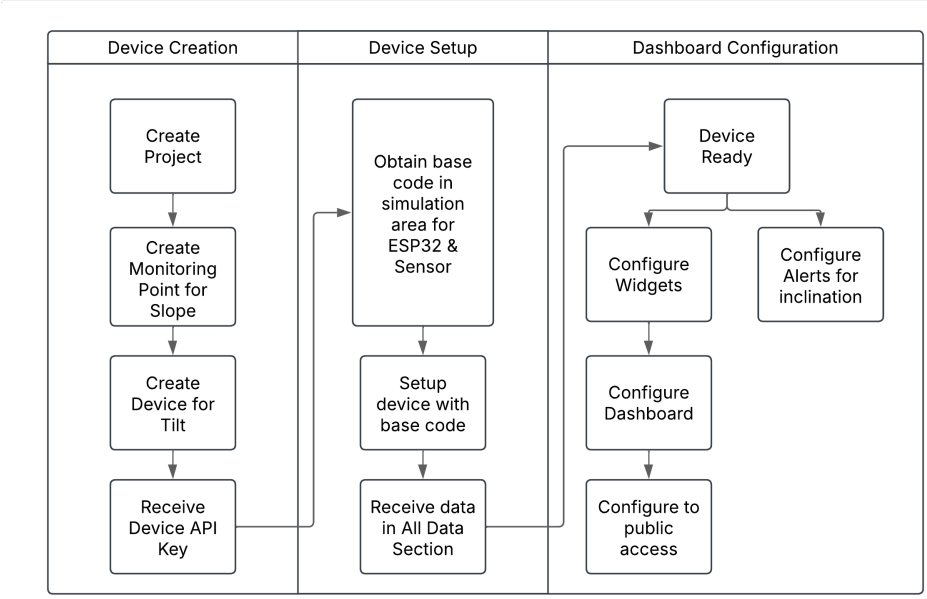

Figura 4.1: Arquitetura Hierárquica TO-BE do Sistema EdgeWidgets

4.2 Definição da Arquitetura

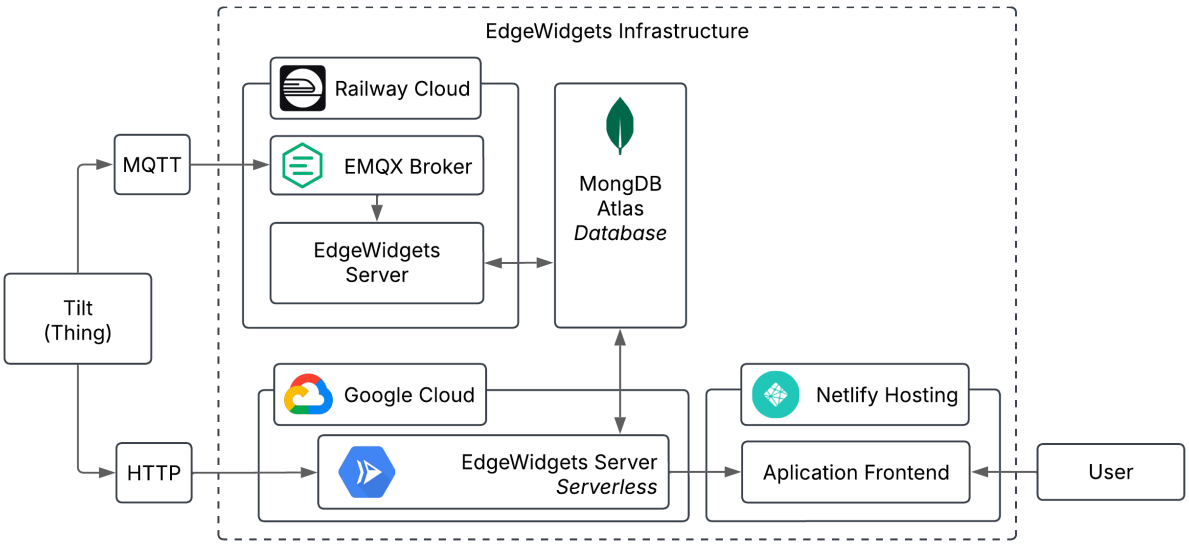

Figura 4.2: Visão Geral da Arquitetura de Infraestrutura Utilizada

A Camada de Sensoriamento utiliza hardware especializado para aquisição de dados físicos do ambiente, utilizando componentes para monitoramento ambiental de precisão. O sensor BNO055 da Bosch combina acelerômetro, giroscópio e magnetômetro com algoritmos de fusão sensorial embarcados, fornecendo dados de orientação com precisão de ±1° em condições normais de operação. O microcontrolador ESP32 C3 SuperMini foi utilizado para coleta dos dados e envio para o servidor, tanto como publisher como via REST API. O encapsulamento foi

estruturado através de impressão 3D customizada.

A Camada de Processamento de Borda implementa computação local para análise de desempenho da comunicação, incluindo a metodologia de medição edge-based para comparação de protocolos. A Camada de Rede gerencia protocolos HTTP e MQTT.

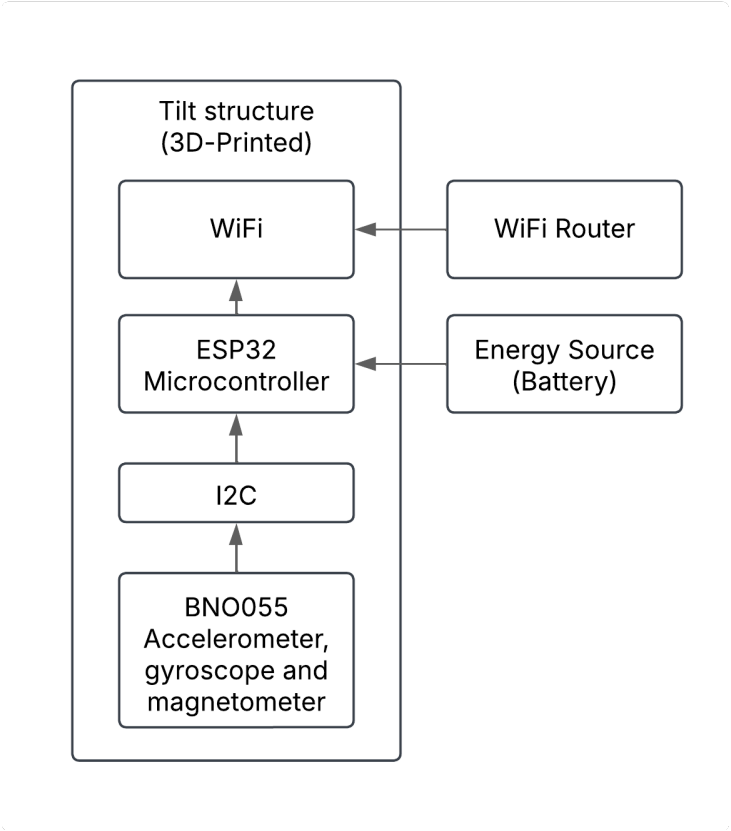

Figura 4.3: Estrutura da Camada de Sensoriamento e Camada de Processamento de Borda

A Camada de Serviços concentra lógica de negócio e orquestração distribuída através de serviços em execução no Google Cloud Run e no Railway. Os serviços implementados abrangem o Serviço de Autenticação responsável por gerenciamento de usuários com JWT. O Serviço de Projetos para gestão organizacional hierárquica. O Serviço de Dispositivos para pipeline de ingestão dual-protocol e armazenamento. O sistema de banco de dados escolhido foi o MongoDB Atlas, aproveitando flexibilidade de schema e capacidades de escalabilidade horizontal.

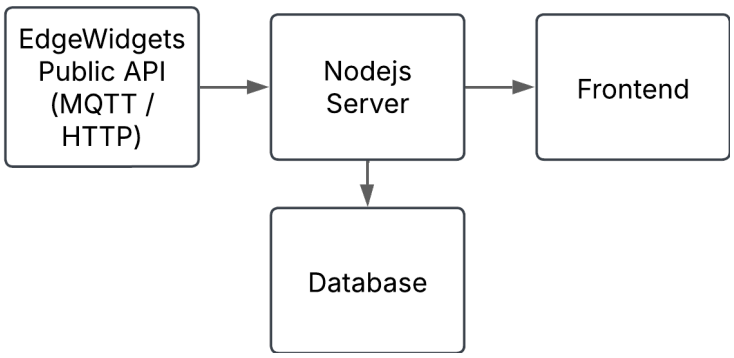

Figura 4.4: Arquitetura de Camada de Serviços

A Camada de Apresentação oferece interfaces responsivas para configuração e monitoramento através de aplicação web React com TypeScript. A interface foi desenvolvida como Single Page Application utilizando React 18 com TypeScript, priorizando type safety e experiência do desenvolvedor. O gerenciamento de estado utiliza Redux para estado global e Tailwind CSS para estilização responsiva.

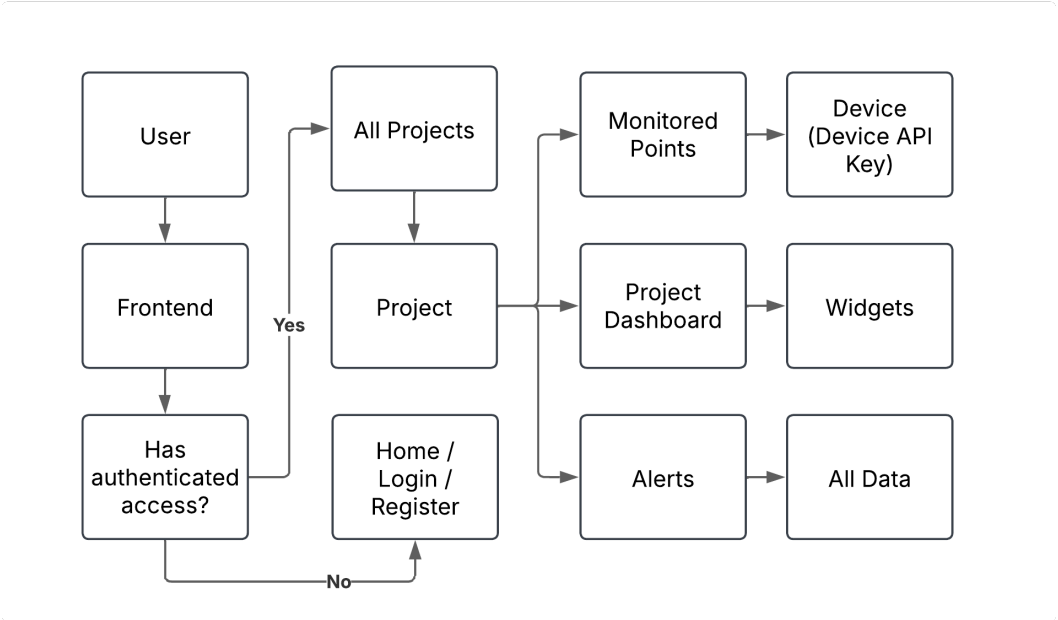

Figura 4.5: Estrutura da Camada de Apresentação

Os modelos de dados implementados compreendem Usuários para perfis, autenticação e preferências. Projetos gerenciam hierarquia organizacional e permissões. Pontos de Monitoramento armazenam localização geográfica e metadados. Dispositivos mantêm configuração, status e histórico de conexão. Widgets controlam configuração de visualização e tipos personalizados. Dashboards gerenciam layout responsivo e composição de widgets. Dados organizam séries temporais com indexação otimizada.

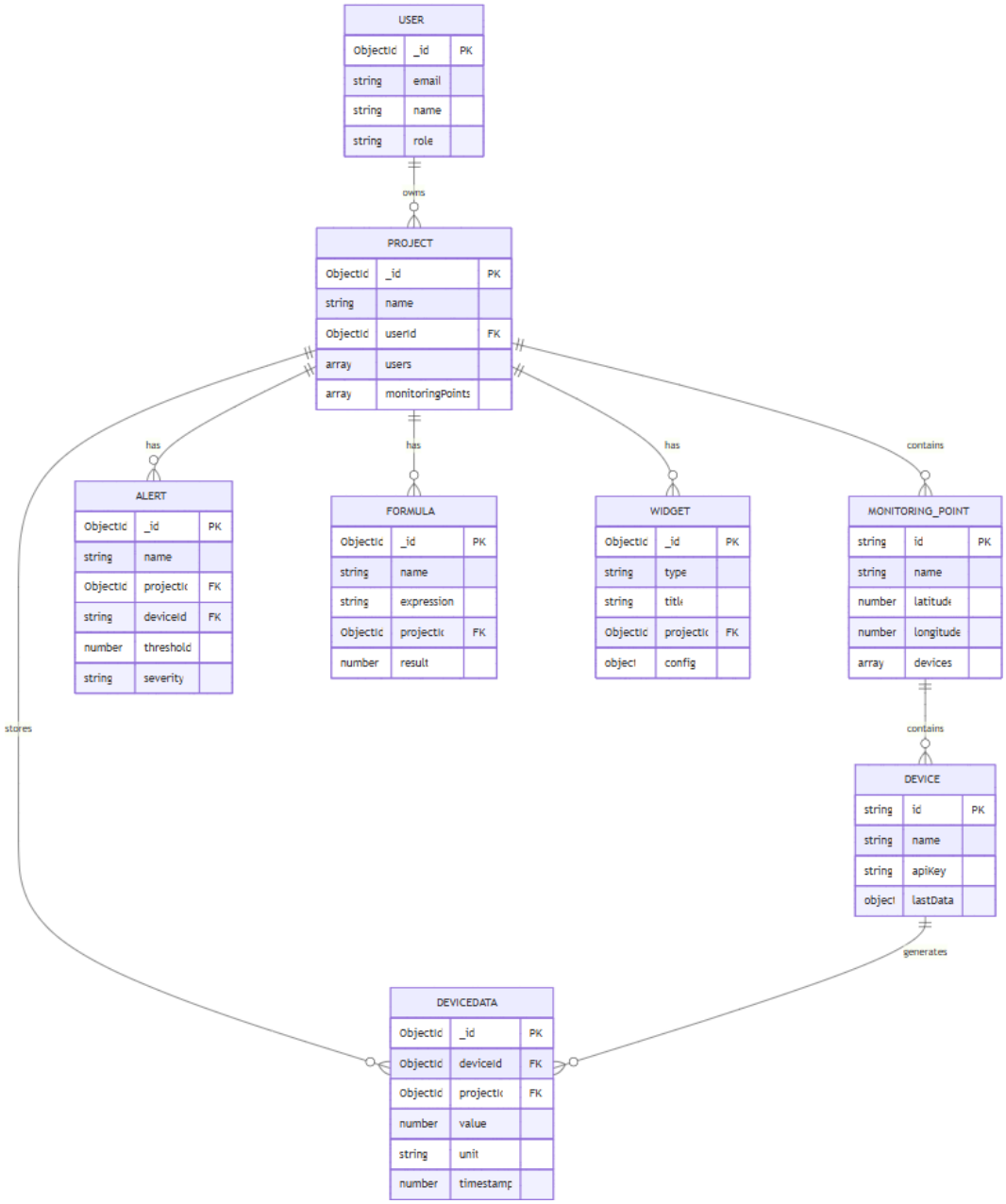

Figura 4.6: Arquitetura do Banco de Dados e Modelos de Dados

4.3 Prototipagem

A terceira etapa envolveu desenvolvimento iterativo de protótipos funcionais através de ciclos de duas semanas com entregáveis funcionais, seguindo práticas de Test-Driven Development e integração contínua para garantir qualidade de código e redução de riscos de integração. O firmware foi implementado seguindo arquitetura modular com separação clara de responsabilidades entre módulos de comunicação HTTP, comunicação MQTT, processamento de dados de sensores, e análise estatística em tempo real. A implementação dual-protocol permite operação simultânea de clientes HTTP e MQTT com medição comparativa de latência. Para medições HTTP, o firmware estabelece conexões com endpoints específicos da API EdgeWidgets e calcula latência como a diferença temporal entre o timestamp de iniciação da requisição e o timestamp de recebimento da resposta completa:

$$\text{Latência}_{HTTP} = T_{response} - T_{request} \tag{4.1}$$

Para medições MQTT, foi implementado padrão de comunicação bidirecional onde o dispositivo publica dados em tópico central e recebe confirmação através de tópico específico de resposta:

$$\text{Latência}_{MQTT} = T_{ack} - T_{publish} \tag{4.2}$$

A aplicação foi validada através de interfaces especializadas para diferentes aspectos do monitoramento ambiental. O sistema compreende Página Inicial, Entrada, Registro, Gestão de Projetos, Visão Geral, Dashboard, Dados, Widgets e Simulação.

Os experimentos foram configurados seguindo design experimental com taxa de amostragem de 0.5 Hz correspondendo a uma medição a cada dois segundos. O ambiente de rede Wi-Fi local controlado opera em frequência de 2.4 GHz com coleta de 150 medições por protocolo baseada em análise prévia. O ambiente experimental incluiu rede Wi-Fi dedicada, configuração padronizada para MQTT (havendo conexão com servidor MQTT EMQX). A coleta de dados foi realizada através de sistema automatizado de logging com timestamping em resolução de microssegundo utilizando clock nativo do ESP32.

4.4 Interfaces Principais

4.4.1 Interface de Autenticação

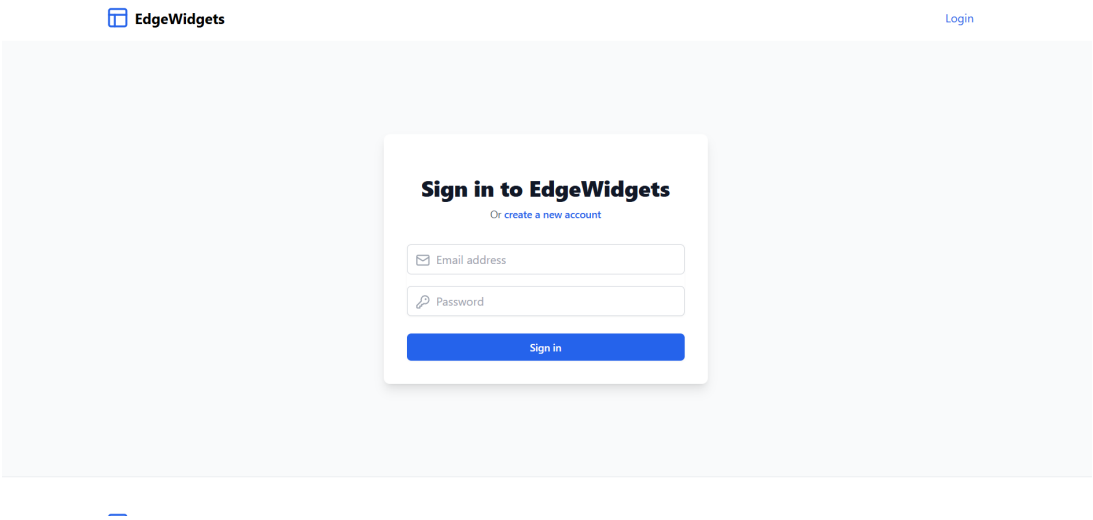

Figura 4.7: Interface de Login com campos de autenticação e acesso ao sistema

4.4.2 Gestão de Projetos

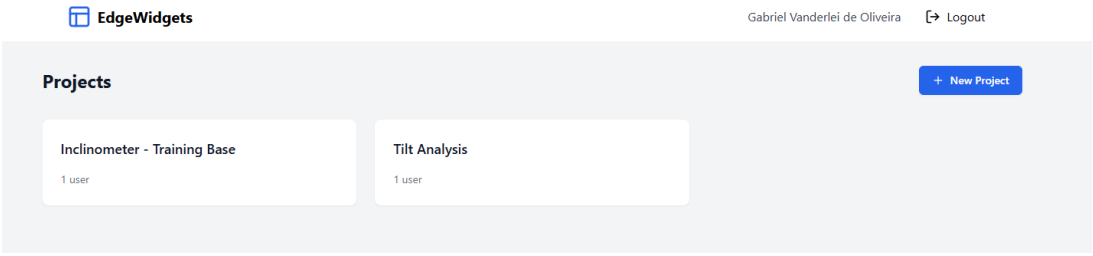

Figura 4.8: Interface de Gestão de Projetos permitindo criação, edição e organização hierárquica

4.4.3 Visão Geral do Monitoramento

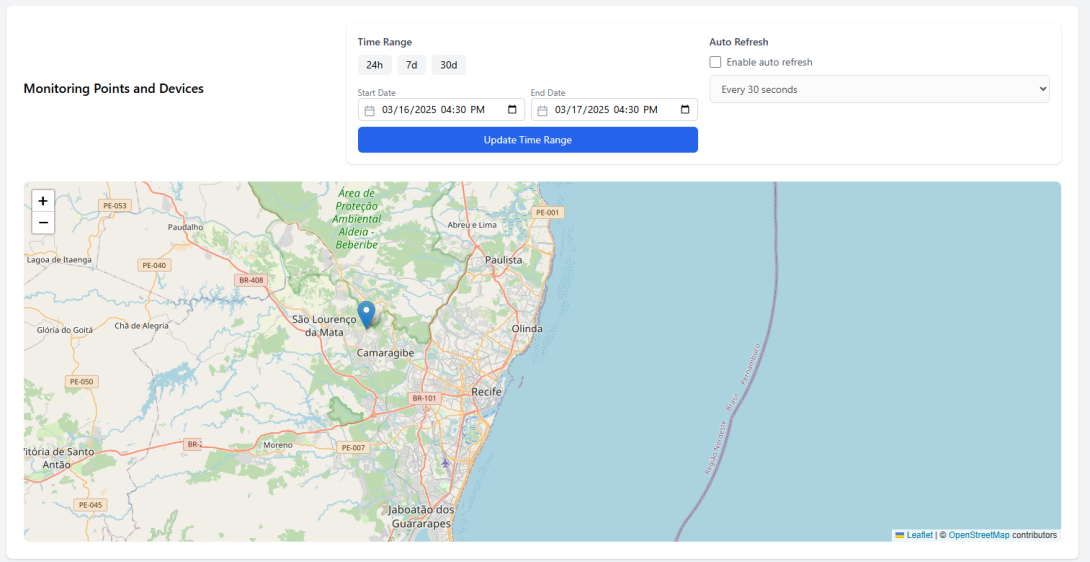

Figura 4.9: Visão Geral dos pontos de monitoramento com status e informações resumidas

4.4.4 Detalhamento dos Pontos de Monitoramento

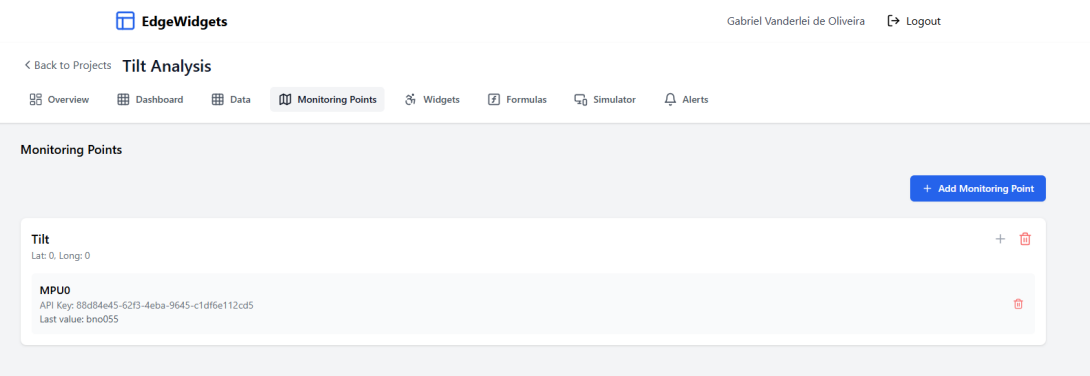

Figura 4.10: Interface detalhada dos Pontos de Monitoramento com configurações e histórico

4.4.5 Configuração de Widgets

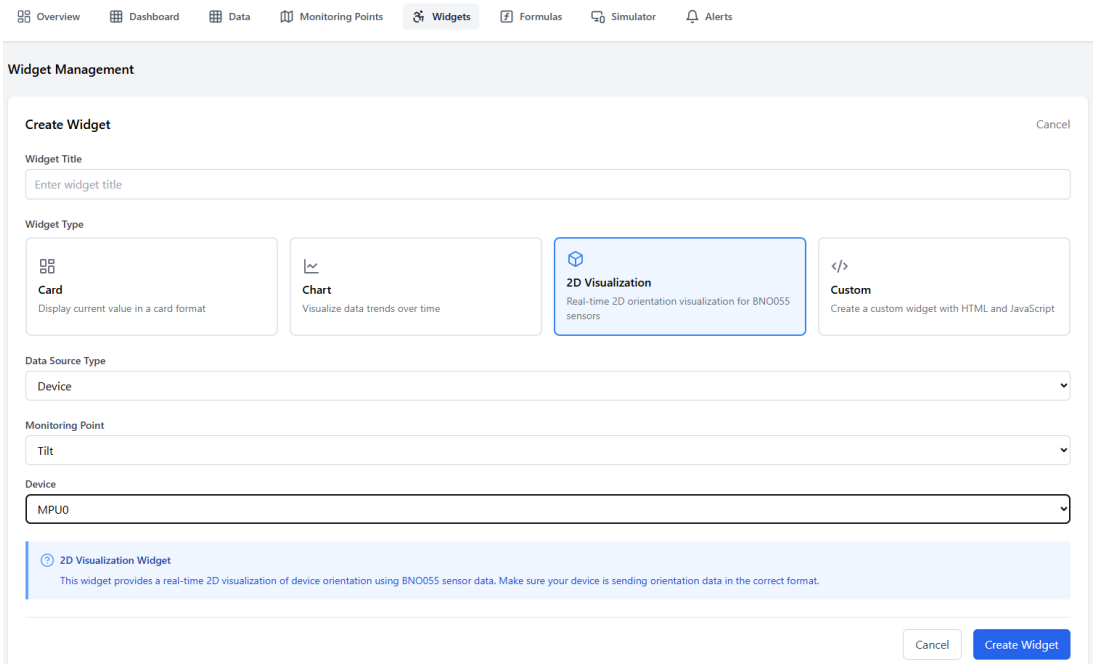

Figura 4.11: Interface de configuração e adição de Widgets personalizados para visualização de dados

4.4.6 Dashboard de Monitoramento

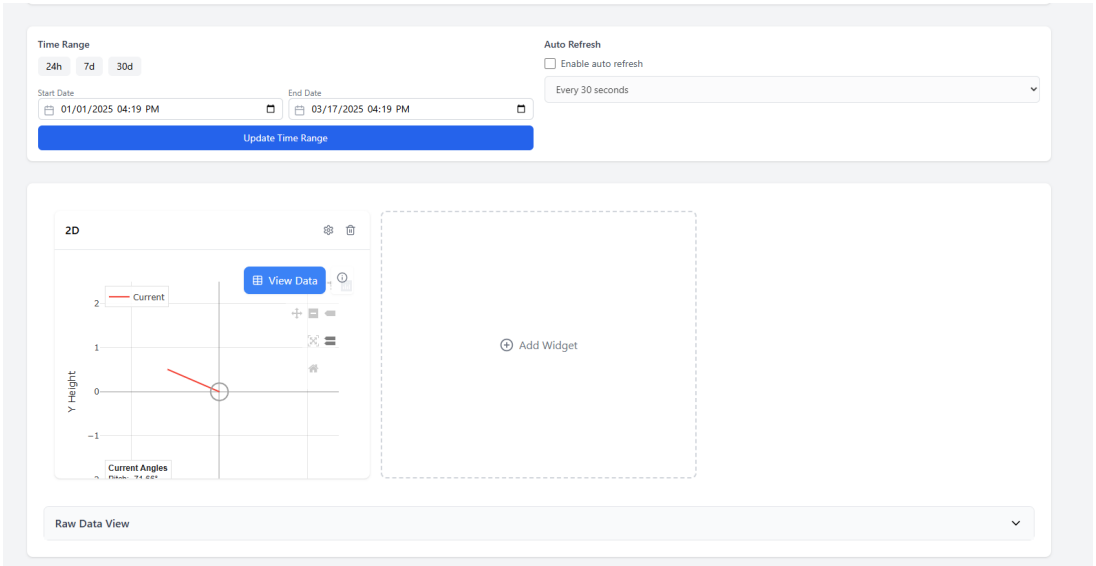

Figura 4.12: Dashboard principal mostrando inclinômetro 2D em tempo real e métricas de monitoramento

4.5 Considerações Finais

O desenvolvimento da plataforma EdgeWidgets resultou em contribuições significativas para o estado da arte em sistemas IoT de monitoramento ambiental. A contribuição arquitetural através da arquitetura hierárquica de cinco níveis compreendendo projetos, pontos de monitoramento, dispositivos, widgets e dashboards demonstrou eficácia na organização escalável de sistemas complexos, superando limitações de fragmentação identificadas na literatura. O desenvolvimento de arquitetura modular para monitoramento ambiental foi alcançado através da implementação e validação da arquitetura hierárquica.

As aplicações imediatas abrangem monitoramento de estabilidade de encostas em áreas de risco, sistemas de alerta precoce para deslizamentos, monitoramento estrutural de edificações, e redes de sensoriamento ambiental distribuído. O potencial de extensão contempla adaptação para outros tipos de sensores incluindo qualidade do ar e nível hídrico, integração com redes LoRaWAN para maior alcance, implementação de algoritmos de machine learning edge, e expansão para protocolos adicionais como CoAP e WebSocket. As validações experimentais futuras abrangem testes de campo através de implementação em condições limitadas de conectividade.

5

Estudos de Caso

5.1 Estudo de Caso 1: Estimação de Risco de Deslizamentos Baseada em RSSI de Redes 5G

5.1.1 Introdução

Este estudo de caso[1] apresenta a aplicação de redes 5G como infraestrutura de sensoriamento distribuído para monitoramento de riscos ambientais, especificamente focando na detecção precoce de deslizamentos através da análise de variações no RSSI (*Received Signal Strength Indication*). A abordagem proposta representa uma inovação significativa ao utilizar infraestrutura de telecomunicações existente como sensor ambiental indireto, eliminando a necessidade de *deployment* de sensores dedicados em áreas de risco. O estudo fundamenta-se na correlação entre precipitação pluviométrica, variações no sinal de redes móveis e ocorrência de deslizamentos. Tradicionalmente, o monitoramento de deslizamentos baseia-se em análises pluviométricas diretas através de estações meteorológicas ou dados satelitais, que podem apresentar limitações de cobertura temporal e espacial. A proposta de utilizar variações de RSSI em redes 5G oferece uma alternativa economicamente viável e tecnicamente robusta para regiões com infraestrutura limitada. A arquitetura do sistema integra conceitos avançados de *Software-Defined Networks* (SDN) para coleta, processamento e análise de dados em tempo real, permitindo a geração de alertas automatizados para comunidades em áreas de risco. Esta abordagem representa um avanço significativo na aplicação de tecnologias 5G para soluções de IoT ambiental, demonstrando como infraestruturas de telecomunicações podem ser reutilizadas para aplicações de monitoramento de desastres naturais.

[1] Baseado no trabalho publicado em (OLIVEIRA et al., 2024) e desenvolvido como parte desta dissertação de mestrado.

5.1.2 Detalhamento do Ambiente Experimental

O estudo foi conduzido utilizando dados históricos da Nova Zelândia, região selecionada devido à alta incidência de deslizamentos e disponibilidade de dados confiáveis. A área de análise compreende coordenadas entre -47.0° a -34.0° de latitude e 167.0° a 179.0° de longitude, cobrindo aproximadamente 1.440.000 km² com topografia diversificada e condições climáticas variadas. Foram analisados 21 eventos de deslizamento ocorridos entre 2013 e 2019, registrados no *Global Landslide Catalog* da NASA (NASA Goddard Space Flight Center, 2015). Esta base de dados fornece informações georreferenciadas sobre localização, data e características dos eventos, permitindo correlação temporal e espacial com dados de infraestrutura de telecomunicações. Os dados de torres de telecomunicações foram obtidos através da base OpenCellID (OpenCellID Project, 2024), que mantém registro global distribuído de estações base móveis. Foram identificadas torres 5G (gNodeB) na região de estudo, com densidade variável entre áreas urbanas e rurais. Para cada evento de deslizamento, foram selecionadas torres dentro de raio de 1 km do local do evento, critério estabelecido para garantir correlação espacial significativa entre variações de sinal e condições meteorológicas locais. O sistema utiliza modelagem matemática baseada em equações de propagação de radiofrequência para simular variações de RSSI causadas por precipitação. A implementação segue padrões ITU-R P.838-3 para atenuação atmosférica (ITU-R P.838-3, 2005), considerando frequências típicas de 5G na faixa de 3,5 GHz.

5.1.3 Arquitetura Proposta

A metodologia adotada seguiu abordagem quantitativa comparativa, correlacionando modelos tradicionais baseados em precipitação com modelos baseados em RSSI de redes 5G. O processo experimental compreendeu oito etapas sequenciais, desde definição de parâmetros até validação estatística dos resultados.

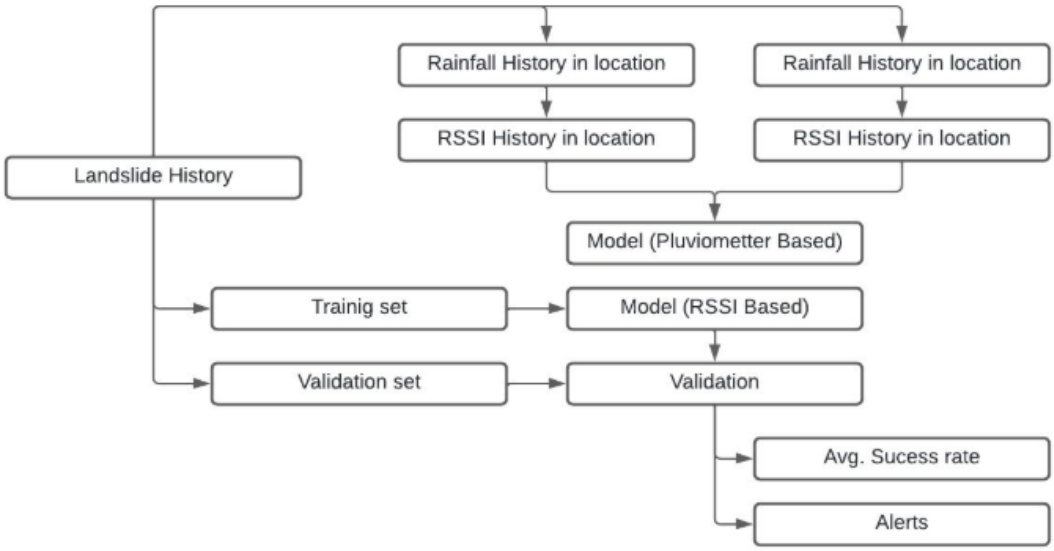

Figura 5.1: Abordagem proposta para correlação entre RSSI e risco de deslizamentos (OLIVEIRA et al., 2024)

O fluxo metodológico seguiu uma sequência estruturada para obtenção e processamento dos dados de deslizamento, torres 5G e RSSI, culminando na aplicação de métodos estatísticos para predição de deslizamentos.

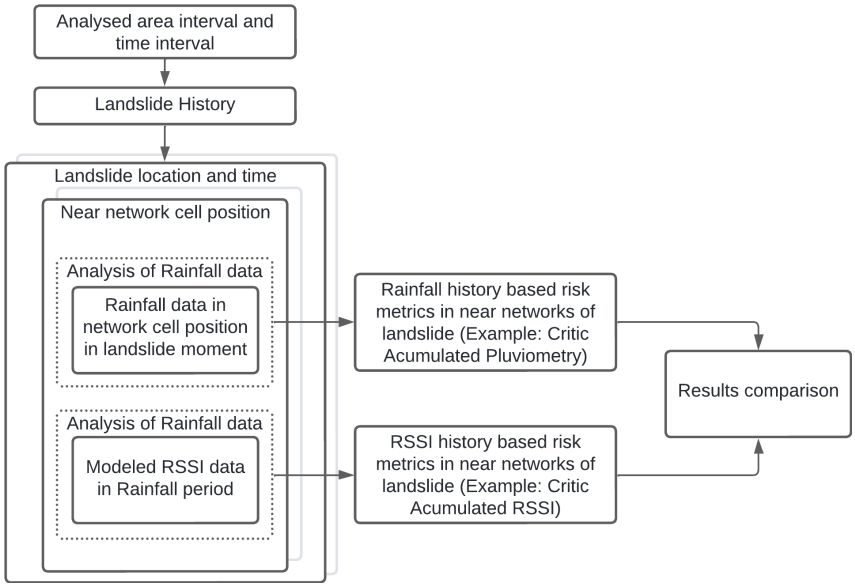

Figura 5.2: Fluxograma detalhado das etapas seguidas para obtenção e processamento de dados de deslizamento, torres 5G e RSSI (OLIVEIRA et al., 2024)

Os parâmetros experimentais foram estabelecidos considerando limitações de dados disponíveis e requisitos de significância estatística. Definiu-se janela temporal de análise de 72 horas anteriores a cada evento de deslizamento, permitindo captura de padrões de precipitação acumulada. O raio de busca de 1 km para torres foi estabelecido para garantir correlação espacial significativa. A coleta de dados utilizou API do *Global Landslide Catalog* da NASA, aplicando

filtros geográficos e temporais específicos. Os dados de precipitação foram obtidos através da NASA Power API (NASA Langley Research Center, 2024), que fornece dados meteorológicos com resolução espacial adequada. A modelagem de RSSI utilizou equações de propagação em espaço livre combinadas com modelos de atenuação atmosférica. A fórmula geral do Free Space Path Loss é definida como ITU-R (2016):

$$FSPL = 20\log_{10}\left(\frac{4\pi d}{\lambda}\right) \text{ dB} \tag{5.1}$$

Para este trabalho, utiliza-se a versão expressa em função da frequência:

$$FSPL = 20\log_{10}(d) + 20\log_{10}(f) - 147{,}55 \text{ dB} \tag{5.2}$$

A seguinte equação permite calcular a perda de potência em espaço livre em decibéis, dada determinada frequência e distância (onde d é a distância entre o gNB e o dispositivo em metros e f é a frequência do sinal em Hz). Para os experimentos realizados a distância foi obtida com base na análise das distâncias entre as torres de celular e locais de deslizamento, enquanto a frequência sendo a utilizada em redes 5G. A fórmula possui origem no padrão ITU-R P.525-3 ajustado para unidades metros e hertz.

A atenuação no sinal gerada pela precipitação da chuva foi obtida através do padrão ITU-R P.838-3 (2005). Esse padrão permite calcular a atenuação específica γ_R (em dB/km) em função da taxa de precipitação e da frequência do sinal através da equação referente a atenuação por precipitação:

$$\gamma_R = kR^{\alpha} \tag{5.3}$$

onde R é a taxa de precipitação em mm/h, e k e α são coeficientes que dependem da frequência e da polarização do sinal. A atenuação total A (em dB) em um percurso de distância d (em km) é obtida multiplicando-se a atenuação específica pela distância:

$$A = \gamma_R \times d \quad \text{(dB)} \tag{5.4}$$

Utilizando a seguinte equação para o RSSI Final:

$$RSSI_{final} = RSSI_{inicial} - (A_{horizontal} + A_{vertical}) \tag{5.5}$$

A arquitetura proposta integra torres 5G, redes SDN e aplicações de monitoramento para criar um sistema abrangente de detecção de riscos de deslizamento baseado em variações de RSSI. A arquitetura SDN proposta permite coleta centralizada de dados de RSSI, processamento em tempo real e distribuição de alertas para usuários em áreas de risco. O controlador SDN gerencia o fluxo de dados e executa algoritmos de análise de risco.

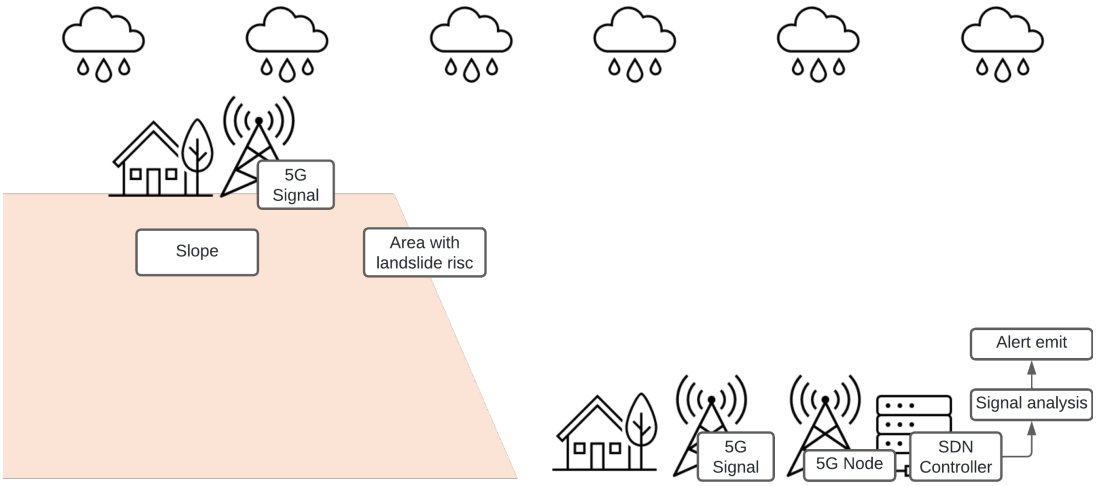

Figura 5.3: Visão geral da arquitetura proposta integrando torres 5G, SDN e aplicações de monitoramento de deslizamentos.

Existem diversos modelos de análise de risco que podem ser utilizados. Para referência. No Brasil, o centro responsável por desastres é o CEMADEN (Centro Nacional de Monitoramentos e Alertas de Desastres Naturais), disponibilizou publicamente informações relacionada a modelagem de risco por meio da plataforma GeoRisk (CAMARINHA, 2025). Onde o risco é calculado com base no índice histórico de deslizamento e acumulado de chuva crítico. Outros modelos de referência podem ser citados como OLIVEIRA et al. (2016) (Análise de incidência de deslizamentos por acumulado de pluviometria com base em múltiplos acumulados para Nova Friburgo), ANDRADE et al. (2023) (Modelo de análise de risco de deslizamento do CEMADEN, SNAKE) e TATIZANA et al. (1987) (Estudo de base com relação a análise estatística entre pluviometria acumulada e incidência de deslizamentos).

Para este trabalho, foram desenvolvidos dois modelos simplificados para o cálculo do risco de deslizamento: um baseado em precipitação (modelo tradicional) e outro baseado em variações de RSSI (modelo proposto). A comparação entre os diferentes mecanismos de cálculo de risco demonstra a viabilidade da abordagem baseada em RSSI como alternativa aos métodos tradicionais.

O modelo baseado em precipitação generalizado foi definido considerando os parâmetros de pluviometria acumulada e média, por serem valores de referência amplamente utilizados em modelagens de pluviometria (embora normalmente variem conforme a localização e o histórico, sendo aqui definidos de forma mais abrangente). A relação entre esses parâmetros é expressa pela Equação 5.6.

$$Risk_{rain} = \frac{P \times P_{accumulated}}{P_{average} \times P_{accumulated_{average}}} \tag{5.6}$$

De forma análoga, o modelo proposto baseado em variações de RSSI considera a diferença e o acumulado do sinal, normalizados por suas médias históricas. Essa relação é representada pela Equação 5.7.

$$Risk_{RSSI} = \frac{RSSI_{diff} \times RSSI_{accumulated}}{RSSI_{average} \times RSSI_{accumulated_{average}}} \tag{5.7}$$

5.1.4 Resultados Experimentais

Para validar a viabilidade da abordagem de medição indireta via rede local, foi desenvolvido um script Python de análise estatística comparativa entre os valores de risco calculados pelo modelo tradicional baseado em pluviometria e pela metodologia RSSI proposta. A análise implementou testes de correlação de Pearson e métricas de erro, conforme apresentado na Tabela 5.1 e visualizado na Figura 5.4.

Tabela 5.1: Estatísticas comparativas entre modelos de risco

Métrica	**Risco Deslizamento**	**Risco RSSI**
Estatísticas Descritivas		
Tamanho da amostra	21	21
Média	0,1546	0,2120
Desvio padrão	0,1643	0,1550
Mediana	0,1044	0,2162
Mínimo	0,0000	0,0006
Máximo	0,4989	0,4708
Análise de Correlação		
Correlação de Pearson	0,8896	
Métricas de Erro		
MSE (Mean Squared Error)	0,008731	

A correlação observada ($r = 0.8896$, $p < 0.000001$) indica que a metodologia RSSI consegue capturar adequadamente os padrões de variabilidade identificados pelo modelo tradicional. Este resultado valida a hipótese de que as variações na intensidade do sinal RSSI podem servir como proxy para as condições pluviométricas locais, possibilitando o sensoriamento indireto através da infraestrutura de rede existente. Ambos os conjuntos atenderam aos pressupostos de normalidade, permitindo a aplicação de testes paramétricos.

A modelagem proposta indica a possibilidade de utilização da infraestrutura de rede existente para sensoriamento, por meio da relação entre as informações já disponíveis mas não utilizadas (RSSI, SNR, entre outras características do sinal de celular) com as condições climáticas locais.

5.1.5 Considerações Finais

O estudo concentrou-se na Nova Zelândia, região com características específicas. A generalização para outras regiões requer validação adicional, considerando diferenças em padrões

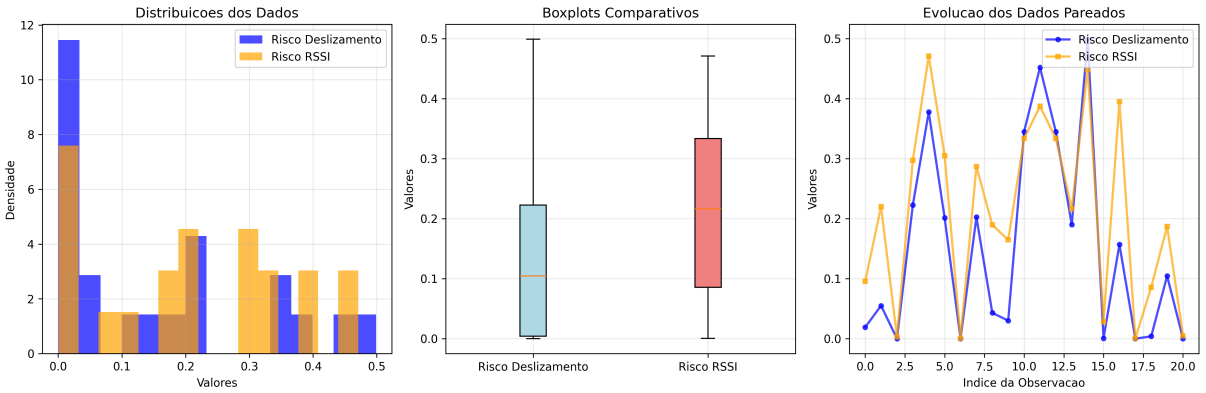

Figura 5.4: Análise estatística comparativa: distribuições, boxplots e evolução temporal dos valores de risco

de precipitação, características do solo, e densidade de infraestrutura de telecomunicações. A janela temporal de análise (2013-2019) pode não capturar variabilidade climática de longo prazo. Validação com séries temporais mais extensas é recomendada para confirmar robustez dos modelos. O estudo baseou-se em modelagem teórica de variações de RSSI, não incluindo medições diretas de campo. A implementação prática requer desenvolvimento de sistemas de coleta de dados em tempo real integrados à infraestrutura de operadoras.

Este estudo de caso demonstrou potencial viabilidade técnica de utilizar variações de RSSI em redes 5G para estimação de risco de deslizamentos, oferecendo alternativa aos métodos tradicionais de monitoramento. A correlação observada (r = 0.8896) entre modelos baseados em precipitação e RSSI confirma potencial da abordagem proposta. A implementação da arquitetura SDN para coleta e processamento de dados representa contribuição significativa para aplicações de redes 5G em IoT ambiental. A capacidade de reutilizar infraestrutura existente oferece novo paradigma para sistemas de alerta precoce. Os resultados estabelecem base sólida para desenvolvimento de sistemas operacionais de monitoramento baseados em RSSI, requerendo validação adicional através de estudos de campo. A metodologia desenvolvida é aplicável a outros tipos de eventos climáticos extremos. Este trabalho, desenvolvido como parte desta dissertação de mestrado, representa avanço significativo na aplicação de infraestruturas de telecomunicações para soluções de monitoramento ambiental distribuído, demonstrando como convergência de tecnologias pode gerar inovações com impacto social e ambiental relevante. Estudos futuros devem incluir validação experimental com medições diretas de RSSI, expansão para outras regiões geográficas, integração com outros tipos de sensores ambientais, e desenvolvimento de algoritmos de *machine learning* para melhoria da precisão preditiva.

5.2 Estudo de Caso 2: Plataforma EdgeWidgets para Monitoramento de Inclinação com Comparação de Protocolos IoT

5.2.1 Introdução

Este segundo estudo de caso[2] apresenta o desenvolvimento e validação experimental da plataforma EdgeWidgets aplicada ao monitoramento de inclinação para detecção precoce de deslizamentos. O foco principal reside na análise comparativa entre protocolos de comunicação HTTP e MQTT em dispositivos IoT de borda, avaliando métricas de latência, confiabilidade e eficiência energética em cenários de monitoramento ambiental. A plataforma EdgeWidgets representa uma solução integrada para sensoriamento ambiental distribuído, combinando hardware de baixo custo baseado em microcontroladores ESP32 com sensores inerciais de alta precisão. O sistema implementa estratégia dual-protocol que permite alternância automática entre HTTP e MQTT baseada em condições de rede, otimizando performance e confiabilidade da transmissão de dados críticos. O estudo fundamenta-se na necessidade de soluções IoT eficientes para monitoramento de riscos geológicos, onde latência de comunicação e confiabilidade de transmissão são fatores críticos para eficácia de sistemas de alerta precoce. A comparação entre protocolos HTTP e MQTT através de metodologia experimental controlada fornece insights valiosos para otimização de arquiteturas IoT em aplicações ambientais.

5.2.2 Detalhamento do Ambiente Experimental

O ambiente experimental foi implementado utilizando dispositivo sensor EdgeWidgets customizado, composto por microcontrolador ESP32 C3 SuperMini com processador RISC-V single-core, conectividade Wi-Fi 802.11 b/g/n na frequência 2,4GHz, interfaces UART, SPI, I2C e GPIO para expansibilidade.

O sistema integra sensor Bosch BNO055 IMU (Inertial Measurement Unit) de 9 graus de liberdade com resolução angular adequada para medições de orientação, taxa de amostragem configurável, interface I2C para comunicação com o microcontrolador, sistema de auto-calibração contínua, temperatura de operação adequada para ambientes externos, e consumo energético otimizado para operação contínua. A estrutura do dispositivo de prototipagem elaborado está disponível na Figura 5.5.

[2]Desenvolvido como parte desta dissertação de mestrado, apresentando validação experimental da plataforma EdgeWidgets.

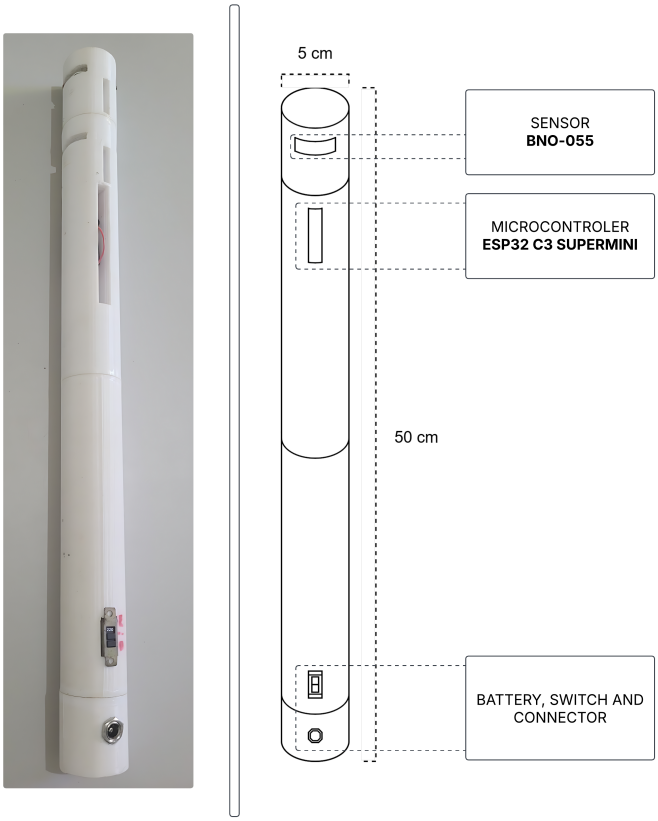

Figura 5.5: Configuração experimental do dispositivo EdgeWidgets mostrando integração entre ESP32, sensor BNO055 e ambiente de teste.

O sistema foi hospedado no *Railway*, com um servidor HTTP e um broker EMQX MQTT, permitindo o recebimento e processamento de dados por ambos os protocolos.

Como resultado deste trabalho, foi montado o circuito com ESP32-C3 SuperMini como microcontrolador e BNO055 como sensor de inclinação. O firmware foi desenvolvido em C++/Arduino, realizando o envio de dados pelos dois protocolos selecionados (HTTP e MQTT). A comunicação com a internet foi realizada configurando conexão Wi-Fi.

5.2.3 Metodologia Experimental

Para o protocolo HTTP, a latência foi medida a partir do timestamp de envio da requisição POST até o momento em que a resposta completa foi recebida do servidor. No caso do MQTT, a latência foi registrada desde o timestamp de publicação da mensagem no tópico principal até o recebimento da confirmação correspondente no tópico de resposta do dispositivo. Cabe destacar que, por se tratar de um protocolo baseado em publicação e subscrição mediado por um broker, o MQTT inclui uma etapa adicional de roteamento e armazenamento intermediário das mensagens, o que deve ser considerado na interpretação dos resultados de latência. As fórmulas utilizadas foram:

$$\text{Latência}_{HTTP} = Tresponse - T_{request} \tag{5.8}$$

$$\text{Latência}_{MQTT} = Tack - T_{publish} \tag{5.9}$$

onde todos os timestamps são coletados utilizando a função 'micros', obtendo assim os microssegundos de execução da aplicação (com base no clock do microprocessador). Os dados foram coletados através de sistema de logging serial do ESP32 com informações estruturadas incluindo timestamp e valores de latência calculados.

5.2.4 Resultados Experimentais

Para validar comparativamente o desempenho dos protocolos HTTP e MQTT, foi desenvolvido um script Python de análise estatística entre as métricas de latência coletadas pelos dispositivos EdgeWidgets. O script completo está disponível no Apêndice B. A análise implementou pré-processamento com remoção de outliers, testes de normalidade Kolmogorov-Smirnov, teste t de Student pareado e análise de correlação, conforme apresentado na Tabela 5.2 e visualizado na Figura 5.6.

Tabela 5.2: Estatísticas comparativas entre protocolos de comunicação IoT

Métrica	**HTTP**	**MQTT**
Estatísticas Descritivas		
Tamanho da amostra	142	142
Média (ms)	1161,25	913,02
Desvio padrão (ms)	116,14	105,69
Mediana (ms)	1149,19	912,20
Mínimo (ms)	929,77	648,32
Máximo (ms)	1478,67	1181,51
Análise de Correlação		
Correlação de Pearson (rr r)	-0,0051	
Métricas de Erro		
MSE (Mean Squared Error)	86226,81	

A comunicação por meio do protocolo HTTP obteve média de 1161,25 ms, enquanto o MQTT apresentou 913,02 ms (latência média 21,4% inferior). A diferença pode ser explicada pelas otimizações implementadas no MQTT, como o uso de conexões persistentes, cabeçalhos mais compactos e uma arquitetura assíncrona baseada em publicação e subscrição, que reduzem a sobrecarga de comunicação e o tempo de transmissão de dados. A correlação observada (r = -0,0051, p = 0,952) indica que os protocolos operam de forma independente, o que é esperado dado que representam abordagens arquiteturais distintas para comunicação IoT. O pré-processamento removeu 8 outliers do HTTP (5,3% da amostra) e 3 outliers do MQTT (2,0% da amostra). Ambos os conjuntos atenderam aos pressupostos de normalidade, permitindo aplicação de testes paramétricos. Os resultados indicam a vantagem do protocolo MQTT para

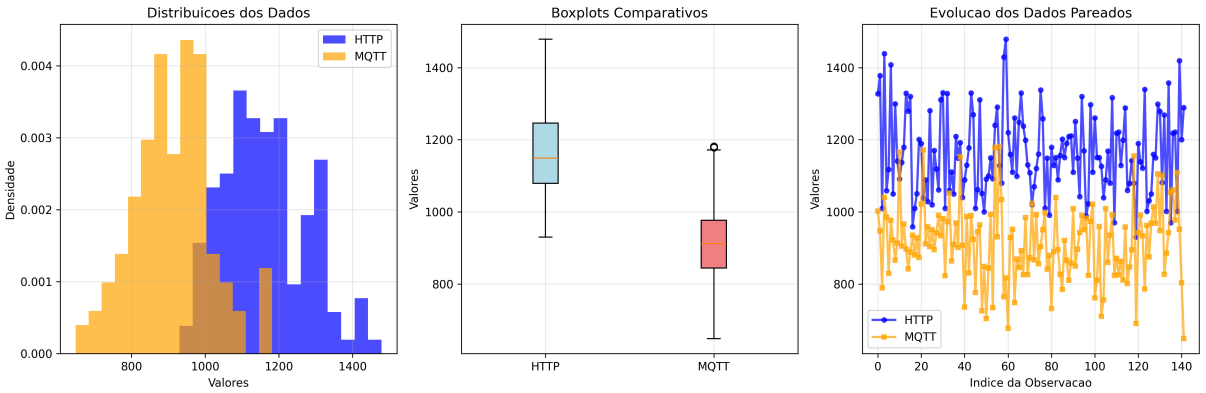

Figura 5.6: Análise estatística comparativa: distribuições de latência, boxplots e evolução temporal para protocolos HTTP e MQTT

aplicações IoT de monitoramento ambiental, apresentando não apenas menor latência média mas também menor variabilidade, características críticas para sistemas de alerta precoce onde consistência temporal é fundamental.

5.2.5 Considerações Finais

As limitações identificadas incluem ambiente controlado que pode não representar totalmente condições de campo extremas, duração limitada dos experimentos, quantidade limitada de dispositivos testados simultaneamente, condições de rede mais estáveis que ambientes remotos reais, e foco em protocolos específicos. Limitações técnicas incluem resolução de timing limitada pela arquitetura do hardware, dependência de conectividade Wi-Fi, capacidade de buffer limitada, e consumo energético ainda otimizável para deployments de longa duração.

Desenvolvimentos futuros incluem integração de sensores adicionais, implementação de algoritmos de machine learning, otimização do consumo energético, expansão para suporte a redes LPWAN, e validação em larga escala com múltiplos dispositivos.

Este estudo de caso aponta a eficácia da plataforma EdgeWidgets para monitoramento ambiental baseado em sensoriamento de inclinação. Os resultados demonstram vantagens significativas do protocolo MQTT sobre HTTP em termos de latência (21,4% menor) e variabilidade, confirmando a validade da estratégia dual-protocol proposta. A metodologia de medição edge-based utilizada permite comparação entre protocolos diminuindo a complexidades de sincronização.

6

Conclusão

O presente trabalho estrutura os desenvolvimentos vinculados a dois estudos de caso, o primeiro relacionado a utilização da rede de 5G local para análise do risco de deslizamento. Se baseando na relação entre pluviometria e o aumento dos riscos e a alteração da força de sinal na rede. Onde as características de rede 5G que permitem maior descentralização da capacidade de análise (por meio de SDN para obtenção de dados em tempo real) permitem a análise de modelos que se utilizem da variação de grandezas como RSSI para comparação. O modelo proposto foi analisado por meio de valores históricos obtidos em fontes públicas e avaliado com base em correlação.

O segundo estudo de caso abordou o desenvolvimento da plataforma EdgeWidgets, assim como sua arquitetura, principais entidades e aplicação em um caso de uso visando a validação de sua utilidade e obtenção de valores para análise. Configurando para tal o ambiente de prototipagem com um módulo de inclinação Tilt visando proximidade com cenários reais de controle para deslizamento de encostas mediante análise da inclinação. Recebendo as informações de inclinação por meio dos protocolos MQTT e HTTP, realizando a análise dos tempos de latência dos envios (assim como métricas adicionais de análise para confirmação da relevância estatística das amostras).

6.1 Trabalhos Futuros e Limitações

Em trabalhos futuros, é possível estender os diferentes protocolos de comunicação utilizados e abordar diferentes tipos de sensores e aplicações de monitoramento, bem como expandir a aplicação da abordagem RSSI para outros tipos de eventos climáticos extremos. Entre os vários usos possíveis desses recursos está a implementação de seleção automática de protocolos em sistemas IoT, a aplicação da solução em serviços disponíveis na internet, e o desenvolvimento de sistemas híbridos que combinem sensoriamento dedicado com reutilização de infraestrutura de telecomunicações. Além da interligação do sistema de monitoramento com ferramentas de análise preditiva e machine learning. Ainda, a expansão do escopo dos protocolos IoT e análise de suas características específicas para cada tipo de aplicação e suas ferramentas.

Desta forma, explorando ainda mais o uso de métricas edge-based e correlações de RSSI para apoiar o desenvolvimento de sistemas de monitoramento.

Para a abordagem de RSSI, trabalhos futuros devem incluir validação experimental com medições diretas de campo, expansão para outras regiões geográficas com diferentes características climáticas e topográficas, e desenvolvimento de algoritmos de machine learning especializados para análise de padrões de RSSI relacionados a diferentes eventos ambientais. A integração com sensores IoT dedicados pode criar sistemas híbridos mais robustos e precisos.

Ainda, pode-se estender o número de protocolos de comunicação conhecidos e realizar os estudos com um número maior de métricas de performance, uma vez que muitas características de comunicação são específicas do hardware e ambiente, tornando a utilização específica e distinta das disponíveis na ferramenta. Além de investigar possíveis aplicabilidades das metodologias apresentadas em ambiente com diferentes sistemas operacionais embarcados. O referido trabalho e as metodologias apresentadas utilizam firmware baseado em Arduino/ESP-IDF, justamente com utilitários e ferramentas do próprio sistema, o que facilita a utilização nos ambientes com as mesmas características, mas pode necessitar adaptações em ambiente com sistemas operacionais diferentes. Portanto, necessitando de algumas modificações e adaptações nesses ambientes.

A implementação de estudos piloto em larga escala é fundamental para validação das premissas teóricas desenvolvidas. A correlação entre RSSI e diferentes fenômenos meteorológicos deve ser investigada para estabelecer biblioteca abrangente de padrões de detecção. Parcerias estratégicas entre academia, operadoras de telecomunicações e agências governamentais podem acelerar a transferência tecnológica dos resultados.

Por fim, uma investigação profunda da correlação direta e indireta das variáveis coletadas no sistema pode auxiliar na extração de características comportamentais e reduzir o uso de variáveis que são correlacionadas. Assim, capturando apenas as características essenciais e mais influentes de cada protocolo e estratégia de sensoriamento. Além de utilizar mais utilitários e ferramentas de monitoramento, implementando os firmwares desenvolvidos para conseguir capturar ainda mais características comportamentais nas diversas situações como ferramentas de análise de rede, ferramentas de monitoramento de conectividade, como análises de qualidade de sinal e ferramentas de monitoramento da aplicação, fazendo a interligação das ferramentas desenvolvidas com diversas outras plataformas de monitoramento e com sistemas de análise de dados de telecomunicações para expansão das capacidades de detecção ambiental.

Apêndice

A

Algoritmo de Análise Estatística

O Algoritmo 1 implementa o teste de normalidade de Kolmogorov-Smirnov, verificando se uma amostra de dados segue distribuição normal.

Algoritmo 1 Teste de Normalidade Kolmogorov-Smirnov

Entrada: *dados*: amostra de dados, α: nível de significância
Saída: *normal*: resultado do teste, D: estatística KS, D_α: valor crítico

TesteNormalidadeKS$dados, \alpha$
1: $\mu \leftarrow$ média$(dados)$
2: $\sigma \leftarrow$ desvio_padrão$(dados)$
3: $n \leftarrow$ tamanho$(dados)$
4: $dados_{ord} \leftarrow$ ordenar$(dados)$
5: $F_n(x) \leftarrow$ calcular_CDF_empírica$(dados_{ord})$
6: $F(x) \leftarrow$ calcular_CDF_teórica$(dados_{ord}, \mu, \sigma)$
7: $D \leftarrow \max |F_n(x) - F(x)|$
8: $D_\alpha \leftarrow 1.36/\sqrt{n}$
9: **se** $D \leq D_\alpha$ **então**
10: **retorna** Verdadeiro$, D, D_\alpha$
11: **senão**
12: **retorna** Falso$, D, D_\alpha$
13: **fim se**

O algoritmo calcula os parâmetros da amostra (linhas 2-4), ordena os dados (linha 5), computa as funções de distribuição cumulativa empírica e teórica (linhas 6-7), determina a estatística D de Kolmogorov-Smirnov (linha 8) e toma a decisão comparando com o valor crítico (linhas 10-14).

O Algoritmo 2 implementa o teste t-Student para amostras pareadas, comparando as médias de dois conjuntos de dados relacionados.

Algoritmo 2 Teste t-Student Pareado

Entrada: X_1, X_2: conjuntos de dados pareados, α: nível de significância

Saída: $significativo$: resultado do teste, t: estatística t, $t_{\alpha/2}$: valor crítico, IC: intervalo de confiança

TesteTPareadoX_1, X_2, α

1: $d_i \leftarrow X_1 - X_2$ {Diferenças pareadas}
2: $n \leftarrow$ tamanho(d)
3: $\bar{d} \leftarrow$ média(d)
4: $s_d \leftarrow$ desvio_padrão(d)
5: $t \leftarrow \bar{d}/(s_d/\sqrt{n})$
6: $df \leftarrow n-1$
7: $t_{\alpha/2} \leftarrow$ valor_crítico_t$(\alpha/2, df)$
8: $IC_{inf} \leftarrow \bar{d} - t_{\alpha/2} \times (s_d/\sqrt{n})$
9: $IC_{sup} \leftarrow \bar{d} + t_{\alpha/2} \times (s_d/\sqrt{n})$
10: **se** $|t| > t_{\alpha/2}$ **então**
11: **retorna** Verdadeiro$, t, t_{\alpha/2}, [IC_{inf}, IC_{sup}]$
12: **senão**
13: **retorna** Falso$, t, t_{\alpha/2}, [IC_{inf}, IC_{sup}]$
14: **fim se**

O algoritmo calcula as diferenças pareadas (linha 2), determina as estatísticas das diferenças (linhas 3-5), computa a estatística t (linha 6), obtém o valor crítico (linha 8), calcula o intervalo de confiança (linhas 9-10) e determina a significância estatística (linhas 11-15).

O Algoritmo 3 coordena todo o processo de análise estatística, integrando os testes de normalidade e t-Student pareado.

Algoritmo 3 Análise Estatística Completa

Entrada: X_1, X_2: conjuntos de dados pareados

Saída: Resultado da análise estatística completa

AnaliseCompletaX_1, X_2

1: $(normal_1, D_1, D_{c1}) \leftarrow$ TesteNormalidadeKS$X_1, 0.05$
2: $(normal_2, D_2, D_{c2}) \leftarrow$ TesteNormalidadeKS$X_2, 0.05$
3: **se** $normal_1 \wedge normal_2$ **então**
4: `imprimir("Ambos conjuntos são normais")`
5: $(signif, t, t_c, IC) \leftarrow$ TesteTPareado$X_1, X_2, 0.05$
6: **se** $signif$ **então**
7: `imprimir("Diferença estatisticamente significativa")`
8: **senão**
9: `imprimir("Não há diferença significativa")`
10: **fim se**
11: `imprimir("Intervalo de Confiança:", IC)`
12: **senão**
13: `imprimir("Dados não são normais. Considere testes não-paramétricos")`
14: **fim se**

O algoritmo principal verifica a normalidade de ambos os conjuntos de dados (linhas 2-3), aplica condicionalmente o teste t-Student se os dados forem normais (linhas 4-11), fornece interpretação dos resultados (linhas 6-10), ou sugere alternativas não-paramétricas caso os pressupostos não sejam atendidos (linha 12).

B

Implementação em Python do Método de Análise

```python
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import pandas as pd

def TesteNormalidadeKS(dados, alpha=0.05):
    """
    Implementa o teste de normalidade Kolmogorov-Smirnov

    Parametros:
    dados: array com os dados da amostra
    alpha: nivel de significancia (padrao 0.05)

    Retorna:
    normal: True se os dados sao normais, False caso contrario
    D: estatistica de Kolmogorov-Smirnov
    D_alpha: valor critico
    """
    # Calcular parametros da amostra
    mu = np.mean(dados)
    sigma = np.std(dados, ddof=1)
    n = len(dados)

    # Ordenar dados para calculo da CDF empirica
    dados_ord = np.sort(dados)

    # Calcular CDF empirica
    cdf_empirica = np.arange(1, n+1) / n

    # Calcular CDF teorica (distribuicao normal)
    cdf_teorica = stats.norm.cdf(dados_ord, mu, sigma)

```

```
33      # Estatistica de Kolmogorov-Smirnov
34      D = np.max(np.abs(cdf_empirica - cdf_teorica))
35
36      # Valor critico para alpha = 0.05
37      D_alpha = 1.36 / np.sqrt(n)
38
39      # Decisao do teste
40      if D <= D_alpha:
41          return True, D, D_alpha
42      else:
43          return False, D, D_alpha
44
45  def TesteTPareado(X1, X2, alpha=0.05):
46      """
47      Implementa o teste t-Student pareado
48
49      Parametros:
50      X1, X2: arrays com os dados pareados
51      alpha: nivel de significancia (padrao 0.05)
52
53      Retorna:
54      significativo: True se ha diferenca significativa
55      t: estatistica t calculada
56      t_alpha_2: valor critico
57      IC: intervalo de confianca [inferior, superior]
58      """
59      # Calcular diferencas pareadas
60      d = X1 - X2
61      n = len(d)
62      d_barra = np.mean(d)
63      s_d = np.std(d, ddof=1)
64
65      # Calcular estatistica t
66      t = d_barra / (s_d / np.sqrt(n))
67      df = n - 1
68
69      # Valor critico para teste bilateral
70      t_alpha_2 = stats.t.ppf(1 - alpha/2, df)
71
72      # Calcular intervalo de confianca
73      margem_erro = t_alpha_2 * (s_d / np.sqrt(n))
74      IC_inf = d_barra - margem_erro
75      IC_sup = d_barra + margem_erro
76
77      # Decisao do teste
78      if abs(t) > t_alpha_2:
79          return True, t, t_alpha_2, [IC_inf, IC_sup]
```

```
80      else:
81          return False, t, t_alpha_2, [IC_inf, IC_sup]
82
83  def AnaliseCompleta(X1, X2):
84      """
85      Executa a analise estatistica completa
86
87      Parametros:
88      X1, X2: arrays com os dados pareados
89
90      Retorna:
91      tupla com resultados dos testes de normalidade e significancia
92      """
93      print("="*60)
94      print("ANALISE ESTATISTICA COMPLETA")
95      print("="*60)
96      print(f"Tamanho da amostra: {len(X1)}")
97      print(f"Media X1: {np.mean(X1):.4f}    {np.std(X1, ddof=1):.4f}")
98      print(f"Media X2: {np.mean(X2):.4f}    {np.std(X2, ddof=1):.4f}")
99      print()
100
101     # Teste de normalidade para ambos os conjuntos
102     normal_1, D_1, D_c1 = TesteNormalidadeKS(X1, 0.05)
103     normal_2, D_2, D_c2 = TesteNormalidadeKS(X2, 0.05)
104
105     print("-" * 50)
106     print("TESTE DE NORMALIDADE KOLMOGOROV-SMIRNOV")
107     print("-" * 50)
108     print(f"X1: Normal = {normal_1}")
109     print(f"    D = {D_1:.4f}, D_critico = {D_c1:.4f}")
110     print(f"X2: Normal = {normal_2}")
111     print(f"    D = {D_2:.4f}, D_critico = {D_c2:.4f}")
112     print()
113
114     # Verificar se ambos os conjuntos sao normais
115     if normal_1 and normal_2:
116         print("    Ambos conjuntos sao normais")
117         print("  Procedendo com teste t-Student pareado...")
118         print()
119
120         # Realizar teste t pareado
121         signif, t, t_c, IC = TesteTPareado(X1, X2, 0.05)
122
123         print("-" * 40)
124         print("TESTE T-STUDENT PAREADO")
125         print("-" * 40)
126         print(f"Estatistica t: {t:.4f}")
```

```
        print(f"Graus de liberdade: {len(X1) - 1}")
        print(f"Valor critico (bilateral):   {t_c:.4f}")
        print(f"Intervalo de Confianca 95%: [{IC[0]:.4f}, {IC[1]:.4f}]")
        print()

        print("-" * 30)
        print("CONCLUSAO")
        print("-" * 30)

        # Interpretar resultados
        if signif:
            print("    DIFERENCA ESTATISTICAMENTE SIGNIFICATIVA")
            print("  Rejeita-se H0: ha evidencia de diferenca entre os
                grupos")
        else:
            print("    NAO HA DIFERENCA ESTATISTICAMENTE SIGNIFICATIVA")
            print("  Nao se rejeita H0: nao ha evidencia de diferenca
                entre os grupos")

        print(f"Intervalo de Confianca: [{IC[0]:.4f}, {IC[1]:.4f}]")

    else:
        print("    Pelo menos um conjunto nao passou no teste de
            normalidade.")
        print("  Dados nao sao normais. Considere testes nao-parametricos")
        print("  Recomendacoes:")
        print("  - Considere transformacoes (Box-Cox, log, etc.)")
        print("  - Use testes nao-parametricos (Wilcoxon, Mann-Whitney)")
        print("  - Verifique outliers nos dados")

    print("="*60)

    # Retornar resultados
    if normal_1 and normal_2:
        return normal_1, normal_2, signif
    else:
        return normal_1, normal_2, None
```

Listing B.1: Implementação das Funções de Análise Estatística

```
def gerar_graficos(X1, X2, nomes=None, salvar_arquivo=True,
    nome_arquivo='analise_estatistica.png', dpi=300):
    if nomes is None:
        nomes = ['X1', 'X2']

    fig, axes = plt.subplots(1, 3, figsize=(15, 5))

    # 1. Histogramas sobrepostos
```

```
8       axes[0].hist(X1, alpha=0.7, label=nomes[0], bins=15, density=True,
            color='blue')
9       axes[0].hist(X2, alpha=0.7, label=nomes[1], bins=15, density=True,
            color='orange')
10      axes[0].legend()
11      axes[0].set_title('Distribuicoes dos Dados')
12      axes[0].set_xlabel('Valores')
13      axes[0].set_ylabel('Densidade')
14      axes[0].grid(True, alpha=0.3)
15
16      # 2. Boxplots comparativos
17      box_plot = axes[1].boxplot([X1, X2], labels=nomes, patch_artist=True)
18      box_plot['boxes'][0].set_facecolor('lightblue')
19      box_plot['boxes'][1].set_facecolor('lightcoral')
20      axes[1].set_title('Boxplots Comparativos')
21      axes[1].set_ylabel('Valores')
22      axes[1].grid(True, alpha=0.3)
23
24      # 3. Gr fico de linha dos dados pareados
25      indices = np.arange(len(X1))
26      axes[2].plot(indices, X1, 'o-', label=nomes[0], color='blue',
            alpha=0.7, linewidth=2, markersize=4)
27      axes[2].plot(indices, X2, 's-', label=nomes[1], color='orange',
            alpha=0.7, linewidth=2, markersize=4)
28      axes[2].legend()
29      axes[2].set_title('Evolucao dos Dados Pareados')
30      axes[2].set_xlabel('Indice da Observacao')
31      axes[2].set_ylabel('Valores')
32      axes[2].grid(True, alpha=0.3)
33
34      plt.tight_layout()
35
36      # Salvar gr fico em arquivo se solicitado
37      if salvar_arquivo:
38          plt.savefig(nome_arquivo, dpi=dpi, bbox_inches='tight',
39                      facecolor='white', edgecolor='none')
40          print(f"\ n   Gr fico salvo como: {nome_arquivo}")
41          print(f"  Resolu  o: {dpi} DPI")
42          print(f"  Formato: {nome_arquivo.split('.')[-1].upper()}")
43
44      plt.show()
45
46  def calcular_estatisticas_descritivas(X1, X2, nomes=None):
47      """
48      Calcula e exibe estatisticas descritivas detalhadas dos dados
49
50      Parametros:
```

```
    X1, X2: arrays com os dados
    nomes: lista com nomes dos grupos (opcional)
    """
    if nomes is None:
        nomes = ['X1', 'X2']

    print("\n" + "="*50)
    print("ESTATISTICAS DESCRITIVAS")
    print("="*50)

    dados = [X1, X2]
    for dados_grupo, nome in zip(dados, nomes):
        print(f"\n{nome}:")
        print(f"  Tamanho da amostra: {len(dados_grupo)}")
        print(f"  Media: {np.mean(dados_grupo):.4f}")
        print(f"  Mediana: {np.median(dados_grupo):.4f}")
        print(f"  Desvio padrao: {np.std(dados_grupo, ddof=1):.4f}")
        print(f"  Variancia: {np.var(dados_grupo, ddof=1):.4f}")
        print(f"  Minimo: {np.min(dados_grupo):.4f}")
        print(f"  Maximo: {np.max(dados_grupo):.4f}")
        print(f"  Q1 (25%): {np.percentile(dados_grupo, 25):.4f}")
        print(f"  Q3 (75%): {np.percentile(dados_grupo, 75):.4f}")

    # Estatisticas das diferencas
    diferencas = X1 - X2
    print(f"\nDiferencas Pareadas ({nomes[0]} - {nomes[1]}):")
    print(f"  Media das diferencas: {np.mean(diferencas):.4f}")
    print(f"  Desvio padrao das diferencas: {np.std(diferencas,
        ddof=1):.4f}")
    print(f"  Correlacao de Pearson: {np.corrcoef(X1, X2)[0,1]:.4f}")

    # Calculo do MSE
    mse = np.mean((X2 - X1)**2)
    rmse = np.sqrt(mse)
    mae = np.mean(np.abs(X2 - X1))

    print(f"\nMetricas de Erro:")
    print(f"  MSE (Mean Squared Error): {mse:.6f}")
    print(f"  RMSE (Root Mean Squared Error): {rmse:.6f}")
    print(f"  MAE (Mean Absolute Error): {mae:.6f}")
```

Listing B.2: Funções Auxiliares para Análise e Visualização

Referências

ACAR, C.; SHKEL, A. M. Environmentally robust MEMS vibratory gyroscopes for automotive applications. **IEEE sensors journal**, [S.l.], v.9, n.12, p.1895–1906, 2009.

AL-FUQAHA, M. et al. Internet of Things: a survey on enabling technologies, protocols, and applications. **IEEE Communications Surveys & Tutorials**, [S.l.], v.17, n.4, p.2347–2376, 2015.

AMARASINGHE, R. et al. Development of miniaturized 6-axis accelerometer utilizing piezoresistive sensing elements. **Sensors and Actuators A: Physical**, [S.l.], v.134, n.2, p.310–320, 2007.

ANDRADE, M. R. M. de et al. The SNAKE System: cemaden's landslide early warning system (lews) mechanism. **International Journal of Geosciences**, [S.l.], v.14, n.11, p.1146–1159, 2023.

ARMBRUST, M. et al. A view of cloud computing. **Communications of the ACM**, [S.l.], v.53, n.4, p.50–58, 2010.

ASHTON, K. That 'Internet of Things' Thing. **RFID Journal**, [S.l.], June 2009.

ATLAM, H. F. et al. Integration of Cloud Computing with Internet of Things: challenges and open issues. In: IEEE INTERNATIONAL CONFERENCE ON INTERNET OF THINGS (ITHINGS), 2017., Exeter, UK. **Anais...** [S.l.: s.n.], 2017. p.670–675.

ATZORI, L.; IERA, A.; MORABITO, G. The Internet of Things: a survey. **Computer Networks**, [S.l.], v.54, n.15, p.2787–2805, 2010.

BANDEIRA, A. P. N.; COUTINHO, R. Q. Critical rainfall parameters: proposed landslide warning system for the metropolitan region of recife, pe, brazil. **Soils and Rocks**, [S.l.], v.38, n.1, p.27–48, 2015.

BANKS, A.; GUPTA, R. **MQTT version 3.1. 1**. 2014. 89p. v.29.

BANSAL, S.; KUMAR, D. IoT ecosystem: a survey on devices, gateways, operating systems, middleware and communication. **International Journal of Wireless and Microwave Technologies**, [S.l.], v.8, n.3, p.37–55, 2020.

BARLIAN, A. A. et al. Review: semiconductor piezoresistance for microsystems. **Proceedings of the IEEE**, [S.l.], v.97, n.3, p.513–552, 2009.

BARMPOUNAKIS, S. et al. Collision avoidance in 5G using MEC and NFV: the vulnerable road user safety use case. **Computer Networks**, [S.l.], v.172, p.107150, 2020.

BEEBY, S. et al. **MEMS mechanical sensors**. [S.l.]: Artech house, 2004.

BELSHE, M.; PEON, R.; THOMSON, M. **Hypertext transfer protocol version 2 (HTTP/2)**. [S.l.]: RFC 7540, 2015.

BISHOP, M. **HTTP/3**. [S.l.]: RFC 9114, 2022.

BRADEN, R. T.; CLARK, D. D. D.; SHENKER, S. **Integrated Services in the Internet Architecture: an overview**. [S.l.]: RFC Editor, 1994. n.1633. (Request for Comments).

Brasil. Ministério do Desenvolvimento Regional. **Nota Técnica nº 1/2023 – SADJ/VISAM/CCPR**. 2023.

BRITO, G. G. de. **Modelo de monitoramento de deslizamento de encostas por meio de sensor multiparamétrico**. 2013. 146p. Dissertação de Mestrado — Universidade Católica de Pernambuco, Recife.

BRITO, G. G. de. **Sistema de monitoramento em tempo real para aplicação geotécnica em encostas – SMTRAGE**. 2023. Tese de Doutorado — Universidade Federal de Pernambuco.

CAMARINHA, P. I. M. **Manual Técnico - v.02-2025. Metodologia aplicada ao Sistema GeoRisk**. [S.l.]: CEMADEN, 2025. Nota Técnica Nº 64/2025.

CHACHULSKI, S. et al. Trading structure for randomness in wireless opportunistic routing. **ACM SIGCOMM Computer Communication Review**, [S.l.], v.37, n.4, p.169–180, 2007.

COLLISCHONN, W. et al. The exceptional hydrological disaster of April-May 2024 in southern Brazil. **Revista Brasileira de Recursos Hídricos**, [S.l.], 2024.

CRUDEN, D. M.; VARNES, D. J. Landslide types and processes. **Transportation research board, US national academy of sciences**, [S.l.], v.247, p.36–75, 1996.

DIETRICH, W. E.; MONTGOMERY, D. R. **SHALSTAB**: a digital terrain model for mapping shallow landslide potential. Berkeley, CA: National Council for Air and Stream Improvement, 1998. Technical Report.

DUNNICLIFF, J. **Geotechnical instrumentation for monitoring field performance**. [S.l.]: John Wiley & Sons, 1988.

Espressif Systems. **ESP32-C3 Series Datasheet**. [S.l.]: Espressif Systems, 2021.

EUGSTER, P. T. et al. The many faces of publish/subscribe. **ACM computing surveys (CSUR)**, [S.l.], v.35, n.2, p.114–131, 2003.

FIELDING, R. T.; TAYLOR, R. N. Principled design of the modern Web architecture. **ACM Transactions on Internet Technology (TOIT)**, [S.l.], v.2, n.2, p.115–150, 2002.

FOSTER, I.; KESSELMAN, C.; TUECKE, S. The anatomy of the grid: enabling scalable virtual organizations. **The International Journal of High Performance Computing Applications**, [S.l.], v.15, n.3, p.200–222, 2001.

GRANJAL, J.; MONTEIRO, E.; SILVA, J. S. Security for the internet of things: a survey of existing protocols and open research issues. **IEEE Communications Surveys & Tutorials**, [S.l.], v.17, n.3, p.1294–1312, 2015.

GUINARD, D. et al. A resource oriented architecture for the web of things. In: INTERNET OF THINGS (IOT), 2011. **Anais...** [S.l.: s.n.], 2011. p.1–8.

HANKERSON, D.; MENEZES, A. J.; VANSTONE, S. **Guide to elliptic curve cryptography**. [S.l.]: Springer Science & Business Media, 2006.

HONG, H. et al. Spatial prediction of landslide hazard at the Yihuang area (China) using two-class kernel logistic regression, alternating decision tree and support vector machines. **Catena**, [S.l.], v.133, p.266–281, 2015.

HUNKELER, U.; TRUONG, H. L.; STANFORD-CLARK, A. MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks. In: INTERNATIONAL CONFERENCE ON COMMUNICATION SYSTEMS SOFTWARE AND MIDDLEWARE AND WORKSHOPS (COMSWARE'08), 2008. **Anais...** [S.l.: s.n.], 2008. p.791–798.

International Telecommunication Union. **The Internet of Things**. Geneva, Switzerland: [s.n.], 2005.

INTRIERI, E. et al. Design and implementation of a landslide early warning system. **Engineering Geology**, [S.l.], v.147, p.124–136, 2012.

ISMAIL, A. A.; HAMZA, H. S.; KOTB, A. M. Performance Evaluation of Open Source IoT Platforms. In: IEEE GLOBAL CONFERENCE ON INTERNET OF THINGS (GCIOT), 2018., Alexandria, Egypt. **Anais...** [S.l.: s.n.], 2018. p.1–5.

ITU-R. **Recommendation ITU-R P.525-3**: calculation of free-space attenuation. Geneva, Switzerland: International Telecommunication Union, 2016. Accessed: 11 de novembro de 2025.

ITU-R P.838-3. **Specific attenuation model for rain for use in prediction methods**. 2005.

Jara Ochoa, H. J. et al. Comparative Analysis of Power Consumption between MQTT and HTTP Protocols in an IoT Platform Designed and Implemented for Remote Real-Time Monitoring of Long-Term Cold Chain Transport Operations. **Sensors**, [S.l.], v.23, n.10, p.4896, 2023.

JING, Q. et al. Security of the internet of things: perspectives and challenges. **Wireless networks**, [S.l.], v.20, n.8, p.2481–2501, 2014.

KALMAN, R. E. A new approach to linear filtering and prediction problems. **Journal of basic Engineering**, [S.l.], v.82, n.1, p.35–45, 1960.

KÖNIG, T.; KUX, H. J.; CORSI, A. C. Advanced models applied for the elaboration of landslide-prone maps, a review. **International Journal of Geosciences**, [S.l.], v.13, n.3, p.174–198, 2022.

LANGLEY, A. et al. The quic transport protocol: design and internet-scale deployment. In: ACM SPECIAL INTEREST GROUP ON DATA COMMUNICATION. **Proceedings...** [S.l.: s.n.], 2017. p.183–196.

LAU, Y. M. et al. Monitoring of rainfall-induced landslides at Songmao and Lushan, Taiwan, using IoT and big data-based monitoring system. **Landslides**, [S.l.], v.20, p.271–296, 2023.

LEMKIN, M.; BOSER, B. E. A three-axis micromachined accelerometer with a CMOS position-sense interface and digital offset-trim electronics. **IEEE Journal of Solid-State Circuits**, [S.l.], v.34, n.4, p.456–468, 1999.

LIU, L.-L. et al. Dynamic prediction of landslide life expectancy using ensemble system incorporating classical prediction models and machine learning. **Geoscience Frontiers**, [S.l.], v.15, n.2, p.101758, 2024.

LIU, Y. et al. Landslide prediction based on low-cost and sustainable early warning systems with IoT. **Bulletin of Engineering Geology and the Environment**, [S.l.], v.82, p.177, 2023.

MACIEL, P. R. M. **Performance, Reliability, and Availability Evaluation of Computational Systems, Volume I**: performance and background. 1st.ed. Boca Raton, FL: CRC Press, 2023.

MAHONY, R.; HAMEL, T.; PFLIMLIN, J.-M. Nonlinear complementary filters on the special orthogonal group. **IEEE Transactions on automatic control**, [S.l.], v.53, n.5, p.1203–1218, 2008.

MathWorks. **ThingSpeak**: iot analytics platform. Accessed: 2025-10-02, `https://thingspeak.com`.

MILLS, D. L. Internet time synchronization: the network time protocol. **IEEE Transactions on communications**, [S.l.], v.39, n.10, p.1482–1493, 1991.

MISHRA, B.; KERTESZ, A. The Use of MQTT in M2M and IoT Systems: a survey. **IEEE Access**, [S.l.], v.8, p.201071–201086, 2020.

NASA Goddard Space Flight Center. **Global Landslide Catalog**. 2015.

NASA Langley Research Center. **NASA POWER Data Access Viewer**. 2024.

OLIVEIRA, G. V. d. et al. Landslide Risk Estimation Based on 5G Network RSSI. **IEEE Access**, [S.l.], 2024.

OLIVEIRA, N. S. de et al. Correlation between rainfall and landslides in Nova Friburgo, Rio de Janeiro—Brazil: a case study. **Environmental Earth Sciences**, [S.l.], v.75, n.20, p.1358, 2016.

OpenCellID Project. **OpenCellID Database**. 2024.

PASSARO, V. M. et al. Gyroscope technology and applications: a review in the industrial perspective. **Sensors**, [S.l.], v.17, n.10, p.2284, 2017.

PATTERSON, D. A. et al. Recovery oriented computing (ROC): motivation, definition, techniques, and case studies. In: UC BERKELEY COMPUTER SCIENCE TECHNICAL REPORT UCB//CSD-02-1175. **Anais...** [S.l.: s.n.], 2002.

PECORARO, G.; CALVELLO, M.; PICIULLO, L. Monitoring strategies for local landslide early warning systems. **Landslides**, [S.l.], v.16, p.213–231, 2019.

PERERA, C. et al. Context Aware Computing for The Internet of Things: a survey. **IEEE Communications Surveys & Tutorials**, [S.l.], v.16, n.1, p.414–454, 2014.

RAMESH, M.; KUMAR, S. Low cost landslide early warning system using wireless sensor network. , [S.l.], p.1232–1236, 2016.

RAWAT, P. S.; BARTHWAL, A. LANDSLIDE MONITOR: a real-time landslide monitoring system. **Environmental Earth Sciences**, [S.l.], v.83, p.226, 2024.

RICHARDSON, L.; RUBY, S. **RESTful web services**. [S.l.]: O'Reilly Media, Inc., 2008.

ROMAN, R.; ZHOU, J.; LOPEZ, J. On the features and challenges of security and privacy in distributed internet of things. **Computer Networks**, [S.l.], v.57, n.10, p.2266–2279, 2013.

SABATINI, A. M. Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing. **IEEE Transactions on Biomedical Engineering**, [S.l.], v.53, n.7, p.1346–1356, 2006.

SATYANARAYANAN, M. The emergence of edge computing. **Computer**, [S.l.], v.50, n.1, p.30–39, 2017.

SAUKOSKI, M.; AALTONEN, L.; HALONEN, K. A. Zero-rate output and quadrature compensation in vibratory MEMS gyroscopes. **IEEE Sensors Journal**, [S.l.], v.7, n.12, p.1639–1652, 2007.

SHARMA, A. et al. An ensemble learning-based experimental framework for smart landslide detection, monitoring, prediction, and warning in IoT-cloud environment. **Environmental Science and Pollution Research**, [S.l.], v.30, p.122677–122699, 2023.

SHI, W. et al. Edge computing: vision and challenges. **IEEE Internet of Things Journal**, [S.l.], v.3, n.5, p.637–646, 2016.

SICARI, S. et al. Security, privacy and trust in Internet of Things: the road ahead. **Computer Networks**, [S.l.], v.76, p.146–164, 2015.

SIMOES, R. et al. Application of MEMS technology in environmental monitoring systems. **Sensors**, [S.l.], v.23, p.4589, 2023.

SINGH, M. et al. MQTT: a lightweight publish/subscribe protocol for the internet of things. **IEEE Potentials**, [S.l.], v.34, n.6, p.41–46, 2015.

SiteWhere. **SiteWhere**: open source iot application enablement platform. Accessed: 2025-10-02, `https://sitewhere.io`.

STANFORD-CLARK, A.; TRUONG, H. L. MQTT for sensor networks (MQTT-SN) protocol specification. **International business machines (IBM) Corporation version**, [S.l.], v.1, p.2, 2013.

STANKOVIC, J. A. Research directions for the internet of things. **IEEE Internet of Things Journal**, [S.l.], v.1, n.1, p.3–9, 2014.

TATIZANA, C. et al. Análise de correlação entre chuvas e deslizamentos aplicada às encostas da Serra do Mar no município de Cubatão. In: CONGRESSO BRASILEIRO DE GEOLOGIA DE ENGENHARIA, 5., São Paulo. **Anais...** [S.l.: s.n.], 1987. v.2, p.225–236.

THANGAVEL, D. et al. Performance evaluation of MQTT and CoAP via a common middleware. **2014 IEEE ninth international conference on intelligent sensors, sensor networks and information processing (ISSNIP)**, [S.l.], p.1–6, 2014.

ThingsBoard. **ThingsBoard**: open-source iot platform. Accessed: 2025-10-02, `https://thingsboard.io`.

WAN, E. A.; VAN DER MERWE, R. The unscented Kalman filter for nonlinear estimation. In: IEEE 2000 ADAPTIVE SYSTEMS FOR SIGNAL PROCESSING, COMMUNICATIONS, AND CONTROL SYMPOSIUM. **Proceedings...** [S.l.: s.n.], 2000. p.153–158.

WELCH, G.; BISHOP, G. et al. An introduction to the Kalman filter. , [S.l.], 1995.

WOODMAN, O. J. An introduction to inertial navigation. **University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696**, [S.l.], v.14, n.15, p.27, 2007.

World Economic Forum. **This is what climate change costs economies around the world**. 2023.

WU, W.-T. et al. Design and fabrication of a micro accelerometer using LIGA process. **Microsystem technologies**, [S.l.], v.8, n.6, p.374–379, 2002.

WURM, J. et al. Security analysis on consumer and industrial IoT devices. , [S.l.], p.519–524, 2016.

YAZDI, N.; AYAZI, F.; NAJAFI, K. Micromachined inertial sensors. **Proceedings of the IEEE**, [S.l.], v.86, n.8, p.1640–1659, 1998.

YOKOTANI, T.; SASAKI, Y. Comparison with HTTP and MQTT on required network resources for IoT. In: ICCEREC), 2016. **Anais...** [S.l.: s.n.], 2016. p.1–6.

ZARGAR, S. T.; JOSHI, J.; TIPPER, D. A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. **IEEE communications surveys & tutorials**, [S.l.], v.15, n.4, p.2046–2069, 2013.