

UNIVERSIDADE FEDERAL DE PELOTAS
Centro de Desenvolvimento Tecnológico
Programa de Pós-Graduação em Computação



Dissertação

Avaliação de Sensores Virtuais em Aplicações IoT Tolerantes a Falhas

Wagner Loch

Pelotas, 2023

Wagner Loch

Avaliação de Sensores Virtuais em Aplicações IoT Tolerantes a Falhas

Dissertação apresentada ao Programa de Pós-Graduação em Computação do Centro de Desenvolvimento Tecnológico da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Júlio Carlos Balzano de Mattos
Coorientador: Prof. Dr. Rafael Iankowski Soares

Pelotas, 2023

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação na Publicação

L812a Loch, Wagner

Avaliação de sensores virtuais em aplicações IoT tolerantes a falhas / Wagner Loch ; Júlio Carlos Balzano de Mattos, orientador ; Rafael Iankowski Soares, coorientador. — Pelotas, 2023.

108 f. : il.

Dissertação (Mestrado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2023.

1. IoT. 2. Sensores virtuais. 3. Tolerância a falhas. 4. Qualidade do ar. I. Mattos, Júlio Carlos Balzano de, orient. II. Soares, Rafael Iankowski, coorient. III. Título.

CDD : 005

Wagner Loch

Avaliação de Sensores Virtuais em Aplicações IoT Tolerantes a Falhas

Dissertação aprovada, como requisito parcial, para obtenção do grau de Mestre em Ciência da Computação, Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas.

Data da Defesa: 28 de março de 2023

Banca Examinadora:

Prof. Dr. Júlio Carlos Balzano de Mattos (orientador)

Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Rafael Iankowski Soares (Co-orientador)

Doutor em Computação pela Pontifícia Universidade Católica do Rio Grande do Sul.

Prof. Dr. Felipe de Souza Marques

Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Guilherme Tomaschewski Netto

Doutor em Oceanografia Física pela Fundação Universidade Federal do Rio Grande.

Prof. Dr. Leomar Soares da Rosa Jr.

Doutor em Microeletrônica pela Universidade Federal do Rio Grande do Sul.

Dedicado ao meu querido tio Valdir (*In Memoriam*), que me inspirou desde jovem a seguir o caminho da computação e cujo legado de sabedoria continuará a me guiar por toda a minha vida. Gostaria também de dedicar esta obra ao meu querido avô Renato (*In Memoriam*), que foi o grande responsável por proporcionar a todos os seus netos a oportunidade de ter estudos de qualidade. Ambos serão lembrados para sempre.

AGRADECIMENTOS

Gostaria de dedicar algumas palavras de agradecimento nesta dissertação, que é fruto de muita dedicação, trabalho duro e muita ajuda de pessoas importantes em minha vida.

Primeiramente, gostaria de agradecer aos meus orientadores Júlio e Rafael, que estiveram sempre presentes durante todo o processo de elaboração deste trabalho. Sem a sua orientação e apoio, eu não teria chegado até aqui. Agradeço pela paciência, pelos conselhos valiosos, pelas sugestões, críticas e incentivos que me ajudaram a crescer como estudante e pesquisador. Vocês foram fundamentais para que eu pudesse dar o meu melhor neste trabalho e por isso sou muito grato.

Não posso deixar de agradecer também à minha família, que sempre esteve ao meu lado, me apoiando e encorajando em todas as etapas da minha jornada acadêmica. Agradeço à minha mãe pelo amor incondicional e pela educação exemplar que me proporcionou. Agradeço ao meu irmão pelo incentivo e apoio nas horas difíceis.

Por fim, gostaria de agradecer de maneira muito especial à minha companheira, Clarissa. Seu amor, compreensão e paciência foram imprescindíveis durante todo este processo. Você foi minha companheira de estudo, minha ouvinte de todas as horas e minha fonte de inspiração. Obrigado por estar sempre ao meu lado, por me motivar quando eu precisava e por me apoiar em todos os momentos.

A todos vocês que de alguma forma contribuíram para a realização deste trabalho, o meu muito obrigado. Espero que este seja apenas o começo de muitas outras conquistas e que possamos continuar juntos nesta caminhada rumo a novos conhecimentos.

*A tecnologia não é apenas uma ferramenta, mas sim uma
extensão da nossa capacidade de pensar e agir.*

— JOHN NAISBITT

RESUMO

LOCH, Wagner. **Avaliação de Sensores Virtuais em Aplicações IoT Tolerantes a Falhas**. Orientador: Júlio Carlos Balzano de Mattos. 2023. 108 f. Dissertação (Mestrado em Ciência da Computação) – Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2023.

Nos últimos anos, a utilização de dispositivos IoT tem crescido de forma exponencial em diversas áreas, como indústria, saúde, agriculturas e cidades inteligentes. Com a crescente demanda por soluções tecnológicas mais eficientes e integradas, a utilização de sensores virtuais em dispositivos IoT vem se tornando cada vez mais relevante. Os sensores virtuais são uma técnica utilizada para estimar dados de sensores em dispositivos IoT. Essa abordagem permite reduzir custos, complexidade e permitir uma maior flexibilidade e adaptabilidade em diferentes cenários de uso. Além disso, a utilização de sensores virtuais também pode contribuir para a tolerância a falhas em dispositivos IoT. Através da simulação de diferentes cenários de uso, os sensores virtuais podem ser utilizados para testar a eficácia dos sistemas, identificar falhas e corrigi-las. Diante desse cenário, o objetivo deste trabalho é realizar um estudo sobre modelos de sensores virtuais voltados a tolerância a falhas em sistemas IoT. No decorrer do trabalho foi implementado um estudo de caso voltado para cenários em contexto de COVID-19, doença que abalou o mundo e obrigou todos a implementarem métodos de prevenção e controle. Os resultados obtidos mostram que os erros introduzidos pelos diferentes modelos de sensores virtuais investigados indicam que o uso de média móvel é mais preciso que o uso de replicação de sensores. De forma geral, o trabalho oferece uma contribuição para a área de sensores virtuais e aplicações IoT tolerantes a falhas, fornecendo direcionamentos para pesquisadores e profissionais que trabalham com essas tecnologias.

Palavras-chave: IoT. Sensores Virtuais. Tolerância a Falhas. Qualidade do ar.

ABSTRACT

LOCH, Wagner. **Virtual Sensors Evaluation in Fault Tolerants IoT**. Advisor: Júlio Carlos Balzano de Mattos. 2023. 108 f. Dissertation (Masters in Computer Science) – Technology Development Center, Federal University of Pelotas, Pelotas, 2023.

In recent years, the use of IoT devices has grown exponentially in several areas, such as industry, health, agriculture, and smart cities. With the growing demand for more efficient and integrated technological solutions, the use of virtual sensors in IoT devices is becoming increasingly relevant. Virtual sensors are a technique used to estimate data from sensors in IoT devices. This approach reduces costs and complexity, allowing flexibility and adaptability in different usage scenarios. In addition, the use of virtual sensors can also contribute to fault tolerance in IoT devices. By simulating different usage scenarios, virtual sensors can be used to test the effectiveness of systems, identify failures and correct them. Given this scenario, this work aims to study models of virtual sensors aimed at fault tolerance in IoT systems. During the work, a case study was implemented focused on scenarios in the context of COVID-19. This disease shook the world and forced everyone to implement prevention and control methods. Allied with this context, we found that replicating sensors is not necessarily the best fault tolerance method and that each context must be analyzed individually. The results show that the errors introduced by the different models of virtual sensors investigated indicate that moving averages are more accurate than sensor replication. The work contributes to virtual sensors and fault-tolerant IoT applications, providing valuable directions for researchers and professionals working with these technologies.

Keywords: IoT. Virtual Sensors. Fault Tolerance. Air Quality.

LISTA DE FIGURAS

Figura 1	Número de dispositivos IoT conectados ao longo dos anos. (BURHAN et al., 2018)	22
Figura 2	Uma rede de dispositivos IoT conectados na nuvem. Possui a capacidade de compartilhar informações, dados e recursos entre si. Disponível em https://pixabay.com/pt/vectors/rede-iot-internet-das-coisas-782707/	23
Figura 3	Sensores disponíveis no mercado já vem prontos para uso em microcontroladores, facilitando o desenvolvimento de novas soluções.	25
Figura 4	Placa de circuito impresso contendo três sensores para mensurar a qualidade do ar.	25
Figura 5	Exemplo de um Smart Campus e suas aplicações. Disponível em (CEIEC SMART CAMPUS SOLUTION, 2020).	28
Figura 6	Ciclo de vida da aplicação proposta por (ÖZCAN et al., 2017).	31
Figura 7	Diagrama do fluxo de execução para o ensino de precisão. Adaptado de (LIANG, 2020).	32
Figura 8	Interface do aplicativo desenvolvido por (VAN MERODE et al., 2016).	32
Figura 9	Diagrama da arquitetura do sistema de estacionamento inteligente. Disponível em (SIECK; CALPIN; ALMALAG, 2020)	33
Figura 10	Sistema de energia proposta por (MALATJI; KWESI NTSALUBA, 2018a).	34
Figura 11	Exemplo de sensores virtuais. Disponível em (MARTIN; KÜHL; SATZGER, 2021)	40
Figura 12	Fluxograma para obtenção de um modelo de sensor virtual.	42
Figura 13	Diagrama das <i>Strings</i> de busca para trabalhos relacionados com Tolerância a Falhas em sistemas IoT.	45
Figura 14	Número de artigos encontrados nas bases de consulta após aplicação dos critérios de inclusão e exclusão por ano.	46
Figura 15	Diagrama das <i>Strings</i> de busca para Sensores Virtuais.	50
Figura 16	Diagrama do sensor virtual proposto capaz de ser aplicado a diferentes variáveis.	52
Figura 17	Fluxograma para detecção de falha em sensores e correção através de sensor virtual.	54
Figura 18	Sensor de temperatura e umidade DHT11.	62
Figura 19	Dimensões em mm do sensor de temperatura e umidade DHT11.	63
Figura 20	Sensor de gases MQ-135.	64

Figura 21	Modelo de Display LCD para Arduino.	64
Figura 22	Arduino Nano com microcontrolador ATmega 328.	65
Figura 23	Gráfico de concentração de CO2 em PPM no Cenário 1.	67
Figura 24	Gráfico de temperatura em °C no Cenário 1.	68
Figura 25	Gráfico de umidade em % no Cenário 1.	68
Figura 26	Gráfico de concentração de CO2 em PPM no Cenário 2.	69
Figura 27	Gráfico de temperatura em °C no Cenário 2.	70
Figura 28	Gráfico de umidade relativa do ar no Cenário 2.	70
Figura 29	Gráfico de concentração de CO2 em PPM no Cenário 3.	71
Figura 30	Gráfico de temperatura em °C no Cenário 3.	71
Figura 31	Gráfico de umidade relativa do ar no Cenário 3.	72
Figura 32	Gráfico de concentração de CO2 em PPM no Cenário 4.	72
Figura 33	Gráfico de temperatura em °C no Cenário 4.	73
Figura 34	Gráfico de umidade relativa do ar no Cenário 4.	73
Figura 35	Gráfico de concentração de CO2 em PPM no Cenário 5.	74
Figura 36	Gráfico de temperatura em °C no Cenário 5.	75
Figura 37	Gráfico de umidade relativa do ar no Cenário 5.	75
Figura 38	Gráfico de Concentração de CO2 no Ambiente Administrativo.	76
Figura 39	Gráfico da Temperatura no Ambiente Administrativo.	77
Figura 40	Gráfico da Umidade Relativa do Ar no Ambiente Administrativo.	78
Figura 41	Aplicação do Modelo Histórico no sensor de Temperatura.	79
Figura 42	Aplicação do Modelo Histórico no sensor de Umidade.	80
Figura 43	Aplicação do Modelo Histórico no sensor de CO2.	80
Figura 44	Aplicação do Modelo Regressão Não-Linear Polinomial 2ª Ordem usando a média histórica no sensor de Temperatura.	81
Figura 45	Aplicação do Modelo Regressão Não-Linear Polinomial 2ª Ordem usando os máximos e mínimos no sensor de Temperatura.	82
Figura 46	Aplicação do Modelo Regressão Não-Linear Polinomial 2ª Ordem usando o vetor de amostras no sensor de Temperatura.	82
Figura 47	Aplicação do Modelo Regressão Não-Linear Polinomial 2ª Ordem usando a média histórica no sensor de Umidade.	83
Figura 48	Aplicação do Modelo Regressão Não-Linear Polinomial 2ª Ordem usando os máximos e mínimos no sensor de Umidade.	84
Figura 49	Aplicação do Modelo Regressão Não-Linear Polinomial 2ª Ordem usando o vetor de amostras no sensor de Umidade.	84
Figura 50	Aplicação do Modelo de Regressão Não-Linear Polinomial 2ª Ordem usando a média histórica no sensor de CO2.	85
Figura 51	Aplicação do Modelo de Regressão Não-Linear Polinomial 2ª Ordem usando os máximos e mínimos no sensor de CO2.	86
Figura 52	Aplicação do Modelo de Regressão Não-Linear Polinomial 2ª Ordem usando o vetor de amostras no sensor de CO2.	86
Figura 53	Aplicação do Modelo de Média Móvel usando o histórico de Temperatura.	87
Figura 54	Aplicação do Modelo de Média Móvel usando o histórico de Umidade.	87
Figura 55	Aplicação do Modelo de Média Móvel usando o histórico de CO2.	88
Figura 56	Aplicação do Modelo Histórico usando vetor de amostras para corrigir falhas no sensor de temperatura.	89

Figura 57	Aplicação do Modelo Histórico usando vetor de amostras para corrigir falhas no sensor de umidade.	90
Figura 58	Aplicação do Modelo Histórico usando vetor de amostras para corrigir falhas no sensor de CO ₂	90
Figura 59	Aplicação do Modelo de Regressão Não-Linear Polinomial de 2 ^a Ordem para corrigir falhas no sensor de temperatura.	91
Figura 60	Aplicação do Modelo de Regressão Não-Linear Polinomial de 2 ^a Ordem para corrigir falhas no sensor de umidade.	92
Figura 61	Aplicação do Modelo de Regressão Não-Linear Polinomial de 2 ^a Ordem para corrigir falhas no sensor de CO ₂	92
Figura 62	Aplicação do Modelo de Média Móvel para corrigir falhas no sensor de temperatura.	93
Figura 63	Aplicação do Modelo de Média Móvel para corrigir falhas no sensor de umidade.	93
Figura 64	Aplicação do Modelo de Média Móvel para corrigir falhas no sensor de CO ₂	94
Figura 65	Aplicação do Modelo de Peniak no sensor de temperatura.	95
Figura 66	Aplicação do Modelo de Peniak no sensor de umidade.	95
Figura 67	Aplicação do Modelo de Peniak no sensor de CO ₂	96
Figura 68	Aplicação de Falhas no Modelo de Peniak no sensor de temperatura.	96
Figura 69	Aplicação de Falhas no Modelo de Peniak no sensor de umidade.	97
Figura 70	Aplicação de Falhas no Modelo de Peniak no sensor de CO ₂	97

LISTA DE TABELAS

Tabela 1	Diferenças entre campus digital e smart campus. Adaptada de (MUHAMAD et al., 2017)	28
Tabela 2	Principais tecnologias aplicadas em um Smart Campus e referências para trabalhos relacionados.	29
Tabela 3	Principais Aplicações para Smart Campus encontradas nos artigos estudados	30
Tabela 4	Número de trabalhos encontrados em cada base de pesquisa.	46
Tabela 5	Principais trabalhos e suas respectivas respostas para as RQs.	47
Tabela 6	Número de trabalhos encontrados em cada base de pesquisa	50
Tabela 7	Estimativas de acurácia e identificação de falhas com base no erro. Baseado em (CASADO-VARA et al., 2018).	57
Tabela 8	Especificações e características elétricas do sensor DHT11	63
Tabela 9	Especificações e características elétricas do sensor MQ-135	63
Tabela 10	Especificações e características do Display LCD.	64
Tabela 11	Especificações Arduino Nano.	66
Tabela 12	Cenários de testes com períodos de duas horas de duração e leituras a cada minuto.	66
Tabela 13	Correlação entre CO ₂ , Temperatura e Umidade pelo Método de Pearson.	78
Tabela 14	Comparação entre os Modelos utilizando o cálculo do erro quadrático nos cenários de falhas.	91

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CMS	Content Management System
CO ₂	Dióxido de Carbono
COVID	Corona Virus Disease
IA	Inteligência Artificial
IOT	Internet of Things
iOS	iPhone Operating System
JSON	JavaScript Object Notation
LCD	Liquid Crystal Display
MSE	Mean Square Error
NFC	Near Field Communication
OMS	Organização Mundial da Saúde
OPAS	Organização Pan-Americana da Saúde
PPM	Partes por Milhão
PWM	Pulse Width Modulation
QoS	Quality of Service
RFID	Radio-Frequency Identification
STQ	Soma Total dos Quadrados
TIC	Tecnologias da Informação e Comunicação
UFPel	Universidade Federal de Pelotas

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Motivação	18
1.2	Objetivos Gerais e Específicos	19
1.3	Organização da Dissertação	20
2	FUNDAMENTAÇÃO TEÓRICA	21
2.1	Internet das Coisas (IoT)	21
2.2	Computação em Nuvem	22
2.3	Sensores	24
2.4	Tecnologias de comunicação sem fio	24
2.5	Cidades e Campus Inteligentes	26
2.5.1	Exemplos de Aplicações	28
2.5.2	Análise das Aplicações e Considerações	34
2.6	Tolerância a Falhas	35
2.6.1	Conceitos e Classificação	36
2.6.2	Técnicas Aplicadas a Sistemas Tolerantes a Falhas	37
2.7	Sensor Virtual	39
2.8	Considerações	43
3	REVISÃO BIBLIOGRÁFICA	44
3.1	Tolerância a Falhas	44
3.2	Sensores Virtuais	49
4	MODELO PROPOSTO	52
4.1	Verificação dos Sensores	53
4.2	Método de Histórico de Amostras	55
4.3	Método de Regressão Não-Linear	55
4.4	Método de Janela Flutuante	56
4.5	Método de Peniak	58
4.6	Discussão e Comparativos	58
5	RESULTADOS E EXPERIMENTOS	60
5.1	Estudo de caso: Investigação da Qualidade do Ar	60
5.2	Definição de Sensores e Dispositivos	62
5.2.1	Sensor de Temperatura e Umidade DHT11	62
5.2.2	Sensor de Gases MQ-135	62
5.2.3	Display LCD	63
5.2.4	Arduino Nano	64

5.3	Definição do Cenário 1 - Ambiente Residencial	66
5.3.1	Cenário 1	67
5.3.2	Cenário 2	69
5.3.3	Cenário 3	70
5.3.4	Cenário 4	72
5.3.5	Cenário 5	74
5.3.6	Considerações Finais	76
5.4	Definição do Cenário 2 - Ambiente Administrativo	76
5.4.1	Avaliação da Correlação entre as Variáveis Monitoradas	77
5.5	Estudo de Caso do Projeto de Sensor Virtual	78
5.5.1	Modelo de Histórico de Amostras	79
5.5.2	Modelo de Regressão Não-Linear (Polinomial de 2ª Ordem)	79
5.5.3	Modelo de Média Móvel	83
5.6	Avaliação de Falhas	85
5.6.1	Experimentos com o Modelo de Histórico de Amostras	85
5.6.2	Experimentos com o Modelo de Regressão Não-Linear	86
5.6.3	Experimentos com o Modelo de Média Móvel	87
5.7	Modelo com Método de Peniak	88
5.7.1	Inserindo Falhas no Método de Peniak	88
5.8	Avaliações de Erros dos Modelos	89
5.9	Considerações Sobre a Temperatura e Umidade	91
5.10	Considerações Sobre a Concentração de CO2	94
6	CONCLUSÕES	98
	REFERÊNCIAS	100

1 INTRODUÇÃO

A evolução das sociedades ocorreu através da incessante busca por novos meios de produção e pela troca de conhecimento por meio de interações ocorridas por meio de redes de relacionamentos existentes ao longo do desenvolvimento das civilizações (SOUZA; CARVALHO, 2018). Com a evolução da tecnologia e das revoluções dos meios de produção surgiram novas possibilidades de comunicação, entre elas destaca-se desde as últimas décadas a Internet e as redes sem fio de comunicação que permitem que dispositivos computacionais móveis tais como smartphones, tablets e notebooks troquem informações geograficamente separados pelo mundo inteiro. O avanço das redes 5G (CHETTRI; BERA, 2020), e a expectativa para uma nova geração com a rede 6G, popularizou o uso das redes sem fio, o que proporcionou uma maior mobilidade e banda de dados dedicada para dispositivos a fim de garantir maior praticidade aos usuários (SOUZA; SILVA; GUIMARÃES, 2009).

Estas tecnologias tem impulsionado a expansão da Internet das Coisas (do inglês, *Internet of Things* - IoT) permitindo que objetos tenham capacidade de produzir informações, transmiti-las e tomar decisões por meio de sistemas microcontrolados, sensores, atuadores e etiquetas eletrônicas (RFIDs) embarcados em objetos interconectados a outros sistemas computacionais locais ou remotos por meio da Internet. A IoT vem sendo objeto de grande atenção de governos e iniciativas privadas (MAGRANI, 2018) onde é constantemente utilizada para solucionar problemas, tais como redução de energia elétrica, assistência domiciliar, segurança entre outras aplicações. É cada vez mais comum o uso de objetos para o monitoramento de ambientes físicos por meio de sensores e transformar tais informações em modelos a fim de reproduzir um ambiente físico em ambiente virtual para a simulação do comportamento de ambientes em situações extremas, conceito este conhecido como gêmeo digital (MIHAI et al., 2022), (ENEYEW; CAPRETZ; BITSUAMLAK, 2022). Segundo (ATZORI; IERA; MORABITO, 2010), podemos classificar a IoT de acordo com três visões. A primeira visão é orientada à internet, a segunda à objetos genéricos e a terceira à visão semântica. Outra característica que podemos mencionar para dispositivos IoT é a sua ubiquidade, ou seja, a capacidade destes dispositivos estarem conectados e mistura-

dos ao ambiente de uma forma não invasiva.

A disseminação vertiginosa do uso da IoT traz por consequência alguns problemas computacionais similares aos da computação tradicional como, por exemplo, distribuição de tarefas em elementos computacionais pertencentes a um mesmo sistema e seus meios de comunicação, a escalabilidade do sistema e o controle quando ocorrerem falhas. Diferentes soluções são propostas na literatura para sistemas IoT a fim de torna-los tolerantes a falhas, onde, grande parte das soluções encontradas na literatura é dedicada a soluções para Computação em Nuvem (*Cloud Computing*) e a Computação em Névoa (*Fog Computing*) (KUMARI; KAUR, 2021).

Assim como em diversas áreas de aplicação, a IoT vem sendo utilizada na área da saúde. Neste sentido, aplicações que vem crescendo significativamente são ferramentas que auxiliem no combate e controle de pandemias, como a do Coronavírus SarsCov-2 (COVID-19) que obrigou a todos a adaptarem seus hábitos de convivência. Escolas e Universidades precisaram se reinventar, levando as metodologias de ensino para o modo remoto. Através de diversos estudos, foi destacado que cerca de 15-45% das transmissões de COVID-19 ocorrem através de pessoas assintomáticas (VERMUND; PITZER, 2020), ou seja, pessoas que não apresentam sintomas e que possivelmente não sabem que estão com a doença.

Pesquisas recentes também indicam que os vírus podem permanecer suspensos em aerossóis por horas, tornando assim fundamental o uso de máscaras em ambientes fechados ou com pouca ventilação (STARIOLO; SOBREIRO et al., 2020). Estas pesquisas evidenciam o quão frágeis e suscetíveis para a propagação de vírus e bactérias podem ser os ambientes acadêmicos e escolares.

Assim como em ambientes educacionais, os ambientes de trabalho também lidam com uma grande concentração de pessoas, tornando estes locais propícios para a propagação e contaminação por doenças infectocontagiosas, tal como o vírus causador da doença COVID-19. Dadas estas condições, estes ambientes podem ser considerados insalubres para o trabalho devido a aglomerações excessivas (SILVA DIAS; BÔAS; LUCENA, 2020).

A OMS (Organização Mundial da Saúde) e especialistas em saúde recomendam a restrição do número de pessoas bem como um correto distanciamento social para reduzir a transmissão dos vírus através do contato social (AQUINO et al., 2020). É possível que este distanciamento seja diferenciado levando em consideração o tipo de ambiente em que o indivíduo está inserido.

1.1 Motivação

Na literatura existe uma concentração de trabalhos buscando soluções para sistemas computacionais tolerantes a falhas hospedados em nuvens computacionais (MU-

DASSAR; ZHAI; LEJIAN, 2022),(REHMAN; AGUIAR; BARRACA, 2022), (RAY et al., 2022). No entanto, o acesso a tais recursos não é trivial por serem serviços privados e de alto custo. Aplicações IoT podem envolver vários níveis de abstração em um projeto. Desde o nível mais baixo de abstração envolvendo soluções para aquisição de sinais de sensores, controle de atuadores e infraestrutura de comunicação de objetos até os níveis mais altos de abstração onde o processamento da aplicação pode ocorrer em um servidor de borda (em inglês, *Edge Computing*), em um nevoeiro ou mesmo na nuvem computacional, existe um grande espaço de projeto para os projetistas. Este trabalho concentra-se nos níveis mais baixos de abstração de um sistema IoT e investiga soluções tolerantes a falhas aplicadas a estes níveis de projeto, ou seja, aos dispositivos sensores da aplicação mais precisamente.

Com a crescente demanda por soluções tecnológicas mais eficientes e integradas, a utilização de sensores virtuais em dispositivos IoT vem se tornando cada vez mais relevante (MARTIN; KÜHL; SATZGER, 2021) (CRISTALDI et al., 2020). Muitas aplicações são desenvolvidas por fabricantes diversos e dependem do fluxo de informação contínuo mas não operam a rede de sensores em si. Em caso de falha do sensor, o fluxo de informações será interrompido e causará problema na aplicação. Os sensores virtuais são uma técnica utilizada para estimar dados de sensores em dispositivos. Essa abordagem permite reduzir custos, complexidade e permitir uma maior flexibilidade e adaptabilidade em diferentes cenários de uso. Além disso, a utilização de sensores virtuais também pode contribuir para a tolerância a falhas em dispositivos IoT.

1.2 Objetivos Gerais e Específicos

O objetivo geral deste trabalho é avaliar o uso de sensores virtuais como um método de baixo custo computacional para fim de criar um sistema tolerante a falhas.

Como forma de validação, este trabalho também tem o objetivo de construir um protótipo que faça uso de um conjunto de sensores, sendo capaz de monitorar a qualidade do ar em ambientes acadêmicos e de identificar possíveis riscos à saúde, principal motivação em tempos de pandemia do COVID-19, enquanto lida com diferentes falhas durante a sua utilização.

Para alcançar o objetivo geral, temos os seguintes objetivos específicos:

1. Revisar a literatura a fim de obter algoritmos e métodos para tolerar falhas em sistemas IoT;
2. Avaliar o método de sensores virtuais como mecanismos de tolerar falhas em IoT;
3. Realizar testes com diferentes tipos de sensores;

4. Realizar estudos de caso com um sistema proposto para monitorar a qualidade do ar em ambientes acadêmicos.

1.3 Organização da Dissertação

O presente trabalho está organizado da seguinte forma: no Capítulo 2, é apresentado uma fundamentação teórica a respeito dos principais tópicos abordados no decorrer deste trabalho. Desde cidades ou campus inteligentes, tecnologias, locais para aplicações, sensores virtuais e tolerância a falhas. No Capítulo 3 são apresentados os principais trabalhos relacionados com o trabalho em termos de tolerância a falhas e sensores virtuais. Neste Capítulo também identificamos as lacunas dos trabalhos e onde a nossa proposta se encaixa. No Capítulo 4, é apresentado o modelo de sensor virtual proposto e capaz de garantir tolerância a falhas em sistemas IoT. No Capítulo 5 são apresentados os sensores e dispositivos utilizados, os cenários onde o sistema desenvolvido foi aplicado em conjunto com o sistema proposto. Neste Capítulo são abordados os testes com diferentes modelos, simulações de falhas e seus respectivos resultados. No Capítulo 6 são apresentadas as conclusões deste trabalho e perspectivas de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os conceitos importantes para o entendimento do desenvolvimento e da aplicação deste trabalho. Primeiramente são reforçados os conceitos e tendência de IoT, bem como são apresentados os objetivos da Computação em Nuvem e a tecnologia de realidade aumentada explorada em aplicações IoT. Em seguida, revisa-se o Aprendizado de Máquina tão explorado atualmente em diversas áreas da Computação e inclusive em aplicações IoT. A seguir, destaca-se a importância do uso de sensores para monitoramento de ambientes. Por fim, na Seção 2.5 apresenta-se a definição de Cidades e Campus Inteligentes, principal foco de atuação deste trabalho, elencando suas características, serviços e desafios. Nesta seção são abordados os principais trabalhos relacionados com o uso de dispositivos IoT. Em seguida discute-se sobre tolerância a falhas e possíveis soluções usando sensores virtuais.

2.1 Internet das Coisas (IoT)

Dispositivos IoT são dispositivos eletrônicos conectados a uma rede que coletam e compartilham seus dados. Estes aparelhos podem ser usados para monitorar e controlar outros dispositivos eletrônicos, automatizar processos e realizar a coleta de dados para análises. Não é difícil encontrar dispositivos IoT em nosso dia-a-dia. Como exemplo, podemos citar os sensores de presença, lâmpadas inteligentes, câmeras de segurança e até mesmo relógios inteligentes. A conexão destes dispositivos normalmente ocorre através de tecnologias sem fio como Wi-Fi e Bluetooth.

É fato notório que a utilização de sensores e dispositivos IoT vem se tornando cada vez mais comum. No início da década passada havia uma estimativa de haver 12,5 bilhões de dispositivos com a expectativa de um rápido crescimento (SIVARAMAN et al., 2018). Essa expectativa de crescimento se confirmou e o número de dispositivos já passa dos 50 bilhões conforme apresentado na Figura 1. A razão pela qual esse crescimento é tão acentuado se dá pelo fato destes dispositivos proverem conforto e otimizar diversos tipos de trabalhos (BURHAN et al., 2018).

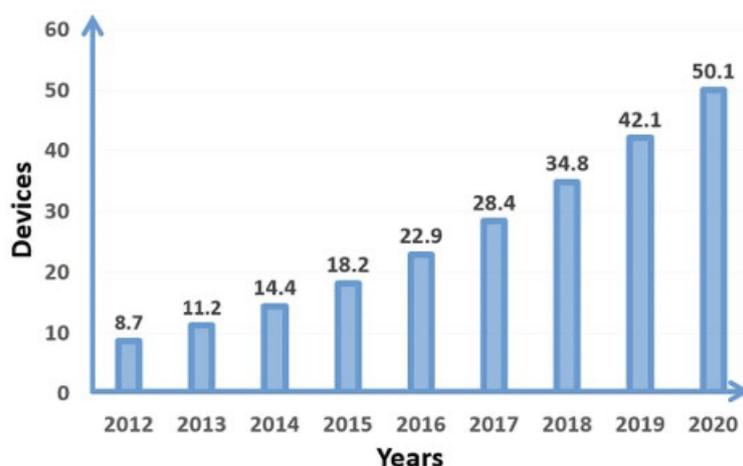


Figura 1 – Número de dispositivos IoT conectados ao longo dos anos. (BURHAN et al., 2018)

Manter estes dispositivos operando sem interrupções é de extrema importância, porém identificar e corrigir rapidamente problemas em dispositivos que podem estar espalhados geograficamente pode ser um enorme desafio (KODESWARAN et al., 2016). A preocupação com os dados gerados também deve ser levada em consideração nos mais diversos cenários. Permanecer com sistemas gerando dados incorretos por um longo período de tempo pode ser catastrófico em análises futuras ou pode acabar permitindo a tomada de decisões equivocadas em sistemas que atuam instantaneamente por exemplo.

A Internet das Coisas, ou simplesmente IoT, tem se tornado cada vez mais comum em pesquisas de ecossistemas *smart*. A sua melhor definição é: "uma rede aberta e abrangente de objetos inteligentes que têm a capacidade de se auto-organizar, compartilhar informações, dados e recursos, reagindo e agindo diante de situações e mudanças no ambiente" (MADAKAM et al., 2015). Sua popularidade se dá pelo baixo custo de implementação e facilidade dos dispositivos. Conectando pessoas através de serviços IoT, podemos facilmente criar aplicações de rastreamento de indivíduos ou objetos em tempo real, por exemplo. Exemplo de uma rede IoT é apresentado na Figura 2.

Segundo (VERSTAEVEL et al., 2018) ainda há muitos desafios a serem resolvidos na utilização de sistemas IoT, dentre os quais, é citado o grande número de dispositivos, a interoperabilidade entre diferentes tecnologias e o grande número de dados coletados a partir de diferentes arquiteturas.

2.2 Computação em Nuvem

Em (WANG et al., 2010), a computação em nuvem é definida como um conjunto de serviços habilitados para rede, fornecendo infraestruturas de computação sob de-

(CARISSIMI, 2015).

A nuvem é uma ótima solução para o tratamento e análise de uma grande quantidade de dados em tempo muitas vezes insignificante, visto ao poder computacional disponível. Através da Computação em Nuvem é possível até mesmo utilizar serviços que façam análise e reconhecimento de imagens em tempo real (LOCH; IANKOWSKI, 2019). As nuvens também possuem soluções para o tratamento de falhas de modo a garantir um serviço em tempo contínuo aos usuários. Uma breve revisão sobre métodos e técnicas usadas para evitar falhas é apresentada na Seção 2.6.

2.3 Sensores

Sistemas inteligentes necessitam de informações para processar. Estas informações podem ser geradas de diferentes maneiras, mas as principais são oriundas de sensores. Sensores são dispositivos capaz de captar informações físicas e transformar em números. Dentre os principais utilizados na construção de sistemas inteligentes podemos citar a detecção por movimentos, temperatura, umidade, luz e até mesmo sensores capazes de detectar vazamentos de água ou gás.

(TSAI; KIONG; SINN, 2018) por exemplo, utiliza em seu trabalho um dispositivo *ESP8266* junto com um sensor de ultrassom para identificar se uma vaga de estacionamento está ocupada ou livre. Para que seja possível a instalação de uma grande quantidade de sensores, a grande maioria utiliza tecnologias móveis para comunicação já que seria inviável a conexão via cabo em locais de difícil acesso.

A Figura 3 apresenta alguns dos diversos sensores disponíveis no mercado que já vem prontos para aplicações em microcontroladores. Estes sensores facilitam e aceleram o desenvolvimento de novas aplicações. A Figura 4 apresenta um sensor tríade para mensurar a qualidade do ar. Nesta pequena placa de apenas 21mm por 19mm estão presentes um sensor de pressão atmosférica, um sensor de dióxido de carbono (CO₂) e um sensor de temperatura e umidade do ar. Este é um exemplo de como podemos encontrar diferentes soluções em uma mesma placa de circuito impresso.

2.4 Tecnologias de comunicação sem fio

Como o objetivo básico de IoT é garantir que objetos comuniquem-se usando qualquer tipo de tecnologia de rede de comunicação entre qualquer tipo de objeto em qualquer lugar, é vital o uso de redes de comunicação sem fio para que viabilize todo e qualquer tipo comunicação entre objetos de um modo geral e dispositivos de computação móvel tais como notebooks, tables, PDAs e smartphones.

Atualmente os meios de comunicação mais comuns existentes para que garan-

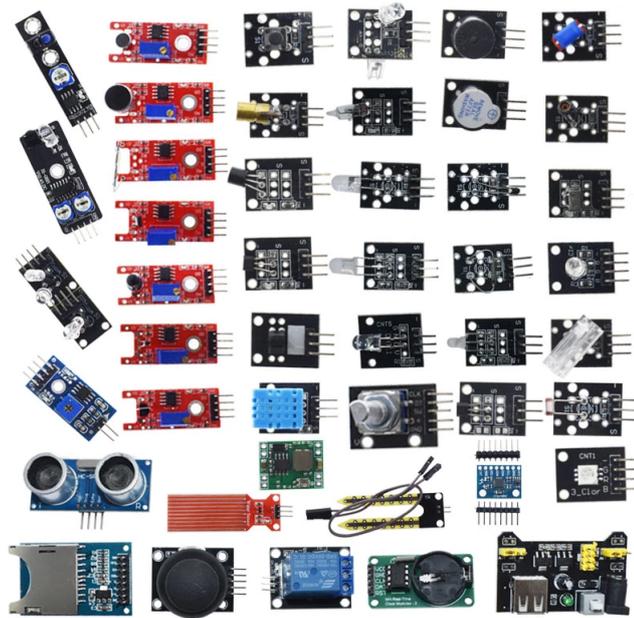


Figura 3 – Sensores disponíveis no mercado já vem prontos para uso em microcontroladores, facilitando o desenvolvimento de novas soluções.

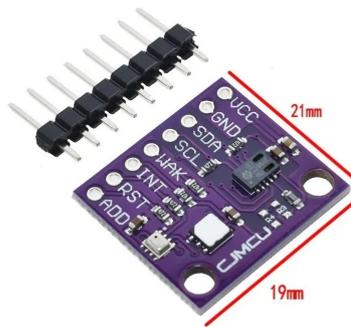


Figura 4 – Placa de circuito impresso contendo três sensores para mensurar a qualidade do ar.

tam esta premissa de IoT são as redes Wifi provendo acesso a Internet por meio de padrões como IEEE 802.11n ou mesmo por meio de redes ponto a ponto formando uma rede pessoal por meio da tecnologia Bluetooth por exemplo (CHANDAN; KHAIRNAR, 2018). Neste contexto é possível também citar a versão de baixo consumo energético do Bluetooth conhecida com BLE, (do inglês, Bluetooth Low Energy) (OLIVEIRA; MATOS, 2016), LoRa (do inglês, Long Range), Zigbee, entre outros.

O emprego de tecnologias de comunicação sem fio também conta com dispositivos como por exemplo sistemas sem contato (*contactless*) como NFC, RFID, QRCode. Segundo (VAN MERODE et al., 2016) estas tecnologias permitem que os dispositivos se comuniquem com o usuário e podem ser empregadas em inúmeras aplicações como revisadas na Seção 2.5.1.

2.5 Cidades e Campus Inteligentes

Esta seção apresenta uma breve definição e contextualização sobre o que são cidades inteligentes e principalmente o campus inteligente, visto que os estudos de casos para a solução aqui proposta ocorre no contexto de campus inteligente realizado no âmbito da Universidade Federal de Pelotas.

A palavra *smart* é um adjetivo em inglês que tem como significado o termo "inteligente". Esta palavra é comumente associada e relacionada com tecnologias que fazem uso de internet para comunicação como em *smartphone* e em *TV Smart* por exemplo. Mas o que de fato, torna um objeto ou algo *smart*? Diversos autores possuem definições diferentes para o conceito daquilo que pode ser considerado *smart*, dentre os quais podemos citar (FERREIRA; ARAÚJO, 2018) que diz que um ambiente inteligente é aquele que é capaz de adquirir e aplicar conhecimento sobre o espaço e em seus habitantes a fim de melhorar a experiência. Já (MUHAMAD et al., 2017) diz que a palavra *smart* é usada para demonstrar que tal objeto é capaz de realizar mais de uma tarefa.

Segundo (STEVENTON; WRIGHT, 2010), um ambiente inteligente é aquele que introduz tecnologias da informação e da comunicação (TIC) e que geram ambientes interativos. A partir deste entendimento, podemos definir uma cidade inteligente (*smart city*) como um ambiente no qual as tecnologias desaparecem à medida que se tornam embutidos nos objetos físicos e nos ambientes nos quais vivemos.

Baseado nos conceitos dos autores citados acima, podemos empregar diversas tecnologias nos ambientes das universidades, criando assim o que chamamos de *smart campus*, *smart university* ou simplesmente campus inteligente.

Existem diversos estudos que relacionam os conceitos de *smart cities* com o ambiente universitário. Podemos associar o campus como uma pequena cidade. As cidades por exemplo, são o futuro da humanidade. Estimativas sugerem que ao final deste século, mais de 80% da população viverá em cidades. Esta rápida transição para ambientes com densa população cria diversos desafios de planejamento, desenvolvimento e principalmente de operações. Segundo (HARRISON; DONNELLY, 2011), as cidades inteligentes provém novos instrumentos que habilitam a observação de sistemas urbanos em um nível micro, isto é, é possível identificar e analisar dados de uma maneira muito mais precisa.

Muitas aplicações de *Smart Cities* podem ser aplicadas em Universidades trazendo consigo diversos benefícios tais como:

- Redução exagerada do consumo de recursos como energia e água, contribuindo assim para a redução da emissão de CO₂.
- Aprimoramento da utilização da infraestrutura existente, evitando a construção de novos prédios.

- Métodos de otimização de transporte individual e coletivo.
- Aprimoramento da interação entre pessoas.
- Aperfeiçoamento de tomadas de decisão através de sistemas.
- Fortalecimento dos métodos de ensino.

O desenvolvimento destas aplicações só é possível em virtude do recente progresso das tecnologias IoT, que tem sua definição apresentada na subseção 2.1. O custo de desenvolvimento, bem como a facilidade de implementação de novas ferramentas vem atraindo cada vez mais pesquisadores para esta área.

Para (HARRISON; DONNELLY, 2011), uma Smart City deve criar um ambiente atrativo tanto para a geração Y (fortemente impactada por avanços tecnológicos) quanto para a geração Z (nativamente digital, ou seja, tiveram contato com as novas tecnologias desde os primeiros dias de vida). O acesso a redes *wireless* deve ser algo pervasivo e a cidade em constante interação com seus cidadãos. As mesmas afirmações podem ser aplicadas e levadas para o ambiente universitário.

Segundo (NIE, 2013) uma universidade que deseja se tornar inteligente passa por 3 estágios: tradicional, *e-campus* e por fim, campus digital ou smart campus. No estágio tradicional, as formas de aprendizado ocorrem da maneira clássica, onde estudantes e professores se encontram de forma presencial num ambiente físico. No estágio *e-campus*, a universidade passa a fornecer e disseminar conteúdo pela internet. Já na fase digital ou *smart*, é quando a universidade consegue se adaptar a diferentes adversidades de forma inteligente com todos os sistemas respondendo de forma integrada.

A Figura 5 apresenta um exemplo de um Smart Campus com algumas aplicações. Percebemos que não é necessário construir um campus do zero para que o mesmo seja classificado como smart. A execução de algumas práticas bem como a interação do ambiente com seus frequentadores já transformam um campus tradicional em smart.

As principais diferenças entre um campus digital e *smart campus* são apresentadas na Tabela 1 adaptada de (MUHAMAD et al., 2017).

O desenvolvimento de uma Cidade ou um Campus Inteligente é inviável sem a utilização de diferentes tecnologias. Ciente disso, existem vários artigos que abordam uma ou várias destas tecnologias. A Tabela 2 apresenta as principais tecnologias utilizadas no desenvolvimento de Cidades e Campus Inteligentes bem como suas referências para trabalhos relacionados.

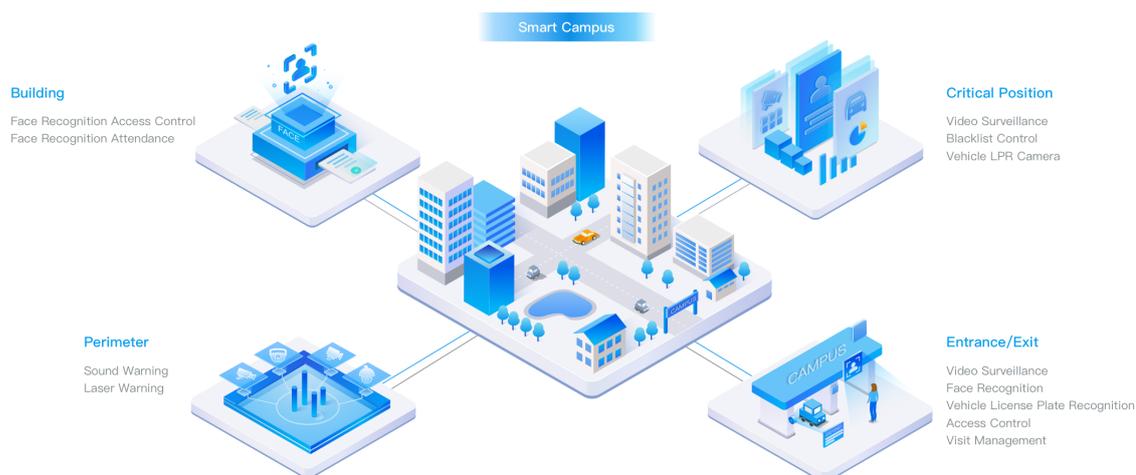


Figura 5 – Exemplo de um Smart Campus e suas aplicações. Disponível em (CEIEC SMART CAMPUS SOLUTION, 2020).

Tabela 1 – Diferenças entre campus digital e smart campus. Adaptada de (MUHAMAD et al., 2017)

	Campus Digital	Smart Campus
Tecnologias Envolvidas	Rede de internet local	RFID, terminais móveis, IoT, computação em nuvem
Aplicações	Aprendizado de forma digital e a distância, biblioteca digital	Sistema inteligente de sensores
Sistema de Gerenciamento	Isolado	Sistema compartilhado e inteligente

2.5.1 Exemplos de Aplicações

Inúmeras aplicações podem ser empregadas num ambiente de um *smart campus* ou *smart city*, para ilustrar, alguns serão apresentados ao longo desta seção. A Tabela 3 apresenta as principais referências destas aplicações encontradas na literatura e são categorizadas pelo domínio em que estão inseridas.

Uso de Realidade Aumentada para Urbanização

O uso de aplicações que fazem uso da realidade aumentada para o planejamento da urbanização de *smart campus* é o interesse do trabalho de (ÖZCAN et al., 2017). O foco principal deste trabalho é adaptar características de *smart cities* para o campus. Um protótipo foi desenvolvido e nele técnicas de detecção de imagem são usadas para identificar lugares. O ciclo de como funciona o procedimento é apresentado na Figura 6.

O usuário é responsável por capturar uma imagem com seu dispositivo através de um software, a imagem é então submetida para um servidor responsável por processar

Tabela 2 – Principais tecnologias aplicadas em um Smart Campus e referências para trabalhos relacionados.

Tecnologia	Referências
Internet das Coisas (IoT)	(VERSTAEVEL et al., 2018), (HOSSAIN; DAS; RASHED, 2019), (AGARWAL; GVV; AGARWAL, 2020), (ZHAMANOV et al., 2017), (SIECK; CALPIN; ALMALAG, 2020), (LIHONG, 2020), (XU; WANG; YU, 2018), (MUHAMMAD et al., 2020), (IQBAL et al., 2019), (TSAI; KIONG; SINN, 2018)
Computação em Nuvem, Big Data	(LIHONG, 2020), (XU; WANG; YU, 2018), (AL NUAIMI et al., 2015), (YANG et al., 2020), (BIBRI, 2019), (TAO et al., 2019)
Inteligência Artificial (IA), Aprendizado de Máquina	(LIANG, 2020), (SIECK; CALPIN; ALMALAG, 2020), (CHIU; LIU; WANG, 2018), (MUHAMMAD et al., 2020), (IQBAL et al., 2018)
Realidade Virtual/Aumentada	(ÖZCAN et al., 2017), (LIU et al., 2017)
Sensores (movimento, temperatura, umidade, luz, energia, biometria)	(LAZAROIU et al., 2015), (MALATJI; KWESINTSALUBA, 2018b), (RUSLI; HALIM, 2019), (CHIU; LIU; WANG, 2018), (TSAI; KIONG; SINN, 2018), (RASHMI; ASHWIN; GUDDETI, 2020), (LIU; LIU; SUN, 2016)
Tecnologias Móveis (Smartphones, NFC, QR Code, GPS, Wireless, Bluetooth, RFID)	(VAN MERODE et al., 2016), (LIU et al., 2017), (RUSLI; HALIM, 2019), (FARIAS et al., 2019), (OLIVEIRA et al., 2019), (TSAI; KIONG; SINN, 2018), (LIU; LIU; SUN, 2016)
Gerenciador de Conteúdo (CMS), Serviços Web	(VAN MERODE et al., 2016)

a imagem e retornar a mesma já com as marcações dos edifícios identificados. Como trabalhos futuros o autor se propõe a aprimorar a aplicação a fim de exibir ainda mais informações e estatísticas sobre os lugares que o usuário está inserido.

O uso de realidade aumentada pode ajudar o usuário a se situar e inclusive ser utilizada para navegação de usuários que não estão habituados com o ambiente que estão inseridos.

Gestão Autônoma para dispositivos IoT

O número de dispositivos IoT vem rapidamente crescendo (CANEDO; SKJELLUM, 2016) e segundo (VERSTAEVEL et al., 2018) em seu trabalho, há mais de 15 bilhões de dispositivos ativos no mundo.

Lidar com uma grande quantidade de dispositivos é um desafio e como solução, é proposta uma arquitetura autônoma para o gerenciamento de dispositivos IoT. Esta

Tabela 3 – Principais Aplicações para Smart Campus encontradas nos artigos estudados

Domínio	Aplicações
Aprendizado	Uso de IA para Ensino de Precisão (LIANG, 2020) Uso de Dispositivos IoT em Aulas Invertidas (ZHAMANOV et al., 2017)
Social	Aplicativo de Interação entre Frequentadores do Campus Baseado na Localização (VAN MERODE et al., 2016)
Administração	Aplicação Móvel para Emitir Notificações Inteligentes com Base na Localização (RUSLI; HALIM, 2019)
Saúde	
Gestão	Uso de Realidade Aumentada para Urbanização (ÖZCAN et al., 2017) Gestão Autônoma para dispositivos IoT (VERSTAEVEL et al., 2018) Uso de <i>Machine Vision</i> e IoT para Gerenciamento Inteligente de Estacionamento (SIECK; CALPIN; ALMALAG, 2020)
Sustentabilidade	Geração Inteligente de Energia (MALATJI; KWESI NTSALUBA, 2018a)

arquitetura, segundo o autor, está sendo implementada em duas universidades, sendo uma delas na França e outra na Austrália.

Uso de IA para Ensino de Precisão

Segundo (MATOS et al., 2020), o ensino de precisão começou em 1965 em salas de aula especiais para crianças com desenvolvimento incomum em um centro de reabilitação da Universidade de Kansas. No início dos anos 1970, o ensino de precisão foi também aplicado em crianças com desenvolvimento normal. Nas salas de aula, os estudantes são responsáveis pelo registro e monitoramento das frequências de seus próprios comportamentos, em um formato de gráfico padronizado. O monitoramento contínuo é o aspecto mais crucial dessa forma de ensino.

O uso de Inteligência Artificial (IA) para promover o ensino de precisão é um dos focos apresentados por (LIANG, 2020). O objetivo é identificar, registrar, categorizar e integrar comportamentos de aprendizagem para ajudar os estudantes a se concentrarem e ajustar as técnicas de aprendizado às necessidades do estudante customizando os planos de aula dos professores. A Figura 7 apresenta um diagrama de como a Inteligência Artificial pode auxiliar no ensino de precisão. Percebemos que são coletados diversos dados como comportamentos, desempenhos e processos que são inseridos em algoritmos de IA que otimizam os conteúdos de ensino.

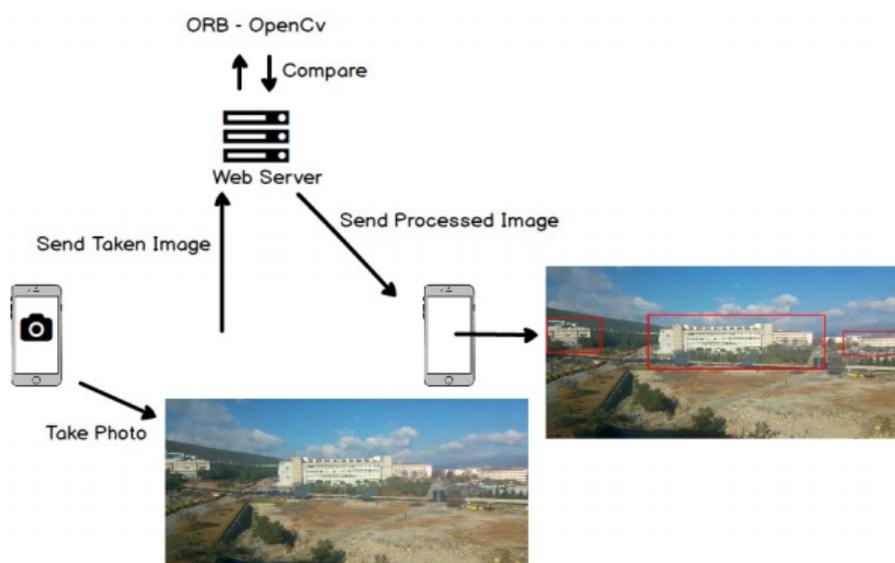


Figura 6 – Ciclo de vida da aplicação proposta por (ÖZCAN et al., 2017).

Uso de Dispositivos IoT em Aulas Invertidas

Segundo (MORAROS et al., 2015), aula invertida é um modelo onde estudantes tem aulas em casa e realizam suas tarefas em sala de aula com a ajuda do professor ou em grupos de colegas. Provas, seminários e outras atividades são realizadas em sala de aula normalmente. Professores preparam vídeo aulas e compartilham com seus alunos que assistem e se preparam antes de participarem das aulas presenciais. Dessa forma, professores conseguem mensurar o conhecimento obtido pelos alunos através das video aulas no momento em que ajuda os alunos a finalizarem as tarefas nos encontros presenciais.

Dessa maneira, o trabalho de (ZHAMANOV et al., 2017) propõe a utilizações de dispositivos IoT para auxiliar nesta modalidade de ensino possibilitando aos estudantes a possibilidade de assistirem às aulas em qualquer local e em qualquer horário, independente do aluno estar em casa, na universidade, metrô ou até mesmo durante o seu almoço.

Segundo o autor, alunos que tiveram contato com dispositivos IoT para o uso de aulas invertidas tiveram melhores resultados de aprendizado do que aqueles que optaram pelo ensino tradicional.

Aplicativo de Interação entre Freqüentadores do Campus Baseado na Localização

A aplicação apresentada por (VAN MERODE et al., 2016) consiste em um aplicativo

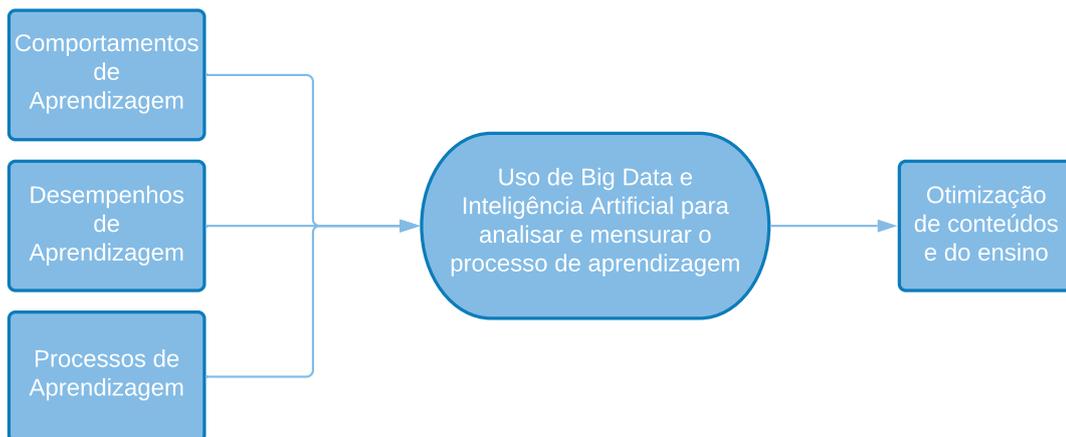


Figura 7 – Diagrama do fluxo de execução para o ensino de precisão. Adaptado de (LIANG, 2020).

para dispositivos *Android* que incentive e facilite a interação entre frequentadores do campus baseado na localização. Também neste aplicativo, informações e avisos a respeito do campus podem rapidamente ser transmitidas. A localização do usuário é fornecida para a aplicação a fim de fornecer conteúdo dinâmico com base na posição.

A Figura 8 apresenta a interface do aplicativo desenvolvido. Com base na localização do usuário é exibida informações e curiosidades sobre os ambientes e há a possibilidade de marcar uma caminhada ou conversa com outros usuários do aplicativo.

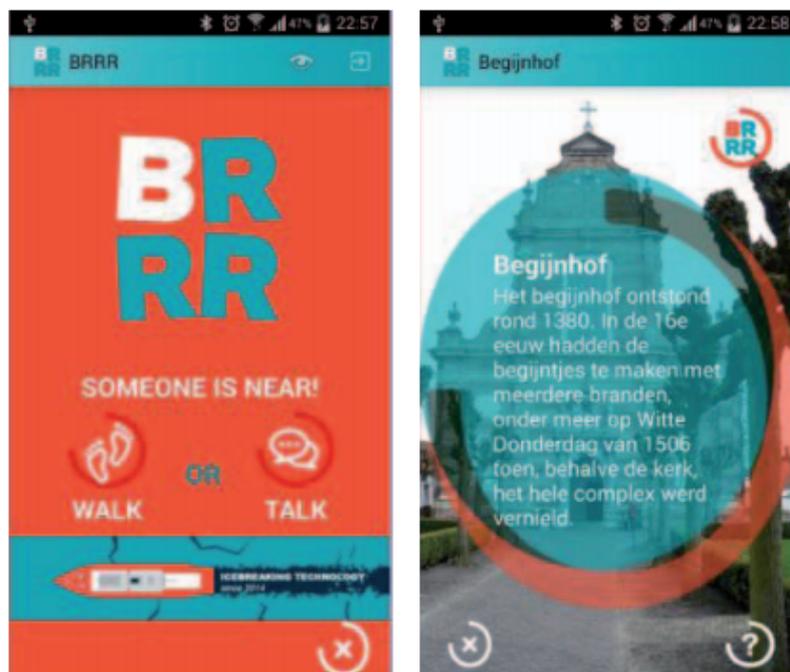


Figura 8 – Interface do aplicativo desenvolvido por (VAN MERODE et al., 2016).

Uso de Visão Computacional e IoT para Gerenciamento Inteligente de Estacionamento

No trabalho de (SIECK; CALPIN; ALMALAG, 2020), os autores propõem o desenvolvimento de um sistema gerenciador de estacionamento inteligente utilizando *Machine Vision* para controlar as vagas disponíveis e apresentando-as através de uma aplicação móvel.

O aplicativo foi desenvolvido para *iOS*, o que abre como possibilidade de trabalho o desenvolvimento para a plataforma *Android*. Também como trabalho futuro, o autor propõe o desenvolvimento de um app "*hands free*", para que os usuários possam utilizá-lo enquanto dirigem e para tal, é sugerido o uso da assistente *Siri*. O autor também ressalta a necessidade da instalação de câmeras em pontos estratégicos, mas destaca que permissões devem ser concedidas pela universidade antes da instalação.

O autor deste trabalho também sugere a implementação de uma análise de dados através de IA para determinar estatísticas, tais como horários de pico e regiões preferidas do estacionamento, ou seja, determinar locais onde há uma demanda maior pelas vagas de estacionamento. O autor ressalta que estas informações poderiam ajudar num gerenciamento mais eficiente das vagas disponíveis. A Figura 9 exibe o diagrama da arquitetura do sistema desenvolvido.

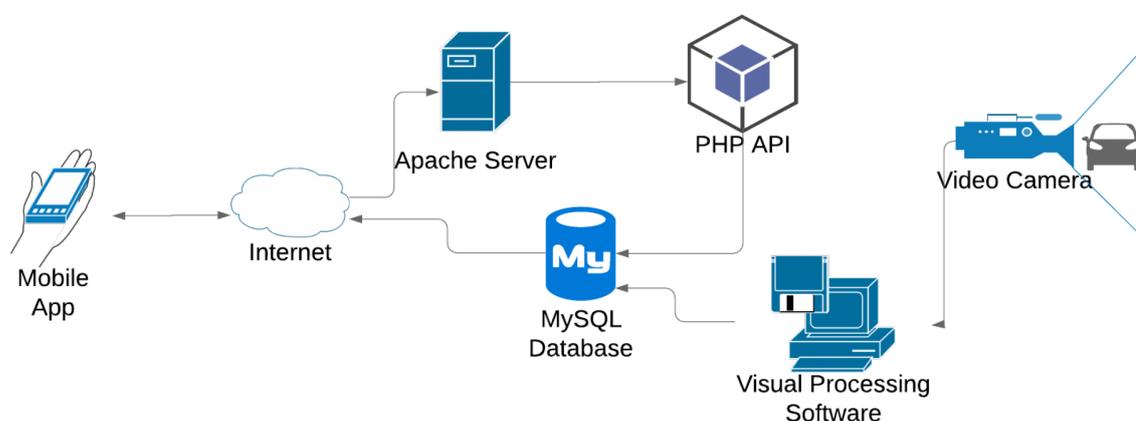


Figura 9 – Diagrama da arquitetura do sistema de estacionamento inteligente. Disponível em (SIECK; CALPIN; ALMALAG, 2020)

Geração Inteligente de Energia

Em (MALATJI; KWESI NTSALUBA, 2018a) é proposto um conceito de um sistema inteligente para gerar energia. Neste sistema há uma interconexão entre diferentes fontes de energia, tais como solar, biogás e armazenamento através de baterias conforme apresentado na Figura 10.

O trabalho apresenta um estudo de caso específico para a Universidade de Joanesburgo porém por se tratar de um conceito simples, a mesma ideia pode ser aplicada em diferentes universidades.

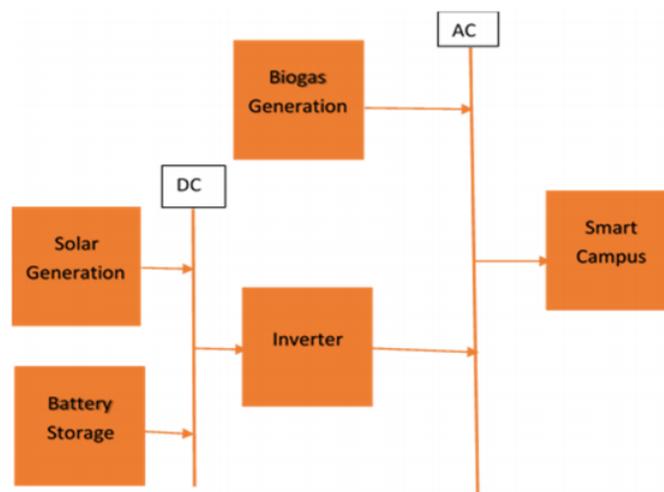


Figura 10 – Sistema de energia proposta por (MALATJI; KWESI NTSALUBA, 2018a).

Aplicação Móvel para Emitir Notificações Inteligentes com Base na Localização

O uso da localização para emitir notificações inteligentes é o foco do trabalho de (RUSLI; HALIM, 2019).

Diferente de (VAN MERODE et al., 2016) que utiliza geo-localização, os autores deste trabalho propõe a utilização do Wi-Fi para emitir notificações com base na rede que o usuário está conectado, dessa forma a gestão da universidade pode comunicar aos funcionários ou alunos problemas referentes a um determinado andar por exemplo, aumentando de forma considerável a precisão.

2.5.2 Análise das Aplicações e Considerações

Conforme evidenciado na Tabela 3 percebemos que há uma lacuna no campo das aplicações que se referem a saúde dos frequentadores de um campus. Dada a pandemia do Coronavírus (Covid-19) no ano de 2020, presume-se que nos anos seguintes este campo seja de potencial interesse para os mais diversos pesquisadores. A mídia já noticia estudos referentes a descontaminação e medidas de propagação do vírus em ambientes escolares.

Também percebemos que há muitos estudos relacionados e preocupados em solucionar problemas com a forma de aprendizagem. Por estar em constante evolução, este campo é muito visado por pesquisadores que tentam aprimorar as metodologias. Já o uso de inteligência artificial vem se tornando popular nesta área, principalmente a aplicação de algoritmos de *machine learning* para analisar e encontrar padrões em

grandes quantidades de dados gerados através de aplicações voltadas para o ensino remoto.

O principal desafio encontrado pelos pesquisadores é resolver os problemas do cotidiano de um campus com poucos recursos. Dessa forma, a utilização de aparelhos móveis que fazem parte do conjunto de dispositivos IoT vem se tornando cada vez mais comum. Dentre os principais benefícios observados nas aplicações, podemos citar a economia de custos operacionais, redução no tempo para realização de tarefas através de automatizações e melhoria dos sistemas de ensino e aprendizagem.

Todas as aplicações analisadas podem facilmente ser implementadas em universidades brasileiras com baixo custo de implantação. A popularização de dispositivos móveis tornam os alunos a figura principal no processo de aprendizagem de novos conteúdos já que a busca por informação é algo quase natural.

2.6 Tolerância a Falhas

Esta seção apresenta os conceitos, modelos e métodos de tolerância a falhas. Em seguida é abordado os principais desafios encontrados atualmente e uma revisão bibliográfica para elencar as principais técnicas utilizadas pela academia.

Sistemas de computação são suscetíveis a falhas. Estas falhas podem ser dos mais variados tipos, desde a leitura errada de um sensor até mesmo a troca de um bit em um chip danificado. A presença de radiação por exemplo, também pode ser um causador de falha. Uma das referências mais conhecidas de falhas deste tipo é o incidente envolvendo o satélite Galaxy IV, ocorrido em 1998. Este satélite, usado para transmitir dados e sinais de televisão, sofreu uma falha devido a uma tempestade de partículas eletrificadas que causaram uma elevação na radiação eletromagnética na atmosfera da Terra. A radiação acabou danificando o sistema de controle de orientação do satélite, levando a uma perda total de comunicação com a Terra. Este incidente ilustra como a radiação pode afetar negativamente sistemas eletrônicos e causar falhas que podem acontecer tanto no hardware como também no software.

Um clássico exemplo de falha em software é o caso da espaçonave Ariane V, onde o software não foi programado para lidar com valores numéricos muito grandes, e quando um valor excessivamente grande foi recebido pelo sistema de orientação, ele falhou e causou a perda total do controle da espaçonave culminando numa grande explosão poucos segundos após o seu lançamento.

Como mencionado, a consequência dessas falhas pode levar a um colapso geral nos sistemas, interrompendo serviços e consequentemente levando a perda de dados. Prever e remover falhas pode não ser suficiente em sistemas que demandam confiabilidade e alta disponibilidade. Nestas situações o sistema deve ser construído usando técnicas de tolerância a falhas (WEBER, 2003). Estas técnicas podem exigir

componentes adicionais ou algoritmos especiais.

A aplicação de determinadas técnicas pode evitar consequências desastrosas. Ao longo dessa Seção discutiremos os desafios de implementar sistemas com tolerância a falhas e algumas técnicas que podemos utilizar.

2.6.1 Conceitos e Classificação

O conceito de tolerar a falhas foi originalmente apresentado em (AVIZIENIS, 1998), porém técnicas para a elaboração de sistemas mais confiáveis já eram utilizadas anteriormente. A idealização principal é projetar sistemas que possam continuar funcionando mesmo quando há falhas em partes individuais. A ideia é projetar sistemas redundantes e distribuídos de tal forma que, quando uma parte falha, outra possa assumir seu trabalho, permitindo que o sistema como um todo continue funcionando sem interrupções significativas. Este conceito é amplamente utilizado em aplicações críticas, como sistemas de navegação aérea e sistemas financeiros, onde a disponibilidade e a confiabilidade são essenciais.

Segundo (PRADHAN, 1996) um sistema está suscetível a três tipos de anomalias. Um **defeito** é um tipo que classificamos como desvio de especificação, ou seja, algo que não estava projetado e não pode ser tolerado. Já um **erro** acontece quando o sistema está em um estado diferente do que o esperado. Este erro pode levar a um defeito. Uma **falha** é ocasionada por algum problema físico ou erro de programação. Esta última é algo inevitável já que estamos suscetíveis ao envelhecimento do hardware e também da alta complexidade de softwares.

Segundo (TANENBAUM A.S; STEEN, 2007), quando ocorre um desvio na especificação do sistema temos um defeito (*failure*). Já um erro pode ser considerado como um estado do sistema cuja etapa posterior poderá levar a um estado inesperado, ou seja, um defeito, sendo que a falha é a causa do erro.

Vemos aqui que ambos autores possuem definições semelhantes para os conceitos e classificação quanto as falhas. De acordo com (DANTAS, 2005) as origens e os tipos de falhas são muito variados. Como exemplo podemos citar um sistema de resfriamento que possui uma falha em um ventilador. Esta falha poderá aquecer os sistemas e apresentar um defeito posteriormente. Porém, esta pode não ser uma falha contínua e sim momentânea que não altera consideravelmente a temperatura e dessa forma não causa nenhum mal funcionamento dos sistemas. Portanto, há de ressaltar que nem todas as falhas são consideradas críticas, ou seja, há falhas que não interferem na execução total de um sistema.

Na literatura as falhas são normalmente classificadas em natureza física e humanas. As falhas físicas são aquelas têm como origem os componentes e normalmente ocorrem devido ao envelhecimento do hardware. Já as falhas humanas, são originadas no projeto, ou seja, são erros de codificação.

Segundo (TANENBAUM A.S; STEEN, 2007), os requisitos que os sistemas devem possuir para serem considerados tolerantes a falhas são:

- **Disponibilidade:** um sistema tem de estar sempre pronto para o uso imediato. De modo geral, quando requisitado, o sistema deverá responder;
- **Confiabilidade:** um sistema deve ser capaz de funcionar sem falhas, ou seja, aquele sistema que funcionará corretamente sem interrupções;
- **Segurança:** se um sistema falhar durante um tempo, nada de catastrófico deverá acontecer;
- **Manutenção:** um sistema deve ser fácil de aplicar manutenções em caso de falhas.

As falhas quando ocorrem também são classificadas em três subtipos. **Falhas transientes** são aquelas que acontecem uma vez e se a operação for repetida não aparecerá novamente. **Falhas intermitentes** são as mais difíceis de diagnosticar pois aparecem e desaparecem sem motivos aparentes. Já as **falhas permanentes** são aquelas que continuam a acontecer mesmo quando o componente é substituído (TANENBAUM A.S; STEEN, 2007).

O texto acima apresenta os conceitos e classificações de sistemas tolerantes a falhas e os principais desafios encontrados na literatura são:

- Métodos para detecção de falhas em hardware e software;
- Métodos de paralelismo para evitar o comprometimento dos resultados no caso de falhas de hardware;
- Desenvolver dispositivos móveis e embarcados que tenham confiabilidade e segurança bem como reduzido consumo de potência;
- Conciliar alta disponibilidade e confiabilidade com alto desempenho.

2.6.2 Técnicas Aplicadas a Sistemas Tolerantes a Falhas

As técnicas tolerantes a falhas são métodos e estratégias usados para garantir a confiabilidade e a disponibilidade de um sistema ou processo, mesmo em caso de falha ou erro. Essas técnicas são projetadas para prevenir ou mitigar o impacto de falhas e para garantir que o sistema ou processo possa continuar a operar de forma eficaz, apesar da presença de falhas ou erros.

Existem muitas técnicas diferentes de tolerância a falhas que podem ser usadas, dependendo da natureza do sistema ou processo e dos tipos de falhas esperados. Algumas técnicas comuns de tolerância a falhas incluem:

- **Redundância:** esta técnica envolve o uso de várias cópias ou componentes de backup, que podem ser usados para assumir o controle se um componente primário falhar. Por exemplo, um servidor com fontes de alimentação redundantes ou discos rígidos pode continuar operando mesmo se um dos componentes falhar.
- **Deteção e correção de erros:** Esta técnica envolve o uso de algoritmos ou hardware para detectar e corrigir erros que ocorrem durante a operação de um sistema ou processo. Por exemplo, um sistema de computador pode usar um código de correção de erros para detectar e corrigir erros na transmissão de dados.
- **Isolamento de falhas:** Esta técnica envolve o uso de estratégias para evitar a propagação de falhas ou erros dentro de um sistema. Por exemplo, um sistema de software pode usar mecanismos de isolamento para evitar que uma falha em uma parte do sistema afete outras partes do sistema.
- **Mecanismos de recuperação:** Esta técnica envolve o uso de estratégias para restaurar um sistema ou processo à operação normal após a ocorrência de uma falha ou erro. Por exemplo, um sistema de banco de dados pode usar pontos de verificação e recuperação baseada em log para restaurar o banco de dados a um estado consistente após uma falha.

No geral, as técnicas tolerantes a falhas são um aspecto importante para garantir a confiabilidade e a disponibilidade de sistemas e processos e são usadas em uma ampla variedade de setores, incluindo computação, telecomunicações e transporte.

Para que um sistema seja classificado como tolerante a falhas devemos construí-lo utilizando técnicas tolerantes a falhas. Estas técnicas garantirão que o sistema continue operando mesmo na presença de anomalias. A maioria destas técnicas aposta na redundância de componentes e na utilização de algoritmos dedicados.

As técnicas de tolerância a falhas podem ser divididas em dois grupos **mascamamento** ou **deteção, localização e reconfiguração**.

O mascaramento de falhas garante a resposta correta mesmo na ocorrência de falhas que não é tratada como um erro, já que o sistema não entra em um estado errôneo. Caso a falha se torne recorrente, ainda é necessário localizar e reparar a origem da falha.

O segundo grupo conta com uma sequência de atividades que devem ser executados na identificação de algum tipo de falha. O primeiro desafio é detectar e identificar uma falha como um erro, pois a mesma pode permanecer no sistema sem se manifestar por um longo período de tempo. Após a localização, podemos reparar a falha substituindo o componente danificado por exemplo.

2.7 Sensor Virtual

É dedicada uma seção neste capítulo de Fundamentação Teórica para tratarmos sobre o que é de fato um Sensor Virtual. Segundo (LIU; KUO; ZHOU, 2009) sensores virtuais são softwares que imitam funcionalidades de dispositivos físicos. Podem ser usados para coletar dados de várias fontes e processá-los de uma forma a reproduzir a saída de um sensor real. Também podem ser utilizados em vários cenários, como por exemplo, um sensor virtual de temperatura pode coletar dados de vários sensores físicos de temperatura e, em seguida, usar uma equação matemática para calcular a temperatura geral de uma área específica.

Outra aplicação é quanto a utilização para estimar ou prever a saída de um sensor que não esteja disponível, seja por falha temporária ou permanente. Este é o foco deste trabalho, realizar a utilização de sensores virtuais para corrigir e estimar dados de um ambiente real cujo sensor físico apresente falha.

Portanto, sensor virtual nada mais é do que uma abordagem que permite que os sensores sejam representados de uma forma digital, sem necessariamente possuir o dispositivo físico. É uma forma de simular o comportamento de um sensor através de modelos matemáticos, algoritmos e ou dados históricos. Então, o objetivo principal do uso de sensores virtuais é fornecer informações precisas e confiáveis sem necessariamente possuir o equipamento físico.

Já a virtualização de sensores é uma abordagem mais ampla que visa fornecer uma camada de abstração entre os sensores e as aplicações que os utilizam. A virtualização de sensores permite a integração de diferentes tipos de sensores, independentemente de suas especificações técnicas, em uma única plataforma. Além disso, a virtualização de sensores pode ser usada para criar sensores virtuais, assim como para gerenciar, monitorar e coletar dados dos sensores físicos.

Em resumo, o sensor virtual é uma aplicação específica da virtualização de sensores, que se concentra em simular o comportamento de sensores. Já a virtualização de sensores é uma abordagem mais ampla que visa fornecer uma camada de abstração para gerenciar e integrar sensores, tanto virtuais quanto físicos.

Para gerar um sensor virtual precisamos que o mesmo seja o mais fiel possível de sua representação física, este conceito é conhecido como Gêmeo Digital (*Digital Twin*) e segundo (PENIAK; BUBENÍKOVÁ; KANÁLIKOVÁ, 2021), consiste em criar um modelo virtual de um processo, produto ou serviço que reflete um objeto físico. Este conceito pode simplificar o desenvolvimento e operações de processos de tolerância a falhas já que o dispositivo irá atuar como um sensor virtual. A saída deste sensor virtual é obtida através de cálculos que fazem uso do grau de confiabilidade.

As principais razões para fazer uso de sensores virtuais são (PENIAK; BUBENÍKOVÁ; KANÁLIKOVÁ, 2021):

- Quando as informações não podem ser obtidas ou calculadas usando um dispositivo físico;
- Quando o sensor físico é muito limitado quanto a suas características e propriedades;
- Quando há altas probabilidades de os resultados dos sensores físicos estarem incorretos;
- Quando os sensores estiverem trabalhando em condições que possibilitem a ocorrência de falhas.

De certa forma, um gêmeo digital é um software que pega dados reais sobre um dispositivo físico e estima como esse dispositivo irá atuar com outras determinadas entradas. A ideia é que o gêmeo esteja sempre atualizado e seja uma cópia exata do dispositivo físico, incluindo forma, posição, status e outros valores.

Em Cristaldi et al. (CRISTALDI et al., 2020) é possível observar o uso de sensores virtuais como uma solução para manter o funcionamento do sistema mesmo na ocorrência de falha de um sensor físico. O conceito de sensor virtual permite aplicar redundância de elementos sensores físicos para geração de informações do sensor virtual (RAVEENDRANATHAN et al., 2012). No trabalho proposto por Cristaldi, o sensor virtual é obtido a partir de 4 diferentes tipos de sensores para inferir uma informação relevante para um sistema de controle em malha fechada aplicado. A Figura 11 apresenta um esboço das formas de conceber um sensor virtual, segundo (MARTIN; KÜHL; SATZGER, 2021).

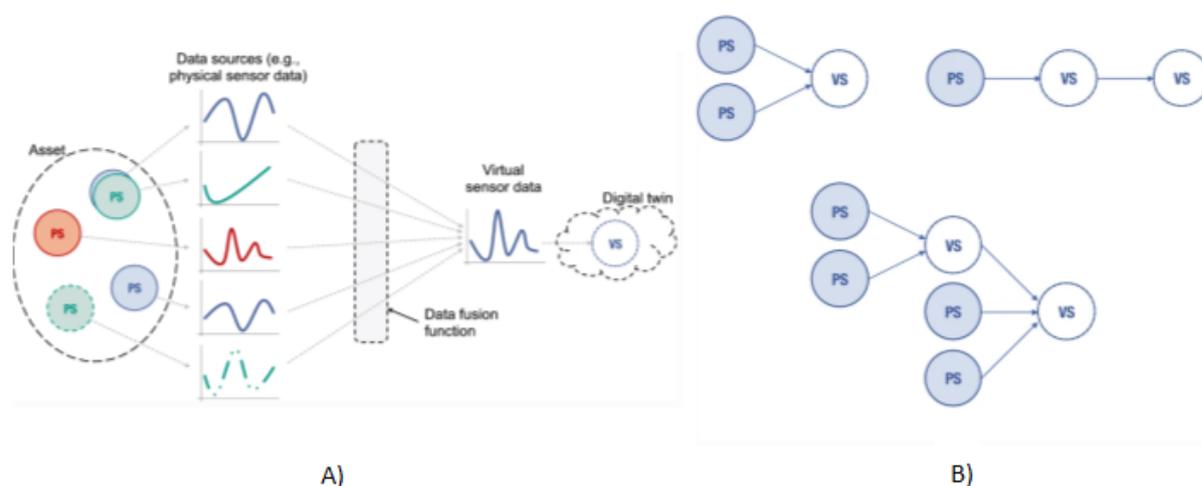


Figura 11 – Exemplo de sensores virtuais. Disponível em (MARTIN; KÜHL; SATZGER, 2021)

Já no trabalho proposto em (PENIAK; BUBENÍKOVÁ; KANÁLIKOVÁ, 2021) é usado a definição de sensores virtuais também como método para tolerância a falhas, porém o sensor virtual é obtido por meio da replicação de 3 sensores físico semelhantes, de

forma que o valor de saída do sensor virtual ocorre por meio de um votador que faz a avaliação dos valores gerados. No momento que um dos sensores falhar, o valor de saída é gerado a partir dos outros 2 sensores. Em diferença aos métodos anteriores da literatura, este trabalho propõe o desenvolvimento de três modelos para concepção de um sensor virtual obtidos a partir de apenas um sensor físico, de modo que a solução tenha baixo custo computacional para detectar e evitar a ocorrência de falhas.

(ABEEL; PEER; SAEYS, 2009) apresenta uma biblioteca Java com diversos algoritmos de Aprendizado de Máquina implementados, a Java Machine Learning Library (Java-ML). Além desta biblioteca, também é muito comum o uso de Python e a biblioteca *SciKit-Learn*. A utilização destas bibliotecas auxiliam na implementação, treinamento e utilização para inferência de novos valores no contexto de sensores virtuais.

Escolher um bom modelo para utilização é de extrema importância. Existem vários modelos que podem ser usados para prever valores com base em outros parâmetros. Algumas destas opções são:

- Regressão linear: Este é um modelo linear simples que pode ser usado para prever um valor contínuo com base em uma ou mais variáveis independentes. É uma boa escolha quando a relação entre as variáveis é aproximadamente linear.
- Regressão logística: Esta é uma variação da regressão linear que é usada para prever um resultado binário (por exemplo, 0 ou 1, verdadeiro ou falso) com base em uma ou mais variáveis independentes. É frequentemente usado para tarefas de classificação.
- Árvores de decisão: Este é um modelo baseado em árvore que faz previsões com base em uma série de decisões com base nos valores das variáveis de entrada. É uma boa escolha para modelar relações complexas entre variáveis.
- Florestas aleatórias (*Random Forests*): Este é um modelo de conjunto que consiste em várias árvores de decisão, cada uma das quais faz uma previsão e a previsão final é feita pela média das previsões de todas as árvores. Geralmente é mais preciso do que uma única árvore de decisão e é resistente ao overfitting.
- Máquinas de vetores de suporte (SVMs): Este é um modelo linear que pode ser usado para tarefas de classificação ou regressão. Funciona encontrando o hiperplano no espaço de recursos que separa ao máximo as diferentes classes.
- Redes neurais: são modelos inspirados na estrutura e função do cérebro. Eles consistem em várias camadas de nós interconectados e são capazes de aprender relacionamentos complexos entre variáveis.

Esses são apenas alguns exemplos, e há muitos outros modelos que podem ser usados para previsão, como *k*-vizinhos (*k-nearest*) mais próximos, aumento de gradiente (*gradient-boosting*) e muito mais. A escolha do modelo dependerá do problema específico que você está tentando resolver e das características dos dados com os quais está trabalhando.

A Figura 12 apresenta o algoritmo para obtenção de um modelo de um sensor virtual. Observamos que é um algoritmo linear, ou seja, precisamos realizar e concluir uma etapa para avançar para a etapa seguinte. Vale ressaltar que o processo de obtenção de dados é de extrema importância para um correto treinamento e funcionamento do modelo. O mais correto é que a captura dos dados ocorra no ambiente real em que o sensor irá ser instalado posteriormente. Se esta captura no ambiente real não for possível, o ideal é que seja realizado em um ambiente que simule as condições reais. Ao final, teremos um modelo de um sensor virtual que representa as características do sensor físico, ou seja, um gêmeo digital.

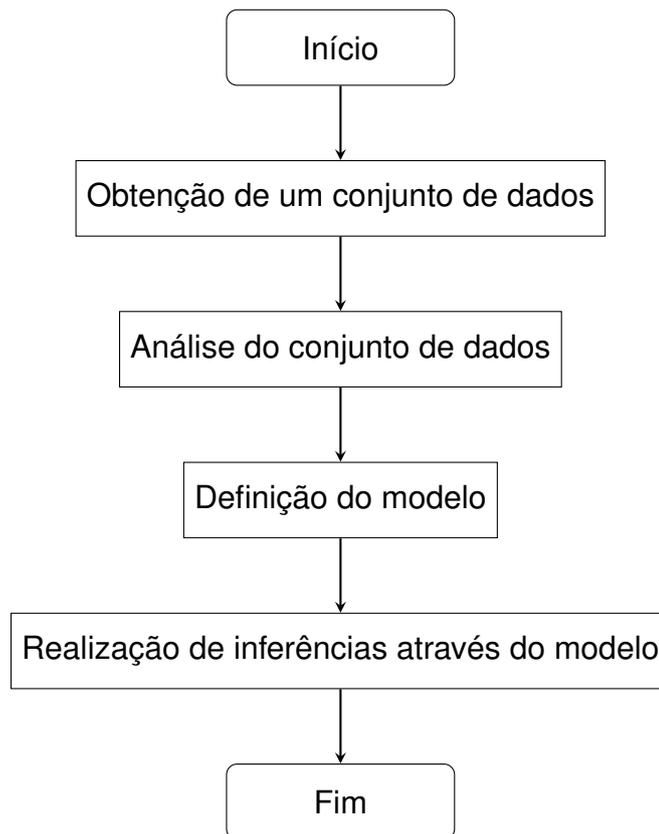


Figura 12 – Fluxograma para obtenção de um modelo de sensor virtual.

No Capítulo 4 é apresentada uma implementação de sensor virtual utilizando os conceitos aqui descritos.

2.8 Considerações

Como o leitor deve ter percebido, todos os sistemas estão suscetíveis a apresentarem diferentes tipos de falhas. A presente Seção apresentou algumas técnicas utilizadas no desenvolvimento de sistemas que podem diminuir a ocorrência destas falhas. Também foram apresentados alguns desafios que os desenvolvedores devem se preocupar quando estão criando novos sistemas e no decorrer deste trabalho, apresentaremos uma solução baseada na técnica de mascaramento e virtualização, utilizando um algoritmo para gerar dados simulados através de equações matemáticas.

3 REVISÃO BIBLIOGRÁFICA

A revisão bibliográfica é uma etapa fundamental em qualquer pesquisa científica, pois permite que os pesquisadores conheçam o estado atual do conhecimento sobre um determinado tema. Por meio da revisão bibliográfica, é possível identificar as principais tendências, lacunas e desafios relacionados ao assunto em questão, bem como as metodologias utilizadas para estudá-lo. Além disso, a revisão bibliográfica é uma ferramenta importante para fundamentar teoricamente a pesquisa e embasar as conclusões obtidas ao longo do estudo.

Neste capítulo de revisão bibliográfica são apresentados os principais estudos e pesquisas relacionados ao tema de Tolerância a Falhas e Sensores Virtuais, com o objetivo de traçar um panorama do estado atual do conhecimento sobre o assunto. Além disso, são abordados os principais conceitos e definições, bem como as principais tendências e desafios relacionados ao tema. São também discutidas as metodologias utilizadas para revisar o tema e as principais conclusões obtidas por meio das pesquisas realizadas até o momento. Inicialmente é conduzida uma busca mais ampla sobre Tolerância a Falhas e posteriormente uma busca dirigida para soluções baseadas em Sensores Virtuais.

3.1 Tolerância a Falhas

Durante a análise de trabalhos relacionados com tolerância a falhas foi utilizado o método de revisão por mapeamento sistemático cujo objetivo principal é encontrar lacunas na área de objeto de estudo. Para tal análise são elencadas algumas perguntas ou RQ (Research Question) para facilitar o processo de análise de trabalhos.

As perguntas utilizadas nesta análise foram:

1. Em qual nível hierárquico do sistema se encontra o processo de tolerância a falha (sensor, edge ou cloud)?
2. Qual a técnica utilizada no processo de identificação de uma falha?
3. Qual a técnica utilizada no processo de tolerância (correção) a falha?

4. Qual o método de validação da técnica proposta?

Uma vez definidas as perguntas da pesquisa devemos definir quais palavras-chaves serão utilizadas para proceder a busca de trabalhos. A Figura 13 apresenta o diagrama das *strings* de busca.

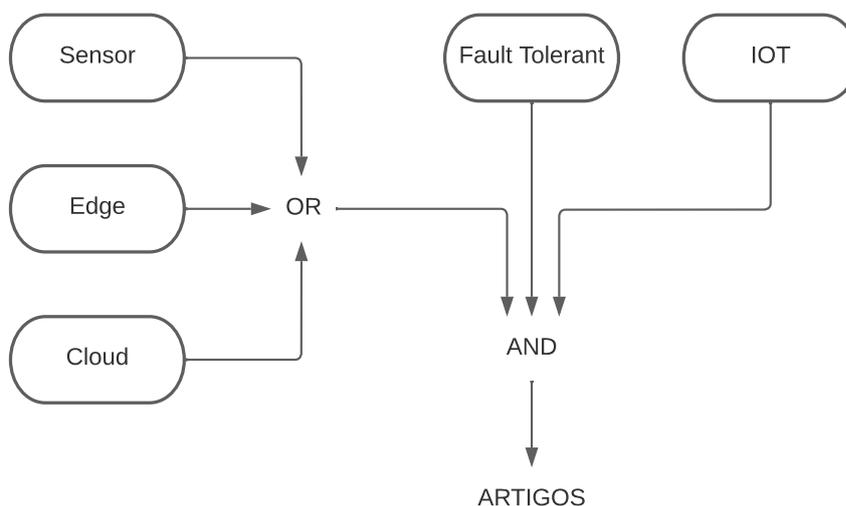


Figura 13 – Diagrama das *Strings* de busca para trabalhos relacionados com Tolerância a Falhas em sistemas IoT.

O próximo passo da revisão bibliográfica é definir em quais bases indexadores serão realizadas as buscas. Para este trabalho foram realizadas buscas na IEEE Xplore e ACM Digital Library.

Para assegurar a qualidade do trabalhos analisados são definidas também alguns critérios de inclusão e exclusão. São incluídos em nossa análise somente artigos que atendam as características desejadas para responder as perguntas e aqueles que não apresentam relevância são desconsiderados nesta etapa.

Os critérios utilizados para inclusão na análise são:

- Trabalhos escritos em inglês ou português;
- Publicações em conferências ou jornais a partir de 2015;
- Artigos relacionados a alguma aplicação com tolerância a falhas e ciências da computação.

Já os critérios utilizados para exclusão de trabalhos são:

- Artigos duplicados;
- Artigos não disponíveis para download;

- Artigos que não estejam relacionados com ciências da computação e com alguma aplicação com tolerância a falhas evidenciada no título.

Uma vez definidas as perguntas, as *strings* de buscas e os critérios de inclusão e exclusão foi realizada a busca nas bases selecionadas aplicando os filtros necessários. A Tabela 4 apresenta os resultados obtidos em cada base consultada.

Tabela 4 – Número de trabalhos encontrados em cada base de pesquisa.

Fonte	Resultados	Após critérios de inclusão e exclusão
IEEE Xplore	197	9
ACM Digital Library	3.788	5

A Figura 14 demonstra a evolução no número de artigos relacionados à esta área ao longo dos últimos anos.

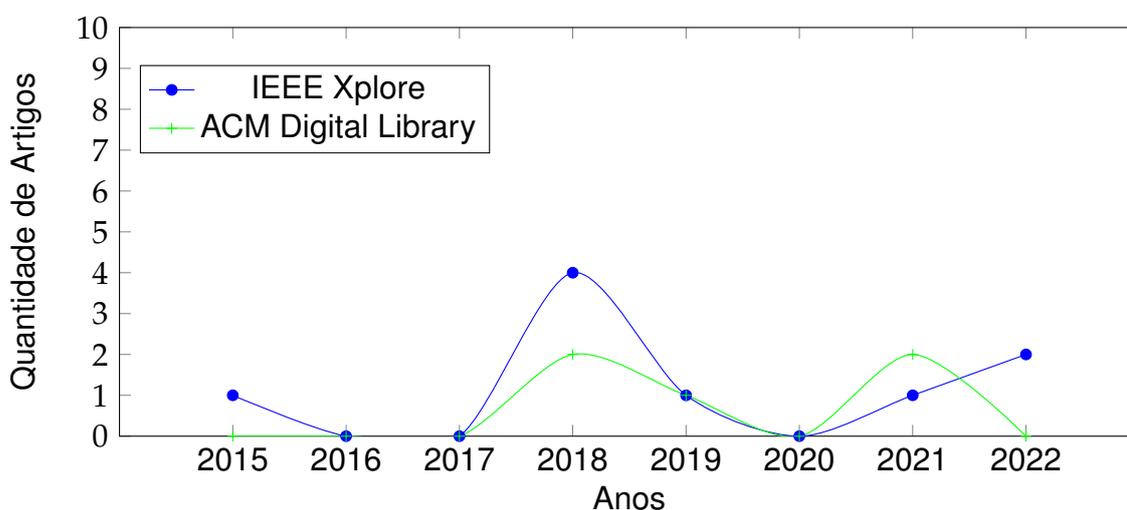


Figura 14 – Número de artigos encontrados nas bases de consulta após aplicação dos critérios de inclusão e exclusão por ano.

A Tabela 5 apresenta os principais artigos selecionados bem como a respostas para a RQs. Através desta tabela percebemos que há uma boa distribuição entre os focos de trabalho, tanto na tolerância a falhas de sensores quanto em níveis mais alto como Fog, Edge e Cloud. Também percebemos que os métodos de detecção de erros são bem semelhantes. Já na forma como a falha é tolerada, percebemos que a replicação de dados e sensores está sendo muito utilizada bem como a validação da maioria dos trabalhos se dá através de simulações.

Quando lidamos com dispositivos tolerantes a falhas normalmente as técnicas empregadas fazem uso de replicação de dados e dispositivos. O trabalho desenvolvido por (QAIM; OZKASAP, 2018) utiliza esta técnica de replicação. Nele é desenvolvido um sistema chamado DRAW que é responsável por replicar dados de um determinado dispositivo para os dispositivos vizinhos. Neste trabalho a troca de mensagens ocorre

Tabela 5 – Principais trabalhos e suas respectivas respostas para as RQs.

Trabalho	RQ1	RQ2	RQ3	RQ4
(KHRIJI et al., 2022)	Cloud	Análise de dados	Replicação de instâncias	Simulação
(PENIAK; BUBENÍKOVÁ; KANÁLIKOVÁ, 2021)	Sensor	Falha total no sensor	Sensor Virtual	Simulação
(MEDJEK et al., 2021)	Edge	IA	Apenas identificação	Simulação
(SAEED et al., 2021)	Edge e Fog	Análise de dados	Replicação de sensores	Simulação
(WANG et al., 2020)	Sensor	Não menciona	Sensor virtual (estimativa)	Simulação
(JAVED et al., 2020)	Edge	Erro de rede, não recebimento de dados	Replicação de dados	Simulação
(CASADO-VARA et al., 2018)	Sensor	Prevê qual valor será o próximo, e caso seja discrepante do real assume que está errado	Sensor virtual (estimativa)	Simulação
(WANG et al., 2018)	Fog	Transmissão de dados	Recoleta de dados	Simulação
(QAIM; OZKASAP, 2018)	Fog	Falha de conexão	Redundância de dados	Simulação
(WOO; LEE; PARK, 2018)	Sensor ou Dispositivo	Falhas de conexão em rotas	São criadas rotas de backup para replicação de dados utilizando o algoritmo Dijkstra	Protótipo em um ambiente simulado
(KAIWARTYA et al., 2017)	Fog	Falha de comunicação	Framework para virtualização de redes	Simulação
(ARDEKANI et al., 2017)	Dispositivo e conexões	Interrupção de serviço	Algoritmo de eleição para escolher o nodo primário	Simulação em ambiente controlado
(XU; POTKONJAK, 2016)	Dispositivo	Foca na recuperação e não na detecção	Redundância de dados	Simulação
(ZHOU et al., 2015)	Sensor	Verifica dados dos sensores vizinhos	Replicação de dados	Testou com um conjunto de sensores criando um protótipo

em períodos pré-determinados e apesar de ser uma solução viável para o compartilhamento de informações entre dispositivos seu modelo não prevê falhas justamente enquanto essa replicação de dados acontece. A mesma replicação de dados ocorre no trabalho apresentado por (XU; POTKONJAK, 2016) onde os dados são salvos em memória não-volátil.

Outra técnica apresentada por (CHUDZIKIEWICZ; FURTAK; ZIELINSKI, 2015) que também utiliza replicação é a realização de testes entre dispositivos. Ou seja, temos dois ou mais dispositivos iguais realizando o mesmo trabalho e caso um deles apresente algum valor discrepante dos outros, este é considerado falho, podendo ser considerado uma solução para detecção de falhas bizantinas que é um tipo de falha que ocorre em sistemas distribuídos, onde uma ou mais entidades do sistema agem de forma não-confiável ou incorreta.

Tratar e corrigir falhas é crucial para um sistema tolerante, porém o primeiro passo é identificar quando uma falha ocorre. Portanto, a identificação de falhas é um fator determinante para a tolerância. Os trabalhos apresentados por (GUTIERREZ et al., 2017), (GOTTSCHO et al., 2017) e (CHAKRABORTY et al., 2018) apresentam diferentes abordagens para monitorar e detectar as falhas.

Em (ZHOU et al., 2015) os autores propõem um esquema para adaptar tolerância a falhas em sistemas IoT por meio de uma arquitetura orientada a serviços utilizando sensores virtuais. O esquema proposto explora a característica de sistemas IoT possuírem vários sensores físicos com uma diversidade de propósitos dependendo da aplicação, o que naturalmente oportuniza redundância de informações desejadas para soluções tolerantes a falhas. Assim os autores propõem serviços virtuais que usam dados a partir de mais de um sensor físico de modo a permitir a substituição de um serviço real, ou seja, produzido por um conjunto pré-definido de elementos sensores físicos, caso ocorra a falha em um deles. Os autores usam regressão linear para identificar e gerar serviços virtuais a partir dos sensores físicos disponíveis. Dependendo do tipo de correlação existente entre os dados dos sensores usados para gerar o serviço é usado mínimos quadrados recursivos (do inglês, recursive least squares – RLS) ou regressão adaptativa multivariada (do inglês, multivariate adaptive regression splines – MARS) para geração de sensores virtuais. A correlação entre sensores é avaliada a partir do histórico de medições obtidas através de um protótipo. O trabalho de (ZHOU et al., 2015) está diretamente relacionado com os objetivos do trabalho aqui proposto, visto que se busca criar sensores virtuais a fim de construir aplicações tolerantes a falhas.

Em (CASADO-VARA et al., 2018) os autores propõem um algoritmo de controle cooperativo baseado em um modelo obtido a partir de cadeias de Markov para identificar e substituir sensores com falhas a fim de manter um sistema IoT tolerante a falhas. A proposta usa sensores virtuais em conjunto ao algoritmo proposto de modo a permitir

manutenção e confiabilidade a sistemas desta natureza. Desta forma sensores com mal funcionamento são substituídos de acordo com o modelo de predição do algoritmo proposto a fim de corrigir o sistema. O algoritmo é validado com dados obtidos a partir do monitoramento de temperatura em um supermercado. Em seu protótipo é utilizado um sensor DHT11 acoplado a uma ESP-01 e não há armazenamento local, os dados são enviados para um servidor em nuvem. Este trabalho usa uma solução de controle em malha fechada e processamento de matrizes usada nas cadeias de Markov o que exige, de certa forma uma complexidade computacional, o que se difere do trabalho aqui proposto que visa uma solução de baixo custo computacional.

Já em Peniak (PENIAK; BUBENÍKOVÁ; KANÁLIKOVÁ, 2021) os autores propõem a criação de um modelo para sensores virtuais redundantes de modo que um serviço virtual a nível de computação em nuvem seja disponibilizado. Este serviço usa o conceito de gêmeos digitais de modo disponibilizar serviço em nuvem a partir de grupos de sensores físicos que compõem sensores virtuais redundantes e assim garantir serviços tolerantes a falhas. Os sensores virtuais disponibilizam dados a partir de dois modelos que dependem do grau de confiabilidade necessário: um modelo que usa 2 sensores físicos e outro com 3 sensores físicos. A integração dos sensores físicos com o sensor virtual ocorre via protocolos de rede IoT, como por exemplo MQTT. O trabalho aqui proposto difere-se da proposta de Peniak (PENIAK; BUBENÍKOVÁ; KANÁLIKOVÁ, 2021) por lidar com o tratamento de falhas a nível do nó sensor na rede, reduzindo a complexidade da infraestrutura necessária para a execução.

Em (CRISTALDI et al., 2020) os autores propõem um modelo de sensor virtual obtido a partir de diferentes tipos de sensores físicos de modo a oferecer um serviço de monitoramento tolerante a falhas. O modelo proposto de sensor está baseado em dois sensores de temperatura situados em pontos diferentes no ambiente, um sensor de tensão elétrica e outro de corrente elétrica a fim de medir dissipação de potência. Os autores usam regressão não-linear para identificar falhas nos dados capturados pelos sensores. O estudo de caso é realizado em um inversor dedicado a geração de energia elétrica a partir de energia solar, monitorando temperatura no fluxo de ar de entrada, de saída, tensão elétrica e dissipação de potência do dispositivo.

Desta forma, destacamos o trabalho de Zhou, Casado-Vara e Peniak como as principais referencias para esta obra.

3.2 Sensores Virtuais

A fim de analisar trabalhos que envolvam especificamente o uso de sensores virtuais em sistemas IoT foi realizada uma revisão bibliográfica dedicada utilizando os seguintes métodos de pesquisa:

- Foram definidas as seguintes palavras chaves para realizar a busca nas bases

de pesquisa: **Virtual Sensor AND IoT AND Fault Tolerant**, conforme Figura 15;

- Foram aplicados filtros para selecionar apenas os trabalhos relacionados com Ciências da Computação e áreas afins relevantes;
- Foram descartados trabalhos que não indicavam conteúdo relacionado com o escopo da pesquisa, isto é, trabalhos com métodos de tolerância a falha utilizando sensores virtuais em dispositivos IoT;

Ao realizar a busca, estamos interessados em conhecer o estado atual da arte, ou seja, conhecer as últimas pesquisas e descobertas do tema e identificar lacunas de conhecimento.

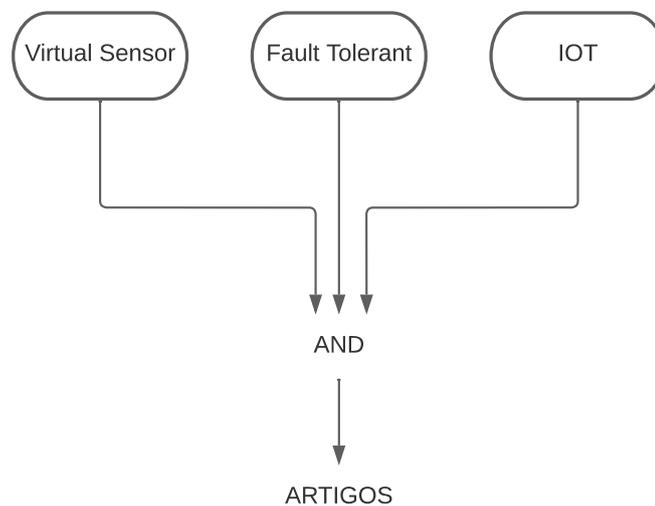


Figura 15 – Diagrama das *Strings* de busca para Sensores Virtuais.

Analisando a Tabela 6, podemos perceber que a base da IEEE Xplore e ACM Digital Library juntas retornaram apenas 11 trabalhos, dos quais após aplicação dos critérios de exclusão, restaram apenas 4. Esses resultados indicam que ainda há poucos estudos avançados sobre o conceito de tolerância a falhas com sensores virtuais aplicado a dispositivos IoT. No entanto, a proposta apresentada neste trabalho se destaca ao fazer uma associação com trabalhos já existentes e oferecer uma solução ultraleve, que pode ser aplicada em qualquer microcontrolador sem a necessidade de conexões extras ou redundância de dados e sensores.

Tabela 6 – Número de trabalhos encontrados em cada base de pesquisa

Fonte	Resultados	Após critérios de inclusão e exclusão
IEEE Xplore	6	2
ACM Digital Library	5	2

Dentre os trabalhos encontrados, estão os já mencionados (ZHOU et al., 2015) e (PENIAK; BUBENÍKOVÁ; KANÁLIKOVÁ, 2021). Além destes, também cabe citar aqui outros trabalhos de extrema relevância: (CHAKRABORTY et al., 2018) apresenta um método de detecção de falhas utilizado uma verificação na tensão de saída dos sensores. É um método bastante simples porém eficaz para detectar falhas permanentes pois são apresentados casos em que os sensores estavam fisicamente danificados. Neste trabalho não são utilizadas técnicas de sensor virtual e seu foco é em apenas identificar a falha e não em corrigir.

(ZANDBERG et al., 2022) apresenta uma implementação de um *middleware* fazendo uso de microsserviços e propõe uma nova técnica de virtualização ultraleve e isolamento de falhas para funções de software pequenas em microcontroladores de baixa potência para IoT. Essa técnica, chamada de Femto-Containers, pode ser utilizada em dispositivos IoT de baixo consumo com conexão apenas à internet. Embora seja uma solução ultraleve, a técnica de Femto-Containers faz uso de computação em nuvem para lidar com grandes quantidades de dados e necessidades de poder computacional. A proposta se destaca por ser uma solução que pode ser aplicada em qualquer microcontrolador sem a necessidade de conexões extras ou redundância de dados e sensores, o que a torna uma alternativa viável e promissora para o desenvolvimento de sistemas IoT. Apesar de ser uma virtualização "Ultra-Lightweight", é importante mencionar que a computação em nuvem é utilizada e portanto uma conexão com a internet durante todo o período se faz necessário.

Embora a técnica de Femto-Containers apresente vantagens, como a capacidade de ser aplicada em dispositivos IoT de baixo consumo e a promessa de isolamento de falhas, é importante destacar que a abordagem também apresenta algumas limitações. De modo geral, a técnica é voltada principalmente para funções de software pequenas, o que pode limitar sua aplicabilidade em certos tipos de sistemas IoT mais complexos.

Apesar de a revisão bibliográfica realizada não ter apresentado muitos resultados, é importante destacar que isso pode ser um indicativo de que o conceito de tolerância a falhas com sensores virtuais ainda carece de estudos mais avançados. A proposta de Femto-Containers apresentada no trabalho, portanto, é uma contribuição significativa para o avanço nessa área, uma vez que apresenta uma nova técnica de virtualização ultraleve e isolamento de falhas que pode ser aplicada em dispositivos IoT de baixo consumo. Essa técnica pode ser vista como uma alternativa promissora e viável para o desenvolvimento de sistemas IoT com menor risco de falhas, e merece uma avaliação mais aprofundada em futuros estudos.

4 MODELO PROPOSTO

Este capítulo apresenta o modelo de sensor virtual proposto e capaz de garantir tolerância a falhas a sistemas IoT. Este modelo possui flexibilidade para ser aplicado a diferentes tipos de sensores devido a sua baixa complexidade computacional. A solução proposta é capaz de detectar e corrigir falhas nas leituras dos sensores de modo a garantir que os dados continuem sendo coletados e analisados mesmo se alguma anomalia ocorrer. Uma visão abstrata do sensor virtual proposto é apresentado na Figura 16. O sensor virtual define um modelo de avaliação de falha aplicado individualmente sobre um sensor físico. Com isso, o método proposto pode ser replicado para diferentes sensores no sistema. Ao final, o modelo proposto é validado através de estudos de caso a fim de demonstrar sua aplicação no monitoramento da qualidade do ar no contexto da pandemia do COVID-19.

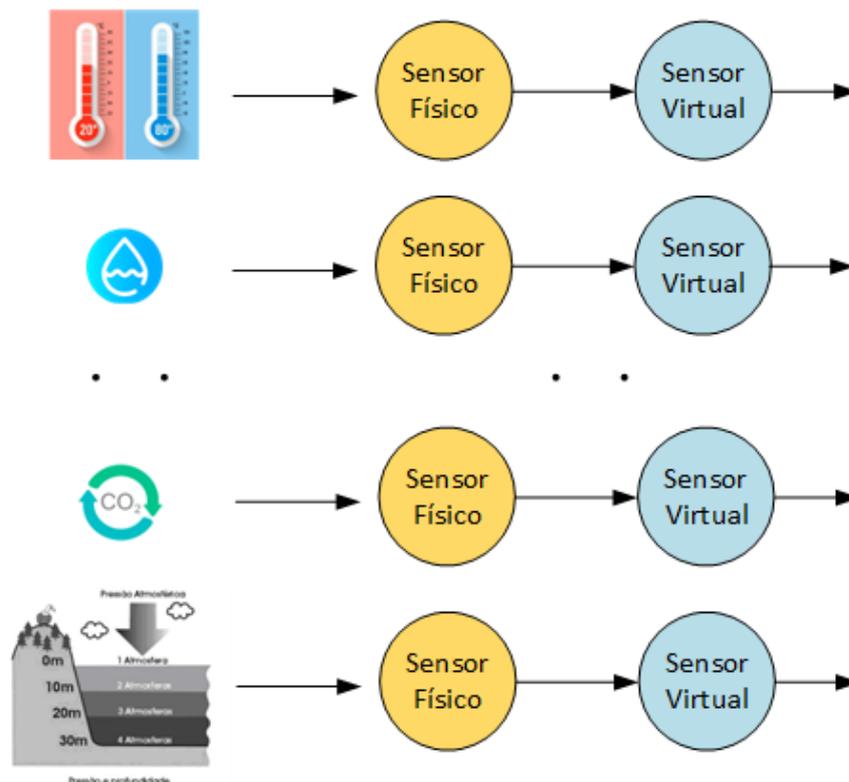


Figura 16 – Diagrama do sensor virtual proposto capaz de ser aplicado a diferentes variáveis.

Combater ou diminuir as taxas de contágio da COVID-19 e outras doenças que se propagam por vias aéreas são objetos de grande aspiração dos cientistas. Enquanto a aplicação de vacinas e novos medicamentos são desenvolvidos, a área da computação tem contribuído com novas soluções, sejam elas na área de inteligência artificial ou desenvolvimento de novos dispositivos para detecções de doenças. Um dos objetivos deste trabalho é desenvolver um sistema IoT composto por um conjunto de sensores que irão monitorar a qualidade do ar de um determinado ambiente. Segundo (EYKELBOSH, 2021) altos níveis de CO₂ presentes no ar podem indicar uma má circulação de ar e conseqüentemente sinalizar se este ambiente é ou não propício para o contágio de doenças respiratórias.

O modelo usado para detecção de falhas e a forma de evitar que a falha afete o funcionamento do sistema pode ser realizada usando diferentes estratégias tal como revisado na literatura. Neste trabalho são apresentadas 4 estratégias diferentes para o tratamento de falhas, sendo elas:

- Histórico de amostras;
- Regressão não-linear;
- Janela Flutuante;
- Método de Peniak.

A Figura 17 apresenta de uma forma abstrata a maneira como o sistema obtém os dados dos sensores e realiza a detecção de falhas. Em caso de alguma inconsistência é utilizado um dos modelos propostos para o tratamento desse erro através do sensor virtual.

4.1 Verificação dos Sensores

A verificação de funcionamento inicial dos sensores físicos é de extrema importância. É necessário que o sensor esteja ativo e gerando dados para que possam ser analisados posteriormente. Neste sentido, é chamada uma função responsável por testar e verificar se os sensores estão ativos, ou seja, realizar algumas leituras de verificação e aferir se houve retorno de dados. O Algoritmo 1 apresenta um exemplo de como podemos verificar se houve retorno de dados em um determinado sensor. A quantidade de tentativas representada pela letra m pode ser ajustado de acordo com a especificação de cada sensor.

Após obter o retorno da verificação, existem dois caminhos a seguir. O primeiro caminho tenta reinicializar o sensor que apresentou falha através de uma função de *reboot* (reinicialização). Caso contrário, o sistema procede com a leitura dos dados caso o mesmo esteja operando normalmente.

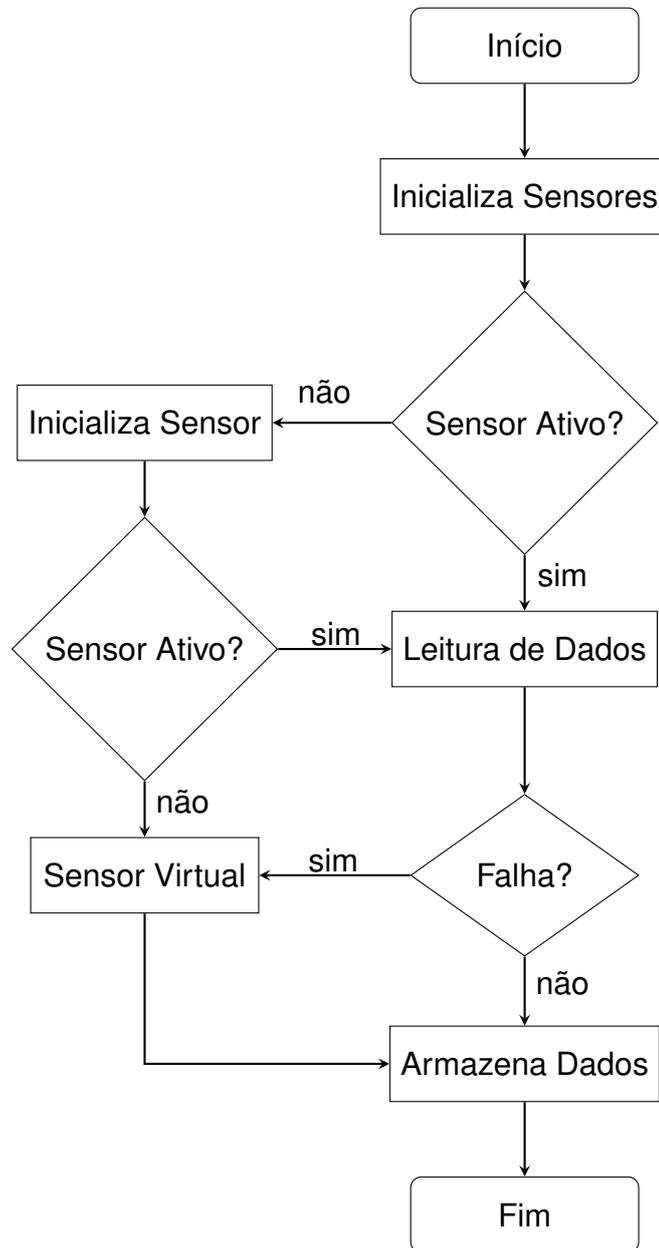


Figura 17 – Fluxograma para detecção de falha em sensores e correção através de sensor virtual.

Algoritmo 1: Verificação de Funcionamento de um Sensor.

Entrada: Sensor a ser testado

Saída: Validação do funcionamento do sensor

início

n = 0;

m = 5;

enquanto $n < m$ **faça**

ler dados do sensor;

se *!dados* **então**

| incrementa n;

senão

| retorna verdadeiro;

fim
fim

retorna falso;

fim

4.2 Método de Histórico de Amostras

Este modelo utiliza as amostras dos três últimos dias para realizar as previsões, ou seja, considera as últimas 72 horas de dados para corrigir uma possível amostra com falha. Este modelo leva em consideração as variações diárias que podem ocorrer. É importante destacar que como o modelo utiliza apenas três dias de amostras, variações climáticas ao longo do ano são minimizadas, como por exemplo diferenças de temperaturas de inverno e verão. Em caso de detecção de uma falha, o valor detectado é substituído pela média das amostras registradas no mesmo instante de tempo nos últimos 3 dias.

4.3 Método de Regressão Não-Linear

Este modelo utiliza uma equação matemática não linear, como por exemplo, um polinômio de grau 2, usada para estimar o comportamento das variáveis observadas. Para estimar a equação é preciso definir o conjunto de amostras a serem modeladas, que podem ser resultado de uma média histórica da variável medida, a média das curvas de máximo e mínimo dentro de um período de medição e também as amostras obtidas no último dia de medição. A Equação 1 é usada para definir o modelo sobre os dados analisados, onde x é uma amostra obtida ao longo do tempo e os coeficientes β_1 e β_2 devem ser ajustados para reproduzir o comportamento dos dados em análise. O ajuste de coeficientes é realizado por aproximações sucessiva seguindo a somatório de erros entre amostras estimados pela Equação 1 e as amostras medidas $STQ = \sum_{i=1}^N (x_i - \bar{x})^2$, onde STQ é a soma total dos quadrados, que representa o erro total gerado pela equação. Por aproximações sucessiva este erro deve ser mi-

nimizado, onde $STQ = 0$ demonstra que a equação representa fielmente o conjunto de dados modelados (FRANCO, 2006). Este ajuste pode ser calculado de maneira automatizada por diferentes ferramentas como Matlab e Excel, por exemplo.

$$Y(x) = \beta_2 x^2 + \beta_1 x + \alpha \quad (1)$$

4.4 Método de Janela Flutuante

O método de janela flutuante é uma técnica de identificação de falhas baseado em uma janela de tempo ou de valores ao redor de um respectivo dado. A técnica consiste em calcular uma média ou mediana dos valores dentro da janela e usar esse valor como uma estimativa de comparação para o dado. A janela pode ser definida com base no número de amostras coletadas imediatamente anteriores ao dado com falha, ou pode ser ajustada dinamicamente com base no período de amostragem dos dados. Por exemplo, se definirmos que determinado sensor utilizará uma janela com 10 amostras, o cálculo da média móvel deve considerar as 10 últimas amostras obtidas para corrigir a falha. É apresentado no estudo de caso a importância deste método que é geralmente mais preciso do que a técnica de média simples ou média retroativa (HANSUN, 2013), especialmente em dados com alta variabilidade.

Portanto, uma vez obtidos os dados dos sensores, é necessário validar os mesmos. Em alguns casos encontramos valores não esperados nas leituras, isto é, valores atípicos que estão fora do contexto para o ambiente monitorado. Para que as análises futuras destes dados não sejam comprometidas por conta destas anomalias, introduzimos os dados em uma equação responsável a fim de avaliá-los quanto a precisão.

A Equação 2 apresenta como é realizada a estimativa de erro com base em uma faixa de valores lidos anteriormente. Na Equação 2, ϵ é o valor de erro da leitura atual que queremos descobrir, x é a leitura atual obtida através do sensor e m é o tamanho da janela que estamos usando como base de cálculo. Já a variável i representa a posição atual que a leitura se encontra no vetor representado por X . Com o valor calculado de erro, verificamos se a leitura atual está dentro de um padrão pré-estabelecido conforme a Tabela 7. Para não comprometer leituras futuras, podemos atribuir pesos para cada nível de acurácia no cálculo da nova média, ou seja, leituras com alta acurácia possuem grandes probabilidades de estarem corretas, já mudanças abruptas de valores que ocasionam baixas acurácias podem indicar níveis anormais e podem comprometer as análises. Nestes cenários podemos indicar pesos menores.

$$\epsilon = \left| \frac{x \times 100 \times m}{\sum_{i=m}^{i-1} X[i]} - 100 \right| \quad (2)$$

Se um valor for classificado como falha sua contribuição não deve ser levada em

consideração para o cálculo de erro de uma leitura posterior. Neste caso, podemos optar por fazer uso de uma estimativa para que ele atribua um valor correspondente ao atual cenário.

Tabela 7 – Estimativas de acurácia e identificação de falhas com base no erro. Baseado em (CASADO-VARA et al., 2018).

Nível de Acurácia	Erro (%)
Alta acurácia	$\epsilon \leq 10$
Boa acurácia	$10 < \epsilon \leq 20$
Baixa acurácia	$20 < \epsilon \leq 35$
Falha	$\epsilon \geq 35$

Em nosso cenário, podemos utilizar uma inferência para determinar quais dados são considerados errôneos, ou seja, quando uma falha é detectada. Segundo (CASADO-VARA et al., 2018) quando o erro for maior que 35% em relação a sua referência podemos concluir que o estado atual representa uma falha. É neste momento que iremos repassar as informações que compõem o estado atual para um método específico, que será capaz de inferir com alta precisão um valor correspondente para o atual cenário utilizando o mesmo conceito de janela flutuante.

O método de janela flutuante é uma técnica que pode ser utilizada também para correção de valores em séries temporais. A janela possui um tamanho previamente conhecido e é deslizada sobre a série, a média dos valores dentro da janela é utilizada para corrigir valores fora da faixa esperada.

Para aplicar o método de janela flutuante, é necessário definir o tamanho da janela que será utilizada para calcular a média dos valores. Em seguida, a janela é deslizada sobre a série, e a média dos valores dentro da janela é calculada. Se um valor fora da faixa esperada é encontrado, ele é corrigido utilizando a média dos valores dentro da janela.

É importante ressaltar que a escolha do tamanho da janela pode afetar a eficácia da correção. Uma janela muito pequena pode não ser capaz de capturar as variações temporais dos valores, enquanto uma janela muito grande pode suavizar demais as variações e perder informações importantes.

O método de janela flutuante é útil em situações em que os valores da série apresentam variações temporais significativas, mas ainda seguem um padrão geral. Esta técnica pode ser aplicado em diversos campos, como na correção de dados meteorológicos ou financeiros por exemplo.

Apesar de ser um método simples, este método apresenta uma série de vantagens tais como a correção em tempo real, ou seja, os valores podem ser corrigidos assim que são coletados, sem a necessidade de armazenar uma grande quantidade de dados. O método também é altamente adaptável e pode ser aplicado em diferen-

tes contextos, pois possui alta efetividade já que a correção utiliza os próprios dados gerados. Também é necessário destacar sua simplicidade de implementação e a economia de recursos computacionais gerados quando comparado com métodos mais complexos que demandam de grandes quantidades de dados para tratamentos.

4.5 Método de Peniak

O Método do modelo de Peniak (PENIAK; BUBENÍKOVÁ; KANÁLIKOVÁ, 2021) é uma técnica utilizada para corrigir possíveis falhas nos valores medidos por sensores. O Método se baseia na replicação de sensores para obter uma medida mais precisa e confiável.

O processo faz uso de pelo menos três sensores para medir a mesma grandeza física. Em seguida, são coletados os valores obtidos por cada um desses sensores. Caso exista uma diferença significativo entre as medidas, ou seja, uma discrepância que possa indicar a falha de um dos sensores, o Método de Peniak busca identificar qual sensor está apresentando valores incorretos. Para isso, o método verifica a menor diferença entre os três sensores e para corrigir a falha utiliza a média dos valores obtidos.

Dessa forma, o Método de Peniak utiliza a replicação de sensores e faz uso de análise estatística dos dados para corrigir possíveis falhas nos sensores.

4.6 Discussão e Comparativos

Este Capítulo apresentou o modelo de tolerância a falhas fazendo uso de sensores virtuais. Foram apresentados os métodos utilizados durante a concepção do modelo e agora dedicamos esta seção para realizar discussões e comparações do modelo proposto com as propostas relacionadas e disponíveis na literatura.

O trabalho de (CASADO-VARA et al., 2018) propõem um algoritmo de controle de temperatura com tolerância a falhas utilizando o conceito de Cadeias de Markov, onde os estados anteriores não são levados em consideração e são considerados irrelevantes para a predição dos estados futuros, desde que o estado atual seja conhecido. De modo geral, o algoritmo encontra o nodo IoT que não esteja funcionando corretamente e coletando dados com problema. As Cadeias de Markov permitem prever o estado futuro do nodo e caso a previsão seja de uma falha depois do tempo de controle ter expirado, um controlador envia um sinal para que o nodo IoT seja substituído, criando assim um sensor virtual naquela determinada posição. Este sensor virtual irá estimar dados de temperatura baseado nos seus nós vizinhos garantindo que os dados sejam coletados de uma maneira contínua, sem perdas de dados e com uma certa confiança. Percebemos que o modelo de (CASADO-VARA et al., 2018) difere

do modelo aqui proposto, haja visto que levamos em consideração os valores retroativos e não necessariamente fazemos uso de sensores vizinhos. Podemos utilizar a implementação aqui proposta com apenas um único sensor.

A proposta de (PENIAK; BUBENÍKOVÁ; KANÁLIKOVÁ, 2021) também faz uso de sensores virtuais com o conceito de gêmeos digitais, porém o que difere da proposta aqui apresentada é que o autor não leva em consideração dados progressos para a produção do sensor virtual e faz uso da redundância de sensores. O cálculo de erro em tempo de execução também garante que somente valores com altos níveis de confiabilidade sejam levados em consideração na composição do sensor virtual.

No trabalho apresentado por (ZHOU et al., 2015) a tolerância a falhas é fortemente baseada na utilização da replicação de sensores e na composição de sensores, isto é, inferir dados de um sensor em falha com o auxílio de outros sensores. A utilização de algoritmos de Inteligência Artificial aprimoram essa técnica visto que estes algoritmos são capazes de identificar e detectar padrões que normalmente não seríamos capazes de detectar. São estes pequenos padrões que fazem uma grande diferença na qualidade dos dados geradores posteriormente. Por outro lado, o uso de Inteligência Artificial exigem um grande banco de dados de informações em conjunto com algoritmos classificadores eficientes para ter uma boa precisão, o que torna a solução mais complexa que o modelo proposto neste trabalho.

5 RESULTADOS E EXPERIMENTOS

Os experimentos apresentados neste capítulo têm como objetivo validar a efetividade do método proposto e apresentar testes com os dados coletados pelos sensores.

Devido às limitações impostas pela pandemia de COVID-19, foram realizadas duas simulações em ambientes distintos. A primeira delas, descrita na Seção 5.3, foi realizada em um ambiente doméstico, com base em condições pré-definidas. Já a segunda simulação foi realizada em um ambiente de trabalho, na sala administrativa da Universidade Federal de Pelotas (UFPel), onde os sensores coletaram dados de 09/06/2022 a 13/10/2022. O Capítulo introduz inicialmente uma breve abordagem sobre COVID-19 e a descrição dos dispositivos utilizados. Em seguida são apresentados os resultados obtidos.

5.1 Estudo de caso: Investigação da Qualidade do Ar

O cenário de estudo de caso consiste em dois ambientes distintos: um ambiente administrativo localizado no campus Porto da Universidade Federal de Pelotas e um ambiente residencial localizado em um bairro próximo. O ambiente administrativo é utilizado diariamente por cerca de 12 funcionários, entre professores e técnicos administrativos. O espaço é composto por diversas salas, onde são realizadas atividades administrativas tais como reuniões, atendimento ao público e atividades de rotina. A ventilação do ambiente é realizada por meio de janelas, portas e sistema de ar-condicionado. Já o ambiente residencial é uma sala com aproximadamente 9 m^2 onde foi possível definir cenários e avaliar como os sensores atuariam nas condições previamente definidas.

O objetivo do estudo é analisar a qualidade do ar nestes ambientes, considerando variáveis como temperatura, umidade e concentração de CO₂. A análise foi realizada por meio da coleta de dados em diferentes horários do dia e em dias distintos da semana, a fim de verificar possíveis variações nos níveis de poluição do ar. A comparação entre os dois ambientes permitirá verificar possíveis diferenças na qualidade do ar e identificar fatores que possam influenciar na poluição do ar em cada um deles.

Sobre o objeto estudo de caso deste trabalho, usamos a COVID-19 que teve seu início em dezembro de 2019, quando a Organização Mundial da Saúde (OMS) foi avisada sobre diversos casos de um tipo de pneumonia na cidade de Wuhan, na China. Pouco tempo depois, as autoridades chinesas confirmaram que haviam detectado um novo tipo de Coronavírus. Segundo a Organização Pan-Americana da Saúde (OPAS) os Coronavírus estão por toda parte e são a segunda principal causa de resfriado comum e raramente causavam doenças mais graves em humanos, o que se provou o contrário. Em 11 de março de 2020, a COVID-19 foi caracterizada como uma pandemia pela OMS, ou seja, uma disseminação mundial com transmissão sustentada de pessoa para pessoa (SCHUELER, 2021).

Com a pandemia de COVID-19, a análise da qualidade do ar em ambientes fechados tornou-se ainda mais relevante, uma vez que a transmissão do vírus pode ocorrer por meio de aerossóis. A análise da concentração de partículas em suspensão e de CO₂, por exemplo, pode indicar a eficácia da ventilação e dos sistemas de ar-condicionado em reduzir a concentração de aerossóis no ambiente. Além disso, a análise da qualidade do ar em ambientes residenciais pode auxiliar na identificação de fatores que possam influenciar na transmissão do vírus entre os membros da família, uma vez que o contato próximo e prolongado em ambientes fechados é um dos principais meios de transmissão da COVID-19. Portanto, o estudo de caso que se propõe a analisar a qualidade do ar em um ambiente administrativo e residencial pode contribuir para a compreensão dos fatores que influenciam na transmissão do vírus em ambientes fechados e para o desenvolvimento de medidas preventivas mais eficazes.

A COVID-19 tem seu contágio e transmissão de diversas maneiras. Por se tratar de uma doença respiratória o meio de transmissão mais comum é o ar. É normal que pequenas gotículas de saliva que possuem tamanhos entre 5-10 μ M de diâmetro são expelidas enquanto falamos, sendo este considerado um dos meios de transmissão. As gotículas mais pesadas tendem a ir para as superfícies enquanto as mais leves podem permanecer em suspensão durante horas e serem levadas a distâncias maiores que 1 m.

O contágio ocorre quando uma pessoa está no mesmo ambiente de outra que possui sintomas respiratórios, sendo os mais comuns a tosse e coriza. Além da transmissão por via aérea, também foram relatados contágios através de contato com superfícies e objetos contaminadas.

Os sintomas do COVID-19 podem variar de leves a graves e podem incluir febre, tosse, dificuldade para respirar, dores no corpo, fadiga e perda do olfato ou paladar. Em casos graves, o COVID-19 pode levar a pneumonia, síndrome do desconforto respiratório agudo e até a morte. Atualmente, já há tratamento específico para COVID-19. Para evitar a propagação do COVID-19, é importante seguir as orientações de saúde pública, como usar máscara, lavar as mãos com frequência e praticar o distanciamento

físico. As vacinas também estão sendo aprimoradas e distribuídas em todo o mundo para ajudar a controlar a propagação do vírus e reduzir o impacto da pandemia.

5.2 Definição de Sensores e Dispositivos

Esta seção apresenta os tipos de sensores utilizados no desenvolvimento deste trabalho para o monitoramento de salas administrativas da UFPel.

5.2.1 Sensor de Temperatura e Umidade DHT11

O DHT11, presente na Figura 18, é um sensor de temperatura e umidade capaz de realizar leituras entre 0 a 50° Celsius e umidade entre 20 a 90%. Segundo sua ficha de dados (em inglês, datasheet), pode ser aplicado em equipamentos de testes e inspeções, automóveis, controladores automáticos, estações meteorológicas, reguladores de umidade, equipamentos médicos e outras aplicações que necessitem o controle de temperatura e umidade.

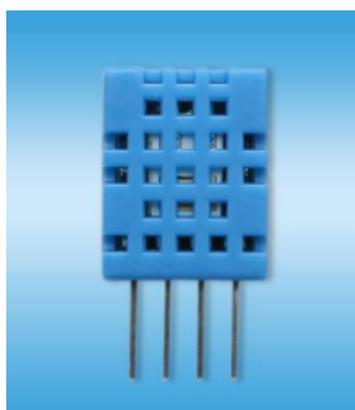


Figura 18 – Sensor de temperatura e umidade DHT11.

Como características podemos citar seu baixo custo, estabilidade a longo prazo, qualidade de medição, respostas rápidas, saída de sinal digital e calibração precisa. Suas pequenas dimensões também são fator extremamente positivo. A Figura 19 exhibe seus tamanhos em milímetros. Por ser um sensor surpreendentemente pequeno, pode ser aplicado em diversos casos, o que facilita sua instalação e manuseio.

A Tabela 8 apresenta as características e especificações elétricas deste sensor.

5.2.2 Sensor de Gases MQ-135

O sensor MQ-135 apresentado na Figura 20 é um sensor para detecção de diversos tipos de gases. Segundo seu *datasheet* pode ser utilizado em equipamentos para mensurar a qualidade do ar de prédios e escritórios. É capaz de detectar gases como dióxido de carbono, benzeno, amônia, óxido nítrico e também fumaça.

Como principais características, o *datasheet* destaca sua estabilidade a longo prazo, rápida resposta, alta sensibilidade e um circuito simples e de fácil aplicação.

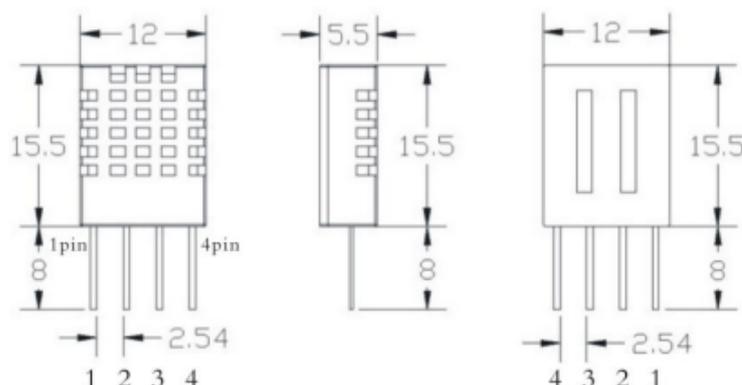


Figura 19 – Dimensões em mm do sensor de temperatura e umidade DHT11.

Tabela 8 – Especificações e características elétricas do sensor DHT11

	Umidade	Temperatura
Resolução	16 Bits	16 Bits
Precisão	±5 % UR	±2 % °C
Faixa de Medição	20 a 90% UR	0 a 50°C
Tempo de Resposta	2s	
Alimentação DC	3.5 ~ 5.5V	
Corrente	0.3mA, em standby 60µA	
Dimensões	23.5 x 12 x 5.5mm	

Seu tamanho também é benéfico para a utilização em projetos pequenos, possuindo apenas 40x25mm. A Tabela 9 apresenta outras especificações deste sensor.

Tabela 9 – Especificações e características elétricas do sensor MQ-135

Faixa de detecção	10 ~ 1000ppm
Alimentação DC	5V
Saída	Digital e analógica
Dimensões	40 x 25 x 22mm

Em nosso projeto, podemos utilizar a detecção de CO₂ para determinar, através de faixas de valores, a capacidade atual de uma determinada sala. Com estas medidas, iremos inferir se o ambiente está com a capacidade acima dos limites estabelecidos para a segurança de seus ocupantes.

5.2.3 Display LCD

Para facilitar a leitura dos dados e deixar o sistema desenvolvido mais flexível e portátil, podemos utilizar um display LCD 16x2 para exibir alguns informações e configurações ao usuário no próprio ambiente de coleta de dados. Sua comunicação com o microcontrolador é simples e há disponível bibliotecas que facilitam a sua utilização.

A Tabela 10 apresenta as especificações do display LCD e a Figura 21 apresenta



Figura 20 – Sensor de gases MQ-135.

um dos vários modelos disponíveis no mercado.

Tabela 10 – Especificações e características do Display LCD.

Alimentação DC	5V
Cor backlight	Azul
Cor escrita	Branca
Dimensões	80 x 36 x 12mm
Dimensão área visível	64,5 x 14mm
Dimensão caractere	3 x 5,02mm
Dimensão ponto	0,52 x 0,54mm

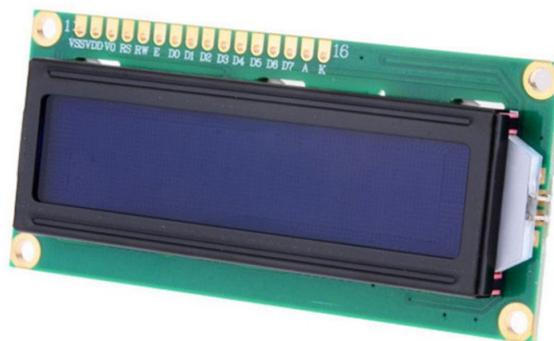


Figura 21 – Modelo de Display LCD para Arduino.

5.2.4 Arduino Nano

Para coletar os dados de todos os sensores é necessário a utilização de um microcontrolador. Neste projeto utilizamos o Arduino Nano, Figura 22. A escolha deste microcontrolador é justificada pela facilidade de encontrá-lo a venda bem como a facilidade de encontrar materiais para referência em livros, artigos disponíveis na internet ou até mesmo em videoaulas.

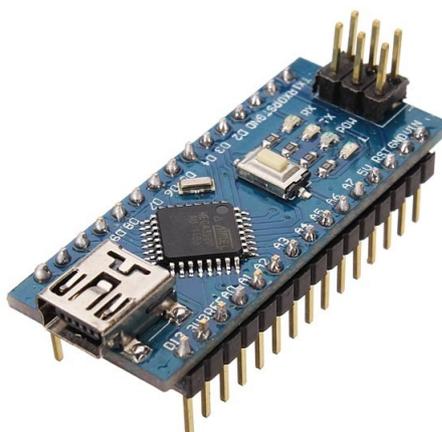


Figura 22 – Arduino Nano com microcontrolador ATmega 328.

O Arduino Nano é uma pequena placa de microcontrolador compatível com protoboard baseada no microcontrolador ATmega328. Ele foi projetado para ser usado em uma variedade de projetos eletrônicos e é especialmente adequado para projetos que exigem tamanho compacto ou baixo consumo de energia.

O Arduino Nano possui vários recursos que o tornam uma escolha popular para projetos eletrônicos. Ele possui vários pinos de entrada/saída (E/S) digitais e analógicos, que podem ser usados para conectar sensores, atuadores e outros componentes eletrônicos. Também possui uma interface USB, que pode ser utilizada para programar a placa e se comunicar com outros dispositivos. A Tabela 11 apresenta as especificações do Arduino Nano.

Além de seus recursos de hardware, o Arduino Nano também possui vários recursos de software que o tornam fácil de usar. É baseado na plataforma Arduino, que fornece um ambiente de programação amigável e uma grande comunidade de usuários que compartilham códigos, ideias e recursos.

No geral, o Arduino Nano é uma placa de microcontrolador versátil que é adequada para uma ampla gama de projetos eletrônicos. É fácil de usar e seu tamanho pequeno e baixo consumo de energia o tornam uma boa escolha para projetos que exigem um design compacto ou portátil.

A programação do dispositivo é realizada através de um software livre chamado de Arduino IDE utilizando a linguagem C. A transferência de código ocorre por meio da porta USB mini (NANO, 2018).

Por utilizar a Linguagem C, sua codificação se torna muito facilitada permitindo a criação de sub-rotinas, o que deixa o código mais organizado e de fácil entendimento para o leitor.

Tabela 11 – Especificações Arduino Nano.

Microcontrolador	ATmega328
Tensão de Operação	5V
Tensão de entrada	7-12V
Portas digitais	14 (6 PWM)
Portas analógicas	8
Memória flahs	32KB
SRAM	2KB
EEPROM	1KB
Velocidade de clock	16MHz
Dimensões	45 x 18mm

5.3 Definição do Cenário 1 - Ambiente Residencial

O ambiente cujo os testes foram realizados possui 9m² com uma janela de 2,4m² e uma porta de aproximadamente 1,7m².

O conjunto de sensores foi instalado aproximadamente no centro da sala. Durante o período de testes seus dados foram analisados no momento em que estavam sendo coletados. Estas análises permitiram uma maior compreensão sobre o funcionamento destes sensores, tais como sua sensibilidade à presença de pessoas, abertura e fechamento de janela e porta, acionamento de ar condicionado e também a detecção de valores anormais. Estes valores anormais podem ser ruídos ou falhas na leitura dos sensores. São estes valores que temos o objetivo de corrigir, dessa forma, a análise não será comprometida e os valores apropriados dos outros sensores podem ser mantidos.

A Tabela 12 apresenta os diferentes cenários que aplicamos em nosso ambiente. Foram captados dados a cada minuto durante duas horas em cada cenário. Os resultados são apresentados em forma de gráficos bem como a situação aplicada sobre cada cenário.

Tabela 12 – Cenários de testes com períodos de duas horas de duração e leituras a cada minuto.

Cenário	Situação
Cenário 1	Sala vazia com janela e porta fechada.
Cenário 2	Sala vazia com janela e porta aberta.
Cenário 3	Sala com uma pessoa e com janela e porta fechada.
Cenário 4	Sala com uma pessoa e com janela e porta aberta.
Cenário 5	Sala com uma pessoa e com janela e porta fechada e ar-condicionado ligado.

Nas seções seguintes são apresentados os resultados referentes a cada Cenário aplicado em nossos experimentos. Antes dos resultados, devemos apresentar as

condições iniciais da sala antes das medições. Por não se tratar de um ambiente completamente controlado, estas elucidacões iniciais se fazem necessárias para entender o comportamento que veremos nos gráficos a seguir.

Nos Cenários 1 e 2 analisamos o mesmo ambiente em três ocasiões distintas. A linha vermelha indica um ambiente que previamente estava altamente populado, neste caso 3 pessoas. Já a linha verde indica um ambiente com estado de partida moderadamente populado, neste caso com uma ou duas pessoas. A linha azul indica um ambiente que estava sem a presença de pessoas por um longo período de tempo.

Nos casos pertinentes, apresentaremos através de uma linha pontilhada preta, a média de todos os casos presentes no Gráfico, dessa forma podemos ter uma visão geral do acontecimento no cenário atual independente da configuração inicial do ambiente.

5.3.1 Cenário 1

As Figuras 23, 24 e 25 apresentam respectivamente os Gráficos para concentração de CO₂ em PPM, temperatura em °C e a umidade relativa do ar no Cenário 1 de nossos testes. No Cenário 1 aplicamos os nossos sensores na condição da sala totalmente fechada. Neste cenário analisamos os sensores de CO₂, Temperatura e Umidade em três condições. A linha vermelha indica a condição do Cenário 1 quando anteriormente haviam três pessoas na sala. A linha verde, indica o qual antes do período analisado haviam duas pessoas na sala. Já a linha azul da Figura 23 apresenta o momento em que o ambiente esteve vazio por várias horas antes de iniciar o experimento.

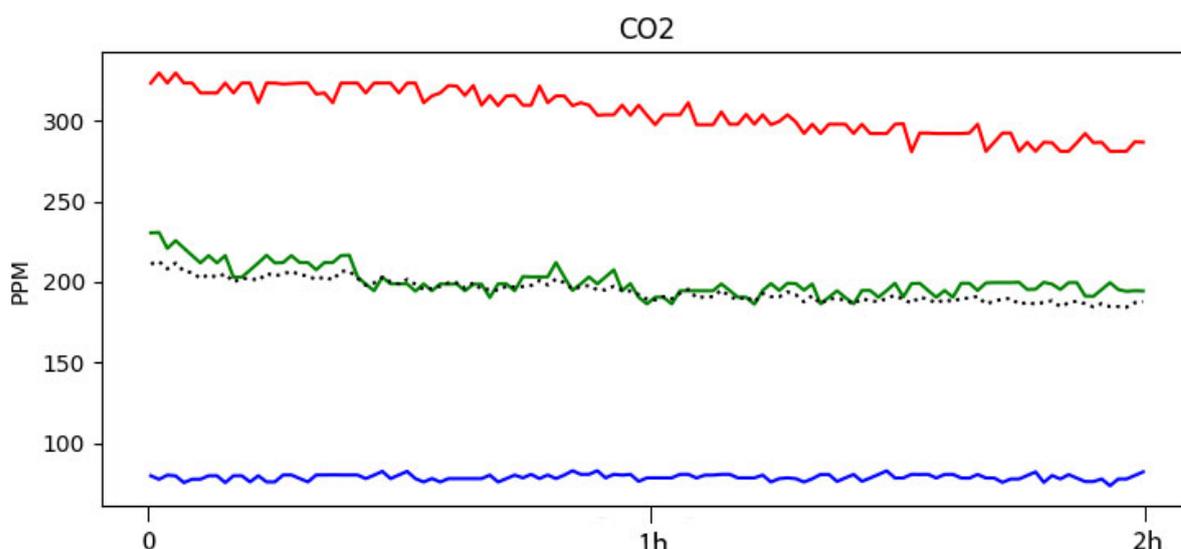


Figura 23 – Gráfico de concentração de CO₂ em PPM no Cenário 1.

O Gráfico presente na Figura 23 que apresenta a concentração de CO₂ durante o período de duas horas demonstra a lentidão da dissipação do CO₂ em ambientes

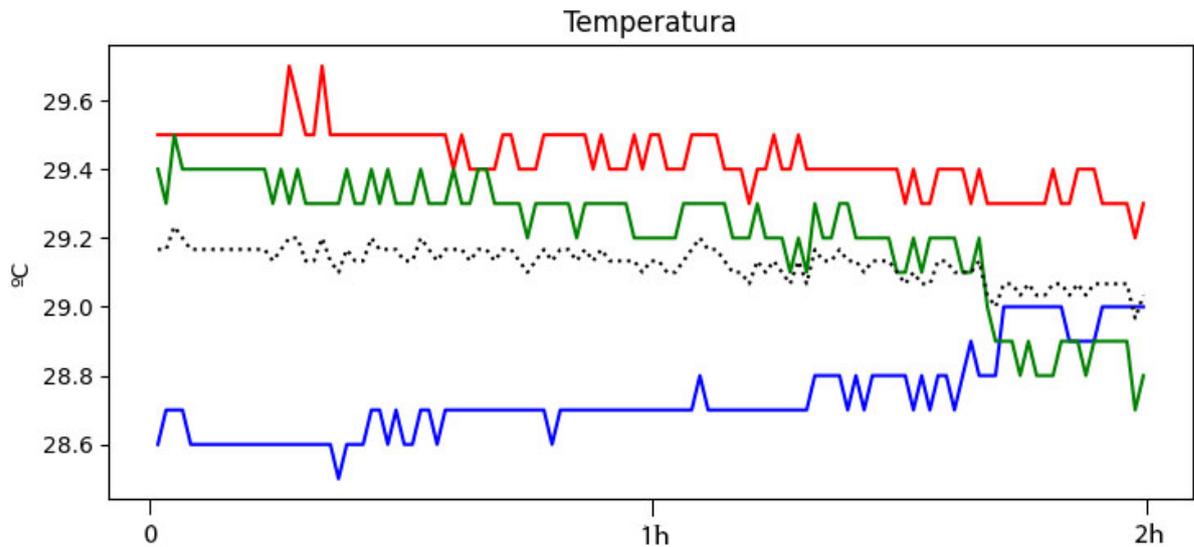


Figura 24 – Gráfico de temperatura em °C no Cenário 1.

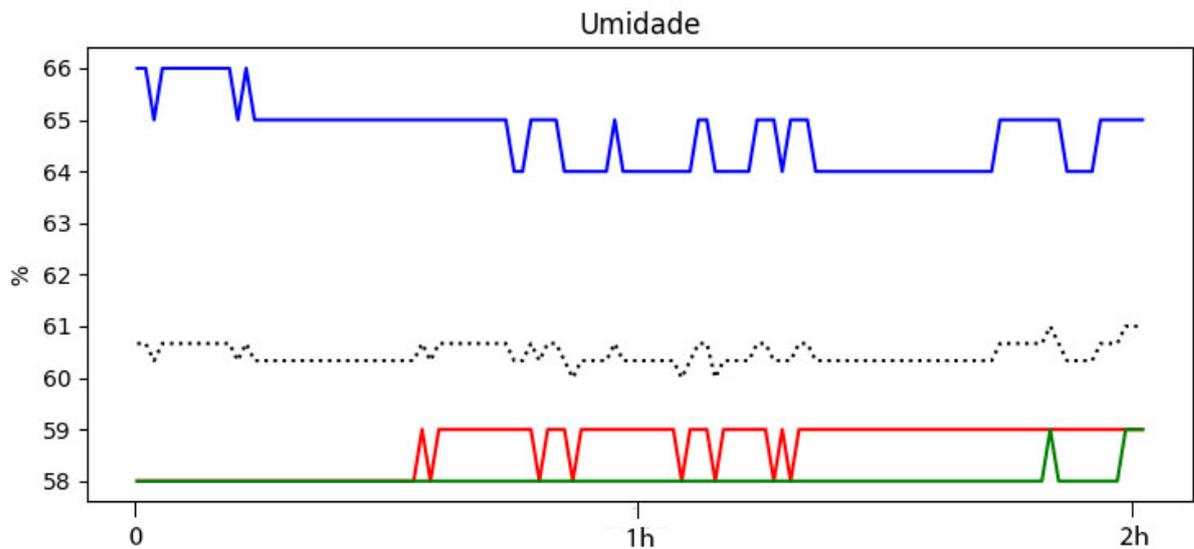


Figura 25 – Gráfico de umidade em % no Cenário 1.

fechados.

Após duas horas de análises, a concentração de CO₂ no ambiente sofreu uma leve diminuição, ficando comprovado que há pouca ou nenhuma troca de ar no ambiente. Manter os ambientes fechados pode ser um fator determinante no contágio de doenças respiratórias, visto que o ar não é renovado nestes ambientes e as partículas contaminadas podem permanecer em suspensão por longos períodos.

Já através dos Gráficos presentes nas Figuras 24 e 25 percebemos que não houveram alterações significativas na Temperatura e Umidade do ambiente durante o período observado.

5.3.2 Cenário 2

O Cenário 2 apresenta o monitoramento do ambiente sem pessoas e com janela e portas abertas, ou seja, com uma corrente de ar passando pelo ambiente. Pelo Gráfico de CO2 podemos comprovar como o fluxo de ar diminui a concentração de CO2 que estava presente no ambiente.

Comparando o Gráfico da Figura 23 com o Gráfico da Figura 26 percebemos como a abertura de porta e janela influenciou positivamente na dissipação da concentração de CO2 na sala testada, comprovando a eficácia desta ação simples de abrir portas e janelas para renovar o ar presente no ambiente.

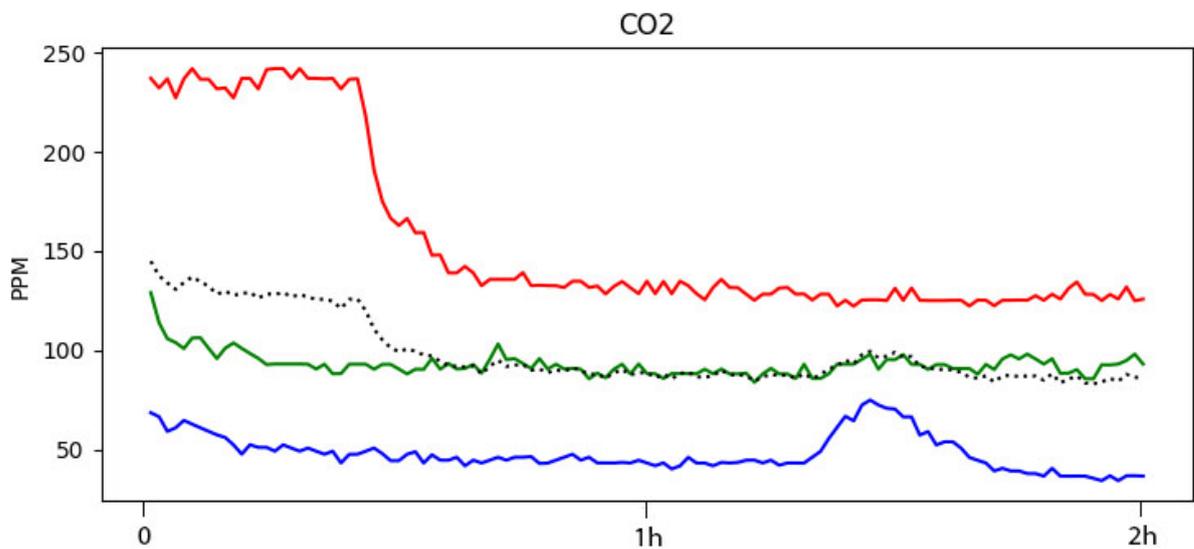


Figura 26 – Gráfico de concentração de CO2 em PPM no Cenário 2.

Tanto o Gráfico de Temperatura, Figura 27, como o de Umidade, 28, não apresentaram novidades neste cenário se mantendo numa faixa de valores já esperados. As variações presentes nestes gráficos são devido as alterações climáticas durante o período observado.

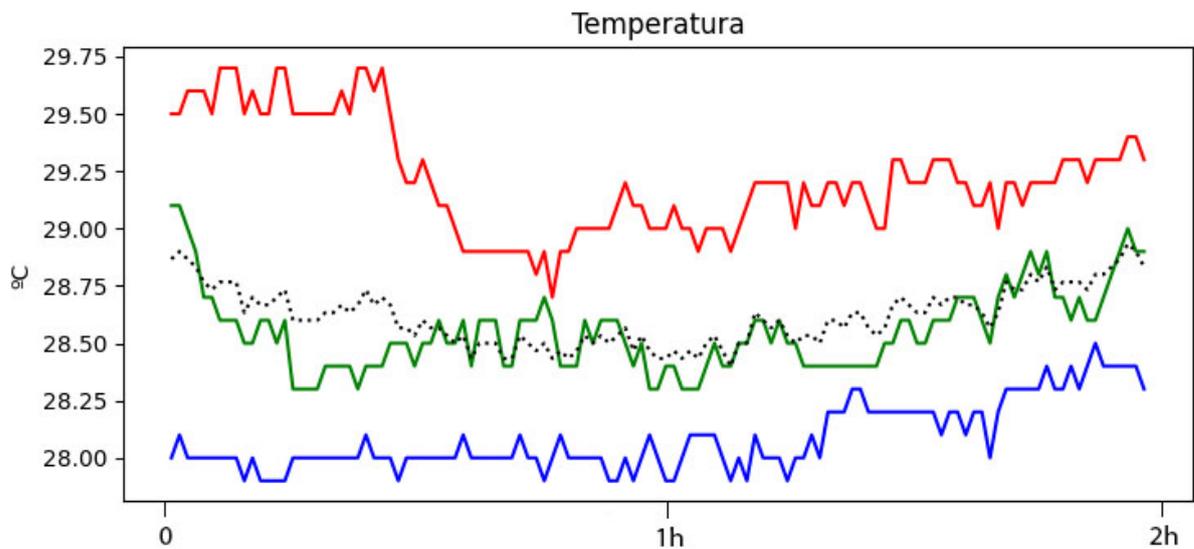


Figura 27 – Gráfico de temperatura em °C no Cenário 2.

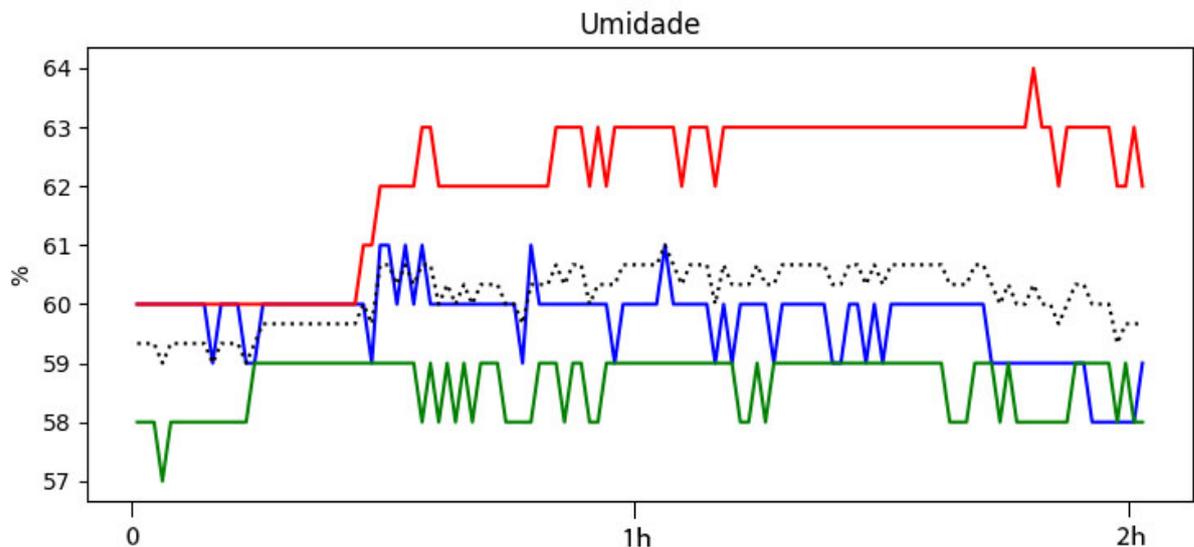


Figura 28 – Gráfico de umidade relativa do ar no Cenário 2.

5.3.3 Cenário 3

O Cenário 3 é o momento em que uma pessoa está presente na sala com janela e porta fechadas. Neste Cenário, através das Figuras 29, 30 e 31, podemos perceber a sensibilidade do sensor para uma única pessoa presente no ambiente, partindo de valores inferiores e aumentando até a casa de 65PPM durante o período observado. Vale ressaltar que a concentração continuou a subir após o período observado até estabilizar na casa de 100PPM. Este gráfico reforça a possibilidade da utilização do sensor para a detecção de pessoas bem como o monitoramento de capacidade de ambientes.

Neste Cenário a Temperatura também apresentou um leve aumento durante o período analisado, porém não é possível concluir que este aumento se deve ao fato de a

sala estar com presença humana. Da mesma forma, a Umidade apresentou-se com pequenas variações durante o período analisado.

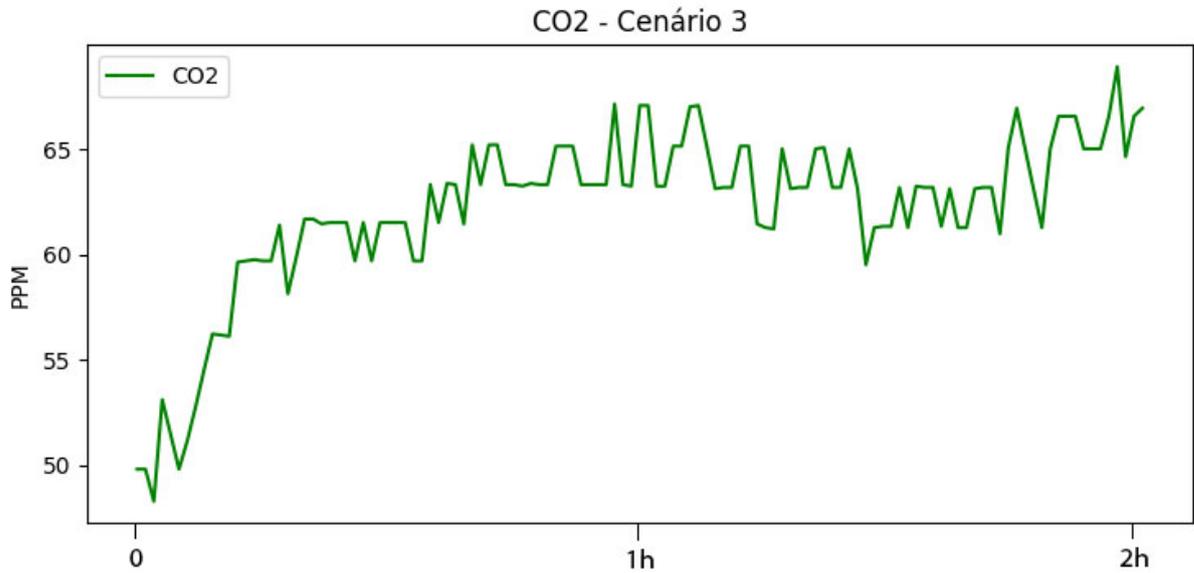


Figura 29 – Gráfico de concentração de CO2 em PPM no Cenário 3.

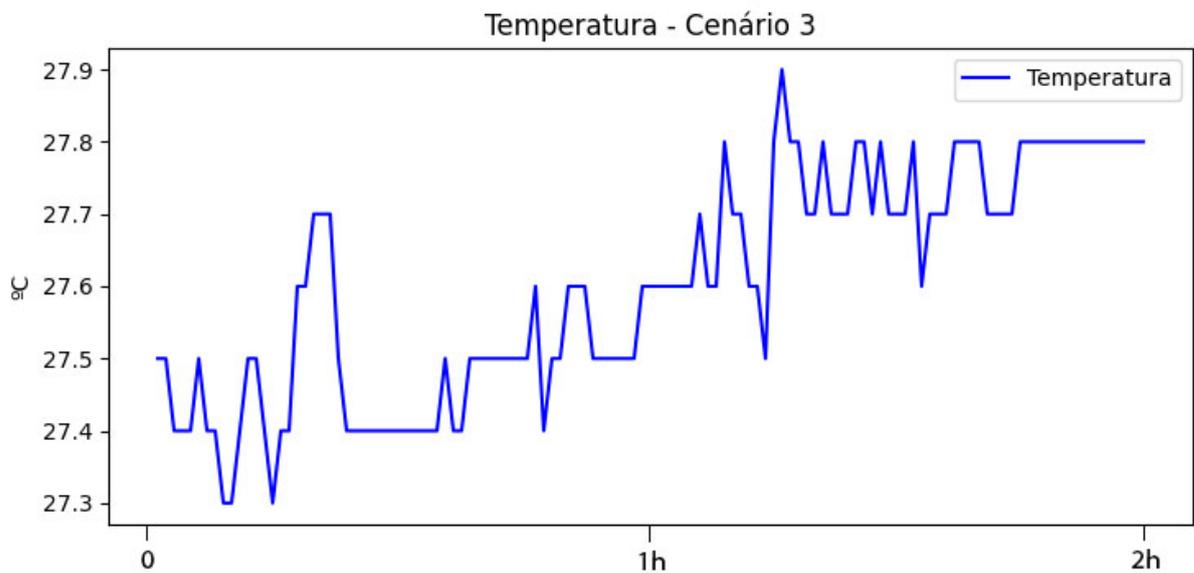


Figura 30 – Gráfico de temperatura em °C no Cenário 3.

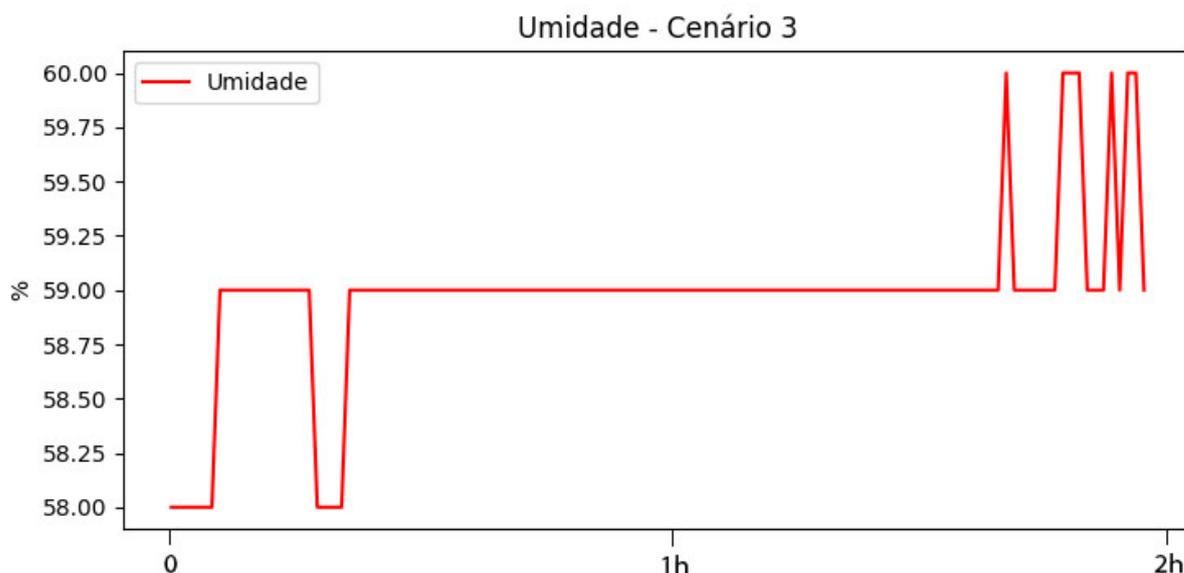


Figura 31 – Gráfico de umidade relativa do ar no Cenário 3.

5.3.4 Cenário 4

O Cenário 4 parte do momento em que a sala estava ocupada e completamente fechada com uma pessoa para o momento em que a janela e a porta são abertas permitindo uma passagem de ar. Percebemos pela Figura 32 como em poucos minutos a concentração de CO₂ diminui a valores semelhantes aos Cenários 1 e 2 onde a sala estava vazia. Isso comprova a eficiência da abertura de janelas que permitem uma circulação do ar.

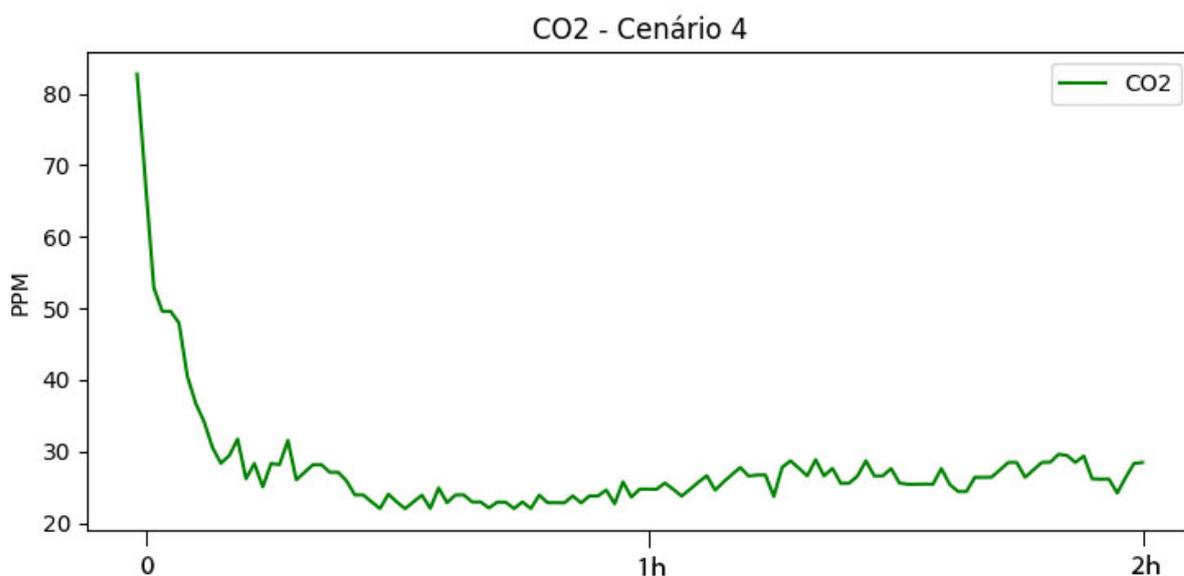


Figura 32 – Gráfico de concentração de CO₂ em PPM no Cenário 4.

A Temperatura e a Umidade, Figuras 33 e 34, apresentaram um leve aumento durante o período analisado.

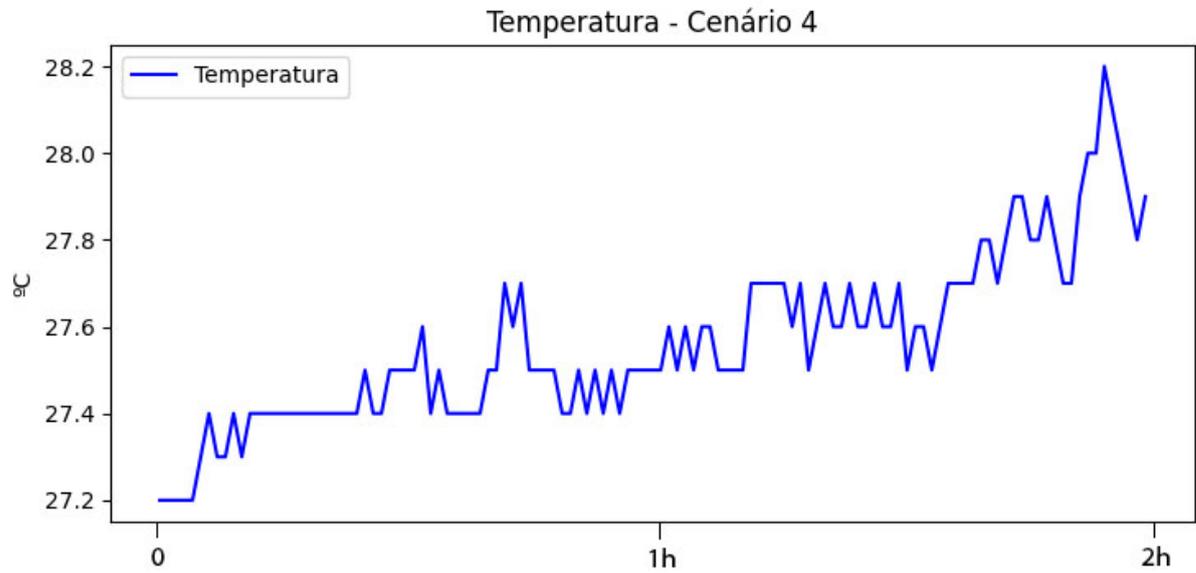
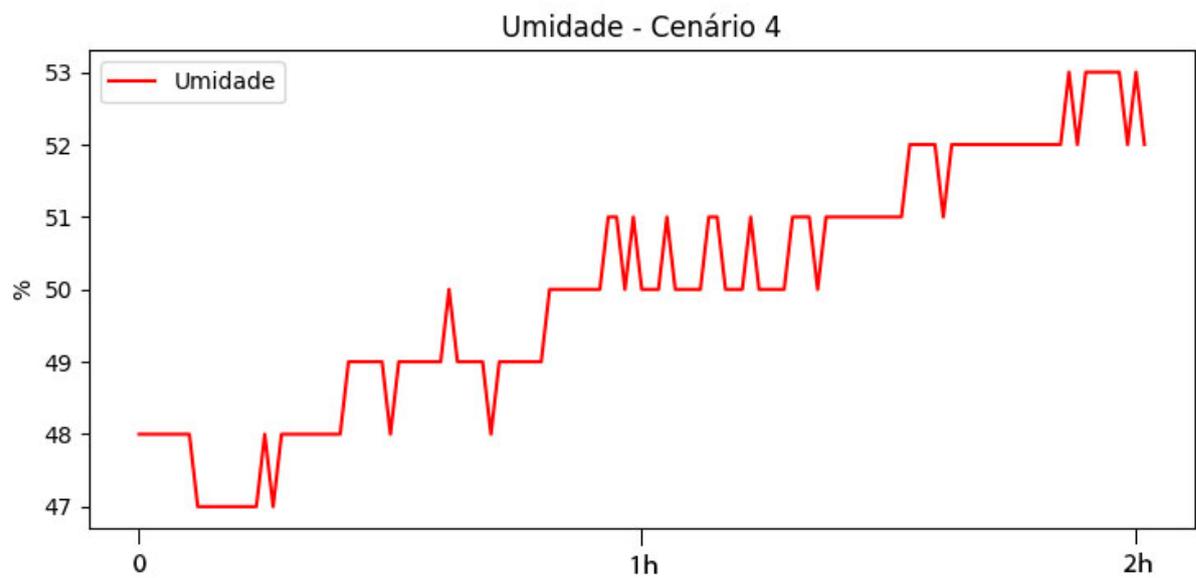
Figura 33 – Gráfico de temperatura em $^{\circ}\text{C}$ no Cenário 4.

Figura 34 – Gráfico de umidade relativa do ar no Cenário 4.

5.3.5 Cenário 5

O Cenário 5 apresenta o ambiente com uma pessoa e com sistema de ar-condicionado ligado. Neste Cenário percebemos pela Figura 35 que a variação de CO₂ se manteve dentro de uma faixa de valores esperada para uma pessoa, isto é, na casa de 60PPM.

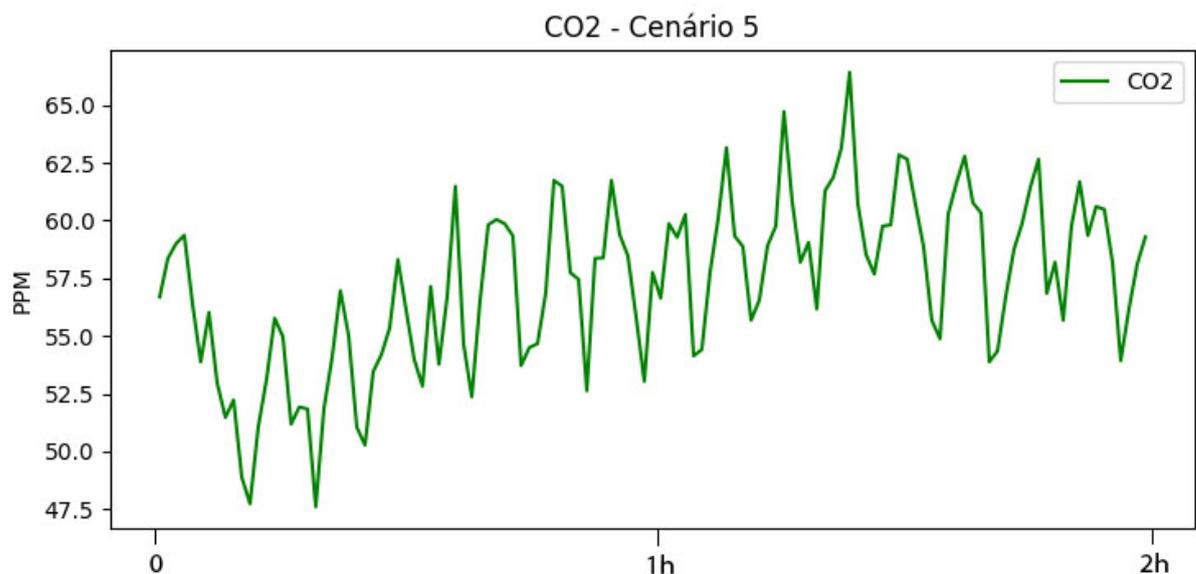


Figura 35 – Gráfico de concentração de CO₂ em PPM no Cenário 5.

Já os Gráficos de Temperatura e Umidade são mais interessantes neste Cenário. Percebemos nitidamente o decréscimo da temperatura no momento em que o ar-condicionado é ligado. O ar-condicionado estava configurado para manter 24°C e é isso que ocorre conforme constatado pelo nosso Gráfico 36. A Umidade, Figura 37, também apresenta variações bruscas com o aparelho ligado, tendendo a diminuir quando o compressor é acionado e voltando a subir quando o mesmo desliga.

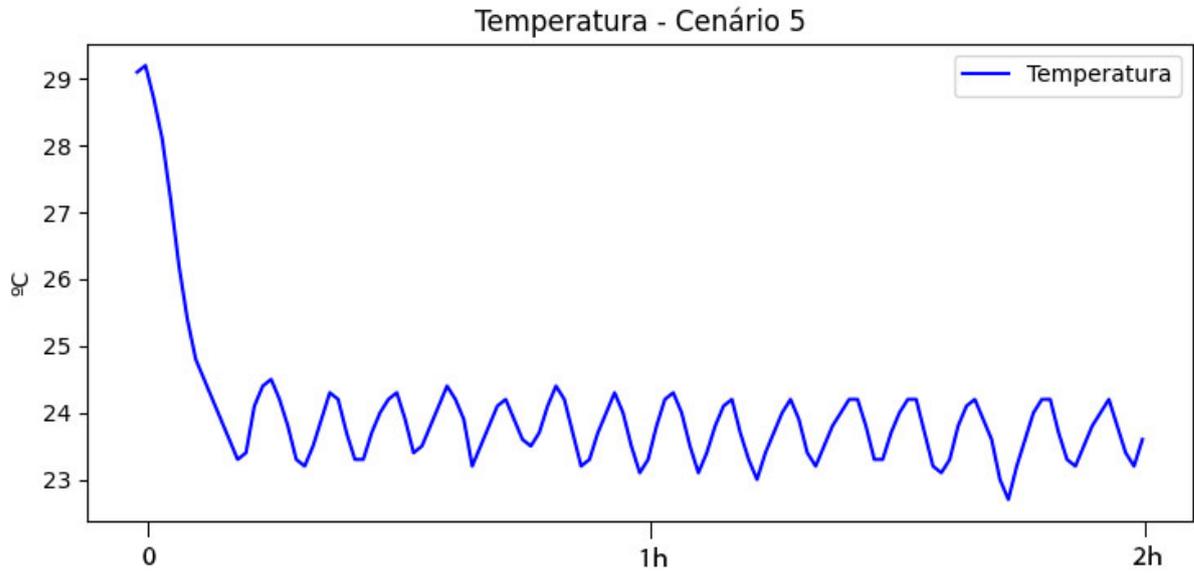
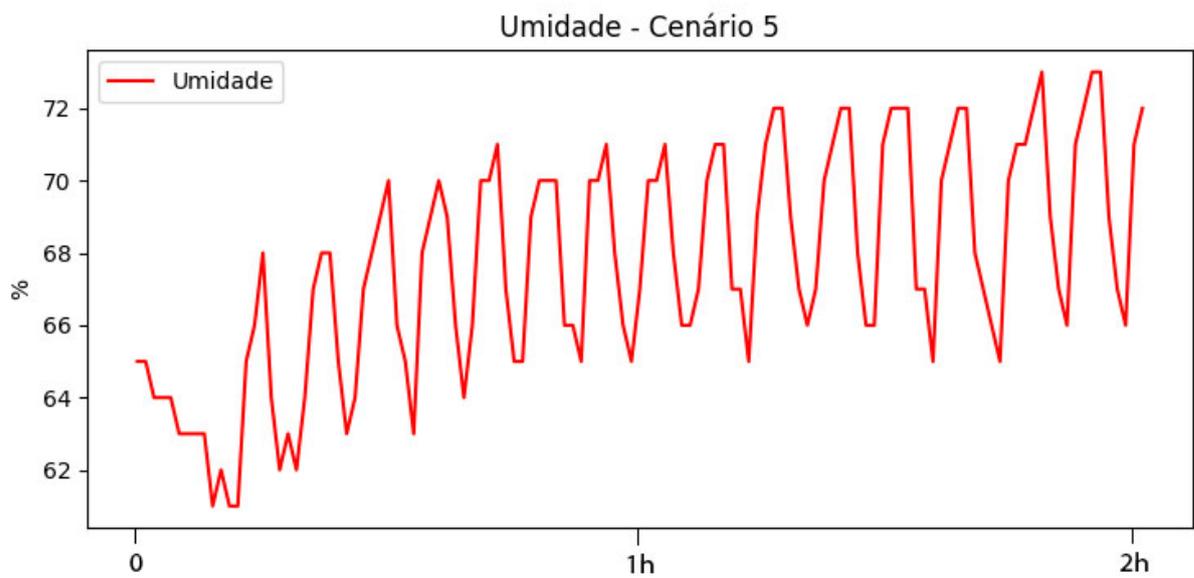
Figura 36 – Gráfico de temperatura em $^{\circ}\text{C}$ no Cenário 5.

Figura 37 – Gráfico de umidade relativa do ar no Cenário 5.

5.3.6 Considerações Finais

A realização do experimento com o auxílio de cenários previamente pensados auxiliaram na compreensão do funcionamento dos sensores utilizados neste trabalho. Todos os cenários apresentaram resultados próximos ao esperado.

A queda nos níveis de concentração de CO₂ quando há uma circulação de ar é um forte indicador para determinar se um ambiente é ou não propício para a transmissão de doenças respiratórias. É de conhecimento geral que ambientes fechados e densamente populados contribuem para a disseminação dessas doenças.

5.4 Definição do Cenário 2 - Ambiente Administrativo

Os ambientes administrativos são espaços usados exclusivamente para fins de trabalho ou negócios. Esses escritórios são normalmente usados por profissionais para realizar seu trabalho em conjunto com outros colegas. Dada essa característica, foram realizados experimentos em uma sala da Universidade Federal de Pelotas no período de 09 de Junho de 2022 a 13 de Outubro de 2022, totalizando 126 dias contínuos de medições. Seguindo o mesmo padrão de leituras do ambiente residencial, ou seja, uma leitura a cada minuto, foram obtidas 181.471 amostras.

As Figuras 38,39 e 40 apresentam todo o período de medição realizados pelo sistema proposto, sendo possível visualizar respectivamente as informações de CO₂, temperatura e umidade. Os gráficos representam no eixo x o tempo de medição que inicia às 00:00 até às 23:59 horas, mostrando o comportamento da variável ao longo de um dia de medição. Deste modo é possível visualizar as variações ocorridas durante os 126 dias monitorados.

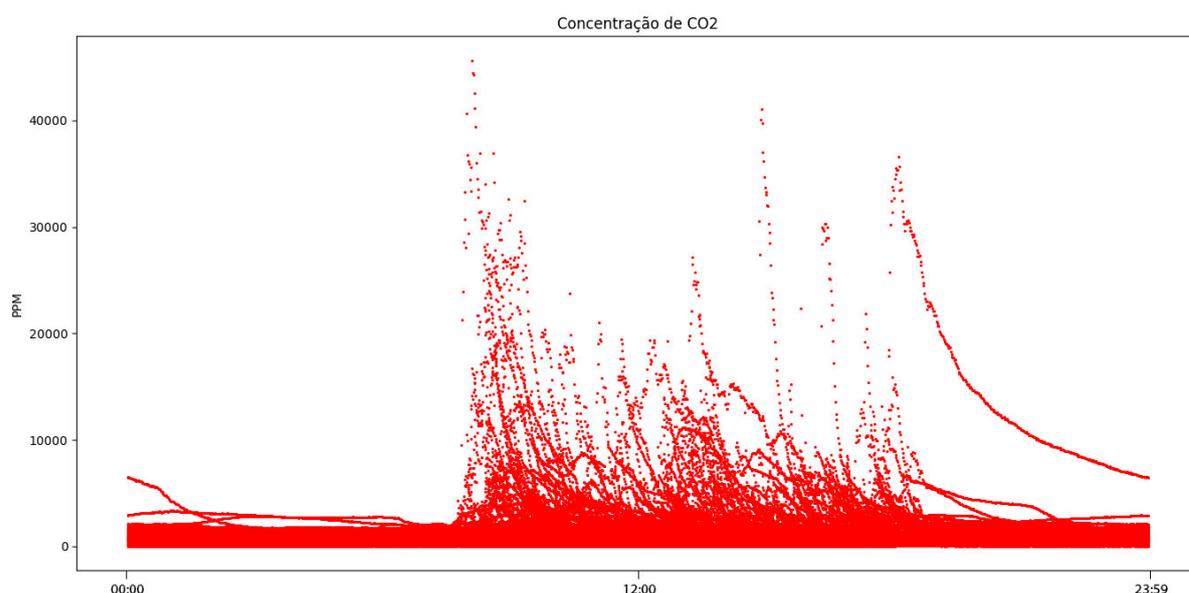


Figura 38 – Gráfico de Concentração de CO₂ no Ambiente Administrativo.

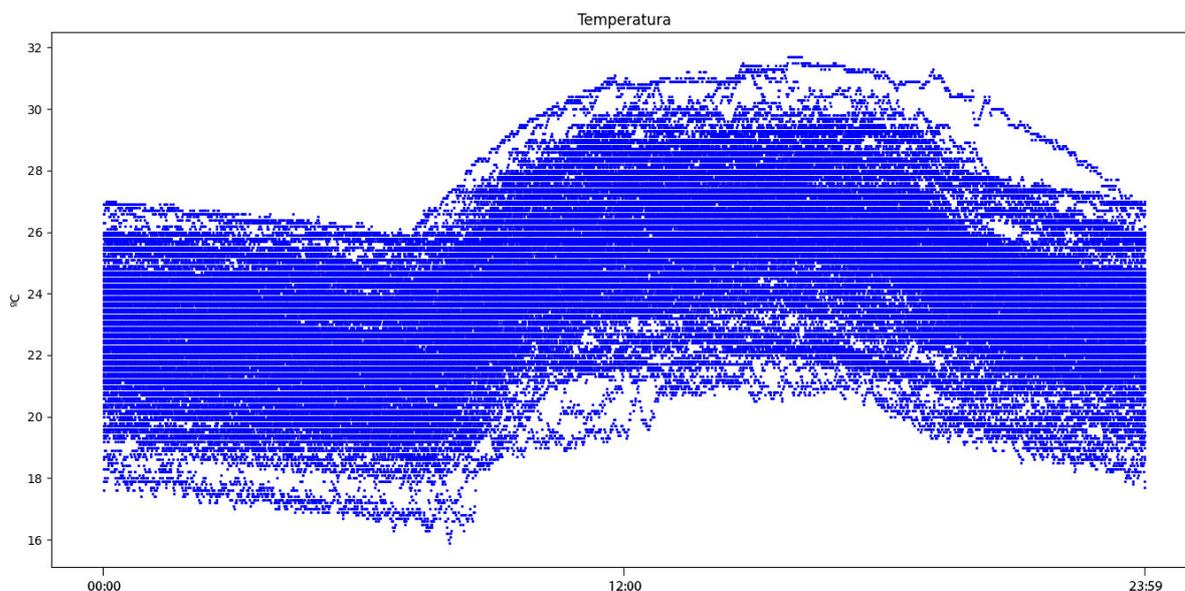


Figura 39 – Gráfico da Temperatura no Ambiente Administrativo.

O Gráfico da Figura 38 apresenta os diferentes níveis de Concentração de CO₂ ao longo dos dias. É percebido um aumento brusco por volta das oito horas da manhã, horário de início de expediente. Esse aumento indica a chegada de pessoas ao ambiente analisado. O Gráfico também nos apresenta altos índices de Concentração de CO₂, em alguns casos ultrapassando 40.000 PPM, o que é considerado altamente prejudicial à saúde segundo a OMS. Estes valores altos são reflexo de uma má circulação de ar no ambiente analisado, indicando que o ar deve ser renovado com maior frequência neste ambiente.

O Gráfico da Figura 39 apresenta as variações da temperatura ao longo dos dias analisados. Percebemos como há um aumento na temperatura também por volta das oito horas da manhã, indicando novamente a presença de pessoas. O repentino aumento de temperatura é resultado do uso intensificado de aparelhos de ar condicionado e aquecedores durante os dias frios compreendidos durante o período analisado.

Assim como no ambiente residencial, o Gráfico da Figura 40 mostra que não houve muitas variações significativas quanto a Umidade Relativa do Ar no ambiente administrativo. A leve queda apresentada pode ser um indicativo do uso de aparelhos de ar condicionado durante o horário de expediente.

5.4.1 Avaliação da Correlação entre as Variáveis Monitoradas

A correlação entre variáveis é uma medida estatística que indica a relação entre duas ou mais variáveis. Quando duas variáveis estão correlacionadas, uma delas pode afetar o comportamento da outra, o que pode ser útil para entender o comportamento de um sistema complexo. A correlação pode ser positiva, quando as variáveis aumentam ou diminuem juntas, ou negativa, quando uma variável aumenta enquanto

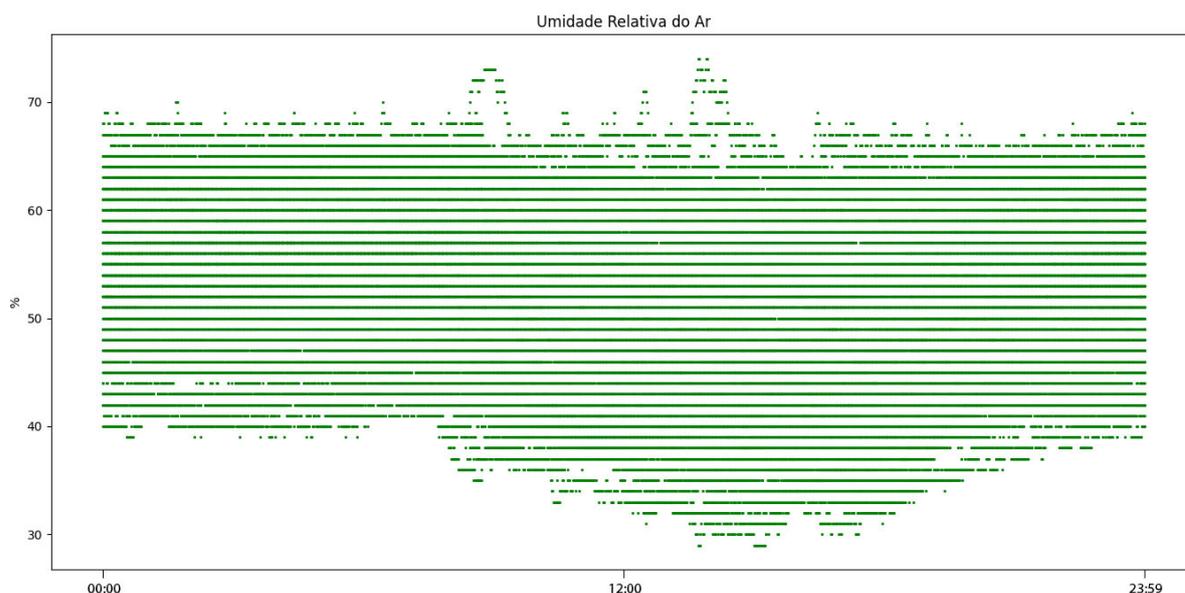


Figura 40 – Gráfico da Umidade Relativa do Ar no Ambiente Administrativo.

a outra diminui.

É importante lembrar que a correlação não implica causalidade, ou seja, apenas porque duas variáveis estão correlacionadas não significa necessariamente que uma cause efeito na outra.

Durante o período observado, foram escolhidos alguns dias aleatoriamente de modo a usar seus dados para análise segundo o método de Pearson (FIGUEIREDO FILHO; SILVA JÚNIOR, 2009) que avalia a correlação entre duas variáveis. Nestes dias o resultado do teste indicou que há pouca correlação entre CO₂ e os dados de temperatura e umidade. Quando analisamos a média geral dos dados o índice de correlação entre os dados aumenta significativamente. A Tabela 13 apresenta os índices de correlação encontrados.

Tabela 13 – Correlação entre CO₂, Temperatura e Umidade pelo Método de Pearson.

	12/07/2022	15/07/2022	20/09/2022	Média Geral
CO ₂ e Temperatura	-0.09719	0.2777245	-0.31611	0.615029
CO ₂ e Umidade	0.091605	-0.18446	0.706015	-0.56477
Temperatura e Umidade	-0.80659	-0.86172	-0.80926	-0.98944

5.5 Estudo de Caso do Projeto de Sensor Virtual

O objetivo de um sensor virtual é representar o comportamento de um sensor físico. Para que esse comportamento seja o mais fiel possível, há diferentes métodos e modelos para sua implementação conforme visto na Seção 2.7.

5.5.1 Modelo de Histórico de Amostras

O primeiro estudo de caso a ser avaliado é o Modelo baseado no histórico de amostras. Neste modelo definimos um vetor de amostras dos últimos 3 dias para estimar os valores mínimos, máximos e média. Caso ocorra alguma falha, utilizamos o vetor para auxiliar na correção, estimando o valor aproximado com base no seu histórico dos últimos dias.

As Figuras 41, 42 e 43 apresentam como exemplo a aplicação deste modelo para análise de Temperatura, Umidade e CO₂.

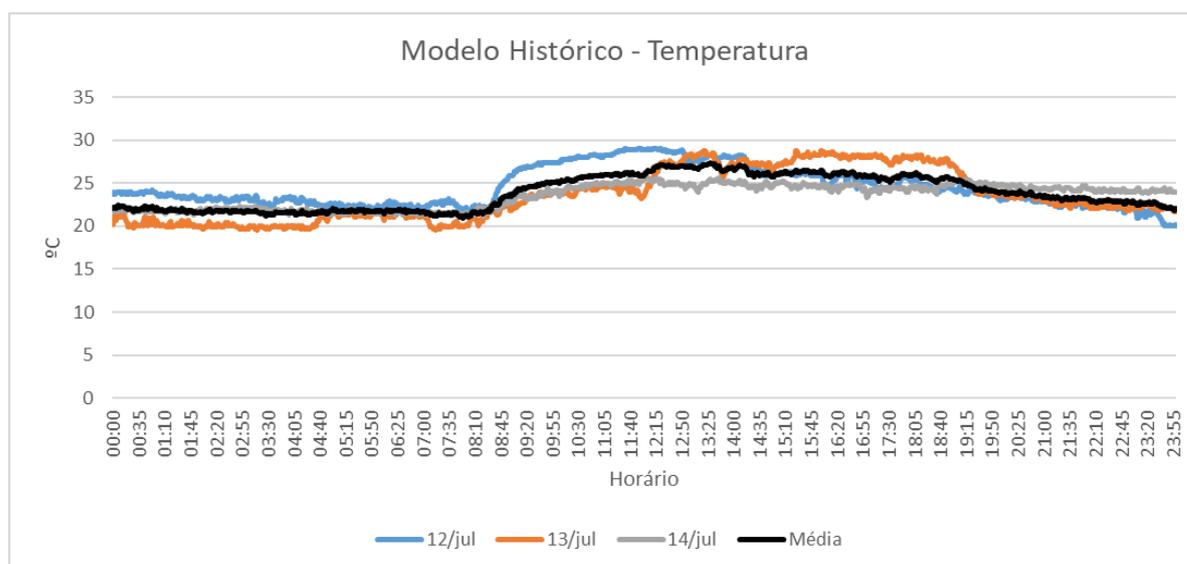


Figura 41 – Aplicação do Modelo Histórico no sensor de Temperatura.

Apesar de ser bastante simples a sua implementação, uma falha deste modelo a ser ponderada é quanto a sua aplicação no caso do sensor de CO₂ que apresenta picos de concentração em momentos variados que podem ser indicados como falhas. Também deve ser ressaltado quanto a mudanças abruptas na temperatura e umidade, que podem levar a interpretações incorretas numa análise futura.

Uma vez que temos definidas as faixas de valores máximos e mínimos, podemos usar como limite de tolerância os valores propostos na Tabela 7, ou seja, se o valor estiver fora em mais de 35%, será considerado como uma falha.

5.5.2 Modelo de Regressão Não-Linear (Polinomial de 2ª Ordem)

A regressão não-linear é uma técnica de análise de dados que prevê o valor de dados desconhecidos usando outro valor de dados relacionado e conhecido. Esta técnica modela matematicamente a variável desconhecida ou dependente e a variável conhecida ou independente como uma equação não-linear (GROSS; GROSS, 2003).

A regressão não-linear polinomial de 2ª ordem é uma técnica de análise de dados que utiliza uma equação polinomial de grau 2 para modelar a relação entre a variável dependente e variável independente, esta última neste caso sendo tempo. Essa

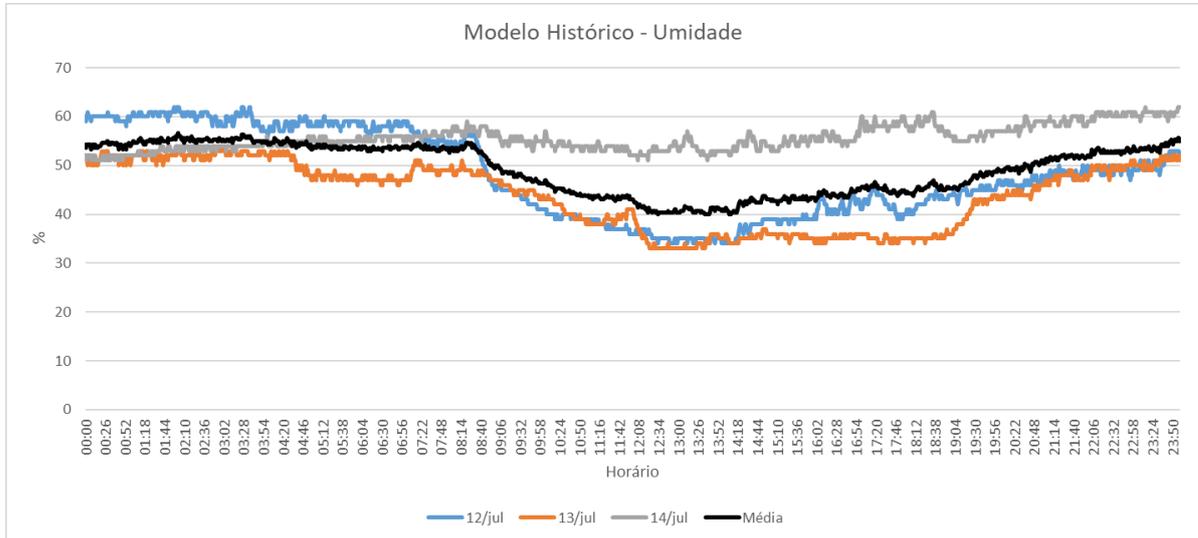


Figura 42 – Aplicação do Modelo Histórico no sensor de Umidade.

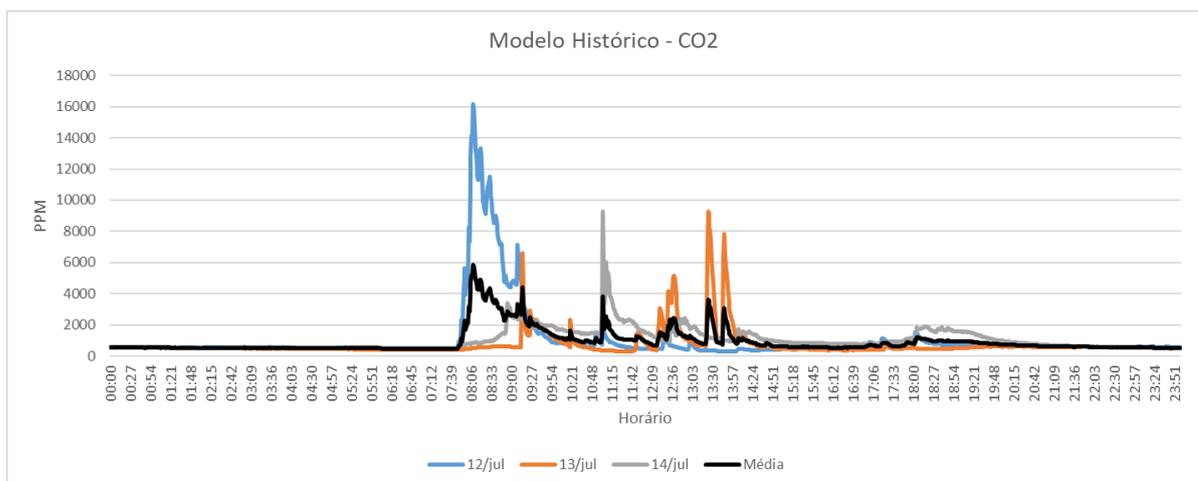


Figura 43 – Aplicação do Modelo Histórico no sensor de CO2.

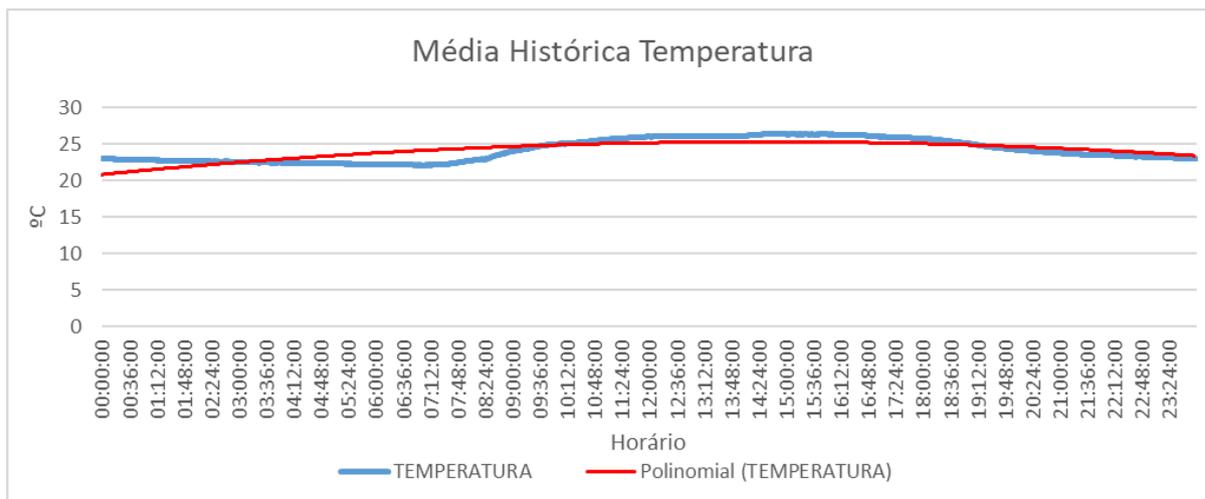


Figura 44 – Aplicação do Modelo Regressão Não-Linear Polinomial 2ª Ordem usando a média histórica no sensor de Temperatura.

técnica é uma extensão da regressão não-linear simples, que utiliza uma equação não-linear para modelar a relação entre duas variáveis. Na regressão não-linear polinomial de 2ª ordem, a equação é dada por:

$$y = a + bx + cx^2 \quad (3)$$

A Equação 3 permite modelar relações não-lineares entre as variáveis e é particularmente útil quando a relação não é bem descrita por uma função não-linear, como é o caso dos nossos sensores.

Analisamos a aplicação do Modelo de Regressão não-linear Polinomial 2ª Ordem para as três variáveis, temperatura, umidade e CO2. O modelo foi gerado baseando-se na média histórica, nas médias das curvas mínima e máxima e na aproximação da média dos últimos 3 dias.

5.5.2.1 Temperatura

Conforme percebemos pelas Figuras 44, 45 e 46 o Modelo de Regressão Não-Linear Polinomial de 2ª Ordem consegue realizar uma aproximação em alguns pontos, porém quando há variações mais bruscas de temperatura o modelo não consegue seguir o comportamento.

Para ter uma aproximação mais exata, teríamos que aumentar a ordem do polinômio, porém quanto maior for a ordem, maior é a necessidade computacional para realizar a aproximação.

5.5.2.2 Umidade

O comportamento do sensor de umidade é semelhante com o sensor de temperatura. Percebemos pelas Figuras 47, 48 e 49 que a aproximação ocorre em alguns

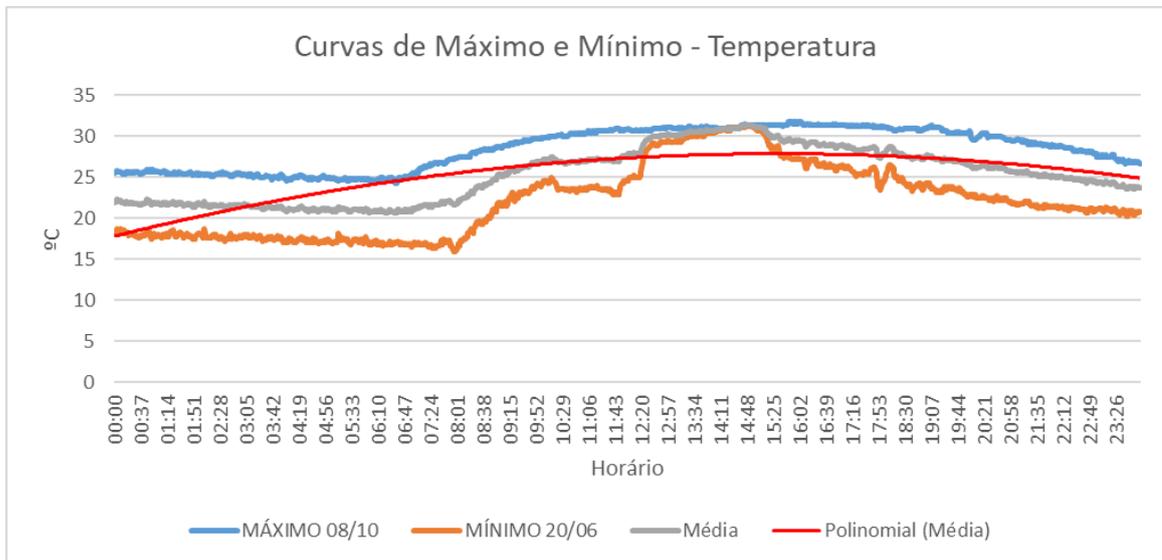


Figura 45 – Aplicação do Modelo Regressão Não-Linear Polinomial 2ª Ordem usando os máximos e mínimos no sensor de Temperatura.

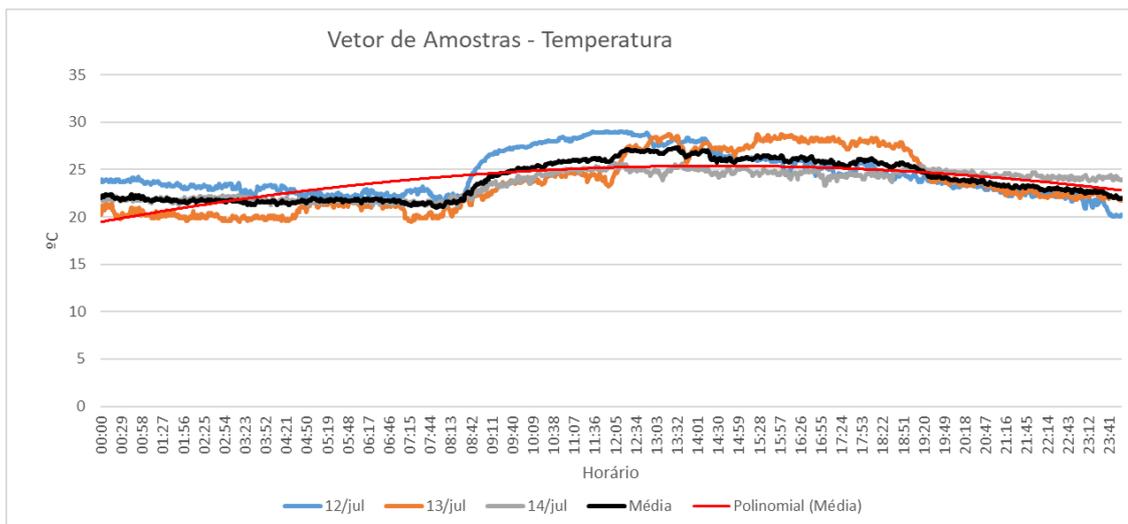


Figura 46 – Aplicação do Modelo Regressão Não-Linear Polinomial 2ª Ordem usando o vetor de amostras no sensor de Temperatura.

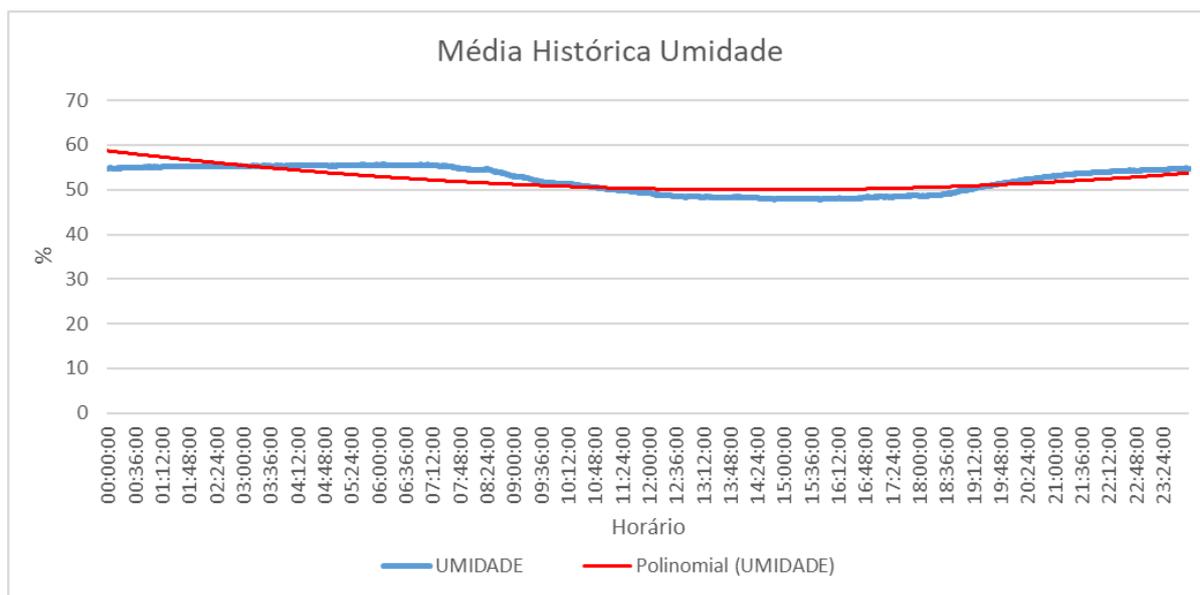


Figura 47 – Aplicação do Modelo Regressão Não-Linear Polinomial 2ª Ordem usando a média histórica no sensor de Umidade.

pontos porém em mudanças mais significativas o modelo deixa a desejar já que não consegue acompanhar as variações ocorridas ao longo do dia.

5.5.2.3 CO₂

O comportamento do Modelo Polinomial de 2ª Ordem no sensor de CO₂ pode ser observado como sendo ainda mais não-linear. Devido as oscilações constantes nos níveis de CO₂ no ambiente monitorado, constatamos pelas Figuras 50, 51 e 52 que o modelo gerado não consegue se aproximar aos dados obtidos pelo sensor, o que demonstra que para esta aplicação, o modelo não é o mais adequado.

5.5.3 Modelo de Média Móvel

O Modelo da Média Móvel é a implementação do Método de Janela Flutuante. Essa técnica consiste em calcular a média dos valores de uma série de amostras em um intervalo de tempo específico, que pode ser fixo ou variável. A média móvel pode ser usada para suavizar uma série temporal, tornando-a mais fácil de visualizar e interpretar. Além disso, a média móvel também pode ser usada para identificar tendências em uma série, como uma tendência de subida ou descida.

Para usar como uma forma alternativa de avaliar as amostras obtidas pelos sensores físicos é aplicado o modelo de média móvel nas médias históricas de Temperatura, Umidade e CO₂ conforme apresentado nas Figuras 53, 54 e 55.

Como esperado, este modelo obtido a partir da Média Móvel produz uma estimativa de valores que mais se aproxima da linha de histórico de amostras a ser seguida. Isso ocorre pois a tendência é recalculada a cada nova ocorrência, o que mantém a fidelidade da amostra.

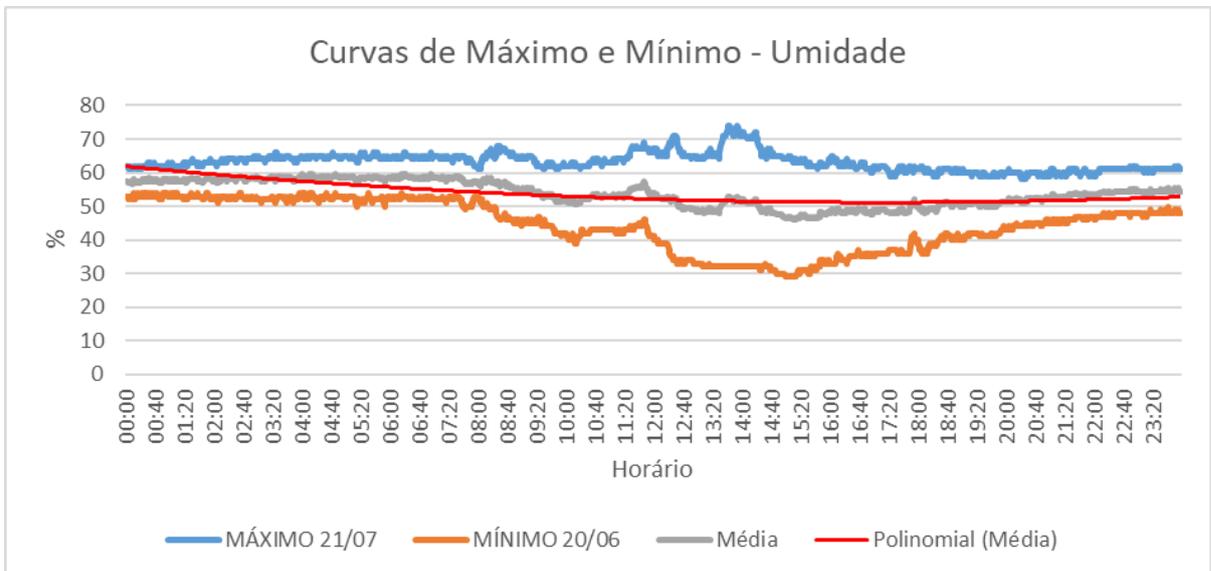


Figura 48 – Aplicação do Modelo Regressão Não-Linear Polinomial 2ª Ordem usando os máximos e mínimos no sensor de Umidade.

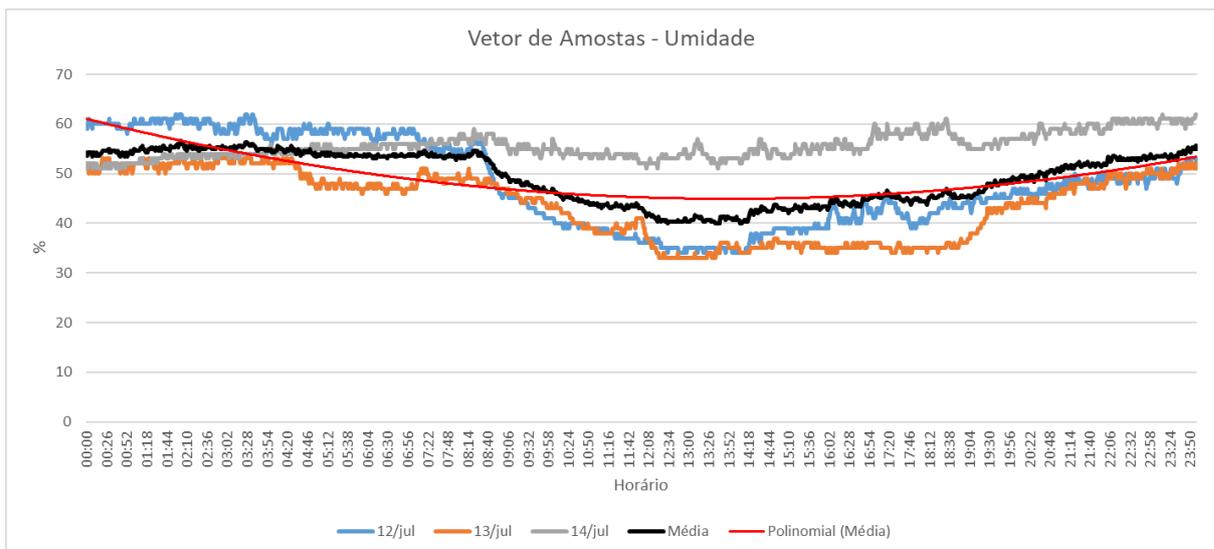


Figura 49 – Aplicação do Modelo Regressão Não-Linear Polinomial 2ª Ordem usando o vetor de amostras no sensor de Umidade.

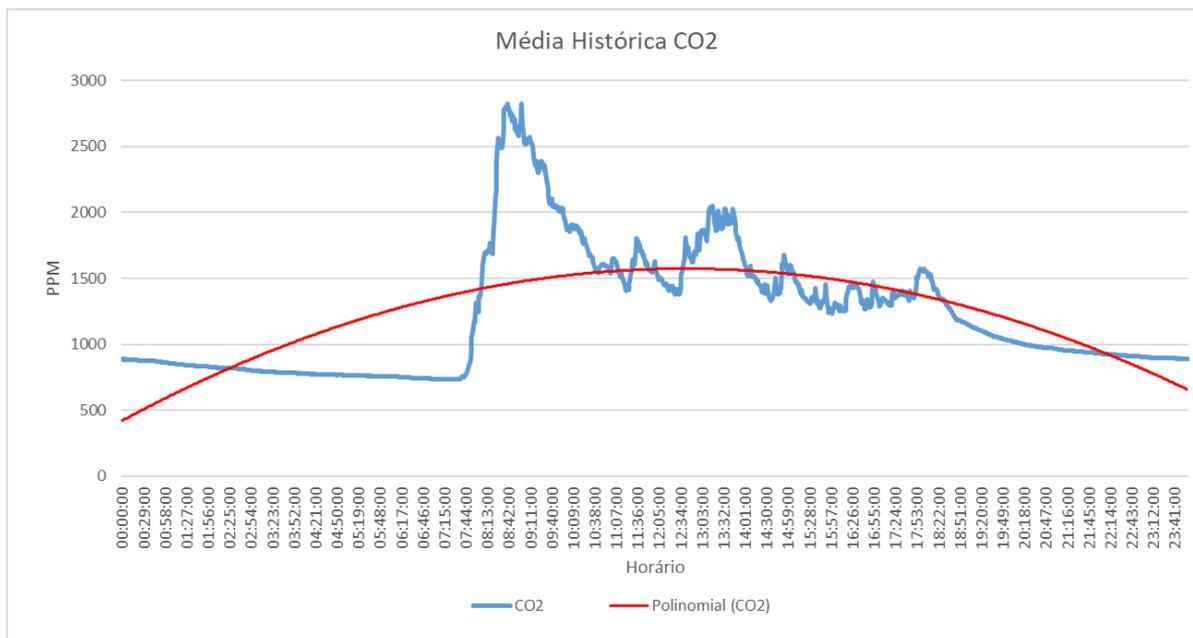


Figura 50 – Aplicação do Modelo de Regressão Não-Linear Polinomial 2ª Ordem usando a média histórica no sensor de CO2.

5.6 Avaliação de Falhas

A avaliação de falhas em modelos de sensor virtual é um aspecto crítico da manutenção preditiva e da operação segura de sistemas complexos. Portanto, nesta Seção são apresentados alguns experimentos utilizando os modelos definidos anteriormente e o comportamento dos sensores virtuais mediante a ocorrência de falhas.

Para testar o tratamento de falhas foram forçados erros nas leituras dos sensores de um conjunto de dados, isto é, valores foram zerados para simular uma falha de sensor. A identificação da falha pode ocorrer através do método proposto na Seção 4.4, que faz uso da janela flutuante. Ao identificar um valor anormal, o algoritmo de tratamento entra em ação.

5.6.1 Experimentos com o Modelo de Histórico de Amostras

Como visto anteriormente, o Modelo Histórico usa um vetor com amostras dos últimos três dias de amostras, o que pode ser uma solução simples para o tratamento de erros.

Para testar a funcionalidade do modelo, utilizamos como parâmetro os dias 12, 13 e 14 de Julho de 2022 para composição do vetor de amostras. No dia seguinte, 15 de Julho, introduzimos algumas falhas entre 06:00-06:59 e 13:00-13:59.

Pela Figura 56 podemos perceber como o modelo atuou minimizando os efeitos dos erros durante o período para o caso da Temperatura, onde obteve o seu melhor resultado.

Como podemos perceber pelas Figuras 57 e 58 o modelo não obteve bons resul-

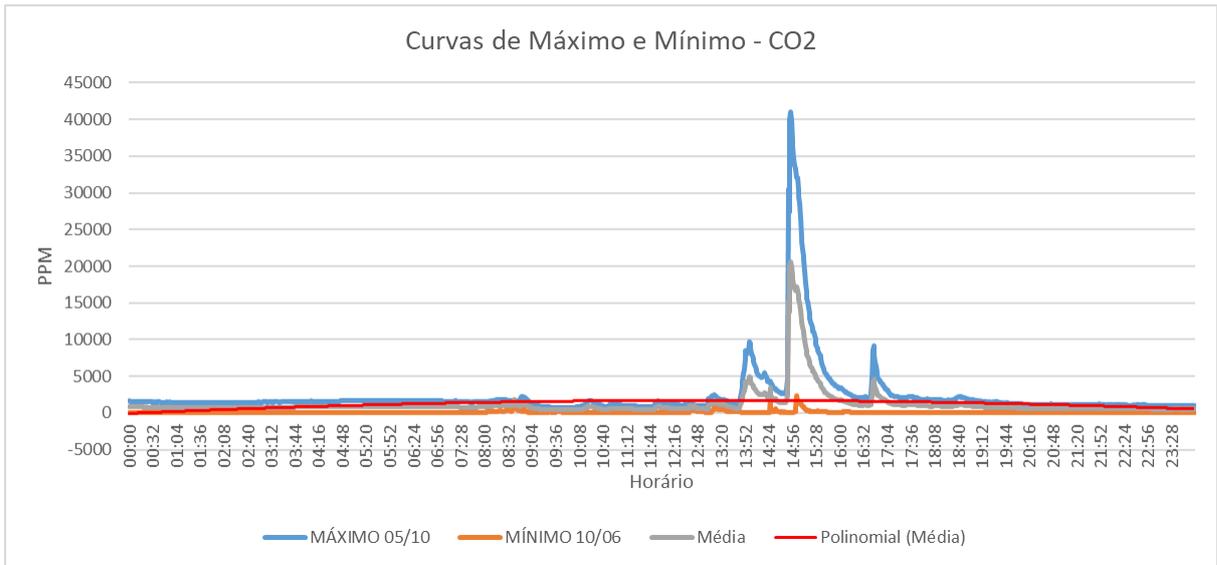


Figura 51 – Aplicação do Modelo de Regressão Não-Linear Polinomial 2ª Ordem usando os máximos e mínimos no sensor de CO2.

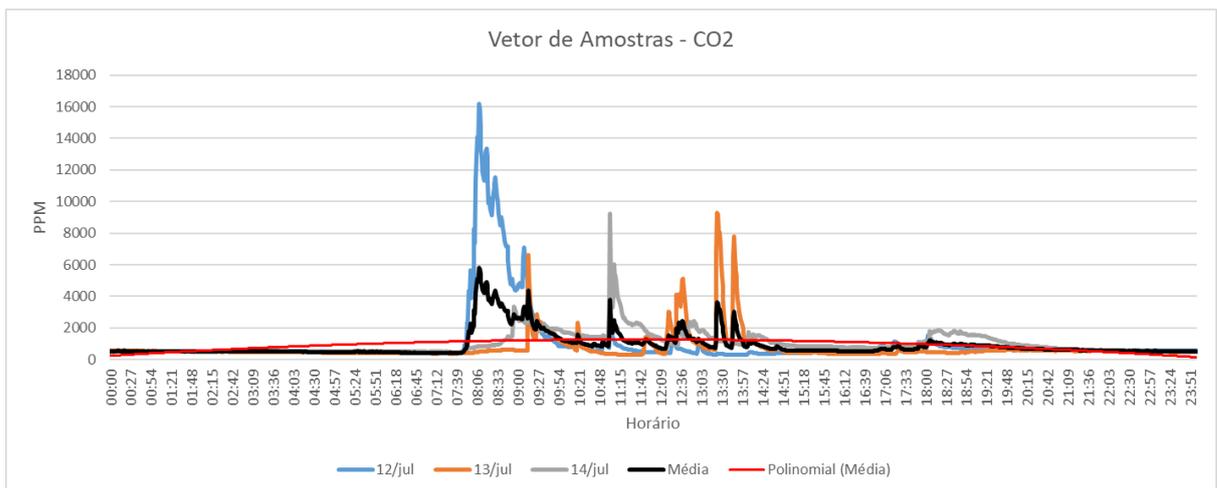


Figura 52 – Aplicação do Modelo de Regressão Não-Linear Polinomial 2ª Ordem usando o vetor de amostras no sensor de CO2.

tados para os casos de Umidade e CO2.

5.6.2 Experimentos com o Modelo de Regressão Não-Linear

O Modelo de Regressão Não-Linear tenta aproximar os valores desejados através de uma equação de segundo grau. Nos casos dos sensores de Temperatura e Umidade a aproximação chega próxima dos valores reais conforme Figuras 59 e 60, porém no caso do sensor de CO2, Figura 61, essa aproximação não se torna tão precisa visto que é uma medida que oscila com uma grande frequência e não há como ter uma previsão precisa dos valores de um determinado instante de tempo.

Como mencionado anteriormente, para ter uma aproximação mais exata, teríamos que aumentar o grau do polinômio, porém ao aumentar o grau, aumentamos também a complexidade e a necessidade de um maior poder computacional. Em nosso cenários

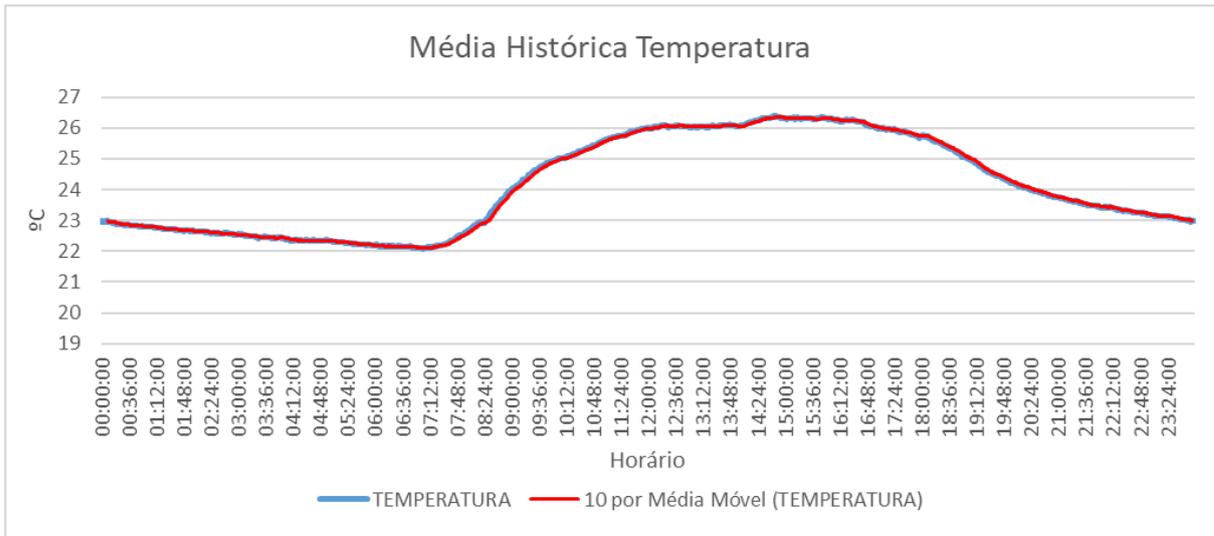


Figura 53 – Aplicação do Modelo de Média Móvel usando o histórico de Temperatura.

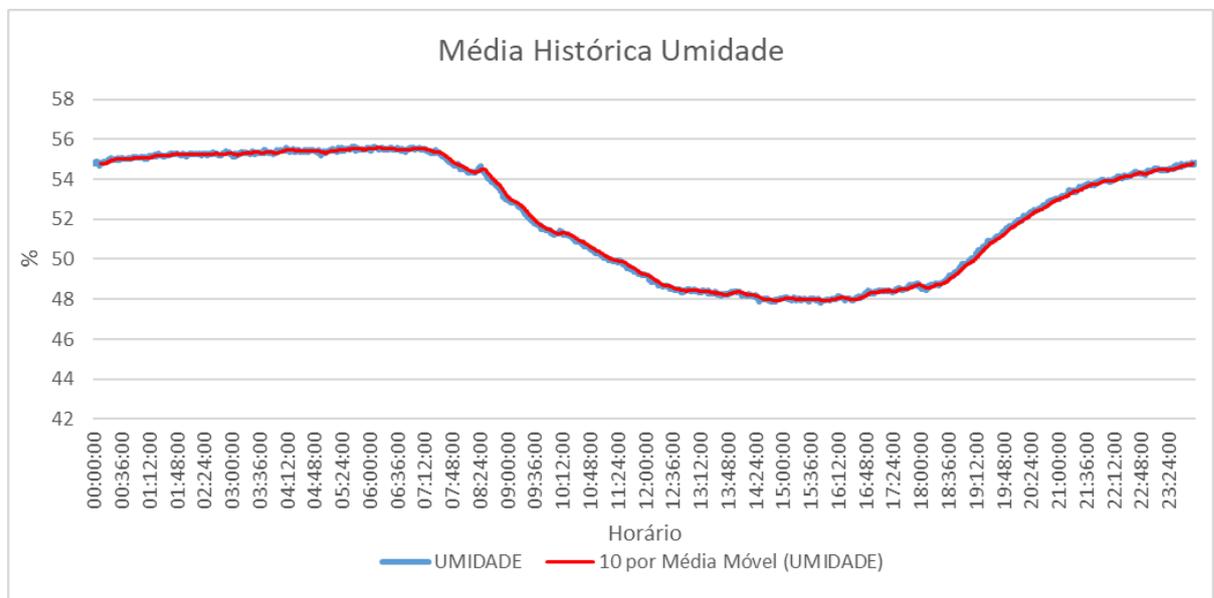


Figura 54 – Aplicação do Modelo de Média Móvel usando o histórico de Umidade.

estamos focados em soluções de baixo custo e que sejam ultra-leves, por isso, a necessidade da utilização de um polinômio de grau 2.

5.6.3 Experimentos com o Modelo de Média Móvel

O Modelo de Média Móvel, dada as suas características, consegue uma aproximação muito mais justa dos cenários reais, conforme Figuras 62, 63 e 64, isto ocorre pois são levados em consideração os valores que vinham ocorrendo em momentos prévios ao cenário de falha. Este modelo tende a conduzir os valores anteriores durante o cenário de falha. É por este motivo, que vemos uma tendência de continuação durante o período observado. Caso a falha persista, uma linha se formará indicando que os novos valores podem não ser os corretos e uma substituição de hardware pode ser necessária.

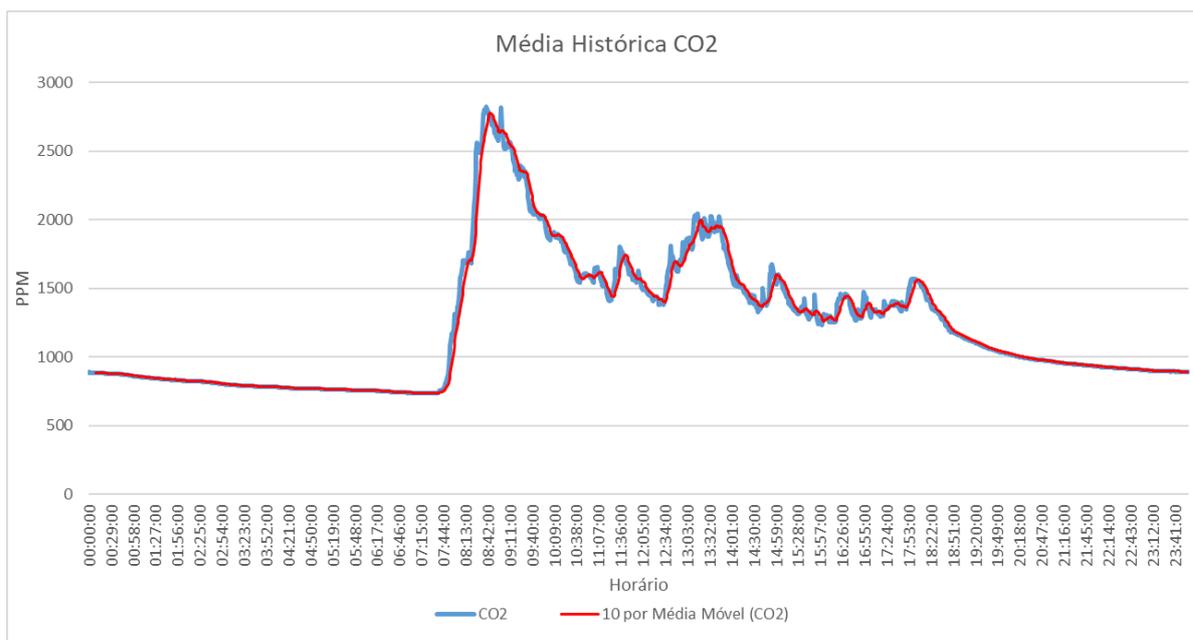


Figura 55 – Aplicação do Modelo de Média Móvel usando o histórico de CO2.

5.7 Modelo com Método de Peniak

Diferente dos modelos analisados anteriormente, o Modelo de Peniak faz uso da replicação de sensores para corrigir falhas. Peniak utiliza três sensores para produzir um sensor virtual. Seu método consiste em identificar a menor diferença de valores produzidos por três sensores e utilizar a média dos dois que apresentaram a menor diferença. Este método é bastante simples porém requer o uso de mais hardware.

Como o Método de Peniak faz uso de múltiplos sensores para gerar um virtual, replicamos dois sensores a partir dos dados adquiridos, utilizando um delta aleatório em cada um para poder realizar os testes e comparações. As Figuras 65, 66 e 67 apresentam os Gráficos com os valores simulados para utilização do Método de Peniak.

Nesta Seção, foi realizado alguns testes com o método proposto por Peniak e na sequência é apresentada a comparação com os modelos vistos anteriormente.

5.7.1 Inserindo Falhas no Método de Peniak

Para analisar como o Modelo atua em casos de falhas, vamos introduzir as mesmas falhas inseridas nos outros modelos. Como o método de Peniak faz uso da replicação de sensores, das 06:00 às 06:59 vamos inserir falhas no sensor 2 e das 13:00 às 13:59 vamos inserir falhas no sensor 3.

As Figuras 68, 69 e 70 apresentam os resultados dos gráficos aplicando falhas em sensores distintos no Modelo de Peniak. Como percebe-se, o comportamento final não é alterado mesmo realizando a inserção de falhas, isto ocorre por conta das falhas terem acontecido em momentos distintos, ou seja, a falha é solucionada com o

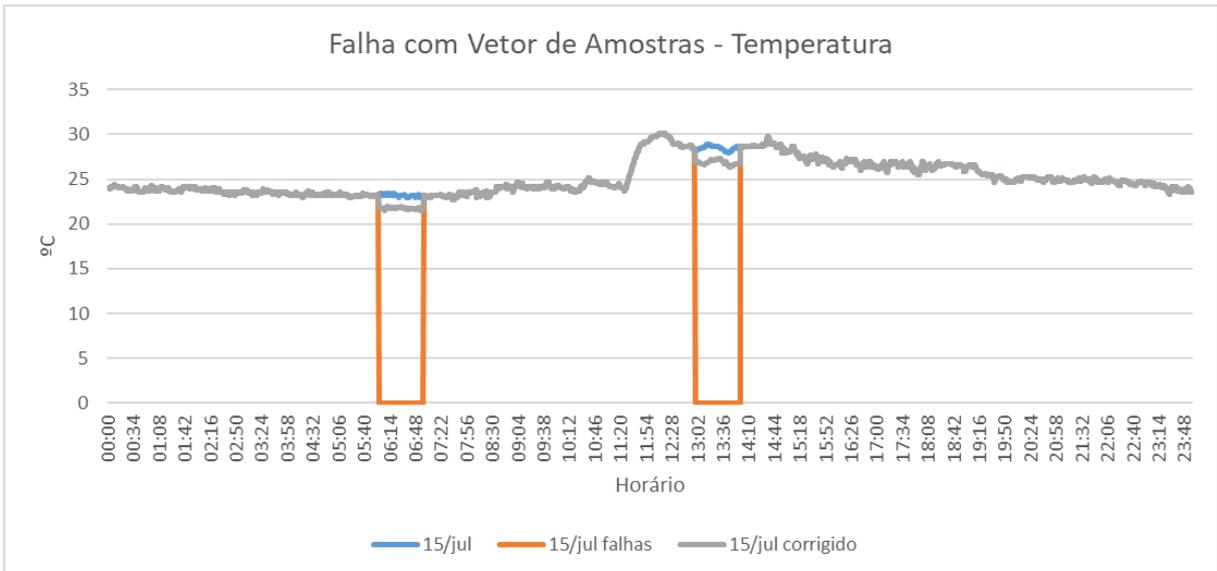


Figura 56 – Aplicação do Modelo Histórico usando vetor de amostras para corrigir falhas no sensor de temperatura.

isolamento daquele sensor.

5.8 Avaliações de Erros dos Modelos

A avaliação de erros é uma técnica utilizada para mensurar o desempenho dos modelos na tarefa de predição. O objetivo da avaliação de erros é estimar o quão bem o modelo está realizando suas predições em relação aos valores reais do conjunto de dados. Uma das métricas de avaliação mais comuns é o erro quadrático.

O erro quadrático é uma medida que calcula a média dos quadrados das diferenças entre os valores preditos pelo modelo e os valores reais do conjunto de dados. Essa medida é útil para avaliar a magnitude dos erros cometidos pelo modelo, pois ela aumenta proporcionalmente ao quadrado das diferenças entre o valor real e o valor predito.

A Equação 4 apresenta o cálculo do erro quadrático usado nas avaliações:

$$MSE = \frac{1}{N} \times \sum (y - \psi)^2 \quad (4)$$

onde MSE é o erro quadrático médio, N é o número de amostras no conjunto de dados, y é o valor real do conjunto de dados e ψ é o valor predito pelo modelo.

A Tabela 14 apresenta os valores encontrados em cada modelos para cada sensor nos cenários de falhas.

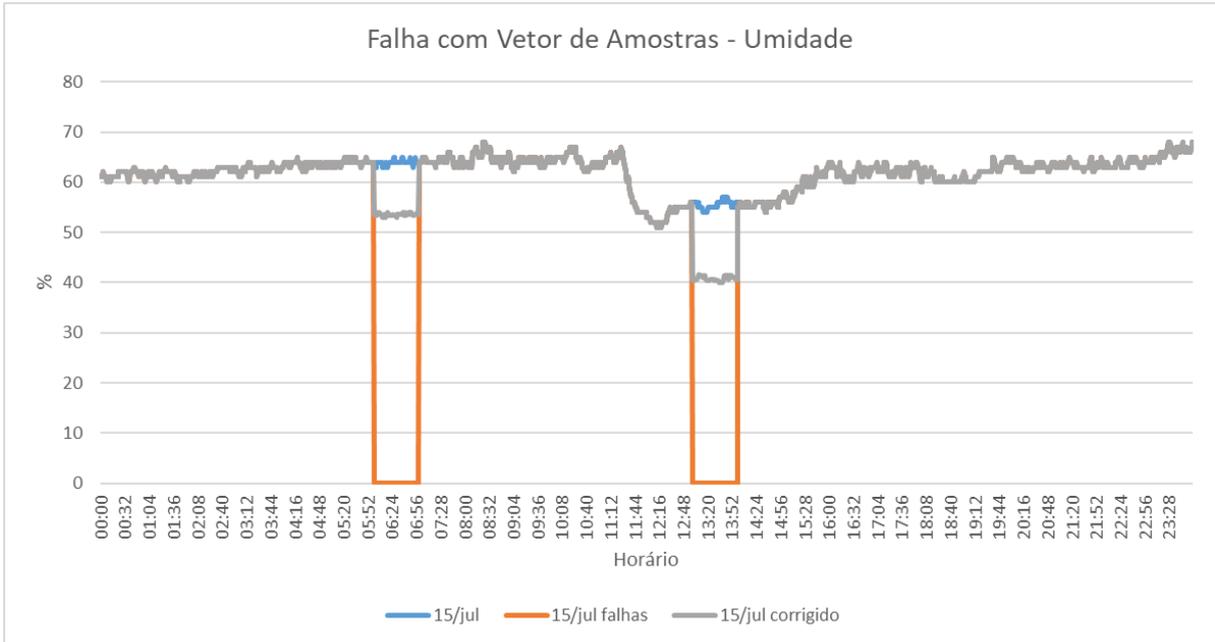


Figura 57 – Aplicação do Modelo Histórico usando vetor de amostras para corrigir falhas no sensor de umidade.

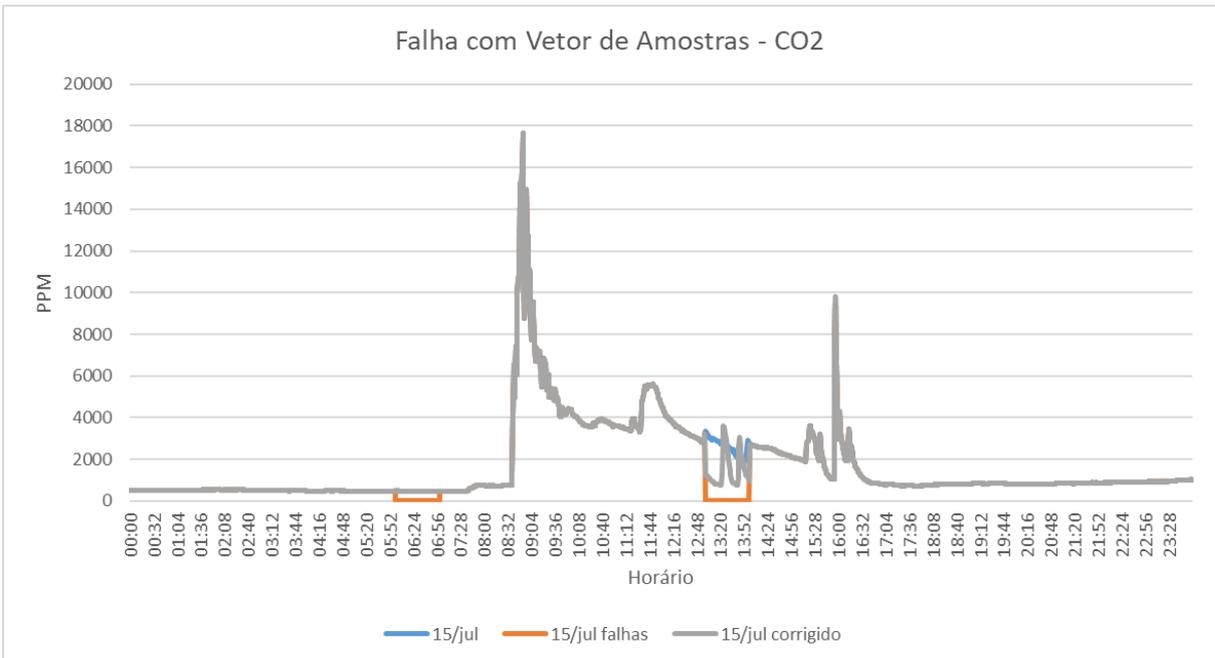


Figura 58 – Aplicação do Modelo Histórico usando vetor de amostras para corrigir falhas no sensor de CO2.

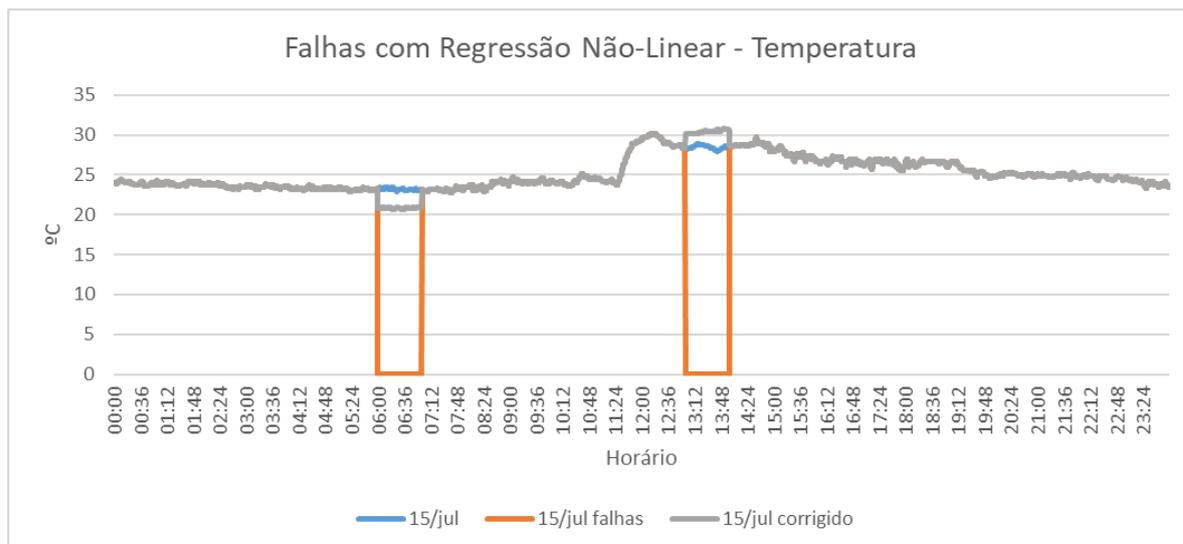


Figura 59 – Aplicação do Modelo de Regressão Não-Linear Polinomial de 2ª Ordem para corrigir falhas no sensor de temperatura.

Tabela 14 – Comparação entre os Modelos utilizando o cálculo do erro quadrático nos cenários de falhas.

Modelo	MSE Temperatura	MSE Umidade	MSE CO2
Peniak	0.16487	0.161051	181.3371
Média Móvel	0.004033	0.039444	11931.67
Regressão Não-Linear	0.397797	2.463124	126757.42
Vetor de Amostras	0.197589	13.63248	93387.63

5.9 Considerações Sobre a Temperatura e Umidade

Os gráficos de temperatura e umidade podem apresentar informações relevantes quanto ao ambiente em que os usuários estão inseridos. Estes sensores são amplamente utilizados em diversas aplicações e através deles conseguimos observar momentos em que o aparelho de ar condicionado esteve em funcionamento, alterando de forma direta a temperatura e a umidade. Este é um indicativo de momentos em que a sala esteve com presença humana, já que não faria sentido deixar o aparelho em funcionamento sem a real necessidade. Através dessa análise, é possível identificar momentos em que a sala esteve ocupada, porém o sensor de CO2 não detectou variações bruscas, indicando que poucas pessoas estavam presentes.

A variação significativa entre uma determinada faixa de valores também pode ser indicativa do tipo de aparelho de ar condicionado, como é o caso de um aparelho convencional que possui um controlador do tipo *Bang-Bang*. Isso significa que quando a temperatura atinge o nível desejado, o compressor é desligado e, quando a temperatura começa a subir novamente, o compressor é acionado novamente, mantendo assim a temperatura da sala dentro de uma faixa confortável para os usuários.

Através da Tabela 14 percebemos que todos os modelos apresentam valores pe-

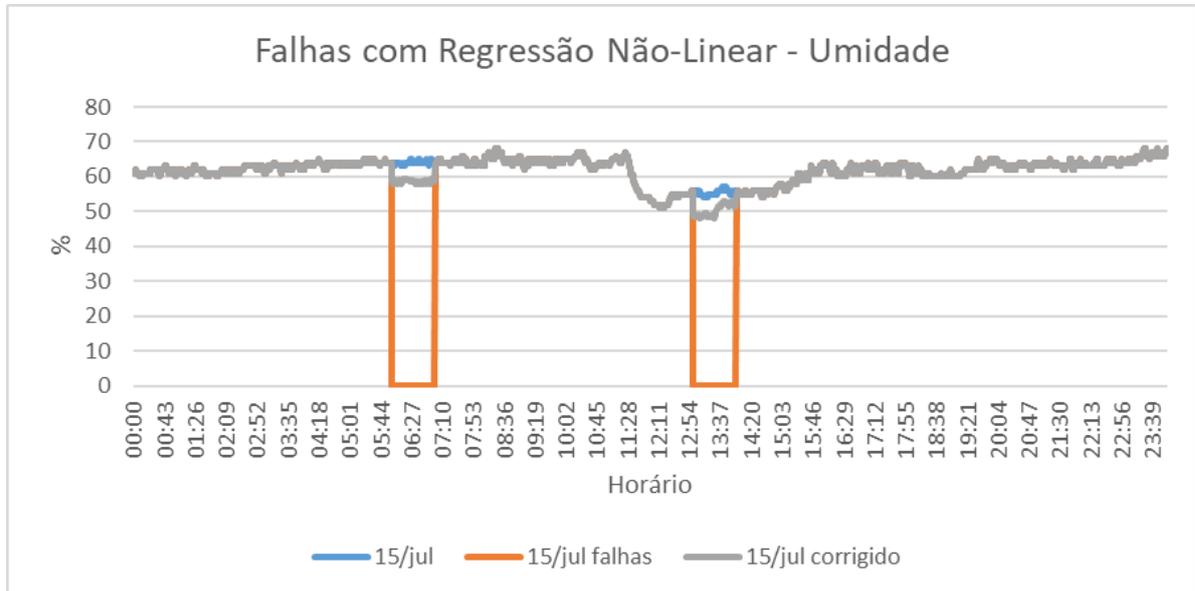


Figura 60 – Aplicação do Modelo de Regressão Não-Linear Polinomial de 2ª Ordem para corrigir falhas no sensor de umidade.

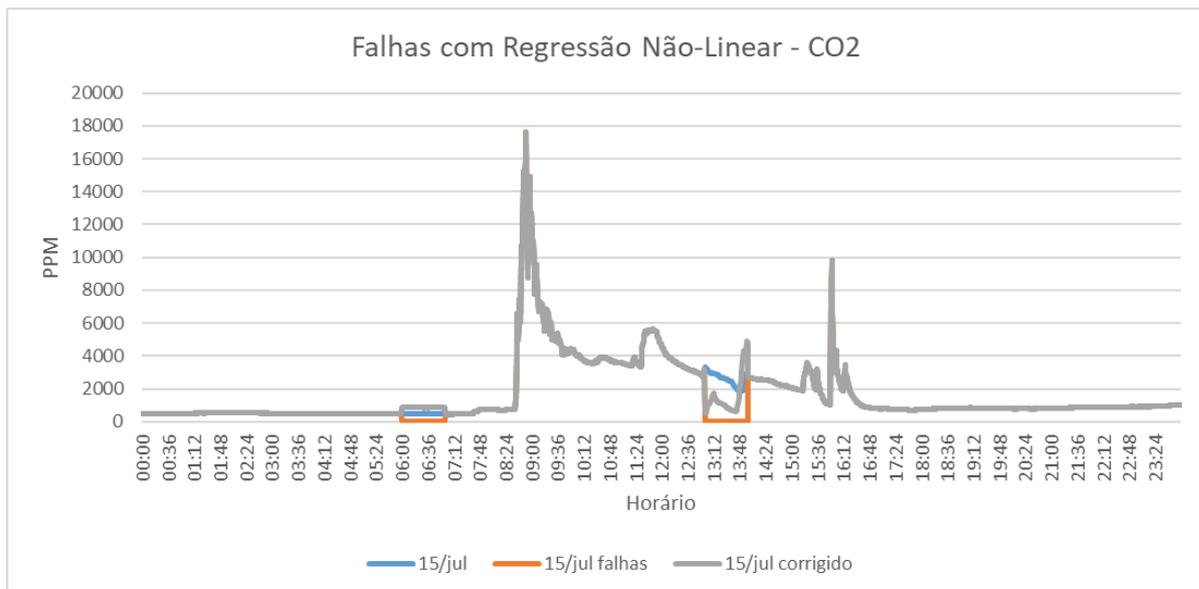


Figura 61 – Aplicação do Modelo de Regressão Não-Linear Polinomial de 2ª Ordem para corrigir falhas no sensor de CO2.

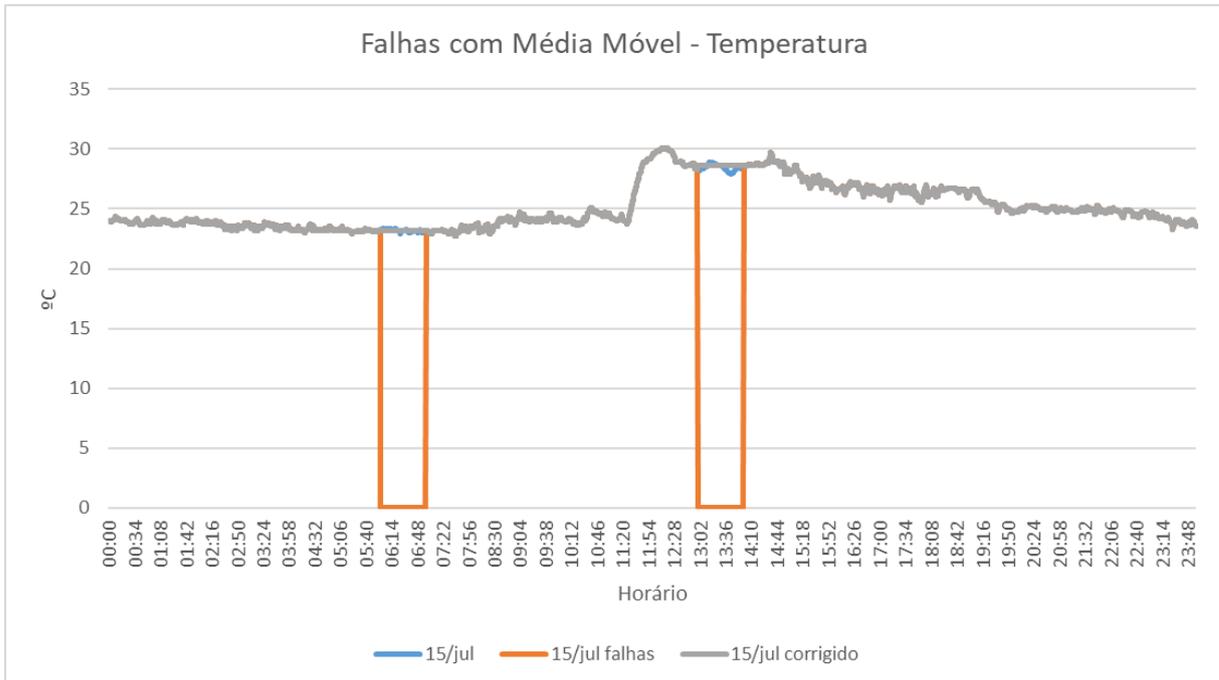


Figura 62 – Aplicação do Modelo de Média Móvel para corrigir falhas no sensor de temperatura.

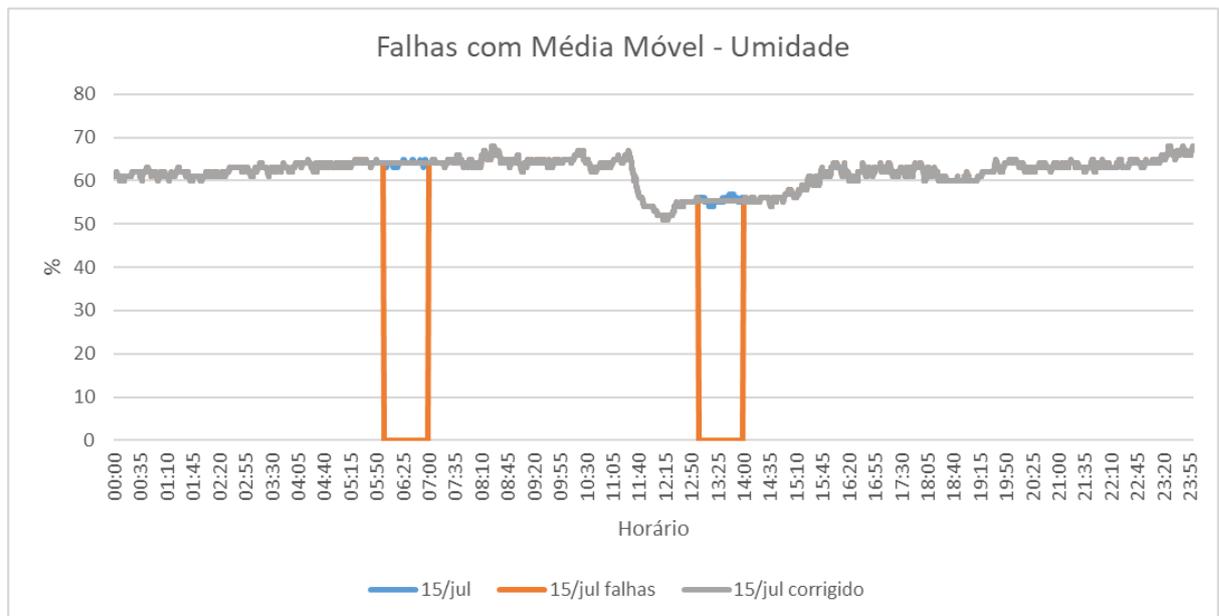


Figura 63 – Aplicação do Modelo de Média Móvel para corrigir falhas no sensor de umidade.

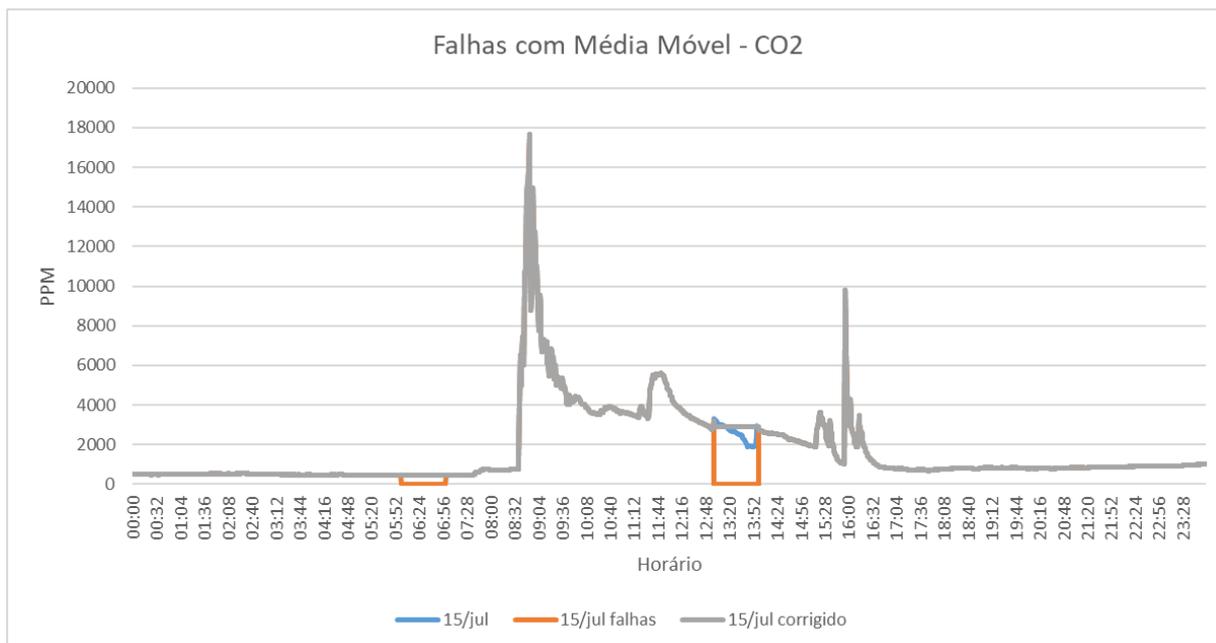


Figura 64 – Aplicação do Modelo de Média Móvel para corrigir falhas no sensor de CO2.

quenos de erro para estas variáveis, então é sensato utilizar um modelo que atenda aos requisitos independente se utiliza ou não da replicação de sensores.

5.10 Considerações Sobre a Concentração de CO2

A concentração de CO2 pode ser um dos indicadores de riscos de transmissão de COVID-19 para ambiente internos conforme observado por (PENG; JIMENEZ, 2021). A concentração de CO2 presente na Figura 29 por exemplo, contata-se a sensibilidade do sensor MQ-135 quanto a presença de pessoas no ambiente. Tendo somente uma pessoa no ambiente, o sensor já foi capaz de identificar variações, entretanto ao inserirmos mais pessoas no ambiente percebe-se os picos de subida no gráfico.

Caso o sensor esteja inserido em uma sala de aula ou em ambientes administrativos, picos acentuados de subida podem caracterizar o início de algum evento como uma aula e a concentração de um grande número de pessoas pode representar um grau maior no risco de infecção por doenças respiratórias tais como a COVID-19.

Pelos modelos analisados, devido a flutuação de valores e a dificuldade em criar previsões para esta grandeza, podemos constatar que a utilização de modelos que façam uso da replicação de sensor, como é o caso de Peniak, sejam a melhor solução para tratamento de falhas.

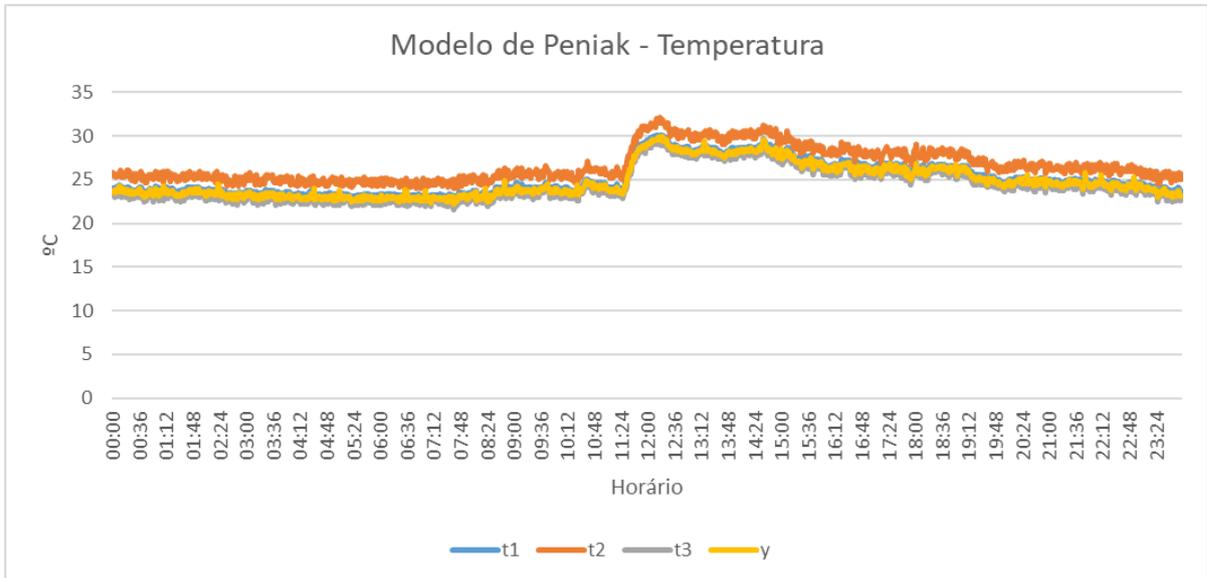


Figura 65 – Aplicação do Modelo de Peniak no sensor de temperatura.

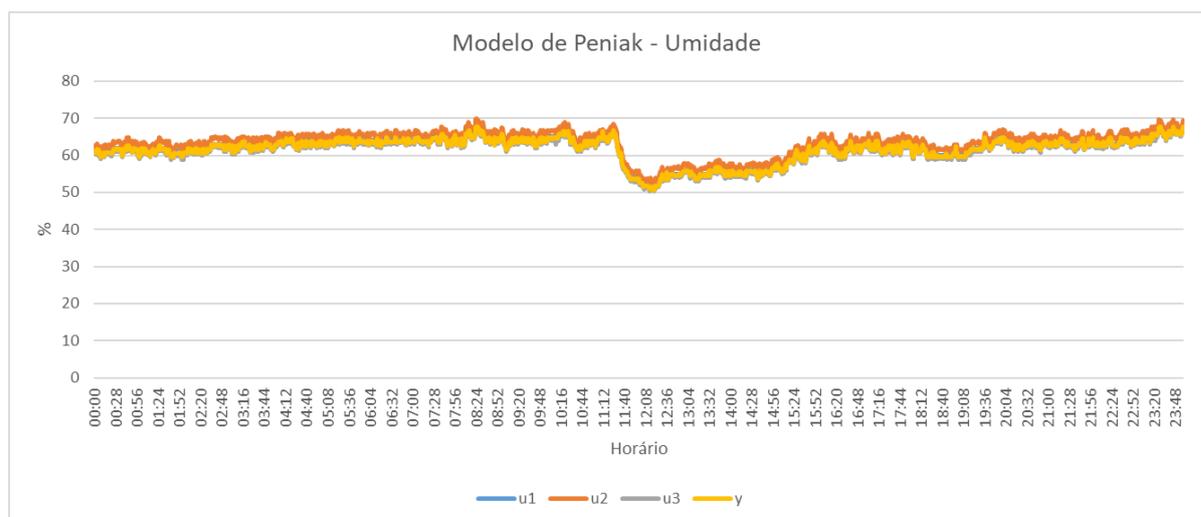


Figura 66 – Aplicação do Modelo de Peniak no sensor de umidade.

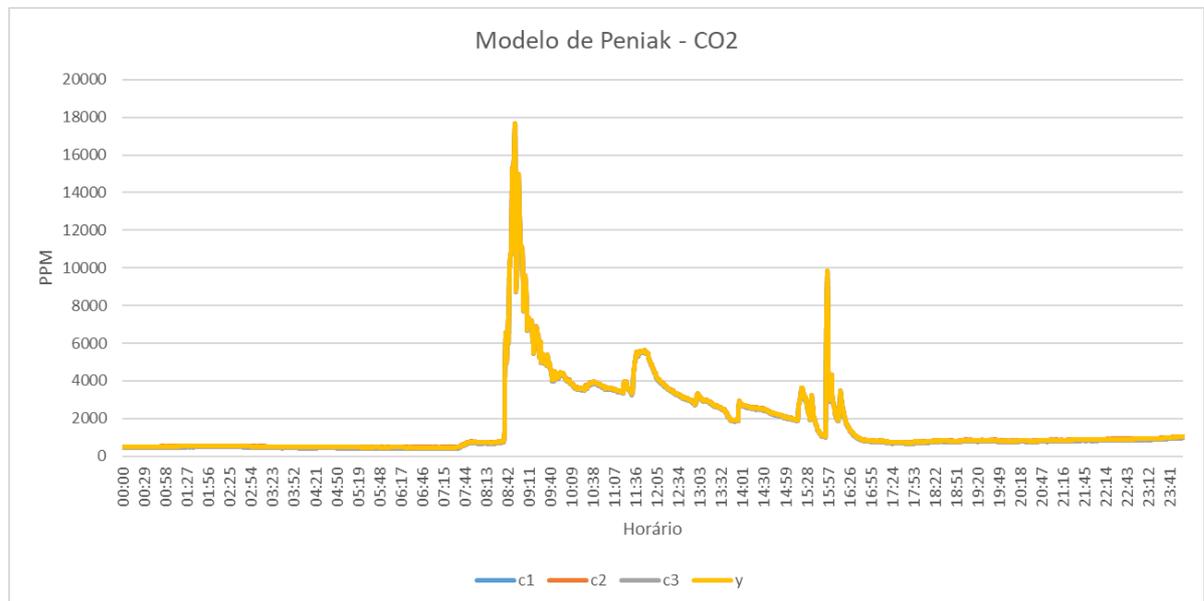


Figura 67 – Aplicação do Modelo de Peniak no sensor de CO2.

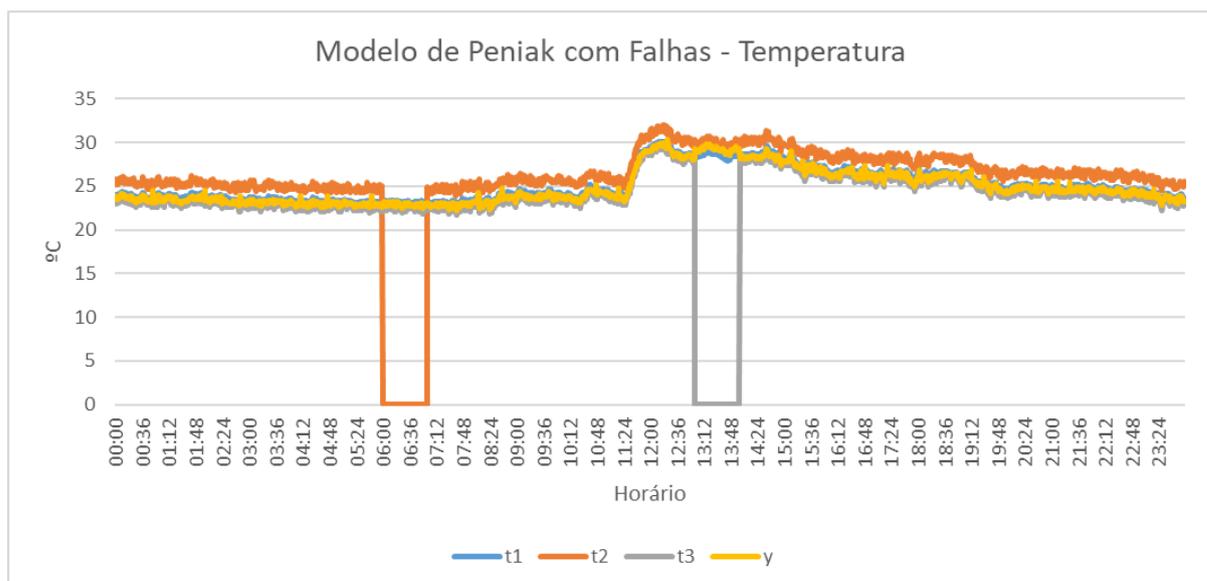


Figura 68 – Aplicação de Falhas no Modelo de Peniak no sensor de temperatura.

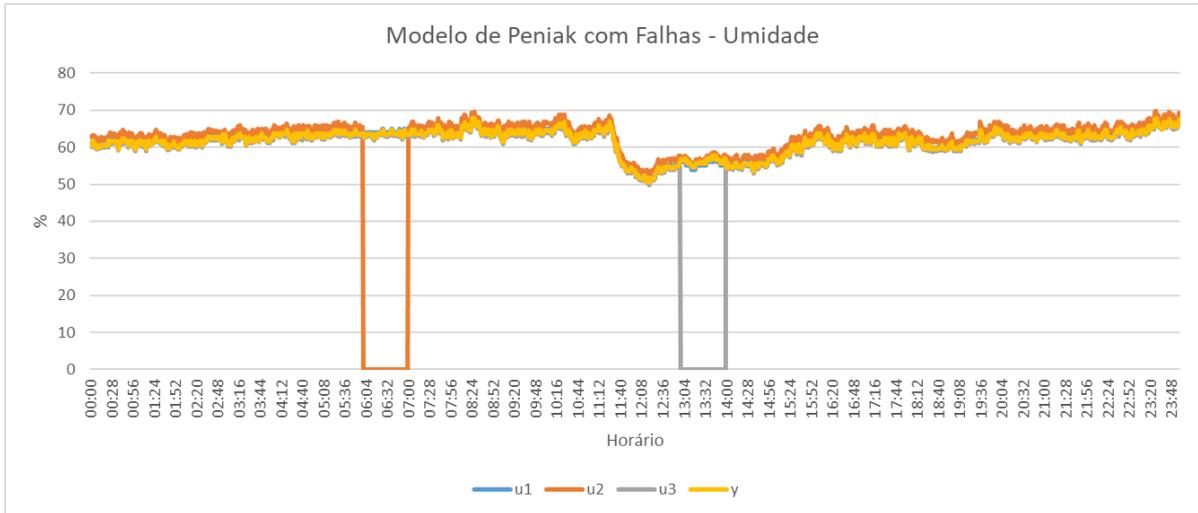


Figura 69 – Aplicação de Falhas no Modelo de Peniak no sensor de umidade.

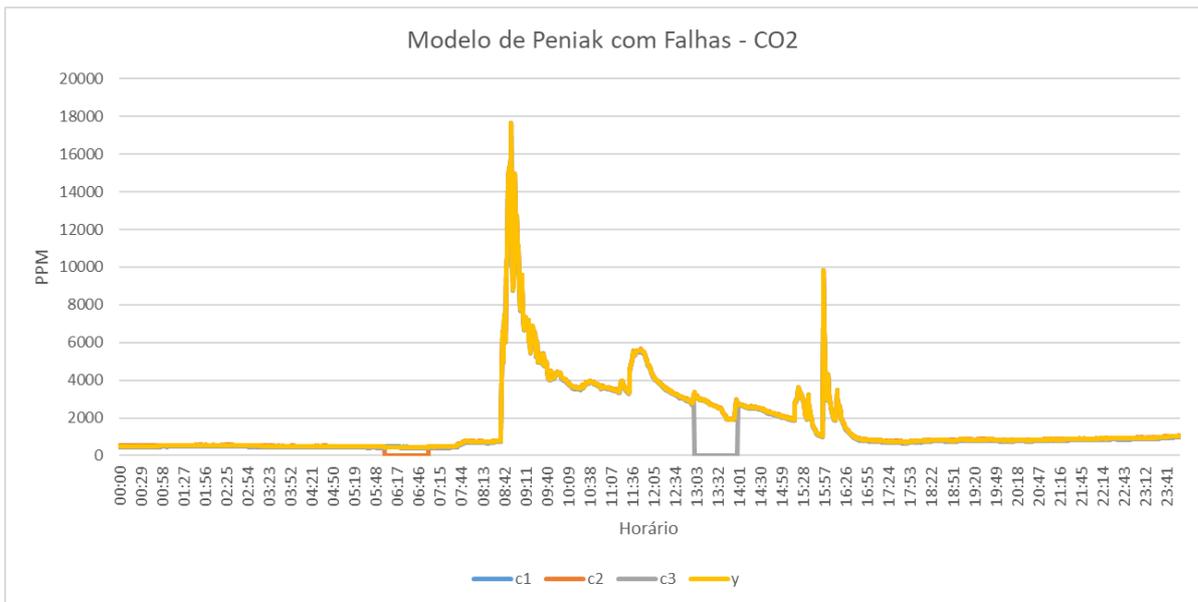


Figura 70 – Aplicação de Falhas no Modelo de Peniak no sensor de CO2.

6 CONCLUSÕES

Este trabalho demonstrou diferentes implementações de modelos para sensores virtuais que podem ser utilizados no contexto de tolerância a falhas e suas aplicações em um ambiente real. Os modelos apresentados foram avaliados quanto aos seus erros e, foi utilizado um conjunto de sensores num ambiente real para monitorar a qualidade do ar, a fim de validar a eficácia dos modelos apresentados.

Os resultados obtidos demonstraram que sistemas que empregam sensores virtuais podem ser tolerantes a falhas, mostrando-se como uma ferramenta útil para garantir a confiabilidade de sistemas críticos, reduzindo a probabilidade de falhas e aumentando a segurança e a disponibilidade do sistema muito importante em alguns tipos de aplicações. No entanto, é importante lembrar que os modelos utilizados apresentam limitações, que devem ser avaliados e considerados na implementação de sistemas de tolerância a falhas.

A utilização de sensores para monitorar a qualidade do ar foi uma forma de validar os modelos apresentados, mostrando que eles foram capazes de detectar e corrigir falhas no sistema de monitoramento quando as falhas apareceram. Essa abordagem demonstra que a combinação de diferentes tecnologias pode ser utilizada para melhorar a confiabilidade e a eficácia dos sistemas.

Como estudo de caso, sabemos que a COVID-19 impactou o mundo todo com uma doença que poucos conheciam. Com o decorrer da Pandemia, novos estudos foram surgindo e ideias para controlar a disseminação do vírus vem sendo implementadas e testadas. A necessidade de desenvolver soluções de baixo custo, eficientes e em um curto intervalo de tempo foi o foco dos cientistas nesta época.

De modo a contribuir para o controle da disseminação da Pandemia de COVID-19 em conjunto com um sistema capaz de se recuperar de falhas foi produzido este projeto. Através de um conjunto de sensores logrou-se êxito em indicar momentos que uma determinada sala pode estar com uma lotação acima da segura. Nos experimentos propostos foram utilizados um conjunto limitado de sensores, porém nada impede de escalar o mesmo conjunto de sensores para um número maior, haja visto que a mesma metodologia aplicada para este sistema pode ser aplicada para qualquer

ambiente.

A recuperação de falhas faz uso de sensores virtuais previamente modelados. Cabendo ao desenvolvedor, optar pelo modelo que melhor se encaixa em seu cenário. No decorrer do trabalho foi apresentado um método de identificação de falhas utilizando uma Janela Flutuante, este método identifica os valores anormais e com uma base de dados anteriores e utilizando um modelo adequado, é capaz de encontrar o melhor valor para aquele determinado momento, sendo esta uma implementação de modelo simples e ultra-leve, que pode ser implementada na grande maioria dos sistemas e microcontroladores. Apesar de ser aplicado em um único nó sensor, o mesmo método pode ser aplicado para um conjunto de sistemas de sensores em diferentes ambientes, desde acadêmicos até residenciais e corporativos. Além de os modelos serem simples e ultra-leves, também são facilmente adaptáveis e confiáveis para que possam ser escaláveis a vários sensores que operem em conjunto ou de forma alheia.

De acordo com a avaliação do erro após a execução dos tratamentos conclui-se que os sensores virtuais propostos são viáveis para produzir sistemas tolerantes a falhas. O sistema proposto pode ser usado para avaliar a qualidade do ar em ambientes fechados tais como um escritório ou sala administrativa indicando potenciais riscos a doenças respiratórias como a gripe por COVID19 devido a má circulação de ar.

Como perspectiva de trabalhos futuros, propõe-se um estudo de comparação de diferentes tipos de sensores para avaliar sua precisão na detecção de determinado fenômeno. Além disso, pode-se realizar comparações entre os métodos mais simples apresentados neste trabalho e métodos mais robustos que fazem uso de modelos com IA. Embora os métodos mais simples tenham apresentado resultados satisfatórios em aplicações anteriores, é importante avaliar seu desempenho em situações mais complexas. Outra possibilidade é explorar outras aplicações para os métodos propostos, como na detecção de padrões ou na previsão de fenômenos naturais, por exemplo. Para realizar esses estudos, será necessário coletar dados adicionais de diferentes fontes e garantir que os resultados sejam representativos. Além disso, é importante comparar os resultados obtidos com outros trabalhos na área, a fim de avaliar a contribuição desses estudos para o avanço do conhecimento sobre o tema.

REFERÊNCIAS

- ABEEL, T.; PEER, Y. V. de; SAEYS, Y. Java-ML: A Machine Learning Library. **Journal of Machine Learning Research**, [S.l.], v.10, n.34, p.931–934, 2009.
- AGARWAL, P.; GVV, R. K.; AGARWAL, P. IoT based Framework for Smart Campus: COVID-19 Readiness. In: FOURTH WORLD CONFERENCE ON SMART TRENDS IN SYSTEMS, SECURITY AND SUSTAINABILITY (WORLDS4), 2020., 2020. **Anais...** [S.l.: s.n.], 2020. p.539–542.
- AL NUAIMI, E.; AL NEYADI, H.; MOHAMED, N.; AL-JAROODI, J. Applications of big data to smart cities. **Journal of Internet Services and Applications**, [S.l.], v.6, n.1, p.25, 2015.
- AQUINO, E. M. et al. Medidas de distanciamento social no controle da pandemia de COVID-19: potenciais impactos e desafios no Brasil. **Ciência & Saúde Coletiva**, [S.l.], v.25, p.2423–2446, 2020.
- ARDEKANI, M. S. et al. Rivulet: a fault-tolerant platform for smart-home applications. In: ACM/IFIP/USENIX MIDDLEWARE CONFERENCE, 18., 2017. **Proceedings...** [S.l.: s.n.], 2017. p.41–54.
- ATZORI, L.; IERA, A.; MORABITO, G. The Internet of Things: A survey. **Computer Networks**, [S.l.], v.54, n.15, p.2787–2805, 2010.
- AVIZIENIS, A. Infraestructure-based design of fault-tolerant systems. In: IFIP INTERNATIONAL WORKSHOP ON DEPENDABLE COMPUTING AND ITS APPLICATIONS. DCIA, 1998. **Proceedings...** [S.l.: s.n.], 1998. v.98, p.12–14.
- BIBRI, S. E. On the sustainability of smart and smarter cities in the era of big data: an interdisciplinary and transdisciplinary literature review. **Journal of Big Data**, [S.l.], v.6, n.1, p.25, 2019.
- BURHAN, M.; REHMAN, R. A.; KHAN, B.; KIM, B.-S. IoT elements, layered architectures and security issues: A comprehensive survey. **Sensors**, [S.l.], v.18, n.9, p.2796, 2018.

CANEDO, J.; SKJELLUM, A. Using machine learning to secure IoT systems. In: PST), 2016., 2016. **Anais...** [S.l.: s.n.], 2016. p.219–222.

CARISSIMI, A. Desmistificando a Computação em Nuvem. **ROSE, Cesar de**, [S.l.], p.3–24, 2015.

CASADO-VARA, R.; VALE, Z.; PRIETO, J.; CORCHADO, J. M. Fault-tolerant temperature control algorithm for IoT networks in smart buildings. **Energies**, [S.l.], v.11, n.12, p.3430, 2018.

CEIEC Smart Campus Solution. **Smart Campus.** [Online; accessed November 30, 2020]. Disponível em: <<https://www.ceiec.com/solution/publicSecurity/AreaManagement/>>.

CHAKRABORTY, T. et al. Fall-curve: A novel primitive for IoT Fault Detection and Isolation. In: ACM CONFERENCE ON EMBEDDED NETWORKED SENSOR SYSTEMS, 16., 2018. **Proceedings...** [S.l.: s.n.], 2018. p.95–107.

CHAKRABORTY, T. et al. Sensor identification and fault detection in IoT systems. In: ACM CONFERENCE ON EMBEDDED NETWORKED SENSOR SYSTEMS, 16., 2018. **Proceedings...** [S.l.: s.n.], 2018. p.375–376.

CHANDAN, A. R.; KHAIRNAR, V. D. Bluetooth Low Energy (BLE) Crackdown Using IoT. In: INTERNATIONAL CONFERENCE ON INVENTIVE RESEARCH IN COMPUTING APPLICATIONS (ICIRCA), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p.1436–1441.

CHETTRI, L.; BERA, R. A Comprehensive Survey on Internet of Things (IoT) Toward 5G Wireless Systems. **IEEE Internet of Things Journal**, [S.l.], v.7, n.1, p.16–32, 2020.

CHIU, Y.-C.; LIU, L.-Y.; WANG, T. Automatic segmentation and summarization for videos taken with smart glasses. **Multimedia Tools and Applications**, [S.l.], v.77, n.10, p.12679–12699, 2018.

CHUDZIKIEWICZ, J.; FURTAK, J.; ZIELINSKI, Z. Fault-tolerant techniques for the Internet of Military Things. In: IEEE 2ND WORLD FORUM ON INTERNET OF THINGS (WF-IOT), 2015., 2015. **Anais...** [S.l.: s.n.], 2015. p.496–501.

CRISTALDI, L. et al. Virtual Sensors: a Tool to Improve Reliability. In: IEEE INTERNATIONAL WORKSHOP ON METROLOGY FOR INDUSTRY 4.0 IOT, 2020. **Anais...** [S.l.: s.n.], 2020. p.142–145.

DANTAS, M. **Computação Distribuída de Alto Desempenho, Redes, Clusters e Grids Computacionais**. [S.l.]: Axcel Books do Brasil Editora, 2005.

ENEYEW, D. D.; CAPRETZ, M. A. M.; BITSUAMLAK, G. T. Toward Smart-Building Digital Twins: BIM and IoT Data Integration. **IEEE Access**, [S.l.], v.10, p.130487–130506, 2022.

EYKELBOSH, A. **Can CO2 sensors be used to assess COVID-19 transmission risk?** [S.l.]: National Collaborating Centre for Environmental Health, 2021. Disponível em: <<https://ncceh.ca/content/blog/can-co2-sensors-be-used-assess-covid-19-transmission-risk>>.

FARIAS, R. S.; SOUZA, R. M. de; MCGREGOR, J. D.; ALMEIDA, E. S. de. Designing smart city mobile applications. **Empirical Software Engineering**, [S.l.], v.24, n.6, p.3255–3289, 2019.

FERREIRA, F. H. C.; ARAÚJO, R. M. Campus Inteligentes: Conceitos, aplicações, tecnologias e desafios. **RelaTe-DIA**, [S.l.], v.11, n.1, 2018.

FIGUEIREDO FILHO, D. B.; SILVA JÚNIOR, J. A. Desvendando os Mistérios do Coeficiente de Correlação de Pearson (r). **Revista Política Hoje**, [S.l.], v.18, n.1, p.115–146, 2009.

FRANCO, N. M. B. **Cálculo Numérico**. [S.l.]: Pearson, 2006.

GOTTSCHO, M. et al. Low-cost memory fault tolerance for iot devices. **ACM Transactions on Embedded Computing Systems (TECS)**, [S.l.], v.16, n.5s, p.1–25, 2017.

GROSS, J.; GROSS, J. **Linear regression**. [S.l.]: Springer Science & Business Media, 2003. v.175.

GUTIERREZ, M. D.; TENENTES, V.; KAZMIERSKI, T. J.; ROSSI, D. Low cost error monitoring for improved maintainability of iot applications. In: IEEE INTERNATIONAL SYMPOSIUM ON DEFECT AND FAULT TOLERANCE IN VLSI AND NANOTECHNOLOGY SYSTEMS (DFT), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p.1–6.

HANSUN, S. A new approach of moving average method in time series analysis. In: CONMEDIA), 2013., 2013. **Anais...** [S.l.: s.n.], 2013. p.1–4.

HARRISON, C.; DONNELLY, I. A. A theory of smart cities. In: ANNUAL MEETING OF THE ISSS-2011, HULL, UK, 55., 2011. **Proceedings...** [S.l.: s.n.], 2011.

HOSSAIN, I.; DAS, D.; RASHED, M. G. Internet of Things Based Model for Smart Campus: Challenges and Limitations. In: INTERNATIONAL CONFERENCE ON COMPUTER, COMMUNICATION, CHEMICAL, MATERIALS AND ELECTRONIC ENGINEERING (IC4ME2), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p.1–4.

IQBAL, M. M. et al. Multimedia based IoT-centric smart framework for eLearning paradigm. **Multimedia Tools and Applications**, [S.l.], v.78, n.3, p.3087–3106, 2019.

IQBAL, M. M.; SALEEM, Y.; NASEER, K.; KIM, M. Multimedia based student-teacher smart interaction framework using multi-agents in eLearning. **Multimedia Tools and Applications**, [S.l.], v.77, n.4, p.5003–5026, 2018.

JAVED, A.; ROBERT, J.; HELJANKO, K.; FRÄMLING, K. IoTEF: A federated edge-cloud architecture for fault-tolerant IoT applications. **Journal of Grid Computing**, [S.l.], v.18, n.1, p.57–80, 2020.

KAIWARTYA, O. et al. Virtualization in wireless sensor networks: Fault tolerant embedding for internet of things. **IEEE Internet of Things Journal**, [S.l.], v.5, n.2, p.571–580, 2017.

KHRIJI, S. et al. Design and implementation of a cloud-based event-driven architecture for real-time data processing in wireless sensor networks. **The Journal of Supercomputing**, [S.l.], v.78, n.3, p.3374–3401, 2022.

KODESWARAN, P. A.; KOKKU, R.; SEN, S.; SRIVATSA, M. Idea: A system for efficient failure management in smart iot environments. In: ANNUAL INTERNATIONAL CONFERENCE ON MOBILE SYSTEMS, APPLICATIONS, AND SERVICES, 14., 2016. **Proceedings...** [S.l.: s.n.], 2016. p.43–56.

KUMARI, P.; KAUR, P. A survey of fault tolerance in cloud computing. **Journal of King Saud University-Computer and Information Sciences**, [S.l.], v.33, n.10, p.1159–1176, 2021.

LAZAROIU, G. C. et al. Smart campus-an energy integrated approach. In: INTERNATIONAL CONFERENCE ON RENEWABLE ENERGY RESEARCH AND APPLICATIONS (ICRERA), 2015., 2015. **Anais...** [S.l.: s.n.], 2015. p.1497–1501.

LIANG, W. Analysis of the Application of Artificial Intelligence Technology in the Construction of Smart Campus. In: INTERNATIONAL WIRELESS COMMUNICATIONS AND MOBILE COMPUTING (IWCMC), 2020., 2020. **Anais...** [S.l.: s.n.], 2020. p.882–885.

LIHONG, W. Research on the Construction of Smart Campus Social Platform Based on Hadoop. In: INTERNATIONAL CONFERENCE ON COMPUTER ENGINEERING AND APPLICATION (ICCEA), 2020., 2020. **Anais...** [S.l.: s.n.], 2020. p.214–217.

LIU, H.; LIU, Y.; SUN, F. Video key-frame extraction for smart phones. **Multimedia Tools and Applications**, [S.l.], v.75, n.4, p.2031–2049, 2016.

LIU, L.; KUO, S. M.; ZHOU, M. Virtual sensing techniques and their applications. In: INTERNATIONAL CONFERENCE ON NETWORKING, SENSING AND CONTROL, 2009., 2009. **Anais...** [S.l.: s.n.], 2009. p.31–36.

LIU, Y. et al. Towards a smart campus: Innovative applications with WiCloud platform based on mobile edge computing. In: INTERNATIONAL CONFERENCE ON COMPUTER SCIENCE AND EDUCATION (ICCSE), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p.133–138.

LOCH, W.; IANKOWSKI, R. S. **Reconhecimento de Caracteres em Placas de Automóveis Utilizando Computação em Nuvem**. Monografia (Bacharel em Engenharia de Computação), UFPEL (Universidade Federal de Pelotas), Pelotas, Brazil.

MADAKAM, S. et al. Internet of Things (IoT): A literature review. **Journal of Computer and Communications**, [S.l.], v.3, n.05, p.164, 2015.

MAGRANI, E. **A internet das coisas**. [S.l.]: Editora FGV, 2018.

Malatji, E. M.; Kwesi Ntsaluba, S. B. Smart Energy Generation for a Smart Campus. In: INTERNATIONAL CONFERENCE ON INTELLIGENT AND INNOVATIVE COMPUTING APPLICATIONS (ICONIC), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p.1–5.

Malatji, E. M.; Kwesi Ntsaluba, S. B. Smart Energy Generation for a Smart Campus. In: INTERNATIONAL CONFERENCE ON INTELLIGENT AND INNOVATIVE COMPUTING APPLICATIONS (ICONIC), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p.1–5.

MARTIN, D.; KÜHL, N.; SATZGER, G. Virtual Sensors. **Business Information Systems Engineering**, [S.l.], v.63, n.3, p.315–323, 2021.

MATOS, D.; HAMASAKI, E.; RIBEIRO, L.; COSTA, I. **Ensino de precisão de leitura e escrita em crianças com desenvolvimento típico**. [S.l.: s.n.], 2020. p.123.

MEDJEK, F.; TANDJAOU, D.; DJEDJIG, N.; ROMDHANI, I. Fault-tolerant ai-driven intrusion detection system for the internet of things. **International Journal of Critical Infrastructure Protection**, [S.l.], v.34, p.100436, 2021.

MIHAI, S. et al. Digital Twins: A Survey on Enabling Technologies, Challenges, Trends and Future Prospects. **IEEE Communications Surveys Tutorials**, [S.l.], v.24, n.4, p.2255–2291, 2022.

MORAROS, J. et al. Flipping for success: evaluating the effectiveness of a novel teaching approach in a graduate level setting. **BMC medical education**, [S.l.], v.15, n.1, p.27, 2015.

MUDASSAR, M.; ZHAI, Y.; LEJIAN, L. Adaptive Fault-Tolerant Strategy for Latency-Aware IoT Application Executing in Edge Computing Environment. **IEEE Internet of Things Journal**, [S.l.], v.9, n.15, p.13250–13262, 2022.

MUHAMAD, W.; KURNIAWAN, N. B.; YAZID, S. et al. Smart campus features, technologies, and applications: A systematic literature review. In: INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY SYSTEMS AND INNOVATION (ICITSI), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p.384–391.

MUHAMMAD, A. N. et al. Deep learning application in smart cities: recent development, taxonomy, challenges and research prospects. **Neural Computing and Applications**, [S.l.], p.1–37, 2020.

NANO, A. **Arduino Nano**. [S.l.]: Obtenido de Arduino: <https://store.arduino.cc/usa/arduino-nano>, 2018.

NIE, X. Research on smart campus based on cloud computing and internet of things. In: APPLIED MECHANICS AND MATERIALS, 2013. **Anais...** [S.l.: s.n.], 2013. v.380, p.1951–1954.

OLIVEIRA, C. H. R. de; COSTA, A. P. F.; THOMAZ, V. F.; SILVA, I. A. Low-cost deployment proposal to urban mobility in smart cities. **The Journal of Supercomputing**, [S.l.], v.75, n.11, p.7265–7289, 2019.

OLIVEIRA, P.; MATOS, P. BLEGen - A Code Generator for Bluetooth Low Energy Services. **Lecture Notes on Software Engineering**, [S.l.], v.4, p.7–11, 01 2016.

ÖZCAN, U.; ARSLAN, A.; İLKYZ, M.; KARAARSLAN, E. An augmented reality application for smart campus urbanization: MSKU campus prototype. In: INTERNATIONAL ISTANBUL SMART GRID AND CITIES CONGRESS AND FAIR (ICSG), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p.100–104.

PENG, Z.; JIMENEZ, J. L. Exhaled CO₂ as a COVID-19 infection risk proxy for different indoor environments and activities. **Environmental Science & Technology Letters**, [S.l.], v.8, n.5, p.392–397, 2021.

PENIAK, P.; BUBENÍKOVÁ, E.; KANÁLIKOVÁ, A. The Redundant Virtual Sensors via Edge Computing. In: INTERNATIONAL CONFERENCE ON APPLIED ELECTRONICS (AE), 2021., 2021. **Anais...** [S.l.: s.n.], 2021. p.1–5.

PRADHAN, D. K. **Fault-tolerant computer system design**. [S.l.]: Prentice-Hall, Inc., 1996.

QAIM, W. B.; OZKASAP, O. DRAW: Data replication for enhanced data availability in IoT-based sensor systems. In: IEEE 16TH INTL CONF ON DEPENDABLE, AUTONOMIC AND SECURE COMPUTING, 16TH INTL CONF ON PERVASIVE INTELLIGENCE AND COMPUTING, 4TH INTL CONF ON BIG DATA INTELLIGENCE AND COMPUTING AND CYBER SCIENCE AND TECHNOLOGY CONGRESS (DASC/PICOM/DATACOM/CYBERSCITECH), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p.770–775.

RASHMI, M.; ASHWIN, T.; GUDDI, R. M. R. Surveillance video analysis for student action recognition and localization inside computer laboratories of a smart campus. **Multimedia Tools and Applications**, [S.l.], p.1–23, 2020.

RAVEENDRANATHAN, N. et al. From Modeling to Implementation of Virtual Sensors in Body Sensor Networks. **IEEE Sensors Journal**, [S.l.], v.12, n.3, p.583–593, 2012.

RAY, B. K.; SAHA, A.; KHATUA, S.; ROY, S. Proactive Fault-Tolerance Technique to Enhance Reliability of Cloud Service in Cloud Federation Environment. **IEEE Transactions on Cloud Computing**, [S.l.], v.10, n.2, p.957–971, 2022.

REHMAN, A. U.; AGUIAR, R. L.; BARRACA, J. P. Fault-Tolerance in the Scope of Cloud Computing. **IEEE Access**, [S.l.], v.10, p.63422–63441, 2022.

RUSLI, A.; HALIM, D. K. Towards an Integrated Hybrid Mobile Application for Smart Campus Using Location-Based Smart Notification. In: INTERNATIONAL CONFERENCE ON ENGINEERING, SCIENCE, AND INDUSTRIAL APPLICATIONS (ICESI), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p.1–6.

SAEED, W. et al. A fault tolerant data management scheme for healthcare Internet of Things in fog computing. **KSII Transactions on Internet and Information Systems (TIIS)**, [S.l.], v.15, n.1, p.35–57, 2021.

SCHUELER, P. **O que É uma pandemia - bio-manguinhos/fiocruz || Inovação ...** Disponível em: <<https://bio.fiocruz.br/index.php/br/noticias/1763-o-que-e-uma-pandemia>>.

SIECK, N.; CALPIN, C.; ALMALAG, M. Machine Vision Smart Parking Using Internet of Things (IoTs) In A Smart University. In: IEEE INTERNATIONAL CONFERENCE ON PERVASIVE COMPUTING AND COMMUNICATIONS WORKSHOPS (PERCOM WORKSHOPS), 2020., 2020. **Anais...** [S.l.: s.n.], 2020. p.1–6.

SILVA DIAS, P. R. da; BÔAS, G. K. E. V.; LUCENA, S. A. G. F. As principais mudanças nas rotinas do setor de pessoal decorrentes da pandemia Covid-19 em um escritório de contabilidade na Cidade de Barreiras-BA. , [S.l.], 2020.

SIVARAMAN, V. et al. Smart IoT devices in the home: Security and privacy implications. **IEEE Technology and Society Magazine**, [S.l.], v.37, n.2, p.71–79, 2018.

SOUZA, F. R. A.; SILVA, C. M. da; GUIMARÃES, C. Segurança em Rede Wireless. **e-xacta**, [S.l.], v.2, n.2, 2009.

SOUZA, T. L.; CARVALHO, T. de. DISPOSITIVOS INTELIGENTES: o uso do RFID em bibliotecas nacionais e internacionais. **ConCI: Convergências em Ciência da Informação**, [S.l.], v.1, n.2, p.115–122, 2018.

STARIOLO, D.; SOBREIRO, R. F. et al. Covid-19: propagação espacial e tempo de suspensão em ambientes fechados. , [S.l.], 2020.

STEVENTON, A.; WRIGHT, S. **Intelligent spaces**: The application of pervasive ICT. [S.l.]: Springer Science & Business Media, 2010.

TANENBAUM A.S; STEEN, M. **Sistemas distribuídos**: princípios e paradigmas. [S.l.]: Pearson Prentice Hall, 2007.

TAO, M.; WEI, W.; YUAN, H.; HUANG, S. Version-vector based video data online cloud backup in smart campus. **Multimedia Tools and Applications**, [S.l.], v.78, n.3, p.3435–3456, 2019.

TAI, M.-F.; KIONG, Y. C.; SINN, A. Smart service relying on Internet of Things technology in parking systems. **The Journal of Supercomputing**, [S.l.], v.74, n.9, p.4315–4338, 2018.

VAN MERODE, D.; TABUNSHCHYK, G.; PATRAKHALKO, K.; YURIY, G. Flexible technologies for smart campus. In: INTERNATIONAL CONFERENCE ON REMOTE ENGINEERING AND VIRTUAL INSTRUMENTATION (REV), 2016., 2016. **Anais...** [S.l.: s.n.], 2016. p.64–68.

VERMUND, S. H.; PITZER, V. E. Asymptomatic transmission and the infection fatality risk for COVID-19: Implications for school reopening. **Clinical Infectious Diseases**, [S.l.], 2020.

VERSTAEVEL, N. et al. An ontology based context-aware architecture for smart campus applications. In: IEEE INTL CONF ON PARALLEL & DISTRIBUTED PROCESSING WITH APPLICATIONS, UBIQUITOUS COMPUTING & COMMUNICATIONS, BIG DATA & CLOUD COMPUTING, SOCIAL COMPUTING & NETWORKING, SUSTAINABLE COMPUTING & COMMUNICATIONS (ISPA/IUCC/BD CLOUD/SOCIALCOM/SUSTAINCOM), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p.1056–1063.

WANG, K. et al. Adaptive and fault-tolerant data processing in healthcare IoT based on fog computing. **IEEE transactions on network science and engineering**, [S.l.], v.7, n.1, p.263–273, 2018.

WANG, L. et al. Cloud computing: a perspective study. **New Generation Computing**, [S.l.], v.28, n.2, p.137–146, 2010.

WANG, Y.; ROTONDO, D.; PUIG, V.; CEMBRANO, G. Fault-tolerant control based on virtual actuator and sensor for discrete-time descriptor systems. **IEEE Transactions on Circuits and Systems I: Regular Papers**, [S.l.], v.67, n.12, p.5316–5325, 2020.

WEBER, T. S. Tolerância a falhas: conceitos e exemplos. **Apostila do Programa de Pós-Graduação–Instituto de Informática-UFRGS. Porto Alegre**, [S.l.], p.24, 2003.

WOO, M. W.; LEE, J.; PARK, K. A reliable IoT system for personal healthcare devices. **Future Generation Computer Systems**, [S.l.], v.78, p.626–640, 2018.

XU, T.; POTKONJAK, M. Energy-efficient fault tolerance approach for internet of things applications. In: INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, 35., 2016. **Proceedings...** [S.l.: s.n.], 2016. p.1–8.

XU, X.; WANG, Y.; YU, S. Teaching performance evaluation in smart campus. **IEEE Access**, [S.l.], v.6, p.77754–77766, 2018.

YANG, C.-T. et al. On construction of an energy monitoring service using big data technology for the smart campus. **Cluster Computing**, [S.l.], v.23, n.1, p.265–288, 2020.

ZANDBERG, K. et al. Femto-Containers: Lightweight virtualization and fault isolation for small software functions on low-power IoT microcontrollers. In: ACM/IFIP INTERNATIONAL MIDDLEWARE CONFERENCE, 23., 2022. **Proceedings...** [S.l.: s.n.], 2022. p.161–173.

ZHAMANOV, A.; SAKHIYEVA, Z.; SULIYEV, R.; KALDYKULOVA, Z. IoT smart campus review and implementation of IoT applications into education process of university. In: INTERNATIONAL CONFERENCE ON ELECTRONICS, COMPUTER AND COMPUTATION (ICECCO), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p.1–4.

ZHOU, S. et al. Supporting service adaptation in fault tolerant internet of things. In: IEEE 8TH INTERNATIONAL CONFERENCE ON SERVICE-ORIENTED COMPUTING AND APPLICATIONS (SOCA), 2015., 2015. **Anais...** [S.l.: s.n.], 2015. p.65–72.