UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Instituto de Ciência e Tecnologia - Campus de Sorocaba

Levi Rodrigues da Silva

Análise Embarcada de Dados de Temperatura e Umidade da Área de Preservação Permanente do *Ibiti Royal Park*

Dissertação apresentada à Universidade Estadual Paulista (UNESP), Instituto de Ciência e Tecnologia, Sorocaba, para obtenção do título de Mestre em Engenharia Elétrica.

Sorocaba - SP
2026

LEVI RODRIGUES DA SILVA

Análise Embarcada de Dados de Temperatura e Umidade da Área de Preservação Permanente do *Ibiti Royal Park*

Dissertação apresentada à Universidade Estadual Paulista (UNESP), Instituto de Ciência e Tecnologia, Sorocaba, para obtenção do título de Mestre em Engenharia Elétrica.

Área de Concentração: Sistemas eletrônicos

Orientador: Prof. Dr. Rodrigo Capobianco Guido

Sorocaba – SP
2026

S586a

Silva, Levi Rodrigues da
Análise embarcada de dados de temperatura e umidade da área de preservação permanente do Ibiti Royal Park / Levi Rodrigues da Silva. -- Sorocaba, 2026
67 p. : il., tabs., fotos

Dissertação (mestrado) - Universidade Estadual Paulista (UNESP), Instituto de Ciência e Tecnologia, Sorocaba
Orientador: Rodrigo Capobianco Guido

1. Internet das coisas. 2. Inteligência artificial-Engenharia de Aplicações. 3. Processamento de sinais. 4. Redes neurais (Computação). 5. Estações meteorológicas. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

UNIVERSIDADE ESTADUAL PAULISTA

Câmpus de Sorocaba

CERTIFICADO DE APROVAÇÃO

TÍTULO DA DISSERTAÇÃO: Análise Embarcada de Dados de Temperatura e Umidade da Área de Preservação Permanente do *Ibiti Royal Park*

AUTOR: LEVI RODRIGUES DA SILVA
ORIENTADOR: RODRIGO CAPOBIANCO GUIDO

Aprovado como parte das exigências para obtenção do Título de Mestre em Engenharia Elétrica, área: Sistemas Eletrônicos pela Comissão Examinadora:

Prof. Dr. RODRIGO CAPOBIANCO GUIDO (Participaçao Virtual)
Departamento de Ciências de Computação e Estatística / UNESP / Câmpus de São José do Rio Preto - Ibilce

Prof. Dr. DIEGO RENAN BRUNO (Participaçao Virtual)
Departamento de Ciências de Computação e Estatística / UNESP / Câmpus de São José do Rio Preto - Ibilce

Prof. Dr. PAULO ROGÉRIO SCALASSARA (Participaçao Virtual)
Departamento Acadêmico de Elétrica / Universidade Tecnológica Federal do Paraná - UTFPR

Sorocaba, 22 de abril de 2026.

gov.br Documento assinado digitalmente
MARCOS DIONIZIO SILVA
Data: 23/04/2026 18:08:47-0300
Verifique em https://validar.iti.gov.br

Marcos Dionizio Silva
Supervisor Técnico de Seção – Substituto
Seção Técnica de Graduação

Agradecimentos

Aos meus pais, *in memoriam*, Antonio Rodrigues da Silva e Maria José Gonçalves, que com certeza estariam orgulhosos com o estudo realizado, pela importância e valorização que davam aos conhecimentos. Aos meus irmãos *in memoriam*, Amauri e Lúcia, que sempre incentivaram os meus estudos, e aos irmãos que representarão muito bem minha família: José Renato, Antonio Carlos, Francisco, Ivone e Elisete.

À minha esposa Eliete, companheira em tudo, aos meus filhos, Rafael e Bruno Filipe, e à razão de minha vida, minhas netas Sophia Bianca e Anna Beatriz.

Ao meu orientador Prof Tit Eng Rodrigo Capobianco Guido, pela paciência e dedicação, representando todos os professores da UNESP.

Ao Instituto de Ciência e Tecnologia de Sorocaba pela grande oportunidade oferecida do saber e apoio irrestrito na busca do conhecimento científico.

Agradeço a todos os citados no corpo desta Dissertação pela contribuição direta.

Estendo minha gratidão aos que não foram nominalmente mencionados, mas que participaram deste resultado com sua torcida, incentivo e contribuições indiretas, agregando valor a este trabalho ou à minha vida. Peço desculpas por eventuais omissões, pois reconheço que cada apoio, por menor que pareça, foi fundamental para esta conquista.

À minha fé, que mesmo no mundo cientista que convivo, ele, Deus, não pode faltar, dando o alicerce necessário para não ficarmos loucos, se do mundo das descobertas e maravilhas que vivemos, fosse olhado pela simplicidade e eficiência da ciência e das causalidades naturais.

“It is far better to grasp the Universe as it really is than to persist in delusion, however satisfying and reassuring.”

Carl Sagan

The Demon-Haunted World: Science as a Candle in the Dark

RESUMO

Pesquisas e coletas de dados por meio de sistemas embarcados de engenharia têm sido primordiais na atualidade, nas mais diversas aplicações. Observam-se, entretanto, lacunas existentes em tais procedimentos, os quais usualmente tornam-se sub-ótimos em algum critério. Desse modo, o objetivo deste trabalho é o de coletar dados da Área de Preservação Permanente localizada às margens do Rio Sorocaba, em uma mata ciliar que margeia o loteamento *Ibiti Royal Park*, para servir como caso-exemplo da estratégia embarcada proposta para modelagem e monitoramento de sinais. Particularmente, valendo-se de dados de temperatura e umidade da microrregião híbrida em apreço, assim como da metodologia utilizada que baseia-se em Internet das Coisas, Inteligência Artificial e Processamento Digital de Sinais provenientes de microcontroladores, caracterizam-se os requisitos de engenharia para suporte ao método proposto e os resultados estatísticos obtidos refutam a hipótese popular de uma diferença térmica de 2℃ entre a APP e a área urbana, demonstrando homogeneidade climática na microrregião. Notavelmente, os resultados obtidos no ecossistema-teste permitem concluir que a contribuição principal está na metodologia empregada, e não na diversidade dos dados coletados, estabelecendo um procedimento científico adequado que pode ser utilizado como prática replicável, isto é, *Yokoten*. O presente trabalho baseia-se no conceito de Yokoten — termo do Sistema Toyota de Produção (Lean) que designa o 'desdobramento horizontal' para o compartilhamento de melhores práticas e soluções. Nesta pesquisa, aplica-se esse princípio à coleta e democratização de dados, propondo um modelo de difusão de conhecimento que visa beneficiar o agronegócio, a indústria contemporânea e a ciência.

Palavras-chave: Sistemas embarcados; Internet das coisas; Inteligência artificial; Processamento de sinais; Estações meteorológicas.

ABSTRACT

Research and data collection using embedded engineering systems have become essential in various applications today. Nevertheless, there are gaps in these procedures, which usually become suboptimal in specific criteria. Therefore, the objective of this research is to collect data from the Permanent Preservation Area (APP) located along the banks of the Sorocaba River, in a riparian forest bordering the Ibiti Royal Park subdivision, to serve as a case study for the proposed embedded strategy for signal modeling and monitoring. Particularly, by utilizing temperature and humidity data from the hybrid micro-region in question, as well as the methodology employed—which is based on the Internet of Things, Artificial Intelligence, and Digital Signal Processing from microcontrollers—the engineering requirements to support the proposed method are characterized, and the statistical results obtained refute the popular hypothesis of a 2℃ thermal difference between the APP and the urban area, demonstrating climatic homogeneity in the micro-region. Notably, the results obtained in the test ecosystem allow us to conclude that the main contribution lies in the methodology employed, and not in the diversity of the collected data, establishing an appropriate scientific procedure that can be used as a replicable practice, that is, Yokoten. The present work is based on the concept of Yokoten term from the Toyota Production System (Lean) that refers to 'horizontal deployment' for sharing best practices and solutions. In this research, this principle is applied to data collection and democratization, proposing a knowledge diffusion model that aims to benefit agribusiness, contemporary industry, and science.

Keywords: Embedded systems; Internet of things; Artificial intelligence; Signal processing; Meteorological stations.

LISTA DE FIGURAS

1.1 Área do *Ibiti Royal Park*, visualizada pelo *Google Earth*. 12

2.1 Método Hipotético-Dedutivo e PDCA 14
2.2 *Loop* de Qualidade e Processo Científico - Metodologia Integrada 16
2.3 Evolução histórica da Indústria 17
2.4 Tela do Simulador Proteus 18

3.1 Arduino UNO - *Datasheet* 2025 21
3.2 Família Pico 1 e o PICO W em destaque. 21
3.3 Raspberry Pi 5 - Reddit 2025 22

4.1 *Overview* do Sistema Embarcado. 23
4.2 Localização dos sensores de coleta de dados: $S_1, S_2, ..., S_{14}$. 24
4.3 Abertura e exemplo do Simulador (circuito Arduino) 26
4.5 Circuito montado do primeiro simulador 26
4.4 Diagrama esquemático do primeiro simulador 27
4.6 Arduino com SD Card, DHT22 e RTC 28
4.7 Protótipo 1 Estação Meteorológica (EM) com Arduino 28
4.8 Ambiente Arduino IDE 29
4.9 Funcionamento do Protótipo 2 Raspberry Pico W 31
4.10 Arquitetura MQTT 32
4.11 Kit do servidor sem embalagem 34
4.12 *Broker* MQTT completo e 2 EM com Raspberry Pico W 34
4.13 Funcionamento Broker Cloud 35

5.1 Detalhes das Estações Meteorológicas – Parte 1 39
5.2 Detalhes das Estações Meteorológicas – Parte 2 39
5.3 Infográfico da arquitetura dos equipamentos 42
5.4 Características do Sensor DHT22 / AM2302 42
5.5 Diagrama em blocos interno 43
5.6 Infográfico da extensão do Wi-Fi via onda de rádio 45
5.7 Comunicação de rádio 46
5.8 Estações Meteorológicas FACENS e FATEC (INMET) 47
5.9 Conceito de Estação Semelhante 48
5.10 Rosa dos Ventos Máximo 48
5.11 Rosa dos Ventos próximo de Sorocaba 49
5.12 Interpolação IDW 49
5.13 *Deep Learning* LSTM 51
5.14 Projeção para 6 meses 51

5.15 Estação Meteorológica no Jardim 55
5.16 Infográfico da estação meteorologica fixa no jardim. 56
5.17 Gráficos individuais por EM em dois tempos de coleta: 10 minutos. 58
5.18 Gráficos individuais por EM em dois tempos de coleta: 10/2 minutos. 58

6.1 Ferramenta Google Colab 61
6.2 Conclusão do Projeto 62
6.3 Guia de como concluir o Master Degree 63
6.4 Cronograma versão final do TAP - Termo de Abertura do Projeto 67

SUMÁRIO

1 INTRODUÇÃO **11**
1.1 Objetivos 11

2 Fundamentação Teórica **14**
2.1 Informações Complementares 15
2.2 Indústria 4.0 16
2.3 Simuladores 18

3 PLACAS DE DESENVOLVIMENTO **20**
3.1 Arduino Uno 20
3.2 Raspberry Pico W 21
3.3 Raspberry Pi 5 22

4 A ABORDAGEM PROPOSTA **23**
4.1 Coleta de Temperatura e Umidade 24
4.2 Desenvolvimento da Estação Meteorológica 25
4.3 *Hardware* e *Software* 25
4.4 Simulador em *Proteus* 25
4.5 Diagrama Esquemático Unifilar 26
4.6 Armazenamento no *SD Card* 27
4.7 Primeiras Amostras com o Protótipo 1 29
4.8 Protótipo 2: Usando processador *Raspberry* PI 5 e PI Pico W 30
4.9 Arquitetura e Componentes MQTT 31
4.9.1 Aspectos Técnicos do MQTT 32
4.9.2 Mosquitto 33
4.9.3 *Hardware* para montar o *Broker MQTT* 33
4.9.4 MQTT *Cloud* 35
4.10 Processamento de Sinais e Dados no Sistema Proposto 36
4.10.1 Aquisição e Processamento Digital de Sinais (DSP) no *Edge* 36
4.10.2 Telemetria e Processamento de Dados 36
4.10.3 Processamento de Sinais Temporais (Reconstrução e Predição) . . . 37

5 COLETA DE DADOS, TESTES E RESULTADOS **38**
5.1 Visão Geral: da coleta à análise 38
5.2 Posicionamento e Distribuição dos Coletores 40
5.3 Registro e Sincronização dos Dados 40
5.4 Frequência e Duração da Coleta 40
5.5 Análise e Ferramentas 41

5.6 Sensor de Temperatura 41
5.7 Wi-FI a longas distâncias 44
5.8 Acessórios 45
5.9 Estações Meteorológicas 46
5.10 Estimativa do Método IDW e LSTM 50
5.11 LSTM 52
5.12 Comparação entre Arquiteturas 53
5.13 Justificação da Escolha da LSTM 53
5.14 Problemas Pontuais Diversos e Obstáculos Encontrados 54
5.15 Lições Aprendidas após operação contínua prolongada (meses de execução) 57

6 CONSIDERAÇÕES FINAIS E CONCLUSÕES 59
6.1 Considerações Finais 59
6.2 Conclusão Final 60

REFERÊNCIAS 64

APÊNDICES 67

1 – INTRODUÇÃO

Atualmente, o interesse geral pelos sistemas embarcados na área de tecnologia e engenharia é crescente. Fundamentalmente, quando tais sistemas integram-se com aqueles baseados em Inteligência Artificial (IA), desafios diversos, tais como os de capacidade de processamento, tempo de execução, entre outros, fazem-se presentes. Os conceitos supracitados fundamentam o desenvolvimento deste trabalho[1].

Delimita-se, assim, o escopo desta pesquisa, originalmente concebida a partir de um desafio industrial de *troubleshooting* em sistemas embarcados e IA alimentada por *datasets* oriundos de grandezas fabris. Notavelmente, esta pesquisa gerou aprendizados valiosos para o autor, fomentando o interesse pela carreira acadêmica. As dificuldades técnicas e barreiras enfrentadas — em especial a restrição de acesso a esses bancos de dados — não apenas motivaram a mudança de direção da pesquisa, mas também proporcionaram valiosas lições sobre adaptação e resolução de problemas, habilidades essenciais para a prática da engenharia e a formação de pesquisadores.

Particularmente, foi realizada a análise, implementação e avaliação de uma ferramenta de monitoramento no Ambiente de Preservação Permanente (APP) *Ibiti Royal Park*, visando, portanto, viabilizar a pesquisa e consolidar o conhecimento técnico e científico necessário para o desenvolvimento deste trabalho. Conforme o Novo Código Florestal Brasileiro (NCFB), Lei 12.651/12, uma APP é definida como um ecossistema com a função primordial de preservar a paisagem, a geologia, a hidrografia, a fauna, a flora, a beleza natural e o bem-estar da população.

Registre-se que, para o desenvolvimento deste projeto, foi elaborado um Termo de Abertura do Projeto (TAP) (SILVA, 2025), utilizando uma metodologia híbrida. Nesse contexto, a detalhada elaboração do cronograma foi fundamental para servir de guia em todas as etapas da pesquisa.

1.1 OBJETIVOS

Diante do exposto preliminarmente, este projeto tem como objetivo projetar, construir e utilizar um sistema embarcado de Engenharia Elétrica para investigar e comparar dados ambientais entre uma Área Urbana e uma APP, particularmente valendo-se das grandezas temperatura e umidade e integrando métodos de engenharia com conceitos de Indústria 4.0 (FERNANDEZ; ARIAS-ARANDA; BUSTIZA-SANCHEZ, 2024). Para isso, são integrados *hardware* e *software* almejando a concepção de um protótipo de estação meteorológica.

[1] Excertos deste trabalho foram publicados na revista Artefactum, em forma de resumo.

SILVA, L. R. da; GUIDO, R. C. Análise embarcada de dados de temperatura e umidade da área de preservação permanente do Ibiti Royal Park. Artefactum, [São Paulo], v. 25, n. 1, p. 1-17, 2026. DOI: https://doi.org/10.23900/artefactum.v25i1.2521. Disponível em: https://artefactumjournal.com/index.php/artefactum/article/view/2521. Acesso em: 10 abr. 2026.

Após a coleta, os dados passam por um processo de tratamento, a fim de validar ou refutar a hipótese popular de que as temperaturas em áreas urbanas são superiores, em cerca de 2℃, às encontradas em áreas de mata ou APP, conforme ilustrado e detalhado nos capítulos seguintes.

Os detalhes que envolvem os objetivos, caracterizando as suas especificidades, incluem:

- **Localizar e Caracterizar o Loteamento**: Conforme a Figura 1.1, o loteamento *Ibiti Royal Park*, situado na zona norte de Sorocaba - SP, compreende 1.177 lotes distribuídos em aproximadamente 700.000 m² e com um perímetro de 3.871 m.

  Seus limites confrontam os bairros Parque Vitória Régia, Jardim Sorocaba Parque, Jardim Santa Catarina, Jardim Santa Lúcia, Jardim Santa Marina e Jardim Santo André II. Adicionalmente, faz divisa com os loteamentos residenciais Ibiti do Paço e Ibiti Reserva, o loteamento industrial LINC e os condomínios de edifícios Flora e Villagio Ibiti. Uma das divisas da área abrange parte de uma mata que integra um corredor ecológico ao longo do Rio Sorocaba. Este corredor se estende por mais de 227 km, conectando diversos nichos de matas florestais nas proximidades de cidades como Ibiúna, Votorantim, Sorocaba, Iperó, Boituva, Tatuí, Cerquilho, Jumirim e Laranjal Paulista. Ele deságua no Rio Tietê, interligandos também a outras áreas de mata, como as da cidade de Itu e Tietê, que conecta a este mesmo corredor.

Figura 1.1 – Área do *Ibiti Royal Park*, visualizada pelo *Google Earth*.



Fonte: Elaborada pelo autor (2026).

- **Tratar os Dados Meteorológicos com IA**: Neste projeto, foi proposta a implementação de um sistema embarcado para coleta de dados meteorológicos locais (in situ), utilizando a estação do INMET de Sorocaba como referência para calibração. A metodologia integra a aplicação da rede neural artificial Long Short-Term Memory (LSTM) sobre a base de dados gerada, visando ao desenvolvimento de modelos preditivos para variáveis específicas

- **Dissertação:** Esta dissertação detalha o desenvolvimento e a implementação de um sistema embarcado para a coleta de dados de temperatura e umidade, visando comparar o microclima da Área de Preservação Permanente (APP) do Ibiti Royal Park com a sua vizinhança urbana para testar a hipótese popular de uma diferença térmica de

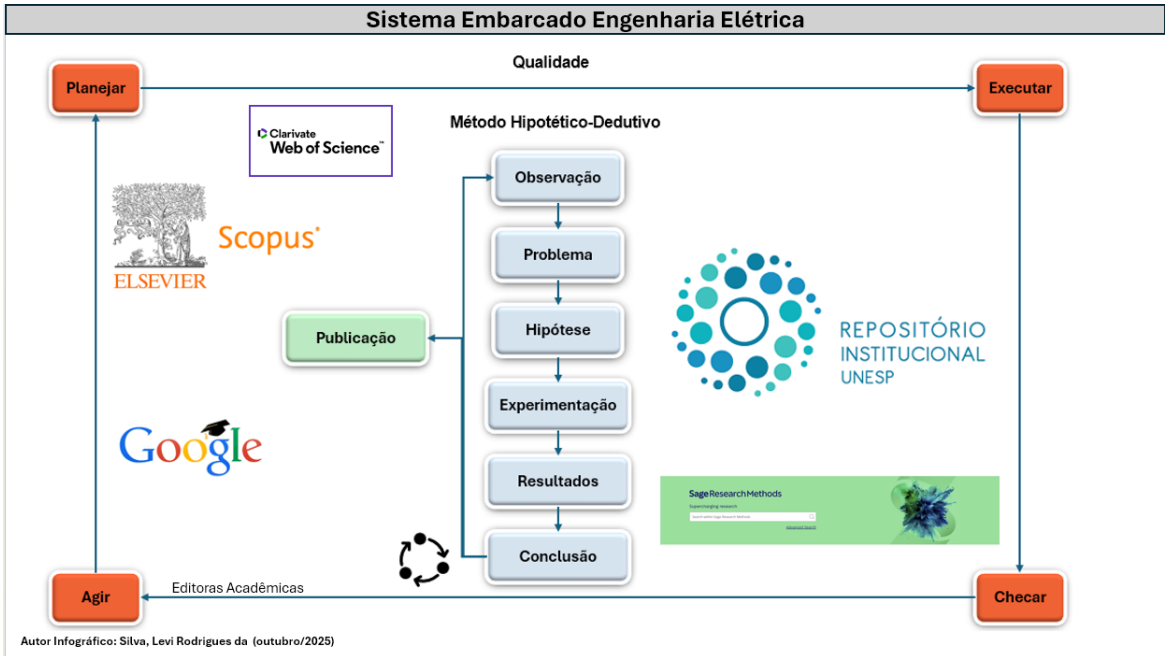
2℃. O documento percorre desde a fundamentação teórica baseada na Indústria 4.0 e Internet das Coisas (IoT) , até a construção dos protótipos das estações meteorológicas utilizando microcontroladores Arduino e Raspberry Pi integrados ao protocolo de comunicação MQTT(RODRIGUES, 2025). Por fim, o texto aborda a metodologia de coleta em campo, a aplicação de modelos estatísticos e de Inteligência Artificial (redes LSTM) para o tratamento dos dados, e culmina na apresentação dos resultados — que refutam a hipótese inicial, além de discutir as lições técnicas aprendidas durante a operação contínua do equipamento.

2 – FUNDAMENTAÇÃO TEÓRICA

Conceitos Gerais Envolvendo Metodologia de Pesquisa Conceitos Gerais Envolvendo Metodologia de Pesquisa Um trabalho científico, independentemente de seu porte, é fundamentalmente uma investigação que parte de uma pergunta ou de um problema e da curiosidade em encontrar respostas e soluções. Para que seja considerado científico, ele deve basear-se em dados observáveis, empíricos e mensuráveis. Existem diversas metodologias para trabalhar com esses dados. Uma abordagem comum e atualmente utilizada é o método hipotético-dedutivo, conforme descrito no livro (MARCONI; LAKATOS, 2003). Este método opera em um ciclo sequencial, geralmente composto por seis etapas, que se realimentam em uma sequência: Observação, Elaboração do Problema, Formulação de Hipóteses, Experimentação, Análise dos Resultados e Conclusão.

Esta abordagem hipotética-dedutiva se alia ao moderno conceito da qualidade e uma de suas ferramentas chamada *Plan-Do-Check-Action* (PDCA) que também é um *loop* que mantêm-se ativo até que se chegue ao resultado desejável. Portanto, planeja-se, executa-se, verifica-se o resultado e toma-se uma ação, conforme a Figura 2.1.

Figura 2.1 – Método Hipotético-Dedutivo e PDCA



Fonte: Elaborada pelo autor (2026).

Assim, a condução de um trabalho científico exige um planejamento meticuloso, o que ressalta a importância do conhecimento gerenciamento de projetos. Nesse sentido, o cronograma é uma ferramenta essencial, atuando como um fio condutor que conecta todas

as etapas, desde o início até a conclusão do projeto.

Toda a documentação e a metodologia empregadas foram rigorosamente registradas nesta monografia. Da mesma forma, as matérias de pesquisa e estudo utilizadas foram devidamente citadas. É importante notar que a ciência, em sua essência, é uma busca contínua por entender o mundo, utilizando a razão, a observação e a experimentação. Portanto, o resultado de um estudo científico não representa uma verdade absoluta e imutável, mas sim o conhecimento mais confiável e útil obtido até o momento, sempre sujeito à avaliação e revisão.

Em tempos de comunicação rápida e abrangente, é essencial ter cuidado para que os dados analisados não sejam frutos de *fake news* e pseudociência. A pseudociência refere-se a alegações, crenças ou práticas apresentadas como científicas, mas que não aderem aos padrões e métodos rigorosos da ciência. Frequentemente, carece de evidências de apoio, testabilidade e revisão por pares. Neste trabalho, houve portanto um cuidado para excluir materiais de pseudociência que pudessem incorporar citações de trabalhos que nunca foram concluídos ou aprovados por pares pois é notável que a dificuldade de acesso do público a materiais científicos genuínos, aliada à crescente presença da mídia e da alta tecnologia na vida das pessoas, tem levado à aceitação de materiais pseudocientíficos como se fossem ciência, acarretando riscos e resultados preocupantes.

2.1 INFORMAÇÕES COMPLEMENTARES

No geral, conforme a Figura 2.2a e 2.2b, observa-se o seguinte conjunto de passos que guiaram este projeto:

- **Método Hipotético-Dedutivo (MARCONI; LAKATOS, 2003):** Como parte do levantamento bibliográfico, realizaram-se buscas sistemáticas em bases de dados e plataformas de pesquisa científica, tais como *Scopus*, Google Acadêmico e o acervo institucional da UNESP. O intuito dessa investigação foi mapear trabalhos e metodologias estritamente semelhantes ao escopo deste projeto. Ressalta-se, contudo, que não foram encontrados estudos prévios que contemplassem integralmente as mesmas características e integrações tecnológicas propostas nesta pesquisa. Tal constatação evidencia a originalidade do presente trabalho e justifica a ausência de uma seção dedicada à comparação com literaturas congêneres.

- Ferramenta de conceito do gerenciamento de gestão da qualidade PDCA, ou seja, planejar (P) Identificar o problema, analisar causas, definir metas em conjunto com o elaborar um plano de ação (TAP).

  Executar (*Do*), realizando cada passo planejado no sistema de gerenciamento híbrido (TAP).

  Checar (*Check*) verificando os resultados, comparando com o planejado e analisar o desempenho.

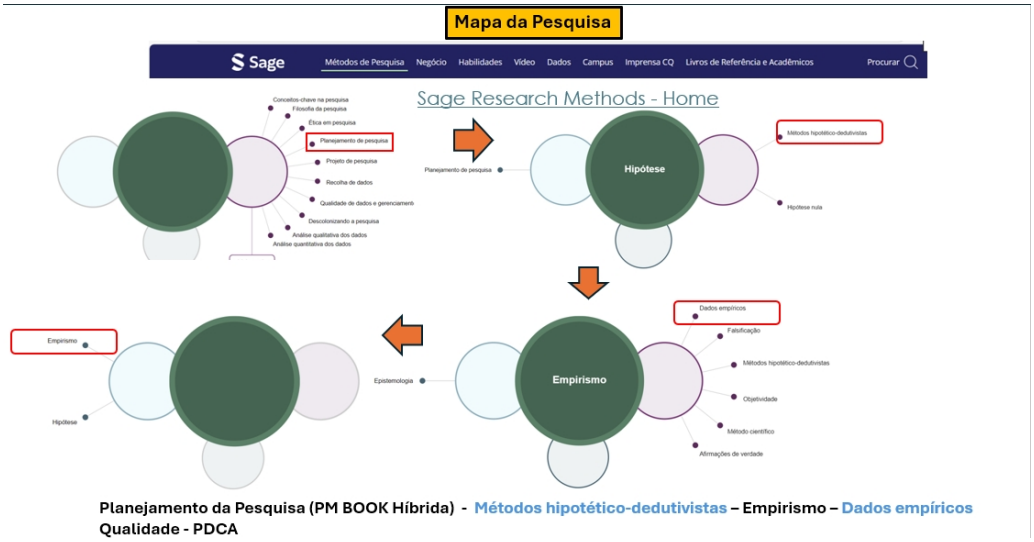
  Agir (*Act*) tomando as ações necessárias, ou seja, se estiver conforme desejado, padronizar o processo, caso negativo reiniciar o *loop* novamente até atingir o esperado.

  A base do ciclo de melhoria contínua foi estabelecida inicialmente por Shewhart (1931), introduzindo o controle estatístico de qualidade. Posteriormente, Shewhart (1939) ex-

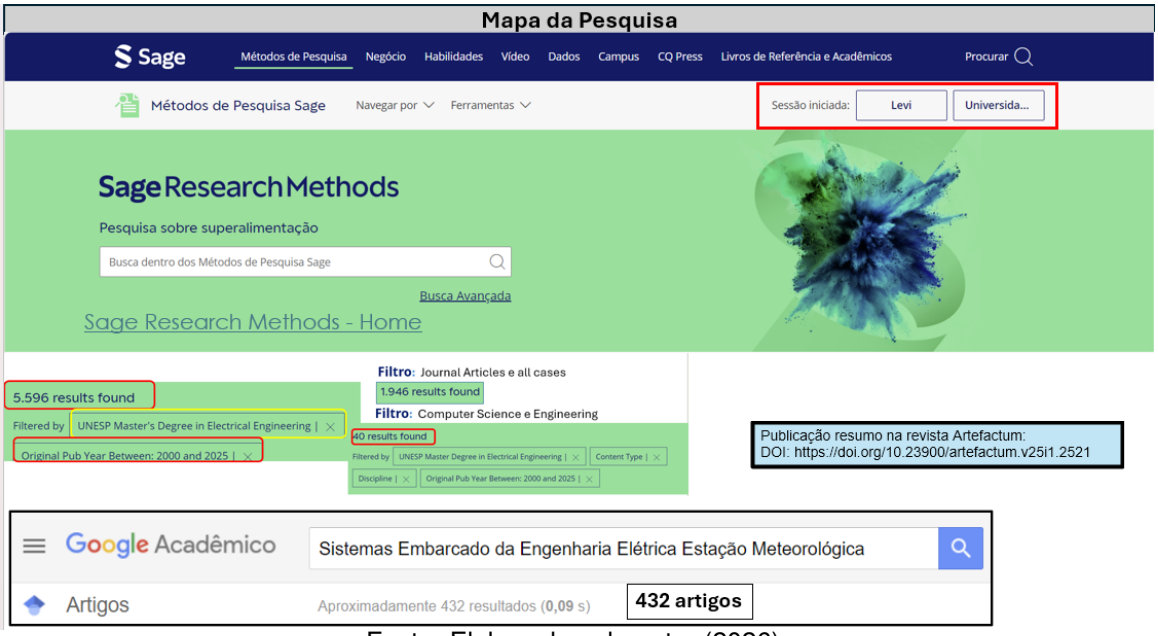
pandiu esses conceitos em uma obra editada por W. Edwards Deming, que viria a popularizar o método mundialmente (DEMING, 1986).

Figura 2.2 – *Loop* de Qualidade e Processo Científico - Metodologia Integrada

(a) *Loop* de Qualidade



(b) Processo Científico



Fonte: Elaborada pelo autor (2026).

2.2 INDÚSTRIA 4.0

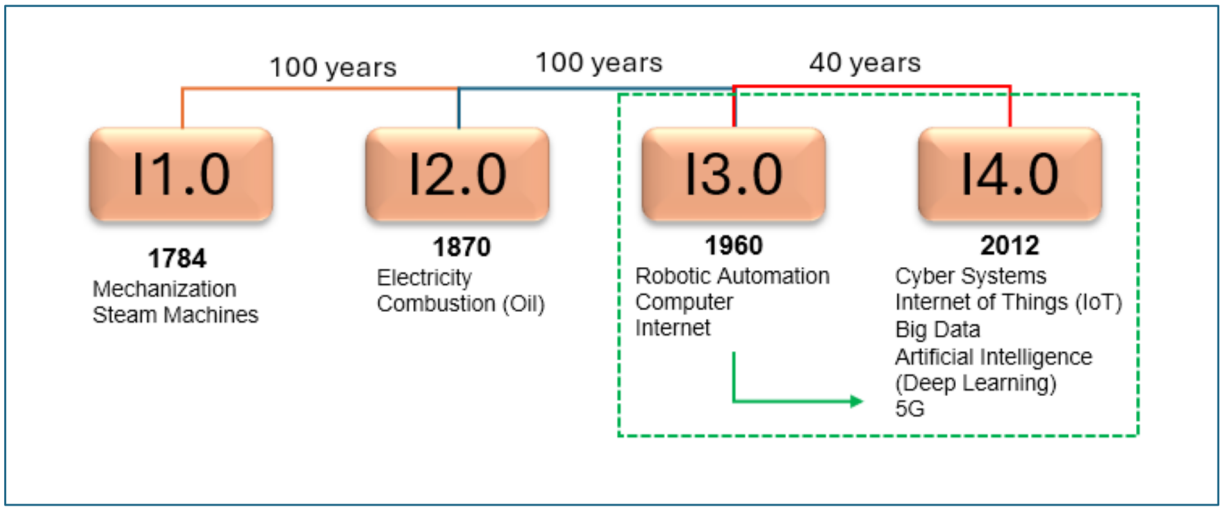
O sistema embarcado (BARROS; CAVALCANTE, 2026) é um pilar essencial para o conceito de Indústria 4.0, a qual remonta ao período em que a produção artesanal já não supria a demanda crescente. O avanço científico impulsionou o desenvolvimento de processos industriais, marcando o início da Indústria 1.0 em 1784. A invenção das máquinas a vapor e a mecanização, como no tear, permitiram a produção em larga escala, viabilizando

a distribuição e venda para um público significativamente maior.

A chegada da eletricidade e dos motores a combustão, derivados do petróleo, desencadeou a Indústria 2.0 em 1870. Esta nova fase revolucionou os processos de produção em escala através da linha de montagem, automatizando tarefas e reduzindo a necessidade de mão de obra altamente especializada. Posteriormente, a evolução da eletrônica, a automação robótica e a introdução de computadores e da internet deram origem à Indústria 3.0 em 1960. Tal revolução trouxe consigo a computação e a automação avançada.

Com o advento dos Sistemas Ciberfísicos, da Internet das Coisas (IoT) e da Inteligência Artificial (IA), em 2012, consolidou-se o conceito de Indústria 4.0. Esta revolução, ainda em desenvolvimento, convive com uma realidade em que muitas empresas permanecem na fase da Indústria 3.0. Tecnologias como o SMT (Surface Mount Technology) (SILVA; SILVEIRA, 2023), predominante na produção de eletrônicos (representando mais de 90% da produção atual), e o remanescente PTH (Through-Hole Technology), que em 2024 detém menos de 10% do mercado, são marcos deste estágio de transição . Historicamente, o contacto prático com os processos de SMT e PTH constituiu a base para a introdução aos conceitos da Indústria 4.0, inserindo-se num contexto em que a organização de atuação obteve a certificação neste novo paradigma, contando com participações técnicas indiretas para este alcance. Conforme ilustrado na Figura 2.2, nota-se que, na Indústria 4.0, a engenharia de sistemas embarcados assume uma importância tão crucial quanto a da IA, mantendo-se em constante aprimoramento.

Figura 2.3 – Evolução histórica da Indústria



Fonte: Elaborada pelo autor (2026) com base em Santos (2018) e Fernandez, Arias-Aranda e Bustiza-Sanchez (2024).

Observa-se que as transições entre a Indústria 1.0 e 2.0, e posteriormente para a 3.0, levaram aproximadamente 100 anos. A passagem para a Indústria 4.0 (FERNANDEZ; ARIAS-ARANDA; BUSTIZA-SANCHEZ, 2024)(SANTOS et al., 2018) ocorreu em pouco mais de 40 anos. Assim, acredita-se que a próxima revolução, rumo à Indústria 5.0, demandará ainda menos tempo, dada a alta intensidade das descobertas tecnológicas, a facilidade de acesso à informação e os processos de pesquisa acelerados.

Complementarmente, é interessante destacar que as saídas de um sistema embarcado podem variar: desde *displays*, LEDs e cartões SD até a visualização em tempo real

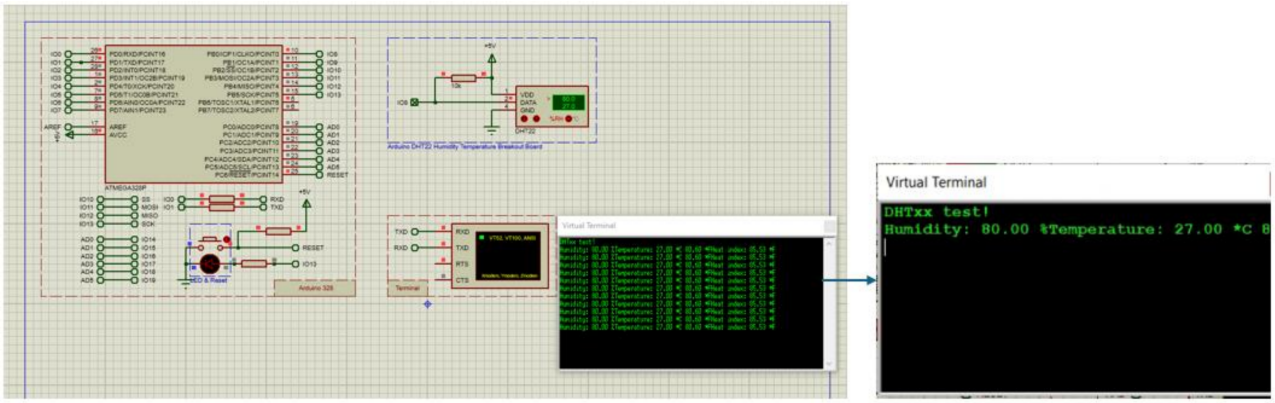
por meio de dispositivos móveis, permitindo ao usuário acompanhar todo o processo. Tais sistemas, portanto, integram o conceito de Internet das Coisas (IoT), combinando *hardware*, *firmware* e *software* embarcados em uma abordagem de Digitalização de Serviços e Tecnologias (DST). A DST refere-se à transformação de processos analógicos em digitais, empregando tecnologias como IoT, IA, computação em nuvem e aplicativos (GOMES, 2022).

O sistema embarcado não é conceito novo: desde a decada de 1960 com os projetos aeroespaciais da NASA, evolui na decada de 1980 na indústria automotiva e chega no século 21 junto com o conceito do *smartphone*, IoT e IA. Um sistema embarcado pode ser todo o sistema ou um subsistema, sendo comum ser um sistema de computação exclusivamente dedicado a uma atividade, otimizado para esta atividade e pode ser em tempo real, que do inglês *Real-Time Operating System* (RTOS), se integrando com um sistema mais amplo com IoT e podendo conter rotinas de IA.

2.3 SIMULADORES

O desenvolvimento de projetos eletrônicos pode usar o recurso de uso de simuladores, como o *Proteus Design Suite*, exemplificado na Figura 2.4, conforme ocorreu neste trabalho.

Figura 2.4 – Tela do Simulador Proteus



Fonte: Elaborada pelo autor (2026).

O *Proteus Design Suite* é uma ferramenta robusta que simplifica o processo de projeto eletrônico, oferecendo um ambiente integrado para captura de projetos, simulação e *layout* de placas de circuito impresso (PCB). Sua capacidade de simular microcontroladores o torna particularmente valioso para desenvolvimento de sistemas embarcados.

Após a verificação das funcionalidades no simulador, é possível criar o esquema elétrico que atende às necessidades do projeto e, posteriormente, desenvolver um protótipo para testes práticos. Caso o protótipo seja bem-sucedido, a versão final do produto pode ser construída. O simulador também oferece a emulação de inúmeros instrumentos de medição, sendo útil na avaliação de quais equipamentos comuns podem ser empregados. Particularmente, para este projeto , a disciplina que contribuiu com este tópico é a *Applied Advanced Embedded System Projects: Machine Learning, IoT and DSP* (AESP), ministrada pelo Prof Dr Ivandro Severino Diniz. Um trabalho final desta disciplina, intitulado “TCC_AESP_Disciplina_final” no Apêndice_3_Sistema Embarcado, serviu de inspiração para o projeto atual. Ressalta-se este simulador foi usado apenas para o projeto do

microcontrolador Arduino. Na sua versão *free*, o microcontrolador Raspberry não é contemplado.

Para este projeto, estratégias de IA foram empregadas em conjunto com ferramentas matemáticas de estatística e probabilidade (CHATER; MANNING, 2006). O objetivo foi o de analisar dados coletados em diferentes posições estratégicas, filtrados e mensurados, para concluir cientificamente se a crença popular de que o APP tem até 2 ℃ a menos que sua vizinha área urbana pode ser refutada ou corroborada.

A manipulação de dados e a geração de análises estatísticas foram realizadas utilizando a linguagem de programação *Python*, em um ambiente de desenvolvimento chamado *Google Colab*. Por meio dessa análise, são obtidas as conclusões do experimento. A IA é um campo vasto, abrangendo diversas subáreas. Para este projeto, a estratégia selecionada é o *Deep Learning*, especificamente o aprendizado supervisionado com redes LSTM, para previsões, sendo uma Rede Neural Recorrente (RNN) projetada especificamente para lidar com séries temporais. Diferente de modelos estatísticos simples, a LSTM consegue memorizar eventos passados por longos períodos, o que é crucial para captar padrões diários e sazonais de temperaturas até obter a previsão futura. Combinada com os princípios do aprendizado supervisionado, a arquitetura LSTM permite treinar a rede com dados históricos rotulados para tarefas de previsão e análise de séries temporais. Sua capacidade de processar dados brutos sequenciais permite identificar padrões térmicos complexos que correlacionam as diferentes posições estratégicas monitoradas.

3 – PLACAS DE DESENVOLVIMENTO

3.1 ARDUINO UNO

O Arduino Uno, exibido na Figura 3.1, é uma placa de microcontrolador baseada no *chip* ATmega328P. É parte da plataforma Arduino, que é uma combinação de *hardware* de código aberto e *software* elementar para usar. O “Uno” significa “um” em Italiano e foi escolhido para marcar o lançamento da versão 1.0 do *software* Arduino.

O principal objetivo do Arduino é tornar a criação de projetos eletrônicos acessível a todos, desde iniciantes até engenheiros experientes. Ele consegue isso por meio de:

- *hardware* simples e aberto: os esquemáticos da placa são públicos, o que significa que qualquer um pode fabricar ou modificar um Arduino Uno. A placa é robusta e fácil de conectar.
- *software* intuitivo: o Ambiente de Desenvolvimento Integrado (IDE) do Arduino é modesto e usa uma linguagem de programação baseada em C++, mas com funções e estruturas que facilitam muito a escrita de código para interagir com o *hardware*.
- Comunidade Ativa: há uma vasta comunidade *online* de usuários, com inúmeros tutoriais, projetos e fóruns de ajuda.

Embora o Arduino Uno seja versátil para uma vasta gama de projetos, desde os mais simples aos mais complexos, na aplicação desenvolvida ele desempenha a função específica de coletar dados do sensor de temperatura e armazená-los em um cartão de dados. Esse processo foi executado por meio de um programa elaborado na plataforma Arduino IDE, o qual foi carregado na memória do microcontrolador via porta USB.

Figura 3.1 – Arduino UNO - *Datasheet* 2025

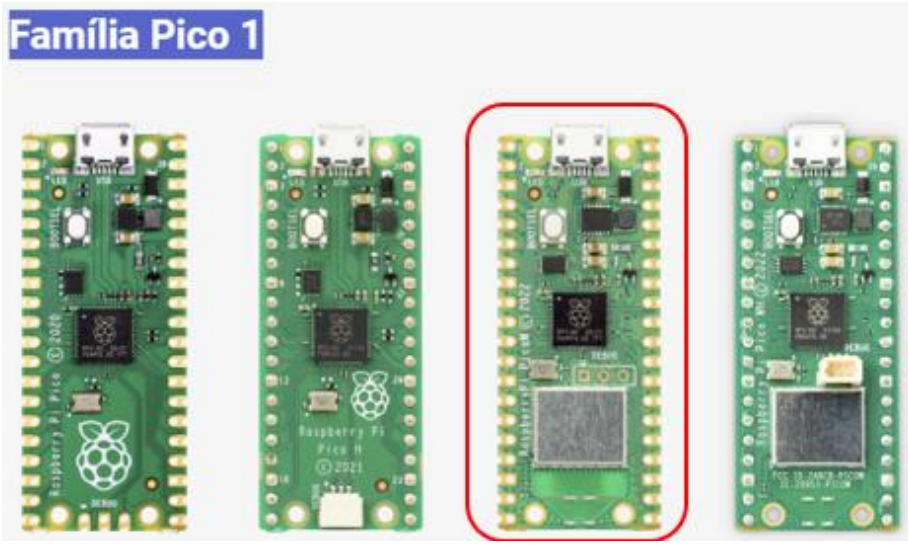


Fonte: *Datasheet* (2026).

3.2 RASPBERRY PICO W

O Raspberry Pi Pico W, exibido na Figura 3.2, é uma versão do microcontrolador Raspberry Pi Pico, mas com conectividade *Wi-Fi* integrada. Ele é ideal para projetos de IoT, automação, robótica e outros sistemas embarcados.

Figura 3.2 – Família Pico 1 e o PICO W em destaque.



Fonte: *Datasheet* (2026).

Em seu *datasheet*, é possível ver as funções de seus terminais e todas os seus detalhes. O Pico 2 W é basicamente uma versão mais potente e segura do Pico W. Enquanto o Pico W continua sendo uma excelente escolha para projetos básicos e com baixo orçamento, o Pico 2 W é ideal para quem precisa de mais poder de processamento, mais memória e recursos de segurança aprimorados, como para aplicações industriais ou de IoT que exigem maior robustez.

3.3 RASPBERRY PI 5

O Raspberry Pi 5, exibido na Figura 3.3, é um sistema de alta performance, caracterizando um computador completo integrado em uma única placa. Ele possui todos os componentes essenciais de um computador tradicional, ideal para uso como:

- miniPC com saída 4K, suporte a múltiplos monitores e boa fluidez gráfica
- estação de trabalho, *media center*, servidores domésticos, automação residencial, entre outros.
- projetos avançados que requerem expansão via PCIe, câmeras MIPI, e conectividade ampla (*Wi-Fi* 5 + *Gigabit Ethernet* + *Bluetooth* 5).

Figura 3.3 – Raspberry Pi 5 - Reddit 2025



Fonte: *Datasheet* (2026).

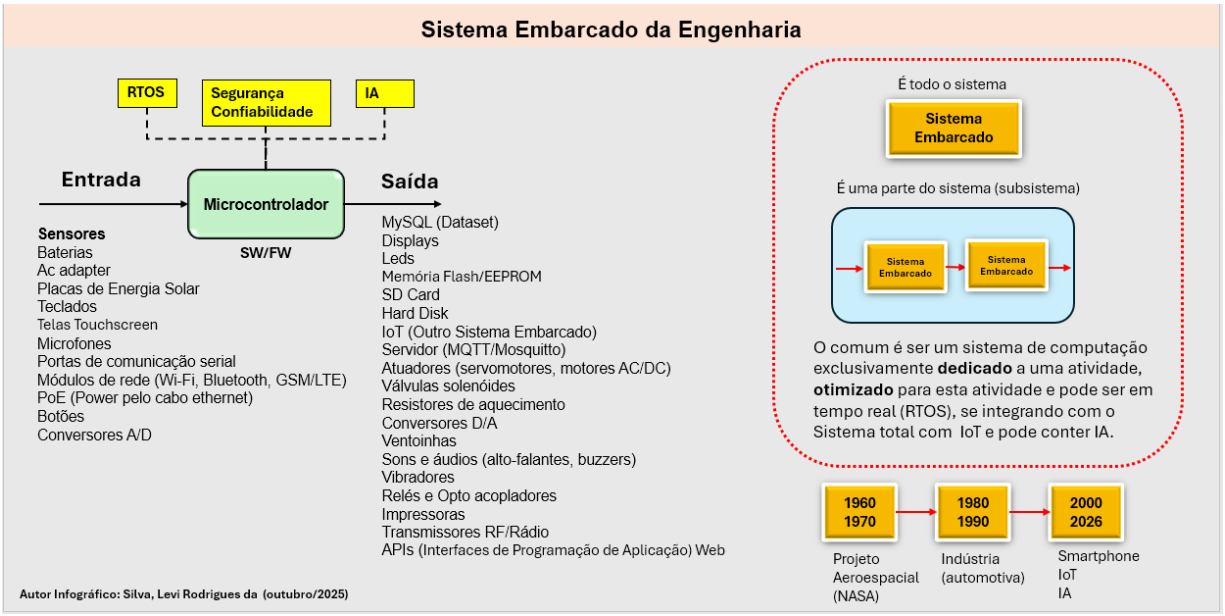
Esta versão, a mais recente no período de 2024/2025, apresenta arquitetura de hardware e software (HW/SW) semelhante à de um computador comum. Em virtude de sua alta performance, o equipamento é frequentemente denominado "PC de bolso". O dispositivo é compatível com sistemas operacionais como Linux e Ubuntu, sendo direcionado a aplicações avançadas que incluem robótica, automação, media centers, servidores e aprendizado de Inteligência Artificial (IA). No presente projeto, a placa atua como servidor, demandando a conexão de periféricos como monitor, teclado e mouse para viabilizar e facilitar a sua manipulação. Adicionalmente, o componente foi adquirido em conjunto com acessórios de montagem e refrigeração, tais como ventoinha, chassi e dissipador de calor (heatsink). O respectivo *datasheet* encontra-se disponível nos anexos do trabalho.

4 – A ABORDAGEM PROPOSTA

Conforme mencionado em Capítulo anterior, este projeto requer a coleta de dados de temperatura e umidade em diversas localizações do *Ibiti Royal Park*. A estrutura completa do sistema embarcado responsável para tal é a que consta na Figura 4.2. Os dados coletados são armazenados em um banco de dados para posterior estudo e análise.

De um modo geral, o conceito de sistema embarcado fundamenta-se em uma abordagem de engenharia destinada à coleta de diversos tipos de dados e ao seu posterior armazenamento para análise e obtenção de conclusões. Conforme ilustrado na Figura 3.1 — que retrata a arquitetura genérica e conceitual desses sistemas —, a etapa de entrada é tipicamente composta por sensores, aliados a outros componentes essenciais para a aquisição de informações. Para o processamento, controle, apresentação ou armazenamento desses dados, faz-se o uso central de microcontroladores. Adicionalmente, em casos onde há a necessidade de comunicação com outras máquinas e usuários, integram-se protocolos de rede para essa interação, a exemplo da arquitetura MQTT operando por meio de um Broker. É fundamentado exatamente sobre essa base teórica e conceitual genérica que o sistema proposto nesta pesquisa foi estruturado.

Figura 4.1 – *Overview* do Sistema Embarcado.



Fonte: Elaborada pelo autor (2026).

4.1 COLETA DE TEMPERATURA E UMIDADE

Na Figura 4.2 apresenta-se um exemplo de possíveis locais onde dados de temperatura e umidade devem ser coletados e armazenados em sistema MySQL, conforme exemplo da Tabela 4.1. Uma vez que todos os dados forem coletados, em diferentes dias e até períodos como as estações do ano, dispõe-se de uma fonte segura de informações para realizar análises.

Figura 4.2 – Localização dos sensores de coleta de dados: $S_1, S_2, ..., S_{14}$.



Fonte: Elaborada pelo autor (2026).

Tabela 4.1 – Exemplo de MySQL para armazenar dados.

Host Name	**Temp. (℃)**	**Umidade (%)**	**Data/Hora**	**Localização**
Pico_Wf0fc	18.1	73.5	2025-09-03 06:58:28	-23.44938° / -47.46318°
Pico_Wf008	20.3	70.5	2025-09-03 06:58:28	-23.44536° / -47.45749°
Pico_Wfabb	21.2	69.5	2025-09-03 06:58:28	-23.44724° / -47.45502°

Fonte: Elaborada pelo autor (2026).

Microcontroladores Arduino, *Raspberry* Pi 5 e Pico W, são responsáveis por gerenciar os sensores nas localizações estabelecidas. Por meio de linguagens de programação como C++/C e *MicroPython*, esses dispositivos coletam os dados necessários, que são posteriormente apresentados ou armazenados para análise.

4.2 DESENVOLVIMENTO DA ESTAÇÃO METEOROLÓGICA

O desenvolvimento do primeiro protótipo de coletor de dados de temperatura e umidade emprega o microcontrolador Arduino Uno. Este conceito de sistema embarcado já é amplamente aplicado e o circuito tem como entrada principal o sensor DHT22, responsável pela detecção de umidade e temperatura. Os dados coletados são processados pelo Arduino e os resultados são então exibidos em um *display* OLED e em uma janela do ambiente de trabalho do Arduino, além de serem armazenados em um banco de dados nos formatos CSV / TXT. Adicionalmente, um LED indica o fornecimento de alimentação ao sistema e um novo dado é capturado a cada intervalo de tempo.

4.3 *HARDWARE* E *SOFTWARE*

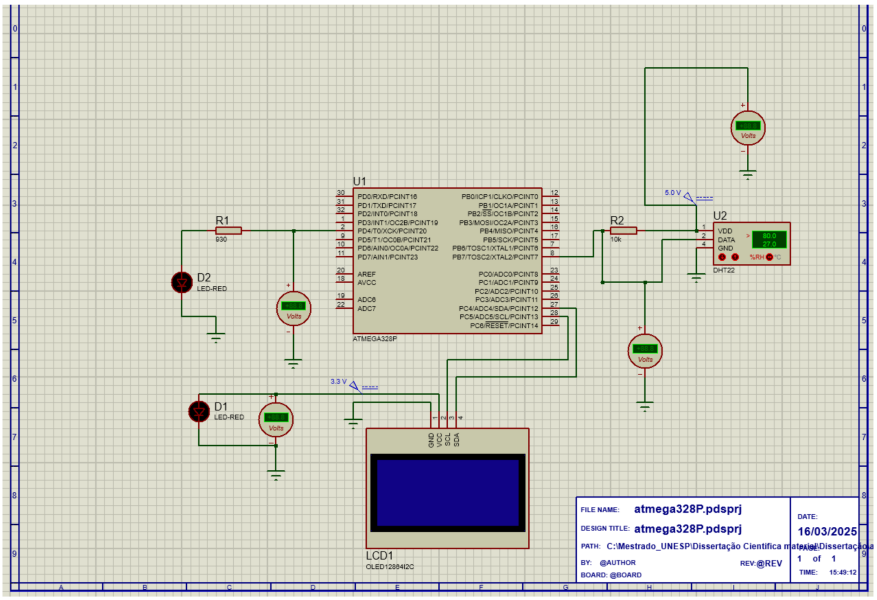
Para a montagem do primeiro protótipo e a subsequente coleta de dados, é crucial ter em mãos o *hardware* necessário e configurar o ambiente de trabalho do *software*. Essa etapa permite programar o sistema e montar os componentes físicos que viabilizarão a obtenção das informações desejadas. A construção deste protótipo inicial implica coletar as informações de temperatura e umidade e verificar se o sistema está operando de maneira adequada e conforme o esperado.

4.4 SIMULADOR EM *PROTEUS*

Para a validação lógica do sistema, recorreu-se ao simulador Proteus 8 Professional, utilizando a infraestrutura e as licenças institucionais disponibilizadas pela UNESP. O emprego desta ferramenta profissional teve como propósito fundamental o reconhecimento da importância estratégica da simulação no ciclo de desenvolvimento, permitindo a identificação de falhas e a otimização do design antes da implementação física. Conforme ilustrado na Figura 4.3 , após a inicialização do ambiente virtual, o circuito foi montado e testado para a verificação do seu comportamento funcional.

O *software*, desenvolvido em C++ utilizando o ambiente Arduino IDE, gera um arquivo *.hex* que é então transferido para o microcontrolador ATmega328P, componente central do Arduino Uno, para simulação. O simulador apresentou inicialmente resultados inesperados, mas a investigação de falhas revelou que os problemas residiam na configuração e nas bibliotecas do simulador, não na proposta do projeto. O funcionamento no *display* da janela do ambiente de programação do *laptop* ocorreu perfeitamente.

Figura 4.3 – Abertura e exemplo do Simulador (circuito Arduino)



Fonte:*Proteus - Application Framework* 2024

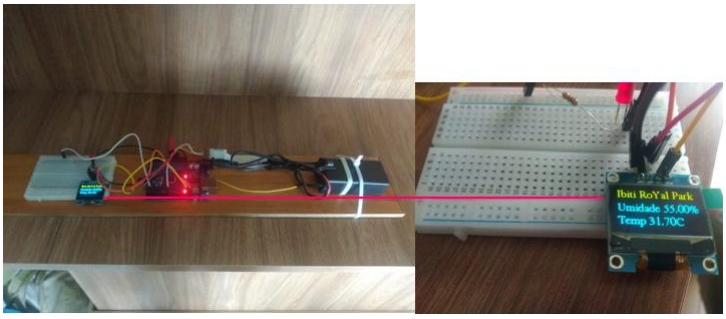
4.5 DIAGRAMA ESQUEMÁTICO UNIFILAR

Inicialmente, a alimentação é provido pelo *laptop* utilizado para dar suporte ao projeto, por meio do cabo USB tipo A/B, conectando na porta USB do Arduino. Esta conexão também é usado para transferir o programa desenvolvido em C++ para o microcontrolador do Arduino, como na Figura 4.4.

O protótipo integrado para a coleta de dados de temperatura e umidade, exibido na Figura 4.5, implicou em funcionamento condizente com as expectativas. Ao ser alimentado por uma fonte DC de 12V conectada ao conector *DC Barrel Plug* (2.5 mm x 5.5 mm) ou por uma bateria de 9.5 VDC, o circuito operou conforme o esperado, alinhando o *hardware* ao *software* desenvolvido.

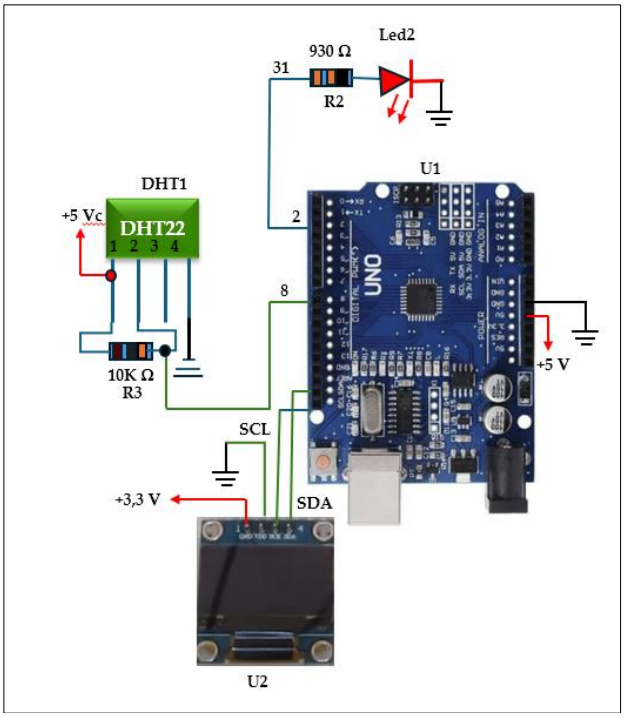
Ao acionar a chave liga/desliga, a bateria energiza o sistema, permitindo que o microcontrolador ATmega328P do Arduino execute o *software* carregado em sua memória. Este *software* é responsável por captar dados de temperatura e umidade do sensor DHT22 a cada 4 segundos, com um LED indicando a realização de cada leitura.

Figura 4.5 – Circuito montado do primeiro simulador



Fonte: Elaborada pelo autor (2026).

Figura 4.4 – Diagrama esquemático do primeiro simulador



Fonte: Elaborada pelo autor (2026).

Os dados coletados são exibidos de duas formas: no monitor serial do Arduino IDE, quando conectado a um *laptop* via cabo USB tipo A/B, ou de forma independente, por meio de um *display* OLED. Embora o primeiro protótipo tenha sido bem-sucedido, a próxima etapa de aprimoramento envolve a transferência desses dados para um cartão SD. O armazenamento em arquivos *.txt* permitirá a posterior análise por rotinas de IA, viabilizando o estudo do impacto dos dados coletados. Esta funcionalidade foi incorporada na montagem do protótipo aprimorado.

4.6 ARMAZENAMENTO NO *SD CARD*

O Arduino Uno, baseado no microcontrolador ATmega328, serve como a unidade central para a coleta de dados. A placa específica utilizada neste projeto apresenta uma ligeira modificação: a entrada do barramento I2C (SDA e SCL).

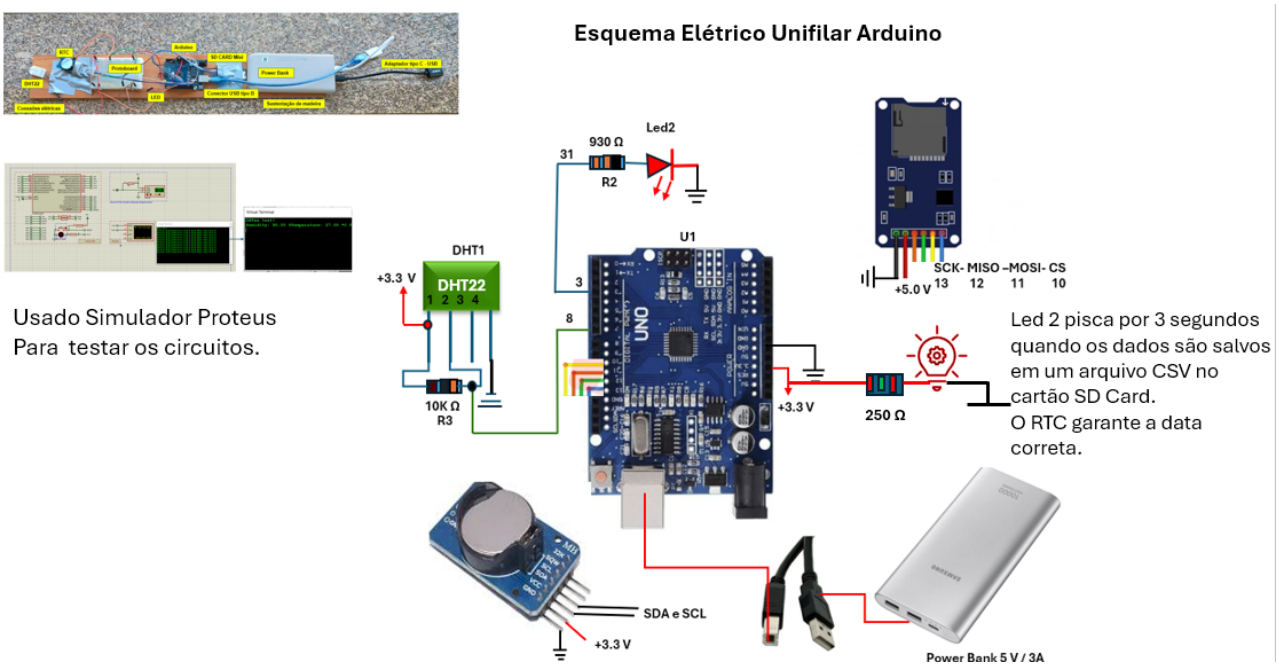
O sensor de temperatura e umidade DHT22 se comunica com o Arduino por meio de um protocolo de pino único, o que simplifica a conexão com um único fio de dados. A troca de informações segue uma comunicação serial que exige temporizações precisas, por isso é necessário o uso de uma biblioteca. Para este projeto, foi utilizado o *DHT sensor library*, disponível no Arduino IDE, que gerencia todo o protocolo de comunicação. A conexão do sensor foi realizada no pino digital 8. Para fornecer um *feedback* visual durante a gravação dos dados, um LED foi conectado ao pino digital 2. O LED piscará por um breve momento, indicando que a leitura de temperatura e umidade foi realizada com sucesso e que os dados estão sendo gravados no *SD Card Mini*.

O diagrama esquemático do projeto atuou como base fundamental para o desenvolvimento do código-fonte em linguagem C++, o qual foi posteriormente inserido e estruturado

no ambiente do Arduino IDE. Após etapas sucessivas de testes e depuração, o código compilado (firmware .hex) foi otimizado e transferido para a memória do microcontrolador por meio da interface USB-B.

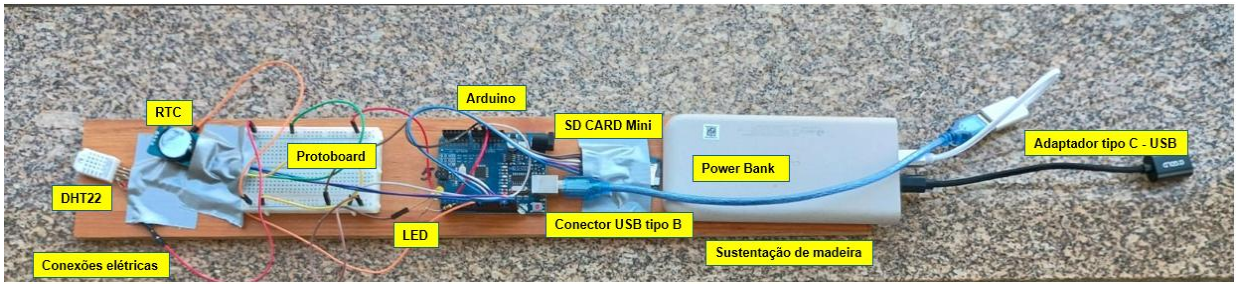
Inicialmente, projetou-se a utilização simultânea do display OLED (via barramento I2C) e do módulo SD Card (via barramento SPI). No entanto, durante os testes práticos, o sistema apresentou travamentos contínuos. A investigação técnica revelou que a falha não decorria de conflitos de comunicação física, mas sim do esgotamento da memória estática (SRAM) do microcontrolador ATmega328P. O Arduino Uno possui apenas 2 KB de SRAM, capacidade insuficiente para alocar simultaneamente o buffer de dados exigido pela biblioteca do cartão SD (512 bytes) e o buffer de vídeo do display OLED (1024 bytes). Diante dessa limitação de hardware, a estratégia de coleta de dados foi ajustada. Nas Figuras 4.6 e 4.7, as montagens estão exibidas.

Figura 4.6 – Arduino com SD Card, DHT22 e RTC



Fonte: Elaborada pelo autor (2026).

Figura 4.7 – Protótipo 1 Estação Meteorológica (EM) com Arduino



Fonte: Elaborada pelo autor (2026).

O *software* final, desenvolvido no ambiente Arduino IDE, é exibido parcialmente na Figura 4.8. Na imagem, ilustra-se o processo de comunicação de teste e a *upload* do có-

digo para a memória *flash* do microcontrolador Arduino Uno por meio da conexão USB. Essa etapa é fundamental, pois garante que a lógica de programação, incluindo a leitura dos dados do sensor DHT22 e a gravação no *SD Card*, seja executada corretamente pelo *hardware*. O sucesso da transferência é validado pelas mensagens de compilação e *upload* no console do Arduino IDE confirmando que o microcontrolador está pronto para iniciar a coleta de dados.

Ressalta-se que a utilização do display OLED foi inviabilizada devido ao esgotamento da memória estática (SRAM) do microcontrolador ATmega328P. Como o Arduino Uno dispõe de apenas 2 KB de SRAM, a alocação simultânea do buffer exigido pela biblioteca do cartão SD (512 bytes) e do buffer de vídeo do OLED (1024 bytes) consumia a quase totalidade dos recursos, resultando no travamento do sistema. Para contornar essa limitação de hardware, a interface foi substituída por um feedback visual simplificado: um LED indicador programado para piscar a cada 20 segundos, confirmando a execução da leitura e a respectiva gravação dos dados no cartão SD.

Figura 4.8 – Ambiente Arduino IDE



Fonte: Elaborada pelo autor (2026).

4.7 PRIMEIRAS AMOSTRAS COM O PROTÓTIPO 1

As primeiras amostras com o Protótipo 1 foram coletadas no ambiente real. Durante o experimento de uma hora, foram coletadas 180 ciclos de amostras, com cada medição sendo lida três vezes para garantir a precisão dos valores, implicando o montante de 540 dados. A geolocalização dos pontos de coleta foi registrada usando um GPS *Garmin MAP* 64X (GARMIN, 2020), como na Tabela 4.2.

O LED serviu como um indicador visual crucial, piscando três vezes a cada minuto, o que confirmou que o sistema estava operando e registrando os dados com sucesso. Nesta fase, a data e a hora das coletas foram anotadas manualmente em uma caderneta, ressaltando a necessidade de integrar um módulo Relógio de Tempo Real (RTC) para registro

automático, o que foi acrescentado no protótipo. Estes dados constam no apêndice de dados.

Tabela 4.2 – Exemplo de MySQL para armazenar dados. Data: 27/03/2025, Média Sorocaba: 26℃ e 62%

Local	Temperatura (℃)	Umidade (%)	Posição Geográfica	Hora
Mata 1	28.05	60.94	S23°26'59.2" W47°27'45.2"	10h46
Mata 2	27.44	64.42	S23°26'59.2" W47°27'45.2"	11h06
Casa 1	31.35	49.30	S23°26'54.8" W47°27'38.9"	11h18
Casa 2	30.69	48.98	S23°26'54.8" W47°27'38.9"	11h38

Fonte: Elaborada pelo autor (2026).

Os principais desafios enfrenteados e análises realizadas foram:

- Duração da Bateria: notou-se um alto consumo de energia; a bateria teve uma autonomia de aproximadamente quatro horas. O consumo energético será um ponto de análise para o desenvolvimento. Utilizando o *Power Bank*, houve uma duração superior de 30 horas, mostrando ser o mais adequado ao projeto.
- Sincronização de Tempo: foi identificada uma defasagem de 4 a 9 segundos entre o cronômetro do celular e o tempo interno do Arduino Uno. Por isso foi adicionado o RTC.

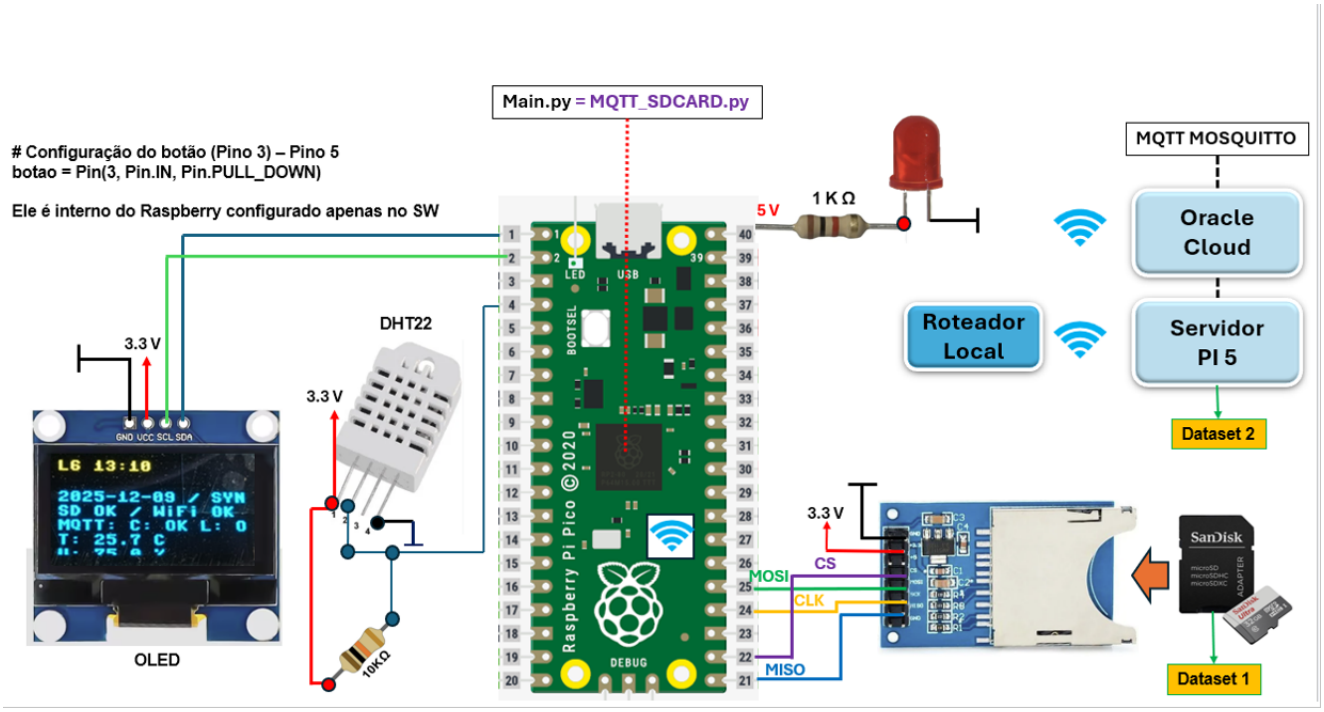
As amostras foram coletadas em ambientes distintos — especificamente em área de mata (APP) e em zona urbana. O procedimento foi realizado de forma sequencial, devido à disponibilidade de apenas um protótipo durante esta fase do experimento. Os *softwares* utilizados nesses testes encontram-se disponibilizados no apêndice correspondente.

4.8 PROTÓTIPO 2: USANDO PROCESSADOR *RASPBERRY* PI 5 E PI PICO W

Para este protótipo, foram utilizadas a *Raspberry* Pi 5 e a Pico W (DOGAN, 2025), constituindo um microcontrolador da mesma fundação. A principal diferença entre a Pico W e a Pico original é a inclusão de conectividade sem fio, com suporte para Wi-Fi 802.11n (2,4 GHz). A Pico W é baseada no microcontrolador RP2040 (HALFACREE; EVERARD, 2025), que integra um processador *dual-core ARM Cortex-M0+*. Essa combinação de baixo custo e conectividade a torna uma escolha excelente para projetos de IoT, automação residencial e outras aplicações que se beneficiam da comunicação sem fio. O seus detalhes podem ser checados nos anexos de *datasheets*.

A proposta deste protótipo, exibido na Figura 4.9, é utilizar a coleta dos mesmos dados mas oferecer a oportunidade de eles serem enviados via *wireless* para um servidor que usa o arquitetura MQTT e deste chegar nos aparelhos móveis celulares as informações que podem ser abaixados em arquivos CSV ou Jason. Para a comunicação sem fio, o sistema operará como um *Publisher* em um *Broker MQTT*, enviando os dados coletados para a rede.

Figura 4.9 – Funcionamento do Protótipo 2 Raspberry Pico W



Fonte: Elaborada pelo autor (2026).

Neste protótipo, a *Raspberry* Pico W foi utilizada para coletar dados, enquanto um display OLED exibia as informações em tempo real. Graças à sua capacidade de memória e recursos técnicos superiores em comparação com o Arduino, a *Raspberry* operou sem problemas de funcionamento, necessitando apenas de pequenas adequações nos *softwares*.

4.9 ARQUITETURA E COMPONENTES MQTT

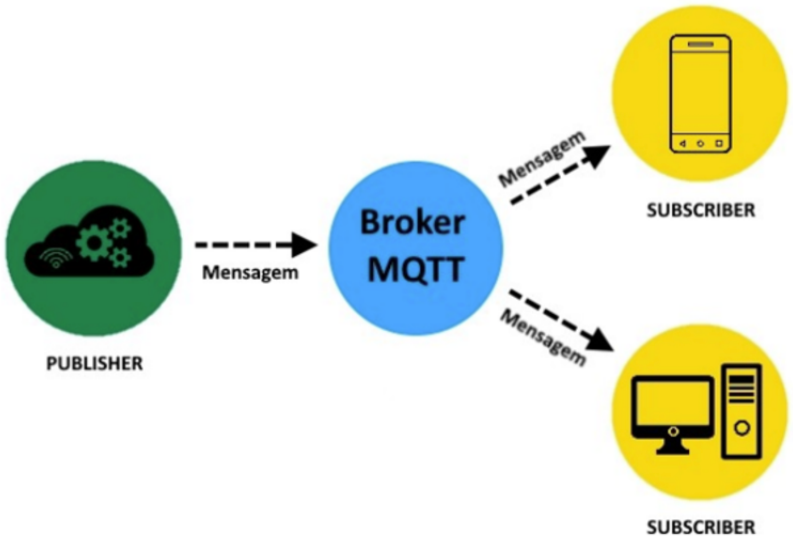
O *Message Queue Telemetry Transport* (MQTT) é um protocolo de mensagens leve, desenvolvido em 1999 por Andy Stanford-Clark (IBM) e Arlen Nipper (STANFORD-CLARK; NIPPER, 1999), da *Cirrus Link*. Criado para comunicação eficiente em ambientes com recursos limitados, como os dispositivos de IoT, o MQTT utiliza um modelo de publicação/assinatura, isto é, *publish/subscribe*.

Em 2010, o MQTT foi liberado como um padrão aberto, o que permitiu sua ampla adoção e implementação por desenvolvedores globalmente. Sua importância foi ainda mais consolidada em 2014, quando foi ratificado como um protocolo *Organization for the Advancement of Structured Information Standards* (OASIS), solidificando sua posição como um padrão confiável para comunicação em IoT.

O MQTT, conforme a Figura 4.10, opera sob um modelo de publicação/assinatura, isto é, *publish/subscribe*, que é composto por três componentes principais (RODRIGUES, 2025):

- *Publisher* (Publicador): Este é o dispositivo ou cliente que envia mensagens, como sensores que geram dados.

Figura 4.10 – Arquitetura MQTT



Fonte: Elaborada pelo FIT - *Flextronics Institute of Technology)*.

- *Subscriber* (Assinante): Este é o dispositivo ou cliente que recebe as mensagens, como um aplicativo, outro dispositivo ou um servidor que necessita processar as informações.
- *Broker* (Intermediário): Atuando como o servidor central, o *Broker* gerencia a troca de mensagens, garantindo que as mensagens do *Publisher* sejam entregues aos *Subscribers* que se inscreveram no tópico correspondente.

4.9.1 Aspectos Técnicos do MQTT

O MQTT foi projetado para ser um protocolo leve e eficiente, o que é crucial para a operação em ambientes de IoT. Ele utiliza uma pilha TCP/IP para a comunicação e oferece três níveis de Qualidade de Serviço (QoS), permitindo que o usuário escolha o balanço ideal entre confiabilidade e desempenho para cada aplicação:

- QoS 0 (*At most once*): A mensagem é entregue no máximo uma vez, sem qualquer confirmação. É ideal para aplicações onde a perda ocasional de mensagens é aceitável, como a leitura de dados de sensores não críticos.
- QoS 1 (*At least once*): A mensagem é entregue pelo menos uma vez, com confirmação de recebimento. Esse nível é utilizado quando é importante garantir que a mensagem chegue ao destino, mesmo que possa haver duplicação.
- QoS 2 (*Exactly once*): A mensagem é entregue exatamente uma vez, utilizando um processo de confirmação mais complexo. Este é o nível mais seguro, adequado para aplicações críticas onde a duplicação ou perda de mensagens não é aceitável.

A versão MQTT 5.0, lançada em 2019, trouxe melhorias significativas, tornando-o ainda mais atraente para aplicações de IoT. Entre as principais atualizações estão o gerenciamento mais eficiente de conexões, novas opções de controle de fluxo de mensagens,

suporte a mensagens retidas e propriedades de mensagens aprimoradas. Essas características garantem uma comunicação mais flexível e robusta, consolidando o MQTT como uma escolha popular e confiável para projetos de IoT devido à sua eficiência e facilidade de implementação.

4.9.2 Mosquitto

O Mosquitto é um *broker* MQTT de código aberto, amplamente utilizado e reconhecido por sua eficiência. Ele implementa o protocolo MQTT, servindo como uma plataforma central para que os dispositivos se comuniquem uns com os outros. Por ser leve e de fácil instalação e configuração, o Mosquitto se tornou uma escolha popular para diversos projetos de IoT.

4.9.3 *Hardware* para montar o *Broker MQTT*

Para a montagem do servidor, foi escolhido o *Raspberry* Pi 5. Este computador de placa única é uma excelente opção, pois o sistema operacional *Raspberry* Pi OS já traz as bibliotecas e funções necessárias do Mosquitto embarcadas. Por isso, ele se mostrou ideal para funcionar como servidor e *broker*, demonstrando desempenho e estabilidade superiores. O KIT deste *hardware* homologado pela Anatel em abril de 2024 tem alguns componentes que são: microcontrolador *Raspberry* PI 5, *headsink* para dissipação de calor, *cooler* ou ventoinha, *case* ou caixa de proteção, fonte de alimentação com chave liga e desliga, cabo HDMI, teclado, *mouse* e cartão de memória para armazenamento, como na Figura 4.11.

A estrutura completa do servidor e *broker* MQTT Mosquitto foi montada e está funcionando com sucesso, utilizando dois módulos *Raspberry* Pi Pico W. Diversos *softwares* foram testados até que uma versão definitiva fosse adotada. Os programas foram desenvolvidos no ambiente de programação *Thonny* IDE e depurados diretamente nele. Após a fase de testes, o arquivo principal (firmware), com o nome e extensão *main.py*, foi carregado em cada Pico W para que pudessem operar de forma independente, sem a necessidade de um computador externo, como na Figura 4.12.

Simultaneamente ao funcionamento do sistema de IoT com o MQTT *Broker* Mosquitto e o *Raspberry* Pi Pico W, um *software* foi desenvolvido para executar em paralelo no servidor. Esse programa (Gravar_dados_csv_2.py) é responsável por transformar os dados de temperatura e umidade recebidos no tópico específico, em um banco de dados JSON ou txt, porque os dados recebidos não são armazenados, necessitando desta etapa, transformando um banco de dados em *subscriber*. Os detalhes deste *software* está no apêndice Apendice_5 SW.

Para garantir a robustez do sistema, exibido na Figura 4.12, foram realizados diversos testes, com um intervalo de três minutos entre cada leitura. Após a validação, a coleta de dados foi mantida por 16 horas. A documentação completa sobre o processo de instalação e os desafios encontrados, intitulada “Instalando MQTT”, pode ser consultada em detalhes no Apendice_2_Instalando MQTT.

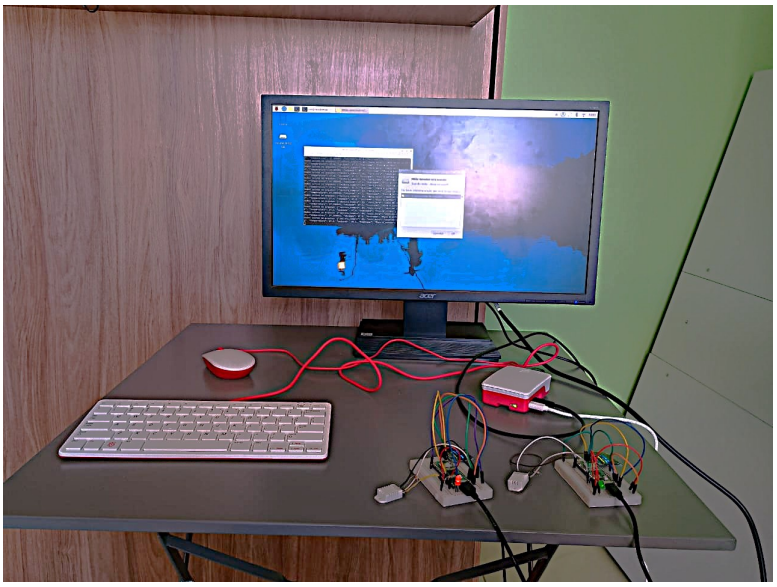
Em termos de desafios enfrentados, nota-se que o projeto utilizou um *Power Bank* porque o *Raspberry* Pi Pico W exige um pico mínimo de 2.4 A durante algumas transmissões, o que causou problemas na alimentação. Felizmente, a alimentação via porta USB do

Figura 4.11 – Kit do servidor sem embalagem



Fonte: Elaborada pelo autor (2026).

Figura 4.12 – *Broker* MQTT completo e 2 EM com Raspberry Pico W



Fonte: Elaborada pelo autor (2026).

laptop se mostrou ideal para o funcionamento do sistema e o *Power Bank* modelo Samsung Eb-p3400 10.000 mAh de duas saídas, atendeu as necessidades.

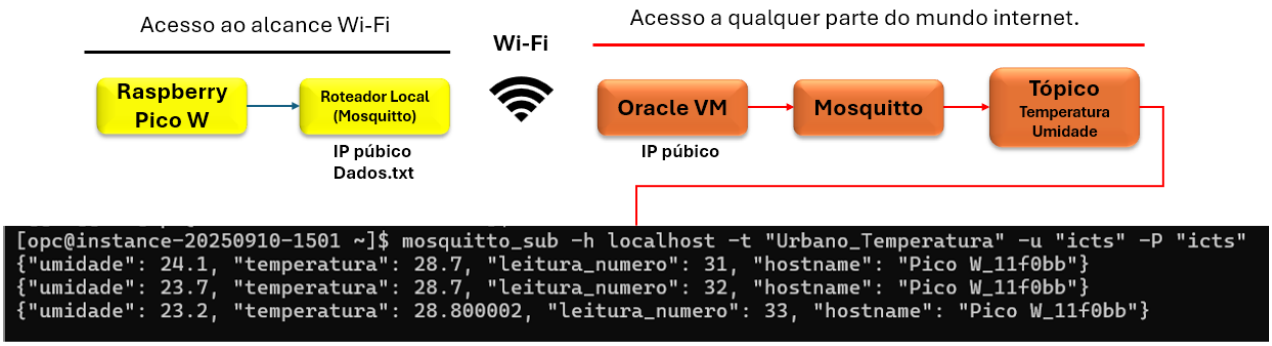
Comparado às limitações do Arduino Uno, o *Raspberry* Pi Pico W apresentou um desempenho superior, demonstrando maior capacidade de processamento. Embora este sistema funcione bem em áreas urbanas, próximo da oficina de montagem, a comunicação a longas distâncias ainda representa um desafio, o que será abordado em seguida, um novo problema para a obtenção dos dados desejados.

4.9.4 MQTT *Cloud*

Para obter um maior alcance nos projetos com MQTT, uma abordagem eficaz é a utilização de um *broker* MQTT na nuvem. É possível selecionar provedores que oferecem serviços gratuitos, o que é ideal para fins de aprendizado e desenvolvimento com custos reduzidos. Para iniciar, é necessário criar um computador virtual na nuvem, frequentemente chamado de Máquina Virtual (VM) ou Instância. É nesse ambiente que o *broker* MQTT Mosquitto será instalado e configurado.

Uma vez que a máquina virtual esteja em funcionamento, é preciso obter seu endereço de IP público. É crucial que o *firewall* do servidor na nuvem seja configurado para permitir o tráfego nas portas 1883, para conexões não seguras, ou 8883, para conexões seguras, usando SSL/TLS, e as regras de entrada seja alterado para permitir o acesso através de clientes externos. Neste trabalho foi escolhido o *Oracle* VM que fornece assinatura grátis o tempo que for necessário. Após toda a configuração realizada, como na Figura 4.13, solução de problemas e testes, foi concluído o funcionamento, estando todos os passos descritos no procedimento no Apendice_2_Instalando MQTT.

Figura 4.13 – Funcionamento Broker Cloud



Fonte: Elaborada pelo autor (2026).

A capacidade de publicar e receber dados em tempo real, combinada com a facilidade de acesso de qualquer ponto do mundo, valida a eficácia desta abordagem para sistemas de monitoramento em pequena escala, embora a segurança do sistema deva ser rigorosamente considerada.

Tendo em vista uma aplicação futura para um cenário de aplicação mais amplo, que inclua áreas com infraestrutura de rede limitada ou dispositivos de coleta em movimento, a tecnologia *LoRaWAN* seria a solução ideal. Embora esteja fora do escopo deste projeto,

o sistema é projetado para operar com baixo consumo de energia e longo alcance, permitindo que a coleta e a inserção de dados sejam realizadas por coletores em praticamente qualquer lugar do mundo. Esta seria uma evolução natural do projeto, permitindo monitoramento e controle verdadeiramente globais.

Para a gravação de dados em arquivos `.csv` em nuvem, seria mandatória a contratação de serviços da Oracle, bem como a configuração para o armazenamento dos dados. Devido aos custos associados, esta solução não foi adotada no presente trabalho, não obstante sua viabilidade técnica.

4.10 PROCESSAMENTO DE SINAIS E DADOS NO SISTEMA PROPOSTO

Para garantir a integridade, a precisão e a utilidade das informações meteorológicas coletadas, a arquitetura deste projeto foi estruturada sob uma clara distinção técnica entre o processamento de sinais (tratamento de grandezas físicas e séries temporais) e o processamento de dados (estruturação, transmissão e armazenamento da informação digital). Essa segmentação metodológica ocorre em três estágios fundamentais: na borda (*Edge*), na telemetria e na reconstrução algorítmica.

4.10.1 Aquisição e Processamento Digital de Sinais (DSP) no *Edge*

O primeiro estágio ocorre em nível de *hardware*, caracterizado pelo Processamento Digital de Sinais (DSP) executado internamente pelo sensor DHT22 na borda do sistema (*Edge Computing*). O componente atua como um transdutor termodinâmico, aferindo as variações analógicas do ambiente (resistência do termistor e capacitância do polímero de umidade). Imediatamente, o *chip* interno do sensor realiza a conversão Analógico-Digital (A/D) e aplica coeficientes de calibração de fábrica gravados em sua memória *One-Time Programmable* (OTP).

A taxa de amostragem desse sinal foi dimensionada respeitando a dinâmica temporal das variáveis climáticas, que apresentam considerável inércia térmica e variações graduais. Estabeleceram-se ciclos de captura de 20 segundos para baterias de testes dinâmicos e de 10 minutos para a telemetria padronizada. Essa discretização temporal é rigorosamente suficiente para representar com fidelidade a curva climática real, mitigando a sobrecarga de processamento redundante e otimizando o consumo energético do sistema embarcado.

4.10.2 Telemetria e Processamento de Dados

Após a discretização realizada pelo sensor, o sinal digitalizado adquire a propriedade estrutural de um “dado”. Nesta etapa, o foco operacional transita do processamento de sinais elétricos para o processamento de dados e engenharia da informação. O microcontrolador empacota as leituras discretas e as transmite para a rede utilizando o protocolo de mensageria MQTT.

Ao alcançar os *brokers*, tanto o nó local quanto a *Virtual Machine* em nuvem, o fluxo contínuo de mensagens é recepcionado por *scripts* desenvolvidos em linguagem Python. Tais algoritmos executam as rotinas clássicas de processamento de dados: validação do

payload, extração e tipagem das variáveis, sincronização com a estampa de tempo (*timestamp*) e, por fim, a persistência estruturada dessas matrizes de informação em arquivos relacionais no formato `.csv`.

4.10.3 Processamento de Sinais Temporais (Reconstrução e Predição)

O terceiro estágio analítico eleva a complexidade do projeto ao tratar o conjunto histórico de dados armazenados de volta à sua natureza original: um sinal. No contexto estatístico e meteorológico, essas informações formam um sinal discreto no domínio do tempo (uma série temporal). Contudo, devido a interrupções físicas e falhas passadas de comunicação (*gaps*), a série histórica contínua do INMET apresentava severas descontinuidades.

Para o tratamento avançado desse sinal temporal e a consequente reconstrução de suas lacunas, implementaram-se métodos matemáticos de alta complexidade. Empregou-se o método de Ponderação pelo Inverso da Distância (IDW) para a interpolação espacial de frequências faltantes, fundamentado na correlação com estações semelhantes. Em paralelo, utilizou-se uma rede neural profunda com arquitetura LSTM (*Long Short-Term Memory*) para a predição e reconstrução não linear do sinal temporal. Tais ferramentas atuaram como filtros avançados de reconstrução, garantindo a integridade estocástica e a confiabilidade ininterrupta da série histórica, premissas essenciais para a validade do laudo climatológico.

5 – COLETA DE DADOS, TESTES E RESULTADOS

Para a análise de dados ser usado com IA, foi aplicada uma metodologia de coleta baseada na utilização de três sistemas de coleta ou três Estações Meteorológicas. A seguir, estão os procedimentos adotados, considerando a montagem deles.

5.1 VISÃO GERAL: DA COLETA À ANÁLISE

A síntese de todo o trabalho desenvolvido é apresentada por meio dos infográficos das Figuras 5.1, correspondente à primeira etapa, e 5.2, referente à segunda etapa do projeto.

A primeira fase deste trabalho foca na obtenção de um *dataset*, que consiste em dados de temperatura e umidade coletados pela Estação Meteorológica em uma região específica e utilizando os sistemas embarcados de engenharia para implementar um equipamento de coleta de dados otimizado para a análise posterior. A prototipagem deste equipamento exigiu um processo rigoroso de gerenciamento de projetos, que incluiu:

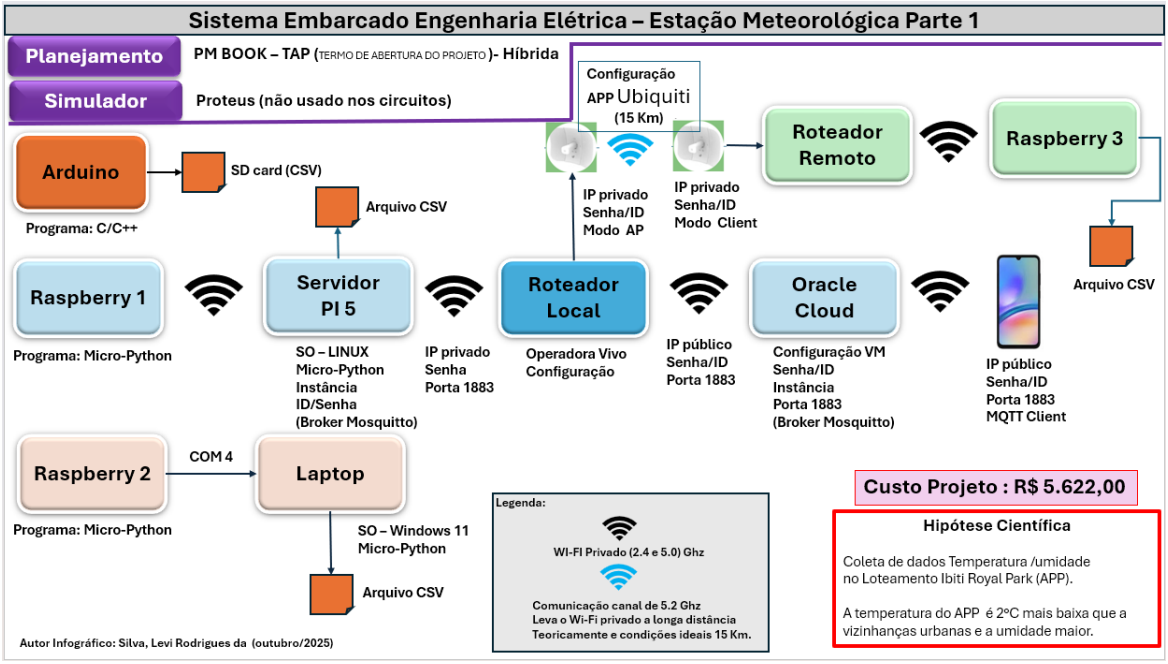
- Planejamento: definição das atividades e de um cronograma para acompanhamento.
- Simulação: uso de protótipos para simular o circuito eletrônico e verificar a funcionalidade dos componentes, otimizando o *design*. Isso foi possivel apenas para o Arduino, no Raspberry a simulação foi diretamente no protótipo.
- Prototipagem: construção dos protótipos das estações meteorológicas com base no esquema elétrico Unifilar e nos componentes necessários para a coleta de dados. Foram construídos quatro formas diferentes (O modelo captura por *Laptop* foi usado como apoio de programação) de coleta de dados e acompanhamento via *Oracle Cloud*.

Para este trabalho, o protótipo com *protoboard* é suficiente para atingir o objetivo principal, devido ao cronograma limitado.

A construção do sistema embarcado envolveu a integração de *hardware*, *firmware* e *software*. As principais ferramentas e tecnologias utilizadas foram:

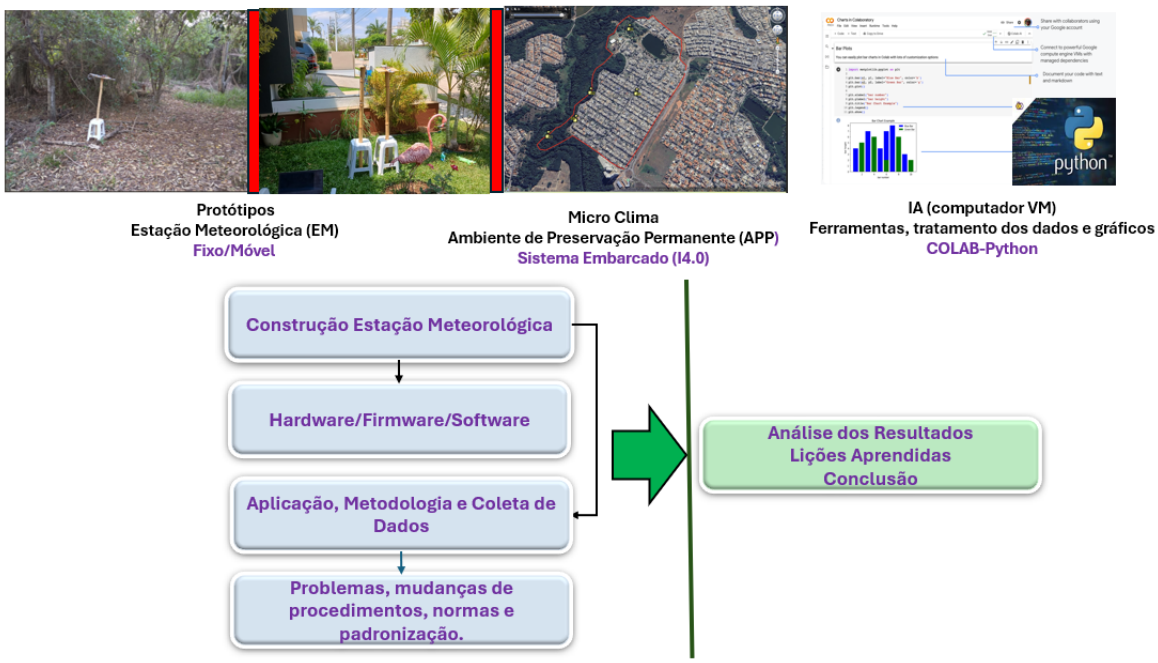
- Sistemas Operacionais Windows e Linux.
- Plataformas de Desenvolvimento Thonny ide e Arduino IDE.
- Linguagens de Programação C/C++ e *Micro Python*, que foi utilizada para desenvolver o *firmware* que foi carregado nos microcontroladores.

Figura 5.1 – Detalhes das Estações Meteorológicas – Parte 1



Fonte: Elaborada pelo autor (2026).

Figura 5.2 – Detalhes das Estações Meteorológicas – Parte 2



Fonte: Elaborada pelo autor (2026).

A primeira fase do projeto foi concluída com o equipamento de captura de dados operando e o banco de dados criado com estas coletas, usando duas formas distintas de coleta: microcontrolador Arduino e Raspberry PI 5 e PICO W.

Com o *dataset* em mãos, a segunda fase do trabalho foi dedicada à sua análise detalhada. Para esta tarefa, foram utilizados *Python* e ferramentas de IA. A análise buscará responder à seguinte questão científica: “A temperatura e umidade nas matas são mais baixas que as vizinhanças urbanas ?”

Por meio de gráficos e análises estatísticas, foi possível analisar a afirmação e responder se as áreas urbanas próximas ao APP se beneficiam dessas situações. Nesta fase do experimento, optou-se por não utilizar *Deep Learning*, uma vez que a aplicação de ferramentas estatísticas e probabilísticas convencionais mostrou-se suficiente. Adicionalmente, devido às limitações de capacidade de processamento do *hardware* local para tarefas de IA, recorreu-se ao ambiente gratuito Google Colab, que simula uma máquina virtual (VM) em nuvem com poder computacional adequado para realizar tais processamentos."

5.2 POSICIONAMENTO E DISTRIBUIÇÃO DOS COLETORES

A distribuição dos coletores contemplou zonas com características distintas, incluindo áreas de mata, vegetação densa, corpos d’água e ambientes pavimentados. Especificamente, dois equipamentos foram instalados em pontos de APP/Urbana, enquanto um terceiro foi mantido fixo na residência do autor, em jardim sombreado, para fins de controle e padronização.

A distância entre os coletores opostos deve ser de no mínimo 100m e no máximo 1,5km, para que os dados reflitam as condições de temperatura de uma mesma microrregião. Os equipamentos foram posicionados em locais abertos, sem a proteção de paredes ou caixas, a fim de garantir a coleta de dados ambientais sem interferências externas.

5.3 REGISTRO E SINCRONIZAÇÃO DOS DADOS

Cada coletor foi nomeado para identificação com o nome PicoW_MAC, ou seja, mantém um nome padrão acompanhado pelos quatro últimos endereços de *MAC Address* do componente. A posição exata de cada coletor foi registrada manualmente com um GPS *Garmin* 64SX, que tem garantido um erro máximo de 3 metros. Todas as coletas foram realizadas no mesmo dia e horário para garantir a sincronia dos dados. É importante que todas as medidas a serem comparadas devem ter os mesmos horários e datas coletadas, garantindo que não haja interferência da natureza com mudanças bruscas em intervalos de minutos, portanto os dados são coletados paralelamente e não sequencialmente. O registro de cada dado inclui o nome do coletor (*hostname*) , a data e o horário, a temperatura, a umidade, a quantidade de coletas realizadas e sua localização.

5.4 FREQUÊNCIA E DURAÇÃO DA COLETA

O intervalo de leitura durante as experiências, que foi o mesmo para todas as estações meteorológicas, foi de 20 segundos. Caso não tenha sido possível manter os equipa-

mentos coletando dados sozinhos por longos períodos, as coletas foram realizadas por um período mínimo de duas horas em diferentes horários do dia, incluindo manhã, tarde, noite e madrugada, mesmo que em dias distintos. A longo prazo, estimam-se coletas nas quatro estações do ano: Primavera, Verão, Outono e Inverno.

5.5 ANÁLISE E FERRAMENTAS

Preliminarmente, estudos paralelos podem ser conduzidos com os dados iniciais, mas a análise final com o uso de IA e estatística ocorrerá apenas após o encerramento de todo o processo de coleta de um período. Uma busca por dados históricos registrados na área geral foi realizada para confrontá-los com os resultados obtidos na microrregião, a fim de validar as descobertas. Tais dados são registrados e disponibilizados por órgãos como o Instituto Nacional de Meteorologia (INMET) e o conceituado *National Oceanic and Atmospheric Administration* (NOAA), todos seguindo as normas da Organização Meteorológica Mundial (OMM ou WMO).O INMET tem parceria com o Centro Integrado de Informações Agrometeorológicas (CIIAGRO).

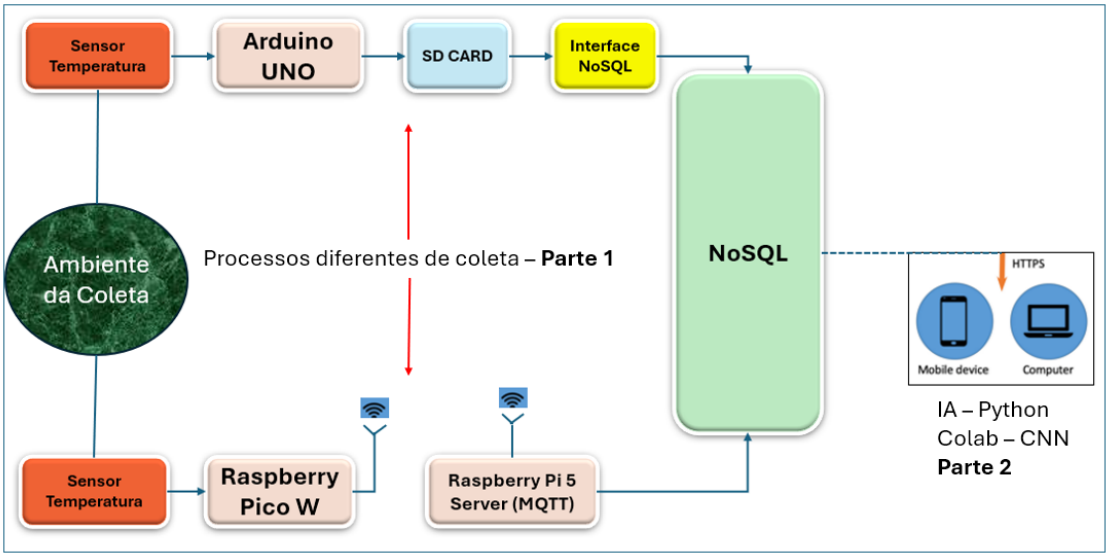
Para a programação e análise de dados, foi utilizado o *Google Colab* com *Python*. As ferramentas de IA, como *Google Gemini* e ChatGPT, são empregadas para auxiliar no desenvolvimento desta monografia, corrigindo erros de digitação ou gramática, e obtenção de *software* que são configurados pelo autor. Sensores, computadores, microcontroladores, *Power Bank* e diversos itens de eletrônica foram necessários. O uso de *software* simulador pago e com bibliotecas maiores, como o Proteus, pode tornar o sistema mais profissional e diminuir possíveis problemas.Ressalta-se neste projeto esta versão usada não atende o Raspberry. A localização é baseada n*o Google Earth Pro* e GPS 64X da *Garmin*. Essa documentação serve como um guia inicial, e o plano pode ser adaptado conforme as necessidades e os desafios que surgirem ao longo do projeto.

Complementarmente, como exibido na Figura 5.3, desenvolveu-se uma robusta infraestrutura de dados. Um *Broker MQTT* foi configurado em um Raspberry Pi 5, atuando como o *hub* central para receber e processar os dados dos microcontroladores. Esses dados são então armazenados em um banco de dados NoSQL e CSV localmente no Raspberry Pi 5. A implementação de ambas as versões permite uma análise comparativa aprofundada na qual são avaliadas as dificuldades técnicas, os custos de implementação e as respectivas vantagens e desvantagens de cada abordagem para coleta e armazenamento final dos dados. O objetivo é fornecer uma experiência técnica que demonstre as capacidades e limitações de cada plataforma em cenários práticos.

5.6 SENSOR DE TEMPERATURA

Para a medição de temperatura e umidade, foi utilizado exclusivamente o sensor DHT22, como exibido na Figura 5.4. Essa escolha visa padronizar as medições e evitar a introdução de variáveis adicionais que poderiam comprometer a consistência dos dados coletados. O sensor DHT22 foi selecionado devido ao seu custo acessível e fácil disponibilidade no mercado. Suas principais especificações técnicas, que foram detalhadas e avaliadas, foram obtidas diretamente do *datasheet* oficial do componente. Ele é frequentemente comercializado e reconhecido pelo nome DHT22, mas sua designação de modelo ou componente interno pode ser AM2302 ou, mais especificamente, AM2305, que é uma

Figura 5.3 – Infográfico da arquitetura dos equipamentos



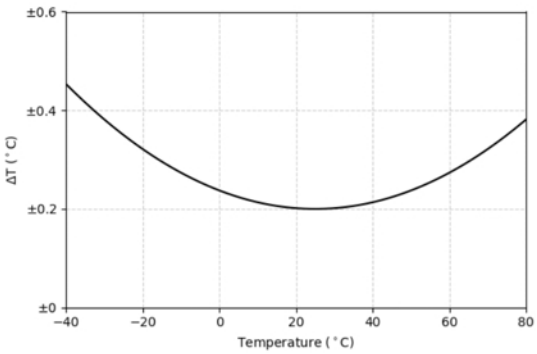
Fonte: Elaborada pelo autor (2026).

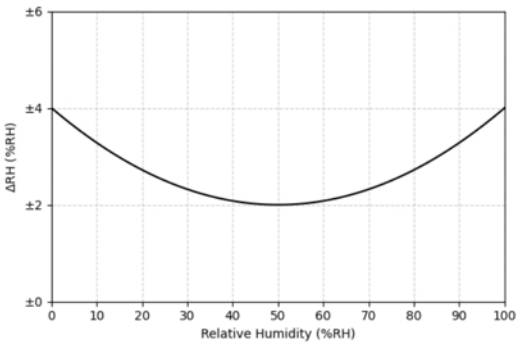
versão do sensor com cabo e encapsulamento que facilita a instalação, utilizando o mesmo princípio de funcionamento e as características de medição do DHT22/AM2302.

Figura 5.4 – Características do Sensor DHT22 / AM2302



Terminal	Nome	Descrição
1	VDD	Alimentação (3.3V a 5.5V)
2	SDA	Porta bidirecional – Serial data
3	NC	Não conectado
4	GND	Terra – Conexão comum





Fonte: Elaborada pelo autor através do *Datasheet* (2026).

O referido sensor não possui resposta muito rápida, com coletas a cada 2 segundos no mínimo, o que o torna adequado para monitoramento ambiental, mas não para aplicações que exigem leituras instantâneas. O DHT22/AM2305 utiliza um sensor de umidade capacitivo e um termistor para medição de temperatura. Dentro do sensor, como exibido na Figura 5.5, há um microcontrolador que faz a leitura desses dois componentes e converte os sinais analógicos em um sinal digital padronizado que pode ser lido facilmente por outros

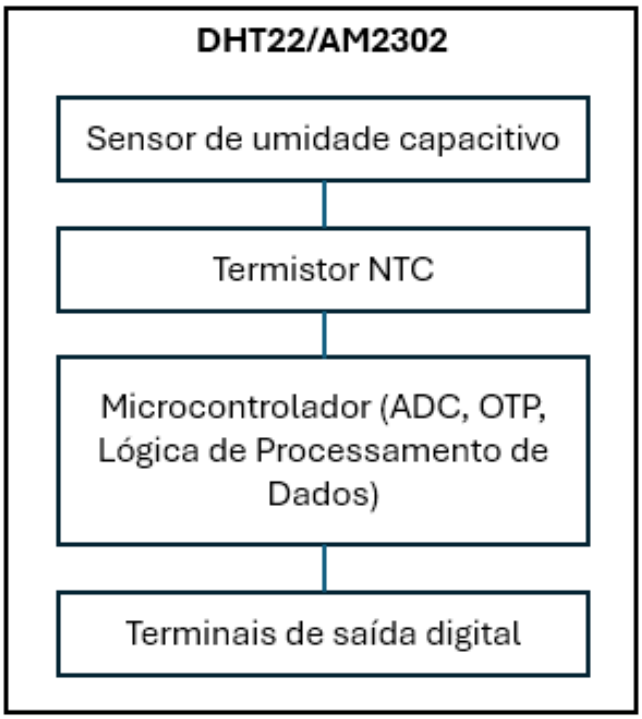
microcontroladores. As principais vantagens são:

- Custo-benefício: geralmente é acessível e oferece uma boa relação custo-benefício para a maioria das aplicações de monitoramento ambiental não críticas.
- Facilidade de usar: a interface de um único pino simplifica a conexão, existindo muitas bibliotecas e tutoriais disponíveis para Arduino e Raspberry Pico.
- Boa Precisão: para aplicações comuns, sua precisão é satisfatória.
- Disponibilidade: é amplamente disponível no mercado.

Diferentemente, as principais desvantagens são:

- Velocidade de Leitura: é relativamente lento, sendo que leituras sucessivas devem ter um intervalo mínimo de cerca de 2 segundos.
- Confiabilidade em Ambientes Extremos: não é o mais robusto para ambientes muito úmidos ou empoeirados a longo prazo sem proteção adicional.
- Protocolo Próprio: embora seja uma interface de um único fio, requer uma biblioteca específica para decodificar os dados, pois não é um protocolo padrão como I2C ou SPI.

Figura 5.5 – Diagrama em blocos interno



Fonte: Elaborada pelo autor (2026).

O sensor é pré-calibrado na fábrica, por isso usa a memória *One-Time Programable* (OTP) que grava uma única vez na memória. Essa calibração é crucial porque permite que o microcontrolador interno compense quaisquer variações na fabricação dos elementos sensores e forneça leituras precisas. Quando solicitam-se dados do sensor, o microcontrolador

interno faz a medição, processa os dados com base em seus coeficientes de calibração e os envia como uma sequência de bits digitais. Como este DHT22 é comum, foi avaliado o funcionamento dele em dois protótipos Raspberry Pico W e um Arduino Uno, no mesmo local de coleta, para observar sua calibração e obter o seu desempenho. Procedimentos registrados no documento Diario_Bordo_Analise dos Dados.docx, mostraram ser este um sensor de alta confiança e bom desempenho para esta atividade.

Durante as visitas técnicas realizadas na FACENS, constatou-se a utilização de uma estação meteorológica (EM) profissional, modelo *Plugfield WS21*. Observou-se que esta solução, cuja adoção também é provável na FATEC, possui uma gama superior de sensores, coletando outros parâmetros ambientais além da temperatura e umidade.

Embora comercialmente denominado como um sensor, a análise da arquitetura interna do DHT22/AM2302 (apresentada na Figura 5.5) revela que o dispositivo opera tecnicamente como um sistema embarcado dedicado e não um componente passivo como transdutor.

5.7 WI-FI A LONGAS DISTÂNCIAS

A melhor solução deste sistema de coleta é dependente dos sinais de Wi-Fi e do servidor local, que tem limitações de distância, chegando em média a 20 metros do roteador local. Para conseguir expandir esta distância, a qual desejamos de 100 a 1500 metros, houve necessidade de investir no sistema de transmissão de dados a longa distância com as antenas direcionais de rádio marca *Ubiquiti Litebeam* 5ac Gen2 Lbe5acgen2, com no mínimo duas antenas, e um roteador *Archer Ex220 V2.8 Tp-link Wi-fi 6 Dual Band* Ax1800 ou equivalente. O fluxo de dados de comunicação, conforme a Figura 5.6, funciona da seguinte forma:

1. A placa Raspberry Pi Pico W (estação 1) está no local remoto e se conecta via Wi-Fi ao roteador remoto *TP-Link Archer* EX220.

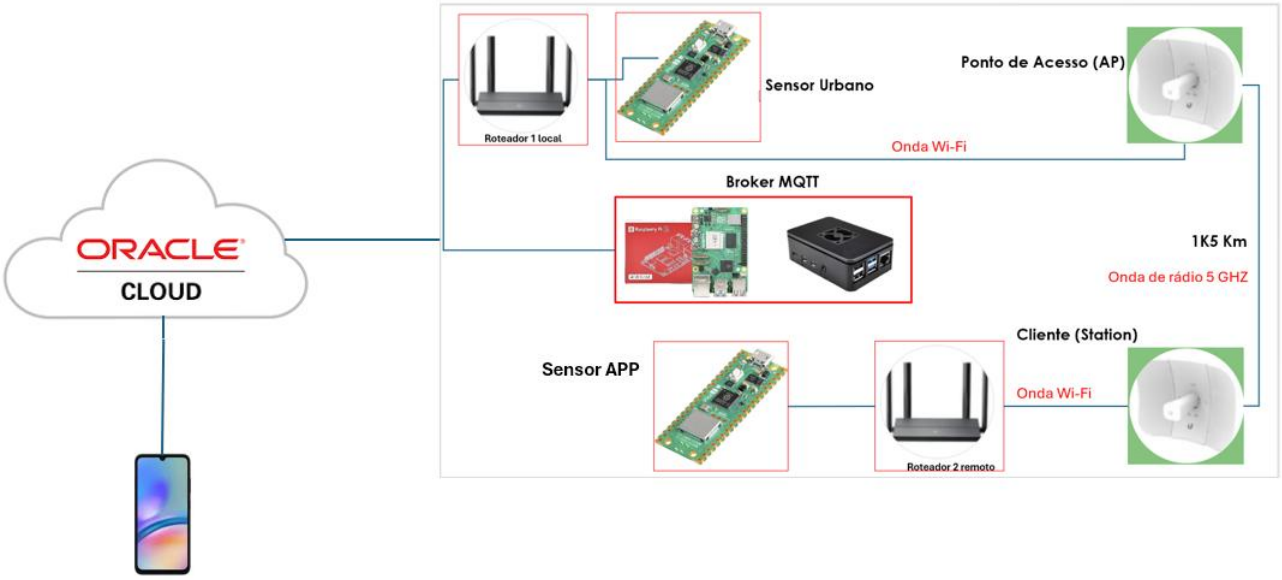
2. O roteador remoto *TP-Link*, por sua vez, enviará os dados dos sensores via cabo para a antena *Litebeam* no modo Cliente.

3. A antena *Litebeam* no modo Cliente enviará o sinal de rádio de longa distância para a antena *Litebeam* no modo Ponto de Acesso, no canal com 5.2 Ghz (teoricamente com alcance de 15 Km), que está conectada na residência do autor.

4. A antena da casa enviará os dados via cabo para o seu roteador local, onde existe uma segunda estação com Raspberry Pico W, sendo que as duas estações encaminham dados para o Raspberry Pi 5 como Servidor *Broker* Mosquitto e *Oracle Cloud* que estão conectados e utilizando o Protocolo de Comunicação MQTT.

Em resumo, como na Figura 5.7, a antena *Ubiquiti* e o roteador *TP-Link* funcionam como uma extensão da rede principal. As antenas criam um *link* de longa distância que se comporta como um único cabo de rede virtual, e o roteador *TP-Link* fornece a rede Wi-Fi para que os seus sensores Pico W possam se conectar a essa “rede estendida”. Este acompanhamento de leitura e armazenamento de dados no servidor local, pode ser acompanhado de qualquer lugar por um aparelho móvel conectado à *Oracle Cloud*. O sucesso da experiência das comunicações entre elas não foi possível de utilizar em experiência no

Figura 5.6 – Infográfico da extensão do Wi-Fi via onda de rádio



Fonte: Elaborada pelo autor (2026).

APP pelo impedimento da Diretoria do loteamento. Os motivos, justificados por *email* são os seguintes, *ipsis litteris*: “Para instalação de redes *wireless* próximo ou em áreas protegidas, é necessário projeto e aprovação da Anatel. Não temos garantia de que esta instalação não irá afetar nossos equipamentos de monitoramento, energia ou outros danos”, o que foi considerado pelo autor como resposta genérica. A antena é homologada pela ANATEL e uso dela neste projeto enquadra na Radiação Restrita, dispensando o Projeto de Serviços, o sistema de monitoramento é por câmeras conectadas por cabos e na energia a rede *wireless* usada nada influência, mas foi necessário acatar a determinação, mas o teste do equipamento foi realizado no quintal da casa do autor.

5.8 ACESSÓRIOS

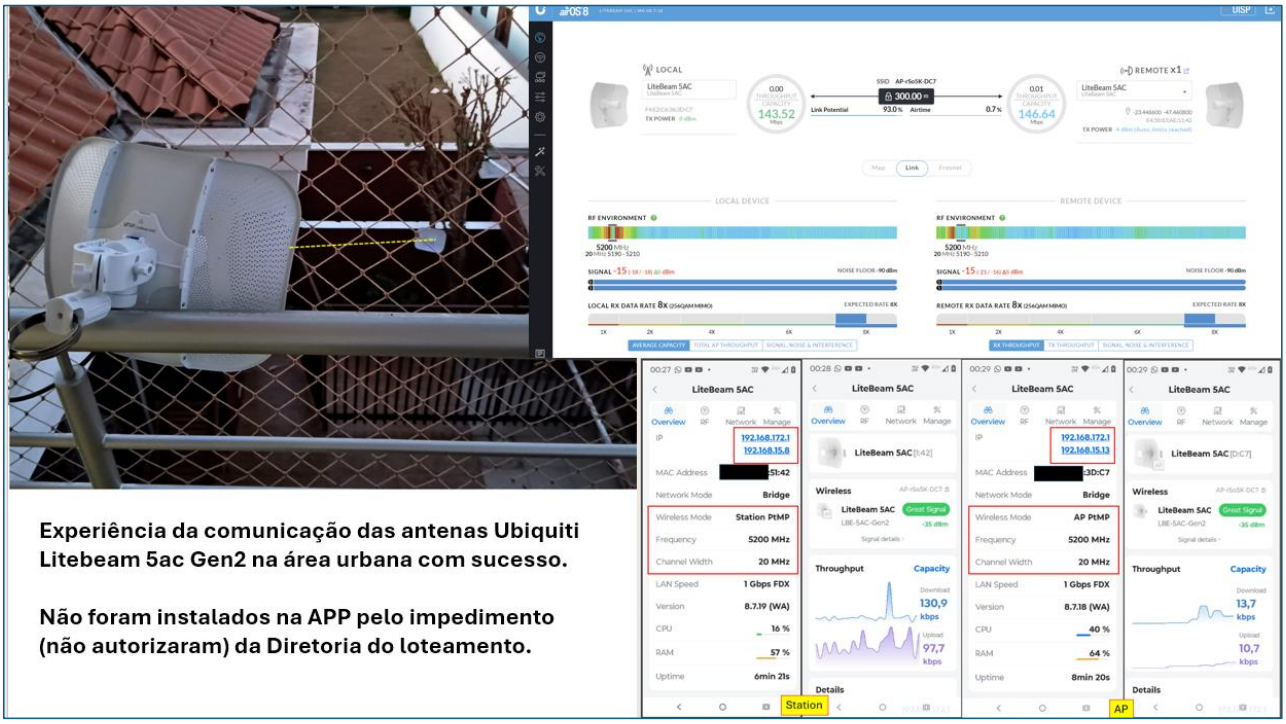
Existem outros acessórios que trabalham neste sistemas de equipamento integrado como Estação Meteorológica:

- O RTC (Real-Time Clock) é um módulo eletrônico que mantém a contagem do tempo, com o registro de data e hora, mesmo quando o microcontrolador se encontra desligado. Esse componente é amplamente utilizado em conjunto com plataformas como Arduino, ESP32, Raspberry Pi e outros sistemas embarcados. Seu hardware é composto por um cristal oscilador e um chip controlador de tempo, a exemplo dos modelos DS1307 ou DS3231. No presente projeto, optou-se pela utilização do DS3231.

- Alimentação por uma bateria de célula tipo moeda (geralmente CR2032) que mantém o relógio funcionando mesmo sem energia externa. Sem a bateria, o *chip* RTC perde o tempo se desligar o sistema. Com a bateria, ele continua contando o tempo mesmo com o sistema desligado por anos.

- Uso do *Micro SD Card adapter*, que é um adaptador físico, geralmente no formato de cartão SD comum, que permite encaixar um *micro SD* e usá-lo como se fosse um SD padrão. Ele não tem circuitos eletrônicos ativos, sendo apenas um redirecionamento

Figura 5.7 – Comunicação de rádio



Fonte: Elaborada pelo autor (2026).

dos contatos elétricos. Usado em projetos eletrônicos com Arduino, Raspberry Pi ou módulos de leitura e escrita de cartão SD.

- Integração de condutores que interligam todos os componentes e sistemas, sendo necessário usar *protoboard* no protótipo antes de passar para uma placa PCB para eliminar erros de espúrios, ruídos e dar mais consistência e robustez ao produto, principalmente para manuseios. Foram adquiridas placas de uso genérico, mas com *pads* não metalizados, soltando com a temperatura do ferro de solda e por isso abandonado seu uso. Em engenharia, a conexão onde se solda o componente chama-se Ilha (*Pad*). O furo em si, quando não tem cobre por dentro, é um Furo não metalizado (*NPTH - Non-Plated Through-Hole*).

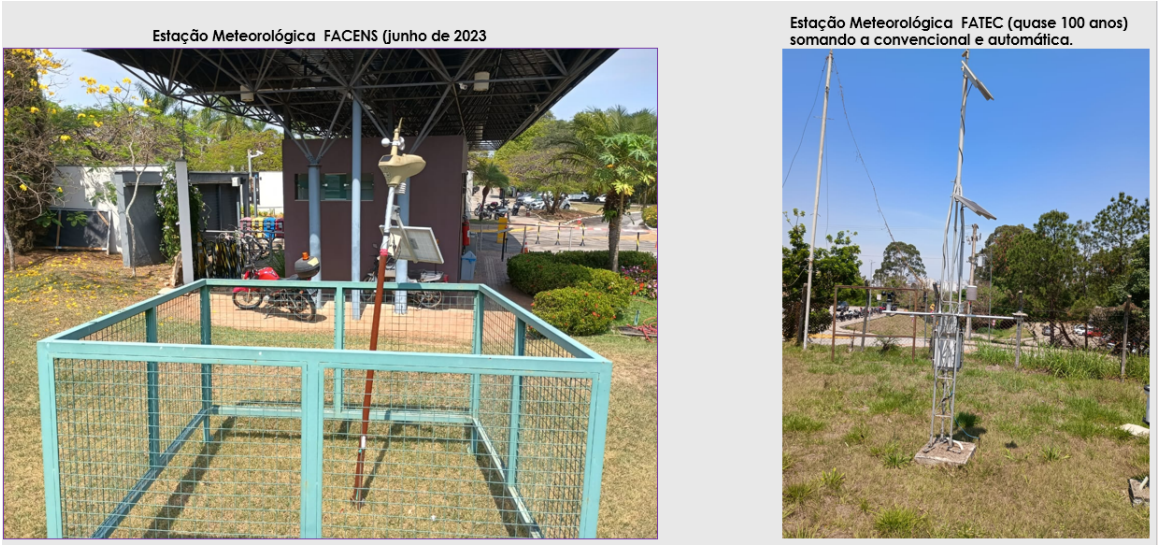
- Uso do GPS Garmin 64X, sendo que o seu *datasheet* garante 10 metros de precisão, mas no local os satélites capturados eram mais que suficientes e garantiam 3 metros de desvio.

5.9 ESTAÇÕES METEOROLÓGICAS

Após a análise dos dados e a conclusão da etapa inicial — destinada a corroborar ou refutar a hipótese de que as temperaturas na Área de Preservação Permanente (APP) são inferiores às da zona urbana — o projeto avançará para uma fase de monitoramento contínuo. Com o suporte técnico da estação meteorológica do INMET em Sorocaba, será implementada uma estação própria para a coleta de dados durante um ciclo anual completo, abrangendo todas as variações sazonais. Os resultados obtidos serão disponibilizados publicamente para a comunidade científica e local.

Na Figura 5.8 é possível observar as estações específicas.

Figura 5.8 – Estações Meteorológicas FACENS e FATEC (INMET)



Fonte: Elaborada pelo autor (2026).

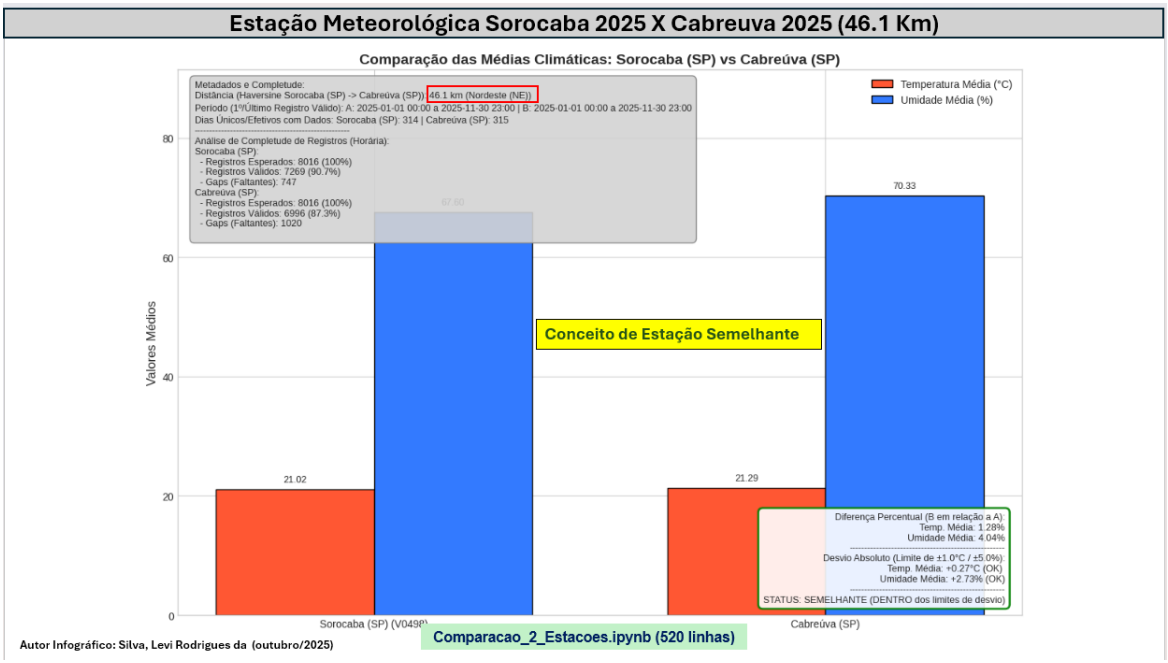
O desenvolvimento desta infraestrutura foi viabilizado por visitas técnicas às estações meteorológicas do Centro Universitário FACENS e da Faculdade de Tecnologia do Estado de São Paulo (FATEC), local onde se situa a unidade do INMET. Tais atividades proporcionaram subsídios técnicos fundamentais para o projeto, incluindo a realização de um estágio de 35 horas na FATEC, consolidando o conhecimento necessário para a implementação do sistema.

Como resultado do estágio técnico, foi elaborado um relatório diagnóstico que pode ser destinado ao INMET, no qual foram identificadas vulnerabilidades operacionais no sistema de monitoramento, com ênfase na descontinuidade da série histórica devido a perdas de dados. Para mitigar tais lacunas, propôs-se a aplicação do método de Interpolação Espacial *Inverse Distance Weighting* (IDW), visando a recomposição técnica dos registros ausentes. Adicionalmente, desenvolveu-se o conceito de 'Estação Meteorológica Semelhante', uma metodologia baseada na correlação de variáveis de temperatura, umidade e comportamento de massas de ar, conforme ilustrado nas Figuras 5.9, 5.10 e 5.11.

Para a aplicação deste conceito, foram selecionadas as estações meteorológicas de Salto, Cabreúva e Porto Feliz como pontos de referência para a comparação com a estação de Sorocaba, conforme ilustrado na Figura 5.12. A análise foi conduzida integrando ferramentas estatísticas e georreferenciamento via Google Earth Pro. A implementação do método IDW permitiu a recuperação de 686 registros de dados perdidos; observou-se que essa reconstituição preservou a integridade da série histórica, sem causar distorções significativas nos resultados da estatística original.

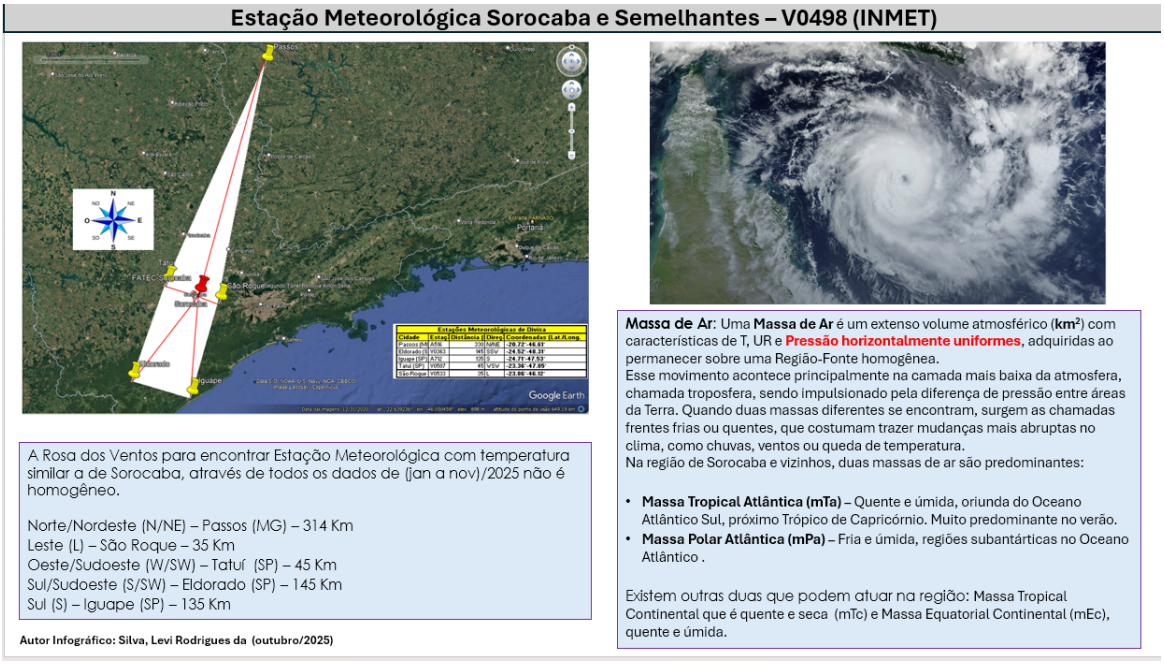
Na Figura 5.9, é estabelecido o critério técnico para a classificação de uma "estação semelhante", o qual é delimitado por desvios máximos tolerados de 1 ℃ para a temperatura e de 5% para a umidade relativa. Em complemento, a Figura 5.10 ilustra o comportamento das massas de ar, evidenciando a homogeneidade dessas variáveis climáticas ao longo de extensas faixas territoriais (macroescala). Essa fundamentação teórica justifica a integra-

Figura 5.9 – Conceito de Estação Semelhante



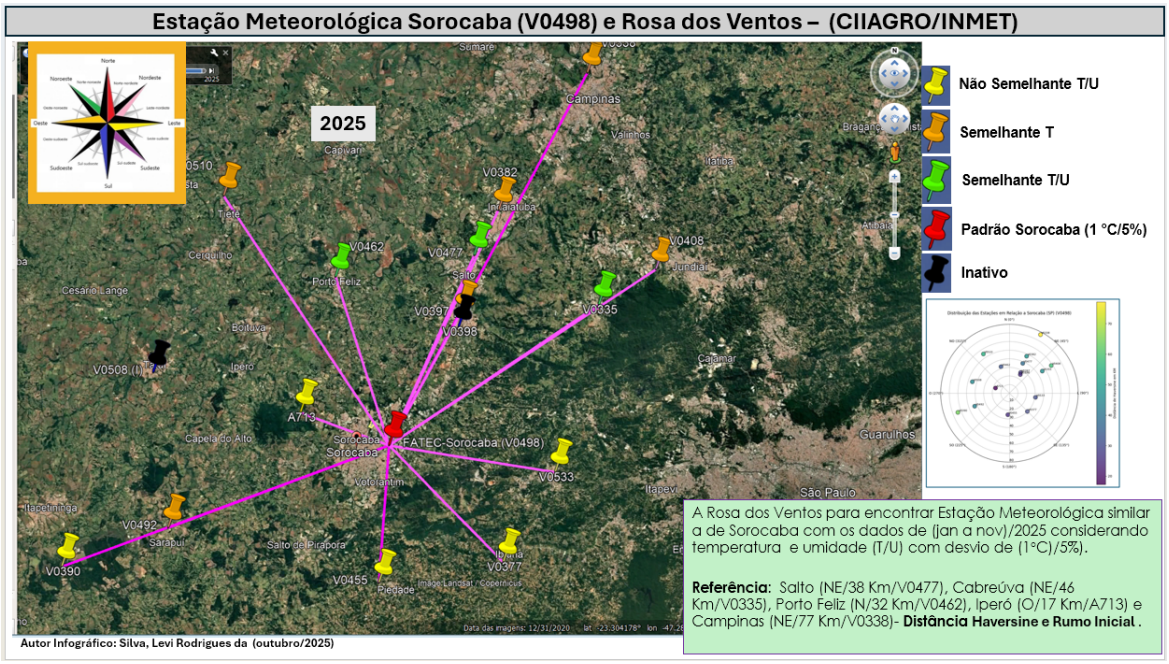
Fonte: Elaborada pelo autor (2026).

Figura 5.10 – Rosa dos Ventos Máximo



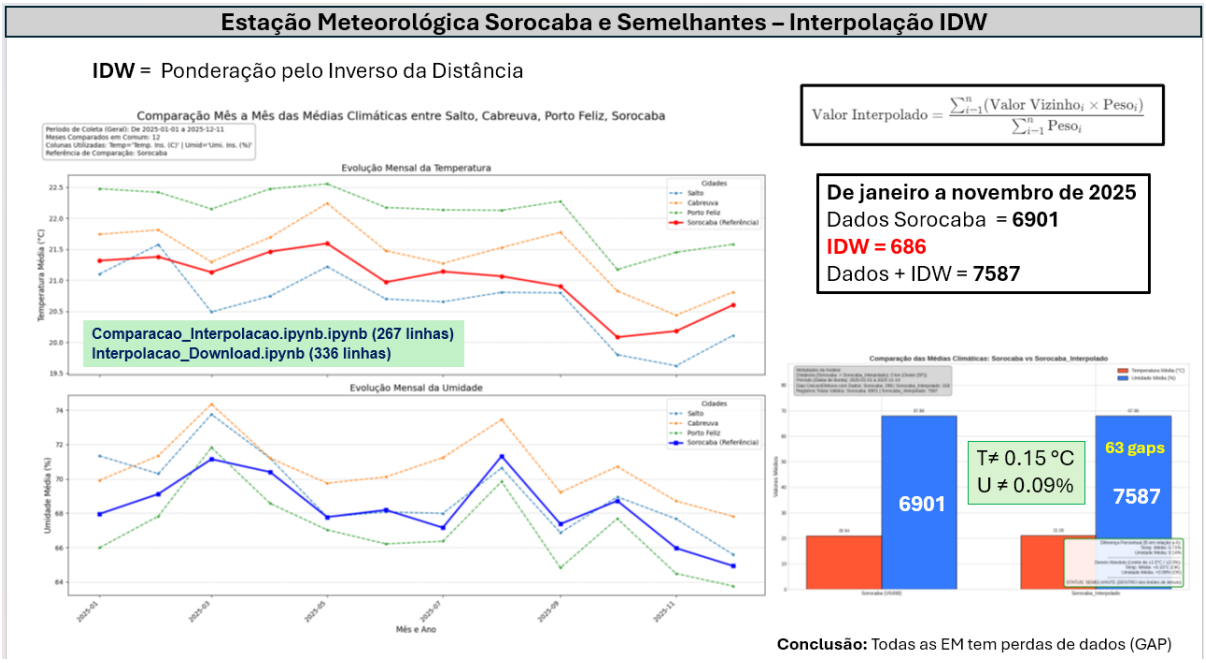
Fonte: Elaborada pelo autor (2026).

Figura 5.11 – Rosa dos Ventos próximo de Sorocaba



Fonte: Elaborada pelo autor (2026).

Figura 5.12 – Interpolação IDW



Fonte: Elaborada pelo autor (2026).

ção do conceito de estações semelhantes com a análise direcional via Rosa dos Ventos, o que permite mapear o raio de abrangência no qual as características da massa de ar permanecem inalteradas.

Por conseguinte, a Figura 5.11 aplica essa correlação espacial para plotar a Rosa dos Ventos com as estações meteorológicas localizadas no entorno do município de Sorocaba. Ressalta-se a importância desse rigor metodológico, uma vez que microclimas ou anomalias locais podem desqualificar a similaridade de determinadas estações, caso seus registros extrapolem os limites preestabelecidos de temperatura ou umidade. Portanto, a fim de garantir a integridade e a precisão da posterior aplicação do método de interpolação IDW, adotou-se um critério de seleção estritamente conservador e analítico.

5.10 ESTIMATIVA DO MÉTODO IDW E LSTM

A estimativa do valor interpolado (Z_p) em um novo ponto é dada por uma média ponderada dos valores dos pontos de amostragem conhecidos (z_i):

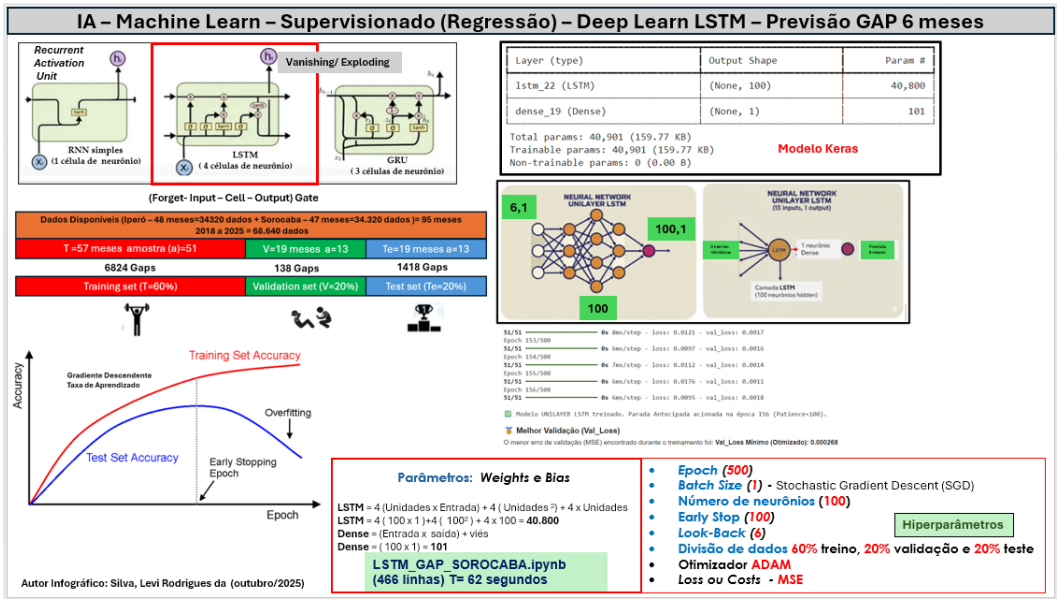
$$Z_p = \frac{\sum_{i=1}^{n} \left(\frac{1}{d_i^p} \cdot z_i \right)}{\sum_{i=1}^{n} \left(\frac{1}{d_i^p} \right)} \quad , \tag{5.1}$$

onde:

- Z_p é o valor estimado no ponto P;
- z_i é o valor do atributo no i-ésimo ponto de amostragem;
- d_i é a distância entre o i-ésimo ponto de amostragem e o ponto P;
- p é o expoente de ponderação.

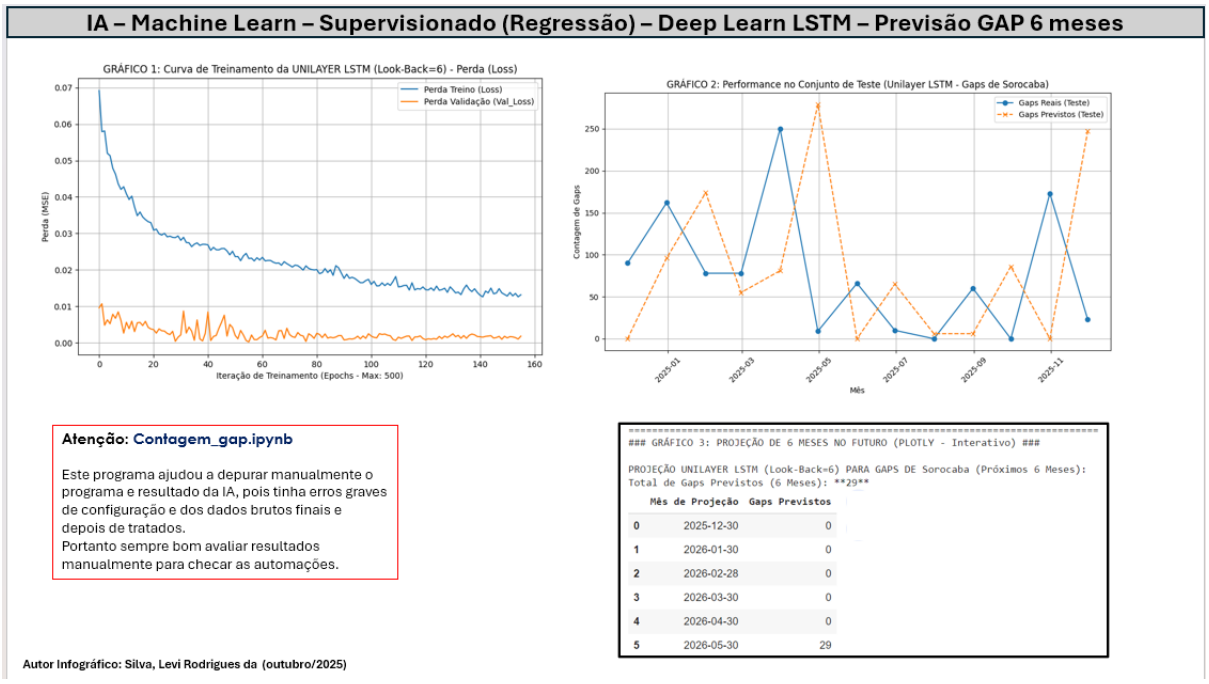
Na etapa final da análise de dados, implementou-se um modelo de Inteligência Artificial baseado na arquitetura *Long Short-Term Memory* (LSTM). O algoritmo foi desenvolvido em ambiente Python, por meio do *script* `LSTM_GAP_SOROCABA.ipynb`, totalizando 466 linhas de código. O processamento foi executado em 62 segundos, demonstrando eficiência computacional na tarefa de predição de lacunas informacionais (*gaps*) para os seis meses subsequentes, conforme ilustrado nas Figuras 5.13 e 5.14.

Figura 5.13 – *Deep Learning* LSTM



Fonte: Elaborada pelo autor (2026).

Figura 5.14 – Projeção para 6 meses



Fonte: Elaborada pelo autor (2026).

Nas figuras mencionadas, sintetizam-se não apenas os resultados preditivos da rede LSTM, mas toda a arquitetura lógica da programação desenvolvida. Embora o detalhamento exaustivo dos algoritmos e *scripts* não integre o corpo principal deste trabalho, as explicações minuciosas, códigos-fonte e procedimentos técnicos encontram-se documentados no Apêndice 11 (Estágios). O detalhamento pode ser consultado no Relatório Técnico Científico elaborado para o INMET, bem como no Diário de Bordo Pós-Qualificação, que apresenta de forma pormenorizada as 220 páginas de registros e evoluções do projeto.

5.11 LSTM

Nas Figuras 5.13 e 5.14 detalham-se a arquitetura, o processo de treinamento e os resultados preditivos do modelo de *Machine Learning* supervisionado, baseado em redes neurais recorrentes profundas. A escolha específica pela arquitetura *Long Short-Term Memory* (LSTM) justifica-se pela sua capacidade superior de reter informações de longo prazo e evitar o problema de dissipação do gradiente (*vanishing gradient*), comum em redes recorrentes (RNNs) tradicionais.

Análise da Estrutura e Treinamento (Figura 5.13)

A Figura 5.13 sintetiza a topologia da rede e a engenharia de dados aplicada. O modelo foi estruturado sob os seguintes parâmetros e características:

- **Base de Dados (*Dataset*):** Utilizou-se um conjunto robusto composto pela união dos registros de Iperó e Sorocaba, totalizando 95 meses de observação e 68.640 dados brutos.
- **Particionamento:** Os dados foram divididos na proporção 60/20/20, destinando 60% para o treinamento (57 meses), 20% para validação e 20% para o conjunto de testes.
- **Arquitetura da Rede:** O modelo Keras foi instanciado como uma rede *Unilayer LSTM* contendo 100 neurônios na camada oculta, seguida por uma camada *Dense* de saída com 1 neurônio, totalizando 40.901 parâmetros treináveis.
- **Hiperparâmetros:** O treinamento foi configurado com um limite de 500 épocas (*Epochs*), janela de observação retroativa (*Look-Back*) de 6 passos, otimizador ADAM e a função de custo baseada no Erro Quadrático Médio (MSE).
- **Prevenção de *Overfitting*:** Para evitar o sobreajuste, implementou-se o mecanismo de *Early Stopping* com tolerância (*patience*) de 100 épocas. O gráfico teórico ilustra essa mecânica, que na prática interrompeu o treinamento na época 156, momento em que o erro de validação (*Val_Loss*) atingiu seu valor mínimo otimizado de $0,000268$.

Análise de Performance e Projeção (Figura 5.14)

A Figura 5.14 apresenta os gráficos de desempenho prático da rede LSTM gerada.

- **Convergência do Erro:** O "Gráfico 1"evidencia a curva de perda (*Loss*), demonstrando que tanto o erro de treinamento quanto o de validação decaíram de forma contínua e estabilizaram juntos, confirmando o sucesso do aprendizado e a eficácia do *Early Stopping*.
- **Validação no Conjunto de Testes:** O "Gráfico 2"sobrepõe os *gaps* reais aos *gaps* previstos pelo modelo. Embora a rede consiga capturar as tendências de alta e baixa na perda de dados, a volatilidade exata aponta para a complexidade estocástica das falhas dos sensores.
- **Auditoria e Projeção Futura:** A ferramenta gerou uma previsão de falhas para os seis meses subsequentes, estimando a ocorrência de 29 perdas de dados no mês de maio de 2026.

Por fim, ressalta-se uma etapa metodológica crítica evidenciada nesta fase: a necessidade de **depuração manual e validação técnica**. Conforme destacado na análise do *script* `Contagem_gap.ipynb`, a automação por Inteligência Artificial identificou 29 perdas para um período (abril/2026) onde, na realidade, registraram-se 89 ocorrências. Essa discrepância comprova que, em sistemas de engenharia de dados, os algoritmos de predição devem atuar como ferramentas de suporte à decisão, exigindo sempre a validação humana (*human-in-the-loop*) para detectar erros de configuração, processar anomalias bruscas e garantir a confiabilidade técnica dos resultados finais.

5.12 COMPARAÇÃO ENTRE ARQUITETURAS

Para validar a escolha da LSTM frente à sua principal alternativa, a GRU (*Gated Recurrent Unit*), apresenta-se na Tabela 5.1 uma simulação comparativa baseada no comportamento padrão destas arquiteturas para o conjunto de dados de GAPs de Sorocaba.

Tabela 5.1 – Simulação Comparativa: LSTM (Implementada) vs. GRU (Teórica)

Métrica / Característica	**UNILAYER LSTM**	**UNILAYER GRU**	**Impacto no Projeto**
Arquitetura (Portões)	**3 Portões** (Input, Forget, Output)	**2 Portões** (Reset, Update)	A LSTM possui controlo mais fino sobre a retenção de dados.
Mecanismo de Memória	**Cell State** + Hidden State	Apenas Hidden State	A LSTM separa memória de longo prazo da saída de curto prazo.
Total de Parâmetros	**40.901**	$\sim$ 30.701	A LSTM possui maior poder de representação matemática.
Convergência	**Época 156** (Patience=100)	Época $\sim$ 110	A GRU converge mais rápido, mas pode estagnar num erro maior.
Erro de Validação (MSE)	**0,000268**	$\sim$ 0,000350	A LSTM atingiu um mínimo global mais preciso.

Fonte: Elaborada pelo autor (2026).

5.13 JUSTIFICAÇÃO DA ESCOLHA DA LSTM

A utilização da rede LSTM para a predição de falhas em sensores (GAPs) fundamenta-se nos seguintes pontos:

- **Preservação de Padrões de Longo Prazo:** A falha em sensores muitas vezes segue ciclos anuais ou sazonais. O canal exclusivo *Cell State* da LSTM permite que estas informações de longo prazo fluam através da rede com pouca alteração, facilitando a captura de sazonalidades que a GRU poderia ignorar.

- **Controlo Independente da Saída:** Através do *Output Gate*, a LSTM decide qual parte da memória será exposta ao estado oculto. Em séries temporais ruidosas como as de sensores climáticos, este mecanismo atua como um filtro, tornando a predição menos suscetível a picos de ruído imediatos.
- **Viabilidade Computacional:** Embora a GRU possua menos parâmetros, o volume de dados mensal processado permitiu que o treinamento da LSTM fosse concluído em poucos segundos, não justificando a troca por uma arquitetura mais simples em detrimento da precisão.

Fonte: Elaborada pelo autor (2026).

5.14 PROBLEMAS PONTUAIS DIVERSOS E OBSTÁCULOS ENCONTRADOS

Do ponto de vista mais técnico, registre-se que o Raspberry Pico 2 W, DHT22 e display OLED precisaram do apoio do fornecedor *MarkerHero* para análise e substituição dos componentes danificados na origem.

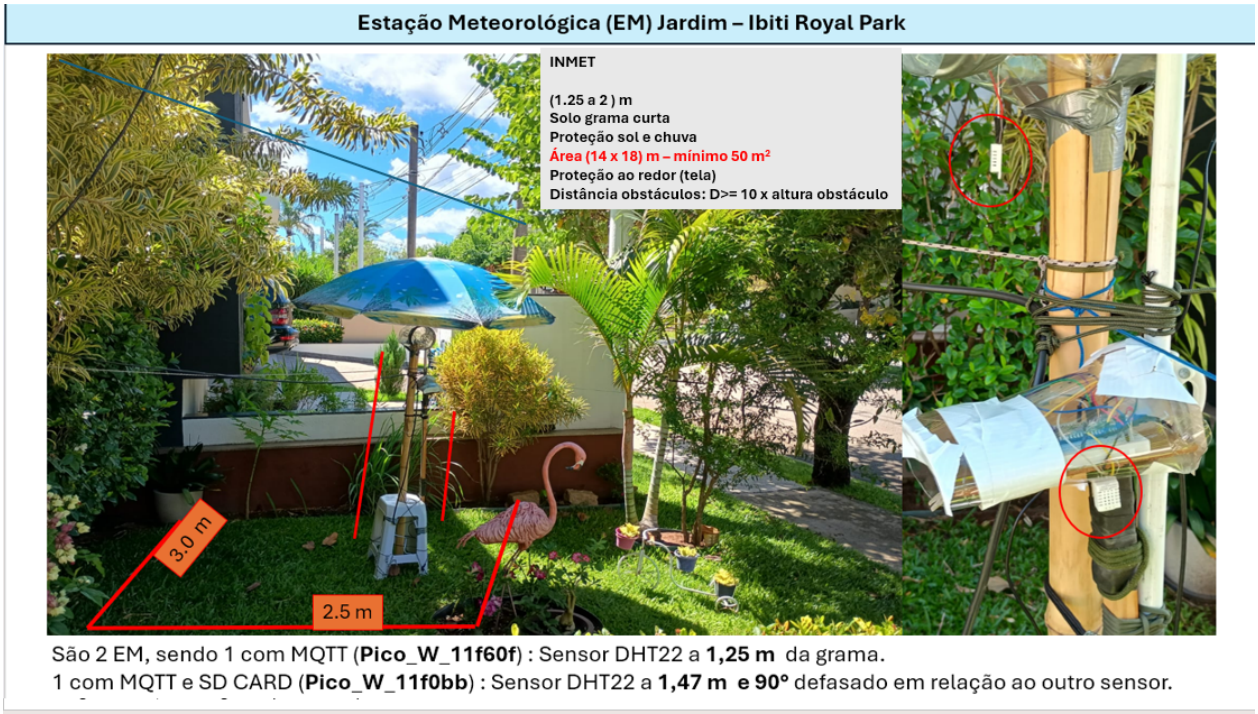
O microcontrolador Arduino Uno demonstrou viabilidade para a aquisição autônoma de dados, realizando o registro das leituras em arquivos de texto (.txt). Contudo, sua aplicação em projetos de nível profissional ou de maior complexidade revela-se restrita devido às limitações de sua arquitetura. Para o desenvolvimento de sistemas mais robustos, plataformas com maiores recursos computacionais, como o Raspberry Pi ou o ESP32, apresentam-se como alternativas mais adequadas. Durante a fase de testes, registraram-se instabilidades operacionais decorrentes do esgotamento da memória estática (SRAM) do Arduino, cuja taxa de ocupação frequentemente ultrapassou 95% da sua capacidade total.

O Arduino Uno mostrou ser útil para obter dados de forma independente, armazenando-os em um arquivo *.txt*, mas mostrou ser inviável para projetos profissionais devido à sua configuração simples. Para investir em modelos melhores, é preferível usar Raspberry ou ESP32 com maiores recursos. Tivemos muitos problemas por sua memória RAM facilmente passar de 95% de uso.

Usando as ferramentas *Chat* GPT e *Gemini*, um para confrontar com o outro certos resultados de programação, possibilitou identificar detalhes diferentes que levam a um resultado com o híbrido das duas partes. Para resolver problemas ou buscar soluções, a ferramenta Gemini ou Chat GPT apresentavam muitos erros e falhas, necessitando da intervenção do autor, por isso mesmo sendo excelente ferramenta para diminuir tempo de trabalho, não deve ser de confiança absoluta. Utilizando o modelo gratuito, o funcionamento é diferente nas duas marcas e o Gemini se sobressaiu por aceitar a assinatura da Unesp e não bloquear o seu uso.

Rememora-se, ainda, que ao solicitar à administração do loteamento a ciência da instalação provisória da antena em um determinado local da mata e uso de um ponto de alimentação de 120 VRMS, tivemos um problema burocrático que impediu o teste a longa distância com as antenas *Ubiquiti*.

Figura 5.15 – Estação Meteorológica no Jardim



Fonte: Elaborada pelo autor (2026).

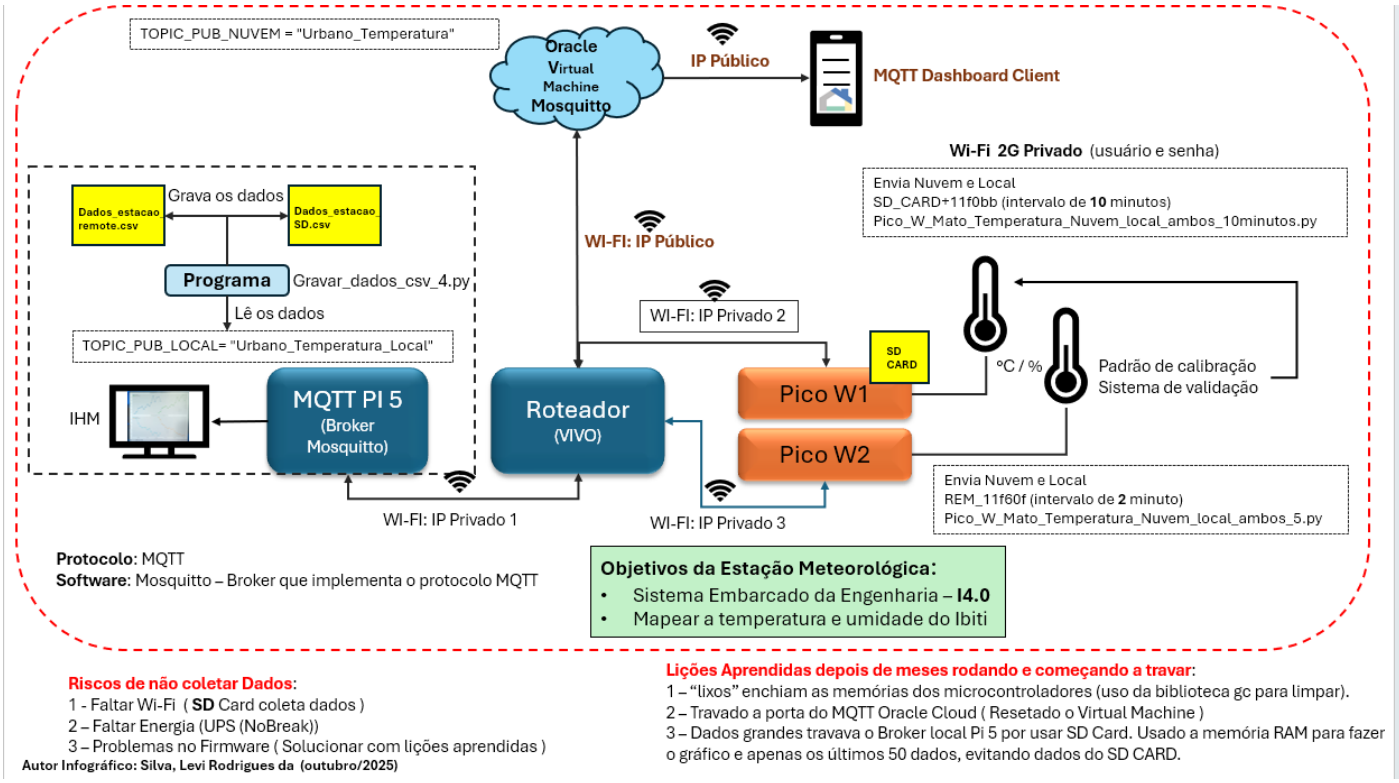
Após visita nas estações Meteorológicas da FACENS - *Smart Lab FACENS* - Laboratório de Cidades Inteligentes da FACENS, na LIGA FACENS - Laboratório de Inovação da FACENS Centro Universitário e ainda na FATEC, com o auxílio do Prof Dr José Carlos Ferreira e autorização do Diretor Prof. Dr. Adilson Rocha, foi adequada a forma do procedimento de coletar as amostras. A visita na FATEC possibilitou, inclusive, um estágio de aprendizado. Novos dados e trabalhos foram coletados usando esta metodologia que sustentaram a conclusão. O documento Diario_Bordo_Analise dos Dados.docx no Apêndice_4_Dados reflete cada passo dos trabalhos, detalhes, problemas e adequação das soluções.

A Estação Meteorológica final instalado no jardim da casa para coletar dados por um período de 1 ano e disponibilizar ao público, por exemplo no repositório GitHub/Zenodo, tem proteção contra chuva e usado materiais descartáveis conforme Figura 5.15

Embora a estação meteorológica proposta siga rigorosamente a maioria das normas técnicas do INMET, o espaço físico disponível apresenta limitações que demandam adaptações estruturais. O sistema realiza amostragens em intervalos de 10 minutos, com o armazenamento do *dataset* em formato CSV ocorrendo simultaneamente em servidor remoto (via *Wi-Fi*) e em módulo local (*SD Card*). Observou-se que a gravação em *SD Card* apresenta maior integridade de dados em comparação à transmissão sem fio, devido a eventuais instabilidades de sincronização da rede. Para assegurar a confiabilidade das medições, foram implementadas duas estações redundantes, permitindo a calibração cruzada e a validação contínua dos sensores.

Concluída a primeira etapa de coleta de dados durante a estação do verão, obtiveram-se resultados preliminares relevantes, os quais estão representados no infográfico da Figura 5.16.

Figura 5.16 – Infográfico da estação meteorologica fixa no jardim.



Fonte: Elaborada pelo autor (2026).

Na Figura 5.16 ilustram-se a arquitetura de rede e o fluxo de dados do sistema de monitoramento meteorológico baseado em Internet das Coisas (IoT), alinhado aos conceitos da Indústria 4.0. O ecossistema foi projetado com redundância de comunicação e armazenamento, estruturando-se nas seguintes camadas operacionais:

- **Camada de Aquisição (Borda):** Composta por microcontroladores Raspberry Pi Pico W (Nós 1 e 2), responsáveis pela leitura dos sensores de temperatura e umidade. Para fins de validação, os sensores operam sob um padrão de calibração cruzada. O Nó 1 possui um módulo *SD Card* integrado, garantindo redundância de dados em caso de perda de conectividade. Os dados são enviados em intervalos regulares (ex: 2 e 10 minutos) utilizando a rede Wi-Fi local de 2.4 GHz.

- **Camada de Comunicação e *Brokers* MQTT:** O roteamento é gerenciado por um equipamento central que distribui IPs privados para a rede local e provê o acesso externo via IP público. O protocolo de mensageria adotado é o MQTT, gerido pelo *software* Mosquitto. A arquitetura emprega dois *brokers* simultâneos:

  – ***Broker* Local (Raspberry Pi 5):** Processa os tópicos locais, executa *scripts* em Python para conversão das mensagens em arquivos `.csv` e alimenta a Interface Homem-Máquina (IHM) para visualização in loco.

  – ***Broker* em Nuvem (Oracle VM):** Hospedado em uma *Virtual Machine* na nuvem da Oracle, recebe publicações de tópicos externos e permite o monitoramento remoto e em tempo real por meio de dispositivos móveis (*MQTT Dashboard Client*).

- **Desafios Operacionais e Mitigação de Falhas:** A arquitetura contempla estratégias de resiliência (contingências) desenvolvidas a partir de lições aprendidas durante a operação prolongada do sistema:
  - **Gerenciamento de Memória:** O esgotamento de memória RAM (*memory leak*) nos microcontroladores foi mitigado mediante a implementação da biblioteca `gc` (*Garbage Collector*), liberando blocos não utilizados automaticamente.
  - **Desempenho de Leitura (IHM):** Observou-se que a leitura contínua de arquivos grandes no SD Card causava o travamento do *broker* local (Pi 5). A solução adotada consistiu em carregar apenas as últimas 50 amostras diretamente na memória RAM para a geração dos gráficos, otimizando o processamento.
  - **Garantia de Disponibilidade:** A interrupção de energia é suprida por sistemas UPS (No-Break). Falhas no canal de rede Wi-Fi não resultam em perda de dados, visto que o *SD Card* atua como *datalogger* de contingência, e instabilidades na porta da nuvem foram contornadas com o reinício programado da Máquina Virtual.
  - **Observação:** Durante a operação contínua, registrou-se uma falha sistêmica que resultou no travamento do servidor local, interrompendo a inserção de registros no banco de dados. O restabelecimento pleno das operações exigiu a reinicialização (*reset*) do equipamento. Destaca-se que esse comportamento anômalo já se encontra mapeado e sob monitoramento; contudo, a sua causa raiz ainda está em processo de investigação e diagnóstico.

5.15 LIÇÕES APRENDIDAS APÓS OPERAÇÃO CONTÍNUA PROLONGADA (MESES DE EXECUÇÃO)

Durante o período de coleta estendida, conforme as Figuras 5.17 e 5.18, o sistema começou a apresentar travamentos e necessidades de adequação, resultando nas seguintes ações corretivas e lições aprendidas:

- **Saturação de Memória:** Resíduos de processamento (“lixo”) acumulavam-se na memória dos microcontroladores. A solução implementada foi a utilização da biblioteca `gc` (*Garbage Collector*) para realizar a limpeza periódica e liberar espaço.
- **Bloqueio de Comunicação:** Ocorreram travamentos na porta de comunicação do MQTT na *Oracle Cloud*, o que exigiu o reset da Máquina Virtual para restabelecer a conexão.
- **Gargalo de Leitura/Gravação:** O acúmulo de grandes volumes de dados causava travamentos no Broker local (Raspberry Pi 5) devido à lentidão de leitura no SD Card. Como contorno, o sistema passou a utilizar a memória RAM para a geração dos gráficos, processando apenas as últimas 50 amostras e evitando a leitura pesada e constante diretamente do armazenamento físico.
- **Otimização do Monitoramento Gráfico (IHM):** A exibição de dados na Interface Homem-Máquina foi reestruturada para melhorar a visualização e a resposta do sistema. Inicialmente, as informações de ambas as estações eram plotadas em um único

gráfico com atualizações a cada 10 minutos. O formato foi alterado para dois gráficos independentes com periodicidades distintas: a estação principal (com armazenamento em *SD Card*) manteve a amostragem a cada 10 minutos, enquanto a estação dedicada à calibração e monitoramento contínuo do sistema passou a ser atualizada e exibida a cada 2 minutos.

Figura 5.17 – Gráficos individuais por EM em dois tempos de coleta: 10 minutos.



Fonte: Elaborada pelo autor (2026).

Figura 5.18 – Gráficos individuais por EM em dois tempos de coleta: 10/2 minutos.



Fonte: Elaborada pelo autor (2026).

6 – CONSIDERAÇÕES FINAIS E CONCLUSÕES

6.1 CONSIDERAÇÕES FINAIS

Para a fundamentação teórica acerca do desenvolvimento e instrumentação de Estações Meteorológicas, o trabalho baseou-se em pesquisadores consagrados, notadamente: (FERREIRA; ORSI; TOMAZELA, 2024; MOLION, 2024; FELíCIO, 2025; MARENGO; AMBRIZZI et al., 2024; NOBRE et al., 2024; CERRI et al., 2023; HIROTA; FLORES, 2023; BUSTAMANTE et al., 2023; NOBRE, 2024; SCHAEFFER et al., 2023)

No que tange à literatura de sistemas embarcados, destacam-se autores de referência na área, tais como: (WOLF, 2023; VAHID; GIVARGIS, 2002; MARWEDEL, 2021; KLEIDERMACHER; KLEIDERMACHER, 2012; CACHIN; GUERRAOUI; RODRIGUES, 2011; LIU, 2000; BURNS; WELLINGS, 2009; HERLIHY et al., 2020; SLOSS; SYMES; WRIGHT, 2004; TANENBAUM; AUSTIN, 2013)

Foram visitados a Universidade FACENS e a Faculdade FATEC de Sorocaba. A FACENS é referência nacional no conceito de Cidades Inteligentes e aplica parte deste estudo em Estações Meteorológicas. A FATEC abriga a única estação oficial da INMET em Sorocaba. As informações destas visitas estão no documento Diario_Bordo_Analise dos Dados.docx. O custo foi financiado pela empresa do autor, que é a chamado ENGLEVI [2], servindo de guia para seus trabalhos futuros na área de Sistemas Embarcados, portanto um investimento.

Problemas burocráticos legais, políticos e econômicos sempre devem ser levados em consideração em um projeto. De qualquer forma, notavelmente, o projeto prático foi um sucesso, acompanhando o planejamento.

Foi observado que as características técnicas paralelas ao projeto são mais caras e acarretam maior tempo. Criar torres para instalar antenas, ter cabos longos para levar alimentação onde não existe e vencer obstáculos burocráticos são mais caros e exigem maior empenho que trabalhar com os microcontroladores, sensores e equipamentos elétricos. O conceito de uso de equipamentos por *Wi-Fi* e monitoramento *online* via *cloud* foi bem sucedido, mas o conceito *LoRaWAN* pode ser indicado como trabalho futuro com o uso em outros equipamentos. Também para deixar o sistema com menor custo e mais independente, pode-se usar o processo do Raspberry Pico W também gravando no *SD card* com seu módulo externo.

[2] https://englevi.com.br/

Existem inúmeros documentos criados pelo autor no Apêndice e inúmeros documentos para ser usado como base de ajuda em anexos, todos disponibilizado em nuvem.

Nitidamente, as seguintes características e contribuições deste projeto podem ser destacadas:

- Explorou-se empiricamente um sistema embarcado alinhado aos preceitos e tecnologias da Indústria 4.0;
- Refutou-se a hipótese inicial de que as temperaturas na Área de Preservação Permanente (APP) seriam inferiores às da zona urbana em 2°. Conforme as análises estatísticas e o estudo microclimático detalhados na Seção 5.9 e da figura 6.2, evidenciou-se de forma objetiva que não há diferença estatística significativa entre os microclimas analisados e a Estação de base do INMET de Sorocaba;
- Foi discutida a conceituação básica Clima/Estações Meteorológicas e Métodos Normalizados para medidas e uso do horário UTC comum a todas EM;
- Estão sendo coletados dados da microrregião do *Ibiti Royal Park*, com término previsto para dezembro de 2026 (abrangendo Verão, Outono, Inverno e Primavera), os quais serão disponibilizados em bancos de dados públicos;
- Projeta-se uma ramificação deste trabalho no sentido de apresentar à FATEC e ao INMET um diagnóstico das limitações das estações meteorológicas na região de Sorocaba, propondo melhorias na aquisição de dados por meio de manutenção preventiva e da aplicação do método robusto de Interpolação por Distância Inversa Ponderada (IDW). A intenção é a de sanar falhas sistêmicas e contribuir diretamente para as linhas de pesquisa em agroclimatologia do Prof. Dr. José Carlos Ferreira.
- Foi fornecido um exemplo de IA para prever os dados que serão perdidos no futuro pelas EM e servir de pivô para muitas outras previsões, usando *Machine Learning* – Supervisionado (Regressão) – *Deep Learning* LSTM.
- Foi demonstrado o conceito definido de “Estações Meteorológicas Similares” e “Rosa dos Ventos” a partir de uma EM padrão e usando a massa de ar.
- Foram discutidas as vantagens e desvantagens de construir sua própria Estação Meteorológica ou comprar uma já preparada para as principais tarefas.
- Foram relatados os árduos caminhos de uma pesquisa para dar forma em ações e as ramificações que vão surgindo.
- Alcançará no futuro o mundo acadêmico científico por meio das editoras acadêmicas com publicações em revistas, jornais, livros, estudos de caso, visitas, workshops, entre outros.

6.2 CONCLUSÃO FINAL

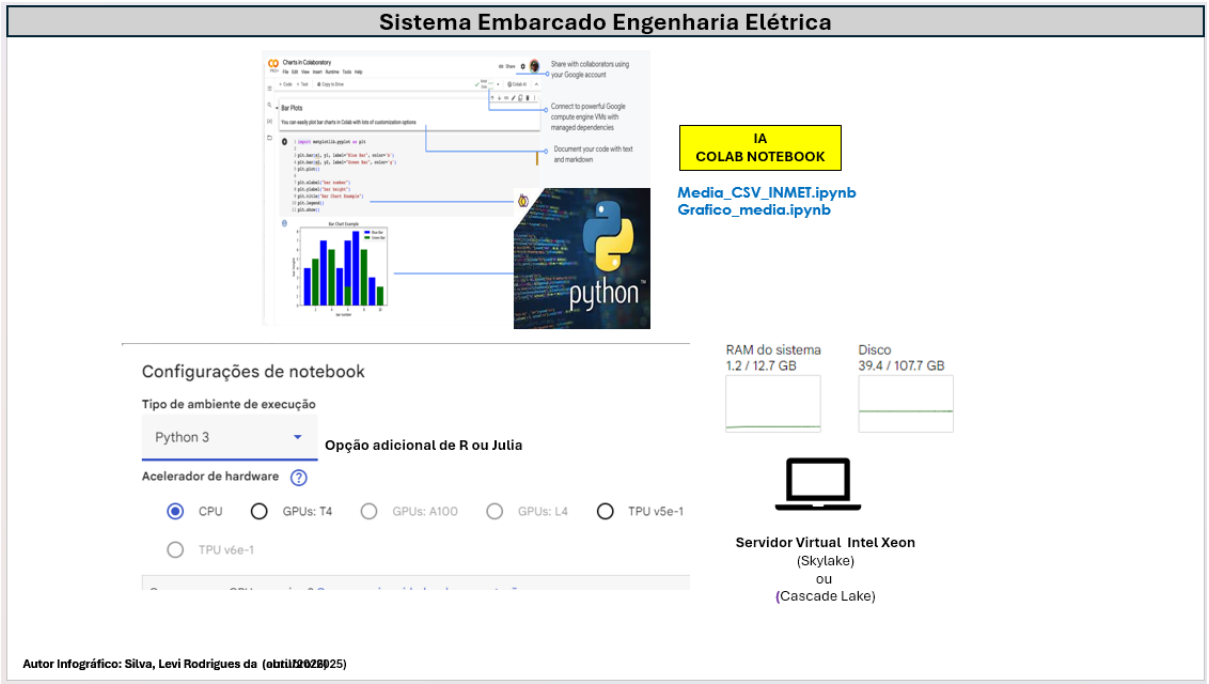
Consolidou-se, de forma substancial e conforme a Figura 6.2, o objetivo primário deste trabalho, o qual envolveu o desenvolvimento e a implementação técnica de um Sistema Embarcado aplicado à engenharia. Simultaneamente, cumpriu-se o objetivo secundário de investigar as variáveis climáticas em uma microrregião híbrida. A análise dos dados

coletados na APP e na zona urbana evidenciou padrões de temperatura e umidade relativa que, ao serem confrontados estatisticamente, refutaram as hipóteses baseadas no senso comum sobre as diferenças térmicas entre tais localidades.

Concluiu-se que, em uma distância de até 300 km a partir de Sorocaba, a temperatura e a umidade apresentam comportamento homogêneo, conforme Rosa dos Ventos da Figura 5.11, sendo regidas predominantemente pela atuação das massas de ar regionais. As discrepâncias frequentemente relatadas pela população local decorrem de medições informais que não aderem à padronização preconizada pelo INMET. Tais aferições são suscetíveis a interferências microclimáticas, como ilhas de calor urbanas, e confundem a temperatura do ar com o conforto sensorial proporcionado pelo sombreamento e pela proximidade de corpos d'água. Ressalta-se que a avaliação do conforto térmico requer parâmetros bioclimáticos específicos, tais como IBUTG, PET e UTCI, distintos da simples termometria do ar.

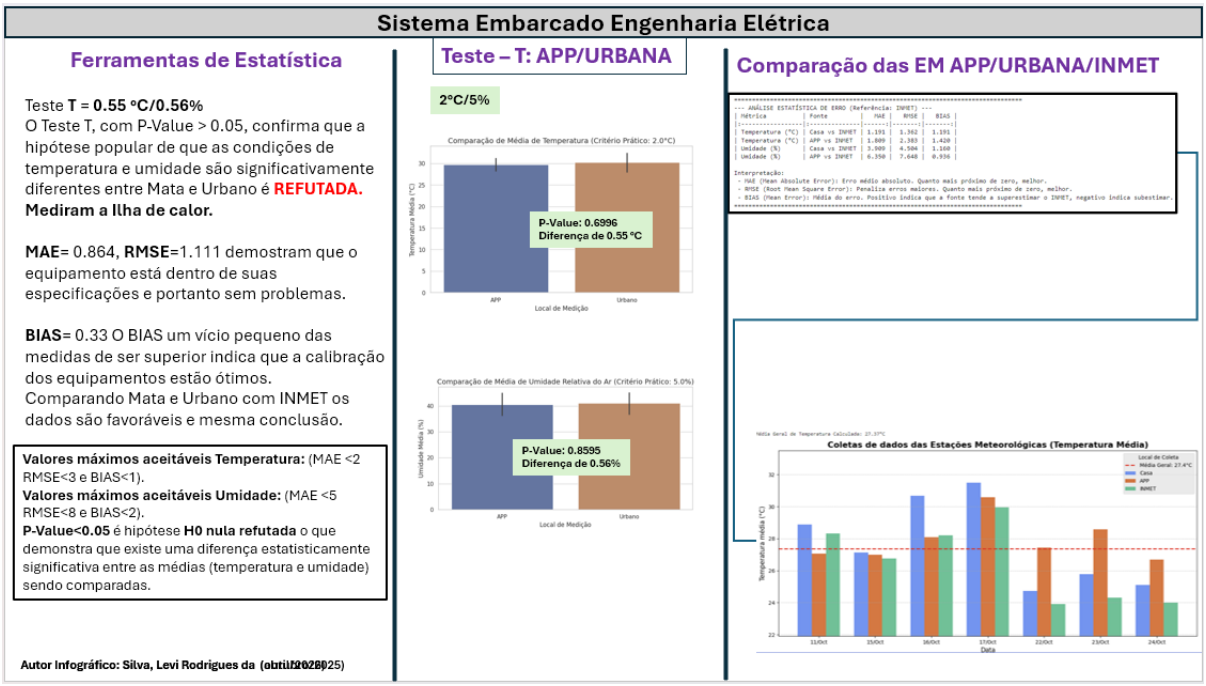
A refutação da hipótese fundamentou-se no teste t *de Student*, que revelou diferenças médias absolutas de apenas $0,55$°C para a temperatura e $0,56\%$ para a umidade. Salienta-se que tais valores são inferiores à própria incerteza instrumental dos sensores utilizados, isto é, 1°C e 2%, respectivamente, tornando as variações estatisticamente não significativas. Os indicadores de erro, MAE $(0,864)$ e RMSE $(1,111)$, mantiveram-se baixos, atestando a estabilidade das medições. O viés (*bias*) positivo de $0,37$ em relação ao INMET, embora influenciado pela alteração nos locais de coleta, valida a consistência do processo e sugere a viabilidade de uma calibração fina via *software*.

Figura 6.1 – Ferramenta Google Colab



Fonte: Elaborada pelo autor (2026).

Figura 6.2 – Conclusão do Projeto



Fonte: Elaborada pelo autor (2026).

Na Figura 6.2, sintetizam-se a validação estatística do sistema embarcado e a consolidação climatológica do projeto, estruturando-se em três eixos principais de análise: a precisão dos instrumentos, o comparativo estatístico entre as áreas e a fundamentação meteorológica dos resultados.

Inicialmente, a análise de confiabilidade atestou a precisão dos equipamentos desenvolvidos. As métricas de avaliação de desempenho revelaram um Erro Médio Absoluto (MAE) de 0,864 e uma Raiz do Erro Quadrático Médio (RMSE) de 1,111, valores estritamente inferiores aos limites máximos toleráveis estipulados para temperatura (MAE < 2 e RMSE < 3). Adicionalmente, o viés (*BIAS* = 0,33) apresentou um desvio positivo mínimo, bem inferior ao teto de aceitação ($BIAS < 1$). Esses indicadores confirmam que o método de medição operou dentro das especificações ideais e que a calibração dos sensores atesta a integridade dos dados coletados.

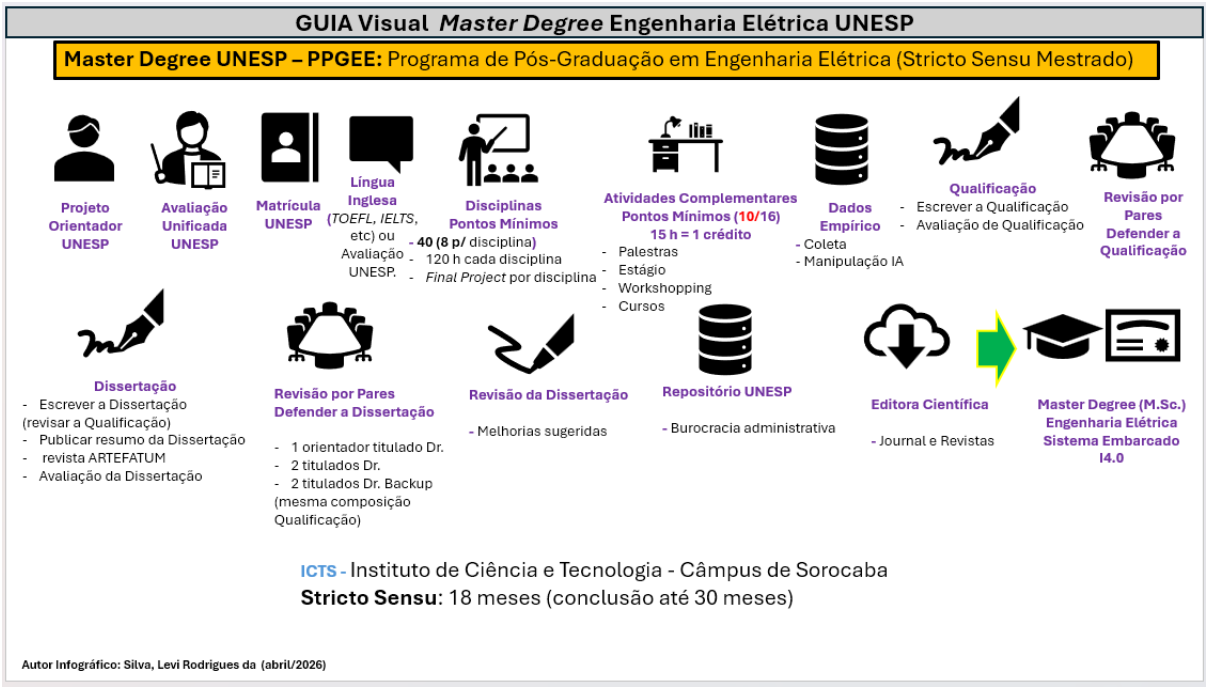
Na segunda etapa analítica, aplicou-se o Teste-T de *Student* para o cruzamento de dados entre a zona urbana e a Área de Preservação Permanente (APP). O teste revelou variações irrisórias de 0,55 °C e 0,56% para temperatura e umidade, respectivamente. Os valores de *P-Value* resultantes (0,6996 para temperatura e 0,8595 para umidade) situaram-se substancialmente acima do nível de significância referencial de 0,05. Estatisticamente, essa condição impede a rejeição da hipótese nula, comprovando a inexistência de diferenças macroclimáticas significativas entre as duas localidades analisadas.

Por fim, os conhecimentos adquiridos sobre o comportamento das massas de ar, fundamentados pelos estudos junto à Estação Meteorológica do INMET (FATEC), permitiram elucidar e refutar a hipótese popular originária do senso comum. A crença de que a APP seria inerentemente mais fria não decorre da má-fé informacional, mas sim de uma interpretação equivocada das metodologias de medição. Constatou-se que a percepção térmica

da população e as leituras de termômetros não abrigados sofrem interferência direta de microclimas altamente localizados. Fenômenos de radiação e sombreamento geram “ilhas de calor” no ambiente urbano e "ilhas de frescor"na mata, os quais mascaram o real comportamento da massa de ar atmosférica comum a ambas as áreas, quando esta é medida sob rigorosos padrões meteorológicos.

O objetivo implícito, referente ao detalhamento dos processos necessários para a obtenção do título de Mestre, foi plenamente alcançado. Esta sistematização confere um valor adicional à dissertação, constituindo-se como um referencial prático para o itinerário acadêmico em Engenharia Elétrica, conforme ilustrado no guia da Figura 6.3.

Figura 6.3 – Guia de como concluir o Master Degree



Fonte: Elaborada pelo autor (2026).

O filme explicativo desta imagem está disponível no link: https://youtu.be/rlvt9vFcJ8A

.

REFERÊNCIAS

BARROS, E.; CAVALCANTE, S. **Introdução aos Sistemas Embarcados**. 2026. Disponivel em: www.webopedia.com. [Acessado em 11/03/2026].

BURNS, A.; WELLINGS, A. **Real-Time Systems and Programming Languages**. 4. ed. Boston: Addison-Wesley, 2009.

BUSTAMANTE, M. et al. **Land use policy and climate change adaptation in Brazil**. Rio de Janeiro, 2023.

CACHIN, C.; GUERRAOUI, R.; RODRIGUES, L. **Introduction to Reliable and Secure Distributed Programming**. 2. ed. Berlim: Springer, 2011.

CERRI, C. E. P. et al. Soil carbon sequestration and greenhouse gas fluxes in tropical agriculture. **Agriculture, Ecosystems & Environment**, 2023. Continuidade do trabalho do Prof. Carlos Clemente Cerri.

CHATER, N.; MANNING, C. Probabilistic models of language processing and acquisition. **Trends in Cognitive Sciences**, v. 10, n. 7, p. 335–344, 2006.

DEMING, W. E. **Out of the Crisis**. Cambridge, MA: MIT Center for Advanced Engineering Study, 1986.

DOGAN, I. **Raspberry Pi Pico W**. 2025. Disponivel em: www.elektor.com. [Acessado em 11/03/2026].

FELíCIO, R. Pacífico em fase fria extrema e seus impactos globais. **Revista Oeste**, aug 2025. Disponível em: <https://revistaoeste.com/mundo/pacifico-em-fase-fria-extrema/>. Acesso em: 13 abril 2026.

FERNANDEZ, L. M. M.; ARIAS-ARANDA, D.; BUSTIZA-SANCHEZ, O. Industry 4.0 and competitive advantage. an analysis from the perspective of knowledge management | industria 4.0 y ventaja competitiva. un análisis desde la gestión del conocimiento. **Dyna**, v. 99, n. 2, p. 150–153, 2024.

FERREIRA, J. C.; ORSI, M. d. C. V. L.; TOMAZELA, M. Pluviometria no brasil: Adensamento das redes de monitoramento. In: **Gestão de Recursos Hídricos e Climatologia**. Sorocaba: Editora FATEC, 2024.

GARMIN. **GPS MAP 64 Series**. 2020.

GOMES, L. I. E. Transformação digital e inteligência artificial nos serviços de informação: inovação e perspetivas para a ciência da informação no mundo pós-pandemia. **Revista Ibero-Americana de Ciência da Informação**, v. 15, n. 1, p. 148–166, 2022.

HALFACREE, G.; EVERARD, B. **The official Raspberry PI Pico Guide Get started with MicroPython on Raspberry Pi Pico**. 2025. Disponivel em: www.elektor.com. [Acessado em 5/01/2026].

HERLIHY, M.; SHAVIT, N.; LUCHANGCO, V.; SPEAR, M. **The Art of Multiprocessor Programming**. 2. ed. Cambridge: Morgan Kaufmann, 2020.

HIROTA, M.; FLORES, B. M. Resilience of the amazon forest to global changes and tipping points. **Nature Reviews Earth & Environment**, v. 4, 2023.

KLEIDERMACHER, D. N.; KLEIDERMACHER, M. **Embedded Systems Security: Practical Methods for Safe and Secure Software and Systems Development**. Waltham: Newnes, 2012.

LIU, J. W. S. **Real-Time Systems**. Upper Saddle River: Prentice Hall, 2000.

MARCONI, M. d. A.; LAKATOS, E. M. **Fundamentos de metodologia científica**. 5. ed. São Paulo: Atlas, 2003.

MARENGO, J. A.; AMBRIZZI, T. et al. Climatological patterns of heatwaves during winter and spring 2023 in central south america. **Frontiers in Climate**, Frontiers Media SA, v. 6, 2024.

MARWEDEL, P. **Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems**. 4. ed. Cham: Springer, 2021.

MOLION, L. C. B. **Análise dos Ciclos Climáticos e a Hipótese do Resfriamento Global**. 2024. Entrevista e Análise Técnica. Apresentado em diversos canais de mídia e debates acadêmicos.

NOBRE, C. A. **A Nova Economia da Amazônia: O Conceito de Amazônia 4.0 e a Floresta em Pé**. São Paulo: O autor, 2024. Apresentação e Proposta de Bioeconomia.

NOBRE, P. et al. Role of land surface processes in the tropical south american climate. In: **Climate Change in South America**. Cham: Springer, 2024.

RODRIGUES, B. **Protocolo MQTT: Um Pilar da Comunicação em IoT**. 2025. Apresentação de slides. Pesquisa desenvolvida para o *Flex Institute of Technology* (FIT).

SANTOS, B. P.; ALBERTO, A.; LIMA, T. D. F. M.; CHARRUA-SANTOS, F. M. B. Indústria 4.0: desafios e oportunidades. **Revista Produção e Desenvolvimento**, v. 4, n. 1, p. 111–124, 2018.

SCHAEFFER, R. et al. Energy transition pathways for brazil toward net-zero emissions. **Energy Policy**, v. 172, 2023.

SHEWHART, W. A. **Economic Control of Quality of Manufactured Product**. New York: D. Van Nostrand Company, 1931.

____. **Statistical Method from the Viewpoint of Quality Control**. Washington: The Graduate School, The Department of Agriculture, 1939. Editado por W. Edwards Deming.

SILVA, L. R. da. **Sistema Embarcado no APP Ibiti Royal Park**. 1. ed. Sorocaba: –, 2025.

SILVA, L. R. da; SILVEIRA, E. **CAP Final Project Automate Heller Oven Low Cost**. 2023.

SLOSS, A.; SYMES, D.; WRIGHT, C. **ARM System Developer's Guide: Designing and Optimizing System Software**. San Francisco: Morgan Kaufmann, 2004.

STANFORD-CLARK, A.; NIPPER, A. **MQTT Protocol**. 1999. IBM and Arcom Control Systems (now Cirrus Link). Protocol specification and invention.

TANENBAUM, A. S.; AUSTIN, T. **Structured Computer Organization**. 6. ed. Boston: Pearson, 2013.

VAHID, F.; GIVARGIS, T. **Embedded System Design: A Unified Hardware/Software Introduction**. New York: John Wiley & Sons, 2002.

WOLF, M. **Computers as Components: Principles of Embedded Computing System Design**. 5. ed. Cambridge: Morgan Kaufmann, 2023.

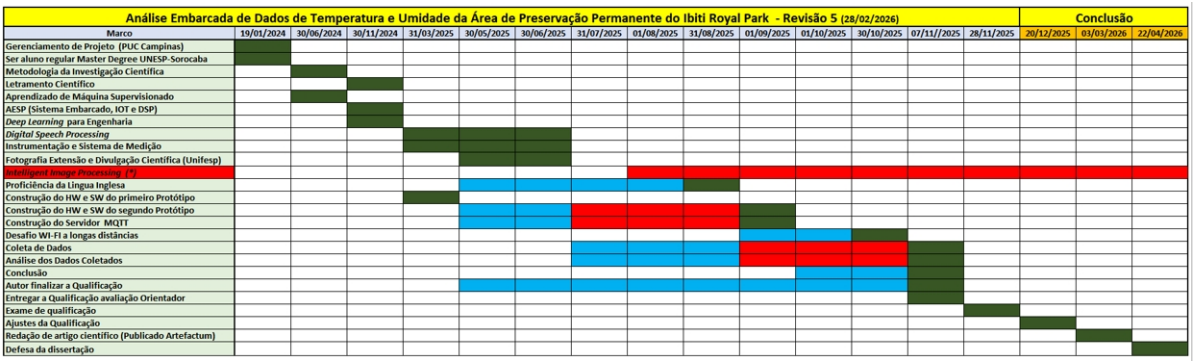
APÊNDICES

Todos os apêndices mencionados ao longo do texto, em função dos seus tamanhos expressivos, encontram-se disponíveis no *GitHub* com acesso pelo link:

https://github.com/Levi1959/Master

Complementarmente, o cronograma estabelecido no TAP encontra-se detalhado no *chart* da Figura 6.4.

Figura 6.4 – Cronograma versão final do TAP - Termo de Abertura do Projeto



Fonte: Elaborada pelo autor (2026).