
Desenvolvimento de uma Estação Meteorológica com Arduino, Java e React

Jhonathan Gontijo da Silva



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Monte Carmelo - MG
2024

Jhonathan Gontijo da Silva

**Desenvolvimento de uma Estação Meteorológica
com Arduino, Java e React**

Trabalho de Conclusão de Curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Área de concentração: Sistemas de Informação

Orientador: Prof. Dr. Adriano Mendonça Rocha

Monte Carmelo - MG

2024

Dedico este trabalho à minha esposa, Lorena Cecila de Melo Gontijo, pela paciência, apoio e por estar sempre ao meu lado nos momentos mais desafiadores. Também dedico este trabalho à minha família, pelo amor e suporte ao longo de toda a minha jornada acadêmica.

Agradecimentos

Primeiramente, agradeço a Deus, por me guiar e abençoar em todos os momentos desta jornada, fortalecendo-me diante dos desafios e concedendo-me a sabedoria necessária para seguir em frente.

Agradeço, com todo o meu coração, à minha esposa, Lorena Cecila de Melo Gontijo, por seu apoio constante, paciência e por acreditar em mim em cada passo dessa caminhada.

Aos meus familiares, que sempre me incentivaram e estiveram ao meu lado, oferecendo amor e suporte, minha eterna gratidão.

Ao meu orientador, que com dedicação e comprometimento me orientou, oferecendo conselhos valiosos e direcionamento, agradeço por sua confiança e por compartilhar seu conhecimento.

Aos professores que me acompanharam ao longo da minha trajetória acadêmica, e aos amigos que contribuíram de forma direta ou indireta com minha formação, deixo aqui meu profundo agradecimento por cada ensinamento, apoio e inspiração. Vocês fizeram parte desta conquista.

*“Faça o teu melhor na condição que você tem, enquanto você não tem condições
melhores para fazer melhor ainda!”
(Mario Sergio Cortella)*

Resumo

Este trabalho teve como objetivo contribuir para o monitoramento climático local da cidade de Monte Carmelo através do desenvolvimento de uma estação meteorológica automatizada e acessível. Utilizando a plataforma Arduino, sensores de pressão atmosférica, temperatura, umidade e precipitação foram integrados para coleta de dados. Foi desenvolvida uma página *web* para que os usuários possam visualizar as informações dos sensores por meio de gráficos, com a opção de consultar o histórico dos últimos 15 dias. A metodologia incluiu a validação dos sensores com instrumentos de referência, comparação dos dados com serviços meteorológicos *online* e estações meteorológicas da região. A análise de usabilidade foi realizada por meio de um formulário aplicado a usuários, que forneceram sugestões de melhoria para o sistema. Os resultados indicam que os dados coletados são precisos e confiáveis, e a interface *web* facilita o acesso às informações. Conclui-se que o sistema é uma solução eficaz e replicável para monitoramento meteorológico em regiões com pouca infraestrutura.

Palavras-chave: Arduino, Monitoramento Climático, Estação Meteorológica, Sensores, Usabilidade.

Lista de ilustrações

Figura 1 – Arduino GIGA R1 WiFi	25
Figura 2 – Pinos I2C do Arduino	26
Figura 3 – Sensor BMP280	26
Figura 4 – Circuito BMP280	27
Figura 5 – Sensor SHT20	28
Figura 6 – Circuito SHT20	28
Figura 7 – Pluviômetro	29
Figura 8 – Circuito pluviômetro	30
Figura 9 – Display LCD 16x2	30
Figura 10 – Circuito display LCD	31
Figura 11 – Circuito completo	32
Figura 12 – Função <i>initializeSensors()</i> no Arduino	34
Figura 13 – Função <i>connectWifi()</i> no Arduino	35
Figura 14 – Função <i>loop()</i> no Arduino	36
Figura 15 – Função <i>postRequest()</i> no Arduino	37
Figura 16 – Endpoint para inclusão dos dados meteorológicos	39
Figura 17 – JSON de inclusão dos dados meteorológicos	40
Figura 18 – Registo na <i>collection</i> informacoes-meteorologicas	40
Figura 19 – Registo na <i>collection</i> agrupamento-dados-meteorologicos	41
Figura 20 – Endpoint para consulta dos dados meteorológicos agregados	41
Figura 21 – Diagrama de caso de uso	43
Figura 22 – Interface para temperatura	44
Figura 23 – Interface para precipitação	44
Figura 24 – Diagrama da arquitetura	45
Figura 25 – MAE - Pressão Atmosférica - Estação x Instrumentos	52
Figura 26 – MAE - Pressão Atmosférica - Estação x Sites	52
Figura 27 – MAE - Temperatura - Estação x Instrumentos	53
Figura 28 – MAE - Temperatura - Estação x Sites	54

Figura 29 – MAE - Umidade - Estação x Instrumentos	55
Figura 30 – MAE - Umidade - Estação x Sites	55
Figura 31 – Localização das Estações Meteorológicas	56
Figura 32 – Precipitação das Estações Meteorológicas	57
Figura 33 – Gráfico - Respostas da 1ª Pergunta do formulário de avaliação	58
Figura 34 – Gráfico - Respostas da 2ª Pergunta do formulário de avaliação	58
Figura 35 – Gráfico - Respostas da 3ª Pergunta do formulário de avaliação	59
Figura 36 – Gráfico - Respostas da 4ª Pergunta do formulário de avaliação	59
Figura 37 – Gráfico - Respostas da 5ª Pergunta do formulário de avaliação	60
Figura 38 – Gráfico - Respostas da 6ª Pergunta do formulário de avaliação	60
Figura 39 – Gráfico - Respostas da 7ª Pergunta do formulário de avaliação	61
Figura 40 – Gráfico - Respostas da 8ª Pergunta do formulário de avaliação	61
Figura 41 – Gráfico - Respostas da 9ª Pergunta do formulário de avaliação	62

Lista de tabelas

Tabela 1 – Grandezas e instrumentos	18
Tabela 2 – Especificações Arduino Giga R1 WiFi	19
Tabela 3 – Especificações BMP280	27
Tabela 4 – Especificações SHT20	29
Tabela 5 – Especificações PB10	30
Tabela 6 – Especificações GM1362	47

Lista de siglas

API Application Programming Interface

CinCi Centro de Inteligência em Cultivos Irrigados

COOXUPÉ Cooperativa Regional de Cafeicultores em Guaxupé

CEMADEN Centro Nacional de Monitoramento e Alertas de Desastres Naturais

HTTP Hypertext Transfer Protocol

IoT Internet of Things

I2C Inter-Integrated Circuit

IP Internet Protocol

JSON JavaScript Object Notation

LCD Liquid Crystal Display

MAE Erro Médio Absoluto - *Mean Absolute Error*

REST Representational State Transfer

SDA Serial Data Pin

SCL Serial Clock Pin

TCC Trabalho de Conclusão de Curso

TCLE Termo de Consentimento Livre e Esclarecido

UFU Universidade Federal de Uberlândia

UniEVANGÉLICA Centro Universitário de Anápolis

Wi-Fi Wireless Fidelity

XML Extensible Markup Language

Sumário

1	INTRODUÇÃO	13
1.1	Motivação	14
1.2	Problema	14
1.3	Objetivos	14
1.4	Organização da Monografia	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Conceitos Básicos	16
2.1.1	Meteorologia	16
2.1.2	Observação Meteorológica	16
2.1.3	Estação Meteorológica	17
2.1.4	Instrumentos Meteorológicos	17
2.1.5	Arduino	18
2.1.6	Protocolo HTTP	20
2.1.7	Arquitetura REST	20
2.1.8	Java	20
2.1.9	Spring Framework	21
2.1.10	MongoDB	21
2.1.11	React	21
2.1.12	Internet das Coisas (IoT)	22
2.2	Trabalhos Correlatos	22
3	ARQUITETURA DO SISTEMA	24
3.1	Estação	24
3.1.1	Componentes	24
3.1.2	Funcionamento	32
3.2	Webserver	38
3.2.1	Linguagem e ferramentas	38

3.2.2	Banco de dados	39
3.2.3	Funcionamento	39
3.3	Webclient	42
3.3.1	Linguagem e ferramentas	42
3.3.2	Funcionamento	42
3.3.3	Diagrama de caso de uso	43
3.3.4	Interface do usuário	43
3.4	Diagrama da arquitetura	45
4	MÉTODO PARA A AVALIAÇÃO	46
4.1	Validação com instrumentos auxiliares	46
4.1.1	Sensor de Pressão Atmosférica BMP280	47
4.1.2	Sensor de Temperatura e Umidade SHT20	47
4.1.3	Pluviômetro de Báscula	47
4.2	Validação com <i>sites</i> de informações meteorológicas	49
4.3	Avaliação dos usuários do sistema	49
5	ANÁLISE DOS RESULTADOS	51
5.1	Resultado da Avaliação dos Sensores	51
5.1.1	Sensor de Pressão Atmosférica BMP280	52
5.1.2	Sensor de Temperatura e Umidade SHT20	53
5.1.3	Pluviômetro de Báscula	55
5.2	Resultado da Avaliação do Sistema com Usuário	58
6	CONCLUSÃO	63
6.1	Principais Contribuições	64
6.2	Trabalhos Futuros	64
	REFERÊNCIAS	65
	 APÊNDICES	 69
APÊNDICE A	– TERMO DE CONSENTIMENTO LIVRE E ES-	
	CLARECIDO (TCLE)	70
APÊNDICE B	– AVALIAÇÃO DA INTERFACE E USABILIDADE	
	DA ESTAÇÃO METEOROLÓGICA	72

Introdução

Com as alterações climáticas mundiais cada vez mais evidentes nas últimas décadas, com fenômenos como o Aquecimento Global e o Efeito Estufa, o monitoramento climático local tornou-se cada vez mais relevante (BAPTISTA, 2007). Não somente para a área agrícola ou outras atividades comerciais/industriais que dependem de dados meteorológicos precisos e atualizados, mas também para o dia-a-dia do cidadão comum que muitas das vezes norteia suas decisões diárias com base no clima.

As estações meteorológicas automáticas oferecem uma solução eficiente para coletar, processar e disponibilizar essas informações em tempo real, auxiliando na tomada de decisões em diversos setores. No entanto, em muitas regiões, como Monte Carmelo (MG), a disponibilidade de estações meteorológicas acessíveis e com dados em tempo real é limitada, o que compromete a precisão das informações disponibilizadas para a população local.

A plataforma Arduino, por sua vez, é uma ferramenta de relativo baixo custo e fácil acesso, amplamente utilizada em projetos de automação. Sua flexibilidade e integração com sensores de diferentes tipos permitem que usuários construam soluções personalizadas para atender a necessidades específicas, como a coleta de dados meteorológicos a nível local (ARDUINO, 2024a).

Este trabalho propõe o desenvolvimento de uma estação meteorológica automatizada utilizando a plataforma Arduino, com sensores para medir temperatura, umidade, pressão atmosférica e precipitação. Os dados coletados são enviados para um servidor, onde são armazenados e disponibilizados para visualização em uma interface interativa. O sistema oferece uma alternativa acessível e de fácil implementação para pequenos agricultores, pesquisadores e entusiastas da meteorologia, preenchendo a lacuna de informações meteorológicas em tempo real de fácil acesso.

1.1 Motivação

A motivação para o desenvolvimento deste trabalho está diretamente relacionada à falta de fontes de dados meteorológicos confiáveis e acessíveis em Monte Carmelo e região. Atualmente, muitos usuários recorrem a *sites* que, embora forneçam informações amplas sobre condições climáticas, frequentemente apresentam inconsistências ou atrasos em relação às condições locais. Este cenário motivou a criação de uma solução local que pudesse fornecer dados meteorológicos precisos e em tempo real, melhorando o acesso à informação e potencializando sua utilização em setores como a agricultura e a pesquisa acadêmica.

A experiência prévia do autor com projetos envolvendo manipulação de microcontroladores, bem como linguagens de programação suficientemente necessárias para a confecção de um projeto desta magnitude, também foram fatores motivacionais para a execução deste trabalho.

1.2 Problema

O principal problema abordado por esta pesquisa é a carência de estações meteorológicas acessíveis que ofereçam dados locais em tempo real, especialmente em regiões como Monte Carmelo. A ausência de tais soluções impacta a tomada de decisões, uma vez que os dados meteorológicos disponíveis em *sites* de abrangência nacional muitas vezes não refletem com precisão as condições microclimáticas da região. Assim, torna-se fundamental o desenvolvimento de uma solução que possa preencher essa lacuna e fornecer informações confiáveis e atualizadas.

1.3 Objetivos

Este trabalho tem como objetivo geral contribuir para a melhoria do monitoramento climático local por meio do desenvolvimento de uma estação meteorológica automatizada, acessível, robusta e confiável, que possa fornecer dados em tempo real para regiões carentes de informações precisas.

Os objetivos específicos são:

- ❑ Investigar a precisão dos sensores de temperatura, umidade, pressão atmosférica e precipitação em comparação com instrumentos de referência e estações meteorológicas estabelecidas;
- ❑ Analisar a eficácia do uso da plataforma Arduino e de tecnologias de código aberto na implementação de uma solução de monitoramento meteorológico acessível;

- ❑ Comparar os dados meteorológicos coletados pela estação desenvolvida com informações fornecidas por serviços meteorológicos *online*, avaliando a acurácia dos dados;
- ❑ Avaliar a experiência de uso e a usabilidade do sistema por meio de um formulário aplicado a diferentes perfis de usuários, identificando pontos de melhoria para o sistema e sua interface;
- ❑ Explorar o potencial de replicabilidade da solução em outras localidades, com foco em regiões onde o acesso a dados meteorológicos em tempo real é limitado.

1.4 Organização da Monografia

Este trabalho está organizado da seguinte forma:

- ❑ **Capítulo 2 - Fundamentação Teórica:** Apresenta conceitos de meteorologia, definições de dispositivos eletrônicos utilizados e tecnologias computacionais aplicadas, além de uma revisão de trabalhos correlatos que embasam o projeto.
- ❑ **Capítulo 3 - Arquitetura do Sistema:** Descreve a estação meteorológica, a forma de processamento e a interface de exibição. Detalha acerca da operação dos sensores e discorre sobre a coleta, a transmissão e o armazenamento dos dados. Explica o funcionamento geral do sistema e como foram implementadas as ferramentas utilizadas.
- ❑ **Capítulo 4 - Método para a Avaliação:** Expõe o processo de validação dos sensores, comparando medições com instrumentos de referência. Também aborda a avaliação do sistema pelos usuários por meio de um formulário.
- ❑ **Capítulo 5 - Análise dos Resultados:** Analisa a precisão dos sensores por meio de erros médios absolutos e avalia o *feedback* dos usuários quanto à usabilidade, interface e funcionalidades, destacando possíveis melhorias para o sistema.
- ❑ **Capítulo 6 - Conclusão:** Apresenta as considerações finais e propõe possíveis trabalhos futuros.

Fundamentação Teórica

Este capítulo apresenta os conceitos fundamentais que sustentam o desenvolvimento desta pesquisa, fornecendo a base teórica necessária para a compreensão do sistema proposto. São explorados os princípios da meteorologia e os instrumentos utilizados para monitoramento climático, com foco na integração de tecnologias modernas, como o Arduino e seus sensores associados, ao contexto de estações meteorológicas automatizadas. Além disso, são discutidas tecnologias computacionais como o protocolo *Hypertext Transfer Protocol* (HTTP), a arquitetura *Representational State Transfer* (REST), Java, Spring Framework e MongoDB, que foram essenciais para a construção do webserver e da interface web. Por fim, são analisados trabalhos correlatos, destacando similaridades e diferenças em relação à solução desenvolvida neste projeto.

2.1 Conceitos Básicos

Com o objetivo de subsidiar melhor compreensão e contextualização, nesta seção serão abordados conceitos que fundamentam a aplicação desenvolvida.

2.1.1 Meteorologia

O Instituto Nacional de Meteorologia (INMET) (2024) define meteorologia como a ciência que investiga os processos e fenômenos que ocorrem na atmosfera, abrangendo os aspectos físicos, dinâmicos e químicos, bem como as interações entre esses elementos e a superfície terrestre subjacente.

2.1.2 Observação Meteorológica

Consiste na medição e registro de todos os elementos que são capazes de representar as condições meteorológicas em determinado momento e lugar, por meio do uso de instrumentos adequados e também da observação humana (Instituto Nacional de Meteorologia (INMET), 2024).

2.1.3 Estação Meteorológica

Em termos gerais, uma estação meteorológica é um conjunto de instrumentos ou sensores que recolhem dados para análise do tempo meteorológico. Eles são capazes de registrar a temperatura do ar, velocidade e direção do vento, umidade do ar, radiação solar, chuva, pressão atmosférica, entre outras variáveis. Podem ser classificadas como convencionais ou automáticas (Agrosmart, 2023).

- ❑ **Convencional:** são aquelas que demandam a presença diária do observador para a coleta de dados. Estas se dividem em classes, conforme a composição observada. As Estações de Primeira Classe são aquelas que medem todos os elementos do clima. As Estações de Segunda Classe são aquelas que não realizam as medidas de pressão atmosférica, radiação solar e vento. Estações de Terceira Classe medem somente a temperatura máxima, a mínima e a chuva. Estas últimas também são referidas como termo-pluviométricas (IBAMA and ESBR - Santo Antônio Energia, 2009).
- ❑ **Automática:** são instalações que automaticamente transmitem ou registram observações obtidas de instrumentos de medição. Neste tipo, as medições são convertidas em sinais elétricos através sensores. Através do processamento e transformação, estes sinais são convertidos em dados meteorológicos e, por fim, são transmitidos como informação. Podem ainda ser subdivididas em estações de tempo real ou estações *off-line*, a depender se os registros serão disponibilizados assim que coletados ou ficarão armazenados em dispositivo físico local, respectivamente (AHMAD et al., 2017).

Os dados obtidos através das estações supracitadas podem ter destinação ampla e diversa. O Instituto Nacional de Meteorologia (INMET) (2024) lista alguns exemplos de finalidade de uso das informações coletadas:

- ❑ Agricultura: garantia de uma boa colheita;
- ❑ Marinha: proteção dos marinheiros, dos navios e dos passageiros;
- ❑ Aeronáutica: proteção e segurança dos pilotos, das aeronaves e dos passageiros;
- ❑ Pesca: condições favoráveis e seguras à pesca;
- ❑ Turismo: garantia de um passeio e de uma viagem feliz e tranquila.

2.1.4 Instrumentos Meteorológicos

Conforme preconizado no Manual de Observações Meteorológicas do Instituto Nacional de Meteorologia (INMET) (1999), os instrumentos/sensores que podem compor uma estação meteorológica medem os seguintes parâmetros:

- ❑ Pressão atmosférica;
- ❑ Temperatura;
- ❑ Umidade;
- ❑ Velocidade do vento;
- ❑ Direção do vento;
- ❑ Precipitação;
- ❑ Radiação.

Este trabalho abordará as medições de pressão atmosférica, temperatura, umidade e precipitação. A Tabela 1 mostra a relação de instrumentos e unidades de medidas para estas grandezas.

Tabela 1 – Grandezas e instrumentos

Grandeza	Instrumento	Unidade Medida
Pressão atmosférica	barômetro	hPa - hectoPascal
Temperatura	termômetro	°C - grau na escala Celsius
Umidade	higrômetro	% - porcentagem
Precipitação	pluviômetro	mm - milímetros

Fonte: Instituto Nacional de Meteorologia (INMET) (1999)

2.1.5 Arduino

Arduino é uma plataforma eletrônica de código aberto baseada em *hardware* e *software* fáceis de usar. As placas Arduino são capazes de ler entradas (luz em um sensor, um dedo em um botão ou uma mensagem de uma rede social) e transformá-las em uma saída (ativando um motor, ligando um LED, publicando algo *online*). Enviando um conjunto de instruções para o microcontrolador da placa é possível estabelecer exatamente o comportamento da mesma. Para tal, é utilizada a linguagem de programação Arduino (baseada em *Wiring* e implementada em C/C++) e o Arduino *Software* (IDE), baseado em *Processing* IDE (ARDUINO, 2024a).

As vantagens do uso desta plataforma listadas na documentação, por Arduino (2024a), são:

- ❑ **Baixo custo:** as placas Arduino são relativamente baratas em comparação com outras plataformas de microcontroladores;

- ❑ **Multiplataforma:** o Arduino *Software* (IDE) roda em sistemas operacionais *Windows*, *Macintosh OSX* e *Linux*;
- ❑ **Ambiente de programação simples e claro:** o Arduino *Software* (IDE) é fácil de usar para iniciantes, mas flexível o suficiente para que usuários avançados também aproveitem;
- ❑ **Software de código aberto e extensível:** o *software* Arduino é publicado como ferramentas de código aberto, disponíveis para extensão por programadores experientes.

2.1.5.1 Modelo Giga R1 WiFi

Dentre as várias plataformas Arduino disponíveis no mercado, a selecionada para este projeto foi a pertencente ao modelo GIGA R1 WiFi. Esta placa possui as especificações demonstradas abaixo na Tabela 2:

Tabela 2 – Especificações Arduino Giga R1 WiFi

Microcontrolador	STM32H747XI dual-core ARM Cortex M7 32-bit @ 480 MHz ARM Cortex M4 32-bit @ 240 MHz
Memória RAM	1 MB
Memória Flash	2 MB
Módulo de Transmissão Sem Fio	Murata 1DX WiFi 802.11b/g/n (2,4 GHz) Bluetooth 5.1
Elemento de Segurança USB	ATECC608A-MAHDA-T USB-C (Porta de Programação / HID) USB-A (Host USB)
Pinos de Entrada e Saída Digitais	76
Pinos de Entrada Analógica	12
Pinos de Saída Analógica	2
Pinos PWM	12
Interface UART	4
Interface I2C	3
Interface SPI	2
Interface CAN	1 (Requer Transceptor Externo)
Conector para Câmera	I2C + D54-D67
Conector para Display	D1N, D0N, D1P, D0P, CKN, CKP + D68-D75
Conector de Áudio	DAC0, DAC1, A7
Tensão de Operação (Nível Lógico)	3,3 V
Tensão de Entrada (VIN)	6-24 V
Corrente CC (pinos E/S)	8 mA
Dimensões	53 mm x 101 mm

Fonte: Arduino (2024a)

Este modelo se destaca entre os demais, e, no momento corrente, é considerado um dos mais avançados da marca, principalmente pelo seu processador poderoso, sua conectividade *wireless* e a diversidade de seus conectores físicos (MakerHero, 2024).

2.1.6 Protocolo HTTP

O HTTP, é um protocolo que reside na camada de aplicação da internet. Se divide em duas partes: o programa cliente e o programa servidor. Ambos são executados em máquinas diferentes, conversam entre si por meio de mensagens que tem sua estrutura definida por este protocolo (LIMA; PETRICA, 2009).

2.1.7 Arquitetura REST

Conforme Junior, Rocha e Maciel (2021), o REST é uma arquitetura para sistemas baseados em redes, criada por Roy Thomas Fielding em sua tese de doutorado apresentado à Universidade da Califórnia. É definido pelo autor como um conjunto de *constraints* (restrições ou princípios) visando:

- ❑ Estabelecer a melhor forma de particionar um sistema;
- ❑ Definir um padrão de identificação na comunicação entre cliente e o servidor;
- ❑ Permitir que os elementos possam evoluir de forma independente.

Outro importante ponto estabelecido por esta arquitetura é o princípio *stateless*, que prega que a comunicação entre cliente e servidor deve ser sem estado. Para que isso se concretize, as mensagens devem conter somente os valores necessários para que a parte compreenda a solicitação. Embora a adoção deste princípio possa acarretar diminuição do desempenho da rede, pelo fato de que a mensagem não será entregue “pronta” para o elemento do sistema, ele garante visibilidade, confiabilidade e escalabilidade. A comunicação *stateless* é geralmente realizada com remessas de arquivos *JavaScript Object Notation* (JSON) ou *Extensible Markup Language* (XML) (JUNIOR; ROCHA; MACIEL, 2021).

2.1.8 Java

O início desta tecnologia se deu em 1991 na empresa Sun Microsystems, onde foi concebida como parte do projeto *Green Project*, liderado por James Gosling. O nome inicial da linguagem foi definido como *Oak*, (em português, “carvalho”), pois era a árvore que se via da janela do local onde os projetistas trabalhavam. O projeto resultou na criação de um controle remoto sensível ao toque chamado *StarSeven*, porém não foi bem recebido na época por estar a frente de seu tempo. Mais tarde, em 1995, foi adaptada para internet, surgindo então a plataforma *Java* (LUCKOW; MELO, 2010).

Dentre as grandes inovações trazidas, uma das mais marcantes é a elencada abaixo:

Uma das principais diferenças entre a plataforma *Java* e as demais linguagens existentes na época é que o *Java* é executado sobre uma *JVM*, ou *Java Virtual Machine*. Qualquer plataforma de *hardware* ou equipamento eletrônico que possa executar uma máquina virtual conseguirá executar *Java*. Isso justifica o slogan “*write once, run anywhere*”, ou, em português, “escreva uma vez, rode em qualquer lugar” (LUCKOW; MELO, 2010, 5.3).

De acordo com Oracle (2024), o *Java* chega à sua versão de número 23. Atualmente consta entre as linguagens mais populares do mundo conforme levantamento realizado pelo canal Stack Overflow (2024).

2.1.9 Spring Framework

O *Spring Framework* foi criado a partir do código contido no livro *Expert One-on-One J2EE Design and Development*, escrito por Rod Johnson. Foi uma ferramenta projetada para trazer mais produtividade ao desenvolver aplicações *Java* para *Web* e ambiente corporativo. Sua implementação não só trouxe mais celeridade no desenvolvimento, como também resolveu gargalos existentes nas versões *JEE* (*Java Enterprise Edition*) da época. Mesmo que novas versões do *JEE* tenham sanado muitos destes problemas, o *Spring* permaneceu muito popular, também por sua simplicidade e oferta de diversos módulos que agregam à aplicação (DEINUM et al., 2021).

2.1.10 MongoDB

O *MongoDB* (nome derivado da palavra em inglês *humongous* que quer dizer “gigantesco”) é um banco de dados “não-relacional”, onde não há conceitos tradicionais como: tabelas, linhas, transações, *joins*, chaves estrangeiras, nem vários outros comuns em bancos de dados relacionais (HOWS; MEMBREY; PLUGGE, 2019).

Ele é um banco orientado a documentos que armazena informações no formato JSON. Os valores armazenados podem ser simples, como números, palavras ou datas, bem como também podem ser listas de informações. Os documentos são agrupados em *collections* e um conjunto de *collections* forma um *database* (BOAGLIO, 2020).

2.1.11 React

Conforme definido pelo Facebook, Inc. (2024), responsável pelo desenvolvimento da tecnologia, o *React* é descrito como uma biblioteca *Javascript* voltada para a criação de interfaces de usuário.

A sua forma de programar declarativa, facilita a construção de telas interativas, auxiliando no desenvolvimento e manutenibilidade do código. A renderização dos elementos é

feita eficientemente, garantindo o desprendimento somente dos recursos necessários (Facebook, Inc., 2024).

O desenvolvimento de interfaces com *React* é baseado em componentes, que encapsulam a própria lógica e estados interdependentemente. A junção destes fragmentos de código monta a interface entregue ao usuário com bom desempenho (Facebook, Inc., 2024).

Atualmente, dentre os pacotes que ofertam bibliotecas e *frameworks* de criação de interfaces com *Javascript*, o *React* é o mais baixado em todo o mundo, conforme o site do *NPM (Node Package Manager)* (npm, Inc., 2024).

2.1.12 Internet das Coisas (IoT)

Internet of Things (IoT) refere-se à rede coletiva de dispositivos conectados e à tecnologia que facilita a comunicação entre os dispositivos e a nuvem, bem como entre os próprios dispositivos. Graças ao advento de *chips* de computador de baixo custo e telecomunicações de alta largura de banda, neste momento existem bilhões de dispositivos conectados à Internet. Isso significa que dispositivos do dia a dia, como escovas de dentes, aspiradores, carros e máquinas, podem usar sensores para coletar dados e responder de forma inteligente aos usuários (Amazon Web Services, 2024).

2.2 Trabalhos Correlatos

Um trabalho correlato relevante foi o desenvolvimento de uma estação meteorológica automatizada de baixo custo apresentado por Silva et al. (2016). O estudo teve como objetivo construir uma estação meteorológica automática utilizando a plataforma Arduino, com sensores de baixo custo e uma interface *web* para acesso remoto aos dados coletados. O projeto alcançou precisão comparável a estações meteorológicas comerciais, oferecendo uma alternativa mais acessível, especialmente para pequenos agricultores e municípios com baixo orçamento. A automatização dos processos de coleta e transmissão de dados foi um diferencial importante, viabilizando o monitoramento remoto de parâmetros como temperatura, umidade e pressão atmosférica, sem necessidade de intervenção humana. São muitas as similaridades entre este projeto e o abordado neste trabalho. Ambos compartilham a proposta de acessibilidade e utilização de uma plataforma Arduino para medição de parâmetros meteorológicos essenciais. A diferença principal reside na estrutura de *software* e na abordagem de armazenamento dos dados. Enquanto aquele utilizou PHP e MySQL para processamento e armazenamento, este empregou Java e MongoDB, oferecendo uma estrutura mais escalável e flexível para armazenar e agregar dados meteorológicos ao longo do tempo. Além disso, o sistema deste trabalho inclui uma rotina automatizada para agrupar dados, facilitando análises históricas, o que não foi abordado por Silva et al. (2016).

Outro trabalho que abordou o tema foi o de Finholdt et al. (2011), que focou no uso de sensores para monitoramento das condições climáticas visando o manejo da irrigação. A estação desenvolvida realizou as medições de parâmetros como precipitação, velocidade do vento, umidade relativa e temperatura, com armazenamento dos dados por até um ano e possibilidade de visualização dos registros anteriores por meio de *display* e comunicação com microcomputadores via protocolo RS-232. O projeto também se destacou por sua autossuficiência energética, através de um painel solar para recarga da bateria. Se assemelha a este TCC em sua ênfase na automação e na coleta de dados sem intervenção humana. A principal diferença, no entanto, é que a estação de Finholdt et al. (2011) foi concebida especificamente para aplicações agrícolas, enquanto a produzida por este TCC tem um propósito mais abrangente, voltado para a coleta e análise de dados meteorológicos em um contexto urbano ou rural, com persistência e exibição em tempo real.

Por fim, o trabalho desenvolvido por Dias e Romano (2018), revela mais um estudo correlato ao desenvolvimento de estações meteorológicas utilizando Arduino. O autor detalha a construção de uma estação relativamente simples empregando essa plataforma. A pesquisa focou na montagem de sensores para medição de temperatura, umidade e pressão atmosférica, validando a eficácia dos dados coletados ao compará-los com os obtidos pela estação meteorológica do Centro Universitário de Anápolis (UniEVANGÉLICA). O projeto destacou-se pela utilização de uma interface simples e pelo baixo custo de implementação, tornando viável a replicação em contextos de recursos limitados. Embora tenham encontrado alguns desafios, como falhas no sistema de energia solar e diferenças nos dados comparados, o uso do Arduino se mostrou promissor para estações meteorológicas acessíveis. Tanto este trabalho, quanto o desenvolvido neste TCC visaram alcançar a construção de uma estação acessível e funcional. No entanto, o projeto desenvolvido por este TCC se mostrou mais robusto no que se refere ao armazenamento e a exibição dos dados. Outra diferença esteve presente na validação dos sensores, onde Dias e Romano (2018) realizaram comparação com uma única estação de referência, enquanto aqui foram utilizados vários instrumentos e serviços para assegurar a precisão da estação.

Arquitetura do sistema

Neste capítulo, será apresentada a arquitetura do sistema da estação meteorológica automatizada, descrevendo os principais componentes e suas interações. O sistema proposto consiste em um conjunto de sensores conectados ao microcontrolador Arduino, que coleta dados meteorológicos como temperatura, umidade, pressão atmosférica e precipitação. Esses dados são transmitidos para uma *Application Programming Interface* (API) por meio de uma comunicação via HTTP, onde são armazenados em um banco de dados e disponibilizados para visualização em uma interface web.

3.1 Estação

Esta seção expõe todos os itens físicos que compõe a estação meteorológica desenvolvida neste Trabalho de Conclusão de Curso (TCC) bem como também o funcionamento e configurações do controlador Arduino.

3.1.1 Componentes

Logo abaixo serão detalhados os componentes de *hardware* com ilustrações e especificações de seus fabricantes.

3.1.1.1 Arduino



Figura 1 – Arduino GIGA R1 WiFi

Fonte: <<https://store.arduino.cc/products/giga-r1-wifi>>

Existem diversos modelos de microcontroladores Arduino e cada um possui uma especificação diferente. São divididos quatro grupos: Família Nano, Família MKR, Família Clássica e Família Mega (ARDUINO, 2024b).

Dentre as opções para o desenvolvimento deste trabalho, o modelo escolhido foi o Arduino GIGA R1 WiFi pertencente a Família Mega, constante na Figura 1. Este controlador possui um módulo *Wireless Fidelity* (Wi-Fi) integrado que atende a necessidade do projeto de conexão com a internet e também possui pinos que obedecem ao protocolo *Inter-Integrated Circuit* (I2C), o que facilita bastante na montagem dos componentes. (ARDUINO, 2024d).

Este protocolo permite a adição de complexidade aos projetos sem aumentar a complexidade da fiação, sendo suportado por todas as placas Arduino. Ele possibilita a conexão de múltiplos dispositivos periféricos, como sensores, *displays* e *drivers* de motor, utilizando apenas alguns fios, o que proporciona maior flexibilidade e acelera a prototipagem (ARDUINO, 2024d).

A Figura 2 traça um paralelo entre os modelos existentes de Arduino e a disponibilidade de pinos para conexão I2C.

Form factor	SDA	SCL	SDA1	SCL1	SDA2	SCL2
UNO	SDA/A4	SCL/A5				
Nano	A4	A5				
MKR	D11	D12				
GIGA	D20	D21	D102	D101	D9	D8
Mega & Due	D20	D21				

Figura 2 Pinos I2C do Arduino

Fonte: <<https://docs.arduino.cc/learn/communication/wire>>

Cada via I2C envolve o uso de duas linhas para enviar e receber dados: um *Serial Clock Pin* (SCL) que define a frequência do sinal em intervalos regulares e *Serial Data Pin* (SDA) pelo qual os dados são enviados entre os dois dispositivos (ARDUINO, 2024d). Como é possível constatar, o modelo adotado neste trabalho (GIGA) é o que mais possui pinos sendo: três Serial Clock Pin (SCL) (D21, D101 e D8) e outros três Serial Data Pin (SDA) (D20, D102 e D9).

Com o objetivo de simplificar o desenvolvimento, foram utilizadas as três portas de interface I2C para os sensores BMP280, SIIT20 e o *Liquid Crystal Display* (LCD).

3.1.1.2 Sensores

Conforme Michaelis (2024), o termo “sensor” se refere a um “dispositivo ou equipamento que, sensível a estímulos magnéticos, motores, de calor, de luz, de pressão, de som, etc., é capaz de converter essa energia e transmitir um impulso correspondente.” São parte essencial do projeto e cada qual é descrito abaixo:

1. Sensor de pressão **BMP280**:



Figura 3 – Sensor BMP280

Fonte: <<https://github.com/Erriez/ErriezBMX280>>

O sensor BMP280A, representado na Figura 3, é amplamente utilizado para medir a pressão atmosférica devido ao seu preço acessível e à facilidade de encontrá-lo no

mercado. Ele se destaca por suas dimensões compactas, baixo consumo de energia e tem um baixo valor de erro de ruído em comparação com outros sensores (KUSUMA et al., 2023).

Ademais, este sensor também é capaz de medir a temperatura ambiente, mas foi utilizado apenas para capturar a pressão atmosférica.

A Figura 4 demonstra o circuito de instalação com o Arduino.

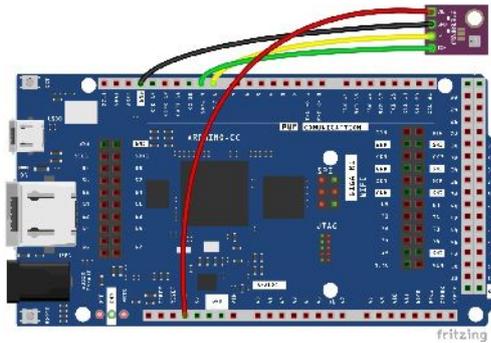


Figura 4 – Circuito BMP280

Abaixo, a Tabela 3 mostra as especificações entregues pelo fabricante.

Tabela 3 – Especificações BMP280

Faixa de medição de pressão	300 hPa a 1100 hPa
Precisão absoluta de pressão	± 1 hPa
Interface	I2C

Fonte: Sensortec (2015)

2. Sensor de temperatura e umidade **SHT20**



Figura 5 – Sensor SHT20

Fonte: <<https://www.usinainfo.com.br/sensor-de-umidade-arduino/sensor-de-umidade-e-temperatura-sht20-encapsulamento-ip65-4999.html>>

O sensor SHT20 da marca *Sensirion*, representado na Figura 5, é amplamente utilizado por sua estrutura compacta (3 x 3 mm) e alta precisão. Ele utiliza sensores capacitivos de umidade e de temperatura baseados em tecnologia *CMOSens®* e fornece sinais digitais calibrados via interface I2C. Cada unidade é individualmente calibrada e testada, garantindo desempenho confiável e estável. O sensor também oferece a possibilidade de ajustar sua resolução e inclui verificação por *checksum* para aumentar a confiabilidade da comunicação, sendo ideal para aplicações que exigem baixo consumo de energia e alta durabilidade (AG, 2020).

A Figura 6 demonstra o circuito de instalação com o Arduino e a Tabela 4 detalha as especificações técnicas fornecidas pelo fabricante.

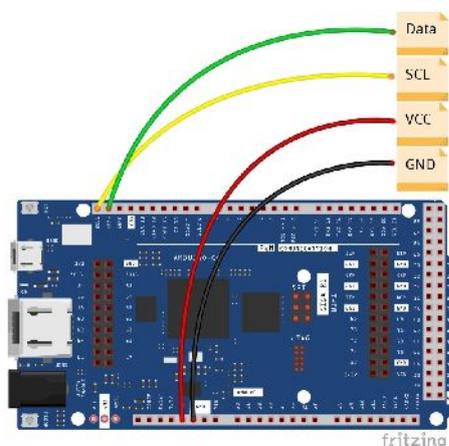


Figura 6 – Circuito SHT20

Tabela 4 – Especificações SHT20

Variável Meteorológica	Tolerância de precisão	Faixa de operação
Umidade	$\pm 3\%$	0% a 100%
Temperatura	0.3°C	-40°C a 125°C

Fonte: AG (2020)

3. Pluviômetro **PB10**



Figura 7 – Pluviômetro

Fonte: <<https://www.usinainfo.com.br/estacao-meteorologica-arduino/pluviometro-de-bascula-digital-para-arduino-e-estacao-meteorologica-pb10-4637.html>>

O pluviômetro de bscula, retratado no Figura 7, funciona com base em um mecanismo simples de coleta de gua da chuva em duas pequenas conchas ou compartimentos. Quando um dos compartimentos se enche com uma quantidade especfica de gua, ele bascula, despejando a gua e acionando um contador. O processo se repete com o outro compartimento. Cada basculada corresponde a uma quantidade conhecida de precipitao, permitindo que o Arduino calcule o volume total de chuva.

A Figura 8 demonstra o circuito de instalao com o Arduino e a Tabela 5 detalha as especificaes tcnicas fornecidas pelo fabricante.

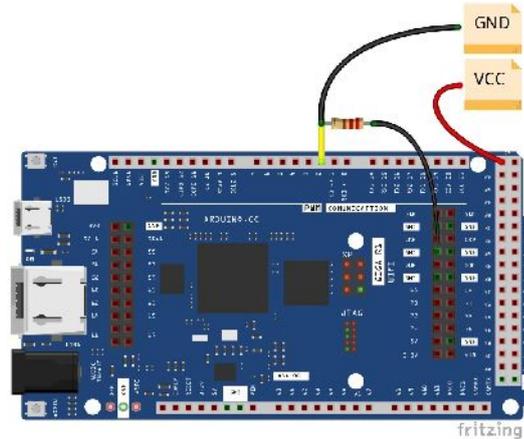


Figura 8 – Circuito pluviômetro

Tabela 5 – Especificações PB10

Diâmetro coletor	147mm
Erro máximo	4% (até 30 mm/h), 5% (até 50mm/h), 7% (até 90 mm/h)

Fonte: UsinaInfo (2024)

3.1.1.3 Demais componentes

1. Display LCD 16x2



Figura 9 – Display LCD 16x2

Fonte: https://m.media-amazon.com/images/I/61FOfovWvEL._AC_SL1010_.jpg

O *display* LCD desempenha um papel importante no projeto fornecendo *feedback* visual em tempo real ao usuário sobre o estado do sistema. Com ele, é possível verificar a conexão do Arduino com a internet, erros dos sensores, envio e retorno de requisições ao servidor. O modelo escolhido, constante na Figura 9, possui 16

colunas e 2 linhas, o que possibilita exibir até 32 caracteres simultaneamente. Sua conexão com o Arduino ocorre através da interface I2C simplificando o circuito e utilizando apenas dois pinos de dados.

A Figura 10 demonstra o circuito de instalação com o Arduino.

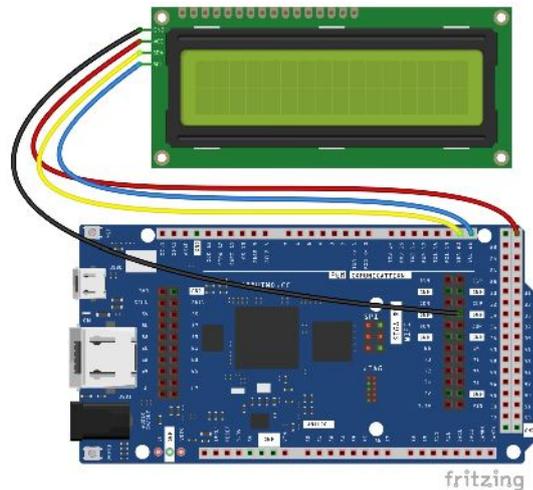


Figura 10 – Circuito display LCD

3.1.1.4 Circuito completo

A Figura 11 demonstra o resultado da montagem de todos os componentes com suas instalações e conexões devidamente realizadas. Por se tratarem de dispositivos eletrônicos sensíveis, tomou-se o cuidado de fixá-los propriamente. Atenção especial também foi despendida à vedação do recipiente de forma a mitigar a influência/interferência de eventuais intempéries ambientais.

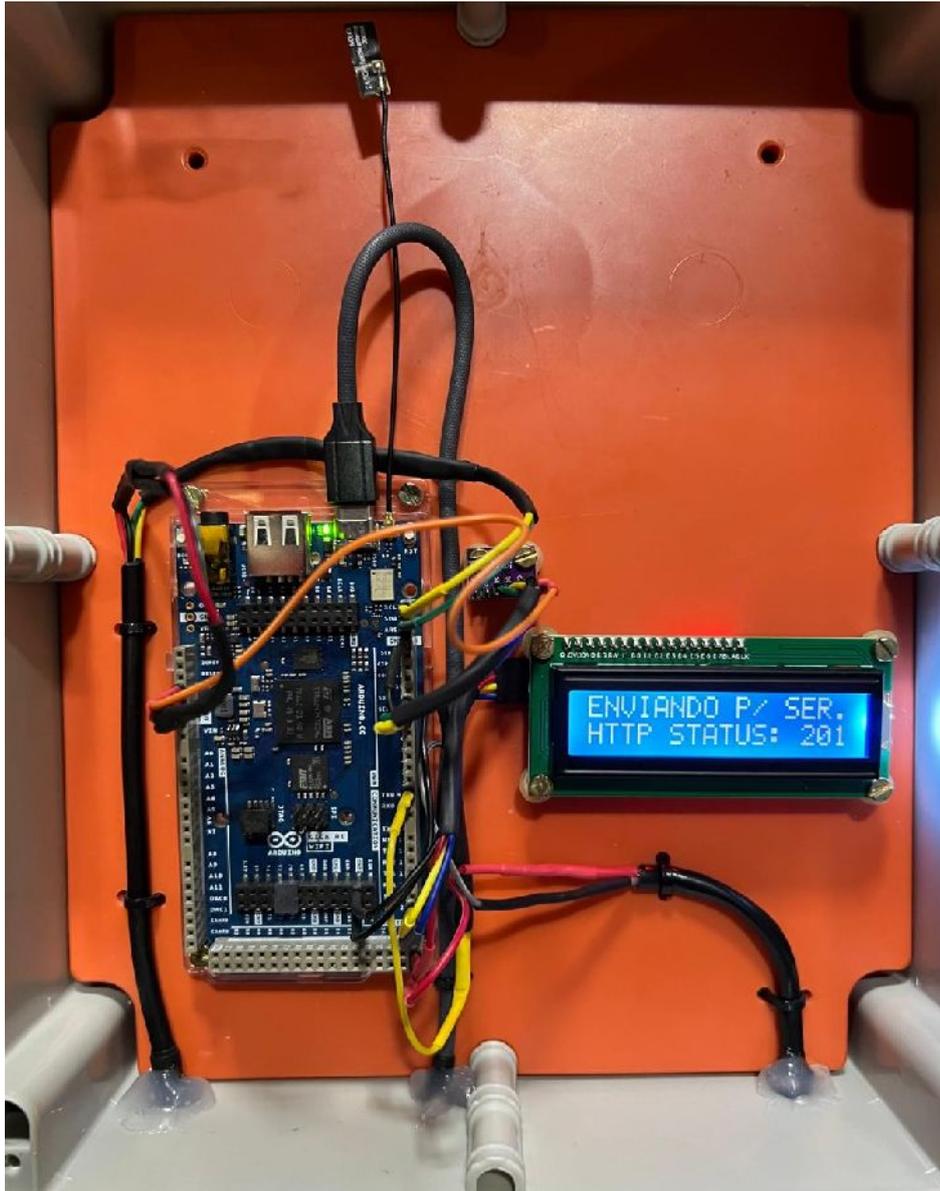


Figura 11 – Circuito completo

3.1.2 Funcionamento

Esta subseção apresenta o funcionamento do sistema embarcado da estação meteorológica automatizada, que utiliza um Arduino para coletar dados. Estas informações são processadas e enviadas para um servidor *web*.

Serão explicadas as etapas de inicialização dos sensores, coleta contínua de dados e envio ao servidor via requisições HTTP, além das bibliotecas que facilitam a comunicação entre os componentes. O objetivo é fornecer uma visão clara de como o sistema integra *hardware* e *software* para monitorar as condições ambientais.

3.1.2.1 Bibliotecas Utilizadas

O código do Arduino faz uso de diversas bibliotecas externas que fornecem interfaces para os sensores e para a comunicação com o servidor. Aqui estão as principais bibliotecas utilizadas:

1. **DFRobot_SHT20**: Utilizada para controlar o sensor SHT20. Documentação: DFRobot (2024).
2. **Adafruit_BMP280**: Utilizada para controlar o sensor BMP280. Documentação: Adafruit (2024).
3. **LiquidCrystal_I2C**: Utilizada para controlar o LCD. Documentação: Brabander (2024).
4. **ArduinoJson**: Utilizada para formatar os dados coletados em um formato JSON. Documentação: Blanchon (2024).
5. **ArduinoHttpClient**: Utilizada para comunicação HTTP com *Webserver*. Documentação: Arduino (2024c).
6. **WiFi**: Utilizada para gerenciar a conexão Wi-Fi. Ela oferece uma interface para conectar o Arduino a redes sem fio e monitorar o status da conexão. Documentação: Arduino (2024e).

3.1.2.2 Inicialização dos Sensores e do Display LCD

Quando o Arduino é energizado, a primeira etapa do processo envolve a inicialização dos sensores e do *display* LCD. Isso é feito pela função *initializeSensors()* (Figura 12), que prepara o sistema para começar a coletar dados.

```
9 void initializeSensors(DFRobot_SHT20 sht20, Adafruit_BMP280 &bmp, LiquidCrystal_I2C &lcd) {
10
11     lcd.begin();
12     lcd.backlight();
13     lcd.clear();
14
15     lcd.setCursor(0, 0);
16     lcd.print("INICIALIZANDO");
17     lcd.setCursor(0, 1);
18     lcd.print("SENSORES...");
19
20     sht20.initSHT20();
21
22     pinMode(RAIN_SENSOR_PORT, INPUT_PULLUP);
23
24     if (!bmp.begin(0x76)) {
25
26         lcd.clear();
27         lcd.setCursor(0, 0);
28         lcd.print("SENSOR BMP280");
29         lcd.setCursor(0, 1);
30         lcd.print("NÃO DETECTADO...");
31         while (1)
32             ;
33     }
34
35     delay(3000UL);
36 }
```

Figura 12 – Função *initializeSensors()* no Arduino

1. **Display LCD:** A função *lcd.begin()* inicializa o *display*, enquanto *lcd.backlight()* ativa a luz de fundo, proporcionando maior visibilidade das informações mostradas. Após a inicialização do LCD é exibida a seguinte mensagem: “INICIALIZANDO SENSORES...”.
2. **Sensor SHT20:** O SHT20 é inicializado pela função *sht20.initSHT20()*.
3. **Sensor BMP280:** O BMP280 é inicializado pela função *bmp.begin()*. Caso o sensor não seja detectado, o Arduino exibe a mensagem de erro “SENSOR BMP280 NÃO DETECTADO...” no *display* LCD, impedindo que o sistema continue sem leituras válidas de pressão e altitude.
4. **Pluviômetro de Bâscula:** O pluviômetro, conectado à porta digital 2, utiliza o modo *INPUT_PULLUP* para detectar a quantidade movimentação da balsa.

3.1.2.3 Conexão à Rede Wi-Fi

Após a inicialização dos sensores, o Arduino tenta se conectar à internet através da função *connectWifi()* (Figura 13).

```
22 void connectWifi(LiquidCrystal_I2C &lcd) {
23
24     if (WiFi.status() != WL_CONNECTED) {
25
26         lcd.clear();
27         lcd.setCursor(0, 0);
28         lcd.print("SEM CONEXAO WIFI");
29
30         if (WiFi.status() == WL_NO_SHIELD) {
31
32             lcd.setCursor(0, 1);
33             lcd.print("FALHA MOD. WIFI");
34             while (true)
35                 ;
36         }
37
38         int statusWifi = WL_IDLE_STATUS;
39
40         while (statusWifi != WL_CONNECTED) {
41
42             lcd.setCursor(0, 1);
43             lcd.print("CONECTANDO...");
44
45             statusWifi = WiFi.begin(SECRET_SSID, SECRET_PASS);
46
47             delay(3000);
48         }
49         printWifiStatus(lcd);
50     }
51 }
```

Figura 13 – Função *connectWifi()* no Arduino

1. **Gerenciamento da Conexão:** A função *WiFi.begin(SECRET_SSID, SECRET_PASS)* conecta o Arduino à rede Wi-Fi cujas credenciais são armazenadas nas variáveis *SECRET_SSID* (nome da rede) e *SECRET_PASS* (senha). Se a conexão não for bem-sucedida, o Arduino mantém a mensagem “CONECTANDO...” no *display* LCD e continua tentando até obter sucesso.
2. **Reconexão Automática:** A cada ciclo no *loop()*, o código verifica se o Arduino ainda está conectado à rede Wi-Fi. Caso a conexão seja perdida, o Arduino tenta se reconectar automaticamente, garantindo que os dados coletados não sejam perdidos por falhas de conexão. Isso torna o sistema robusto em ambientes onde a conectividade Wi-Fi possa oscilar.
3. **Exibição de Status:** Uma vez conectado, o Arduino exibe o nome da rede e o endereço *Internet Protocol* (IP) local no LCD, permitindo ao usuário verificar o sucesso da conexão.

3.1.2.4 Coleta de Dados dos Sensores

A cada ciclo de execução da função *loop()* (Figura 14), o Arduino coleta dados dos sensores em intervalos regulares.

```
29 void loop() {
30
31   WiFiClient wifiClient;
32   HttpClient client = HttpClient(wifiClient, SERVER, PORT);
33
34   readRain(amountOfRain);
35   float rain = amountOfRain * 0.2238;
36   float humidity = sht20.readHumidity();
37   float temperature = sht20.readTemperature();
38   float pressure = (bmp.readPressure() / 100);
39   float altitude = bmp.readAltitude(1013.25)
40
41   postRequest(client, lcd, humidity, temperature, pressure, altitude, rain);
42
43   delay(741UL);
44   connectWifi(lcd);
45 }
```

Figura 14 – Função *loop()* no Arduino

Os dados coletados incluem:

1. **Temperatura:** Dado obtido através da função *sht20.readTemperature()*.
2. **Umidade:** Dado obtido através da função *sht20.readHumidity()*.
3. **Pressão atmosférica:** Dado obtido através da função *bmp.readPressure()*. Esta função retorna o valor em Pascal (Pa). Para converter em hectopascal (hPa), o resultado é dividido por 100.
4. **Altitude:** Dado obtido através da função *bmp.readAltitude(float seaLevelhPa)*. Esta função recebe como parâmetro a pressão padrão ao nível médio do mar que é 1013,2 hectopascal (hPa) (BRASILEIRA, 2024).
5. **Quantidade de chuva:** O pluviômetro de balança detecta a quantidade de chuva acumulada a partir de uma série de leituras de alta e baixa tensão (representando o movimento da balança). O número de basculadas é multiplicado por um fator de calibração (0,2238 mm por basculada) para calcular o volume de chuva em milímetros.

3.1.2.5 Formatação e Envio dos Dados ao Servidor

Os dados coletados são formatados em um objeto JSON utilizando a biblioteca *ArduinoJson.h*. A função *postRequest()* (Figura 15) é responsável por enviar esses dados ao servidor web via uma requisição HTTP *POST*.

```

7 void postRequest(HttpClient client, LiquidCrystal_I2C &lcd, float humidity, float temperature,
8   float pressure, float altitude, float rain) {
9
10  lcd.clear();
11  lcd.setCursor(0, 0);
12  lcd.print("ENVIANDO P/ SER.");
13
14  JsonDocument request;
15  request[String("temperatura")] = temperature;
16  request[String("umidade")] = humidity;
17  request[String("pressao")] = pressure;
18  request[String("altitude")] = altitude;
19  request[String("chuva")] = rain;
20
21  String jsonRequest;
22  serializeJson(request, jsonRequest);
23
24  client.post("/v1/estacao-meteorologica", "application/json", jsonRequest);
25
26  int statusCode = client.responseStatusCode();
27
28  lcd.setCursor(0, 1);
29  lcd.print("HTTP STATUS: ");
30  lcd.setCursor(13, 1);
31  lcd.print(statusCode);
32 }

```

Figura 15 – Função *postRequest()* no Arduino

1. **Formato JSON:** A biblioteca *ArduinoJson* serializa os valores de temperatura, umidade, pressão, altitude e chuva em um objeto JSON através da função *serializeJson()*. A estrutura do JSON é organizada da seguinte maneira:

```

{
  "temperatura": 0,
  "umidade": 0,
  "pressao": 0,
  "altitude": 0,
  "chuva": 0
}

```

2. **Requisição HTTP POST:** A função *client.post()* envia o objeto JSON para o *webservice* no *endpoint* */v1/estacao-meteorologica*. O status da resposta HTTP é lido pela função *client.responseStatusCode()* e exibido no LCD conforme demonstrado na Figura 11.

3.1.2.6 Ciclo de Operação Contínua

O ciclo de operação do Arduino segue uma sequência contínua, onde a cada iteração ele:

1. Coleta os dados dos sensores;

2. Envia esses dados ao servidor;
3. Verifica a conexão Wi-Fi e tenta reconectar, se necessário.

Entre cada ciclo, o Arduino aguarda um curto intervalo de tempo antes de repetir o processo, o que permite a coleta regular de dados e a comunicação eficiente com o servidor. O LCD exibe informações importantes ao longo do processo, como o status da conexão Wi-Fi, envio da requisição ao servidor e o código de resposta HTTP.

3.2 Webservice

O *Webservice* nada mais é do que o servidor que receberá e processará as requisições oriundas tanto do Arduino quanto do *Webclient*. Após sua construção, o mesmo é disponibilizado em serviço *web* tendo seu acesso realizado por meio de *endpoints* devidamente confeccionados para receber e tratar dados brutos, armazená-los quando conveniente, e reter informações enriquecidas.

3.2.1 Linguagem e ferramentas

Neste tópico serão abordados os meios utilizados para o desenvolvimento do *Webservice*. Buscou-se optar por tecnologias consolidadas no mercado corrente.

3.2.1.1 Java

A escolha da linguagem de programação *Java* para o desenvolvimento deste projeto foi fundamentada na experiência e familiaridade prévia do pesquisador, o que garantiu maior produtividade e segurança na implementação das funcionalidades essenciais. A utilização de uma linguagem já dominada pelo desenvolvedor permitiu focar no aprimoramento das funções específicas da aplicação, reduzindo o tempo de desenvolvimento e minimizando a ocorrência de erros comuns relacionados à curva de aprendizado com novas tecnologias.

Essa escolha se mostrou ainda mais acertada pela extensa comunidade de desenvolvedores e pela rica documentação disponível, fatores que proporcionam um suporte contínuo ao longo do desenvolvimento, facilitando a resolução de problemas.

No contexto do *Webservice*, *Java* oferece suporte para o desenvolvimento de APIs, que são responsáveis por receber e processar as requisições do Arduino. Essas requisições, que contêm os dados dos sensores em formato JSON, são processadas e armazenadas em um banco de dados *MongoDB*.

3.2.1.2 Spring Boot

O *framework Spring Boot* foi escolhido por sua simplicidade e rapidez no desenvolvimento de aplicações *Java*. Ele reduz a necessidade de configurações complexas, permitindo

uma integração fácil com bancos de dados e oferecendo suporte nativo para a criação de APIs. Sua vasta documentação e comunidade facilitam a resolução de problemas, enquanto sua escalabilidade garante que a aplicação possa crescer sem grandes mudanças na arquitetura.

3.2.2 Banco de dados

O *MongoDB* foi utilizado neste projeto por ser um banco de dados *NoSQL* que trabalha com documentos no formato JSON, o que facilita a manipulação de dados não estruturados, como os que são enviados pela estação meteorológica. Outra razão para sua escolha é sua fácil integração com o *Spring Boot*, simplificando a comunicação entre a API e o banco de dados. Assim, o *MongoDB* foi escolhido por sua flexibilidade, desempenho e boa integração com o ambiente *Java*, tornando-se uma opção prática e eficiente para este projeto.

3.2.3 Funcionamento

O *Webserver* foi desenvolvido na estrutura de microsserviço utilizando *Java 17*, *framework Spring Boot 3* e banco de dados *MongoDB*. Suas principais funções são receber, armazenar, processar e fornecer dados meteorológicos. Abaixo estão explicações detalhadas do seu funcionamento:

3.2.3.1 Recepção e armazenamento dos dados

A API expõe um *endpoint* principal para a inserção de dados meteorológicos, implementado na classe **EstacaoMeteorologicaController**. Esse *controller* é responsável por receber as requisições HTTP *POST* enviadas pelo Arduino. O *endpoint* correspondente é o `/v1/estacao-meteorologica`, descrito na Figura 16. Este recebe as informações coletadas pelos sensores de temperatura, umidade, pressão atmosférica e milímetros de chuva no corpo da requisição, no formato JSON, conforme Figura 17.

```
28  @PostMapping("/v1/estacao-meteorologica") no usages ± JGontijo
29  @ResponseStatus(CREATED)
30  void incluir(@RequestBody InformacaoMeteorologicaDto request) {
31
32      logRequestController(INCLUIR_DADOS_METEOROLOGICOS, request);
33
34      var informacaoMeteorologica = informacaoMeteorologicaDtoMapper.mapToInformacaoMeteorologica(request);
35      estacaoMeteorologicaApplication.incluir(informacaoMeteorologica);
36
37      logResponseController(INCLUIR_DADOS_METEOROLOGICOS);
38  }
```

Figura 16 – Endpoint para inclusão dos dados meteorológicos

```
1  {
2    "temperatura": 0,
3    "umidade": 0,
4    "pressao": 0,
5    "altitude": 0,
6    "chuva": 0
7  }
```

Figura 17 – JSON de inclusão dos dados meteorológicos

Ao receber essa requisição, a API processa os dados e os armazena em um banco de dados *MongoDB*, dentro de uma *collection* chamada **informacoes-meteorologicas**. Cada novo conjunto de dados meteorológicos gera um novo documento na coleção, permitindo o registro histórico das medições realizadas pela estação meteorológica, conforme exemplo presente na Figura 18.

```
_id: ObjectId('66e979f0fc11264fd7e7da86')
temperatura : 30.63883018
umidade : 36.80853271
pressao : 917.2243652
altitude : 832.0327759
chuva : 0
dataLeitura : 2024-09-17T12:45:36.464+00:00
_class : "br.com.estacaometeorologica.adapter.secondary.persistence.estacaometeo..."
```

Figura 18 – Registo na *collection* informacoes-meteorologicas

Após o armazenamento bem-sucedido das informações no banco de dados, o *endpoint* responde ao Arduino com o código de status HTTP 201 (*created*), indicando que a requisição foi bem-sucedida e um novo recurso foi criado. O corpo da resposta é vazio, pois o foco do *endpoint* é apenas a persistência dos dados. Caso ocorra algum erro no processo de inclusão, a API retorna o código de erro HTTP 500 (*internal server error*).

3.2.3.2 Rotina para agrupamento

A API possui um *scheduler* que desempenha um papel importante no processamento dos dados meteorológicos. Sua função principal é realizar, a cada hora, uma tarefa automática que agrupa os dados armazenados no banco de dados. O propósito é otimizar o armazenamento e análise ao consolidar os dados coletados ao longo do tempo.

Especificamente, a API consulta a *collection* **informacoes-meteorologicas**, onde os dados brutos são inseridos a cada minuto, e seleciona todos os registros dos últimos 60 minutos. Em seguida, o *scheduler* calcula a média dos valores de temperatura, umidade, pressão atmosférica e acumulado de milímetros de chuva. Esses cálculos são realizados para criar um panorama consolidado do comportamento dos dados meteorológicos naquela hora específica.

Após o cálculo das médias e acumulado, os resultados são armazenados em uma *collection* separada, chamada **agrupamento-dados-meteorologicos**. Esta *collection* possui uma estrutura semelhante à dos dados originais, mas reflete o valor médio de cada parâmetro e acumulado de chuva, como é possível observar na Figura 19. Além disso, cada registro consolidado recebe um *timestamp* com a data e a hora de referência, facilitando a organização e consulta dos dados de forma eficiente.

A execução automatizada do *scheduler* ocorre de forma periódica, garantindo que a cada hora haja um novo registro agregado, mantendo os dados disponíveis para futuras análises, relatórios e visualizações gráficas pelo *Webclient*. Esse mecanismo torna o sistema mais eficiente e escalável, evitando consultas desnecessárias a grandes volumes de dados brutos a cada requisição.

```
_id: ObjectId('66df37aac0141f10ac25a10d')
temperatura : 33.34191998
umidade : 24.409732107288136
pressao : 916.1037359728814
altitude : 839.1382446
chuva : 0
dataReferencia : 2024-09-09T03:00:00.000+00:00
horaReferencia : 15
_class : "br.com.estacaometeorologica.adapter.secondary.persistence.estacaometeo..."
```

Figura 19 – Registro na *collection* agrupamento-dados-meteorologicos

3.2.3.3 Fornecimento de Dados para o Frontend

Para expor as informações agrupadas ao *frontend*, foi desenvolvido o *endpoint GET* no recurso `/v1/estacao-meteorologica/resumo-diario`, descrito na Figura 20. Ele recebe como parâmetro uma data de referência, a partir da qual a API consulta a *collection* **agrupamento-dados-meteorologicos** para obter os dados meteorológicos agregados de cada hora ao longo do dia solicitado. Esses dados são utilizados para construir os gráficos de linhas exibidos no *frontend*, permitindo a visualização do comportamento das variáveis meteorológicas ao longo do dia.

```
40 @GetMapping("/v1/estacao-meteorologica/resumo-diario") no usages ± JGontijo
41 @ResponseStatus(OK)
42 ConsultaResumoDiarioDto buscarResumoDiario(@RequestParam LocalDate data) {
43
44     LogRequestController(BUSCAR_RESUMO_DIARIO);
45
46     var resumoDiario = estacaoMeteorologicaApplication.buscarResumoDiario(data);
47     var response = consultaResumoDiarioDtoMapper.mapToConsultaResumoDiarioDto(resumoDiario);
48
49     LogResponseController(BUSCAR_TODOS_DADOS_METEOROLOGICOS, response);
50     return response;
51 }
```

Figura 20 – Endpoint para consulta dos dados meteorológicos agregados

Além de fornecer os valores agregados de cada hora do dia solicitado, a resposta da API também inclui:

1. Valores máximos e mínimos das variáveis meteorológicas de temperatura, umidade e pressão atmosférica no dia da consulta;
2. Acumulado de milímetros de chuva;
3. Última medição bruta de cada variável meteorológica, com base na última requisição recebida do Arduino. Esse valor é útil para o *frontend* exibir o estado mais recente de cada parâmetro meteorológico;
4. A primeira altitude registrada no dia solicitado. Como a altitude tende a permanecer constante, ela é registrada apenas uma vez e reaproveitada ao longo do dia.

3.3 Webclient

O *Webclient* foi concebido com o objetivo de consumir as informações ofertadas pelo *Webserver* e expô-las de forma inteligível ao usuário final. Logo abaixo serão descritos os recursos utilizados em sua confecção bem como seu funcionamento.

3.3.1 Linguagem e ferramentas

Conforme Flanagan (2012), *Javascript* é a linguagem de programação da *Web*. Para a confecção do *Webclient*, disposto em ambiente *web*, foi utilizado esta linguagem juntamente à biblioteca *React*.

Outra biblioteca complementar ao *React* que foi importante na construção do *frontend*, foi a *Ant Design Charts*, responsável pela formatação dos dados recebidos pelo *Webserver* em gráficos de linhas interativos.

3.3.2 Funcionamento

Após o *Webserver* receber e tratar os dados oriundos do Arduino/Estação Meteorológica, os mesmos são persistidos no banco de dados onde ficarão disponíveis para serem consumidos pelo *Webclient*.

O contrato estabelecido, demanda apenas que o cliente remeta a data de interesse, constando dia, mês e ano. De posse desta informação, o servidor entrega ao *frontend* uma série de dados meteorológicos, apropriadamente organizados em intervalos de hora em hora, contendo os valores do dia selecionado até o momento da requisição.

Para cada uma das grandezas trabalhadas: Temperatura, Umidade, Pressão atmosférica e Volume de chuva, a interface renderizará na tela um gráfico. O eixo x será graduado em horas completas e terá a representação das 24(vinte e quatro) horas do dia. O eixo

y será graduado na unidade de medida da grandeza correspondente e sua representação será dinâmica, se adequando ao intervalo recebido.

Acompanharão cada um destes gráficos, os mostradores contendo os respectivos valores máximos e mínimos. Excetua-se o Volume de chuva que mostrará somente o acumulado do dia. Caso a data remetida seja correspondente à atual, também serão exibidos mostradores contendo as últimas medições realizadas, extraídas daquelas feitas a cada minuto pela Estação.

Foi estabelecido como regra de negócio, que o usuário poderá escolher uma data que resida no período entre o dia corrente os 15(quinze) dias anteriores. Para garantir a integridade desta regra, o componente visual seletor de datas foi configurado para restringir a escolha de qualquer dia fora do intervalo o supracitado.

3.3.3 Diagrama de caso de uso

Como a interação do usuário com o *Webclient* é restrita somente à seleção do dia em que serão exibidos os dados meteorológicos coletados, há somente um caso de uso possível, conforme demonstrado na Figura 21 abaixo.

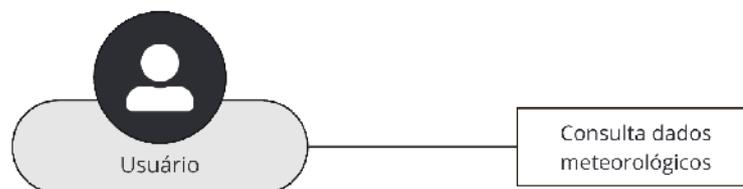


Figura 21 – Diagrama de caso de uso

3.3.4 Interface do usuário

A Figura 22 apresenta a interface do usuário, destacando o gráfico de linha referente à variável de temperatura ao longo do dia selecionado. O ponto indicado pela letra **A** representa o seletor de data, que permite ao usuário escolher o dia para visualização dos dados. Já o ponto marcado com a letra **B** corresponde ao botão “Selecionar”, responsável por enviar a requisição ao *Webserver* e carregar os dados meteorológicos da data selecionada.

O gráfico exibido é idêntico para pressão atmosférica e umidade, alterando-se apenas as informações apresentadas.

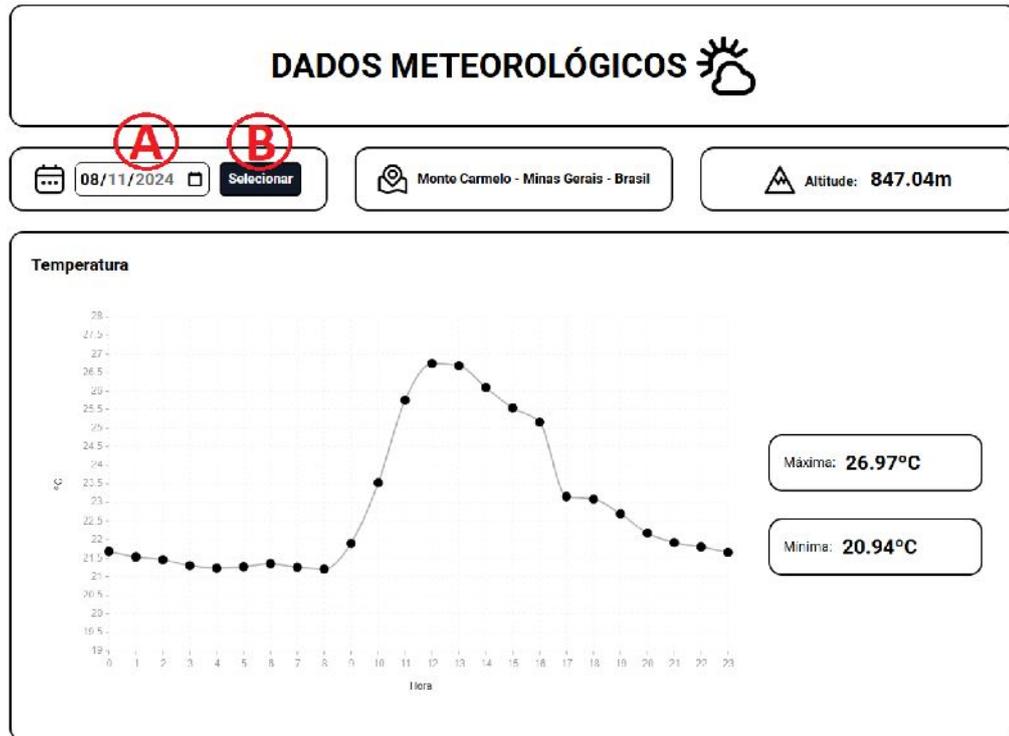


Figura 22 – Interface para temperatura

A Figura 23 apresenta o gráfico de linha referente à precipitação acumulada ao longo do dia selecionado. Além do gráfico, que exibe a distribuição da chuva em intervalos de horários, é apresentado também o valor total do acumulado diário, permitindo ao usuário uma visão completa sobre o volume de precipitação registrado na data escolhida.

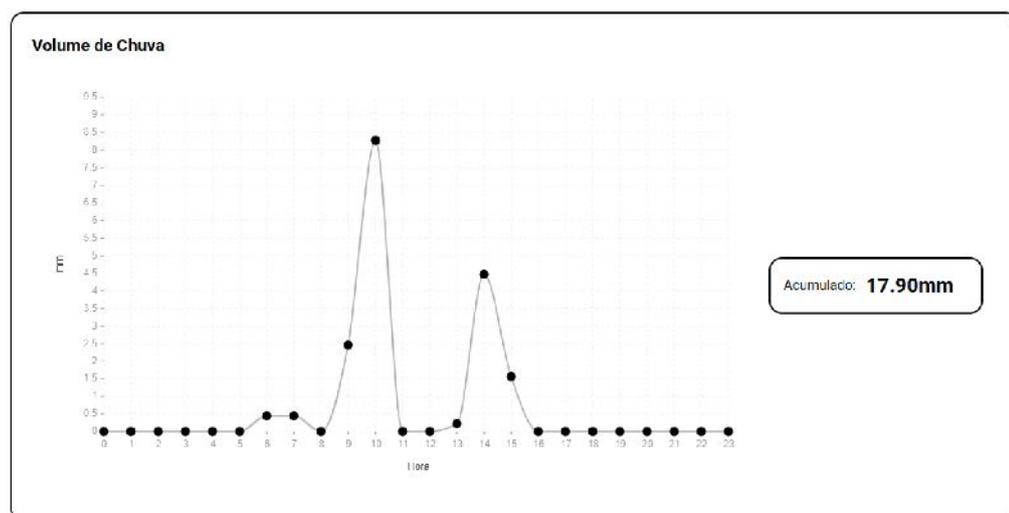


Figura 23 – Interface para precipitação

3.4 Diagrama da arquitetura

O fluxo de dados no sistema ocorre da seguinte forma:

1. O Arduino realiza medições a cada 60 segundos com o auxílio dos sensores;
2. Processa as leituras e, por meio do módulo Wi-Fi, envia os dados para a API usando o protocolo HTTP;
3. O *Webserver* recebe os dados e os persiste no banco de dados *MongoDB*;
4. Uma rotina é executada no *Webserver* a cada hora para agregar os dados brutos dos últimos 60 minutos em uma nova *collection*;
5. O *frontend* realiza requisição ao servidor para obter os dados e exibí-los ao usuário em forma de gráfico;



Figura 24 – Diagrama da arquitetura

A Figura 24 demonstra de forma gráfica e sintética o trajeto das informações e as tecnologias utilizadas em cada camada.

O código-fonte de todo o projeto está disponível em repositório público¹ na plataforma *Github*.

¹ <<https://github.com/EstacaoMeteorologicaArduino/estacao-meteorologica-arduino-spring-react>>.

Método para a Avaliação

Para garantir a confiabilidade dos dados coletados pela estação meteorológica automatizada, foram realizadas validações de todos os sensores utilizados no sistema. O objetivo foi verificar a precisão dos sensores em comparação com instrumentos e serviços existentes e consolidados, fornecendo assim um respaldo técnico para a qualidade das informações obtidas.

Além da validação dos componentes físicos, a interface do *Webclient*, bem como seu funcionamento junto a todo o sistema, foi submetido a uma análise heurística junto aos usuários finais.

4.1 Validação com instrumentos auxiliares

Inicialmente, foram realizadas medições paralelas utilizando sensores auxiliares. As coletas foram realizadas no mesmo local da estação meteorológica. Estas ocorreram no período compreendido entre os dias 16/09/2024 e 20/09/2024, para os sensores de temperatura, umidade e pressão atmosférica. As coletas foram realizadas em intervalos de 4 horas no decorrer do dia, obedecendo os seguintes horários: 03:00, 07:00, 11:00, 15:00, 19:00 e 23:00. Os dados estão disponíveis no *GitHub*¹.

Para a validação da precipitação de chuva (pluviômetro), foram considerados os valores acumulados diários. Para tanto, este sensor foi avaliado com base em comparações com outras três estações localizadas na região de Monte Carmelo, em datas distintas dos demais sensores. A diferença nas datas ocorreu devido à ausência de precipitação durante o período de validação dos sensores.

A seguir, será detalhado o processo de avaliação de cada sensor.

¹ <<https://github.com/EstacaoMeteorologicaArduino/estacao-meteorologica-arduino-spring-react>>

4.1.1 Sensor de Pressão Atmosférica BMP280

O instrumento utilizado para medir a pressão atmosférica é o barômetro (Instituto Nacional de Meteorologia (INMET), 1999).

Para validar a precisão do sensor BMP280, foi utilizado o *iPhone 13 Pro Max* que possui barômetro integrado (Apple Support, 2024).

A *Apple* não disponibiliza um aplicativo nativo para acessar os dados do sensor, por isso as informações de pressão do barômetro do *iPhone* foram obtidas com o auxílio do aplicativo *Barometer* disponível na *App Store*.

4.1.2 Sensor de Temperatura e Umidade SHT20

Para a medição de temperatura e umidade, foi utilizado o termômetro digital GM1362. Adicionalmente, um termômetro de mercúrio com escala de -40°C a 50°C foi empregado exclusivamente para as leituras de temperatura.

O GM1362 é um dispositivo portátil que possui sensores sensíveis que garantem alta precisão, alta estabilidade e resposta rápida às mudanças nas condições ambientais. Ele vem equipado com um *display* LCD que exibe simultaneamente as informações coletadas Benetech (2024).

As especificações fornecidas pelo fabricante para o GM1362 são descritas abaixo na Tabela 6.

Tabela 6 – Especificações GM1362

Variável Meteorológica	Faixa de Medição	Precisão da Medição
Umidade	10% a 99%	$\pm 5\%$
Temperatura	-10°C a 50°C	$\pm 1^{\circ}\text{C}$

Fonte: Benetech (2024)

4.1.3 Pluviômetro de Bâscula

Para a validação do pluviômetro, foi necessário realizar um processo de calibração para determinar a quantidade exata de água que faz cada balsa se mover, assim como calcular quantos milímetros de precipitação são representados por cada movimento.

Após a calibração, foram realizadas comparações com três estações meteorológicas automáticas instaladas na cidade de Monte Carmelo, entre 07/10/2024 e 12/10/2024, a saber:

- Estação Meteorológica do Centro de Inteligência em Cultivos Irrigados (CinCi), localizada na UFU, no campus Monte Carmelo, na rodovia LMG 746, km 01;

- Estação Meteorológica da Cooperativa Regional de Cafeicultores em Guaxupé (COOXUPÉ), localizada no Núcleo de Monte Carmelo da empresa, na rodovia MG 190, km 03;
- Estação Meteorológica do Centro Nacional de Monitoramento e Alertas de Desastres Naturais (CEMADEN), localizada na Av. Paranaíba, 190, Res. Dona Quita, Monte Carmelo - MG.

Os dados da estação meteorológica da CinCi foram cedidos pela UFU por meio de acesso histórico de medições colhidas. Os dados da estação da COOXUPÉ foram coletados no sítio eletrônico², disponibilizado pela cooperativa na internet. Os dados do CEMADEN foram encontrados no portal³, disponibilizado pela unidade de pesquisa.

4.1.3.1 Calibração da báscula

O primeiro passo foi determinar quantos mililitros (ml) de água são necessários para bascular a báscula do pluviômetro. Utilizando uma seringa, foram conduzidos vários testes onde quantidades controladas de água foram adicionadas à báscula. A cada teste, foi observado o momento em que a báscula basculava. Após múltiplas medições, chegou-se à conclusão que cada báscula precisa de 3,8 ml de água para bascular.

4.1.3.2 Cálculo da quantidade de água equivalente a 1 mm de precipitação

O pluviômetro utilizado possui um diâmetro de 14,7 cm, e o cálculo de volume de água para 1 mm de precipitação foi feito com base na área da superfície de coleta. A área foi calculada utilizando a fórmula da área de um círculo e o volume de um cilindro conforme exposto por Souza (2016):

$$A = \pi r^2$$

Onde r é o raio, que corresponde à metade do diâmetro:

$$r = \frac{14,7}{2} = 7,35cm$$

Substituindo na fórmula da área:

$$A = 3,1416 \times (7,35)^2 \approx 169,83cm^2$$

O volume de água correspondente a 1 mm de chuva (ou seja, 1 mm de altura de água acumulada sobre essa área) é o produto da área pela altura da água:

$$V = A \times 0,1 = 169,83 \times 0,1 = 16,98ml$$

² <<https://sismet.cooxupe.com.br:9000/dados/estacao/pesquisarDados/?estCooxupe=1&cdEstacao=12>>

³ <https://resources.cemaden.gov.br/graficos/interativo/grafico_CEMADEN.php?idpcd=9368&uf=MG>

Assim, concluiu-se que 16,98 ml de água são necessários para representar 1 mm de precipitação.

4.1.3.3 Cálculo da quantidade de chuva correspondente a uma basculada

Sabendo que uma balança se move com 3,8 ml de água e que são necessários 16,98 ml para 1 mm de chuva, foi realizada uma regra de três para determinar quantos milímetros de precipitação cada basculada representa:

$$x = \frac{3,8 \times 1}{16,98} \approx 0,2238mm$$

Com base nesses cálculos, foi possível calibrar o pluviômetro de maneira precisa para medir a quantidade de precipitação. Cada movimento da balança corresponde a 0,2238 mm de chuva, o que garante a confiabilidade e precisão dos dados coletados pela estação meteorológica. Esse valor foi programado no Arduino, que o utiliza para multiplicar pela quantidade de basculadas registradas, determinando assim o total acumulado de precipitação.

4.2 Validação com *sites* de informações meteorológicas

Como complemento, também foram coletados dados de *sites* de informações meteorológicas. O período de coleta também foi entre 07/10/2024 e 11/10/2024. A escolha dos serviços foi baseada no ranqueamento dos primeiros resultados da pesquisa no *Google* com a frase “Temperatura agora”. Os selecionados foram:

- ❑ <<https://weather.com>>
- ❑ <<https://tempoagora.uol.com.br>>
- ❑ <<https://www.climatempo.com.br>>

As coletas também foram realizadas em intervalos de 4 horas no decorrer do dia, obedecendo os seguintes horários: 03:00, 07:00, 11:00, 15:00, 19:00 e 23:00.

4.3 Avaliação dos usuários do sistema

A avaliação da usabilidade e interface da estação meteorológica foi realizada por meio de um formulário de pesquisa aplicado entre os dias 12 e 18 de outubro de 2024. O formulário foi direcionado a alunos e professores dos cursos de Sistemas de Informação e Agronomia da Universidade Federal de Uberlândia, além de outros usuários da estação meteorológica, visando obter um *feedback* amplo e diversificado. As respostas foram

anônimas, garantindo a confidencialidade dos participantes. Um Termo de Consentimento Livre e Esclarecido (TCLE) foi incluído para garantir que os respondentes compreendam os objetivos da pesquisa e autorizem sua participação. O TCLE pode ser encontrado no apêndice A.

O questionário, composto por 10 perguntas, sendo 9 de múltipla escolha e 1 aberta para críticas e sugestões, teve como foco a usabilidade, a interface gráfica, a precisão dos dados meteorológicos e as funcionalidades do sistema. O tempo médio de resposta foi de aproximadamente 10 minutos.

Os participantes avaliaram diversos aspectos do sistema, incluindo:

- Facilidade de acesso e uso do *website*;
- Claridade e atratividade da interface gráfica;
- Utilidade das funcionalidades que apresentam valores máximos e mínimos de temperatura, umidade, pressão atmosférica e acumulado de precipitação;
- Compreensão dos dados meteorológicos apresentados em gráficos de linha;
- Adequação da frequência de coleta de dados, realizada a cada minuto e agregada por hora;
- Precisão dos dados exibidos, comparando com fontes externas;
- Relevância das variáveis meteorológicas para suas necessidades;
- Disposição em recomendar o sistema a outras pessoas.

Além dessas perguntas, os usuários foram incentivados a fornecer sugestões e críticas construtivas, permitindo a identificação de possíveis melhorias, tanto na usabilidade quanto na apresentação das informações.

O formulário utilizado para a coleta de dados pode ser consultado no Apêndice B.

Análise dos Resultados

Neste capítulo serão exibidos os resultados obtidos através das coletas de dados executadas no capítulo anterior. Além disso, serão também dispostas discussões sobre as informações construídas e eventuais tendências e indeterminações.

5.1 Resultado da Avaliação dos Sensores

Como mencionado no Capítulo 4, a avaliação dos sensores contidos na estação meteorológica, foi realizada com a comparação dos valores registrados com outros instrumentos e serviços de referência.

Considerando o grande volume de dados coletados e visando uma exibição mais concisa dos resultados obtidos, vislumbrou-se mais adequado a demonstração estatística das diferenças(erro) entre as aferições. Para tanto, foi selecionada a métrica **Erro Médio Absoluto (MAE)** que mede a média das diferenças absolutas entre os valores medidos e os valores de referência, sem considerar o sinal da diferença (positivo ou negativo). É uma métrica que oferece uma noção clara da magnitude média do erro, pelo fato de operar nas mesmas unidades das grandezas envolvidas, o que a torna fácil de interpretar (SANTOS, 2018).

Obedece à seguinte fórmula:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |X_{\text{estacao},i} - X_{\text{referencia},i}|$$

Onde:

- n é o número total de medições;
- $X_{\text{estacao},i}$ é o valor medido pela estação meteorológica na i -ésima observação;
- $X_{\text{referencia},i}$ é o valor medido pelo instrumento/serviço de referência na i -ésima observação;
- $|\cdot|$ representa o valor absoluto da diferença.

5.1.1 Sensor de Pressão Atmosférica BMP280

A Figura 25 traz o gráfico que contém os valores das diferenças das medições do sensor BMP280, instalado na Estação Meteorológica, com as realizadas pelo barômetro contido no *smartphone iPhone 13 Pro Max*.

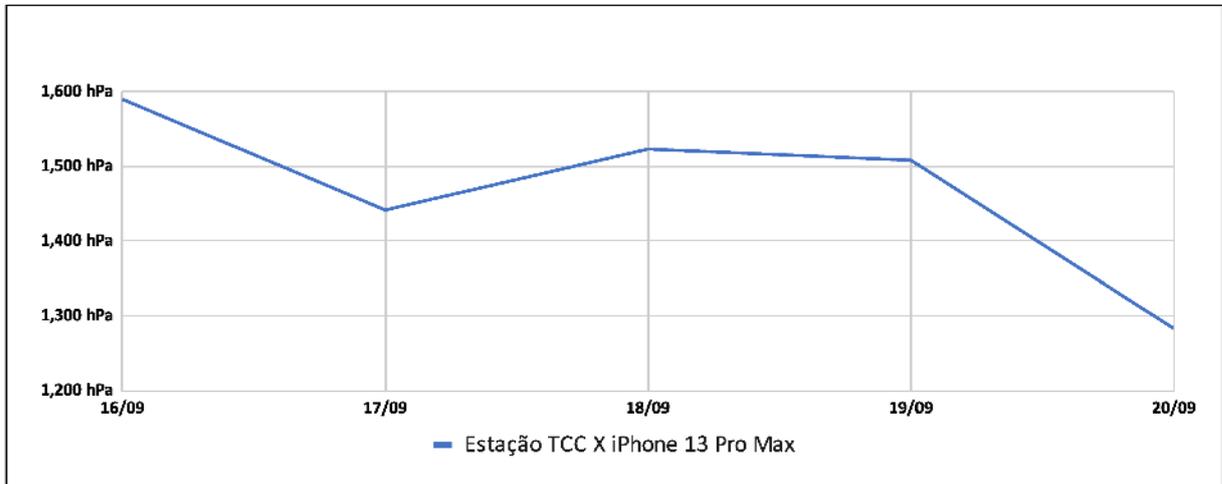


Figura 25 MAE - Pressão Atmosférica - Estação x Instrumentos

Os valores registrados acima refletem alta precisão da Estação quando comparada com um instrumento *in loco* para a grandeza de pressão atmosférica. Nenhuma média de erro ultrapassou a marca de 2 hPa, mesmo com os valores nominais chegando três dígitos.

Abaixo, a Figura 26 mostra a comparação com os *sites* de serviços meteorológicos.

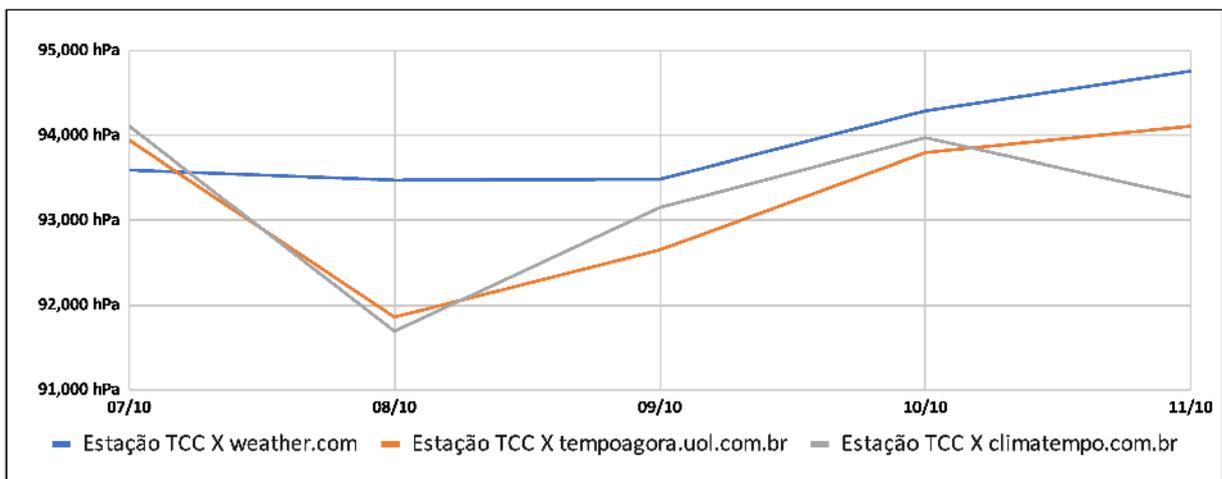


Figura 26 MAE - Pressão Atmosférica - Estação x Sites

Todos os *sites* mostraram valores acima dos registrados pela Estação. A diferença permaneceu estável entre 90 e 95 hPa. Os valores encontrados nestes serviços externos no

período de avaliação estiveram sempre acima da marca de 1000 hPa, o que se aproxima consideravelmente do valor obtido ao nível do mar (1013,25 hPa) (AHRENS, 2009).

Ahrens (2009) traz uma simplificação da Equação Barométrica que roga que a cada 100 metros de altitudes ganhos há uma queda de cerca 12 hPa. Considerando que a região de Monte Carmelo fica a aproximadamente de 800 metros acima do nível do mar, pelo padrão matemático supracitado, teria então 96 hPa ($800 \div 100 \times 12$) a menos do que o nível do mar, resultado em um valor médio aproximado de 917,25 hPa ($1013,25 - 96$). O resultado destes cálculos se aproxima mais do que foi obtido do sensor BMP280 do que os valores constantes nos *sites* elencados.

Diante do exposto, é possível inferir que a Estação Meteorológica apresenta precisão superior em comparação às suas referências online. É provável que as medições realizadas pelos *sites* especializados sejam baseadas em cálculos de interpolação e estimativa, o que justificaria a discrepância.

5.1.2 Sensor de Temperatura e Umidade SHT20

O sensor SHT20 é responsável pelas medidas de temperatura e umidade na Estação Meteorológica confeccionada neste trabalho. Em seguida serão tratadas individualmente as análises de cada uma destas grandezas.

5.1.2.1 Análise da Temperatura

A Figura 27 expõe o gráfico de Erro Médio Absoluto da Temperatura aferida na Estação Meteorológica em comparação a dois instrumentos diferentes: o termômetro de mercúrio e o dispositivo GM1362.

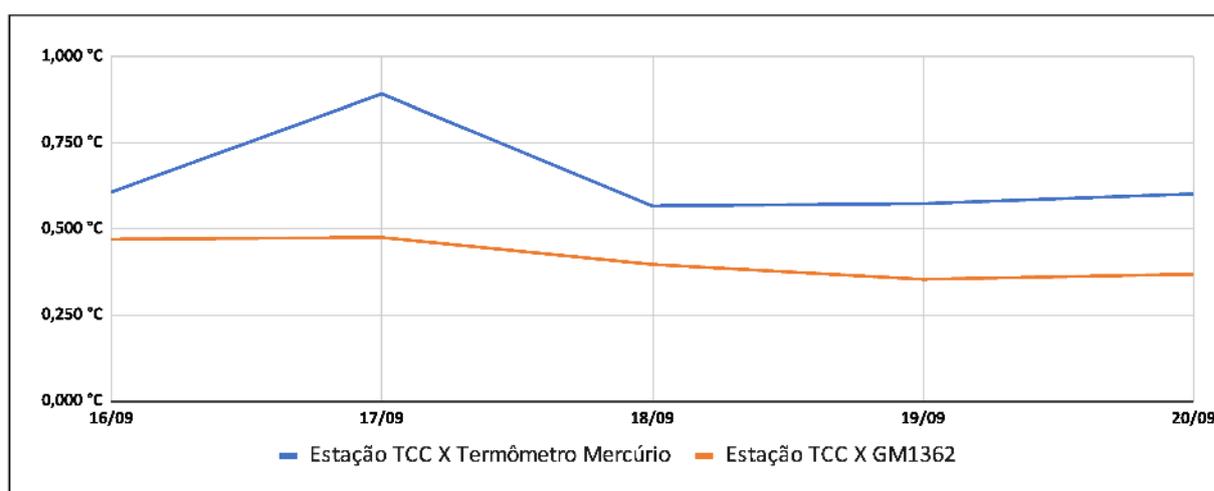


Figura 27 – MAE - Temperatura - Estação x Instrumentos

O gráfico acima revela que houve uma diferença muito pequena entre as aferições da estação e os sensores de referência. Não foram registrados valores de erros maiores do que 1°C para este experimento em seu período de observação.

Cabe salientar que, conforme mencionado no Capítulo 3, nas especificações do sensor SHT20, a margem de erro de suas medições de temperatura é de $\pm 1^\circ\text{C}$. Sendo assim, é possível afirmar que, tecnicamente, não houve diferença/erro entre as medições dos três dispositivos por estarem todas dentro da tolerância informada.

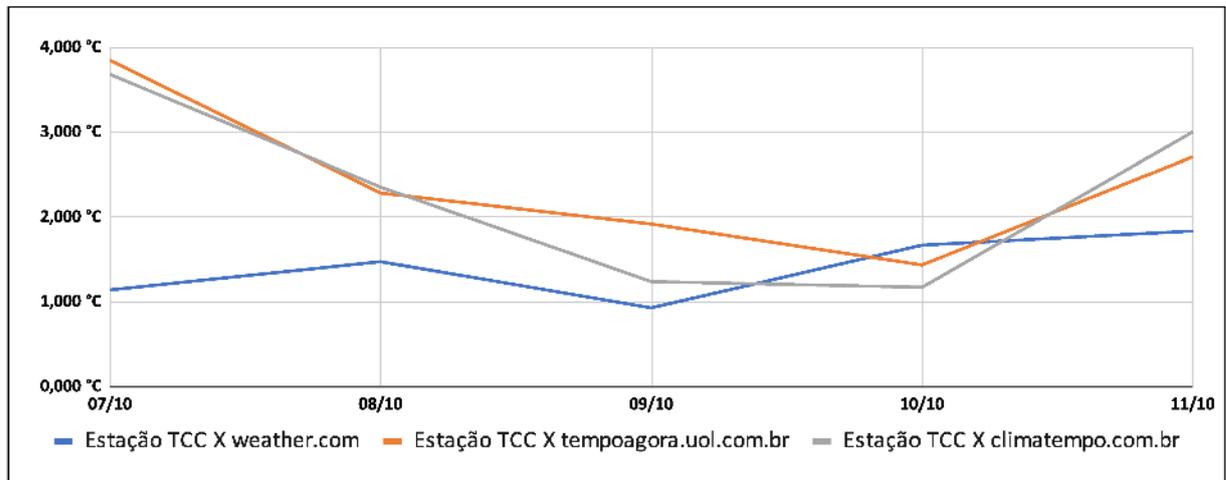


Figura 28 – MAE - Temperatura - Estação x Sites

O gráfico constante na Figura 28 mostra que na comparação dos valores obtidos na estação e os oriundos dos *sites* selecionados, alguns dias trouxeram valores de erros mais importantes, se aproximando de 4°C. Houve destaque para *weather.com* que se aproximou mais das medidas do sensor SHT20, enquanto os outros dois serviços se distanciaram similarmente.

Em uma análise mais aprofundada do comportamento do gráfico da Figura 28, percebe-se que em 07/10/2024 foram anotados maiores erros. Nesta data foi registrada a maior amplitude térmica do período, com máxima de 38,03°C e mínima de 21,32°C (aferidos pela Estação Meteorológica deste trabalho). Observou-se que em dias com estas características, principalmente nos momentos próximos aos valores máximos, a discrepância entre as referências se evidenciou mais.

5.1.2.2 Análise da Umidade

A seguir, outra comparação utilizando o dispositivo digital GM1362, desta vez para umidade, conforme gráfico abaixo contido na Figura 29.

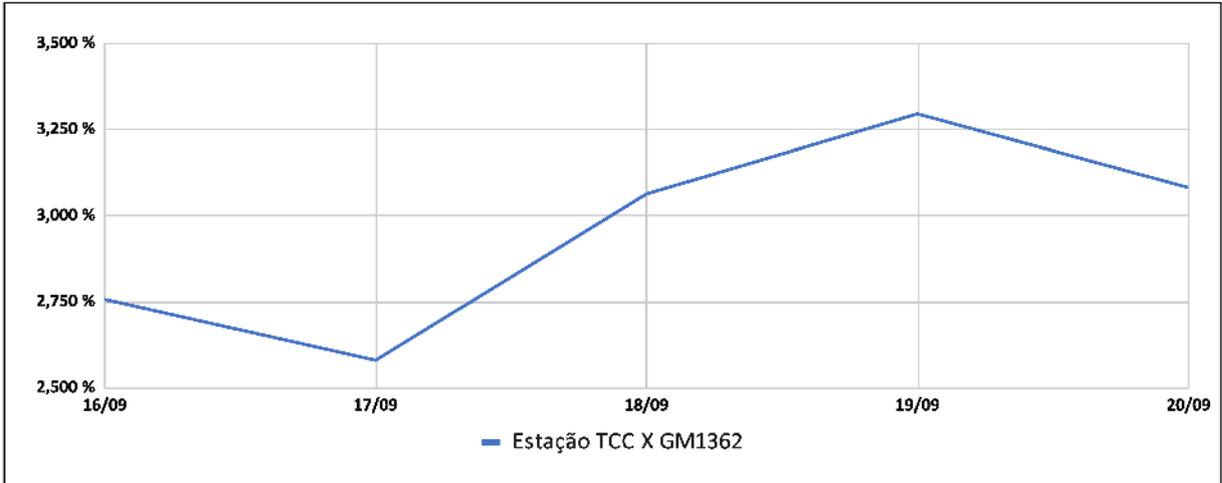


Figura 29 MAE - Umidade - Estação x Instrumentos

Nas especificações do sensor SIIT20, contido no Capítulo 3, a margem de erro informada pelo fabricante para as medições de umidade é de $\pm 5\%$. Como não houve registro de erros que superaram este valor no gráfico da Figura 29, a Estação Meteorológica novamente se mostrou precisa e livre de erros substanciais quando comparada a instrumentos físicos *in loco*.

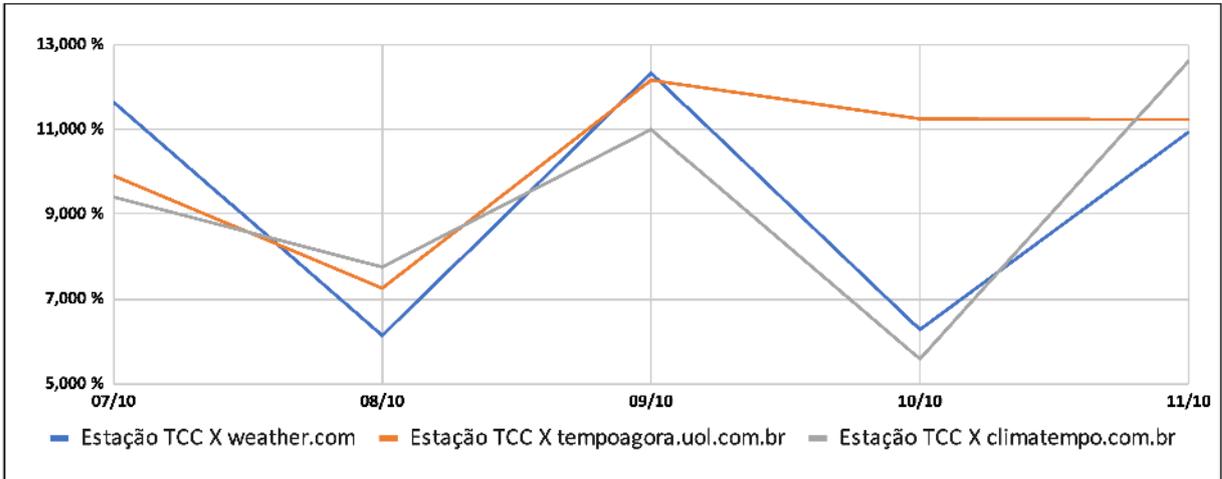


Figura 30 – MAE - Umidade - Estação x Sites

Na comparação contida na Figura 30, os valores dos MAEs ficaram entre 6 e 13%. Nos dias 09, 10 e 11 foi registrado precipitação de chuva, porém isto não afetou a comparação de forma substancial.

5.1.3 Pluviômetro de Bâscula

A análise comparativa entre as quatro estações meteorológicas automáticas, no período de 07/10/2024 a 12/10/2024, revelou variações significativas nos volumes de precipitação

registrados. Essas variações são justificadas pela diferença de localização geográfica de cada estação, que se encontram em pontos distintos da cidade, o que naturalmente influencia nos dados de chuva coletados.

Na Figura 31, que ilustra a distribuição geográfica das estações, foram utilizadas as marcações “A”, “B”, “C” e “D” para simbolizar a localização de cada estação meteorológica. Abaixo, estão detalhadas as correspondências:

- ❑ **A:** Estação meteorológica da Cooperativa Regional de Cafeicultores em Guaxupé (COOXUPÉ);
- ❑ **B:** Estação meteorológica do Centro de Inteligência em Cultivos Irrigados (CinCi);
- ❑ **C:** Estação meteorológica do Centro Nacional de Monitoramento e Alertas de Desastres Naturais (CEMADEN);
- ❑ **D:** Estação meteorológica desenvolvida neste trabalho de conclusão de curso.

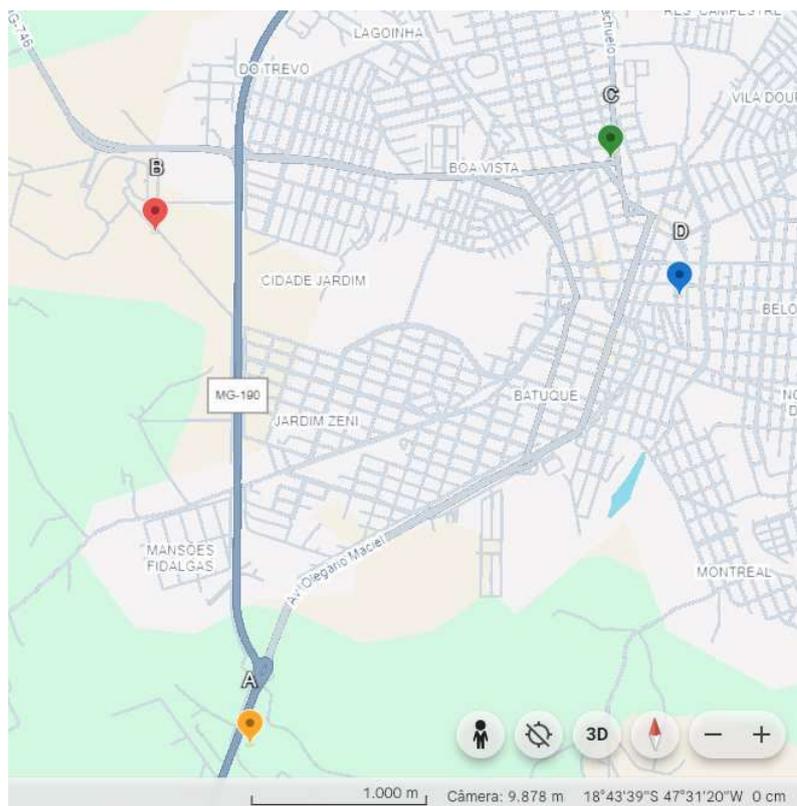


Figura 31 – Localização das Estações Meteorológicas

Fonte: <<https://earth.google.com/>>

O gráfico presente na Figura 32, ilustra visualmente os volumes diários de precipitação para cada uma das estações, facilitando a comparação e a análise dos dados.

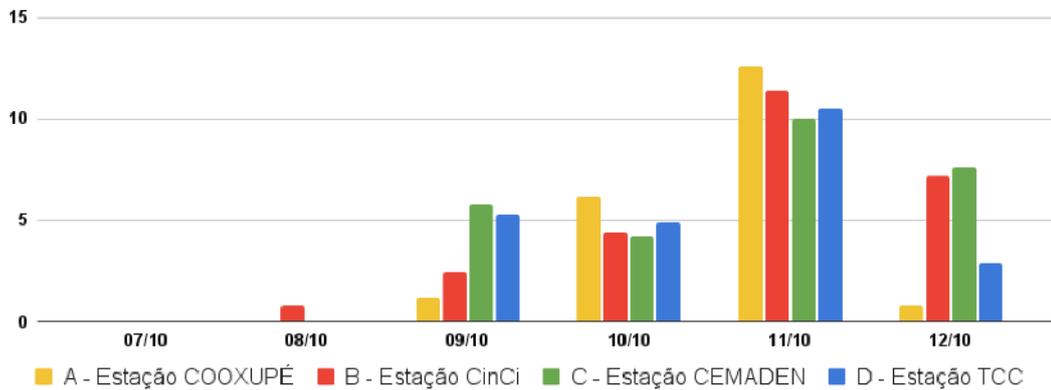


Figura 32 – Precipitação das Estações Meteorológicas

Ao longo dos dias analisados, foram observadas diferenças nos volumes de chuva registrados entre as estações, o que reforça a importância de considerar a distribuição espacial para entender as variações meteorológicas. Abaixo está uma análise diária dos dados de precipitação comparados:

- ❑ **07/10/2024:** Nenhuma das estações registrou precipitação, indicando ausência de chuva nesta região da cidade.
- ❑ **08/10/2024:** A estação do CinCi registrou 0,8 mm de chuva, enquanto as demais estações não registraram precipitação. Isso sugere uma chuva localizada no oeste do mapa, onde se encontra a estação do CinCi.
- ❑ **09/10/2024:** A estação do TCC registrou 5,3 mm de chuva enquanto a estação do CEMADEN registrou 5,8 mm. Isso indica uma maior concentração de chuva na região nordeste do mapa. De acordo com o <<https://www.google.com.br/maps>> as duas estações estão a uma distância de 870 metros.
- ❑ **10/10/2024:** Neste dia, a diferença entre os volumes de chuva registrados foi pequena entre as quatro estações. A estação da COOXUPÉ apresentou 6,2 mm, seguida pela estação do TCC, com 4,9 mm, e a estação do CinCi, com 4,4 mm. A CEMADEN registrou 4,2 mm. Esses dados indicam uma precipitação relativamente uniforme, sem grandes variações, sugerindo que a chuva foi bem distribuída entre as regiões monitoradas.
- ❑ **11/10/2024:** A estação da COOXUPÉ novamente apresentou o maior volume de chuva com 12,6 mm, enquanto a estação do CinCi apresentou 11,4 mm, seguida de perto pela estação do TCC, com 10,5 mm, e a CEMADEN, com 10 mm. Esses dados indicam uma precipitação semelhante na região do mapa, mas com uma ligeira concentração ao sul.

- **12/10/2024:** No último dia de coleta, a estação do CinCi e a CEMADEN registraram valores mais elevados (7,2 mm e 7,6 mm, respectivamente), em comparação com a estação do TCC, que registrou 2,9 mm, e a COOXUPÉ, com 0,8 mm, sugerindo uma maior precipitação na região norte e oeste do mapa.

5.2 Resultado da Avaliação do Sistema com Usuário

A avaliação com os usuários contou com 16 respostas, obtidas de alunos e professores dos cursos de Sistemas de Informação e Agronomia da Universidade Federal de Uberlândia, além de outros usuários da estação meteorológica. Segue abaixo a análise de cada pergunta.

1. O quão fácil foi acessar e utilizar o *website* da estação meteorológica?

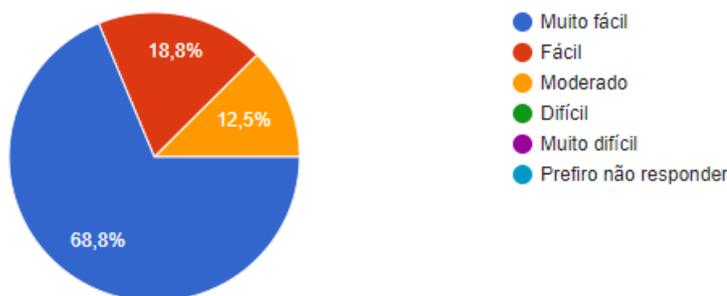


Figura 33 – Gráfico - Respostas da 1ª Pergunta do formulário de avaliação

O gráfico da Figura 33 mostra que a maioria dos usuários (68,8%) achou “muito fácil” acessar e utilizar o *website*, com 18,8% respondendo que foi “fácil” e 12,5% relatando que foi “moderado”. Não houve respostas que classificaram o acesso como difícil. Isso indica que o *site* foi bem projetado em termos de acessibilidade e facilidade de uso, proporcionando uma boa experiência desde o primeiro contato.

2. A interface gráfica do *site* é visualmente agradável e fácil de entender?

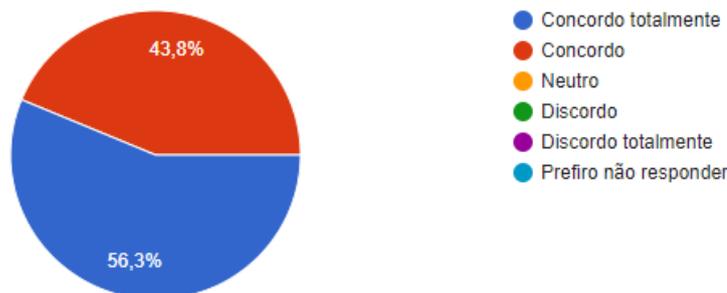


Figura 34 – Gráfico - Respostas da 2ª Pergunta do formulário de avaliação

O gráfico da Figura 34 mostra que, dos 16 participantes, 56,3% “concordaram totalmente” que a interface é agradável e 43,8% “concordaram”. Não houve respostas neutras ou negativas. Isso sugere que a interface, visualmente clara e bem organizada, foi bem recebida pelos usuários, cumprindo seu papel de facilitar a navegação e compreensão das informações.

3. A funcionalidade de exibir os valores máximos e mínimos de temperatura, umidade, pressão atmosférica, além do acumulado de precipitação, é útil para você?

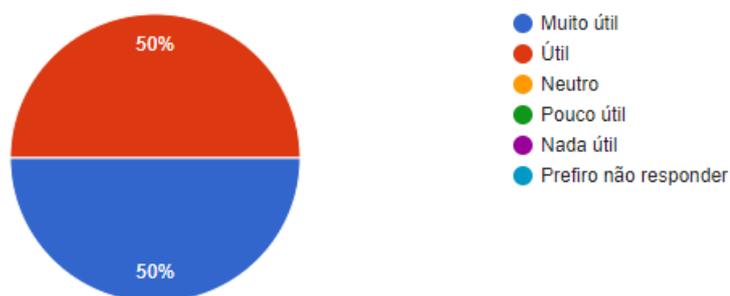


Figura 35 – Gráfico - Respostas da 3ª Pergunta do formulário de avaliação

O gráfico da Figura 35 mostra que metade dos respondentes considerou essa funcionalidade “muito útil” e os outros 50% acharam “útil”. Não houve respostas neutras ou negativas, o que reforça a importância dessa funcionalidade para os usuários, ajudando-os a compreender melhor as condições meteorológicas ao longo do tempo.

4. A apresentação dos dados em forma de gráficos de linha facilitou a compreensão das informações?

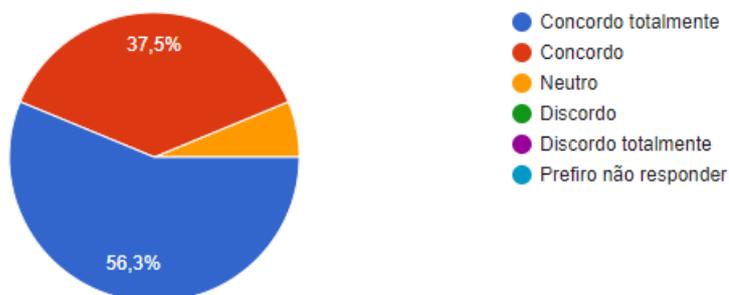


Figura 36 – Gráfico - Respostas da 4ª Pergunta do formulário de avaliação

O gráfico da Figura 36 mostra que, dos usuários, 56,3% “concordaram totalmente” que a apresentação dos dados em gráficos de linha facilitou a compreensão, enquanto

37,5% “concordaram” e apenas 6,3% foram “neutro”. Isso confirma que os gráficos são uma ferramenta eficaz para visualizar e interpretar as variações meteorológicas, tornando as informações acessíveis e claras.

5. A coleta dos dados a cada minuto e o agregado a cada hora é suficiente para suas demandas?

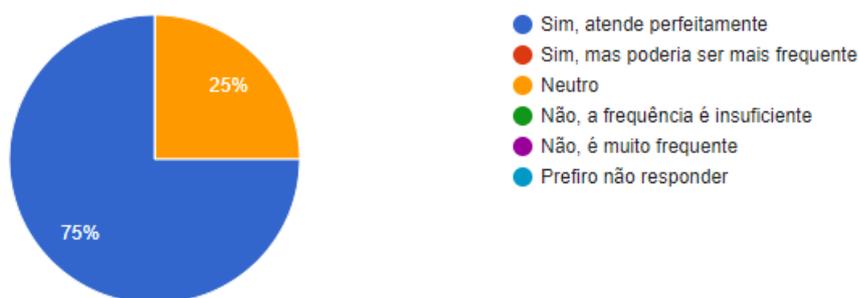


Figura 37 – Gráfico - Respostas da 5ª Pergunta do formulário de avaliação

O gráfico da Figura 37 mostra que para 75% usuários, a frequência de coleta de dados é suficiente, atendendo perfeitamente suas demandas, enquanto 25% se mantiveram neutros. Não houve respostas negativas. A frequência de coleta e agregação por hora parece ser bem adequada para a maioria dos casos de uso prático, garantindo um equilíbrio entre precisão e volume de dados.

6. Como você avalia a precisão dos dados exibidos na estação meteorológica?

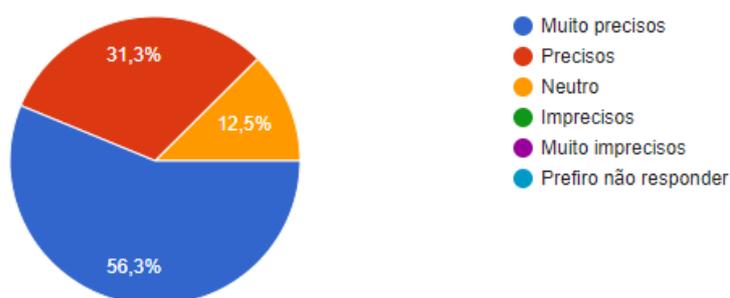


Figura 38 – Gráfico - Respostas da 6ª Pergunta do formulário de avaliação

O gráfico da Figura 38 mostra que 56,3% dos usuários classificaram os dados como “muito precisos” e 31,3% como “precisos”, com 12,5% respondendo de forma neutra. Não houve avaliações negativas. Isso indica que os sensores da estação meteorológica estão fornecendo dados confiáveis e comparáveis a fontes externas, o que aumenta a confiança no sistema.

7. A possibilidade de consultar os dados meteorológicos dos últimos 15 dias atende as suas necessidades?

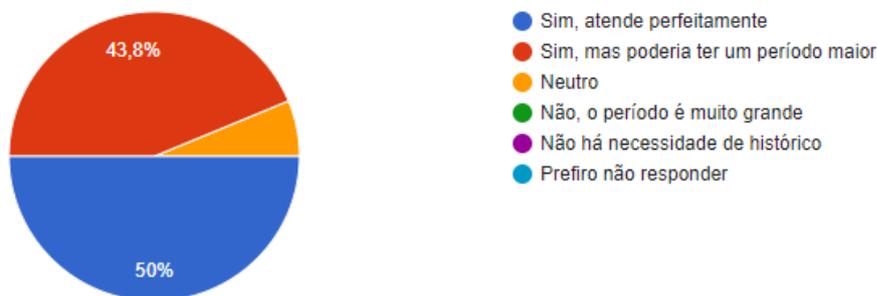


Figura 39 – Gráfico - Respostas da 7ª Pergunta do formulário de avaliação

O gráfico da Figura 39 mostra que metade usuários afirmaram que o histórico de 15 dias atende perfeitamente, enquanto 43,8% sugeriram que poderia haver um período maior. Apenas 6,3% foram neutros, sem respostas negativas. Embora a funcionalidade de histórico atenda à maioria, há uma demanda para estender o período disponível, o que pode ser uma melhoria futura a ser considerada.

8. Quais variáveis meteorológicas são relevantes para você?

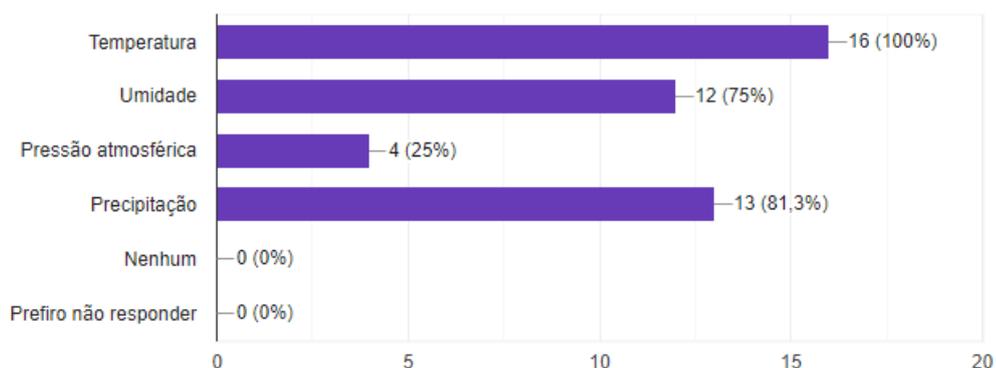


Figura 40 – Gráfico - Respostas da 8ª Pergunta do formulário de avaliação

O gráfico da Figura 40 mostra que a variável “Temperatura” foi considerada relevante por todos os 16 respondentes, seguida por “Precipitação”, selecionada por 13, “Umidade” por 12, e “Pressão atmosférica” por 4. Isso demonstra que as variáveis de temperatura e precipitação são as mais importantes para os usuários, enquanto a pressão atmosférica tem menor relevância.

9. Você recomendaria o uso desta estação meteorológica para outra pessoa?

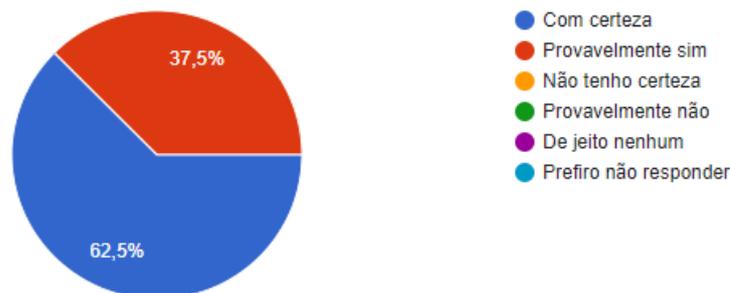


Figura 41 – Gráfico - Respostas da 9ª Pergunta do formulário de avaliação

O gráfico da Figura 41 mostra que a maioria dos respondentes (62,5%) afirmaram que “com certeza” recomendaria a estação, enquanto 37,5% “provavelmente” a recomendariam. Não houve respostas negativas. Esse resultado demonstra uma alta aceitação e satisfação com o sistema, indicando que a estação meteorológica atendeu ou superou as expectativas dos usuários.

10. **Por favor, deixe aqui suas críticas ou sugestões para melhorar este projeto, como a usabilidade, a qualidade das informações, a interface da estação meteorológica, ou qualquer outro aspecto que considere relevante.**

Os comentários abertos ofereceram diversas sugestões construtivas. Os usuários sugeriram a inclusão de uma funcionalidade para visualizar dados históricos de períodos mais longos em uma única tela, a possibilidade de exportar os dados, e a comparação dos dados da estação meteorológica com informações de fontes externas como INMET, Climatempo e *Weather*. Essas sugestões indicam oportunidades de melhoria para o sistema, especialmente na oferta de ferramentas mais avançadas de análise e comparação de dados meteorológicos.

Conclusão

Este capítulo tem como objetivo concluir o desenvolvimento do trabalho, destacando como os objetivos propostos foram alcançados e as principais contribuições trazidas pelo projeto.

Os objetivos específicos foram atingidos de maneira satisfatória. Foi investigada a precisão dos sensores BMP280 (Pressão atmosférica), SHT20 (temperatura e umidade) e do pluviômetro (precipitação), comparando seus dados com instrumentos de referência e estações meteorológicas. A análise revelou uma boa acurácia, validando o uso desses sensores no contexto de monitoramento climático local.

Além disso, foi analisada a eficácia da plataforma Arduino em um sistema de monitoramento meteorológico de relativo baixo custo. O uso de tecnologias de código aberto, como o *Spring Boot* e *React*, se mostrou eficiente para a implementação de uma solução acessível e escalável, que atende às necessidades de usuários da região de Monte Carmelo com informações precisas em tempo real.

Foi também possível comparar os dados da estação com informações de serviços meteorológicos *online*. Esta comparação demonstrou que para algumas grandezas aferidas, com destaque para pressão atmosférica e umidade, houve uma importante discrepância entre os dados coletados. Conforme análise destes resultados, verificou-se que os valores disponíveis em *sites* ficaram mais distantes daqueles medidos tanto pela estação meteorológica produzida por este TCC quanto também pelos sensores de referência. Isso demonstra um maior nível de confiabilidade às estações e o quanto estas são importantes para o fornecimento de dados mais precisos.

A análise mostrou que estações localizadas em diferentes pontos da cidade de Monte Carmelo trouxeram diferentes valores de medições. Isso demonstrou o dinamismo do clima e como em curtas distâncias podem existir variações importantes. Portanto, a existência de instalações meteorológicas distribuídas pela cidade é algo positivo e que certamente confere ao usuário demandante o resultado mais preciso possível para a região de interesse.

Em suma, os resultados foram positivos, confirmando a viabilidade do projeto. Por fim, o *feedback* dos usuários, coletado por meio de um formulário, avaliou a usabilidade

e a interface do sistema, identificando melhorias para futuras versões e revelando que o *Webclient* atendeu bem às expectativas.

6.1 Principais Contribuições

O trabalho contribuiu de maneira significativa para o desenvolvimento de uma solução local para monitoramento climático em Monte Carmelo, proporcionando aos usuários acesso a dados meteorológicos em tempo real por meio de um *site* acessível. A arquitetura modular e flexível do sistema também abre espaço para futuras melhorias e adaptações por outros desenvolvedores e alunos.

As contribuições principais incluem:

- ❑ A criação de um *site* onde é possível acessar dados meteorológicos detalhados e precisos da região de Monte Carmelo, beneficiando a agricultura, a pesquisa local e o cotidiano dos cidadãos.
- ❑ Uma arquitetura de sistema que pode ser replicada e melhorada por outros alunos e profissionais, permitindo expansões como a adição de novos sensores ou funcionalidades.
- ❑ Validação experimental do sistema, confirmando que a hipótese de desenvolver uma solução local, acessível e precisa foi alcançada.

6.2 Trabalhos Futuros

Algumas limitações foram identificadas e podem ser melhoradas em versões futuras. A exportação de dados meteorológicos é uma funcionalidade solicitada por profissionais, o que permitiria análises mais detalhadas e armazenamento de longo prazo. Outra sugestão é a implementação de gráficos mensais, que proporcionariam uma visão mais ampla das variações climáticas.

Além disso, é possível expandir o projeto para incluir novas estações meteorológicas em diferentes pontos de Monte Carmelo, o que permitiria uma análise mais rica das variações microclimáticas na região. O *feedback* dos usuários também sugeriu melhorias na interface, como a comparação com dados de serviços meteorológicos externos, que pode ser incorporada em futuras atualizações.

Referências

- ADAFRUIT. **Adafruit BMP280 Library**. 2024. Acesso em: 29 set. 2024. Disponível em: <<https://www.arduino.cc/reference/en/libraries/adafruit-bmp280-library/>>. Citado na página 33.
- AG, S. **Humidity and Temperature Sensor SHT20 Datasheet**. 2020. Acessado em: 30 set. 2024. Disponível em: <https://sensirion.com/media/documents/CCDE1377/635000A2/Sensirion_Datasheet_Humidity_Sensor_SHT20.pdf>. Citado 2 vezes nas páginas 28 e 29.
- Agrosmart. **Como funciona uma estação meteorológica e qual a sua importância para a agricultura**. 2023. Acessado em: 17 set. 2024. Disponível em: <<https://agrosmart.com.br/blog/estacao-meteorologica-funciona-importancia-agricultura/>>. Citado na página 17.
- AHMAD, L. et al. Automatic weather station. In: _____. **Experimental Agrometeorology: A Practical Manual**. Springer, 2017. cap. 12, p. 83–87. Disponível em: <https://doi.org/10.1007/978-3-319-69185-5_12>. Citado na página 17.
- AHRENS, C. **Meteorology Today: An Introduction to Weather, Climate, and the Environment**. Brooks/Cole, CengageLearning, 2009. (International student edition). ISBN 9780495564935. Disponível em: <<https://books.google.com.br/books?id=SrkRAQAIAAJ>>. Citado na página 53.
- Amazon Web Services. **O que é IoT (Internet das Coisas)?** 2024. Acessado em: 17 set. 2024. Disponível em: <<https://aws.amazon.com/pt/what-is/iot/>>. Citado na página 22.
- Apple Support. **iPhone 13 Pro Max: Especificações e Recursos do Barômetro**. 2024. Acessado em: 03 out. 2024. Disponível em: <<https://support.apple.com/pt-br/111870>>. Citado na página 47.
- ARDUINO. **Arduino Documentation**. 2024. Acesso em: 19 set. 2024. Disponível em: <<https://docs.arduino.cc/learn/starting-guide/whats-arduino/>>. Citado 3 vezes nas páginas 13, 18 e 19.
- _____. **Arduino Hardware**. 2024. Acesso em: 23 set. 2024. Disponível em: <<https://www.arduino.cc/en/hardware#boards/>>. Citado na página 25.

- _____. **ArduinoHttpClient**. 2024. Acesso em: 29 set. 2024. Disponível em: <<https://www.arduino.cc/reference/en/libraries/arduinohttpclient/>>. Citado na página 33.
- _____. **Inter-Integrated Circuit (I2C) Protocol**. 2024. Acesso em: 23 set. 2024. Disponível em: <<https://docs.arduino.cc/learn/communication/wire/>>. Citado 2 vezes nas páginas 25 e 26.
- _____. **WiFi**. 2024. Acesso em: 29 set. 2024. Disponível em: <<https://www.arduino.cc/reference/en/libraries/wifi/>>. Citado na página 33.
- BAPTISTA, J. C. R. O estado da arte do estudo das conseqüências do aquecimento global para o brasil. **Revista de informação legislativa**, v. 44, n. 174, p. 273–278, 2007. Citado na página 13.
- BENETECH. **Digital Thermometer GM1362**. 2024. <<http://www.benetechco.net/en/products/gm1362.html>>. Acessado em: 6 de out. de 2024. Citado na página 47.
- BLANCHON, B. **ArduinoJson**. 2024. Acesso em: 29 set. 2024. Disponível em: <<https://www.arduino.cc/reference/en/libraries/arduinojson/>>. Citado na página 33.
- BOAGLIO, F. **MongoDB: construa novas aplicações com novas tecnologias**. [S.l.]: Editora Casa do Código, 2020. Citado na página 21.
- BRABANDER, F. de. **LiquidCrystal I2C**. 2024. Acesso em: 29 set. 2024. Disponível em: <<https://www.arduino.cc/reference/en/libraries/liquidcrystal-i2c/>>. Citado na página 33.
- BRASILEIRA, D. de Controle do E. A. F. A. **QNE - Pressão Padrão ao Nível Médio do Mar**. 2024. Acesso em: 29 set. 2024. Disponível em: <<https://www.decea.mil.br/index.cfm?i=utilidades&p=glossario&single=2314#:~:text=C%C3%B3digo%20que%20corresponde%20%C3%A0%20press%C3%A3o,conforme%20venham%20a%20ser%20observadas.>> Citado na página 36.
- DEINUM, M. et al. Spring framework fundamentals. **Pro Spring MVC with WebFlux**, 2021. Citado na página 21.
- DFROBOT. **Dfrobot sht20**. 2024. Acesso em: 29 set. 2024. Disponível em: <https://www.arduino.cc/reference/en/libraries/dfrobot_sht20/>. Citado na página 33.
- DIAS, R. d. P.; ROMANO, Y. S. d. S. Desenvolvimento de estação meteorológica utilizando arduino. **Repositorio Institucional RINCÓN - Academia de Ciencias de Bolivia**, 2018. Citado na página 23.
- Facebook, Inc. **React - A JavaScript library for building user interfaces**. 2024. Acessado em: 17 set. 2024. Disponível em: <<https://github.com/facebook/react>>. Citado 2 vezes nas páginas 21 e 22.
- FINHOLDT, G. et al. Desenvolvimento de uma estação meteorológica automática para manejo de irrigação. **Revista Engenharia na Agricultura-REVENG**, v. 19, n. 4, p. 313–328, 2011. Citado na página 23.

FLANAGAN, D. **JavaScript: o guia definitivo**. [S.l.]: Bookman Editora, 2012. Citado na página 42.

HOWS, D.; MEMBREY, P.; PLUGGE, E. **Introdução ao MongoDB**. [S.l.]: Novatec Editora, 2019. Citado na página 21.

IBAMA and ESBR - Santo Antônio Energia. **Relatório de Monitoramento Climatológico - Usina Hidrelétrica Santo Antônio (Rio Madeira)**. 2009. Acessado em: 17 set. 2024. Disponível em: <[https://licenciamento.ibama.gov.br/Hidreletricas/SantoAntonio\(RioMadeira\)-%202002001.000508_2008-99/Relatorios/POS%20LI/Relat%C3%B3rio%20n%C2%B0%208/Anexos%20Monitoramento%20Climatol%C3%B3gico/5.2/5.2%20NOVO_PBA_ESBR_SAE_IBAMA_30102009%5B1%5D.pdf](https://licenciamento.ibama.gov.br/Hidreletricas/SantoAntonio(RioMadeira)-%202002001.000508_2008-99/Relatorios/POS%20LI/Relat%C3%B3rio%20n%C2%B0%208/Anexos%20Monitoramento%20Climatol%C3%B3gico/5.2/5.2%20NOVO_PBA_ESBR_SAE_IBAMA_30102009%5B1%5D.pdf)>. Citado na página 17.

Instituto Nacional de Meteorologia (INMET). **Manual de Observações Meteorológicas**. 3^a. ed. [S.l.], 1999. Acesso em: 19 set. 2024. Disponível em: <<https://portal.inmet.gov.br/manual/manual-de-observa%C3%A7%C3%B5es-meteorol%C3%B3gicas>>. Citado 3 vezes nas páginas 17, 18 e 47.

_____. **Sobre Meteorologia**. 2024. Acessado em: 17 set. 2024. Disponível em: <<https://portal.inmet.gov.br/sobre-meteorologia>>. Citado 2 vezes nas páginas 16 e 17.

JUNIOR, E. A. G.; ROCHA, R. D.; MACIEL, R. de S. Desenvolvimento de api rest com spring boot. **Revista Científica do UniRios**, 2021. Acessado em: 17 set. 2024. Disponível em: <<https://www.publicacoes.unirios.edu.br/index.php/revistarios/article/view/102>>. Citado na página 20.

KUSUMA, H. A. et al. Sensor bmp280 statistical analysis for barometric pressure acquisition. In: IOP PUBLISHING. **IOP Conference Series: Earth and Environmental Science**. [S.l.], 2023. v. 1148, n. 1, p. 012008. Citado na página 27.

LIMA, L.; PETRICA, E. Protocolo http. **Viva o Linux - Artigos da Comunidade**, 2009. Acessado em: 17 set. 2024. Disponível em: <<https://www.vivaolinux.com.br/imagens/artigos/comunidade/Protocolo%20HTTP.pdf>>. Citado na página 20.

LUCKOW, D. H.; MELO, A. A. de. **Programação Java para a WEB**. [S.l.]: Novatec Editora, 2010. Citado 2 vezes nas páginas 20 e 21.

MakerHero. **Comparativo Arduino GIGA, Due e Mega: Qual é o melhor para o seu projeto?** 2024. Acessado em: 17 set. 2024. Disponível em: <<https://www.makehero.com/blog/comparativo-arduino-giga-due-mega/>>. Citado na página 20.

Michaelis. **Michaelis Moderno Dicionário da Língua Portuguesa**. Melhoramentos, 2024. Acesso em: 29 out. 2024. Disponível em: <<https://michaelis.uol.com.br/>>. Citado na página 26.

npm, Inc. **Node Package Manager (NPM) - A package manager for JavaScript**. 2024. Acessado em: 17 set. 2024. Disponível em: <<https://www.npmjs.com/>>. Citado na página 22.

Oracle. **Java Releases**. 2024. Acessado em: 17 set. 2024. Disponível em: <<https://www.java.com/releases/>>. Citado na página 21.

- SANTOS, G. M. M. dos. **Método de Previsão de Vendas e Estimativa de Reposição de Itens no Varejo da Moda**. Dissertação (Dissertação de Mestrado) — Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, abr. 2018. Aprovado em 26 de abril de 2018. Disponível em: <<https://www.pucrs.br>>. Citado na página 51.
- SENSORTEC, B. Digital pressure sensor. **May 7th**, 2015. Citado na página 27.
- SILVA, A. C. M. da et al. Estação meteorológica automática de baixo custo-low cost automatic meteorological station. **Multiverso: Revista Eletrônica do Campus Juiz de Fora-IF Sudeste MG**, v. 1, n. 1, p. 46–56, 2016. Citado na página 22.
- SOUZA, B. N. P. A. **Guia do Professor - Medindo a Chuva**. [S.l.], 2016. Acesso em: 14 out. 2024. Disponível em: <<http://www.matematica.seed.pr.gov.br/arquivos/File/manuais/medindoachuva.pdf>>. Citado na página 48.
- Stack Overflow. **Stack Overflow Developer Survey 2024: Most Popular Technologies**. 2024. Acessado em: 17 set. 2024. Disponível em: <<https://survey.stackoverflow.co/2024/technology/#most-popular-technologies>>. Citado na página 21.
- UsinaInfo. **Pluviômetro de Bâscula Digital Arduino para Estação Meteorológica PB10**. 2024. Acessado em: 03 out. 2024. Disponível em: <<https://www.usinainfo.com.br/estacao-meteorologica-arduino/pluviometro-de-bascula-digital-arduino-para-estacao-meteorologica-pb10-4637.html>>. Citado na página 30.

Apêndices

Termo de Consentimento Livre e Esclarecido (TCLE)

Termo de Consentimento Livre e Esclarecido (TCLE) está sendo obtido de forma virtual antes do início da sua participação na pesquisa e coleta de dados. Antes de concordar em participar da pesquisa, você pode entrar em contato por email com o aluno Jhonathan Gontijo Da Silva cujo email é o: jhonathan_gontijo@hotmail.com, para discutir as informações do estudo.

Você tem o tempo que for necessário para decidir se quer ou não participar da pesquisa (conforme item IV da Resolução nº 466/2012 ou Capítulo. III da Resolução nº 510/2016).

A data de início e término da coleta de dados é: 12 de Outubro de 2024 a 17 de Outubro de 2024.

Na sua participação, você responderá algumas perguntas. A pesquisa será feita através de um formulário do *Google*, sua participação será anônima, os resultados dessa pesquisas serão usados apenas para fins de pesquisa. A pesquisa é separada em 10 perguntas sobre a estação meteorológica proposta pela atual pesquisa. O tempo mínimo estimado é de, aproximadamente, 10 minutos.

Você tem o direito de não responder qualquer questão, sem necessidade de explicação ou justificativa para tal. No questionário haverá opções de preferência para não serem respondidas.

Você não terá nenhum gasto e nem ganho financeiro por participar na pesquisa.

Nós, pesquisadores, atenderemos as orientações das Resoluções nº 466/2012, Capítulo XI, Item XI.2: f e nº 510/2016, Capítulo VI, Art. 28: IV - manter os dados da pesquisa em arquivo, físico ou digital, sob sua guarda e responsabilidade, por um período mínimo de 5 (cinco) anos após o término da pesquisa.

Para minimizar alguns riscos do ambiente virtual, é importante que você tenha todo o cuidado com a segurança e privacidade do local quando realizar o acesso às etapas virtuais da pesquisa para que sejam garantidos o sigilo e a confidencialidade necessários.

Não haverá benefícios, o intuito da pesquisa é a obtenção de dados para finalizar a

pesquisa de graduação.

Havendo algum dano decorrente da pesquisa, você terá direito a solicitar indenização através das vias judiciais (Código Civil, Lei 10.406/2002, Artigos 927 a 954 e Resolução CNS nº 510 de 2016, Artigo 19).

Você é livre para deixar de participar da pesquisa a qualquer momento sem qualquer prejuízo ou coação. No questionário não será solicitado a identificação dos dados pessoais dos participantes, portanto esteja informado a impossibilidade de exclusão dos dados da pesquisa.

Em qualquer momento, caso tenha qualquer dúvida ou reclamação a respeito da pesquisa, você poderá entrar em contato com Jhonathan Gontijo Da Silva - (34) 99221-6644 ou Adriano Mendonça Rocha - adriano.rocha@ufu.br.

Você poderá também entrar em contato com o Comitê de Ética na Pesquisa com Seres Humanos – CEP, da Universidade Federal de Uberlândia, localizado na Av. João Naves de Ávila, nº 2121, bloco A, sala 224, campus Santa Mônica – Uberlândia/MG, 38408-100; pelo telefone (34) 3239-4131 ou pelo e-mail cep@propp.ufu.br. O CEP/UFU é um colegiado independente criado para defender os interesses dos participantes das pesquisas em sua integridade e dignidade e para contribuir para o desenvolvimento da pesquisa dentro de padrões éticos conforme resoluções do Conselho Nacional de Saúde.

Avaliação da interface e usabilidade da estação meteorológica

Formulário utilizado para a coleta de dados sobre a interface e usabilidade da estação meteorológica.

O quão fácil foi acessar e utilizar o *website* da estação meteorológica?

- Muito fácil
- Fácil
- Moderado
- Difícil
- Muito difícil
- Prefiro não responder

A interface gráfica do *site* é visualmente agradável e fácil de entender?

- Concordo totalmente
- Concordo
- Neutro
- Discordo
- Discordo totalmente
- Prefiro não responder

A funcionalidade de exibir os valores máximos e mínimos de temperatura, umidade, pressão atmosférica, além do acumulado de precipitação, é útil para você?

- Muito útil
- Útil
- Neutro
- Pouco útil
- Nada útil
- Prefiro não responder

A apresentação dos dados em forma de gráficos de linha facilitou a compreensão das informações?

- Concordo totalmente
- Concordo
- Neutro
- Discordo
- Discordo totalmente
- Prefiro não responder

A coleta dos dados a cada minuto e o agregado a cada hora é suficiente para suas demandas?

- Sim, atende perfeitamente
- Sim, mas poderia ser mais frequente
- Neutro
- Não, a frequência é insuficiente
- Não, é muito frequente
- Prefiro não responder

Como você avalia a precisão dos dados exibidos na estação meteorológica? Se possível, compare com informações externas.

- Muito precisos
- Precisos
- Neutro

- Imprecisos
- Muito imprecisos
- Prefiro não responder

A possibilidade de consultar os dados meteorológicos dos últimos 15 dias atende as suas necessidades?

- Sim, atende perfeitamente
- Sim, mas poderia ter um período maior
- Neutro
- Não, o período é muito grande
- Não há necessidade de histórico
- Prefiro não responder

Quais variáveis meteorológicas são relevantes para você? (Pode selecionar mais de uma opção)

- Temperatura
- Umidade
- Pressão atmosférica
- Precipitação
- Nenhum
- Prefiro não responder

Você recomendaria o uso desta estação meteorológica para outra pessoa?

- Com certeza
- Provavelmente sim
- Não tenho certeza
- Provavelmente não
- De jeito nenhum
- Prefiro não responder

Por favor, deixe aqui suas críticas ou sugestões para melhorar este projeto, como a usabilidade, a qualidade das informações, a interface da estação meteorológica, ou qualquer outro aspecto que considere relevante: