



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS ARARANGUÁ
CENTRO DE CIÊNCIAS, TECNOLOGIA E SAÚDE (CTS)
TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO

Jessé Emerim Schardosim

**Protótipo de sistema embarcado utilizando sensor ultrassônico para
monitoramento do nível de corpos d'água em tempo real e alerta de enchentes**

Araranguá

2026

Jessé Emerim Schardosim

Protótipo de sistema embarcado utilizando sensor ultrassônico para monitoramento do nível de corpos d'água em tempo real e alerta de enchentes

Trabalho de Conclusão de Curso submetido ao curso de Tecnologias da Informação e Comunicação do Campus Araranguá da Universidade Federal de Santa Catarina como requisito parcial para a obtenção do título de Bacharel em Tecnologias da Informação e Comunicação.

Orientador: Prof. Cristian Cechinel, Dr.
Coorientador: Prof. Fabrício Herpich, Dr.

Schardosim, Jessé Emerim

Protótipo de sistema embarcado utilizando sensor ultrassônico para monitoramento do nível de corpos d'água em tempo real e alerta de enchentes / Jessé Emerim Schardosim ; orientador, Cristian Cechinel, coorientador, Fabrício Herpich, 2026.

82 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Campus Araranguá, Graduação em Tecnologias da Informação e Comunicação, Araranguá, 2026.

Inclui referências.

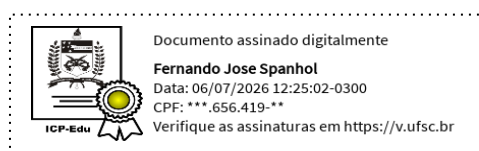
1. Tecnologias da Informação e Comunicação. 2. Sistemas embarcados. 3. Nível da água. 4. Sensor ultrassônico. 5. Enchentes. I. Cechinel, Cristian. II. Herpich, Fabrício. III. Universidade Federal de Santa Catarina. Graduação em Tecnologias da Informação e Comunicação. IV. Título.

Jessé Emerim Schardosim

**Protótipo de sistema embarcado utilizando sensor ultrassônico para monitoramento
do nível de corpos d'água em tempo real e alerta de enchentes**

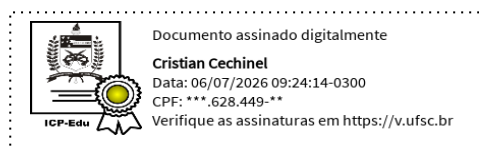
Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de Bacharel em Tecnologias da Informação e Comunicação e aprovado em sua forma final pelo Curso Tecnologias da Informação e Comunicação.

Araranguá, 19 de junho de 2026.

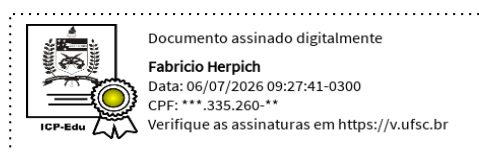


Prof. Fernando José Spanhol, Dr.
Coordenador do Curso

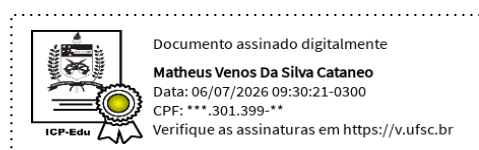
Banca examinadora



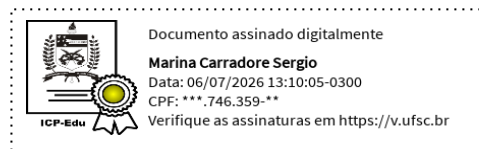
Prof. Cristian Cechinel, Dr.
Orientador



Prof. Fabrício Herpich, Dr.
Coorientador



Prof. Matheus Venos Da Silva Catâneo
Universidade Federal de Santa Catarina



Prof.^a Marina Carradore Sérgio, Dr.^a
Universidade Federal de Santa Catarina

Araranguá, 2026.

AGRADECIMENTOS

Agradeço ao professor e orientador Cristian Cechinel que possibilitou a apresentação e finalização do meu trabalho de conclusão de curso (TCC). Agradeço também ao professor e coorientador Fabrício Herpich que acompanhou e sanou minhas dúvidas em relação ao desenvolvimento do TCC. E agradeço também ao professor Matheus Venos Da Silva Catâneo e à professora Marina Carradore Sérgio que prontamente aceitaram participar da banca do meu TCC.

A dúvida é o princípio da sabedoria.

Aristóteles

RESUMO

As enchentes representam um problema recorrente em diversos municípios brasileiros, causando impactos sociais, econômicos e ambientais significativos. A ausência de sistemas automatizados e de baixo custo para monitoramento contínuo de corpos d'água dificulta a emissão antecipada de alertas à população. Este trabalho de conclusão de curso tem o objetivo de desenvolver um protótipo de sistema embarcado de baixo custo utilizando sensor ultrassônico para medir o nível de corpos d'água em tempo real com o intuito de mitigar o impacto de enchentes. Foi realizada uma pesquisa bibliográfica para embasamento técnico e científico da solução proposta e adotou-se a metodologia *Design Science Research Methodology* (DSRM) que guiou a pesquisa através de uma estrutura sistemática para desenvolvimento do artefato. Primeiramente foi identificado o problema, que no caso são as enchentes que ocorrem em Araranguá e região. Depois foram definidos os objetivos, que inclui o desenvolvimento do protótipo com base no conhecimento e estado da arte sobre o problema e soluções já apresentadas. Depois desenvolveu-se o protótipo em ambiente controlado. E então demonstrou-se o seu funcionamento avaliando os dados de leitura do mesmo utilizando-se de métricas para determinar sua acurácia. Foi desenvolvido também um website para monitoramento do mesmo. E um *listener* assíncrono que é um *software* que funciona no *backend* que fica escutando alterações na *Real-Time Database* do *Firebase* nos valores atuais das sondas dos usuários, grava no respectivo gráfico de onda do website de hora em hora os valores lidos pela sonda e envia um e-mail ao usuário quando o nível da água ultrapassar determinado limite definido por ele. Foi feita uma prova de conceito para verificar se o protótipo funciona. E também medido e registrado os valores lidos pela sonda para calcular a acurácia do protótipo, neste caso foram utilizadas 12 medições. Foi usado o erro absoluto médio, o erro de viés médio e o erro relativo médio para fins de cálculo da precisão do protótipo. Além disso, foi comparada a precisão do protótipo desenvolvido (erro relativo médio) na presente pesquisa aplicada com a de um estudo identificado na revisão bibliográfica. O protótipo apresentou erro relativo médio de 2,96%, demonstrando precisão compatível com aplicações de monitoramento de nível da água para fins de alerta de enchentes. Os resultados indicam que a solução proposta é viável para aplicações de monitoramento hidrológico de baixo custo e possui potencial para utilização em sistemas de alerta precoce de enchentes. Por fim, o protótipo passou na prova de conceito e a presente pesquisa aplicada conseguiu alcançar com êxito os objetivos propostos.

Palavras-chave: sistemas embarcados; nível da água; sensor ultrassônico; enchentes.

ABSTRACT

Floods represent a recurring problem in several Brazilian municipalities, causing significant social, economic, and environmental impacts. The lack of low-cost, automated systems for the continuous monitoring of water bodies hinders the early issuance of warnings to the population. This undergraduate monograph aims to develop a low-cost embedded system prototype using an ultrasonic sensor to measure water body levels in real time in order to mitigate flood impacts. A bibliographic research was conducted to provide a technical and scientific foundation for the proposed solution, and the Design Science Research Methodology (DSRM) was adopted, guiding the research through a structured research model for the development of the artifact. First, the problem was identified, which in this case refers to the floods occurring in Araranguá and its surrounding region. Subsequently, the objectives were defined, which included developing the prototype based on the knowledge and state-of-the-art regarding the problem and previously proposed solutions. Next, the prototype was developed in a controlled environment. Its functionality was then demonstrated by evaluating its reading data using metrics to determine its accuracy. A website was also developed to monitor the system. Furthermore, an asynchronous listener was implemented — a backend software that listens for changes in the users' probe current values within the Firebase Real-Time Database, records the probe readings hourly on the website's respective waveform graph, and sends an email to the user when the water level exceeds a specific limit defined by them. A proof of concept was conducted to verify the prototype's functionality. Additionally, the values read by the probe were measured and recorded to calculate the prototype's accuracy; for this purpose, 12 measurements were used. The mean absolute error, mean bias error, and mean relative error were used to calculate the prototype's precision. In addition, the precision of the developed prototype (mean relative error) in this applied research was compared with that of a study identified in the literature review. The prototype presented a mean relative error of 2.96%, demonstrating a precision compatible with water level monitoring applications for flood warning purposes. The results indicate that the proposed solution is viable for low-cost hydrological monitoring applications and has the potential to be used in early flood warning systems. Finally, the prototype passed the proof of concept, and this applied research successfully achieved its proposed objectives.

Keywords: embedded systems; water level; ultrasonic sensor; floods.

LISTA DE FIGURAS

Figura 1 – Esquema do Sensor Ultrassônico para medição do nível da água	27
Figura 2 – Placa UNO R4 WiFi.....	31
Figura 3 – Módulo ultrassônico AJ-SR04M	32
Figura 4 – Etapas da DSRM.....	43
Figura 5 – Sensor ultrassônico instalado	45
Figura 6 – Regras de acesso ao banco de dados	46
Figura 7 – Estrutura do banco de dados	47
Figura 8 – Documentação da API Brevo	48
Figura 9 – Biblioteca: FirebaseClient by Mobizt	49
Figura 10 – Biblioteca: ESP_SSLClient by Mobizt	49
Figura 11 – Circuito do protótipo	50
Figura 12 – Uno R4 WiFi Pinout.....	51
Figura 13 – Uno R4 WiFi Pinout: Legenda.....	51
Figura 14 – Histórico fictício de nível da água da sonda	53
Figura 15 – Dashboard do website com dados fictícios	54
Figura 16 – Página inicial do website	55
Figura 17 – Página de login do website	56
Figura 18 – Dashboard do website com dados legítimos lidos pelo sensor	57
Figura 19 – Circuito do Protótipo montado na protoboard.....	58
Figura 20 – Primeira medição do nível da água com a régua: 6 cm	59
Figura 21 – Primeira medição do nível da água com o sensor: 6.1 cm.....	60
Figura 22 – Hora, temperatura e pressão atmosférica da primeira medição.....	60
Figura 23 – Primeiro valor lido pelo sensor na Tabela 2 (listener)	62
Figura 24 – Terceiro valor lido pelo sensor na Tabela 2 (listener).....	62
Figura 25 – E-mail recebido pelo usuário do terceiro valor lido (Tabela 2)	62
Figura 26 – Segunda medição do nível da água com a régua: 9,5 cm	63
Figura 27 – Sensor ultrassônico utilizado e sua inclinação	65

LISTA DE QUADROS

Quadro 1 – Comparação do presente projeto com trabalhos correlatos	21
Quadro 2 – Descrição do módulo ultrassônico utilizado.....	33
Quadro 3 – Custos dos materiais utilizados no protótipo	44

LISTA DE TABELAS

Tabela 1 – Propriedades dos meios envolvidos no protótipo	29
Tabela 2 – Valores lidos pelo sensor ultrassônico no protótipo.....	61
Tabela 3 – Comparação da precisão dos sensores da literatura e do protótipo desenvolvido	69

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
ARM	Advanced RISC Machines
CSS	Cascading Style Sheets
FEWS	Flood Early Warning System
GSM	Global System for Mobile Communications
hPa	hectopascal
HTML	HyperText Markup Language
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
JS	JavaScript
JSON	JavaScript Object Notation
LoRa	Long Range
LPWAN	Low-Power Wide-Area Network
MAE	Mean Absolute Error
MBE	Mean Bias Error
MRE	Mean Relative Error
MSE	Mean Square Error
NoSQL	Not Only SQL
ODS	Objetivos de Desenvolvimento Sustentável
ONU	Organização das Nações Unidas
POC	Proof of Concept
RDBMS	Relational Database Management System
RTDB	Real-Time Database
SDK	Software Development Kit
SQL	Structured Query Language
TS	timestamp
TTN	The Things Network
UFSC	Universidade Federal de Santa Catarina
WAN	Wide Area Network
Wi-Fi	Wireless Fidelity

SUMÁRIO

1	INTRODUÇÃO.....	16
1.1	PROBLEMA DE PESQUISA.....	16
1.2	OBJETIVOS.....	17
1.2.1	Objetivo Geral.....	17
1.2.2	Objetivos Específicos.....	17
1.3	JUSTIFICATIVA.....	17
1.4	ORGANIZAÇÃO DO TRABALHO.....	19
2	FUNDAMENTAÇÃO TEÓRICA.....	20
2.1	SISTEMAS EMBARCADOS.....	20
2.2	TRABALHOS CORRELATOS.....	20
2.3	PROTÓTIPO.....	25
2.4	REFLEXÃO DE ONDAS ULTRASSÔNICAS.....	28
2.5	PLACA UNO R4 WIFI.....	31
2.6	MÓDULO ULTRASSÔNICO.....	32
2.7	MICROCONTROLADOR.....	33
2.8	FIREBASE.....	34
2.9	BANCO DE DADOS.....	34
2.10	LISTENER ASSÍNCRONO DO BANCO DE DADOS.....	35
2.11	WEBSITE.....	36
2.11.1	Gráfico de nível da água.....	36
2.12	ERRO ABSOLUTO MÉDIO E ERRO DE VIÉS MÉDIO.....	37
2.13	ERRO RELATIVO E ERRO RELATIVO MÉDIO.....	39
3	METODOLOGIA.....	41
3.1	CUSTOS PARA CONSTRUÇÃO DO PROTÓTIPO.....	44
3.2	FUNCIONAMENTO DO PROTÓTIPO.....	45
3.3	INTEGRAÇÃO COM O BANCO DE DADOS.....	46
3.4	ENVIO DE EMAILS.....	48
3.5	BIBLIOTECAS UTILIZADAS NO CÓDIGO DA PLACA UNO R4 WIFI.....	49
3.6	CIRCUITO E LEITURA DO NÍVEL DA ÁGUA.....	50
3.7	WEBSITE DE MONITORAMENTO.....	52
3.8	SERVIÇOS BACKEND DE MONITORAMENTO.....	52

3.9	ACURÁCIA DA Sonda.....	54
4	RESULTADOS E DISCUSSÃO.....	55
4.1	PROVA DE CONCEITO.....	57
4.2	DEMONSTRAÇÃO DO NÍVEL DA ÁGUA LIDO PELO SENSOR.....	61
4.3	CÁLCULO DA ACURÁCIA DO SENSOR	64
4.3.1	Erro Absoluto Médio	64
4.3.2	Erro de Viés Médio.....	65
4.3.3	Erro Relativo e Erro Relativo Médio	66
4.4	COMPARAÇÃO DO EXPERIMENTO COM A LITERATURA.....	68
5	CONCLUSÃO	70
	REFERÊNCIAS	72
	APÊNDICE A – CÓDIGO UNO R4 WIFI	77
	APÊNDICE B – CREDENCIAIS.H (MASCARADO).....	82

1 INTRODUÇÃO

Com o aquecimento global e mudanças climáticas os desastres naturais têm se tornado cada vez mais frequentes em escala global, refletindo-se no Brasil por meio do agravamento e da recorrência das enchentes e inundações. Segundo o Emergency Database (2023 *apud* Faleiro *et al.*, 2024, p. 96), o Brasil é um dos países que mais sofre com desastres hidrológicos no mundo, causando um número preocupante de óbitos e impactando direta e indiretamente a população, seja por meio de doenças, afogamentos, danos materiais e/ou psicológicos. Nesse contexto, se faz necessário prevenir ou diminuir o impacto desses desastres num âmbito socioeconômico.

As áreas urbanas são as que mais tem interferência do homem no meio ambiente, que incluem desmatamento, poluição, canalizações de cursos d'água, construções, entre outros, resultando em aumento da temperatura nos centros urbanos, aumento das chuvas e, conseqüentemente, aumento também das enchentes (Poli, 2013).

Com o intuito de mitigar o impacto dessas enchentes foi desenvolvido um protótipo de sistema embarcado utilizando sensor ultrassônico que mede o nível de corpos d'água em tempo real.

1.1 PROBLEMA DE PESQUISA

Contextualizando, enchentes são a elevação do nível d'água no curso d'água, atingindo a cota máxima do canal, porém sem extravasar e inundações ocorrem quando a água transborda e inunda uma planície (Vieira *et al.*, 2025). Nesse sentido, a elevação de níveis de corpos d'água, como rios, que podem acarretar enchentes e inundações deve ser de alguma forma monitorada em tempo hábil para evitar ou mitigar o impacto das mesmas na população. No entanto, não são todos os municípios que possuem equipamentos automatizados para medir o nível de rios, por exemplo, para alertar a população em tempo suficiente sobre as enchentes e/ou inundações deles. Este é o caso do município de Araranguá, que ainda utiliza réguas para monitorar o nível do rio Araranguá, por exemplo (Araranguá, 2023).

Então o problema de pesquisa fundamenta-se em verificar se é possível desenvolver um sistema embarcado de baixo custo, utilizando sensor ultrassônico,

capaz de medir o nível de corpos d'água em tempo real com precisão adequada para fins de alerta de enchentes.

1.2 OBJETIVOS

Devido ao cenário supracitado, o presente trabalho tem os seguintes objetivos, delimitados em objetivo geral e objetivos específicos para um melhor entendimento.

1.2.1 Objetivo Geral

Desenvolver um protótipo de sistema embarcado de baixo custo utilizando sensor ultrassônico para monitorar em tempo real níveis de corpos d'água, visando prevenir ou mitigar o impacto de cheias na população em geral.

1.2.2 Objetivos Específicos

- Construir um protótipo funcional do sistema embarcado.
- Determinar a acurácia do protótipo funcional do sistema embarcado.
- Desenvolver um website para que o usuário possa visualizar o nível do corpo d'água que está monitorando e a funcionalidade de enviar um e-mail ao usuário alertando-o quando o corpo d'água medido estiver acima de determinado nível definido por ele.

1.3 JUSTIFICATIVA

Historicamente, a falta de monitoramento eletrônico do nível do rio Araranguá tem ocasionado muitos problemas socioeconômicos para o município e região. Por esse motivo, o presente trabalho visa a construção de um protótipo para este fim.

De acordo com o estudante de geografia da UNESC Djalma Santos Niles:

Os resultados demonstram que a inundação em Araranguá é do tipo lenta e gradual. A quantidade e intensidade de precipitação que ocorre nas cabeceiras da bacia são mais importantes em causar inundações em Araranguá do que aquela que ocorre na própria cidade. As marés meteorológicas e astronômicas realmente represam o fluxo do rio, o que é

problemático mesmo em episódios de chuvas não tão intensas e/ou continuadas. As canchas de arroz quando estão cheias podem contribuir para aumentar a intensidade da inundação, contudo essa relação deve ser investigada com maior profundidade. Percebeu-se a partir de observações que nos eventos de grande precipitação nas cabeceiras, a onda de cheia no canal leva aproximadamente dezoito horas para chegar no sítio urbano de Araranguá. É importante montar uma rede de monitoramento para lidar com eventos de inundação na cidade de Araranguá e em toda bacia. (Niles, 2009, p. 7)

E, ainda referente ao rio Araranguá: “Em sua trajetória, o rio atravessa inúmeras comunidades, acumulando em suas margens e leito resíduos e poluentes, que são descartados de maneira inadequada” (Araranguá, 2024). O que significa que se deve monitorar o nível do rio Araranguá para evitar que as inundações do mesmo não só causem problemas econômicos (como perda de bens materiais), mas também evitar que causem problemas sociais como mortes e a proliferação de doenças durante as enchentes devido ao acúmulo de resíduos e poluentes, conforme supracitado. Pois, ao receberem um alerta antecipado de que uma inundação está prestes a ocorrer, os habitantes de Araranguá e região, por exemplo, podem se deslocar e até conservar seus bens materiais antes de que a inundação chegue em suas casas.

Dessa forma, este projeto está justificado por abranger alguns Objetivos de Desenvolvimento Sustentável (ODS) da agenda 2030 da Organização das Nações Unidas (ONU), mais especificamente: o “Objetivo 9. Construir infraestruturas resilientes, promover a industrialização inclusiva e sustentável e fomentar a inovação” (ONU, 2015) e o “Objetivo 11. Tornar as cidades e os assentamentos humanos inclusivos, seguros, resilientes e sustentáveis” (ONU, 2015), particularmente o 11.1 – que fala sobre garantir o acesso da população à habitação segura – e o 11.5 – que fala sobre reduzir significativamente o número de mortes e de pessoas afetadas por catástrofes (ONU, 2015).

Assim, o presente trabalho tem o intuito de verificar se é possível desenvolver um sistema embarcado para medir nível de corpos d’água com os materiais e métodos supracitados. E assim contribuir com formas de monitoramento de corpos d’água mais eficientes e de baixo custo. Corroborando, assim, para diminuir ou prevenir o impacto de enchentes e inundações, especialmente nos centros urbanos e rurais do Brasil.

1.4 ORGANIZAÇÃO DO TRABALHO

O presente trabalho está estruturado em cinco capítulos de forma a conduzir o leitor de maneira lógica pelo conteúdo abordado no decorrer do mesmo:

Capítulo 1 – Introdução: contextualiza o tema abordado no trabalho e apresenta o problema de pesquisa, objetivos, justificativa e importância do estudo.

Capítulo 2 – Fundamentação Teórica: aborda os principais conceitos e suas definições e traz a revisão bibliográfica com trabalhos correlatos referentes ao presente trabalho.

Capítulo 3 – Metodologia: Demonstra e explica a metodologia utilizada para desenvolvimento do trabalho, os custos e como o protótipo foi desenvolvido.

Capítulo 4 – Resultados e Discussão: Demonstra os resultados obtidos, se o protótipo passou na prova de conceito e avalia utilizando métricas a precisão do protótipo, inclusive comparando-o com um estudo encontrado na revisão bibliográfica.

Capítulo 5 – Conclusão: Traz as conclusões do trabalho baseado nos objetivos propostos, as limitações e escalabilidade do protótipo e o que se espera para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

O presente projeto trata-se de um protótipo de sistema embarcado para monitoramento em tempo real de corpos d'água com o intuito de mitigar o impacto de enchentes na população em geral. Foi realizada uma revisão bibliográfica, na qual analisaram-se as tecnologias e métodos já utilizados atualmente para monitoração, controle e mitigação do impacto de cheias num âmbito socioeconômico. E, a partir disso, foram comparados também os objetivos, métodos, resultados e diferenciais dos trabalhos acadêmicos encontrados na revisão bibliográfica com os do presente trabalho.

2.1 SISTEMAS EMBARCADOS

Para o embasamento tecnológico desta pesquisa, é imprescindível compreender o conceito de sistemas embarcados. Segundo Oshana (2006, p. 27, tradução nossa):

Um sistema embarcado é um sistema computacional especializado que geralmente é integrado como parte de um sistema maior. Um sistema embarcado consiste em uma combinação de componentes de hardware e software para formar um mecanismo computacional que desempenhará uma função específica. Ao contrário dos sistemas de desktop, que são projetados para realizar funções de propósito geral, os sistemas embarcados são restritos em sua aplicação. Sistemas embarcados frequentemente operam em ambientes reativos e com restrições de tempo [...]. Um particionamento aproximado de um sistema embarcado consiste no hardware, que fornece o desempenho necessário para a aplicação (e outras propriedades do sistema, como segurança), e no software, que fornece a maioria das funcionalidades e a flexibilidade do sistema. (Oshana, 2006, p. 27, tradução nossa)

Dessa forma, o conceito principal do sistema embarcado desenvolvido está demonstrado na Figura 1 (adiante) e o seu funcionamento é explicitado no decorrer deste trabalho.

2.2 TRABALHOS CORRELATOS

Foram analisados inicialmente 5 trabalhos acadêmicos correlacionados com o objetivo do presente projeto. O quadro comparativo do presente trabalho com os trabalhos correlatos, onde são destacados o autor, o objetivo, os resultados, o

dashboard, o sistema de aviso e o diferencial de cada um, pode ser visto a seguir (Quadro 1).

Quadro 1 – Comparação do presente projeto com trabalhos correlatos

Autor	Objetivo	Resultados	Dashboard	S. aviso	Diferencial
Simatupang e Arrazaq (2023)	Projetar FEWS residencial acessível e portátil para detecção precisa.	Precisão de 99,71% e comunicação estável via rádio.	Blynk (app/web) + LCD	App Blynk + Buzzer/L ED	Rede rádio HC-12 local independente do Wi-Fi para o alarme físico.
Che Zulkifli (2019)	Notificar áreas de risco usando cobertura robusta de rede de celular (GSM).	Monitoramento eficaz sem Wi-Fi e mitigação de falsos positivos via software.	Não possui (apenas local para testes)	SMS (GSM) via rede móvel	Operação 100% offline focada exclusivamente em infraestrutura SMS.
Permatasari et al. (2023)	Prevenção ativa com bombas d'água via IoT na região de Bandung (Indonésia).	Controle preciso das bombas atuadoras e modelagem matemática validada.	Website customizado na nuvem	Acompanhamento web visual apenas	Acionamento autônomo de bombas usando Lógica Fuzzy.
Shukla et al. (2025)	Sistema "FloodGuard" para detectar risco e liberar água represada.	Ação física imediata em 1.5s, SMS em 3s e consome menos de 300mA.	ThingSpeak + LCD	SMS (GSM) + Buzzer	Híbrido (IoT/Offline) e com mitigação física por meio de Válvulas Solenoides.
Zakaria, Jabbar e Sulaiman (2023)	Sistema focado em grande alcance e eficiência energética para grandes bacias.	Rede validada até 1km com baixíssima perda de pacote e alta estabilidade IoT.	TTN, TagoIO e ThingSpeak	TagoIO (SMS, E-mail, chamadas)	Nós solares autossuficientes e arquitetura LPWAN de longo alcance.
Este trabalho (2026)	Sistema "Sonus Velox" de monitoramento remoto de corpos d'água com alerta ao usuário.	Acesso remoto ao nível da água e erro relativo médio de 2,96%.	Website hospedado no <i>Firebase</i> .	Envio de e-mail ao usuário quando o nível está alto.	Controle de acesso com usuário e senha (privacidade e segurança)

Fonte: elaborado pelo autor

O primeiro trabalho correlato, intitulado: "Design Implementation of Flood Early Warning System for Residential Monitoring" de Simatupang e Arrazaq (2023) foi encontrado no IEEE Xplore com as *keywords*: "water level" AND "ultrasonic sensor" AND "floods".

O segundo trabalho correlato, intitulado: "Development of Flood Notification System" de Che Zulkifli (2019) foi encontrado no Portal de Periódicos da CAPES com as *keywords*: "water level" AND "ultrasonic sensor".

O terceiro trabalho correlato, intitulado: "Development of IoT Control System Prototype for Flood Prevention in Bandung Area" de Permatasari *et al.* (2023) foi encontrado no Portal de Periódicos da CAPES com as *keywords*: "embedded systems" AND "water level" AND "ultrasonic sensor" AND "floods".

O quarto trabalho correlato, intitulado: "FloodGuard: An Embedded System for Real-Time Monitoring and Flood Management System" de Shukla *et al.* (2025) foi encontrado no IEEE Xplore com as *keywords*: "water level" AND "ultrasonic sensor" AND "floods".

O quinto trabalho correlato, intitulado: "Development of a smart sensing unit for LoRaWAN-based IoT flood monitoring and warning system in catchment áreas" de Zakaria, Jabbar e Sulaiman (2023) foi encontrando no Portal de Periódicos da CAPES com as *keywords*: "embedded systems" AND "water level" AND "ultrasonic sensor" AND "floods".

O trabalho de Simatupang e Arrazaq (2023) foca em projetar e implementar um protótipo de Sistema de Alerta Precoce de Inundações (*Flood Early Warning System* – FEWS) acessível para áreas residenciais, que possa ser facilmente utilizado a qualquer hora e local, com leituras precisas do nível da água vistas pelo Blynk. O sistema foi construído em duas unidades (transmissora e receptora). A unidade transmissora lê os níveis da água usando um sensor ultrassônico (HC-SR04) e um NodeMCU para enviar dados para a nuvem. A comunicação com a unidade receptora (que controla o alarme e o visor LCD) é feita localmente usando um módulo de comunicação serial sem fio HC-12 com uma placa Arduino Nano. O sistema atingiu uma precisão de 99,71% na leitura do nível da água (testado via métodos contínuos e não contínuos). Os alarmes locais e os aplicativos refletiram as mudanças de status (seguro, alerta e perigo) com alta consistência. A principal diferença é a abordagem de arquitetura dividida: utiliza uma rede de rádio local (HC-12) para a unidade de

alarme funcionar independentemente da internet, e usa o Wi-Fi simultaneamente para o monitoramento na nuvem via Blynk.

Che Zulkifli (2019) tem como objetivo desenvolver um sistema prático de notificação de inundações capaz de emitir avisos precoces via rede de celular, com o objetivo de reduzir a perda de propriedades e alertar a tempo os moradores e as autoridades das áreas afetadas. O respectivo protótipo usa um sensor ultrassônico conectado a um Arduino UNO R3. Os dados lidos são processados pelo Arduino e, quando atingem os níveis de "Atenção" (*Caution*) ou "Perigo" (*Danger*), o sistema aciona um Shield GSM (SIM900A) para disparar mensagens de texto. Há uma lógica de atraso (delay) de 10 segundos para checagem e prevenção de falsos alarmes causados por obstáculos temporários, no caso ele checa se o valor é o mesmo nesse intervalo, se for significa que é a altura real da água, caso contrário provavelmente é uma interferência ou objeto externo. O sistema foi capaz de medir as distâncias do nível da água com sucesso. O módulo GSM disparou com precisão os alertas SMS programados indicando o momento estimado da inundação e direcionamentos de abrigos, operando de forma estável. A vantagem é que não depende de internet Wi-Fi ou servidores em nuvem, baseando-se 100% na rede celular GSM (SMS), o que o torna ideal para locais remotos com pouca infraestrutura de internet. Além disso, a mensagem de perigo inclui uma estimativa do tempo de inundação.

O trabalho de Permatasari *et al.* (2023) tem o intuito de projetar um protótipo de sistema de controle para prevenir o impacto de inundações na região do rio Citarum, na Indonésia, utilizando Internet das Coisas (IoT) juntamente com lógica Fuzzy. A leitura do nível de água que pode ser visto num website é feita via sensores ultrassônicos e enviada por meio de módulos LoRa (ESP32) para um Raspberry Pi. O Raspberry Pi atua como o "cérebro" executando um algoritmo de Lógica Fuzzy com três conjuntos Fuzzy (raso, médio e alto). O processamento é feito em um Raspberry Pi, que avalia 9 regras condicionais (do tipo "Se-Então") baseadas nos estados de dois recipientes. O algoritmo de Lógica Fuzzy teve sucesso em acionar as bombas d'água no momento correto (erro aceitável constatado). O sistema também conseguiu gerar uma previsão (usando regressão linear simples) para o comportamento do nível da água nos próximos dias. O diferencial neste caso é que em vez de apenas emitir alertas, ele atua ativamente para tentar mitigar a enchente (ligando bombas de sucção de água), tudo operado e balanceado de forma autônoma por Lógica Fuzzy.

E o trabalho de Shukla *et al.* (2025) foca em criar um sistema "FloodGuard" acessível e de ação precoce que, além de detectar os níveis críticos da água (que pode ser visto pelo ThingSpeak) e alertar, toma medidas físicas autônomas para redirecionar o escoamento. Emprega um Arduino Uno e sensor ultrassônico. Ao ultrapassar o nível crítico estipulado, o Arduino liga um relé que abre automaticamente duas válvulas solenoides de 12V, desviando a água para "canais" secundários. Simultaneamente, o módulo GSM (SIM800L) manda alertas e um NodeMCU joga os dados na nuvem. As válvulas solenoides redirecionaram a água em um período máximo de 1,5 segundos após a detecção. Os alertas via SMS foram entregues entre 3 a 5 segundos. O projeto demonstrou extrema estabilidade operando por 24 horas sob baixo consumo de energia (menos de 300 mA). Neste caso, ele se destaca por ser um sistema híbrido robusto: alertas online e offline, além de realizar a mitigação física do problema abrindo válvulas solenoides para o escape mecânico da água (funcionando como monitoramento e escoamento preventivo simultaneamente).

Já o trabalho de Zakaria, Jabbar e Sulaiman (2023) introduziu um sistema inteligente para bacias hidrográficas focado em usar tecnologia LoRaWAN para cobrir longas distâncias sem fio, usar o mínimo de bateria e permitir atualizações em tempo real escaláveis. O nó principal é alimentado por painel solar e baterias, utilizando Arduino, sensor ultrassônico e shield LoRa. Os nós se comunicam com um gateway LoRaWAN construído em Raspberry Pi. A comunicação e a telemetria (incluindo PDR (*Packet Delivery Ratio* ou razão de entrega de pacotes), latência (atraso) e relação sinal-ruído – SNR (*Signal-to-Noise Ratio*)) foram rigorosamente testadas em distâncias de 100m a 1km. Comprovou-se uma transmissão de rede incrivelmente estável via LoRaWAN (sem perdas de pacotes dependendo do *Spreading Factor* usado – quanto maior o SF maior o alcance do sinal, maior o consumo de energia e menor a taxa de dados que se pode enviar por segundo). E os dados medidos alcançaram a nuvem perfeitamente mesmo em condições de baixo consumo. No caso, esse trabalho se diferencia pelo forte enfoque na tecnologia de rede LoRaWAN, oferecendo longuíssimo alcance e autonomia de energia via placas solares. O sistema não depende de Wi-Fi ou pacote de dados local da operadora no sensor e possui a mais completa integração multiplataformas IoT (usa 3 nuvens simultaneamente – TTN, TagoIO e ThingSpeak).

O presente trabalho consiste em desenvolver um protótipo de sistema embarcado de baixo custo utilizando sensor ultrassônico conectado à placa UNO R4 WiFi para medir o nível de corpos d'água em tempo real com o intuito de mitigar o impacto de enchentes. No qual os níveis da água são registrados num gráfico acessível através do website “Sonus Velox” – que requer autenticação (usuário e senha) para acesso. O destaque, neste caso, é que além de possuir um gráfico de onda do nível da água interativo no website, o sistema também envia um e-mail de alerta ao usuário quando o nível da água ultrapassa determinado valor definido por ele. Sendo então um sistema inovador, independente e acessível por uma ampla gama de dispositivos.

2.3 PROTÓTIPO

O sensor ultrassônico deste trabalho consiste em um *trigger* que dispara pulsos de onda ultrassônica e um leitor (*echo*) que mede o tempo que a onda ultrassônica levou para ir e voltar até sensor, dessa forma, através de uma equação que utiliza a velocidade do som, calcula-se a distância do objeto medido pelo tempo de ida e volta desta onda. Assim usando a velocidade do som no ar, que é de aproximadamente 343 metros por segundo, calcula-se a distância pela equação: $distancia = \frac{tempo_{onda} \times velocidade_{som}}{2}$, as variáveis são divididas por 2 porque considera-se que o $tempo_{onda}$ inclui o tempo de ida e volta da onda – o que resultaria no dobro da distância caso não estivesse dividido por 2 (Parajuli, 2024). A função para medir o $tempo_{onda}$ utilizada na plataforma Arduino é a *pulseIn()* que mede o pulso da onda (tempo de ida e volta da mesma) em microssegundos. Ou seja, divide-se por 10^6 a quantidade em microssegundos e assim obtêm-se a quantidade equivalente em segundos, porque um segundo tem 10^6 microssegundos (um milhão de microssegundos). Dessa forma obteve-se a equação para calcular a distância do sensor ultrassônico até a superfície da água (o sensor está suspenso acima da água):

$$d(cm) = \frac{(p/10^6 \times 343 \times 10^2)}{2}$$

Em que,

$d(cm)$ é a distância do sensor até a água, medida em centímetros (cm);

p é o pulso, ou tempo da onda, medido em microssegundos (μs);

Assim, pode-se obter a distância do sensor até a superfície da água em centímetros, pois segundo a equação acima o pulso (p) é medido em microssegundos (μs) e ao dividi-lo por 10^6 obtêm-se o *tempo_{onda}* (ou pulso) em segundos (s). No entanto, este valor é o tempo de ida e volta da onda (o dobro do tempo de ida da onda), o que faz necessário dividi-lo por 2 (conforme a equação acima) para obter-se o valor correto (apenas o tempo de ida da onda). E ao multiplicar este valor (tempo de ida da onda) pela velocidade do som que é de aproximadamente 343 m/s (trezentos e quarenta e três metros por segundo) obtêm-se a distância do sensor até a água em metros. E por último, ao multiplicar esse valor obtido anteriormente por 10^2 (cem), obtêm-se a distância do sensor ultrassônico até a superfície da água em centímetros, porque um metro tem cem centímetros. O que, após algumas manipulações algébricas resulta em:

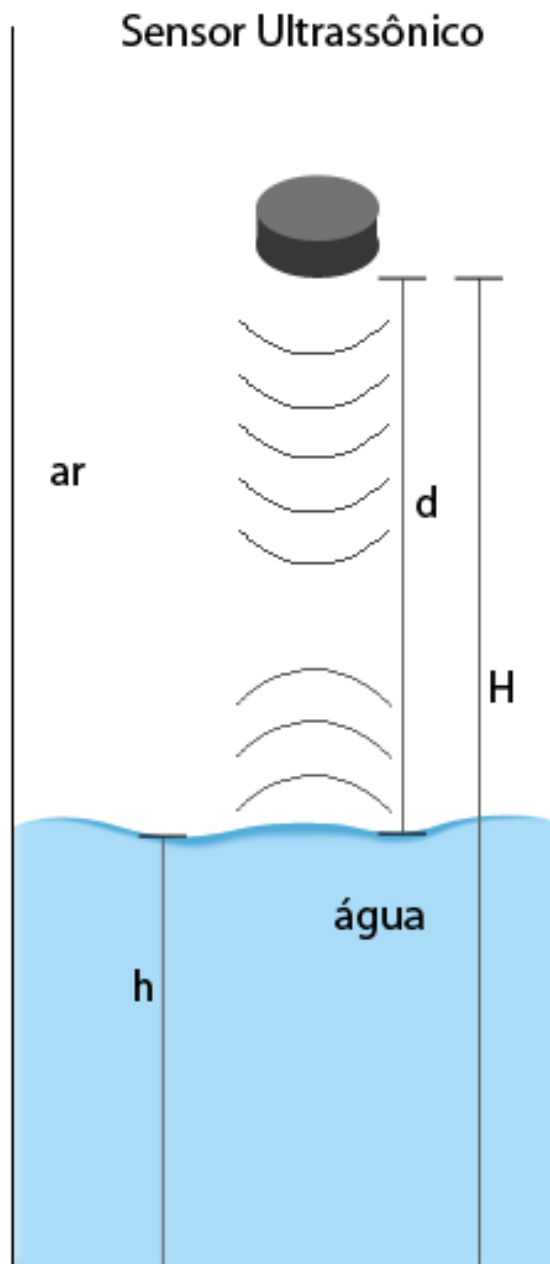
$$d(cm) = \frac{(p \times 343 \times 10^{-4})}{2}$$

E, por conseguinte:

$$d(cm) = \frac{(p \times 0,0343)}{2}$$

Em que, as variáveis e constantes da equação supracitada já foram explicadas previamente. Basicamente, o que ocorre é que uma onda ultrassônica é emitida pelo sensor ultrassônico (que está suspenso acima do corpo d'água), depois essa onda é refletida pela superfície da água e por último o sensor lê o tempo de ida e volta da onda (pulso). E através da equação acima obtêm-se a distância em centímetros (cm) do sensor até a superfície da água a partir do pulso explicado anteriormente. E assim, ao diminuir a altura H do sensor em relação ao chão pela distância d do sensor até a água tem-se a altura h do corpo d'água – conforme pode ser visto na Figura 1 (a seguir).

Figura 1 – Esquema do Sensor Ultrassônico para medição do nível da água



Fonte: elaborado pelo autor

Em que,

d é a distância do sensor até a superfície da água (cm);

H é a altura do sensor ultrassônico em relação ao chão;

h é o nível da água;

Dessa forma, conhecendo-se a altura H que o sensor ultrassônico instalado está do chão (até a base do corpo d'água) pode-se obter o nível da água h pela equação citada, subtraindo-se a distância d medida pelo sensor da altura total H do mesmo.

2.4 REFLEXÃO DE ONDAS ULTRASSÔNICAS

Segundo Urone e Hinrichs (2012, tradução nossa), "Reflexões nas fronteiras entre dois meios diferentes ocorrem devido a diferenças em uma característica conhecida como a impedância acústica Z de cada substância. A impedância é definida como:"

$$Z = \rho v$$

Em que,

Z é a impedância acústica do meio ($\text{kg} / (\text{m}^2 \cdot \text{s})$);

ρ é a densidade do meio (kg / m^3);

v é a velocidade do som no meio (m / s).

Na interface entre meios de diferentes impedâncias acústicas, parte da energia da onda é refletida e parte é transmitida. Quanto maior a diferença de impedância acústica entre os dois meios, maior a reflexão e menor a transmissão (Urone; Hinrichs, 2012, tradução nossa).

No caso, para saber a impedância acústica de determinado meio – como a da água e a do ar na Tabela 1 (a seguir) – basta multiplicar a densidade do meio (kg/m^3) pela velocidade de propagação da onda ultrassônica nele (m/s). Dessa forma, obtêm-se sua respectiva impedância acústica ($\text{kg} / (\text{m}^2 \cdot \text{s})$).

A seguir, na Tabela 1, pode-se ver algumas propriedades do ar e da água – que são os meios utilizados para fazer o protótipo da presente pesquisa, inclusive a impedância acústica de cada um dos meios.

Tabela 1 – Propriedades dos meios envolvidos no protótipo

Meio	Densidade (kg/m ³)	Velocidade de propagação da onda ultrassônica (m/s)	Impedância acústica (kg / (m ² * s))
Ar	1,3	343	445,9
Água	1000	1500	1,5 × 10 ⁶

Fonte: Adaptado de Urone e Hinrichs (2012). Nota: o valor da velocidade de propagação da onda ultrassônica no ar, que originalmente era 330, e a impedância acústica do ar, que originalmente era 429, foram substituídos por 343 e 445,9, respectivamente, pois esse valor de 343 m/s (velocidade de propagação da onda ultrassônica no ar) foi o admitido para fins de cálculo e uso na presente pesquisa, com base em Parajuli (2024).

Resumidamente, "O ultrassom é uma forma de radiação não ionizante que utiliza ondas sonoras de alta frequência." (Powles *et al.*, 2018, p. 202, tradução nossa) e "A refletância é um parâmetro usado para quantificar a capacidade de corpos, materiais ou superfícies de refletir a radiação que os atinge." (Fernández-García *et al.*, 2017, tradução nossa).

A refletância na interface entre o ar e os materiais é determinada pela diferença entre suas impedâncias acústicas. Como a impedância acústica do ar é quatro ou cinco ordens de grandeza menor do que a de líquidos ou sólidos, a refletância não muda drasticamente com alterações nas propriedades acústicas dos materiais. (Tanaka *et al.*, 2009, p. 617, tradução nossa)

Portanto, a refletância é a eficiência ou a porcentagem da energia (neste caso, onda ultrassônica) incidente que foi refletida. Segundo Urone e Hinrichs (2012, tradução nossa): o coeficiente de intensidade da reflexão α é definido como a razão da intensidade da onda refletida em relação à onda incidente (transmitida). Esta afirmação pode ser escrita matematicamente como:

$$\alpha = \frac{(Z_2 - Z_1)^2}{(Z_1 + Z_2)^2}$$

Em que,

α é o coeficiente de intensidade da reflexão (quanto maior, mais energia foi refletida, ou seja, maior a reflexão);

Z_1 é a impedância acústica do primeiro meio (de onde a onda vem) – neste caso, o ar (kg / (m² * s));

Z_2 é a impedância acústica do segundo meio (para onde a onda vai) – neste caso, a água ($\text{kg} / (\text{m}^2 \cdot \text{s})$).

(Urone; Hinrichs, 2012, tradução nossa)

Um coeficiente de reflexão zero (correspondente à transmissão total e ausência de reflexão) ocorre quando as impedâncias acústicas dos dois meios são iguais. Um 'casamento' de impedância (sem reflexão) proporciona um acoplamento eficiente da energia sonora de um meio para outro (Urone; Hinrichs, 2012, tradução nossa). O que não ocorre no nosso protótipo, pois a impedância acústica do ar é exorbitantemente menor do que a da água. Calculando o coeficiente de intensidade da reflexão α para o nosso protótipo com Z_1 sendo 445,9 ($\text{kg} / (\text{m}^2 \cdot \text{s})$) (impedância acústica do ar) e Z_2 sendo $1,5 \times 10^6$ ($\text{kg} / (\text{m}^2 \cdot \text{s})$) (impedância acústica da água), temos:

$$\alpha = \frac{(Z_2 - Z_1)^2}{(Z_1 + Z_2)^2}$$

Substituindo as variáveis na equação, obtêm-se:

$$\alpha = \frac{((1,5 \times 10^6) - 445,9)^2}{(445,9 + (1,5 \times 10^6))^2}$$

Resolvendo:

$$\alpha = \frac{(1500000 - 445,9)^2}{(445,9 + 1500000)^2}$$

$$\alpha = \frac{(1499554,1)^2}{(1500445,9)^2}$$

O que resulta em:

$$\alpha = \frac{2248662498826,81}{2251337898826,81}$$

$$\alpha \cong 0,998812$$

Isso significa que cerca de 99,88% da onda ultrassônica é refletida nas condições padrões e como a impedância acústica do ar é muito menor que a da água,

pequenas alterações nas propriedades acústicas dos materiais não mudam drasticamente a refletância para o nosso projeto (Tanaka *et al.*, 2009, tradução nossa). O que torna a medição do nível da água, neste caso, confiável e precisa, pois a detecção do eco (onda ultrassônica) não depende da pureza ou composição química da água, mas sim da grande barreira acústica natural existente entre o meio gasoso (ar) e o meio líquido (água).

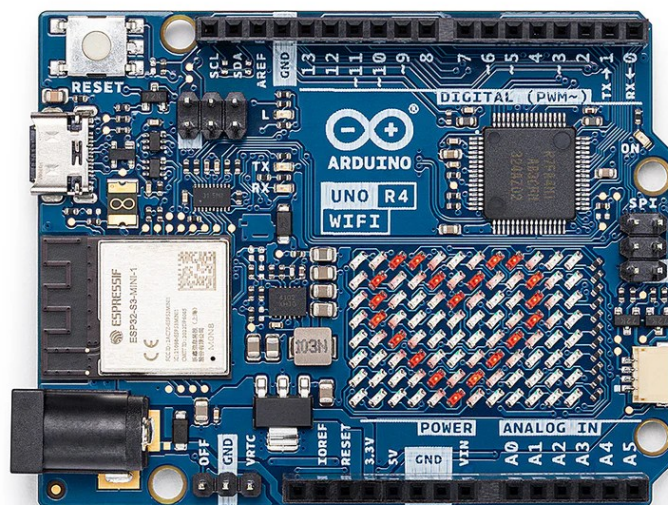
2.5 PLACA UNO R4 WIFI

Referente a placa UNO R4 WiFi, basicamente:

O microcontrolador (MCU) da placa é o Renesas RA4M1 de alto desempenho (Arm® Cortex®-M4), com velocidade de clock de 48 MHz, 32 kB de memória SRAM e 256 kB de memória flash. Este MCU dispõe de um relógio de tempo real (RTC), um conversor digital-analógico (DAC), um barramento CAN e suporte para dispositivos de interface humana (HID) via USB. O UNO R4 WiFi também conta com um módulo ESP32-S3 para conectividade Wi-Fi® e Bluetooth® [...] (Arduino, 2026, tradução nossa)

A placa utilizada pode ser vista na imagem a seguir (Figura 2). Resumidamente, ela contém as mesmas funções da placa UNO R4 Minima, mas com o acréscimo do módulo ESP32-S3 que vem integrado nela, o que permite conectividade Wi-Fi e possibilita a comunicação direta da placa com o banco de dados em tempo real da *Firebase*.

Figura 2 – Placa UNO R4 WiFi



Fonte: Arduino ([202-?])

2.6 MÓDULO ULTRASSÔNICO

Referente ao módulo ultrassônico AJ-SR04M (conforme pode ser visto na Figura 3 a seguir):

O módulo de medição de distância ultrassônico AJ-SR04M é projetado utilizando uma sonda *stripline* à prova d'água, com transceptor e receptor integrados, e tecnologia de detecção ultrassônica sem contato. Na faixa de 20 cm a 800 cm, o produto pode detectar com precisão a distância entre objetos planos e, na faixa de 20 cm a 250 cm, pode medir com precisão a distância até pessoas. (Arduino..., [20--?], p. 1, tradução nossa)

Figura 3 – Módulo ultrassônico AJ-SR04M



Fonte: Digital Concepts (2026)

“Os seres humanos conseguem detectar sons numa faixa de frequência de aproximadamente 20 Hz a 20 kHz.” (Purves *et al.*, 2001, tradução nossa). No Quadro 2 está a descrição do módulo ultrassônico utilizado. Segundo esse quadro o módulo emite e lê ondas ultrassônicas de 40 kHz – que, conforme citado anteriormente, é uma frequência acima da audição humana – por isso a onda disparada pelo sensor não é audível. E, também, há um “Recent range” de 20 cm no Quadro 2, isso significa que o objeto ao qual está se medindo a distância deve estar a pelo menos 20 cm de distância para se ter uma leitura correta. Segundo a descrição do módulo ultrassônico

usado (Quadro 2), o mesmo possui uma resolução de 2 mm. Entretanto, a precisão e acurácia do sensor são verificadas empiricamente no decorrer desta pesquisa.

Quadro 2 – Descrição do módulo ultrassônico utilizado

Parameter	Specification
Module	AJ-SR04M ultrasonic module
Working Voltage	DC 3-5.5V
Working Current	40mA Duration less than 50us
Standby Current	2mA
Low Power Current	Mode 2 current is 40μA, mode 4 and 5 current is 20μA
Working Frequency	40KHz
Furthest Range	8m
Recent Range	20cm
Measuring Angle	75 degrees
Input Trigger Signal	2:Trig/RX Trigger/Serial Receiver/Switch Enablation 2:Echo/TX Pulse Width Output/Serial Output/Switch Output
Output Echo Signal	Output TTL, Serial Port 5 Mode Selection
Serial Port Output Format	9600 n 8 1
Resolution	about 2mm
Working Temperature	-20°C ~ 75°C
Storage Temperature	-40°C ~ 80°C
Status Indicator	LED Indicating Status, One Flash/Switch Output Status
Size	41.3x28.5x23mm

Fonte: Adaptado de Arduino... ([20--?]).

2.7 MICROCONTROLADOR

Microcontroladores são amplamente utilizados em sistemas embarcados devido à capacidade de controlar dispositivos em tempo real e executar tarefas específicas. Ou seja, eles “[...] são pequenos dispositivos de computação usados em sistemas embarcados para controlar e monitorar funções e processos” (Souza, 2022).

[...] os microcontroladores integram processamento, memória e periféricos de entrada/saída (E/S), incluindo timers, contadores e transformadores analógico-digitais (ADCs), em uma unidade autônoma eficiente e econômica. Ao combinar vários componentes em um único sistema, os microcontroladores são adequados para aplicações que exigem processamento de sinal em tempo real, como controle de motores e assistentes e interface com vários tipos de sensores e comunicações. (Schneider; Smalley, 2024)

2.8 FIREBASE

Em sistemas embarcados conectados à internet é comum a necessidade de armazenar, acessar e analisar dados em tempo real. Logo, serviços de *back-end* acessíveis se tornam essenciais, principalmente para aqueles que não possuem capital para contratar um servidor dedicado. Basicamente, um servidor dedicado é:

Um servidor dedicado é um servidor físico atribuído a uma única organização. Você precisa de um servidor para hospedar ou executar os aplicativos e sites para que seus usuários possam acessar os serviços. O provedor de hospedagem gerencia e mantém os recursos físicos do servidor e os aloca para diferentes clientes. Dependendo do tipo de solução de hospedagem escolhida, talvez seja necessário compartilhar os recursos subjacentes do servidor físico com outras organizações. No entanto, na hospedagem dedicada, você tem controle total sobre como sua organização usa os recursos do servidor físico. Você obtém um ambiente de servidor físico isolado que pode ser configurado e gerenciado conforme necessário. Um servidor dedicado é útil para aplicativos de missão crítica que exigem alta disponibilidade ou para workloads de computação intensiva, onde o aplicativo exige muitos recursos do servidor. (Amazon Web Services, [s. d.])

Desta forma, para viabilizar o acesso remoto aos dados, foi utilizada a plataforma Google® *Firebase*. O *Firebase* do Google® possui hospedagem para aplicativos e websites e é considerado um *back-end as a service*, ou seja, “*back-end* como serviço”. Basicamente ele oferece uma variedade de serviços de *back-end* que já vêm prontos, como “[...] autenticação, análise de dados, armazenamento de dados, notificações, banco de dados e muitos outros [...]” (Ribeiro, 2023), tudo sem precisar criar um *back-end* do zero. Isso faz dessa plataforma uma opção prática e fácil para quem quer desenvolver aplicativos e serviços online.

2.9 BANCO DE DADOS

O serviço específico empregado foi o *Firebase Realtime Database* que é um banco de dados NoSQL hospedado na nuvem. Esse banco de dados é uma boa

escolha para o projeto, pois permite que ele possa acessar e enviar dados em tempo real. Assim, quando o microcontrolador envia um novo valor do nível da água para o banco de dados, esse valor é replicado para todos os clientes conectados, como, por exemplo, através do website desenvolvido. Além disso, esse banco de dados é *rule-based*, ou seja, permite configurar regras de acesso ao banco de dados. E isso está incluso em todos os níveis hierárquicos dos objetos do mesmo, dessa forma essas regras podem ser definidas pelo administrador para evitar acesso não autorizado ao banco de dados.

Em relação a estrutura de dados do *Firebase Realtime Database*:

Todos os dados do Firebase Realtime Database são armazenados como objetos JSON. Pense no banco de dados como uma árvore JSON hospedada na nuvem. Ao contrário de um banco de dados SQL, não há tabelas nem registros. Quando você adiciona dados à árvore JSON, eles se tornam um nó na estrutura JSON com uma chave associada.

(Firebase, 2025)

Para saber o que é um banco de dados NoSQL é preciso primeiro entender a diferença entre um banco de dados SQL (relacional) – ou seja, que utiliza um sistema de gerenciamento de banco de dados relacional (RDBMS) – de um NoSQL (não relacional). Segue a distinção do funcionamento deles:

Os dados em um RDBMS são armazenados em objetos de banco de dados que são chamados de tabelas. Uma tabela é uma coleção de entradas de dados relacionadas e consiste em colunas e linhas. Esses bancos de dados requerem a definição do esquema com antecedência, ou seja, todas as colunas e seus tipos de dados associados devem ser conhecidos de antemão para que os aplicativos possam gravar dados no banco de dados. Eles também armazenam informações vinculando várias tabelas por meio do uso de chaves, criando assim um relacionamento entre várias tabelas. No caso mais simples, uma chave é usada para recuperar uma linha específica para que ela possa ser examinada ou modificada.

Por outro lado, em bancos de dados NoSQL, os dados podem ser armazenados sem definir o esquema com antecedência—o que significa que você tem a capacidade de se mover e iterar rapidamente, definindo o modelo de dados à medida que avança. Isso pode ser adequado para requisitos específicos de negócios, seja baseado em gráficos, orientado a colunas, orientado a documentos ou como um armazenamento de valor-chave.

(Oracle, 2021)

Bancos de dados NoSQL são flexíveis, não exigindo um esquema pré-definido, o que é ideal para o desenvolvimento ágil de protótipos.

2.10 LISTENER ASSÍNCRONO DO BANCO DE DADOS

Referente ao *listener assíncrono*:

Listeners assíncronos: os dados armazenados em um Firebase Realtime Database são recuperados anexando um listener assíncrono a uma referência de banco de dados. Ele é acionado uma vez no estado inicial dos dados e posteriormente, quando há alterações. Um listener de eventos pode receber diversos tipos de eventos diferentes. Esse modo de recuperação de dados é compatível com SDKs Admin para Java, Node.js e Python. (Firebase, 2026)

Basicamente, ele é um programa, que no caso da presente pesquisa foi desenvolvido em *Python*, que fica operando em *background* monitorando a *Realtime Database do Firebase*, utilizando-se do SDK Admin para Python do *Firebase*, e disparando gatilhos (*triggers*) para executar ações quando há determinadas alterações no banco de dados supracitado. Inclusive é ele que, neste caso, envia o e-mail de alerta do usuário pela API da Brevo quando a altura do nível da água ultrapassa o limite definido pelo mesmo.

2.11 WEBSITE

Foram utilizados o Bootstrap, CSS, HTML, JavaScript e os serviços de *back-end* do *Firebase* para construção do website. Basicamente, o “Bootstrap é um framework front-end. Ele oferece uma coleção de componentes prontos, como botões, formulários, grids, modais, e muito mais — tudo baseado em HTML, CSS e JavaScript” (Magno, 2025). Então, dessa forma, ele facilita o desenvolvimento do *layout* do site. Os serviços de *back-end* do *Firebase* utilizados já foram citados, como hospedagem do website, autenticação de usuário e o banco de dados em tempo real (*Firebase Realtime Database*).

2.11.1 Gráfico de nível da água

A biblioteca utilizada para fazer os gráficos de nível da água foi a Chart.js. Basicamente, “[...] Chart.js é uma biblioteca JavaScript gratuita para a criação de gráficos baseados em HTML. É uma das bibliotecas de visualização mais simples para JavaScript [...]” (W3schools, [s. d.], tradução nossa).

Foram utilizados *plugins* para a biblioteca Chart.js para que os gráficos de nível da água das sondas ficassem como esperado, mais especificamente o *Luxon adapter* e o *Annotation plugin*.

De forma simples, “O Luxon oferece suporte nativo para fusos horários e internacionalização [...] uma vez carregado, este adaptador sobrescreve o adaptador de data padrão fornecido no Chart.js (como um efeito colateral)” (Chart.js, [20--], tradução nossa). Este adaptador (*Luxon adapter*) permite mostrar a data exata do registro através do *timestamp* armazenada no banco de dados, ou seja, o dia, mês, ano e a hora, minutos e segundos correspondente ao mesmo. Isso é muito útil nesta pesquisa, pois possibilita saber o dia e horário exato do registro, o que não seria possível sem este *plugin*. Caso não fosse utilizado este *plugin* a biblioteca Chart.js mostraria apenas o horário do registro (sem o dia do ano).

Referente ao *Annotation plugin*, “Este plugin desenha linhas, caixas, rótulos, pontos, polígonos e elipses na área do gráfico [...] As anotações não funcionarão em nenhum gráfico que não possua exatamente dois eixos” (Chart.Js, 2026, tradução nossa). Basicamente, este plugin foi utilizado para traçar o limite (linha em vermelho) no gráfico de nível da água das sondas. Dessa forma, o usuário pode saber quando será ativado o evento que alerta que o nível da água está alto. Quando ativado esse evento, um e-mail de alerta é enviado ao usuário informando que o nível da água de determinada sonda ultrapassou o limite definido pelo usuário. Incluindo a data e horário em que o evento ocorreu e informações relevantes sobre o nível da água e a respectiva sonda.

2.12 ERRO ABSOLUTO MÉDIO E ERRO DE VIÉS MÉDIO

Para medir a acurácia da sonda foi usado o Erro Absoluto Médio ou *Mean Absolute Error* (MAE). Através dele pode-se obter a média dos erros absolutos medidos. E o Erro de Viés Médio ou *Mean Bias Error* (MBE). Pois através dele pode-se obter a média dos erros e sua direção (se a sonda tende a superestimar ou subestimar as medições – como explicitado a seguir).

O Erro Absoluto Médio, no caso do sensor ultrassônico, é medido somando os módulos das subtrações do valor medido pelo valor real e dividindo o resultado dessa soma pelo número de medições. Segue a equação do Erro Absoluto Médio a seguir:

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i - x|$$

Em que,

MAE é o Erro Absoluto Médio;

n é a quantidade de medições;

x_i é a valor da altura da água medida na sonda;

x é valor real da altura da água medida na régua.

(Burch, 2023)

Foi escolhido o MAE no lugar do Erro Quadrático Médio ou *Mean Square Error* (MSE) pelo seguinte motivo:

O MAE é uma medida de perda regressiva que analisa a diferença de valor absoluto entre as previsões de um modelo e a verdade fundamental, calculada em média em todo o conjunto de dados. Ao contrário do Erro Quadrático Médio (MSE), o MAE é ponderado em uma escala linear e, portanto, não coloca tanto peso nos valores discrepantes. Isso fornece uma medida mais uniforme de desempenho [...]

(Burch, 2023)

Já o Erro de Viés Médio ou *Mean Bias Error* (MBE) pode indicar uma tendência do protótipo de subestimar ou superestimar um valor lido. A subestimação resulta em um valor negativo de MBE, enquanto um valor positivo representa uma superestimação. O valor desejável de MBE é zero (Jamil; Akhtar, 2017).

O Erro de Viés Médio, referente ao sensor ultrassônico, é medido somando as subtrações do valor medido pelo valor real do nível da água e dividindo o resultado dessa soma pelo número de medições. Segue a equação do Erro de Viés Médio a seguir:

$$MBE = \frac{1}{n} \sum_{i=1}^n x_i - x$$

Em que,

MBE é o Erro de Viés Médio;

n é a quantidade de medições;

x_i é a valor da altura da água medida na sonda;

x é valor real da altura da água medida na régua.

(Jamil; Akhtar, 2017).

2.13 ERRO RELATIVO E ERRO RELATIVO MÉDIO

O erro relativo fundamenta-se em: “A primeira etapa consiste em obter o erro absoluto entre a medição real (valor verdadeiro) e o valor do sensor (valor medido)” (Sari *et al.*, 2023, p. 4, tradução nossa). O erro absoluto pode ser calculado pela seguinte equação:

$$E_a = |x_i - x|$$

Em que,

E_a é o Erro Absoluto (cm);

x_i é o valor da altura da água medida no sensor ultrassônico (cm);

x é o valor real da altura da água medida na régua (cm).

(Sari *et al.*, 2023, p. 4, tradução nossa)

Segundo Sari *et al.* (2023, p. 4, tradução nossa): o valor que demonstra a precisão do sistema embarcado proposto pode ser determinado calculando o erro relativo, que pode ser obtido comparando o erro absoluto com a medição real através da seguinte equação:

$$E_r = \frac{E_a}{x} \times 100\%$$

Em que,

E_r é o Erro Relativo (%);

E_a o Erro Absoluto (cm);

x é o valor real da altura da água medida na régua (cm).

(Sari *et al.*, 2023, p. 4, tradução nossa)

Então, por dedução, tem-se o erro relativo médio ou *Mean Relative Error* (MRE) que nada mais é do que a média aritmética dos erros relativos de todas as medições feitas pelo protótipo.

Logo, a equação do erro relativo médio é a seguinte:

$$MRE = \frac{1}{n} \sum_{i=1}^n \frac{|x_i - x|}{x} \times 100\%$$

Em que,

MRE é o Erro Relativo Médio (%);

n é a quantidade de medições;

x_i é o valor da altura da água medida na sonda (cm);

x é o valor real da altura da água medida na régua (cm).

Dessa forma, obtêm-se com mais acurácia a precisão do protótipo desenvolvido.

3 METODOLOGIA

Este trabalho de conclusão de curso foi desenvolvido com uma metodologia que guia a pesquisa, coleta e análise de dados, além dos passos técnicos para criar e validar a solução proposta, a qual está detalhada a seguir.

A pesquisa é classificada como aplicada, ou seja, busca resolver problemas práticos e gerar conhecimento útil (Gil, 2008). Neste caso, o intuito é criar uma solução tecnológica que ajude a monitorar corpos d'água e reduzir os efeitos de enchentes.

Ela utiliza uma abordagem quantitativa que foca em dados numéricos para testar hipóteses e entender a relação entre variáveis usando métodos estatísticos (Creswell, 2010). Para validar o sensor ultrassônico, será fundamental coletar dados como distância da base do corpo d'água até o sensor (em centímetros), a altura da água calculada pelo sistema (em centímetros) e a altura da água medida pela régua. Analisando esses dados com métricas como o Erro Médio Absoluto (MAE), o Erro de Viés Médio (MBE) e o Erro Relativo Médio (MRE), será possível medir o desempenho do protótipo de forma clara. Mais especificamente, foi utilizada a metodologia *Design Science Research Methodology* (DSRM) que segundo Çağdaş e Stubkjær (2011 *apud* Freitas Junior *et al.*, 2017, p. 25) basicamente é: “Design Science Research se constitui em um processo rigoroso de projetar artefatos para resolver problemas, avaliar o que foi projetado ou o que está funcionando, e comunicar os resultados obtidos”. A DSRM foi adotada por ser adequada ao desenvolvimento e avaliação de artefatos tecnológicos destinados à solução de problemas reais.

A DSRM, segundo Peffers *et al.* (2007 *apud* Freitas Junior *et al.*, 2017, p. 25), é desenvolvida a partir de seis etapas procedurais, que são sugeridas na ordem especificada na Figura 4, mas que podem ser executadas de acordo com a necessidade de determinado projeto:

Etapa 1 - Identificação do problema e sua motivação: esta etapa é dedicada à definição do problema de pesquisa específico, apresentando-se uma justificativa para a sua investigação. É importante que a definição deste problema seja empregada na construção de um artefato que pode efetivamente oferecer a solução para este problema. Tem-se como recursos necessários para esta etapa o estado da arte do problema e da relevância da solução apresentada.

Etapa 2 - Definição dos objetivos para a solução: tendo-se como ponto de partida o conhecimento acerca do problema, bem como a noção do que é viável e factível, delineiam-se os objetivos da solução a ser desenvolvida. Elencam-se como requisitos desta etapa novamente o estado da arte do problema e o conhecimento das possíveis soluções já previamente apresentadas.

Etapa 3 - Projetar e desenvolver: etapa destinada à criação do artefato, determinando-se a sua funcionalidade desejada para o artefato, sua arquitetura e em seguida a criação do próprio artefato. Os recursos necessários para a terceira etapa compreendem o conhecimento da teoria que pode ser exercida em uma solução.

Etapa 4 – Demonstração: momento de demonstração do uso do artefato resolvendo uma ou mais instâncias do problema por meio de um experimento ou simulação, estudo de caso, prova formal ou outra atividade apropriada. Os recursos relacionados para esta etapa incluem o conhecimento efetivo de como usar o artefato para resolver o problema.

Etapa 5 – Avaliação: nesta etapa deve-se observar e mensurar como o artefato atende à solução do problema, comparando-se os objetivos propostos para a solução com os resultados advindos da utilização do artefato. Pode-se definir pela recursividade da metodologia, isto é, o retorno às etapas 3 ou 4, de modo a aprimorar o artefato.

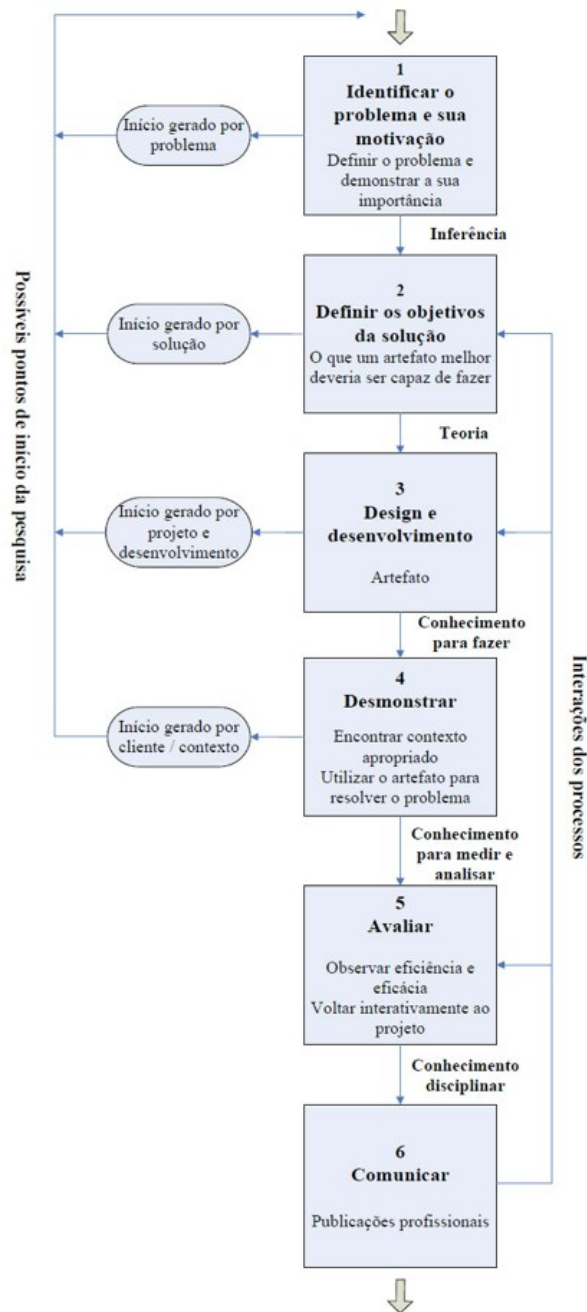
Etapa 6 – Comunicação: momento de divulgação do problema e da relevância da propositura de uma solução para o mesmo, além da apresentação do artefato desenvolvido.

(Peffer *et al.*, 2007 *apud* Freitas Junior *et al.*, 2017, p. 25)

Na Etapa 1 ocorreu a identificação do problema e motivação, relacionada à necessidade de monitoramento automatizado e de baixo custo de corpos d'água. Na Etapa 2 definiu-se os objetivos da solução, estabelecendo os requisitos funcionais do sistema. Na Etapa 3 houve o projeto e desenvolvimento do artefato, envolvendo a construção do sistema embarcado, do banco de dados em nuvem, website de monitoramento e do *listener* assíncrono. Na Etapa 4 ocorreu a demonstração, por meio da implementação e funcionamento do protótipo, na Etapa 5 a avaliação, realizada através da análise da precisão do sistema utilizando as métricas MAE, MBE e MRE, bem como comparação com estudos da literatura e na Etapa 6 a comunicação dos resultados por meio deste Trabalho de Conclusão de Curso.

A Figura 4 a seguir demonstra a sequência metodológica da DSRM.

Figura 4 – Etapas da DSRM



Fonte: Jappur (2014)

Quanto aos procedimentos técnicos, o estudo incluiu duas partes: bibliográfica e experimental.

A parte bibliográfica baseia-se em materiais já existentes, como livros e artigos científicos (Gil, 2008). Isso serve como a base teórica do projeto, ajudando a

entender os princípios da reflexão de onda ultrassônicas, o funcionamento dos sistemas embarcados, as arquiteturas de aplicações conectadas à internet e como validar sensores.

A pesquisa experimental, por outro lado, foca em um objeto de estudo, selecionando as variáveis que podem influenciá-lo e definindo como observar os efeitos dessas variáveis (Marconi; Lakatos, 2003). Essa parte prática envolve construir um protótipo da sonda em um ambiente conhecido, monitorar o nível da água através da leitura da distância lida pelo sistema embarcado usando o sensor ultrassônico, coletar dados e comparar os resultados com um padrão de referência (altura real da água) para validar a prova de conceito do dispositivo.

Assim, a metodologia combina a teoria da pesquisa bibliográfica com a prática da pesquisa experimental e quantitativa para desenvolver e validar um protótipo funcional de um sistema embarcado usando um sensor ultrassônico.

Os recursos que foram utilizados para montar o protótipo são os seguintes: plataforma Arduino®, placa UNO R4 WiFi (possui o microcontrolador Renesas RA4M1 (Arm® Cortex®-M4) e o módulo ESP32-S3-MINI-1-N8 integrados) (Arduino, 2026), módulo ultrassônico AJ-SR04M, fios, protoboard e um recipiente de acrílico.

3.1 CUSTOS PARA CONSTRUÇÃO DO PROTÓTIPO

Contextualizando, partindo dos recursos descritos acima para montagem do protótipo, todos os materiais e seus respectivos custos utilizados para construção do sistema embarcado podem ser vistos no Quadro 3 (a seguir).

Quadro 3 – Custos dos materiais utilizados no protótipo

Material utilizado	Custo
1 pote acrílico	R\$ 94,32
1 mangueira em pvc/silicone	R\$ 8,41
Kit jumper Macho x Femea 10 unidades	R\$ 5,01
Kit jumper Macho x Macho 65 unidades	R\$ 14,63
Protoboard 1660 Pontos	R\$ 40,00
Placa UNO R4 WiFi	R\$ 130,73
Waterproof Ultrasonic module AJ-SR04M	R\$ 51,45
Type-C Cable for UNO R4 WiFi 1pcs	R\$ 16,50
Pote quadrado Plasvale 1,3 L 1pcs	R\$ 19,37
Total	R\$ 380,42

Fonte: elaborado pelo autor

3.2 FUNCIONAMENTO DO PROTÓTIPO

Dentre os componentes utilizados está o sensor ultrassônico AJ-SR04M. O sensor ultrassônico instalado pode ser visto a seguir (Figura 5).

Figura 5 – Sensor ultrassônico instalado



Fonte: elaborado pelo autor

Basicamente o protótipo funciona assim: a placa compatível com a plataforma Arduino® (UNO R4 WiFi) foi utilizada para medir a distância até a água através da leitura do pulso (tempo de onda) de um sensor ultrassônico (suspense acima da água), que através de uma equação a mesma consegue determinar o nível da água e o envia para a RTDB do *Firebase* do Google®. Foi utilizado o *Firebase* para hospedagem e desenvolvimento do website de monitoramento. E foram calculados o Erro Absoluto Médio, o Erro de Viés Médio e o Erro Relativo Médio para checar a precisão da sonda. Inclusive comparando a métrica de Erro Relativo Médio com a do artigo de Sari *et al.* (2023) que usou também sensores ultrassônicos.

3.3 INTEGRAÇÃO COM O BANCO DE DADOS

Conforme diz Ribeiro (2023): o “Firebase Realtime Database é um banco de dados NoSQL hospedado na nuvem que permite armazenar e sincronizar dados em tempo real”. Foram definidas as regras de acesso ao banco de dados conforme a Figura 6 a seguir (referenciado também no Apêndice A – na parte em que ocorre o envio dos dados da leitura do nível da água para o banco de dados).

Figura 6 – Regras de acesso ao banco de dados

```

1  {
2    "rules": {
3      //".read": "now < 1766804400000", // 2025-12-27
4      //".write": "now < 1766804400000", // 2025-12-27
5      "usuario": {
6        "$uid": {
7          ".read": "$uid === auth.uid",
8          ".write": "$uid === auth.uid"
9        }
10     }
11   }
12 }
```

Fonte: elaborado pelo autor

A partir da Figura 6 nota-se que dentro do objeto “usuario” só pode ter acesso aos \$uid (número único que identifica o usuário) o usuário com o \$uid correspondente.

Dessa forma, cada usuário está restrito a modificar e ler apenas os seus dados correspondentes.

A Figura 7 exemplifica como ficou estruturado o banco de dados.

Figura 7 – Estrutura do banco de dados



Fonte: elaborado pelo autor

Dessa forma um usuário com determinado \$uid pode configurar quantas sondas forem necessárias sem precisar ter criado o esquema previamente, pois o banco de dados é não relacional.

O código para comunicação com o banco de dados da placa utilizada UNO R4 WiFi está referenciado no Apêndice A e os dados para autenticação e login no banco de dados estão demonstrados no Apêndice B (os dados estão mascarados por questão de segurança).

A placa utilizada para fazer o protótipo funciona da seguinte maneira:

1. **Conexão com o banco de dados:** a placa UNO R4 WiFi se conecta ao Wi-Fi local utilizando as credenciais definidas no Apêndice B e faz login na *Firebase* com as credenciais que estão neste mesmo Apêndice. Tudo isso através do código em que foi feito o *Sketch* (upload) para ela (Apêndice A).
2. **Medição da distância até a superfície da água:** a placa UNO R4 WiFi conectada por um circuito pela protoboard ao sensor ultrassônico conta o

tempo de pulso (tempo de ida e volta) da onda ultrassônica até a superfície da água.

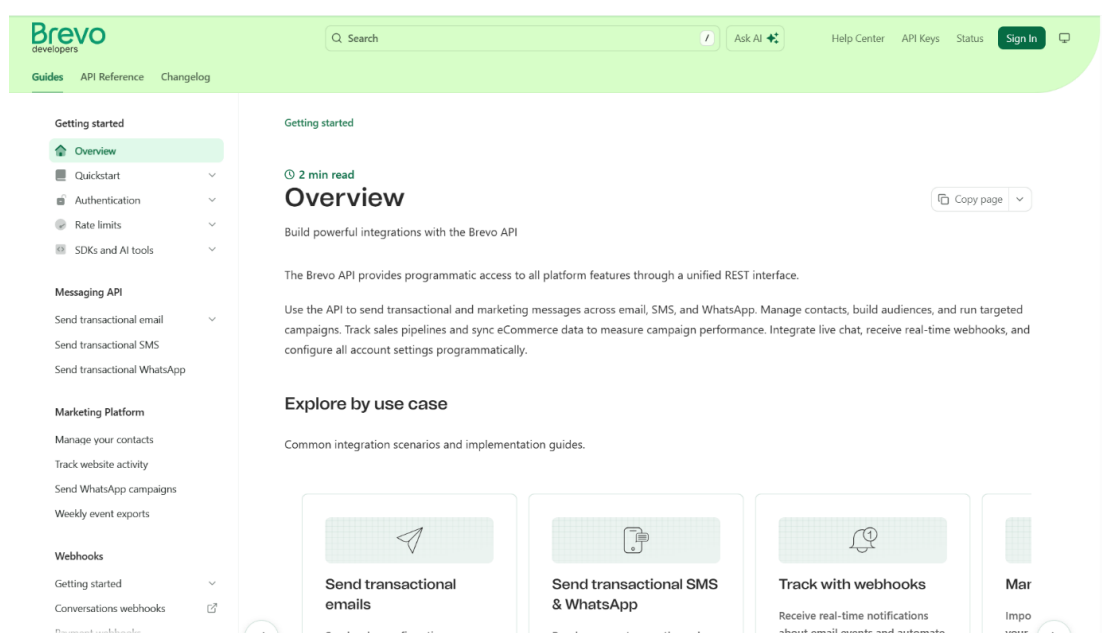
3. **Cálculo do nível da água:** através da equação demonstrada na fundamentação teórica a placa UNO R4 WiFi calcula a distância até a superfície da água em centímetros. E com isso, ao diminuir a distância até a superfície da água da altura total do sensor até o chão (90 cm) têm-se o nível da água.
4. **Transmissão de Dados:** O módulo WiFi (ESP32-S3) envia o dado de nível da água calculado para a plataforma em nuvem.

Essa abordagem permite que o dispositivo opere de forma autônoma e em tempo real, fornecendo dados contínuos para análise. Dessa forma, a placa pode ao mesmo tempo usar as funções do Arduino® Uno e a do módulo Wi-Fi. Permitindo assim a integração da placa com o banco de dados.

3.4 ENVIO DE EMAILS

A integração com o serviço de e-mails (envio de e-mail de alerta ao usuário) foi realizada através da API da plataforma Brevo (BREVO, 2026). A mesma foi usada no *listener* assíncrono desenvolvido em *Python*. A página da documentação da plataforma Brevo (API) pode ser vista na Figura 8 (a seguir).

Figura 8 – Documentação da API Brevo

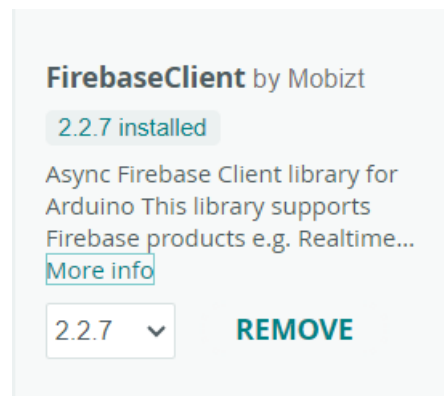


Fonte: Brevo (2026)

3.5 BIBLIOTECAS UTILIZADAS NO CÓDIGO DA PLACA UNO R4 WIFI

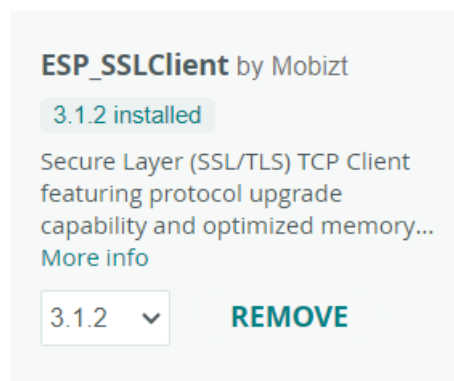
Foram utilizadas as bibliotecas `FirebaseClient` (Figura 9) e `ESP_SSLClient` (Figura 10), ambas do Mobizt (usadas no Apêndice A – código da placa UNO R4 WiFi). As mesmas foram instaladas na IDE do Arduino utilizada, conforme pode ser visto nas figuras a seguir.

Figura 9 – Biblioteca: `FirebaseClient` by Mobizt



Fonte: elaborado pelo autor

Figura 10 – Biblioteca: `ESP_SSLClient` by Mobizt



Fonte: elaborado pelo autor

Com essas bibliotecas foi possível fazer a comunicação da placa UNO R4 WiFi com o banco de dados em tempo real da *Firebase* do Google®. Permitindo o envio do nível da água lido pelo sensor ultrassônico para o banco de dados.

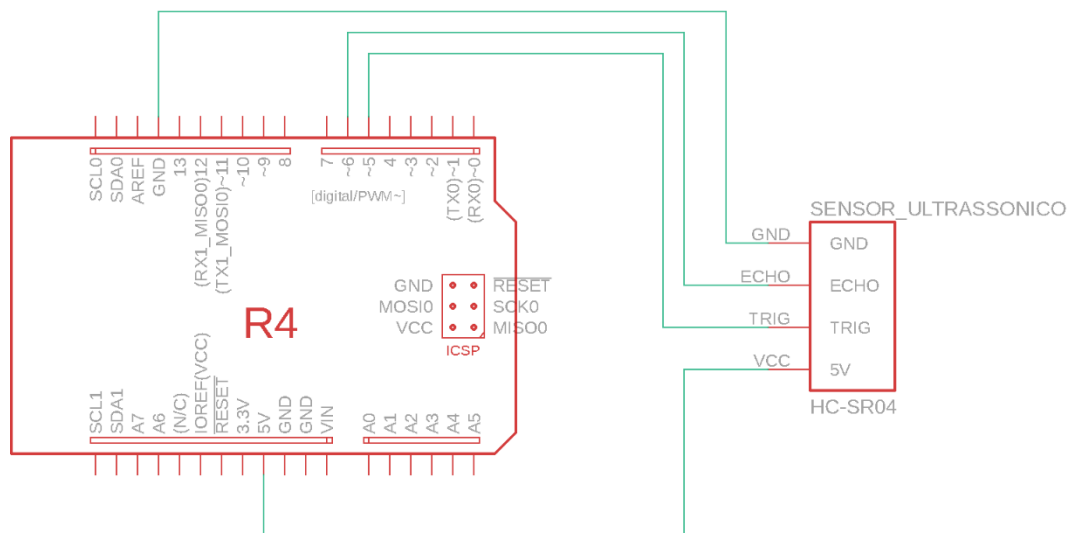
3.6 CIRCUITO E LEITURA DO NÍVEL DA ÁGUA

O código para leitura do nível da água está referenciado no Apêndice A. Já o circuito utilizado para montagem do sistema embarcado pode ser visto na Figura 11.

Assim, a placa UNO R4 WiFi lê o tempo do pulso do módulo ultrassônico AJ-SR04M (que está conectado nela) e através de uma equação determina o nível da água. Mais especificamente, a placa UNO R4 WiFi determina a distância do sensor ultrassônico utilizado até a superfície da água através de uma equação explicada anteriormente que envolve o tempo de onda e a velocidade do som. Assim a placa calcula o nível da água diminuindo a altura total do sensor ultrassônico até o chão pela distância do mesmo até a superfície da água. Dessa forma o valor do nível da água é enviado para a *Realtime Database* do *Firebase* do Google®. E assim é feito o tratamento desse valor no projeto do *Firebase* do Google® o qual pode ser visto no website e alertado através do e-mail do usuário.

Foi utilizado um sensor ultrassônico para a construção do sistema embarcado que mede a altura da água. Basicamente o mesmo está ligado ao seguinte circuito (esquema na Figura 11).

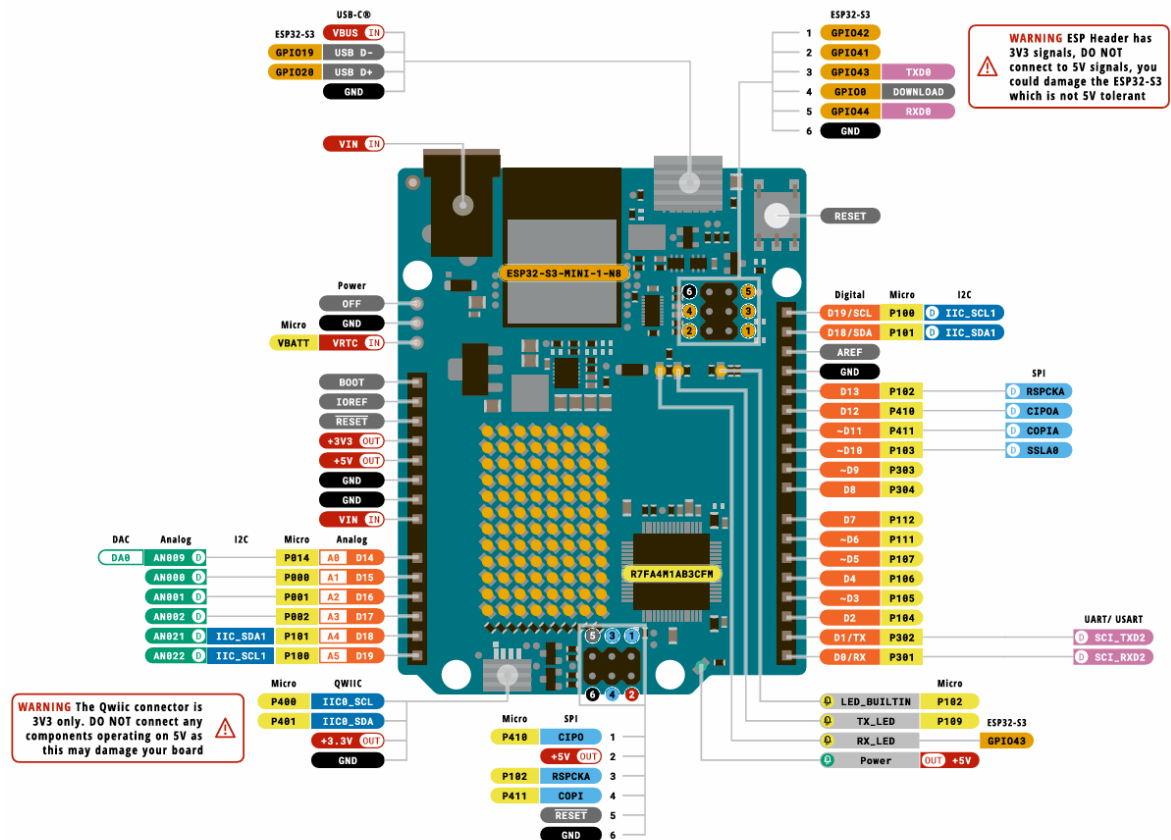
Figura 11 – Circuito do protótipo



Fonte: elaborado pelo autor

A seguir temos o *Pinout* (Figura 12) que nada mais é que um diagrama que, com a sua respectiva legenda (Figura 13), mostra as funções de cada pino da placa utilizada (UNO R4 WiFi).

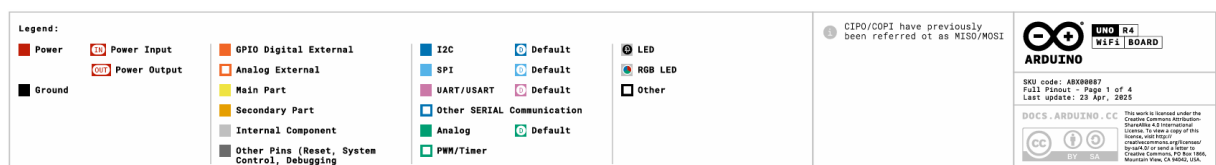
Figura 12 – Uno R4 WiFi Pinout



Fonte: Arduino (2026)

Conforme pode ser visto na legenda (Figura 13) do *pinout* da placa Uno R4 WiFi utilizada (Figura 12) os pinos utilizados D5 e D6 são digitais, o que permite a leitura digital e controle do sensor ultrassônico utilizado. Mais especificamente, a emissão de pulsos (ondas ultrassônicas) ocorre através do *trigger* do módulo ultrassônico utilizado (conectado ao pino D5 da placa) e a leitura do tempo da onda ocorre através do *echo* do módulo ultrassônico (conectado ao pino D6 da placa). Foi escolhido especificamente estes pinos (D5 e D6) porque não possuem nenhuma outra atribuição além de serem pinos principais e digitais (ver Figura 12 e 13), o que os torna confiáveis em não estarem sendo usados para outra função.

Figura 13 – Uno R4 WiFi Pinout: Legenda



Fonte: Arduino (2026)

3.7 WEBSITE DE MONITORAMENTO

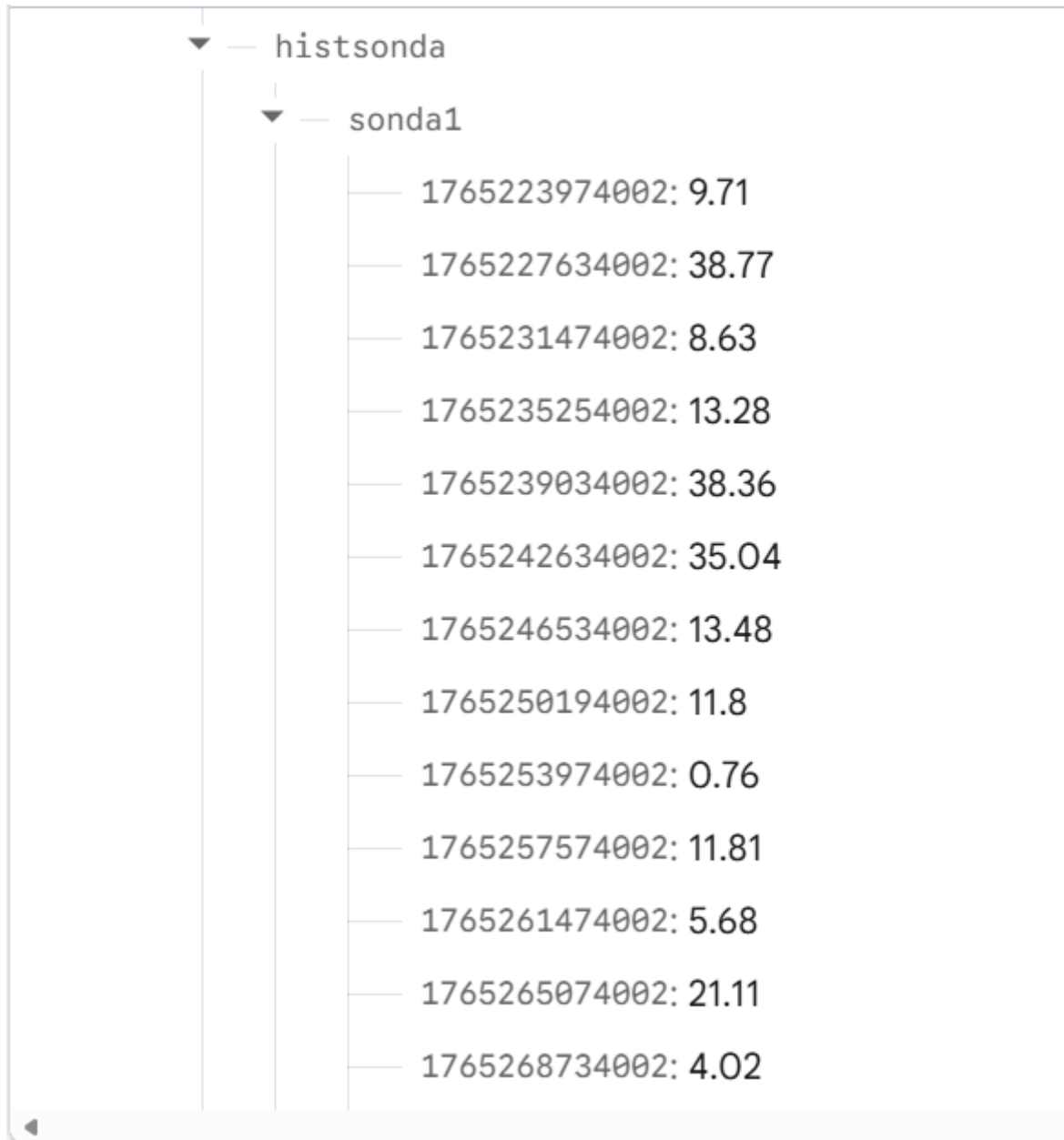
Foi feito um website para consulta e alerta do nível da água para o usuário. O website obtém os dados da *Firebase Realtime Database* e trata esses dados mostrando também em forma de gráfico de onda a altura da água lida pela sonda. O usuário é alertado por e-mail quando a altura da água ultrapassa determinado limite pré-definido por ele.

3.8 SERVIÇOS BACKEND DE MONITORAMENTO

Foi desenvolvido um *listener* assíncrono em *Python* que funciona no *backend* para “escutar” alterações feitas na *Realtime Database* do *Firebase* e registrar a cada 1 hora o valor do nível da água do sensor no seu respectivo gráfico de nível da água que aparece no *dashboard* do website criado. E, também, enviar um e-mail ao usuário alertando-o quando o nível da água lido por determinado sensor ultrassônico ultrapassar o limite predefinido pelo usuário, com intervalo de pelo menos 1 hora entre os avisos do respectivo sensor.

Mais especificamente, o *listener* registra o timestamp como key e o valor lido do sensor como value no *path*: “usuario/\$uid/histsonda/\$nome_sonda/” criando até 25 registros por sonda (\$nome_sonda), conforme pode ser visto na Figura 14. Ele registra a cada 1 hora com base no valor atual (value) lido do *path*: “usuario/\$uid/sonda/\$nome_sonda” em que \$nome_sonda é o nome da sonda e o value dessa sonda por exemplo da sonda1 o último valor lido (conforme pode ser visto na Figura 15) é 9.36.

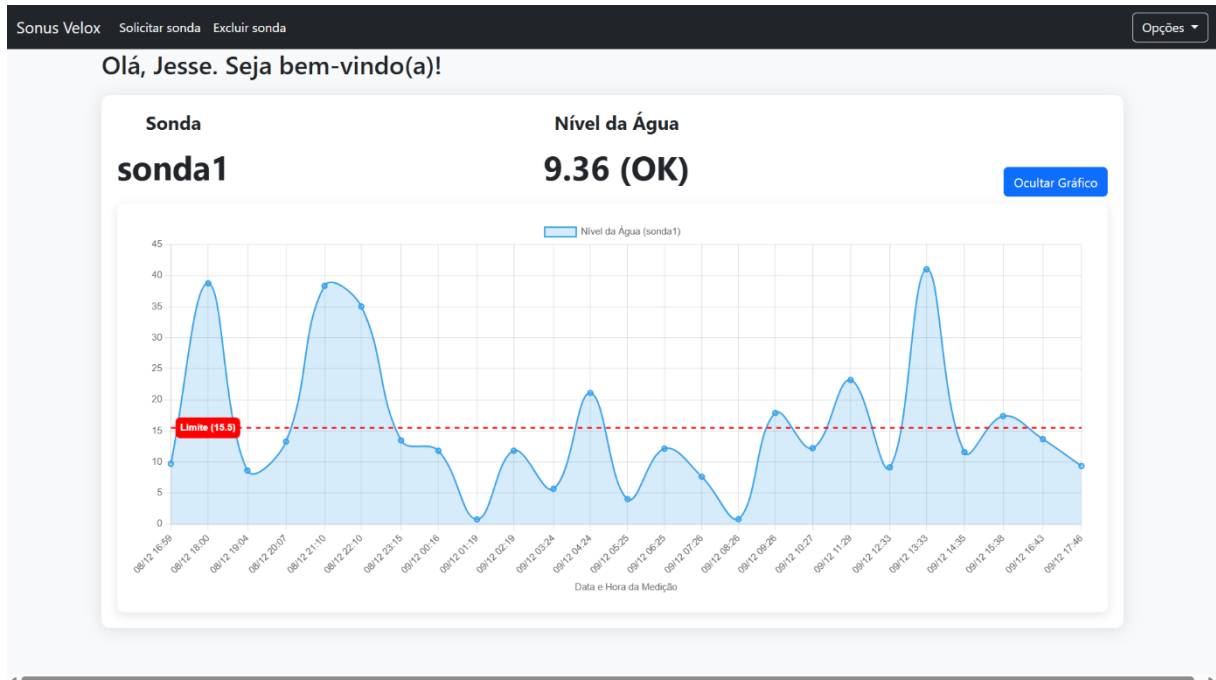
Figura 14 – Histórico fictício de nível da água da sonda



Fonte: elaborado pelo autor

O gráfico de nível da água fictício da sonda1 referente aos registros demonstrados na Figura 14 está representado na Figura 15 (a seguir).

Figura 15 – Dashboard do website com dados fictícios



Fonte: elaborado pelo autor

3.9 ACURÁCIA DA SONDA

Como detalhado na fundamentação teórica, foram usados o Erro Absoluto Médio (MAE), o Erro de Viés Médio (MBE) e o erro relativo para determinar a acurácia da sonda. Assim foi feito um determinado número de medições com a sonda em relação ao valor da altura real da água (medida pela régua). E foram calculados o MAE e o MBE. O primeiro determina a taxa de erro absoluto médio e o segundo retorna, além da taxa de erro, se os valores estão sendo superestimados (MBE positivo) ou subestimados (MBE negativo) pela sonda.

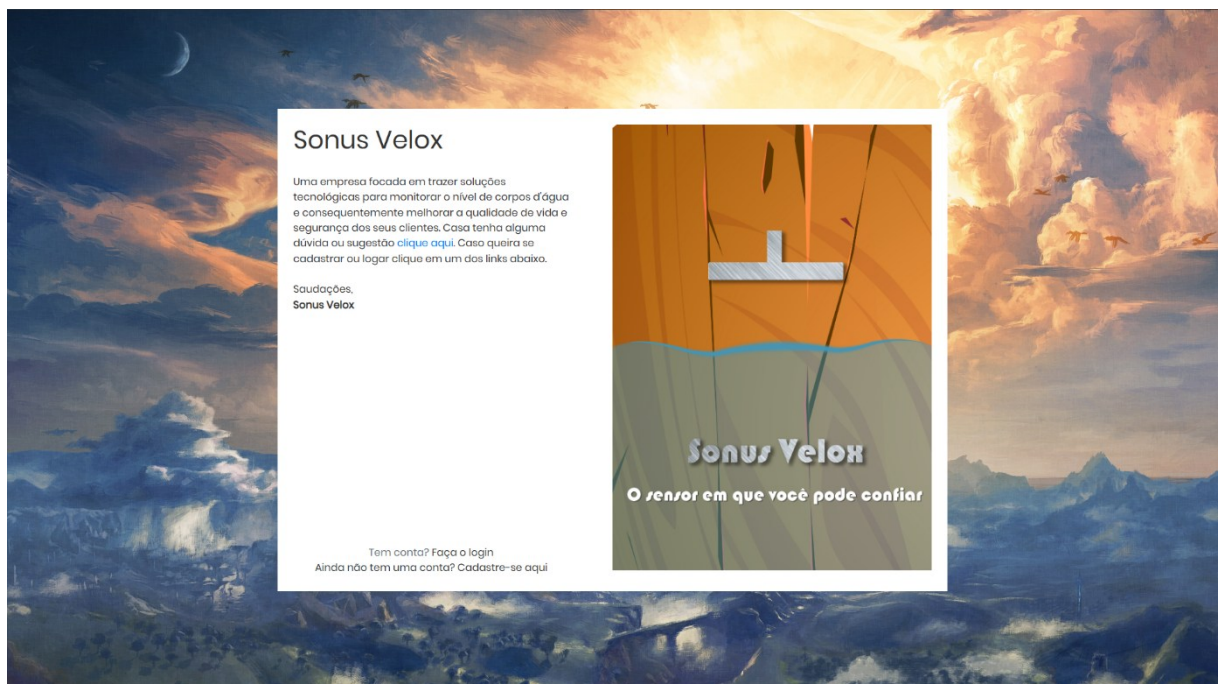
Com o intuito de verificar a acurácia do presente experimento com a do artigo de Sari *et al.* (2023) foi calculado também o erro relativo médio do sensor utilizado no sistema embarcado desenvolvido para comparação.

4 RESULTADOS E DISCUSSÃO

O protótipo consegue ler a altura do nível da água através de uma equação que envolve a velocidade do som explicada na fundamentação teórica do presente trabalho de conclusão de curso. O mesmo conseguiu passar na POC (*Proof of Concept* ou Prova de Conceito), o que será demonstrado a seguir. E, também, será calculado a precisão do protótipo comparando o valor lido por ele com o valor real lido por uma régua do nível da água. E utilizando a plataforma Arduino® e a *Firebase* do Google® conseguiu-se mostrar a altura do nível da água através de um website hospedado na *Firebase*.

Foi criado um website, através da plataforma *Firebase* do Google®, para monitoramento do nível de corpos d'água. O nome do website é o mesmo do sensor: “Sonus Velox” – que é a combinação das palavras em latim “Sonus” (som) com “Velox” (veloz). A página inicial do mesmo está na Figura 16 (a seguir).

Figura 16 – Página inicial do website

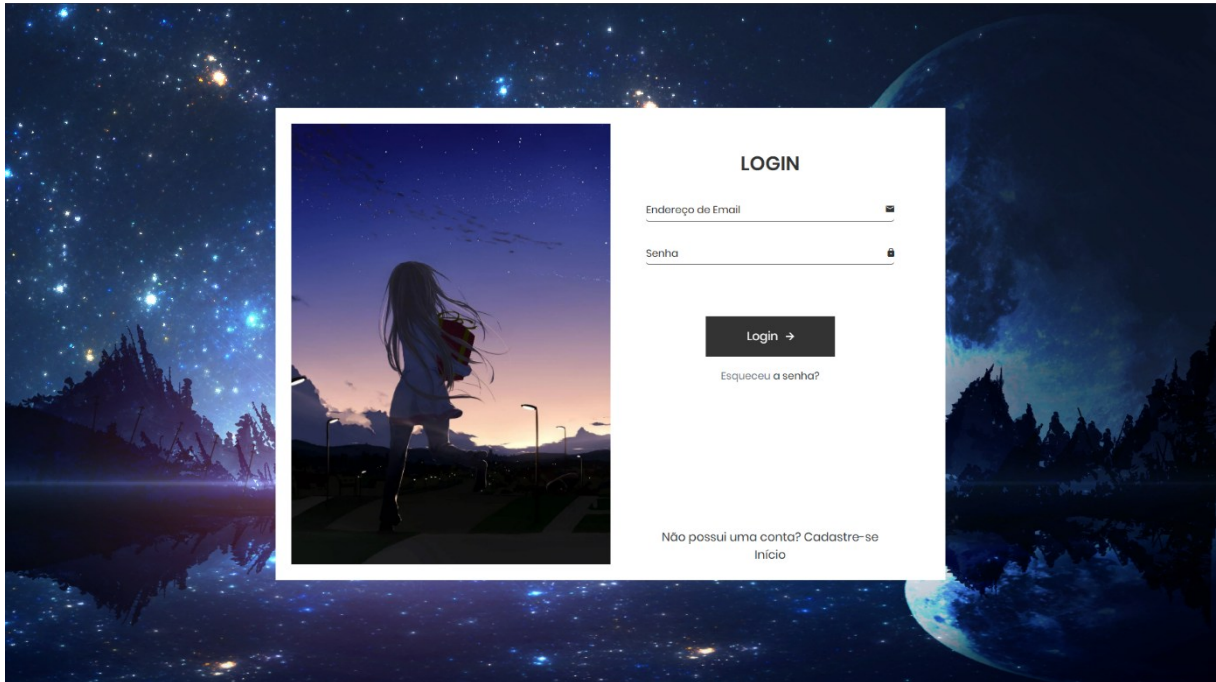


Fonte: elaborado pelo autor

Incluindo também a funcionalidade de enviar um e-mail ao usuário quando a altura do nível da água ultrapassar determinado valor escolhido pelo mesmo. E também, espera-se determinar a acurácia do protótipo criado.

Foram implementadas as funcionalidades de autenticação do usuário (login, cadastro e recuperação de senha) utilizando a *Firebase*. A página de login está na Figura 17 (a seguir).

Figura 17 – Página de login do website

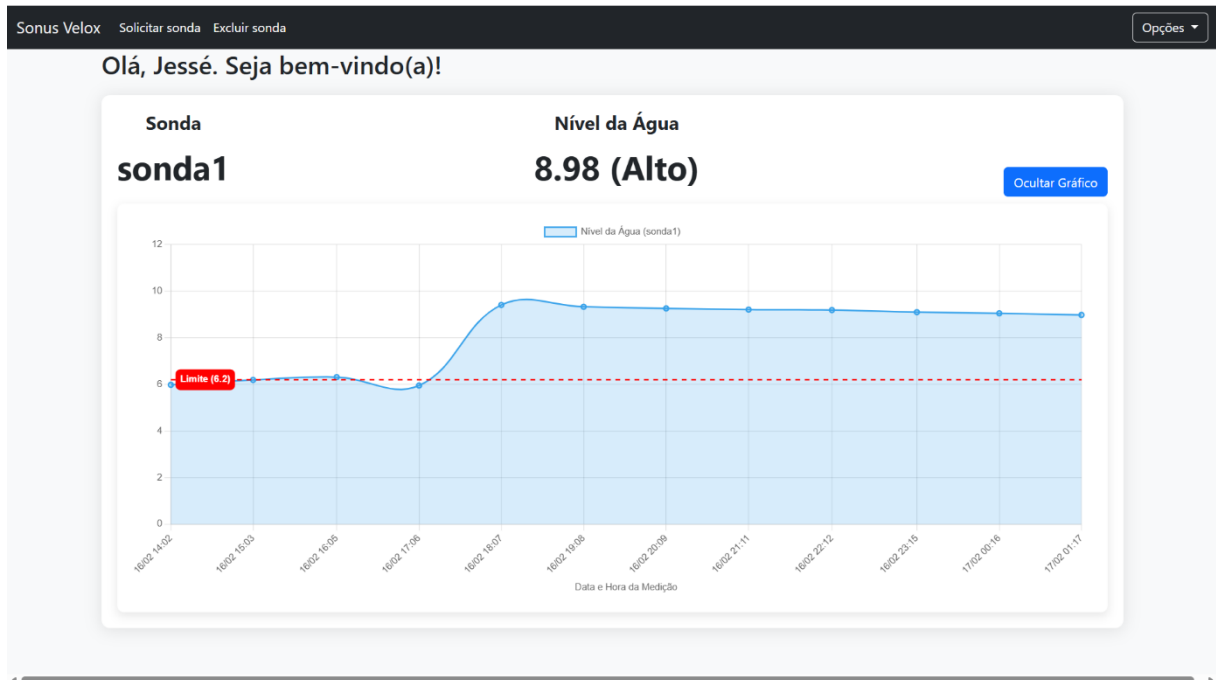


Fonte: elaborado pelo autor

Dessa forma o usuário pode entrar com o endereço de Email e senha que se cadastrou e assim visualizar as sondas (sensores ultrassônicos) que estão vinculadas a sua conta. O *dashboard* com os dados legítimos lidos pelo sensor ultrassônico utilizado no protótipo estão na Figura 18 (a seguir). Nota-se que o último nível registrado foi de 8.98 cm, logo está “Alto” porque está acima do limite definido pelo usuário (neste caso, 6.2 cm). O usuário recebe um e-mail de alerta de hora em hora quando for lido um nível acima do limite especificado por ele.

A seguir tem-se a prova de conceito e a demonstração da leitura do nível da água pelo sensor e calculado também a precisão do mesmo pelo erro relativo médio, inclusive comparando com a precisão do artigo de Sari *et al.* (2023) encontrado na revisão bibliográfica do presente trabalho.

Figura 18 – Dashboard do website com dados legítimos lidos pelo sensor



Fonte: elaborado pelo autor

4.1 PROVA DE CONCEITO

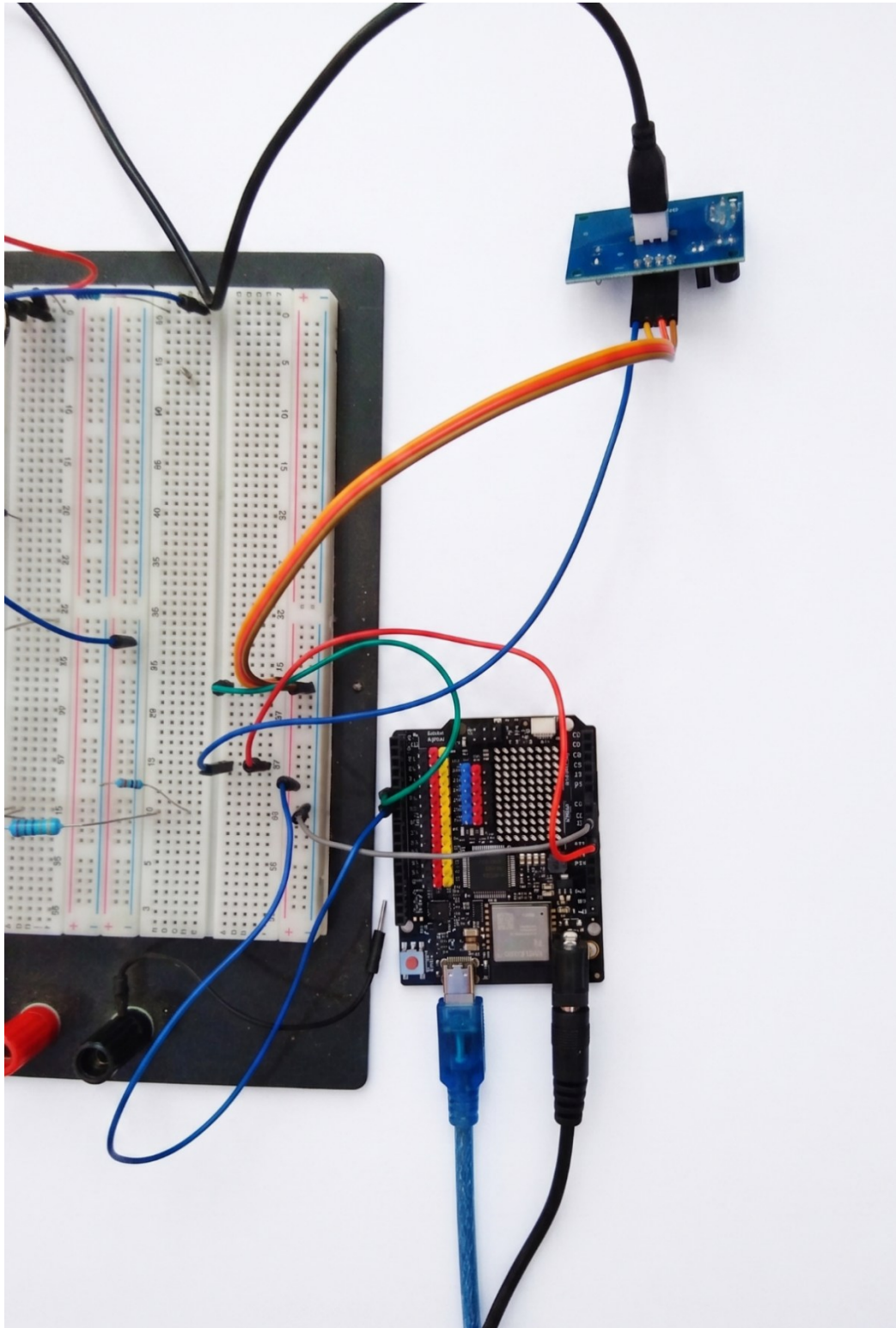
Primeiro atenta-se ao circuito montado na protoboard (Figura 19), o qual foi descrito anteriormente na metodologia.

Basicamente tem-se as seguintes conexões:

- Pino D6 conectado ao ECHO: o pino D6 da placa UNO R4 WiFi está conectado ao ECHO do sensor, o que significa que é esse pino que “lê” quanto tempo demorou para a onda ultrassônica ir e voltar até o sensor ultrassônico.
- Pino D5 conectado ao TRIGGER: o pino D5 da placa UNO R4 WiFi está conectado ao TRIGGER do sensor, o que significa que é por este pino que é emitida a onda ultrassônica.
- Pino GND conectado ao GND: o pino GND (Ground) da placa UNO R4 WiFi está conectado ao GND do sensor, ou seja, o pino terra do sensor está conectado ao pino terra da placa permitindo que o sistema funcione corretamente.

- Pino 5V conectado ao VCC: o pino 5V (5 Volts) da placa UNO R4 WiFi está conectado ao VCC do sensor, o que permite que o sensor ultrassônico tenha a alimentação necessária para seu funcionamento.

Figura 19 – Circuito do Protótipo montado na protoboard



Fonte: elaborado pelo autor

Primeiramente, mediu-se o nível da água do pote de acrílico com uma régua milimétrica e encontrou-se uma altura da água de 6 cm, conforme pode ser visto a seguir (Figura 20).

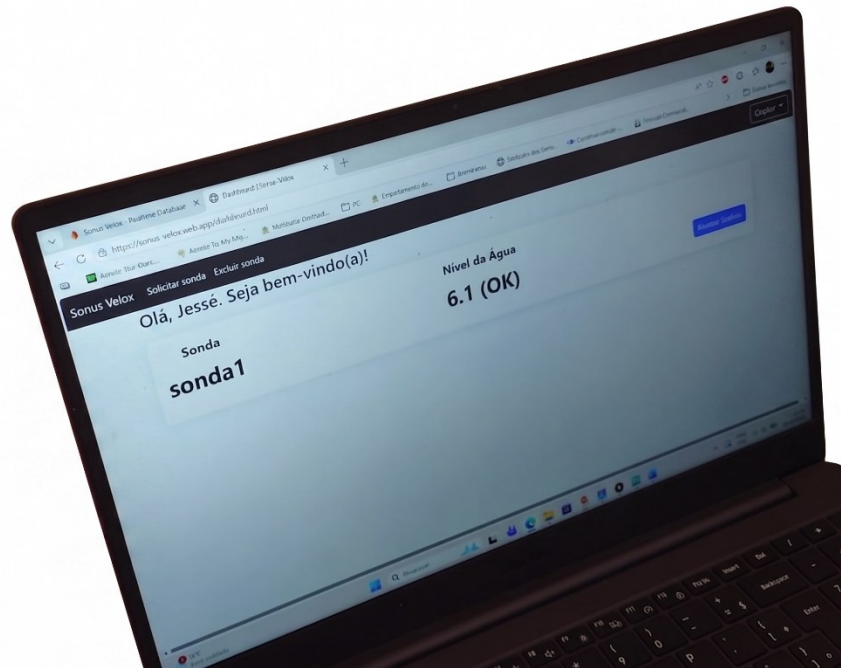
Figura 20 – Primeira medição do nível da água com a régua: 6 cm



Fonte: elaborado pelo autor

E ao medir o mesmo nível da água com o sensor ultrassônico utilizado no protótipo encontrou-se um valor bem aproximado, mais especificamente 6.1 cm, conforme pode ser visto na Figura 21 a seguir, na qual o valor é exibido no website de monitoramento criado.

Figura 21 – Primeira medição do nível da água com o sensor: 6.1 cm



Fonte: elaborado pelo autor

Foi também registrado a temperatura (em graus Celsius), a pressão atmosférica e a hora e o dia da medição do nível da água pelo sensor ultrassônico. O dia foi 16/02/2026 e a hora foi 12:28. O valor real lido pela régua foi de 6 cm. O valor lido pelo sensor foi de 6.1 cm. A temperatura foi de 29.4 °C. E a pressão atmosférica foi de 1011.7 hPa. Conforme pode ser visto na Figura 22 (a seguir).

Figura 22 – Hora, temperatura e pressão atmosférica da primeira medição



Fonte: elaborado pelo autor

Em síntese, o protótipo passou na prova de conceito, pois foi demonstrado que ele consegue ler o nível da água com determinada precisão.

4.2 DEMONSTRAÇÃO DO NÍVEL DA ÁGUA LIDO PELO SENSOR

A seguir tem-se a Tabela 2 que demonstra os valores do nível da água lido pelo sensor ultrassônico no ano de 2026 (o gráfico referente a mesma pode ser visto na Figura 18).

Tabela 2 – Valores lidos pelo sensor ultrassônico no protótipo

Nº	Valor real	Valor sensor	Temp.	Pressão atmosférica	Dia	Hora
#1	6 cm	5,98 cm	31,3 °C	1009,2 hPa	16/02/2026	14:02:54
#2	6 cm	6,19 cm	30,5 °C	1009,1 hPa	16/02/2026	15:03:50
#3	6 cm	6,31 cm	30,5 °C	1009,9 hPa	16/02/2026	16:05:48
#4	6 cm	5,95 cm	30,6 °C	1010,2 hPa	16/02/2026	17:06:48
#5	9,5 cm	9,41 cm	30,6 °C	1010,5 hPa	16/02/2026	18:07:48
#6	9,5 cm	9,33 cm	30,3 °C	1010,4 hPa	16/02/2026	19:08:48
#7	9,5 cm	9,26 cm	29,9 °C	1011,7 hPa	16/02/2026	20:09:58
#8	9,5 cm	9,21 cm	29,7 °C	1012,4 hPa	16/02/2026	21:11:03
#9	9,5 cm	9,19 cm	29,3 °C	1012,4 hPa	16/02/2026	22:12:48
#10	9,5 cm	9,1 cm	29,6 °C	1012,2 hPa	16/02/2026	23:15:46
#11	9,5 cm	9,05 cm	28,4 °C	1012,0 hPa	17/02/2026	00:16:50
#12	9,5 cm	8,98 cm	27,9 °C	1011,5 hPa	17/02/2026	01:17:44

Fonte: elaborado pelo autor

E, também, conforme pode ser visto na Figura 23 (a seguir), foi capturado o momento em que o *listener* desenvolvido “grava” o primeiro valor do nível da água da Tabela 2 lido pelo sensor ultrassônico no histórico da sonda1 na RTDB da *Firebase* (banco de dados em tempo real utilizado no website para armazenamento do valor lido).

Figura 23 – Primeiro valor lido pelo sensor na Tabela 2 (listener)

```
16:58 ~/fire_monitor_sonus_velox $ python monitor_fire_2.py
Conexão com o Firebase estabelecida com sucesso!
Monitoramento iniciado. Escutando por alterações no Firebase...
Processando: UID=[redacted], Sonda=sonda1, Altura=5.98 cm
AVISO: A sonda 'sonda1' não possui nenhum registro em 'histsonda'.
Registro de histórico adicionado para a sonda 'sonda1' no ts: 1771261374580
Total de registros (1) está dentro do limite de 25.
-> Altura 5.98 cm está abaixo do limite de 18.0. Nenhuma ação.
```

Fonte: elaborado pelo autor

E, pode-se ver na Figura 24 (a seguir), o momento em que o *listener* feito “grava” o terceiro valor do nível da água da Tabela 2 lido pelo sensor ultrassônico no histórico da sonda1 na RTDB da *Firebase*. É possível ver também na Figura 24 que o *listener* envia um e-mail de alerta para o usuário, pois o valor lido da sonda1 (6.31 cm) ultrapassa o limite definido pelo usuário que é de 6.2 cm.

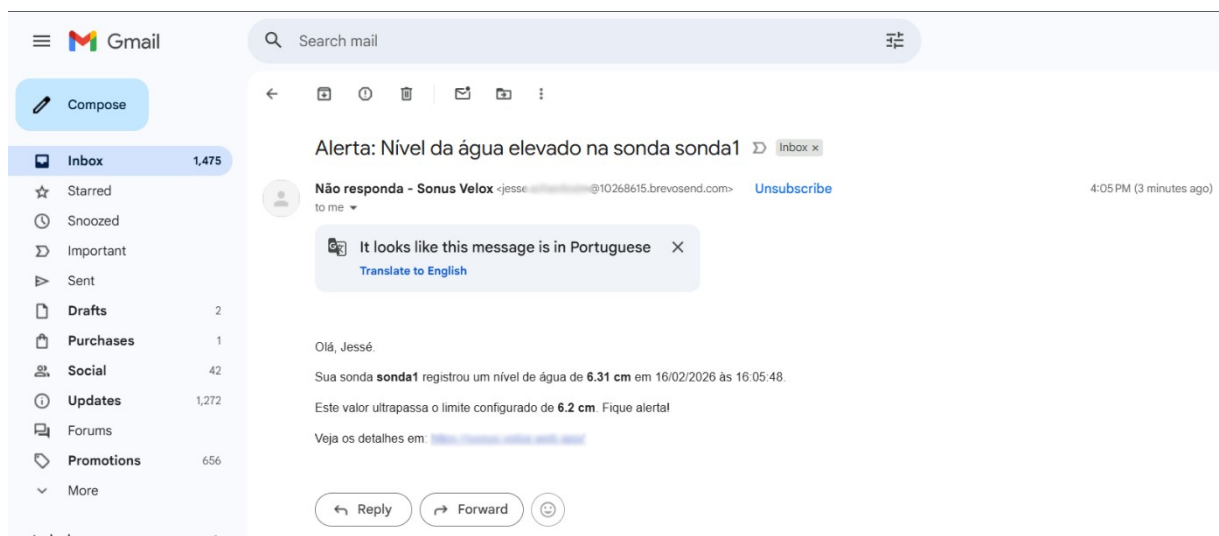
Figura 24 – Terceiro valor lido pelo sensor na Tabela 2 (listener)

```
18:56 ~/fire_monitor_sonus_velox $ python monitor_fire_2.py
Conexão com o Firebase estabelecida com sucesso!
Monitoramento iniciado. Escutando por alterações no Firebase...
Processando: UID=[redacted], Sonda=sonda1, Altura=6.31 cm
Registro de histórico adicionado para a sonda 'sonda1' no ts: 1771268748276
Total de registros (3) está dentro do limite de 25.
-> ALERTA: Altura 6.31 cm ultrapassou o limite de 6.2 cm.
-> E-mail de alerta enviado com sucesso para: j[redacted]@gmail.com via API Brevo.
-> Registro 'avisolimite' atualizado no Firebase.
```

Fonte: elaborado pelo autor

A seguir tem-se o e-mail recebido pelo usuário (Figura 25) conforme descrito anteriormente.

Figura 25 – E-mail recebido pelo usuário do terceiro valor lido (Tabela 2)



Fonte: elaborado pelo autor

A seguir (Figura 26) está a leitura do nível da água pela régua (valor real) do quinto valor em diante da Tabela 2. A leitura do valor real pela régua dos quatros primeiros valores da Tabela 2 está na Figura 20.

Figura 26 – Segunda medição do nível da água com a régua: 9,5 cm



Fonte: elaborado pelo autor

4.3 CÁLCULO DA ACURÁCIA DO SENSOR

Conforme descrito na metodologia foram calculados o Erro Absoluto Médio, o Erro de Viés Médio e o Erro Relativo Médio para verificar a precisão do sensor instalado. Segue os cálculos adiante baseados nos valores da Tabela 2.

4.3.1 Erro Absoluto Médio

Seguindo a equação descrita na fundamentação teórica para calcular o Erro Absoluto Médio e usando os valores da Tabela 2, temos:

$$MAE = \frac{1}{12} (|-0,02| + |0,19| + |0,31| + |-0,05| + |-0,09| + |-0,17| + |-0,24| \\ + |-0,29| + |-0,31| + |-0,4| + |-0,45| + |-0,52|)$$

O que equivale a:

$$MAE = \frac{1}{12} (0,02 + 0,19 + 0,31 + 0,05 + 0,09 + 0,17 + 0,24 + 0,29 + 0,31 + 0,4 + 0,45 \\ + 0,52)$$

Resultando em:

$$MAE = \frac{1}{12} (3,04)$$

Ou seja:

$$MAE = 0,253333 \dots$$

Em que,

MAE é o Erro Absoluto Médio.

Portanto, temos um erro absoluto médio de 0,25333. Isso significa que o sensor instalado tende a ler um valor com uma divergência média de 0,253 cm em relação à altura real da água, representando a margem média de oscilação absoluta do protótipo durante as leituras. E isso se deve principalmente ao fato de o sensor ultrassônico estar levemente inclinado quando foram feitas as leituras do nível da água por ele, conforme pode ser visto na Figura 27 a seguir.

Figura 27 – Sensor ultrassônico utilizado e sua inclinação



Fonte: elaborado pelo autor

Caso fosse feito o experimento em condições mais ideais, o erro absoluto médio seria menor.

4.3.2 Erro de Viés Médio

Agora, seguindo a equação descrita na fundamentação teórica para calcular o Erro de Viés Médio e usando os valores da Tabela 2, temos:

$$MBE = \frac{1}{12} (-0,02 + 0,19 + 0,31 - 0,05 - 0,09 - 0,17 - 0,24 - 0,29 - 0,31 - 0,4 - 0,45 - 0,52)$$

Assim, obtêm-se:

$$MBE = \frac{1}{12}(-2,04)$$

O que resulta em:

$$MBE = -0,17$$

Em que,

MBE é o Erro de Viés Médio.

O erro de viés médio calculado mostra que a sonda tende subestimar em aproximadamente 0,17 cm o valor real da altura da água. O que indica uma boa precisão, mesmo com o sensor estando inclinado durante as leituras. Pois, segundo Gomes (2011, p. 53) referente ao rio Araranguá: “As enchentes acontecem com magnitudes diferentes e esse fator foi considerado no planejamento deste levantamento. Para tanto foi realizado uma divisão dos pontos vulneráveis em 4 níveis de intensidade das enchentes”:

- Nível 1 - até 3,0 metros: enchente de pequena proporção;
- Nível 2 - de 3,0 a 3,5 metros: enchente média proporção;
- Nível 3 – de 3,5 a 4,0 metros: enchente de grande proporção;
- Nível 4 – de 4,0 a 5,0 metros: enchente de enorme proporção.

(Gomes, 2011, p. 53)

O que denota que a variação de alerta para níveis de enchente ocorre de 0,5 metro (50 centímetros) a 1,0 metro (100 centímetros), então o sensor subestimar o valor real da altura da água em 0,17 cm é uma precisão mais do que aceitável, pode-se dizer ótima para fins de alerta de enchentes.

4.3.3 Erro Relativo e Erro Relativo Médio

Agora, seguindo a equação descrita na fundamentação teórica para calcular o Erro Relativo e usando os valores da prova de conceito, temos primeiramente que calcular o erro absoluto.

Segue o erro absoluto da prova de conceito:

$$E_a = |6.1 - 6|$$

Ou seja:

$$E_a = 0,1$$

Em que,

E_a é o Erro Absoluto (cm).

Agora segue o erro relativo da prova de conceito:

$$E_r = \frac{0,1}{6} \times 100\%$$

O que resulta em:

$$E_r \cong 1,67\%$$

Em que,

E_r é o Erro Relativo (%).

No entanto, esse erro relativo supracitado é apenas para a medição feita na prova de conceito do protótipo. Para se ter uma acurácia melhor do protótipo é preciso calcular o erro relativo médio para todas as medições do experimento. Com base na equação do erro relativo médio ou Mean Relative Error (MRE) explicitada na fundamentação teórica e para as 12 medições da Tabela 2, temos:

$$MRE = \frac{1}{12} \left(\frac{|-0,02|}{6} + \frac{|0,19|}{6} + \frac{|0,31|}{6} + \frac{|-0,05|}{6} + \frac{|-0,09|}{9,5} + \frac{|-0,17|}{9,5} + \frac{|-0,24|}{9,5} + \frac{|-0,29|}{9,5} + \frac{|-0,31|}{9,5} + \frac{|-0,4|}{9,5} + \frac{|-0,45|}{9,5} + \frac{|-0,52|}{9,5} \right) \times 100\%$$

Calculando obtêm-se:

$$MRE = \frac{1}{12} (0,355) \times 100\%$$

O que resulta em:

$$MRE \cong 2,9583\%$$

Em que,

MRE é o Erro Relativo Médio (%).

O que significa que o erro relativo médio do protótipo é de aproximadamente 2,96%, indicando ser um sensor bem preciso para a leitura do nível da água mesmo

nas condições em que o presente experimento foi realizado. Pois, conforme Gomes (2011) a variação de níveis de alerta da proporção de enchentes ocorre entre 50 e 100 centímetros e como o erro relativo médio do protótipo é de 2,96%, mesmo que estivéssemos medindo 800 cm (leitura máxima do sensor ultrassônico) teríamos um erro máximo de 23,68 cm, o que é aceitável para fins de alerta de enchentes.

4.4 COMPARAÇÃO DO EXPERIMENTO COM A LITERATURA

Atenta-se agora aos valores do erro relativo dos sensores utilizados no artigo de Sari *et al.* (2023) encontrado na revisão bibliográfica (que são: US-100, HY-SRF05 e JSN-SR04T) para fins de comparação com o erro relativo encontrado no protótipo da presente pesquisa aplicada (na qual foi utilizado o sensor ultrassônico AJ-SR04M). Esse artigo foi encontrado na pesquisa bibliográfica com as seguintes *keywords*: “ultrasonic sensor” + “water level” + “measurement”, no IEEE Xplore, intitulado “Assessment of External Factors on the Accuracy of Ultrasonic Sensors for Water Level Measurement in Peatlands”, ou seja, “Avaliação de fatores externos na precisão de Sensores ultrassônicos para medição do nível da água em Turfeiras”. Que basicamente, segundo Sari *et al.* (2023, p. 1, tradução nossa): “Este estudo visa determinar o efeito de fatores externos na precisão dos sensores ultrassônicos HY-SRF05, JSN-SR04T e US-100”.

Os resultados de precisão para a medição do nível da água indicam que o sensor ultrassônico US-100 é mais preciso do que os sensores HY-SRF05 e JSN-SR04T. O valor do erro relativo do US-100 é de 0,93%, do HY-SRF05 é de 2,13% e do JSN-SR04T é de 2,83% (Sari *et al.*, 2023, p. 1, tradução nossa)

O artigo de Sari *et al.* (2023) serve como base para comparar a precisão dos sensores utilizados no mesmo com a do sensor e sistema da presente pesquisa. Avaliando, assim, os pontos fortes e fracos de cada um e as circunstâncias em que foram usados para medição do nível da água e claro o desempenho e as vantagens de cada um dos sistemas apresentados, com isso temos a tabela de comparação a seguir:

Tabela 3 – Comparação da precisão dos sensores da literatura e do protótipo desenvolvido

Sensor	Erro Relativo
US-100	0,93%
HY-SRF05	2,13%
JSN-SR04T	2,83%
AJ-SR04M (protótipo)	2,96%

Fonte: Adaptado de Sari *et al.* (2023).

Dessa forma, pode-se ver que o sensor ultrassônico utilizado na presente pesquisa aplicada apresentou um erro relativo médio um pouco maior do que o erro relativo dos sensores utilizados no artigo de Sari *et al.* (2023). Isso se deve ao fato do sensor da presente pesquisa aplicada (protótipo) estar um pouco inclinado durante as leituras do nível da água pelo mesmo. Ainda assim, o sistema embarcado desenvolvido obteve um erro relativo próximo à maioria dos outros sensores ao qual foi comparado.

Com isso, tem-se um protótipo funcional de um sistema embarcado utilizando sensor ultrassônico e que cumpre todos os objetivos da pesquisa.

5 CONCLUSÃO

Este trabalho de conclusão de curso consolidou com êxito as bases teóricas e metodológicas para o desenvolvimento de um protótipo de sistema embarcado utilizando sensor ultrassônico com o intuito de monitorar o nível de corpos d'água em tempo real e emitir alertas para mitigar o impacto de enchentes na população em geral. Ou seja, foi possível desenvolver o sistema embarcado proposto, no entanto a precisão do mesmo ficou um pouco prejudicada porque no experimento o sensor ultrassônico estava um pouco inclinado. Na fase de implementação, utilizou-se as bibliotecas *FirestoreClient* (Suwatchai, 2026b) e *ESP_SSLClient* (Suwatchai, 2026a) na plataforma Arduino® para enviar os dados lidos pelo sensor ultrassônico para *Realtime Database* do *Firestore* do Google®. E para medir a distância do sensor até a superfície da água foi utilizado o sensor ultrassônico AJ-SR04M ligado a uma placa compatível com Arduino®. Mais especificamente, a Uno R4 WiFi – usando sua biblioteca oficial na plataforma Arduino®, permitindo usar as funções do módulo Wi-Fi ESP32 e do seu microcontrolador que vem integrados nela. E, dessa forma, calcular o nível da água utilizando a leitura do sensor ultrassônico. Possibilitando mostrar, assim, o valor do nível da água num website hospedado pela *Firestore* do Google® com um gráfico de onda para melhor visualização. E, inclusive, foi desenvolvido um *listener* assíncrono que grava os valores lidos pelo sensor ultrassônico de hora em hora no gráfico de onda da sonda. E que também envia um e-mail ao usuário caso o valor lido pelo sensor ultrassônico ultrapassar o limite definido por ele.

As principais limitações incluem que o sensor consegue ler uma distância máxima de 8 metros e que a placa utilizada UNO R4 WiFi depende de fonte de energia e Wi-Fi próximos para seu funcionamento. No entanto, isto nem chega a ser uma grande limitação porque para fins de alerta de enchentes 8 metros de aumento do nível de um rio, por exemplo, seria uma enchente de grandíssima proporção, então para fins de alerta de enchentes este intervalo de leitura do nível da água é mais que suficiente, pode-se dizer ótimo. Quanto aos custos para implementação em larga escala, conforme visto na Metodologia do presente trabalho, cada sonda custaria menos de 300 reais para sua implementação para leitura do nível de um rio, por exemplo. E todas poderiam usar o mesmo website e sistema de alerta, no caso do presente trabalho o website "Sonus Velox", e o sistema de alerta seria o envio de e-

mail ao usuário, possibilitando privacidade e segurança com o usuário e senha vinculados a cada sonda quando configurada na sua respectiva instalação.

Em resumo, foi construído um protótipo funcional de um sistema embarcado utilizando sensor ultrassônico, um website para monitoramento e um *listener* assíncrono conforme supramencionado. O protótipo passou na prova de conceito, e quando calculado o erro relativo médio segundo as leituras obtidas no experimento obteve-se apenas 2,96%. O que denota ser um sensor bem preciso, pois segundo Gomes (2011) as variações de níveis de alerta de enchentes ocorrem entre 50 e 100 centímetros e com 2,96% de erro relativo temos um erro máximo de 23,68 cm para a leitura máxima do sensor (800 cm), tornando-o um sensor adequado para este fim. Ou seja, a presente pesquisa alcançou com êxito todos os objetivos propostos. E, para trabalhos futuros espera-se que o protótipo criado sirva como base para os mesmos e que seja melhorado a acurácia do protótipo do sistema embarcado criado.

REFERÊNCIAS

AMAZON WEB SERVICES. **O que é um servidor dedicado?** [S. l.]: Amazon Web Services, [s. d.]. Disponível em: <https://aws.amazon.com/pt/what-is/dedicated-server/>. Acesso em: 29 jun. 2025.

ARARANGUÁ. Prefeitura Municipal. **Novas réguas para monitoramento do Rio Araranguá**. Araranguá, 20 dez. 2023. Disponível em: <https://ararangua.sc.gov.br/novas-reguas-para-monitoramento-do-rio-ararangua/>. Acesso em: 17 abr. 2025.

ARARANGUÁ. Prefeitura Municipal. **Recuperação e preservação do Rio Araranguá**. Araranguá, 26 nov. 2024. Disponível em: <https://ararangua.sc.gov.br/recuperacao-e-preservacao-do-rio-ararangua/>. Acesso em: 29 maio 2026.

ARDUINO. **Arduino® UNO R4 WiFi**. [S. l.], [202-?]. Disponível em: <https://store-usa.arduino.cc/products/uno-r4-wifi>. Acesso em: 25 fev. 2026.

ARDUINO Distance Measuring Transducer Sensor Model: AJ-SR04M. [S. l.: s. n.], [20--?]. Disponível em: <https://www.fabian.com.mt/viewer/42586/pdf.pdf>. Acesso em: 27 maio 2026.

ARDUINO. **UNO R4 WiFi**. [S. l.], 2026. Disponível em: <https://docs.arduino.cc/hardware/uno-r4-wifi/>. Acesso em: 14 jan. 2026.

BREVO. **Brevo API**: technical documentation. [S. l.], 2026. Disponível em: <https://developers.brevo.com/>. Acesso em: 16 fev. 2026.

BURCH, David. **Erro absoluto médio no aprendizado de máquina: o que você precisa saber**. [S. l.], 2023. Disponível em: <https://arize.com/blog-course/mean-absolute-error-in-machine-learning-what-you-need-to-know/>. Acesso em: 11 jun. 2025.

CHART.JS. **chartjs-adapter-luxon**. [S. l.]: GitHub, [20--]. Disponível em: <https://github.com/chartjs/chartjs-adapter-luxon>. Acesso em: 26 jan. 2026.

CHART.JS. **chartjs-plugin-annotation: Guide**. [S. l.], 2026. Disponível em: <https://www.chartjs.org/chartjs-plugin-annotation/latest/guide/>. Acesso em: 26 jan. 2026.

CHE ZULKIFLI, Che Ahmad Hafizie Bin. Development of Flood Notification System. **Journal of Applied Engineering & Technology (JAET)**, [S. l.], v. 3, n. 2, p. 17–28, 2019. DOI: 10.55447/jaet.03.02.18. Disponível em: <https://jae-tech.com/index.php/jaet/article/view/18>. Acesso em: 26 maio 2026.

CRESWELL, John W. **Projeto de pesquisa**: métodos qualitativo, quantitativo e misto. 3. ed. Porto Alegre: Artmed, 2010.

DIGITAL CONCEPTS. **AJ-SR04M Ultrasonic Distance Sensor**. [S. l.], 2026. Disponível em: <https://www.digitalconcepts.net.au/arduino/?op=Ultrasonic>. Acesso em: 27 maio 2026.

FALEIRO, Giovana Teodora de Jesus; FERREIRA, Gabriela Calafate; CASTRO, Felipe Abdala Rumanos de; VALÉRIO, Antonio Batista Fernandes de; ARMOND, Núbia Beray. **IMPACTOS SOCIOAMBIENTAIS E ESTRATÉGIAS DE ADAPTAÇÃO ÀS ENCHENTES E INUNDAÇÕES: AS EXPERIÊNCIAS DOS RESIDENTES DA BACIA DO RIO JEQUITINHONHA**. Caminhos de Geografia, Uberlândia, v. 25, n. 101, p. 95–111, 2024. DOI: 10.14393/RCG2510171663. Disponível em: <https://seer.ufu.br/index.php/caminhosdegeografia/article/view/71663>. Acesso em: 17 abr. 2025.

FERNÁNDEZ-GARCÍA, A.; SUTTER, F.; FERNÁNDEZ-RECHE, J.; LÜPFERT, E. Mirrors. In: HELLER, Peter (org.). **The performance of concentrated solar power (CSP) systems: analysis, measurement and assessment**. Duxford: Woodhead Publishing, 2017. cap. 3, p. 67-98. Disponível em: <https://doi.org/10.1016/B978-0-08-100447-0.00003-1>. Acesso em: 04 fev. 2026.

FIREBASE. **Estruturar seu banco de dados**. [S. l.], [2025]. Disponível em: <https://firebase-dot-devsite-v2-prod.appspot.com/docs/database/web/structure-data?hl=pt&authuser=0>. Acesso em: 12 jun. 2025.

FIREBASE. **Recuperar dados**. [S. l.], [2026]. Disponível em: <https://firebase.google.com/docs/database/admin/retrieve-data?hl=pt-br>. Acesso em: 03 fev. 2026.

FREITAS JUNIOR, Vanderlei; CECI, Flavio; WOSZEZENKI, Cristiane Raquel; GONÇALVES, Alexandre Leopoldo. Design Science Research Methodology enquanto estratégia metodológica para a pesquisa tecnológica. **Revista Espacios**, v. 38, n. 6, p. 25, 2017. Disponível em: <https://www.revistaespacios.com/a17v38n06/17380625.html>. Acesso em: 1 jul. 2026.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2008.

GOMES, Rodolfo Batista. **Levantamento, análise de dados e de fatores condicionantes para a ocorrência das enchentes com proposta de medidas mitigadoras para a cidade de Araranguá**. 2011. 107 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Ambiental) – Universidade do Extremo Sul Catarinense, Criciúma, 2011. Disponível em: <https://repositorio.unesc.net/handle/1/1359>. Acesso em: 31 maio 2026.

JAPPUR, R. F. **Modelo conceitual para criação, aplicação e avaliação de jogos educativos digitais**. Universidade Federal de Santa Catarina. Florianópolis, SC. 2014.

JAMIL, Basharat; AKHTAR, Naiem. Comparative analysis of diffuse solar radiation models based on sky-clearness index and sunshine period for humid-subtropical

climatic region of India: a case study. **Renewable And Sustainable Energy Reviews**, [S. l.], v. 78, p. 329-355, out. 2017. Elsevier BV. <http://dx.doi.org/10.1016/j.rser.2017.04.073>.

MAGNO, Rafael. **Bootstrap**: O que é e por que ele ainda é tão usado?. [S. l.], 2025. Disponível em: <https://www.dio.me/articles/bootstrap-o-que-e-e-por-que-ele-ainda-e-tao-usado-2175a2891523>. Acesso em: 16 jan. 2026.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. **Fundamentos de metodologia científica**. 5. ed. São Paulo: Atlas, 2003.

NILES, Djalma Santos. **Eventos de inundação em Araranguá**: tipos de ocorrências. 2009. 65 f. Trabalho de Conclusão de Curso (Graduação em Geografia) - Universidade do Extremo Sul Catarinense, Criciúma, 2009. Disponível em: <https://bib.unesc.net/acervo/92582/>. Acesso em: 29 maio 2026.

ORACLE. **O que é NoSQL?** Austin, 2021. Disponível em: <https://www.oracle.com/br/database/nosql/what-is-nosql/>. Acesso em: 07 jun. 2025.

ORGANIZAÇÃO DAS NAÇÕES UNIDAS (ONU). **Agenda 2030 para o Desenvolvimento Sustentável**. [S. l.]: ONU, 2015. Disponível em: <https://brasil.un.org/pt-br/91863-agenda-2030-para-o-desenvolvimento-sustentavel>. Acesso em: 17 abr. 2025.

OSHANA, Robert. Overview of embedded systems and real-time systems. In: OSHANA, Robert. **DSP software development techniques for embedded and real-time systems**. Oxford: Newnes, 2006. p. 19-34. (Embedded Technology). Disponível em: <https://www.sciencedirect.com/science/article/pii/B9780750677592500041>. Acesso em: 05 fev. 2026

PARAJULI, Alsan. **Waterproof Ultrasonic Sensor with Arduino to Measure Water Level**. 2024. Disponível em: <https://iotprojectsideas.com/waterproof-ultrasonic-sensor-with-arduino-to-measure-water-level/>. Acesso em: 05 jan. 2026.

PERMATASARI, Yessy *et al.* Development of IoT Control System Prototype for Flood Prevention in Bandung Area. **JOIV : International Journal on Informatics Visualization**, v. 7, n. 3, p. 1016-1021, 2023. DOI: 10.30630/joiv.7.3.2083. Disponível em: <https://doi.org/10.30630/joiv.7.3.2083>. Acesso em: 27 maio 2026.

POLI, Cláudia Maria Basso. As causas e as formas de prevenção sustentáveis das enchentes urbanas. **II Seminário Nacional de Construções Sustentáveis**, 2013.

POWLES, Alexander EJ.; MARTIN, David J.; WELLS, Irving TP.; GOODWIN, Christopher R. Physics of ultrasound. **Anaesthesia & Intensive Care Medicine**, v. 19, n. 4, p. 202-205, 2018. Disponível em: <https://doi.org/10.1016/j.mpaic.2018.01.005>. Acesso em: 04 fev. 2026.

PURVES, Dale; AUGUSTINE, George J.; FITZPATRICK, David; KATZ, Lawrence C.; LAMANTIA, Anthony-Samuel; MCNAMARA, James O.; WILLIAMS, S. Mark. The

audible spectrum. *In*: PURVES, D. *et al.* (Ed.). **Neuroscience**. 2. ed. Sunderland (MA): Sinauer Associates, 2001. Disponível em: <https://www.ncbi.nlm.nih.gov/books/NBK10924/>. Acesso em: 06 jan. 2026.

RIBEIRO, Andre Louis Souza. **Entendendo o Firebase e suas principais funcionalidades**. [S. l.], 2023. Disponível em: <https://www.alura.com.br/artigos/entendendo-firebase-principais-funcionalidades>. Acesso em: 07 jun. 2025.

SARI, Yuslena; ARIFIN, Yudi Firmanul; NOVITASARI, Novitasari; PRATAMA, Ferry; WIJAYA, Eka Setya; PRAMUNENDAR, Ricardus Anggi; MAULIDA, Mutia; MUSTAMIN, Nurul Fathanah; BASKARA, Andreyan Rizky; MAULIDIYA, Erika. Assessment of External Factors on the Accuracy of Ultrasonic Sensors for Water Level Measurement in Peatlands. *In*: INTERNATIONAL CONFERENCE ON INFORMATICS AND COMPUTING (ICIC), 8., 2023, Manado. **Proceedings...** [S.l.]: IEEE, 2023. p. 1-7. Disponível em: <https://doi.org/10.1109/ICIC60109.2023.10381918>. Acesso em: 14 jan. 2026.

SCHNEIDER, Josh; SMALLEY, Ian. **O que é um microcontrolador?** [S. l.], 2024. Disponível em: <https://www.ibm.com/br-pt/think/topics/microcontroller>. Acesso em: 07 jun. 2025.

SHUKLA, P. K.; TIWARI, S.; SINGH, D. K.; SINGH, A. FloodGuard: An Embedded System for Real-Time Monitoring and Flood Management System. *In*: INTERNATIONAL CONFERENCE ON ELECTRICAL AND ELECTRONICS ENGINEERING (ICE3), 2025, Gorakhpur. **Proceedings [...]**. [S. l.]: IEEE, 2025. p. 1-6. DOI: 10.1109/ICE367573.2025.11448841. Disponível em: <https://doi.org/10.1109/ICE367573.2025.11448841>. Acesso em: 27 maio 2026.

SIMATUPANG, J. W.; ARRAZAQ, T. A. Design Implementation of Flood Early Warning System for Residential Monitoring. *In*: INTERNATIONAL CONFERENCE ON RADAR, ANTENNA, MICROWAVE, ELECTRONICS, AND TELECOMMUNICATIONS (ICRAMET), 2023, Bandung. **Proceedings [...]**. [S. l.]: IEEE, 2023. p. 158-163. Disponível em: <https://doi.org/10.1109/ICRAMET60171.2023.10366589>. Acesso em: 26 maio 2026.

SOUZA, Fábio. **O que são sistemas embarcados?** [S. l.], 2022. Disponível em: <https://embarcados.com.br/o-que-sao-sistemas-embarcados/>. Acesso em: 07 jun. 2025.

SUWATCHAI, K. **ESP_SSLClient**. [S. l.]: GitHub, 2026a. Disponível em: https://github.com/mobizt/ESP_SSLClient. Acesso em: 02 mar. 2026.

SUWATCHAI, K. **FirestoreClient**. [S. l.]: GitHub, 2026b. Disponível em: <https://github.com/mobizt/FirebaseClient>. Acesso em: 02 mar. 2026.

TANAKA, Katsuhiko; SUDA, Takefumi; HIRAI, Kazuhiro; SAKO, Kazunari; FUAKGAWA, Ryoichi; SHIMAMURA, Makoto; TOGARI, Asako. Monitoring of soil moisture and groundwater levels using ultrasonic waves to predict slope failures. *In*: IEEE SENSORS, 2009, Christchurch. **Proceedings [...]**. Christchurch: IEEE, 2009. p.

617-620. Disponível em: <https://doi.org/10.1109/ICSENS.2009.5398322>. Acesso em: 04 fev. 2026.

URONE, Paul Peter; HINRICHS, Roger. Ultrasound. *In*: URONE, Paul Peter; HINRICHS, Roger. **College Physics**. Houston: OpenStax, 2012. Cap. 17, sect. 7. Disponível em: <https://openstax.org/books/college-physics/pages/17-7-ultrasound>. Acesso em: 04 fev. 2026.

VIEIRA, Douglas Santos; ROCHA, Ana Victória Machado Vilela; VIEIRA, José Moacir de Sousa; MENDES, Rodolfo Moreda; VALÉRIO FILHO, Mario. **SEGURO PARA ENCHENTES E INUNDAÇÕES NO BRASIL: UM ESTUDO BASEADO EM INDICADORES INTERNACIONAIS**. Revista Políticas Públicas & Cidades, [S. l.], v. 14, n. 1, p. e1675, 2025. DOI: 10.23900/2359-1552v14n1-119-2025. Disponível em: <https://journalppc.com/RPPC/article/view/1675>. Acesso em: 17 abr. 2025.

W3SCHOOLS. **Chart.js**. [S. l.], [s. d.]. Disponível em: https://www.w3schools.com/ai/ai_chartjs.asp. Acesso em: 20 jan. 2026.

ZAKARIA, Muhammad Izzat; JABBAR, Waheb A.; SULAIMAN, Noorazliza. Development of a smart sensing unit for LoRaWAN-based IoT flood monitoring and warning system in catchment areas. **Internet of Things and Cyber-Physical Systems**, [S. l.], v. 3, p. 249-261, 2023. DOI: <https://doi.org/10.1016/j.iotcps.2023.04.005>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2667345223000263>. Acesso em: 27 maio 2026.

APÊNDICE A – CÓDIGO UNO R4 WIFI

```

/*
  Jesse Emerim Schardosim
  Codigo criado com base nos exemplos das bibliotecas e placas utilizadas
*/

#include <Arduino.h>
#include <math.h>

#include <WiFiS3.h>

#define ENABLE_USER_AUTH
#define ENABLE_DATABASE

#include <FirebaseClient.h>

#include "credenciais.h"

char ssid[] = WIFI_SSID;      // your network SSID (name)
char pass[] = WIFI_PASSWORD;  // your network password (use for WPA, or use as
key for WEP)
int status = WL_IDLE_STATUS;  // the WiFi radio's status

void processData(AsyncResult &aResult);

WiFiSSLClient ssl_client;

using AsyncClient = AsyncClientClass;
AsyncClient aClient(ssl_client);

UserAuth user_auth(API_KEY, USER_EMAIL, USER_PASSWORD, 3000 /* expire period in
seconds (<3600) */);
FirebaseApp app;
RealtimeDatabase Database;
AsyncResult databaseResult;

#define trigPin 5
#define echoPin 6

const float ALTURA = 90.0;

float h = 0.0;

unsigned long sendDataPrevMillis = 0;

long duracao;
float distancia;

bool signOK = false;
String uid;

bool uidCapturado = false;

```

```

void setup() {
    digitalWrite(trigPin, LOW);

    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    Serial.begin(115200);
    delay(50); // Intervalo para aguardar a estabilização do sistema

    if (WiFi.status() == WL_NO_MODULE) {
        Serial.println("Communication with WiFi module failed!");
        // don't continue
        while (true);
    }

    while (status != WL_CONNECTED) {
        Serial.print("Attempting to connect to WPA SSID: ");
        Serial.println(ssid);
        // Connect to WPA/WPA2 network:
        status = WiFi.begin(ssid, pass);

        // wait 10 seconds for connection:
        delay(10000);
    }
    Serial.println();

    // you're connected now, so print out the data:
    Serial.print("You're connected to the network");
    printCurrentNet();
    printWifiData();

    Firebase.printf("Firebase Client v%s\n", FIREBASE_CLIENT_VERSION);

    Serial.println("Initializing app...");
    initializeApp(aClient, app, getAuth(user_auth));

    // Or initialize the app and wait.
    // initializeApp(aClient, app, getAuth(user_auth), 120 * 1000,
    auth_debug_print);

    app.getApp<RealtimeDatabase>(Database);

    Database.url(DATABASE_URL);

    signOK = true;
}

void loop() {
    app.loop();

    if (app.ready() && !uidCapturado) {

        // PEGA O UID AQUI
        uid = app.getUid();

        // Verifica se veio um UID válido (não vazio)
        if (uid.length() > 0) {

```

```

        Serial.print("Usuário Logado! UID: ");
        Serial.println(uid);

        // Marca como capturado para não imprimir de novo
        uidCapturado = true;

        // Agora pode usar esse UID nos caminhos do banco
    }
}

if (app.ready() && signOK && uidCapturado && (millis() - sendDataPrevMillis >
15000 || sendDataPrevMillis == 0)) { // && (Serial.available() > 0)) {
    sendDataPrevMillis = millis();

    digitalWrite(trigPin, LOW);
    delayMicroseconds(5);

    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    duracao = pulseIn(echoPin, HIGH, 40000);

    distancia = duracao*0.0343/2;

    Serial.print("Distancia = ");
    Serial.print(distancia);
    Serial.println(" cm");

    h = ALTURA - distancia;

    // Set float
    Serial.println("Setting the float altura value... ");
    Database.set<number_t>(aClient, "/usuario/"+(uid)+"/sonda/sonda1",
number_t(h, 2), processData, "setAlturaTask");

    processData(databaseResult);
    delay(10000);
}

}

void processData(AsyncResult &aResult)
{
    // Exits when no result is available when calling from the loop.
    if (!aResult.isResult())
        return;

    if (aResult.isEvent())
    {
        Firebase.printf("Event task: %s, msg: %s, code: %d\n",
aResult.uid().c_str(), aResult.eventLog().message().c_str(),
aResult.eventLog().code());
    }

    if (aResult.isDebug())

```

```

    {
        Firebase.printf("Debug task: %s, msg: %s\n", aResult.uid().c_str(),
aResult.debug().c_str());
    }

    if (aResult.isError())
    {
        Firebase.printf("Error task: %s, msg: %s, code: %d\n",
aResult.uid().c_str(), aResult.error().message().c_str(), aResult.error().code());
    }

    if (aResult.available())
    {
        Firebase.printf("task: %s, payload: %s\n", aResult.uid().c_str(),
aResult.c_str());
    }
}

void printWifiData() {
    // print your board's IP address:
    IPAddress ip = WiFi.localIP();
    Serial.print("IP Address: ");

    Serial.println(ip);

    // print your MAC address:
    byte mac[6];
    WiFi.macAddress(mac);
    Serial.print("MAC address: ");
    printMacAddress(mac);
}

void printCurrentNet() {
    // print the SSID of the network you're attached to:
    Serial.print("SSID: ");
    Serial.println(WiFi.SSID());

    // print the MAC address of the router you're attached to:
    byte bssid[6];
    WiFi.BSSID(bssid);
    Serial.print("BSSID: ");
    printMacAddress(bssid);

    // print the received signal strength:
    long rssi = WiFi.RSSI();
    Serial.print("signal strength (RSSI):");
    Serial.println(rssi);

    // print the encryption type:
    byte encryption = WiFi.encryptionType();
    Serial.print("Encryption Type:");
    Serial.println(encryption, HEX);
    Serial.println();
}

void printMacAddress(byte mac[]) {
    for (int i = 0; i < 6; i++) {
        if (i > 0) {
            Serial.print(":");

```

```
    }  
    if (mac[i] < 16) {  
        Serial.print("0");  
    }  
    Serial.print(mac[i], HEX);  
}  
Serial.println();  
}
```

APÊNDICE B – CREDENCIAIS.H (MASCARADO)

```
// Insert your network credentials
#define WIFI_SSID "Nome_Rede_Wifi"
#define WIFI_PASSWORD "Senha_Rede_Wifi"

// Insert Firebase project API Key
#define API_KEY "API_KEY_PROJETO_FIREBASE"

// Insert RTDB URLdefine the RTDB URL */
#define DATABASE_URL "https://nome-projeto-*-rtdb.firebaseio.com/"

#define USER_EMAIL "email@mail.com"
#define USER_PASSWORD "senha_usuario"
```