

UNIVERSIDADE FEDERAL DE SÃO CARLOS  
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA  
DEPARTAMENTO DE QUÍMICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM QUÍMICA

**INTEGRAÇÃO E MINIATURIZAÇÃO DE SENSORES  
COMERCIAIS DE BAIXO CUSTO PARA MONITORAMENTO  
HÍDRICO**

Fernanda Renata Almeida Ribeiro\*

Dissertação apresentada como parte dos  
requisitos para obtenção do título de  
MESTRA EM QUÍMICA, área de  
concentração: QUÍMICA ANALÍTICA.

Orientador: Ernesto C. Pereira

\*Bolsista CNPq

SÃO CARLOS – SP

2025

Ribeiro, Fernanda Renata Almeida

Integração e Miniaturização de Sensores Comerciais de Baixo Custo para Monitoramento Hídrico / Fernanda Renata Almeida Ribeiro -- 2025.  
104f.

Dissertação (Mestrado) - Universidade Federal de São Carlos, campus São Carlos, São Carlos  
Orientador (a): Ernesto Chaves Pereira de Souza  
Banca Examinadora: Eryza Guimarães de Castro, Elton Fabiano Sitta  
Bibliografia

1. Sensores; pH; Turbidez;. 2. Condutividade; Temperatura; Arduino. I. Ribeiro, Fernanda Renata Almeida. II. Título.

Ficha catalográfica desenvolvida pela Secretaria Geral de Informática  
(SIn)

DADOS FORNECIDOS PELO AUTOR

Bibliotecário responsável: Arildo Martins - CRB/8 7180



**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

Centro de Ciências Exatas e de Tecnologia  
Programa de Pós-Graduação em Química

---

**Folha de Aprovação**

---

Defesa de Dissertação de Mestrado da candidata Fernanda Renata Almeida Ribeiro, realizada em 27/08/2025.

**Comissão Julgadora:**

Prof. Dr. Ernesto Chaves Pereira de Souza (UFSCar)

Profa. Dra. Eryza Guimarães de Castro (UNICENTRO)

Prof. Dr. Elton Fabiano Sitta (UFSCar)

O Relatório de Defesa assinado pelos membros da Comissão Julgadora encontra-se arquivado junto ao Programa de Pós-Graduação em Química.

*“Eu desejo que meninas negras e indígenas possam sonhar com outros futuros e não só aqueles que estão socialmente dados para nós.”*

- *Silvana Bahia*

*À minha avó Berenice, que me ensinou o valor do amor e perseverança, e em memória do meu avô Lúcio, que me mostrou a força da honestidade.*

## AGRADECIMENTOS

Chegar até aqui foi um caminho desafiador, feito de muitos dias de dedicação, dúvidas, conquistas e recomeços. Por isso, antes de tudo, agradeço a mim mesma, pela persistência, pela coragem de continuar mesmo quando tudo parecia difícil, e pela capacidade de aprender e crescer ao longo do processo.

Agradeço imensamente à minha mãe, Gisele Aparecida de Almeida, pelo apoio incondicional, pelo amor que me sustenta nos dias bons e nos dias difíceis, e por sempre acreditar em mim, mesmo quando eu mesma duvidava e por simplesmente existir, eu sou porque você é. Sou grata também ao meu irmão Guilherme Diogo, por ser meu companheiro de vida, sua existência me alegra e me faz melhor a cada dia, obrigada!

Aos meus amigos, que estiveram ao meu lado com palavras de incentivo, conversas leves, escuta generosa e aquele carinho essencial que ajuda a manter a sanidade. A presença de vocês fez toda a diferença deixando o banquinho da depressão menos triste. Excepcionalmente ao Dr. Felipe Staciaki da Luz pelas boas discussões de hipóteses e pelos momentos de respiro nesses anos, aos Drs. Alex Silva de Moraes e Roger Gonçalves por toda eutímia e conhecimento sobre arduino e programação compartilhados, sem os quais eu não teria chegado até aqui, obrigada!

Ao meu orientador, professor Dr. Ernesto Pereira, pela orientação, pela paciência, pelas sugestões valiosas e por me conduzir com seriedade e respeito ao longo dessa jornada acadêmica, seu entusiasmo pela ciência me inspirou a não desistir mesmo quando nada parecia funcionar, pois o importante é o que se aprende no caminho, obrigada pela confiança prof!

Agradeço também a todos os professores, colegas, técnicos, secretárias e demais profissionais que, de alguma forma, contribuíram para meu aprendizado durante o mestrado. Cada troca, cada dúvida discutida, cada colaboração deixou sua marca neste trabalho. Muito obrigada a todos!

Agradeço também ao Programa MAI-DAI UFSCAr (Mestrado e Doutorado para Inovação), em parceria com a empresa ImportNow e CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico do Brasil), pelo apoio fundamental por meio da concessão desta bolsa (131138/2023-5), que foi essencial para o desenvolvimento da minha pesquisa. Investir em ciência é investir no futuro do país, e o incentivo do governo brasileiro à pesquisa nacional fortalece não apenas a produção de conhecimento, mas também o desenvolvimento tecnológico e social. Muito Obrigada!

Agradeço a família são carlense que criei aqui (fyah!) e por fim, fica aqui o meu sincero agradecimento a todas as experiências, fáceis e difíceis, que me trouxeram até este momento. Essa caminhada foi feita por muitas mãos e por muito esforço próprio. E eu me orgulho disso.

## LISTA DE SIGLAS E SÍMBOLOS

### **Siglas:**

**ANA** - Agência Nacional de Águas e Saneamento Básico

**CNPq** - Conselho Nacional de Desenvolvimento Científico e Tecnológico

**EC** - Condutividade Elétrica

**EEPROM** - Electrically Erasable Programmable Read-Only Memory

**FAU** - Unidades de Atenuação de Formazina

**IoT** - Internet das Coisas

**ISFET** - Ion-Sensitive Field-Effect Transistor

**MEMS** - Sistemas Microeletromecânicos

**NTU** - Unidades Nefelométricas de Turbidez

**RTDs** - Resistance Temperature Detectors

**TDS** - Sólidos Totais Dissolvidos

**UNESCO** - Organização das Nações Unidas para a Educação, a Ciência e

Cultura

**WMO** - Organização Meteorológica Mundial

**YSI** - Empresa fabricante de equipamentos para monitoramento de água

### **Símbolos:**

**°C** - Graus Celsius

**μS/cm** - Microsiemens por centímetro

**mS/cm** - Milisiemens por centímetro

**ppm** - Partes por milhão

**pH** - Potencial Hidrogeniônico

**H<sup>+</sup>** - Íon Hidrogênio

**OH<sup>-</sup>** - Íon Hidroxila

**CO<sub>2</sub>** - Dióxido de Carbono

**KOH** - Hidróxido de Potássio

**NaOH** - Hidróxido de Sódio

**HCl** - Ácido Clorídrico

**Pt** - Platina

**Ag/AgCl** - Eletrodo de Prata/Cloreto de Prata

## LISTA DE FIGURAS

FIGURA 1. 1 - Parcela da população com uso de água potável com gestão segura.....	1
FIGURA 1. 2 - Consumo setorial de água .....	2
FIGURA 1. 3 - Arduino Uno Rev3 .....	7
FIGURA 1. 4 - Termistor NTC KL15L010 10R 5A.....	8
FIGURA 1. 5 - Resistance Temperature Detectors (RTDs).....	8
FIGURA 1. 6 - Resistance Temperature Detectors (RTDs).....	9
FIGURA 1. 7 - Termopar de Mola, Termopar M12 Tipo K.....	9
FIGURA 1. 8 - Sensor De Temperatura Ir Infravermelho Mlx90614 Para Arduino .....	10
FIGURA 1. 9 - Sensor de condutividade condutivo LDL101 .....	11
FIGURA 1. 10 - Sensor de condutividade indutivo LDL210 .....	11
FIGURA 1. 11 - Módulo Sensor de EC Condutividade da Água com Eletrodo analógico .....	12
FIGURA 1. 12 - Medidor de TDS Sensor de Condutividade da Água Analógico.....	12
FIGURA 1. 13 - Eletrodo Combinado Universal De Vidro.....	13
FIGURA 1. 14 - Pocket ISFET pH Meter.....	14
FIGURA 1. 15 - Eletrodo de pH/ORP Série 2724-2726 - SIGNET .....	14
FIGURA 1. 16 - Eletrodo de Pt com Pani eletrodepositada para sensor de pH .....	15
FIGURA 1. 17 - Eletrodo de vidro para sensor de pH .....	15
FIGURA 1. 18 - DTF16 Haze Control Turbidimetro de Processo.....	16
FIGURA 1. 19 - DTF16 Haze Control Turbidimetro de Processo.....	16
FIGURA 1. 20 - Sensor de turbidez Turbimax CUS52D.....	17
FIGURA 1. 21 - Turbidity Sensor InPro8610ie .....	17
FIGURA 1. 22 - Sensor de Turbidez para Arduino ST100 + Módulo de Leitura.....	18
FIGURA 1. 23 - MEMS – Sistemas Microeletromecânicos .....	20
FIGURA 1. 24 - DTLstar C-200 Electrochemical Sensor Silk Screen Printed.....	20
FIGURA 1. 25 - Lab-on-a-chip form parameter inspection.....	21
FIGURA 1. 26 - Um exemplo de nanosensor .....	21
FIGURA 3. 1 - Diagrama de Circuitos do Sistema Integrado.....	23
FIGURA 3. 2 - Representação do sistema integrado montado durante medida.....	24
FIGURA 3. 3 - Sensor de Temperatura DS 18B20 à Prova D'Água.....	25
FIGURA 3. 4 - Medidor de TDS Sensor de Condutividade da Água Analógico.....	26
FIGURA 3. 5 - Protótipo de Sensor de Condutividade Miniaturizado .....	27
FIGURA 3. 6 - Sensor de pH para Arduino + Módulo de Leitura BNC PH4502C.....	28
FIGURA 3. 7 - Protótipo de Sensor de pH Miniaturizado .....	29
FIGURA 3. 8 - Sistema de Titulação para estudo do funcionamento do sensor de pH miniaturizado .	29
FIGURA 3. 9 - Sensor de Turbidez para Arduino ST100 + Módulo de Leitura.....	30
FIGURA 3. 10 - Diagrama de Blocos referente ao código do Sensor de Temperatura (DS18B20).....	32
FIGURA 3. 11 - Diagrama de Blocos referente ao código do Sensor de Condutividade (TDS).....	33
FIGURA 3. 12 - Diagrama de Blocos referente ao código do Sensor de pH (PH4502C).....	34
FIGURA 3. 13 - Diagrama de Blocos referente ao código do Sensor de Turbidez (ST10) .....	35
FIGURA 3. 14 - Diagrama representando a sequência de integração dos sensores.....	35
FIGURA 3. 15 - Diagrama de Blocos referente ao código de integração dos sensores.....	37
FIGURA 4. 1 – Medida e Replicata da Temperatura em função do tempo para uma solução Padrão 1413 µS/cm .....	42
FIGURA 4. 2 – Medida e Replicata da Variação da Temperatura em função do tempo para água destilada.....	42
FIGURA 4. 3 - Relação entre a CE Padrão e o Potencial medido pelo sensor de TDS .....	44

FIGURA 4. 4 - Medida e Replicata da Variação da Condutividade Elétrica em função do tempo para água destilada.....	45
FIGURA 4. 5 - Medida e Replicata da Variação da Condutividade Elétrica em função do tempo para Solução Padrão 1413 $\mu\text{S}/\text{cm}$ .....	45
FIGURA 4. 6 - Relação entre o pH da Solução Padrão e o Potencial medido pelo sensor de pH .....	46
FIGURA 4. 7 - Medida e Replicata da Variação do pH em função do tempo para Solução Padrão pH 7 .....	47
FIGURA 4. 8 - Medida e Replicata da Variação do pH em função do tempo para Água Destilada ....	48
FIGURA 4. 9 - Relação entre a Turbidez da Solução Padrão e o Potencial medido pelo sensor de Turbidez .....	50
FIGURA 4. 10 - Medida e Replicata da Variação da Turbidez em função do tempo para Solução Padrão 100 NTU.....	50
FIGURA 4. 11 - - Medida e Replicata da Variação da Turbidez em função do tempo para Água Destilada.....	52
FIGURA 4. 12 - Titulação de HCl (0,1 M) com NaOH (0,5 M).....	67
FIGURA 4. 13 - Relação entre condutividade de soluções padrão e a condutância obtida pelo sensor miniaturizado de condutividade .....	67

## RESUMO

### INTEGRAÇÃO E MINIATURIZAÇÃO DE SENSORES COMERCIAIS DE BAIXO CUSTO PARA MONITORAMENTO HÍDRICO

Esta dissertação apresenta o desenvolvimento de uma plataforma de baixo custo para o monitoramento da qualidade da água, utilizando o microcontrolador Arduino UNO R3 e uma integração de sensores comerciais. O sistema proposto visa oferecer uma solução acessível e eficiente para a coleta de dados de parâmetros hídricos essenciais, como temperatura, condutividade elétrica, pH e turbidez. O trabalho detalha a metodologia de integração dos sensores, o firmware desenvolvido para a aquisição e processamento dos dados, e a arquitetura modular que permite a expansão para diferentes cenários de aplicação. O sistema foi projetado para ser robusto e de fácil replicação, contribuindo para a democratização do monitoramento ambiental e a detecção precoce de alterações na qualidade da água. Embora os resultados detalhados de validação e calibração ainda estejam em fase de obtenção, a plataforma demonstra o potencial para ser uma ferramenta valiosa em estudos ambientais e gestão de recursos hídricos.

**Palavras-chave:** sensores; pH; turbidez; condutividade; temperatura; arduino.

## ABSTRACT

### INTEGRATION AND MINIATURIZATION OF LOW-COST COMMERCIAL SENSORS FOR WATER MONITORING

This dissertation presents the development of a low-cost platform for water quality monitoring, utilizing the Arduino UNO R3 microcontroller and an integration of commercial sensors. The proposed system aims to offer an accessible and efficient solution for collecting data on essential water parameters, such as temperature, electrical conductivity, pH, and turbidity. The work details the sensor integration methodology, the firmware developed for data acquisition and processing, and the modular architecture that allows for expansion to different application scenarios. The system was designed to be robust and easily replicable, contributing to the democratization of environmental monitoring and the early detection of changes in water quality. Although detailed validation and calibration results are still being obtained, the platform demonstrates the potential to be a valuable tool in environmental studies and water resource management.

**Keywords:** sensors; pH; turbidity; conductivity; temperature; arduino.

# SUMÁRIO

1 INTRODUÇÃO .....	1
1.1 Motivação.....	1
1.1.1 Cenário Global da Água.....	1
1.1.2 Monitoramento .....	3
1.1.3 Integração dos Sensores Comerciais .....	5
1.2 Revisão Bibliográfica.....	6
1.2.1. Plataformas de Monitoramento da Qualidade da Água.....	6
1.2.2 Parâmetros de Qualidade da Água e Tecnologias de Sensoriamento.....	7
1.2.3 Desafios na Integração de Sensores .....	18
1.2.4 Miniaturização de Sensores: Perspectivas e Abordagens.....	19
2 OBJETIVOS .....	22
3 MATERIAIS E MÉTODOS .....	23
3.1 Visão Geral do Sistema de Monitoramento .....	23
3.2 Aquisição de Dados de Qualidade da Água .....	25
3.2.1 Medição de Temperatura.....	25
3.2.2 Medição de Condutividade.....	26
3.2.3 Medição de pH .....	28
3.2.4 Medição de Turbidez.....	29
3.3 Processamento e Controle .....	30
3.3.1 Plataforma Arduino .....	30
3.3.2 Firmware Desenvolvido .....	31
3.4 Transmissão e Armazenamento de Dados.....	37
3.4.1 Sistema de Comunicação.....	37
3.4.2 Armazenamento Local .....	38
3.5 Validação do Sistema Integrado.....	39
3.5.1 Metodologia de Testes .....	39
4 RESULTADOS E DISCUSSÃO .....	41
4.1 Desempenho do Sistema de Aquisição de Dados .....	41
4.1.1 Medição de Temperatura.....	41
4.1.2 Medição de Condutividade.....	43
4.1.3 Medição de pH .....	46
4.1.4 Medição de Turbidez.....	49
4.2 Avaliação do Sistema de Processamento e Controle.....	53
4.2.1 Desempenho da Plataforma Arduino .....	53
4.2.2 Eficiência do Firmware .....	54

4.2.3 Integração do Sensores.....	55
4.3 Desempenho da Transmissão e Armazenamento de Dados.....	57
4.3.1 Sistema de Comunicação.....	57
4.3.2 Armazenamento Local.....	59
4.4 Análise Comparativa e Discussão Geral.....	61
4.4.1 Comparação com Sistemas Comerciais.....	61
4.4.2 Potenciais Aplicações.....	63
4.4.3 Limitações Identificadas e Oportunidades de Melhorias.....	65
5 CONCLUSÃO.....	68
6. BIBLIOGRAFIA.....	69
7. APÊNDICE.....	73

# 1 INTRODUÇÃO

## 1.1 Motivação

### 1.1.1 Cenário Global da Água

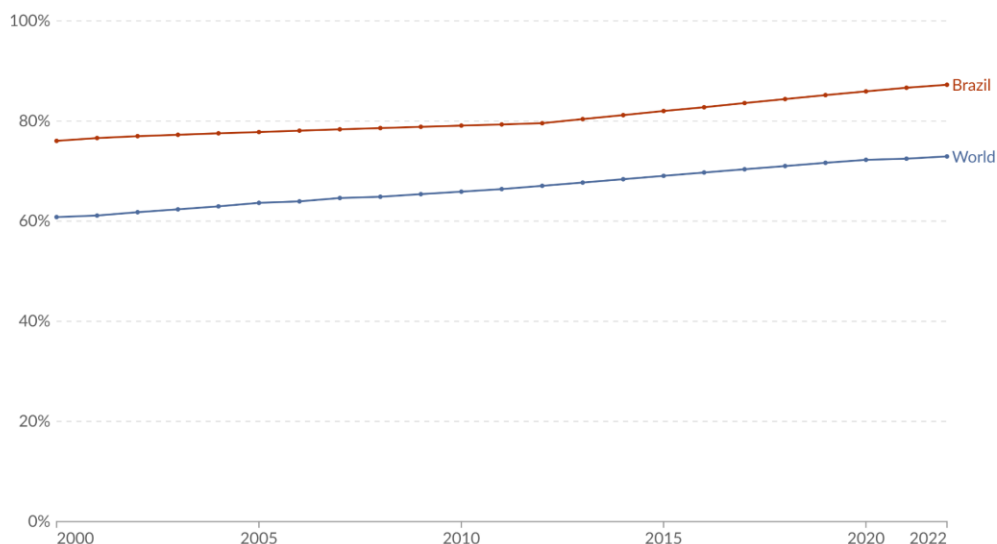
A água é o alicerce da vida na Terra, fundamental para todos os ecossistemas, atividades humanas e equilíbrio climático. Embora cubra 71% da superfície do planeta, apenas 2,5% é doce, e menos de 1% dessa parcela está acessível para consumo direto (UNESCO, 2024). Dados do *Our World in Data* (2024) mostram que em torno de 27% da população global ainda não tem acesso a água potável gerenciada com segurança (Figura 1.1).

#### Percentual de pessoas com acesso à água potável com gestão segura

##### Share using safely managed drinking water



Safely managed drinking water service is defined as an improved water source<sup>1</sup> located on the premises, available when needed, and free from contamination.



Data source: WHO/UNICEF Joint Monitoring Programme for Water Supply, Sanitation and Hygiene (JMP) (2024)  
OurWorldinData.org/water-access | CC BY

1. Improved drinking water sources Improved drinking water sources are those that have the potential to deliver safe water by nature of their design and construction, and include: piped water, boreholes or tubewells, protected dug wells, protected springs, rainwater, and packaged or delivered water.

FIGURA 1. 1 - Parcela da população com uso de água potável com gestão segura

Fonte: WHO/UNICEF Joint Monitoring Programme for Water Supply, Sanitation and Hygiene (JMP) (2024) – with major processing by Our World in Data

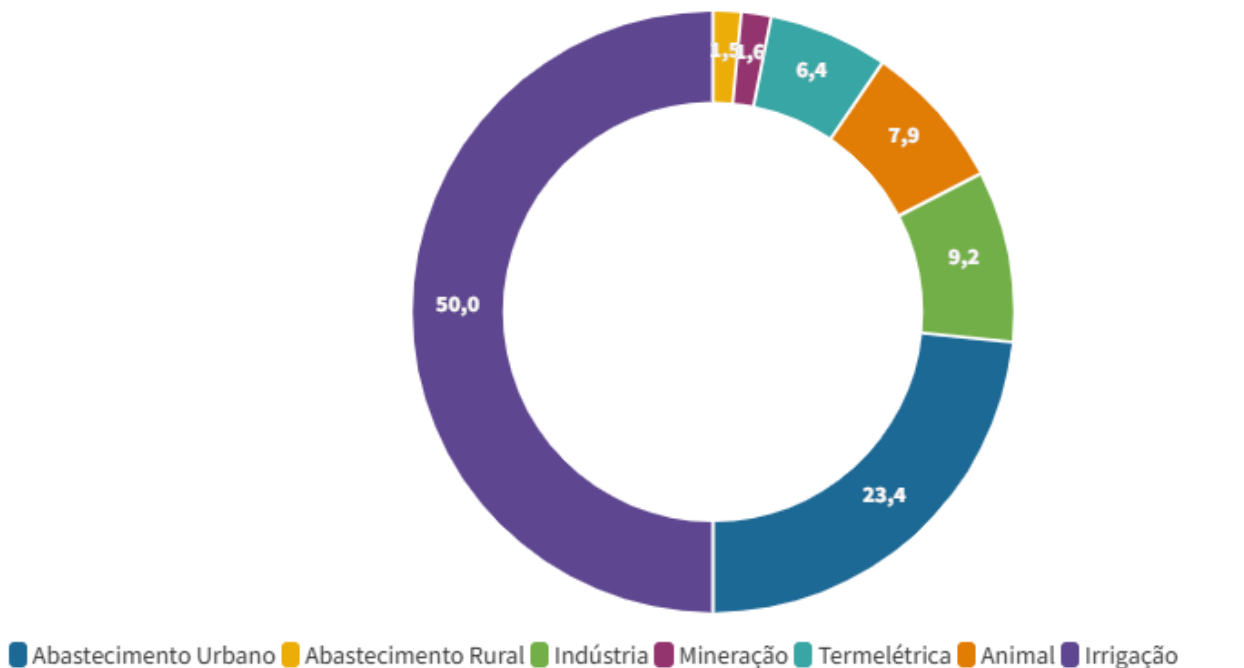
Enquanto no Brasil, a Agência Nacional de Águas (ANA, 2024) aponta que por volta de 50% dos corpos hídricos do Brasil correspondem à demanda de irrigação do país (Figura 1.2).

## Usos Consuntivos Setoriais

Em 2023

Retirada total dos usos setoriais: 2.103,6 m<sup>3</sup>/s ≈ 66,52 trilhões de L/ano

m<sup>3</sup>/s %



Fonte: Agência Nacional de Águas e Saneamento Básico • Base Nacional de Referência de Usos Consuntivos da Água  
Figura atualizada em junho de 2024.

FIGURA 1. 2 - Consumo setorial de água  
Fonte: Agência Nacional de Águas e Saneamento Básico

O estresse hídrico, agravado pelas mudanças climáticas, já afeta 2 bilhões de pessoas em regiões críticas (WMO, 2024), exigindo soluções urgentes para monitoramento e gestão sustentável.

Como já apontado, a dependência humana da água revela-se em setores primários, como a agricultura, que consome 70% dos recursos hídricos globais. Um exemplo de problemas relacionados à não homogeneidade são as secas como as do *Corn Belt* americano reduziram significativamente a produção de milho, elevando os preços mundiais (Yang; Wang, 2023). Na indústria, a escassez na região metropolitana de São Paulo (2014-2015) paralisou parte das fábricas por falta de água, com perdas econômicas substanciais onde a gestão hídrica inadequada foi um fator crucial. (Empinotti; Budds; Aversa, 2019). Casos

extremos como o "Dia Zero" na Cidade do Cabo (2018) - quando a cidade quase esgotou seu abastecimento, demonstram como a crise hídrica ameaça diretamente a vida urbana. (Simpson; Shearing; Dupont, 2020).

No Brasil, desastres como o rompimento da barragem em Mariana (2015) e Brumadinho (2019) contaminaram extensas áreas dos Rios Doce e Paraopeba com metais pesados, tornando a água imprópria para consumo em diversos municípios (Governo de Minas Gerais, 2024). Paralelamente, milhões de brasileiros vivem sem acesso à água tratada (SNIS, 2022), cenário que se repete em países em desenvolvimento, onde a falta de monitoramento agrava a vulnerabilidade. Esses exemplos evidenciam a necessidade crítica de sistemas de baixo custo para detecção precoce de contaminação e gestão eficiente.

Nesse contexto, a plataforma proposta nesta pesquisa, com sensores comerciais integrados e algoritmos de correção de dados, torna-se uma ferramenta vital para democratizar o monitoramento da qualidade da água.

Sua capacidade de fornecer informações em tempo real a um custo significativamente menor que sistemas convencionais pode revolucionar a gestão hídrica em comunidades remotas, pequenos produtores rurais e áreas de risco ambiental. Ao integrar acessibilidade e precisão, esta solução representa um avanço concreto para enfrentarmos os desafios globais da água no século XXI.

### **1.1.2 Monitoramento**

O monitoramento da qualidade da água representa um pilar fundamental na gestão sustentável dos recursos hídricos. Tradicionalmente, este processo depende de coletas manuais de amostras e análises laboratoriais, método que, embora preciso, apresenta limitações críticas: alto custo operacional (US\$ 200-500 por amostras que variam de 0,5 a 1L), baixa frequência temporal (tipicamente mensal ou trimestral) e, conseqüentemente, um significativo atraso entre coleta e resultado (ANA, 2023). Essa abordagem convencional cria "pontos cegos" temporais que podem não detectar eventos críticos de contaminação, como

descargas industriais intermitentes ou florações de algas tóxicas, que frequentemente ocorrem entre os períodos de amostragem.

Estudos recentes demonstram que eventos de poluição de curta duração podem ter impactos severos nos ecossistemas aquáticos. Por exemplo, (Schulz; Costa, 2015) documentaram que descargas intermitentes de efluentes industriais e urbanos no Rio dos Sinos (RS) causaram hipóxia e recorrentes eventos de mortalidade massiva de peixes, incluindo um evento severo em 2006 que resultou na morte de aproximadamente 100 toneladas de peixes. Estes episódios agudos de contaminação, frequentemente não detectados por monitoramento convencional mensal, evidenciam a necessidade de sistemas de monitoramento contínuo, capazes de fornecer dados em tempo real ou quase real.

A revolução digital e o avanço da Internet das Coisas (IoT) têm transformado o paradigma do monitoramento ambiental. Sensores conectados, plataformas de transmissão remota e análise de dados em nuvem permitem hoje o que era tecnicamente inviável há uma década: o monitoramento contínuo, autônomo e acessível de parâmetros críticos da qualidade da água (De Camargo et al., 2023). No entanto, sistemas comerciais integrados de alta precisão ainda apresentam custos proibitivos (R\$10.000-50.000) para implementação em larga escala, especialmente em países em desenvolvimento e comunidades de baixa renda (UNESCO, 2024).

Este cenário cria uma "divisão digital hídrica" onde comunidades com menos recursos permanecem vulneráveis por falta de acesso a tecnologias de monitoramento. Dados da UNESCO (2023) revelam que apenas uma pequena fração dos corpos d'água em países de baixa renda possuem monitoramento regular, comparado a taxas muito superiores em nações desenvolvidas. Esta disparidade amplifica riscos de saúde pública e ambientais precisamente nas regiões onde a proteção dos recursos hídricos é mais urgente.

### **1.1.3 Integração dos Sensores Comerciais**

Diante dos desafios apresentados, a integração de sensores comerciais em plataformas de baixo custo emerge como uma solução promissora para democratizar o monitoramento da qualidade da água. Esta abordagem utiliza componentes eletrônicos e sensores disponíveis no mercado, combinados através de microcontroladores de código aberto como o Arduino, para criar sistemas funcionais a uma fração do custo de equipamentos profissionais dedicados (Bogdan et al., 2023).

A utilização de sensores comerciais apresenta algumas vantagens em comparação com o desenvolvimento de sensores customizados ou miniaturizados. Primeiramente, estes componentes já passaram por processos de validação industrial, oferecendo confiabilidade e repetibilidade aceitáveis para muitas aplicações de monitoramento (Kumar; Roy; Bano, 2021).

Adicionalmente, sua ampla disponibilidade no mercado facilita a manutenção, substituição e replicação dos sistemas, aspectos críticos para implementações em áreas remotas ou com recursos limitados.

Estudos comparativos demonstram que, embora sensores comerciais de baixo custo possam apresentar precisão inferior a equipamentos laboratoriais em medições absolutas, sua capacidade de detectar variações relativas e tendências é suficiente para muitas aplicações de monitoramento ambiental. Pesquisas demonstram que plataformas de baixo custo e algoritmos avançados podem atingir altas taxas de detecção de eventos de contaminação, rivalizando com a eficácia de sistemas profissionais muito mais caros. (Bencs et al., 2022)

A integração destes sensores, entretanto, apresenta desafios técnicos significativos. A calibração adequada, compensação de interferências cruzadas entre parâmetros, correção de deriva temporal e processamento de sinais são aspectos críticos que determinam a confiabilidade final do sistema (Zhang et al., 2024). Adicionalmente, a operação em campo exige robustez contra intempéries, autonomia energética e capacidade de transmissão de dados em condições

adversas.

Neste contexto, a presente pesquisa propõe uma abordagem pragmática: desenvolver uma plataforma que integre sensores comerciais de temperatura, pH, condutividade e turbidez, otimizando sua operação conjunta através de calibração cruzada, processamento de dados e transmissão remota. Esta solução busca equilibrar custo-benefício, confiabilidade e acessibilidade, criando um sistema viável para implementação em larga escala, especialmente em contextos de recursos limitados.

## **1.2 Revisão Bibliográfica**

### **1.2.1. Plataformas de Monitoramento da Qualidade da Água**

O monitoramento da qualidade da água tem evoluído significativamente nas últimas décadas, passando de análises laboratoriais pontuais para sistemas de sensoriamento contínuo e em tempo real. As plataformas de monitoramento modernas podem ser classificadas em três categorias principais: sistemas comerciais de alta precisão, estações de monitoramento fixas e plataformas de baixo custo baseadas em microcontroladores (Hamel et al., 2024).

Os sistemas comerciais de alta precisão, como os fabricados por YSI, Hach e In-Situ, oferecem excelente confiabilidade e precisão, mas com custos que podem variar de R\$10.000,00 a R\$100.000,00, tornando-os inacessíveis para aplicações de larga escala ou em regiões com recursos limitados (In-Situ, 2025; YSI, 2025). As estações de monitoramento fixas, frequentemente mantidas por agências ambientais governamentais, fornecem dados de qualidade, mas com cobertura espacial limitada devido aos altos custos de instalação e manutenção. (ANA, 2025)

Em resposta a essas limitações, a última década testemunhou o surgimento de plataformas de baixo custo baseadas em microcontroladores de código aberto, como Arduino (Figura 1.3) e Raspberry Pi (Hong et al., 2021).

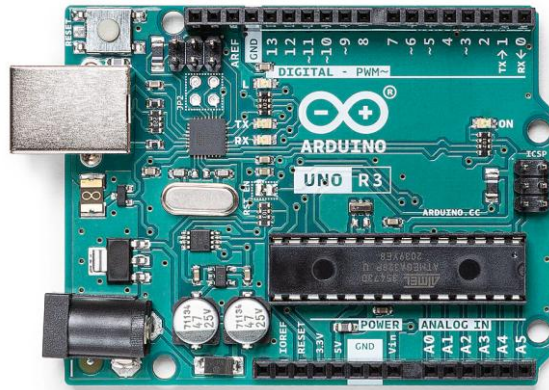


FIGURA 1.3 - Arduino Uno Rev3

Fonte: Adaptado de <https://store.arduino.cc/products/arduino-uno-rev3>

Estas plataformas utilizam sensores comerciais acessíveis e componentes eletrônicos padronizados para criar sistemas de monitoramento com custo reduzido, permitindo maior densidade espacial de pontos de monitoramento e democratizando o acesso a dados de qualidade da água (Forhad et al., 2024).

A integração de sensores em plataformas baseadas em Arduino apresenta desafios técnicos significativos, incluindo calibração adequada, compensação de interferências cruzadas, consumo energético e robustez ambiental. No entanto, estudos recentes demonstram que, quando adequadamente implementadas, estas plataformas podem alcançar precisão comparável a equipamentos profissionais para muitas aplicações de monitoramento ambiental (De Camargo et al., 2023).

## **1.2.2 Parâmetros de Qualidade da Água e Tecnologias de Sensoriamento**

### **1.2.2.1 Temperatura**

A temperatura é um parâmetro fundamental na caracterização da qualidade da água, influenciando diretamente processos físicos, químicos e biológicos nos ecossistemas aquáticos. Variações de temperatura afetam a solubilidade de gases (especialmente oxigênio), taxas metabólicas de organismos aquáticos, cinética de reações químicas e estratificação em corpos d'água (Lehmann et al., 2023). Além disso, a temperatura é essencial para a compensação

de outros parâmetros, como pH e condutividade elétrica, que são dependentes da temperatura.

### **Tecnologias de Sensoriamento de Temperatura:**

**1. Termistores:** São semicondutores cuja resistência varia significativamente com a temperatura. Os termistores NTC (Negative Temperature Coefficient) são os mais comuns em aplicações aquáticas, oferecendo boa sensibilidade e resposta rápida. São relativamente baratos e robustos, mas requerem circuitos de condicionamento de sinal e calibração para alta precisão (Figura 1.4). (Koestoer et al., 2019).



FIGURA 1. 4 - Termistor NTC KL15L010 10R 5A

Fonte: Adaptado de <https://www.erkatron.com.br/termistor-ntc-kl15l010-10r-5a>

**2. RTDs (Resistance Temperature Detectors):** Baseados na variação da resistência de metais puros (tipicamente platina - Pt100 ou Pt1000) com a temperatura. Oferecem excelente estabilidade e precisão, especialmente em medições de longo prazo, mas são mais caros que termistores e geralmente requerem circuitos mais complexos (Figura 1.5). (Khatri; Gupta; Gupta, 2019).



FIGURA 1. 5 - Resistance Temperature Detectors (RTDs)

Fonte: Adaptado de <https://www.thermosensors.com/rtd>

**3. Sensores Digitais Integrados:** Incorporam o elemento sensor e circuitos

de condicionamento e conversão em um único chip, fornecendo saída digital diretamente proporcional à temperatura. O DS18B20 (Maxim Integrated) é um exemplo popular, utilizando protocolo 1-Wire, que permite conectar múltiplos sensores em um único barramento digital. Oferecem boa precisão (tipicamente  $\pm 0.5^{\circ}\text{C}$ ), facilidade de integração e custo moderado (Figura 1.6). (Maxim Integrated Products, Inc, 2019).

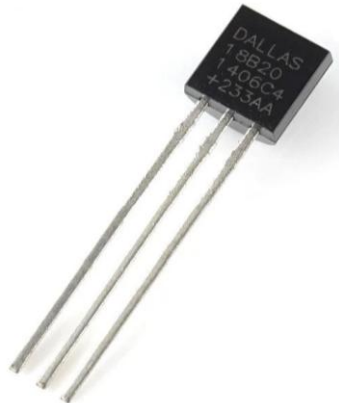


FIGURA 1. 6 - Resistance Temperature Detectors (RTDs)

Fonte: Adaptado de [DS18B20 DALLAS 18B20 TO-92 1 Wire Digital Temperature Sensor](#)

**4. Termopares:** Baseados no efeito Seebeck, geram uma pequena tensão proporcional à diferença de temperatura entre duas junções de metais diferentes. São robustos e podem operar em amplas faixas de temperatura, mas oferecem menor precisão em aplicações aquáticas e requerem circuitos de compensação de junção fria (Figura 1.7). (Michalski; Briard; Michel, 2001) (“Efeito Seebeck”, 2025)



FIGURA 1. 7 - Termopar de Mola, Termopar M12 Tipo K

Fonte: Adaptado de <https://www.amazon.com.br/Termopar-Metros-Termopares-Inoxidável-Temperatura/dp/B0C372C8BD>

**5. Sensores Infravermelhos:** Medida de temperatura sem contato, baseados

na radiação infravermelha emitida pelos corpos. Raramente utilizados em monitoramento subaquático devido à forte absorção da radiação infravermelha pela água (Figura 1.8). (Childs et al., 2008).



FIGURA 1. 8 - Sensor De Temperatura Ir Infravermelho Mlx90614 Para Arduino  
Fonte: Adaptado de <https://www.mercadolivre.com.br/sensor-de-temperatura-ir-infravermelho-mlx90614-para-arduino>

Para aplicações de monitoramento da qualidade da água, os sensores digitais integrados como o DS18B20 e os termistores NTC encapsulados são as escolhas mais comuns, oferecendo um bom equilíbrio entre precisão, facilidade de integração e custo. Para esse projeto optou-se pelo DS18B20, na versão encapsulada e aprova d'água, por apresentar maior robustez da sonda que pode ser submersa na amostra, e leituras precisas já que o sinal digital apresenta menos ruído, e sua resolução pode ser configurada.

### 1.2.2.2 Condutividade Elétrica

A condutividade elétrica (CE) mede a capacidade da água de conduzir corrente elétrica, sendo diretamente proporcional à concentração de íons dissolvidos. Este parâmetro fornece uma indicação rápida do conteúdo total de sais dissolvidos (TDS) e é frequentemente utilizado como indicador de poluição, intrusão salina ou alterações na composição química da água. A CE é expressa em microsiemens por centímetro ( $\mu\text{S}/\text{cm}$ ) ou milisiemens por centímetro ( $\text{mS}/\text{cm}$ ) e varia significativamente com a temperatura (aumento de aproximadamente 2% por  $^{\circ}\text{C}$ ), exigindo compensação para comparações válidas (Hayashi, 2004).

### Tecnologias de Sensoriamento de Condutividade:

**1. Sensores Condutivos (Contato Direto):** Utilizam dois ou mais eletrodos em contato direto com a água, aplicando uma tensão alternada para evitar polarização e medindo a corrente resultante. A geometria dos eletrodos (área e distância) define a "constante de célula" do sensor. São os mais comuns em aplicações de monitoramento, com variantes de dois, três ou quatro eletrodos para diferentes faixas de medição (Figura 1.9). (Radiometer Analytical, 2004).



FIGURA 1. 9 - Sensor de condutividade condutivo LDL101  
Fonte: Adaptado de <https://www.ifm.com/br/pt/product/LDL101>

**2. Sensores Indutivos (Sem Contato):** Utilizam bobinas indutivas para criar um campo magnético na água, induzindo correntes proporcionais à condutividade. São mais adequados para águas com alta condutividade ou soluções corrosivas, pois não têm eletrodos em contato direto com o meio. São mais caros e menos sensíveis em baixas condutividades (Figura 1.10). (McCleskey et al., 2012).



FIGURA 1. 10 - Sensor de condutividade indutivo LDL210  
Fonte: Adaptado de <https://www.ifm.com/br/pt/product/LDL210>

**3. Sensores Capacitivos:** Medem alterações na capacitância entre eletrodos

isolados, que variam com a condutividade do meio. Menos comuns em aplicações aquáticas, mas úteis em situações específicas onde sensores condutivos ou indutivos não são adequados (Figura 1.11). (Gupta; Pandey, 2016).



FIGURA 1. 11 - Módulo Sensor de EC Condutividade da Água com Eletrodo analógico  
Fonte: Adaptado de <https://www.usinainfo.com.br/sensores/modulo-sensor-de-ec-condutividade>

**4. Sensores TDS:** Não medem diretamente a condutividade, mas utilizam a relação aproximada entre CE e TDS (Sólidos Totais Dissolvidos), tipicamente assumindo  $TDS (ppm) \approx CE (\mu S/cm) \times 0.5-0.7$ . São comuns em aplicações de baixo custo, mas menos precisos que condutímetros dedicados (Figura 1.12) (Wirman; Wardhana; Isnaini, 2019).

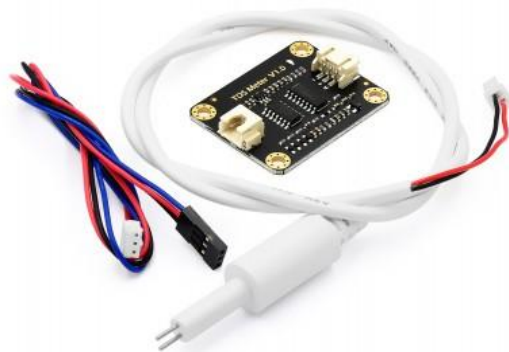


FIGURA 1. 12 - Medidor de TDS Sensor de Condutividade da Água Analógico  
Fonte: Adaptado de <https://www.usinainfo.com.br/outros-sensores-arduino/medidor-de-tds>

Para plataformas de monitoramento baseadas em Arduino, sensores condutivos de dois eletrodos com módulos de interface analógica são os mais utilizados devido ao seu baixo custo e facilidade de integração. No entanto, requerem compensação de temperatura e podem sofrer polarização e contaminação dos eletrodos em uso prolongado (González-Pérez et al., 2022),

esse modelo foi empregado no projeto principalmente pelo seu baixo custo e fácil operação.

### 1.2.2.3 pH

O pH é uma medida da concentração de íons hidrogênio em solução, indicando sua acidez ou alcalinidade em uma escala logarítmica de 0 a 14. Este parâmetro é crítico para ecossistemas aquáticos, pois influencia a solubilidade e disponibilidade biológica de nutrientes e substâncias potencialmente tóxicas, como metais pesados e amônia. A maioria dos organismos aquáticos é adaptada a faixas específicas de pH, com alterações significativas podendo causar estresse fisiológico ou mortalidade (Wurts; Durborow, 1992).

#### Tecnologias de Sensoriamento de pH:

**1. Eletrodos de Vidro:** A tecnologia mais estabelecida para medição de pH, consiste em um bulbo de vidro sensível a íons  $H^+$  que desenvolve um potencial elétrico proporcional à concentração desses íons. Geralmente incorporam um eletrodo de referência (frequentemente Ag/AgCl) para formar um eletrodo combinado. Oferecem boa precisão ( $\pm 0.01-0.1$  unidades de pH) e ampla faixa de medição, mas requerem calibração regular, manutenção cuidadosa e têm vida útil limitada (Figura 1.13). (Bates, 1955).



FIGURA 1. 13 - Eletrodo Combinado Universal De Vidro

Fonte: Adaptado de <https://www.dsylab.com.br/acessorios/eletrodos/eletrodo-de-ph-vidro>

**2. Sensores ISFET (Ion-Sensitive Field-Effect Transistor):** Utilizam um transistor de efeito de campo modificado onde a corrente é modulada pela concentração de íons  $H^+$ . Oferecem maior robustez mecânica que eletrodos de

vidro, resposta mais rápida e possibilidade de miniaturização. No entanto, são mais caros e podem sofrer deriva ao longo do tempo (Figura 1.14) (Bergveld, 2003).



FIGURA 1. 14 - Pocket ISFET pH Meter

Fonte: Adaptado de <https://deltatrak.com/product/pocket-isfet-ph-meter>

**3. Sensores de pH de Estado Sólido:** Utilizam materiais como óxidos metálicos (ex: óxido de irídio) que desenvolvem potencial superficial dependente do pH. Oferecem maior durabilidade, e maior precisão que eletrodos de vidro convencionais, entretanto se mostram uma opção mais cara. (Figura 1.15) (Zimer et al., 2010).



FIGURA 1. 15 - Eletrodo de pH/ORP Série 2724-2726 - SIGNET

Fonte: Adaptado de <https://www.tecnolog.ind.br/eletrodo-de-ph-orp-serie-2724-2726-signet-726.html>

**4. Sensores Baseados em Polímeros Condutores:** Utilizam polímeros eletricamente condutores cuja condutividade varia com o pH. Ainda em desenvolvimento para aplicações comerciais, oferecem potencial para sensores flexíveis e de baixo custo (Figura 1.16) (Lakard et al., 2007).



FIGURA 1. 16 - Eletrodo de Pt com Pani eletrodepositada para sensor de pH  
Fonte: A autora.

Para plataformas de monitoramento baseadas em microcontroladores, os eletrodos de vidro combinados com módulos de interface analógica (como o PH4502C) são a opção mais comum, oferecendo bom equilíbrio entre precisão, disponibilidade e custo e por isso foi o escolhido para o desenvolvimento deste projeto (Figura 1.17). (Wilson et al., 2022).



FIGURA 1. 17 - Eletrodo de vidro para sensor de pH  
Fonte: A autora.

#### **1.2.2.4 Turbidez**

A turbidez é uma medida da claridade ou opacidade da água, indicando a quantidade de material particulado em suspensão que dispersa ou absorve a luz. Partículas suspensas podem incluir sedimentos (argila, silte), matéria orgânica, algas, microrganismos e outros materiais microscópicos. A

turbidez é um importante indicador de qualidade da água, afetando a penetração de luz (e conseqüentemente a fotossíntese), a temperatura e a estética. Além disso, partículas suspensas podem transportar contaminantes adsorvidos e proteger microrganismos patogênicos da desinfecção (Anderson, 2005).

### **Tecnologias de Sensoriamento de Turbidez:**

**1. Sensores Nefelométricos:** Medem a luz dispersa em um ângulo específico (geralmente 90°) em relação ao feixe incidente. São os mais comuns e precisos para baixas turbidez, sendo o método padrão para água potável. A unidade de medida é NTU (Unidades Nefelométricas de Turbidez) (Figura 1.18) (Sadar, 1998).



FIGURA 1. 18 - DTF16 Haze Control Turbidímetro de Processo  
Fonte: Adaptado de <https://www.optek.com/pt/dtf16.asp>

**2. Sensores de Transmissão (Turbidímetros de Absorção):** Medem a atenuação da luz ao passar pela amostra (180° em relação ao feixe incidente). São mais adequados para águas com alta turbidez, onde a dispersão seria excessiva para medição nefelométrica. A unidade típica é FAU (Unidades de Atenuação de Formazina). (Figura 1.19) (Omar; Matjafri, 2009)



FIGURA 1. 19 - DTF16 Haze Control Turbidímetro de Processo  
Fonte: Adaptado de <https://www.anderson-negele.com/br/p/analise-de-processo/turbidimetro>

**3. Sensores de Retroespalhamento:** Medem a luz refletida em ângulos próximos à fonte (tipicamente  $<45^\circ$ ). São úteis em aplicações in-situ onde o volume de amostragem deve ser minimizado ou em águas muito turvas (Figura 1.20). (Downing, 2006).



FIGURA 1. 20 - Sensor de turbidez Turbimax CUS52D

Fonte: Adaptado de <https://www.br.endress.com/pt/produtos/analitica/CUS52D?t.tabId=product-overview>

**4. Sensores de Razão:** Combinam medições em diferentes ângulos para compensar variações na cor da água e outras interferências. Oferecem maior precisão, mas com maior complexidade e custo (Figura 1.21). (Lambrou et al., 2014).



FIGURA 1. 21 - Turbidity Sensor InPro8610ie

Fonte: Adaptado de <https://www.mt.com/br/pt/home/products/Process-Analytics/turbidity-meter/inpro8600i>

Para plataformas de monitoramento baseadas em Arduino, sensores nefelométricos infravermelhos (como o ST100) (Figura 1.22) são os mais

utilizados, oferecendo boa relação custo-benefício. Estes sensores tipicamente utilizam um LED infravermelho como fonte de luz e um fotodetector posicionado a 180° para medir a luz dispersa. A calibração é realizada com padrões de formazina ou equivalentes (Sayahi et al., 2019).



FIGURA 1. 22 - Sensor de Turbidez para Arduino ST100 + Módulo de Leitura  
Fonte: Adaptado de <https://www.usinainfo.com.br/outros-sensores-arduino/sensor-de-turbidez-para-arduino-st100>

### 1.2.3 Desafios na Integração de Sensores

A integração de múltiplos sensores em uma única plataforma de monitoramento apresenta diversos desafios técnicos que precisam ser superados para garantir medições confiáveis e operação robusta em campo:

- **Interferências Cruzadas:** Sensores próximos podem interferir entre si, seja eletronicamente (ruído, acoplamento capacitivo) ou fisicamente (alteração do fluxo de água, sombreamento). O design cuidadoso do layout físico e a blindagem adequada são essenciais (Wagner et al., 2006).
- **Calibração Multivariável:** Muitos parâmetros são interdependentes (ex: pH e temperatura, condutividade e temperatura), exigindo calibração cruzada e compensação algorítmica (De Camargo et al., 2023).
- **Consumo Energético:** O monitoramento contínuo com múltiplos sensores demanda energia significativa, um desafio para sistemas autônomos alimentados por baterias ou energia solar. Estratégias de gerenciamento de energia, como amostragem intermitente e modos de baixo consumo, são necessárias (Khorasani; Naji, 2017).
- **Bioincrustação:** O crescimento de biofilmes e a acumulação de detritos nos sensores podem alterar significativamente as medições ao

longo do tempo. Sistemas de auto limpeza ou manutenção regular são necessários para operação prolongada (Delauney; Compère; Lehaitre, 2010).

- **Robustez Ambiental:** Sistemas de monitoramento em campo estão sujeitos a condições adversas (variações de temperatura, umidade, radiação UV, submersão), exigindo encapsulamento adequado e materiais resistentes (Glasgow et al., 2004).
- **Processamento e Transmissão de Dados:** A aquisição simultânea de múltiplos parâmetros gera volumes significativos de dados que precisam ser processados, armazenados e transmitidos eficientemente, considerando as limitações de memória e largura de banda (Hart; Martinez, 2006).

#### **1.2.4 Miniaturização de Sensores: Perspectivas e Abordagens**

A miniaturização de sensores para monitoramento ambiental representa uma fronteira tecnológica promissora, com potencial para reduzir custos, consumo energético e impacto ambiental dos sistemas de monitoramento.

Os desafios na miniaturização de sensores para monitoramento ambiental incluem manter a precisão e estabilidade em tamanho reduzido, garantir robustez em condições adversas, e desenvolver interfaces adequadas com sistemas de aquisição de dados. Apesar destes desafios, a tendência de miniaturização continua a avançar, impulsionada por aplicações em redes de sensores distribuídos, monitoramento pessoal e dispositivos vestíveis (Korostynska; Mason; Al-Shamma, 2013).

##### **Principais Abordagens de Miniaturização:**

- **Tecnologia MEMS (Sistemas Microeletromecânicos):** Integra elementos mecânicos, sensores, atuadores e eletrônica em substratos de silício através de técnicas de microfabricação. Sensores MEMS de pH, temperatura e condutividade já estão disponíveis comercialmente, oferecendo tamanho reduzido

e baixo consumo energético (Figura 1.23). (Brand; Pourkamali, 2015).



FIGURA 1. 23 - MEMS – Sistemas Microeletromecânicos  
Fonte: Adaptado de <https://embarcados.com.br/mems>

- **Sensores Impressos:** Utilizam técnicas de impressão (serigrafia, inkjet, impressão 3D) para depositar materiais funcionais (condutores, semicondutores, polímeros) em substratos flexíveis ou rígidos. Oferecem potencial para produção em massa de baixo custo, mas geralmente com menor precisão que sensores convencionais (Figura 1.24) (Matzeu; Florea; Diamond, 2015).

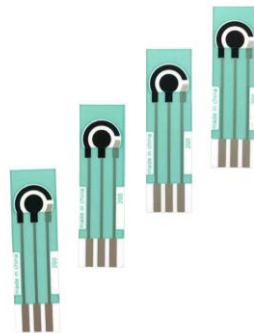


FIGURA 1. 24 - DTLstar C-200 Electrochemical Sensor Silk Screen Printed  
Fonte: Adaptado de <https://www.alibaba.com/product-detail/electrochemical-sensor-Electrochemical>

- **Lab-on-a-Chip:** Integram múltiplas funções analíticas em um único dispositivo miniaturizado, frequentemente combinando microfluídica e sensores eletroquímicos ou ópticos. Permitem análises complexas com volumes mínimos de amostra (Figura 1.25) (Whitesides, 2006).

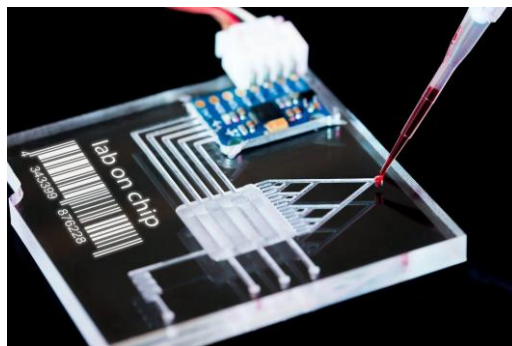


FIGURA 1. 25 - Lab-on-a-chip form parameter inspection

Fonte: Adaptado de <https://www.polytec.com/int/surface-metrology/areas-of-application/b/lab-on-a-chip>

- **Nanosensores:** Utilizam nanomateriais (nanotubos de carbono, nanopartículas, grafeno) como elementos sensores, aproveitando suas propriedades únicas e alta relação superfície/volume. Ainda predominantemente em fase de pesquisa para aplicações ambientais (Figura 1.26) (Willner; Vikesland, 2018).

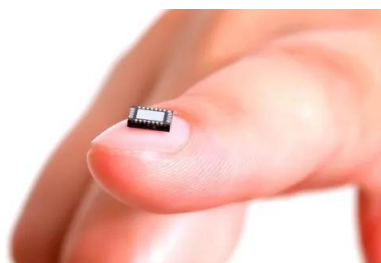


FIGURA 1. 26 - Um exemplo de nanosensor

Fonte: Adaptado de <https://ahsen35813.medium.com/nanosensors-11235813-39333fdd4c13>

Dada essa perspectiva, o desenvolvimento deste trabalho visa a utilização de tecnologias de *Open source* e os sensores de baixo custo encontrados no mercado.

## 2 OBJETIVOS

O presente trabalho tem como objetivo geral o desenvolvimento e a validação de uma plataforma de baixo custo para monitoramento in-situ da qualidade da água, baseada na integração de sensores comerciais com a plataforma Arduino.

Para alcançar este objetivo geral, foram definidos os seguintes objetivos específicos:

- Integrar hardware e software dos sensores comerciais selecionados, especificamente, sensor de temperatura, sensor de condutividade elétrica, sensor de pH e sensor de turbidez, com a placa microcontroladora Arduino UNO R3;
- Desenvolver o código embarcado (firmware) para a placa Arduino, capaz de realizar a aquisição de dados de forma confiável e, preferencialmente, simultânea ou quase-simultânea dos múltiplos sensores integrados;
- Implementar um sistema para a transmissão remota dos dados coletados pela plataforma;
- Avaliar o desempenho da plataforma integrada por meio de testes em laboratório, incluindo a calibração individual dos sensores e a verificação da precisão e estabilidade das leituras conjuntas;
- Se possível, realizar testes preliminares em ambiente de campo;
- Integrar a plataforma a um sistema de alimentação autônomo, utilizando painel solar e bateria, visando a operação contínua e independente da rede elétrica.

## 3 MATERIAIS E MÉTODOS

### 3.1 Visão Geral do Sistema de Monitoramento

O sistema de monitoramento da qualidade da água desenvolvido consiste em uma plataforma de baixo custo baseada no microcontrolador Arduino UNO R3, integrando múltiplos sensores comerciais para medição de parâmetros críticos (Figura 3.1). A arquitetura do sistema foi projetada para ser modular e de fácil replicação.

#### Componentes Principais:

- Microcontrolador: Arduino UNO R3
- Sensor de Temperatura: DS18B20
- Sensor de pH: PH4502C
- Sensor de Condutividade Elétrica: Medidor TDS (TDS Meter V1.0)
- Sensor de Turbidez: ST100 (com módulo MC-9011)

#### Diagrama de Circuitos:

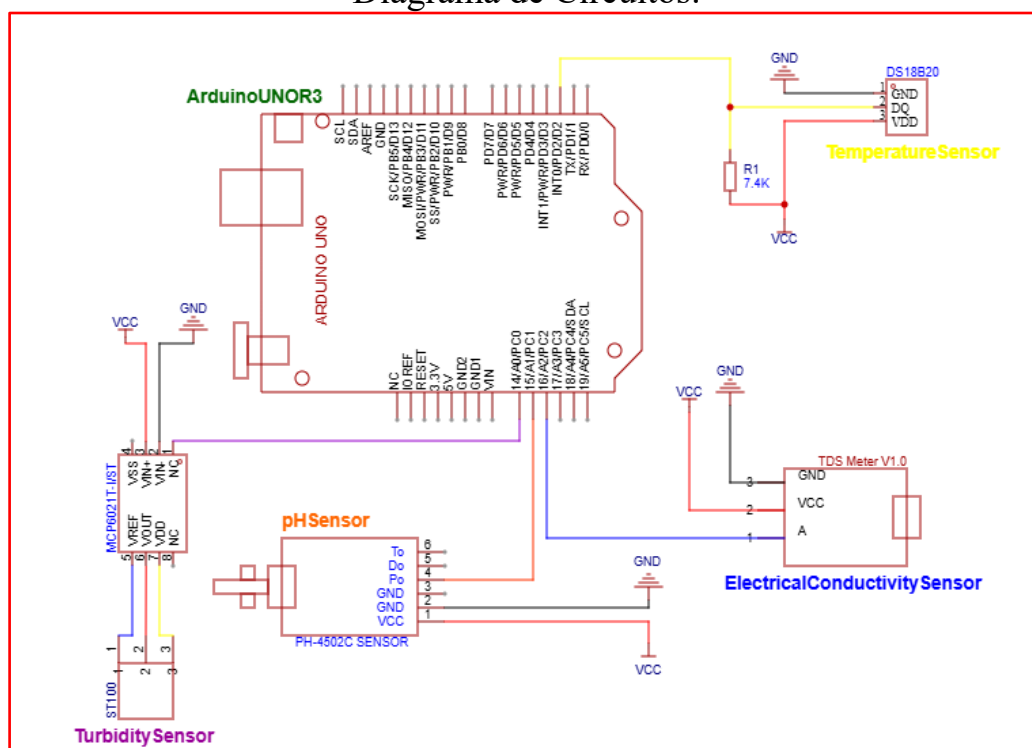


FIGURA 3. 1 - Diagrama de Circuitos do Sistema Integrado

Fonte: Gerado pela autora via: <https://easyeda.com>

O diagrama esquemático detalha as conexões elétricas entre o

Arduino UNO R3 e os sensores. O sensor de temperatura DS18B20 é conectado a uma porta digital do Arduino, utilizando um resistor pull-up de 4.7k $\Omega$ , característico para comunicação One-Wire. O sensor de pH (PH4502C) e o sensor de condutividade elétrica (TDS Meter V1.0) são conectados às portas analógicas do Arduino (A0 e A1, respectivamente). O sensor de turbidez (ST100) é conectado ao módulo MC-9011, que por sua vez se conecta a uma porta analógica do Arduino (A2).

### Sistema Montado:

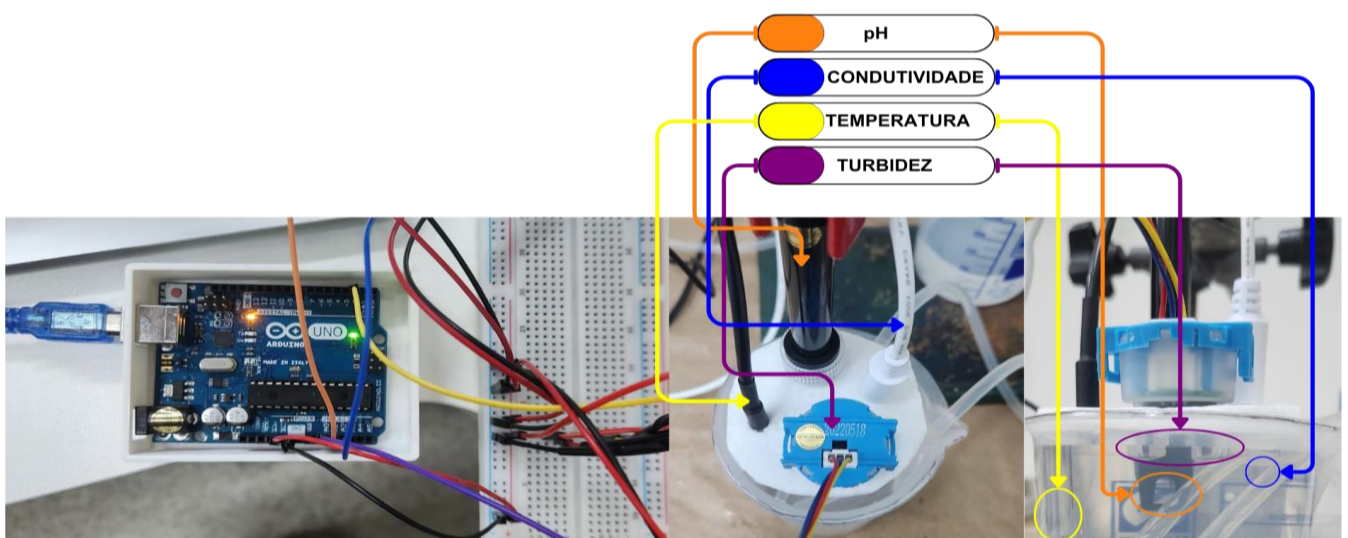


FIGURA 3. 2 - Representação do sistema integrado montado durante medida  
Fonte: A autora

A Figura 3.2 ilustra a implementação física da plataforma. É possível observar a protoboard utilizada para as conexões dos sensores ao Arduino, bem como o arranjo dos sensores em um béquer para testes. As fotos mostram os sensores submersos na amostra para obtenção dos dados.

### Princípios de Operação:

A plataforma opera através da aquisição contínua de dados dos sensores. O Arduino lê os sinais analógicos e digitais de cada sensor, processa esses dados (incluindo compensação de temperatura para pH e condutividade, se aplicável) e os prepara para transmissão. O sistema é projetado para monitoramento in-situ, permitindo a coleta de dados em tempo real ou em

intervalos programados, o que é crucial para a detecção de eventos de poluição de curta duração e para a gestão eficiente dos recursos hídricos.

## **3.2 Aquisição de Dados de Qualidade da Água**

### **3.2.1 Medição de Temperatura**

Os testes de desempenho do sensor de temperatura DS18B20 (Figura 3.3) foram realizados em ambiente controlado, especificamente em uma sala termostaticada mantida a 24°C com o uso de ar-condicionado, com a amostra em constante agitação sendo aquecida e resfriada ao longo do tempo. O objetivo principal desta etapa foi avaliar a estabilidade e a capacidade do sensor em registrar variações de temperatura em comparação com um termômetro de bulbo de referência, em vez de uma calibração multiponto formal.



FIGURA 3. 3 - Sensor de Temperatura DS 18B20 à Prova D'Água  
Fonte: Adaptado de [UsinaInfo-DS18B20](#)

Para a medição de temperatura, foi utilizado um termômetro sem mercúrio da marca Alla France (produzido no Brasil) como referência. A amostra foi mantida em agitação constante utilizando um agitador magnético IKA Topolino. O sensor DS18B20 foi mergulhado na amostra de forma que toda a ponta de prova metálica ficasse submersa, garantindo o contato adequado com o meio a ser monitorado.

O circuito do sensor DS18B20 foi montado em uma protoboard, conectando-se a resistência *pull-up* entre a saída de sinal e a entrada de energia, conforme o esquema elétrico padrão para este sensor. A aquisição de dados foi realizada utilizando a IDE do Arduino na versão 1.8.19. Inicialmente, as leituras

eram acompanhadas via Monitor Serial do Arduino, com gravação da tela do computador para registro das variações. O intervalo de leitura variou entre 1s e 1h, visando compreender a resolução dos dados para otimização da leitura.

A avaliação da estabilidade e da precisão de sensores em sistemas de monitoramento integrados é um passo crucial para garantir a confiabilidade dos dados coletados. Neste estudo, a performance do sensor de temperatura digital DS18B20, operando em conjunto com sensores de pH, condutividade elétrica e turbidez via uma plataforma Arduino, foi investigada sob condições controladas. Os experimentos foram conduzidos em duplicata, utilizando como meios uma solução padrão de condutividade (1413  $\mu\text{S}/\text{cm}$ ) e água destilada, com o ambiente termostaticado a 24 °C e agitação constante da solução.

Não foram realizadas modificações de miniaturização específicas no sensor de temperatura além de seu encapsulamento padrão à prova d'água, o que já o torna compacto e robusto para imersão em ambientes aquáticos.

### 3.2.2 Medição de Condutividade

A calibração do sensor de condutividade (TDS Meter V1.0) (Figura 3.4) foi realizada utilizando soluções padrão com concentrações de 10, 100, 500, 1000 e 1413  $\mu\text{S}/\text{cm}$ . O processo de calibração foi acompanhado pelo sensor de temperatura, e os dados de temperatura foram utilizados para a correção da condutividade obtida, garantindo maior precisão nas medições.

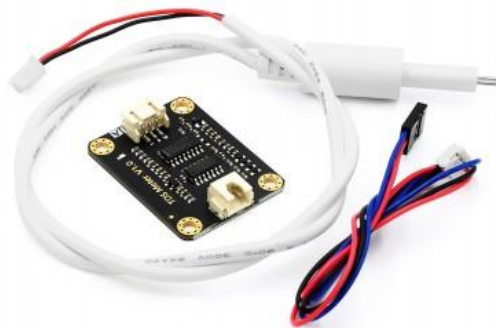


FIGURA 3. 4 - Medidor de TDS Sensor de Condutividade da Água Analógico  
Fonte: Adaptado de [UsinaInfo-TDS](#)

As medições foram realizadas em água destilada e em uma solução

padrão de 1413  $\mu\text{S}/\text{cm}$ , mantendo as condições controladas de sala termostaticada a 24°C e agitação constante da solução.

O controle do sensor de condutividade foi realizado de forma similar ao sensor anterior, utilizando a mesma IDE do Arduino e o mesmo modelo de Arduino, com as bibliotecas específicas para comunicação com o sensor de condutividade. A curva de calibração foi estabelecida utilizando soluções padrão de condutividade nas concentrações de 10, 100, 500, 1000 e 1413  $\mu\text{S}/\text{cm}$ . Durante a calibração, a sonda do sensor foi mantida submersa na solução por 2 minutos para permitir a estabilização do sinal. Entre cada medida, o sensor foi lavado com água destilada e cuidadosamente seco para evitar contaminação cruzada.

Os testes de estabilidade em água destilada e em solução padrão de condutividade foram conduzidos sob as mesmas condições controladas dos experimentos anteriores, incluindo a manutenção da temperatura ambiente a 24°C e a agitação constante da amostra em cada caso. A correção da condutividade pela temperatura foi realizada utilizando os dados do sensor de temperatura, conforme descrito na seção 3.1.1, garantindo a precisão das medições.

Foi desenvolvido um protótipo miniaturizado do sensor de condutividade (Figura 3.5), entretanto somente estudos preliminares foram realizados a fim de se obter a constante de célula do sensor.

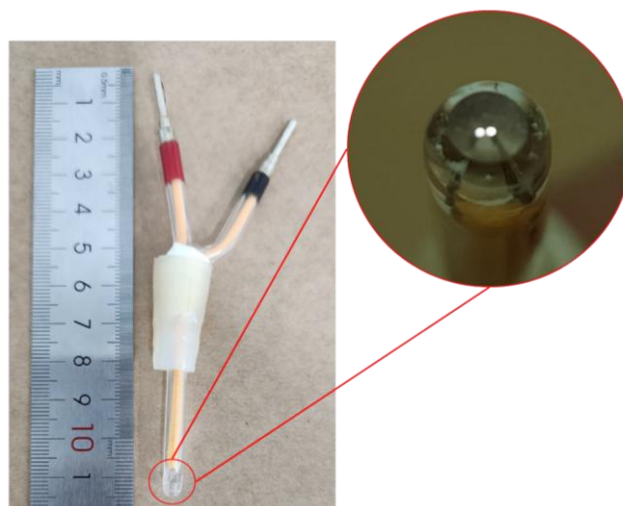


FIGURA 3. 5 - Protótipo de Sensor de Condutividade Miniaturizado  
Fonte: A autora

### 3.2.3 Medição de pH

A calibração do sensor de pH (PH4502C) (Figura 3.6) foi realizada utilizando soluções padrão de pH 4, 7 e 10, conforme procedimento recomendado para eletrodos de pH. A relação entre o sinal de saída do sensor e o valor de pH foi linearizada.



FIGURA 3. 6 - Sensor de pH para Arduino + Módulo de Leitura BNC PH4502C  
Fonte: Adaptado de [UsinaInfo-PH](#)

Para a medição de pH, as leituras do sensor PH4502C foram comparadas com um Indicador Universal de Fita de pH (pH 0 - 14, MQuant® da Merck). As amostras foram mantidas em agitação constante utilizando o agitador magnético IKA Topolino, em um béquer de 250 mL.

A calibração do sensor de pH foi realizada mergulhando o eletrodo em soluções tampão padrão da Êxodo Científica nos pHs 4, 7 e 10. Em cada etapa de calibração, a ponta do eletrodo de vidro foi mantida em contato com a solução por 5 minutos. Após cada medida, o eletrodo foi lavado com água destilada, seguido de um intervalo de 5 minutos antes da próxima imersão. Para as medidas em solução tampão de pH 7 e em água destilada, o sensor permaneceu imerso por mais de 15 horas, sob agitação constante, para avaliar a estabilidade das leituras a longo prazo.

A aquisição de dados foi realizada utilizando a mesma IDE do Arduino (versão 1.8.19) e o mesmo modelo de Arduino previamente mencionado, com a inclusão das bibliotecas específicas necessárias para a comunicação com o sensor de pH e seguindo as mesmas práticas laboratoriais descritas anteriormente, com controle da temperatura da sala e agitação constante da amostra.

Para avaliar o desempenho do sensor de pH miniaturizado desenvolvido baseado no trabalho de (Zimer et al., 2010), (Figura 3.7) foi realizada uma titulação de HCl com NaOH (Figura 3.8), comparando suas leituras com as de um pHmetro de bancada de referência.

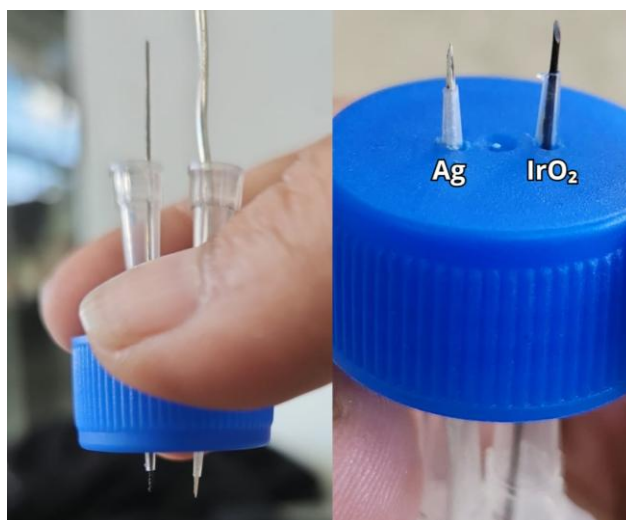


FIGURA 3. 7 - Protótipo de Sensor de pH Miniaturizado

Fonte: A autora

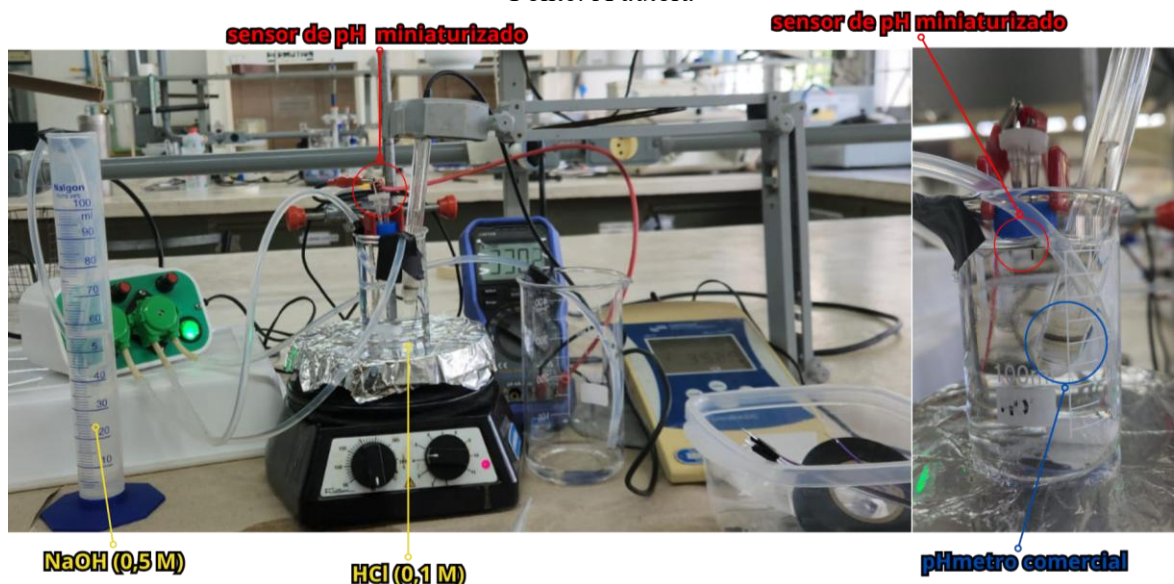


FIGURA 3. 8 - Sistema de Titulação para estudo do funcionamento do sensor de pH miniaturizado

Fonte: A autora

### 3.2.4 Medição de Turbidez

Para a medição de turbidez, foi utilizado o sensor ST100 (Figura 3.9). As soluções padrão utilizadas para calibração foram fornecidas prontas pela marca Specsol. A calibração foi realizada visando estabelecer a relação entre a leitura do sensor e os níveis de turbidez, ajustando a relação a um modelo

polinomial de segunda ordem. O sensor de temperatura foi utilizado em conjunto para a correção da leitura de turbidez, garantindo maior precisão, uma vez que a turbidez é sensível à temperatura.



FIGURA 3. 9 - Sensor de Turbidez para Arduino ST100 + Módulo de Leitura  
Fonte: Adaptado de [UsinaInfo-ST100](#)

As medições foram conduzidas em água destilada e em soluções padrão, sob condições controladas de temperatura (24°C) e agitação constante da solução, similares aos experimentos dos demais sensores. Durante os testes, foi observado que a variação da luz ambiente na sala influencia as leituras do sensor de turbidez, sendo um fator a ser considerado na análise dos resultados. Não foi realizada miniaturização específica para este sensor.

O controle do sensor de turbidez foi realizado de forma similar aos sensores anteriores, utilizando a IDE do Arduino (versão 1.8.19) e o mesmo modelo de Arduino UNO R3. Foi utilizada uma biblioteca específica para a calibração do sensor e salvamento dos dados na EEPROM (Electrically Erasable Programmable Read-Only Memory) do sensor, que consiste em um tipo de memória não volátil usada para armazenar dados que precisam ser retidos mesmo quando o dispositivo é desligado, permitindo a persistência dos valores de calibração.

### **3.3 Processamento e Controle**

#### **3.3.1 Plataforma Arduino**

A plataforma central para o processamento e controle do sistema de

monitoramento é o microcontrolador Arduino UNO R3. O Arduino UNO R3 é uma placa de desenvolvimento de código aberto baseada no microcontrolador ATmega328P. Ele possui 14 pinos digitais de entrada/saída (dos quais 6 podem ser usados como saídas PWM), 6 pinos de entrada analógica, um cristal oscilador de 16 MHz, uma conexão USB, um jack de alimentação, um cabeçalho ICSP e um botão de reset. Esta plataforma foi escolhida devido à sua facilidade de uso, vasta comunidade de suporte, baixo custo e flexibilidade para integração com diversos sensores e módulos, tornando-a ideal para prototipagem e desenvolvimento de sistemas embarcados (Monk, 2021).

### **3.3.2 Firmware Desenvolvido**

O firmware desenvolvido para a plataforma Arduino UNO R3 é responsável pela aquisição, processamento e, posteriormente, pela integração dos dados provenientes dos sensores de temperatura, condutividade, pH e turbidez. O desenvolvimento ocorreu de forma modular, iniciando com códigos independentes para cada sensor e evoluindo para uma solução integrada. A linguagem de programação utilizada é baseada em C/C++, empregando a IDE do Arduino.

#### **3.3.2.1 Firmware Individual dos Sensores**

Inicialmente, foram desenvolvidos e testados códigos para cada sensor individualmente, garantindo o correto funcionamento e a compreensão de suas respectivas bibliotecas e lógicas de leitura.

##### **Sensor de Temperatura (DS18B20):**

O código (Apêndice A) utiliza as bibliotecas `OneWire.h` e `DallasTemperature.h`. Ele define o pino digital (`PINO_ONEWIRE = 2`) para comunicação 1-Wire, inicializa o sensor e, no loop principal, solicita a leitura da temperatura (`sensors.requestTemperatures()`) e obtém o valor em Celsius (`sensors.getTempCByIndex(0)`). A temperatura é então

impressa no Serial Monitor. Um `delay(1000)` é utilizado para aguardar 1 segundo entre as leituras.

A Figura 3.10 é um diagrama de blocos desenvolvido para facilitar o entendimento de cada segmento do código e será aplicado para todos os códigos descritos a seguir, onde toda função dentro do bloco rosa é referente a inicialização e definição dos pinos, bibliotecas e entre outras funções que ocorrem no momento que o arduino é energizado, o bloco azul contém funções referentes a leitura e detalhes de processamento do sensor, o bloco lilás se refere ao monitor serial, a interface com o usuário, e tudo isso integra o *looping* da lógica da linguagem empregada na comunicação com o arduino.

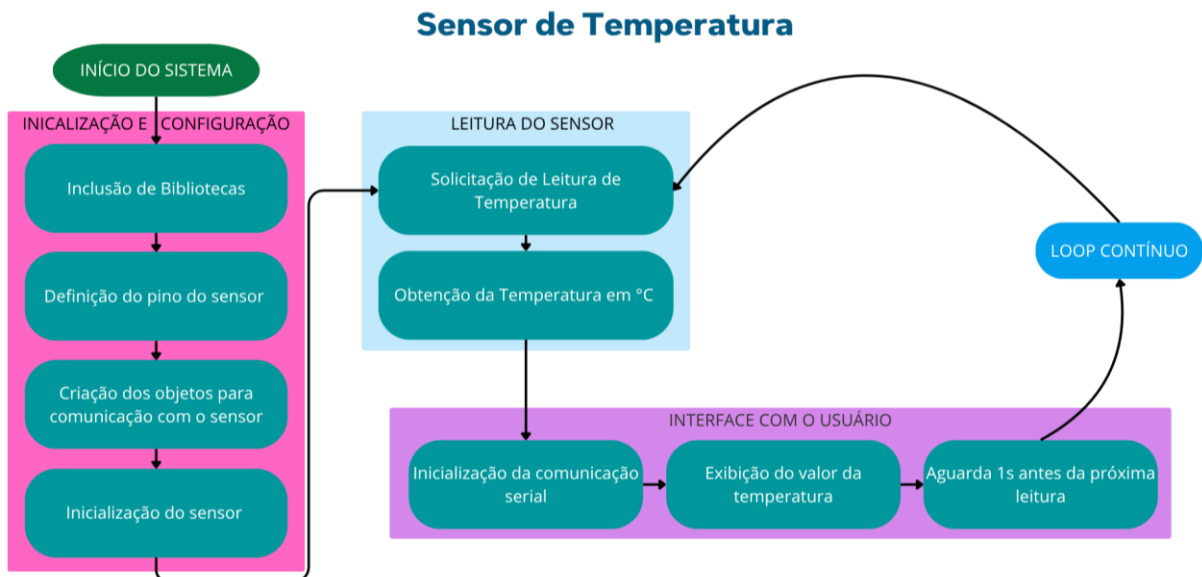


FIGURA 3. 10 - Diagrama de Blocos referente ao código do Sensor de Temperatura (DS18B20)  
Fonte: A autora.

### Sensor de Condutividade (Medidor TDS):

O código (Apêndice B) emprega a biblioteca `GravityTDS.h` e `EEPROM.h`. Ele define o pino analógico (`TdsSensorPin = A0`) para o sensor TDS, configura a tensão de referência (5.0V) e a faixa do ADC (1024 para 10 bits). No loop, a função `gravityTds.update()` realiza a amostragem e o cálculo do valor de TDS, que é então obtido por `gravityTds.getTdsValue()` e impresso no Serial Monitor. A temperatura

é um parâmetro importante para a compensação da condutividade, e o código original prevê a leitura de um sensor de temperatura.

A Figura 3.11 é um diagrama de blocos desenvolvido para facilitar o entendimento de cada segmento do código do sensor de condutividade, que inicialmente não tinha funções claramente definidas para calibração do sensor, além da possibilidade de incluir as medidas do sensor de temperatura para correção da condutividade, representada como TDS no código inicial.

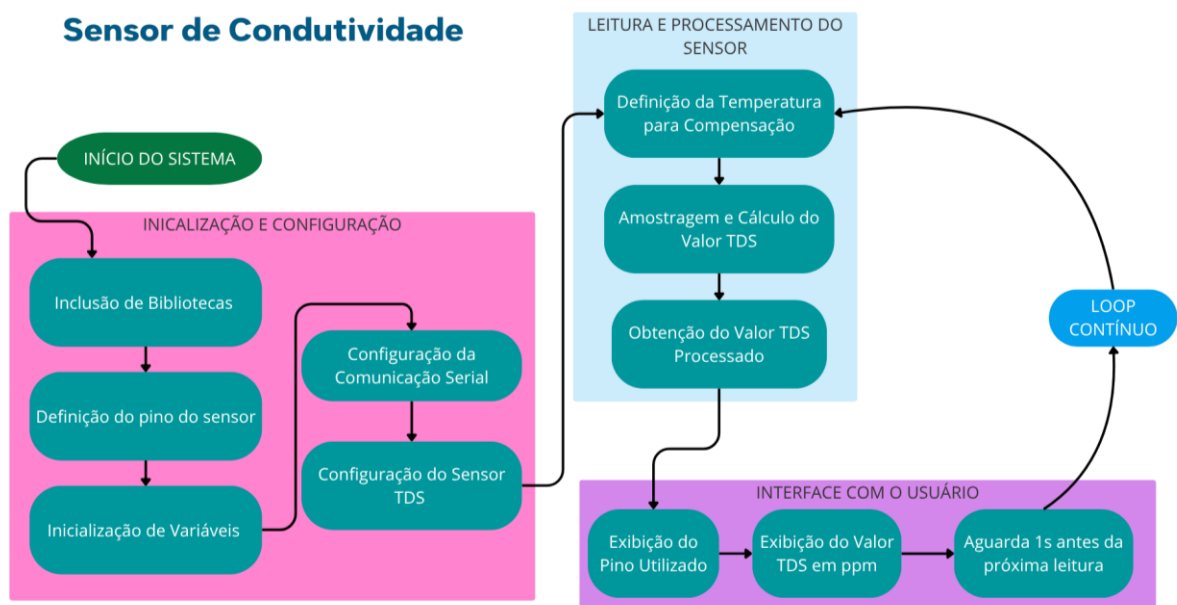


FIGURA 3. 11 - Diagrama de Blocos referente ao código do Sensor de Condutividade (TDS)  
Fonte: A autora.

### Sensor de pH (PH4502C):

O código (Apêndice C) realiza a leitura do sensor de pH conectado ao pino analógico A0. Ele implementa uma rotina de calibração que requer a definição de tensões obtidas em soluções de pH 7, 4 e/ou 10. A leitura do sensor é feita através de múltiplas amostras (10 amostras), que são ordenadas e uma média é calculada a partir das 6 amostras centrais para reduzir ruído. A tensão média é então convertida para o valor de pH utilizando uma equação linear ( $ph = m * tensao + b$ ), onde m e b são constantes calculadas a partir dos pontos de calibração. O valor de pH é impresso no Serial Monitor.

A Figura 3.12 é um diagrama de blocos desenvolvido para facilitar o entendimento de cada segmento do código do sensor de pH.

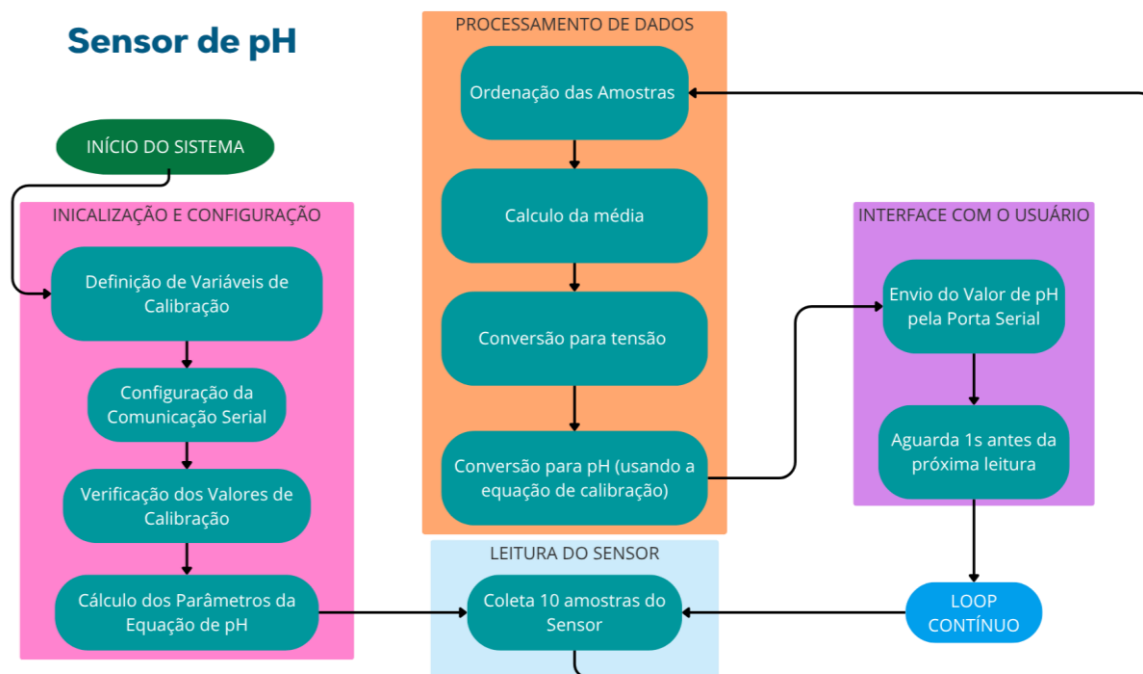


FIGURA 3. 12 - Diagrama de Blocos referente ao código do Sensor de pH (PH4502C)  
 Fonte: A autora.

Note que a calibração do sensor está inclusa no bloco de inicialização, e o bloco laranja presente pela primeira vez, nesse diagrama, representa a sessão de processamento de dados de forma isolada da leitura que coleta as amostras para o cálculo mencionado anteriormente.

### **Sensor de Turbidez (ST100):**

O código (Apêndice D) lê o sensor de turbidez conectado ao pino analógico A0. Ele realiza múltiplas leituras (800 amostras) para calcular uma média da voltagem lida. A voltagem é então convertida para o valor de NTU (Unidades Nefelométricas de Turbidez) utilizando uma função polinomial. O código também inclui um sistema de arredondamento para a leitura da voltagem. O valor de NTU é impresso no Serial Monitor.

A Figura 3.13 é um diagrama de blocos desenvolvido para facilitar o entendimento de cada segmento do código do sensor de turbidez, que representa também uma função auxiliar de arredondamento no bloco cinza.

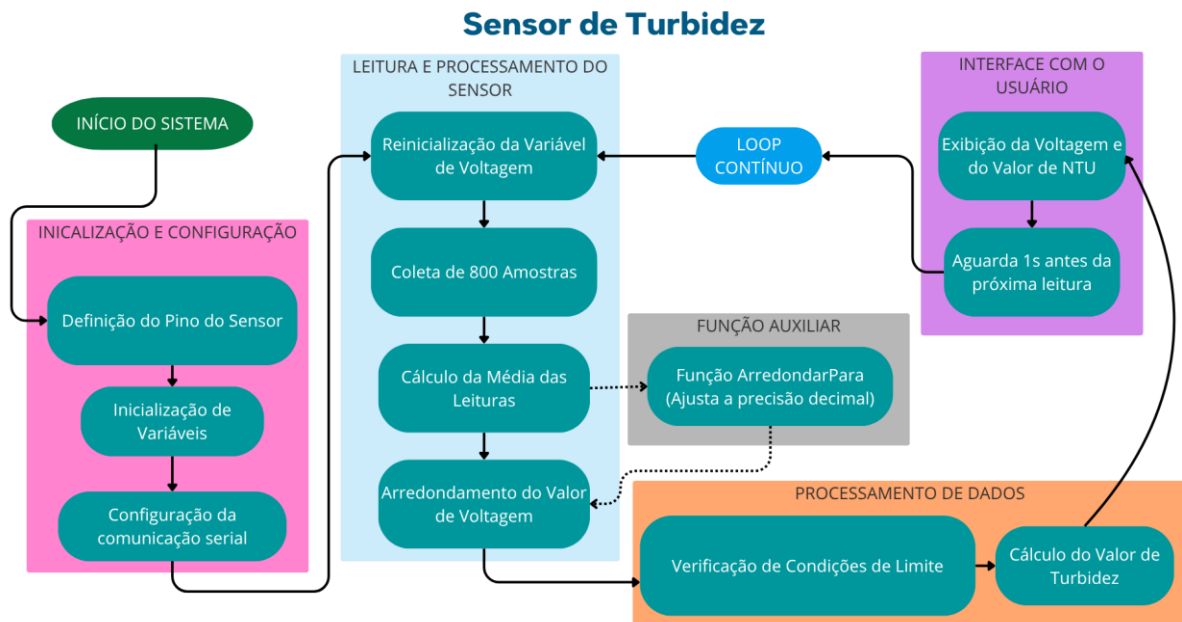


FIGURA 3. 13 - Diagrama de Blocos referente ao código do Sensor de Turbidez (ST10)  
Fonte: A autora.

### 3.3.2.2 Evolução do Firmware Integrado

O desenvolvimento do firmware seguiu uma abordagem incremental, integrando os sensores passo a passo para garantir a funcionalidade e a estabilidade de cada combinação antes de adicionar um novo componente. A evolução do código pode ser acompanhada pelos Apêndices E e F.

Essa metodologia permitiu identificar e resolver desafios de compatibilidade e otimização de recursos, com a reorganização e inclusão de funções na sequência representada na Figura 3.14 abaixo.



FIGURA 3. 14 - Diagrama representando a sequência de integração dos sensores  
Fonte: A autora.

### **Integração Final:**

O diagrama de blocos representado na Figura 3.15 é referente ao código de integração dos sensores (Apêndice G), onde é possível observar como as correções realizadas nos códigos ao longo da integração, aprimorando a clareza na comunicação entre cada componente, e conseqüentemente a possibilidade de obtenção de dados mais fideis aos parâmetros da amostra. Cada bloco representa um grupo de funções do código integrado:

- Bloco Rosa - Inicialização e Configuração: Inclui todas as bibliotecas dos sensores, define cada pino utilizado, se comunica com cada EEPROM possibilitando a calibração dos sensores, zera todas as variáveis e completa a comunicação com o computador.
- Bloco Verde - Calibração dos Sensores: Se durante a transmissão serial a placa receber o comando “enter” essa função é ativada para realizar a calibração com soluções padronizadas para cada equipamento (PH4502C e TDS Meter V1.0), com o comando “sair” encerra a calibração.
- Bloco Azul - Leitura dos Sensores: Faz a leitura das informações, digitais ou analógicas, obtidas por cada sensor.
- Bloco Laranja - Processamento de Dados: Calcula os valores obtidos a partir das equações de calibração e correção de cada sensor, e converte os sinais em cada parâmetro em análise.
- Bloco Lilás - Interface com o Usuário: Exibe as leituras convertidas, processa os comandos de calibração.

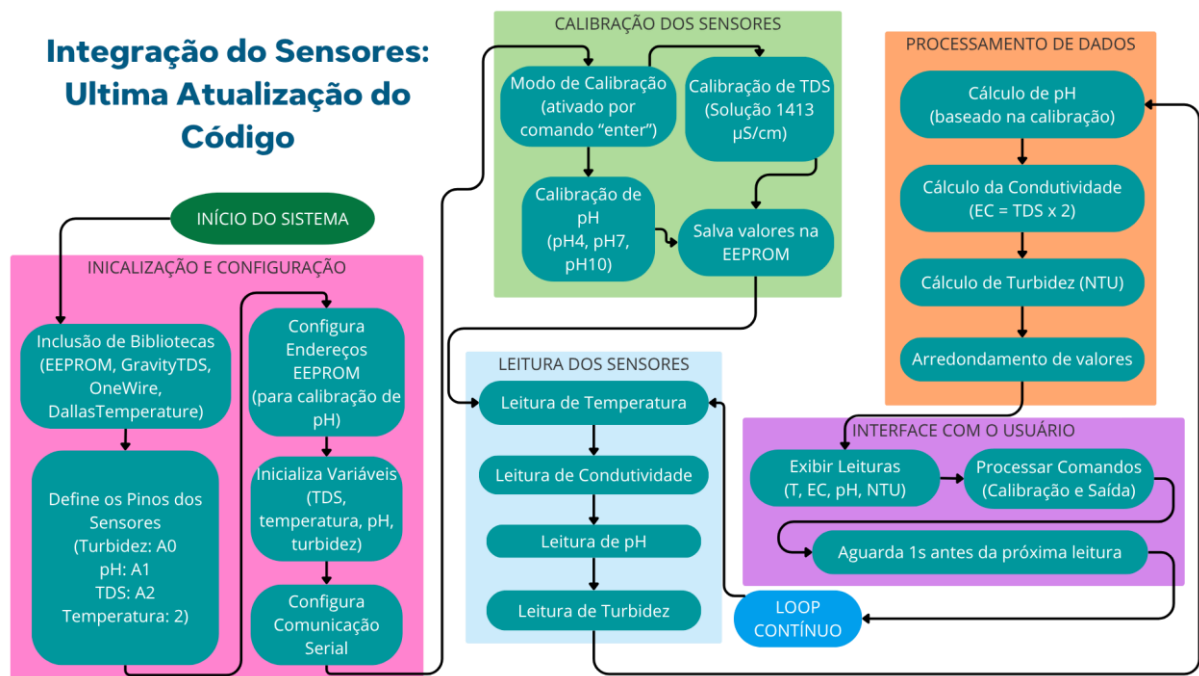


FIGURA 3. 15 - Diagrama de Blocos referente ao código de integração dos sensores  
Fonte: A autora.

A integração dos sensores foi realizada de forma sequencial e modular, permitindo a validação de cada etapa antes da incorporação de um novo sensor. A lógica de programação garante que as leituras sejam realizadas de forma coordenada, e que as dependências entre os sensores (como a compensação de temperatura para as leituras de condutividade e pH) sejam devidamente consideradas. A comunicação serial é utilizada para depuração e para a interação com o usuário, permitindo a calibração dos sensores em tempo real. A utilização da memória EEPROM garante que os valores de calibração sejam persistidos mesmo após o desligamento do sistema, otimizando o processo de inicialização em campo.

### **3.4 Transmissão e Armazenamento de Dados**

#### **3.4.1 Sistema de Comunicação**

O sistema de comunicação da plataforma de monitoramento foi implementado através de uma interface serial entre o microcontrolador Arduino e o computador host. Esta arquitetura permite a transmissão em tempo real dos dados coletados pelos sensores, garantindo a integridade e a sincronização das

informações durante o processo de aquisição.

A comunicação serial foi estabelecida utilizando o protocolo UART (*Universal Asynchronous Receiver-Transmitter*) nativo do Arduino, operando a uma taxa de transmissão (*baud rate*) de 9600 bps. Esta velocidade foi selecionada para garantir a estabilidade da comunicação e compatibilidade com diferentes sistemas operacionais, proporcionando um equilíbrio adequado entre velocidade de transmissão e confiabilidade dos dados. O Arduino foi programado para transmitir os dados dos sensores em formato de texto estruturado através da porta serial USB, permitindo a captura e armazenamento das informações no computador receptor.

A estrutura de dados transmitida inclui identificadores únicos para cada sensor, timestamps relativos e os valores medidos em suas respectivas unidades. Este formato padronizado facilita o processamento posterior dos dados e garante a rastreabilidade das medições ao longo do tempo. A comunicação bidirecional também permite o envio de comandos do computador para o Arduino, possibilitando o controle remoto de parâmetros de aquisição, como intervalos de medição e configurações de calibração dos sensores.

### **3.4.2 Armazenamento Local**

Para a captura e armazenamento dos dados transmitidos pela porta serial do Arduino, foi utilizado o software TeraTerm (“Tera Term - Terminal Emulator for Windows”, 2025), um emulador de terminal amplamente reconhecido por sua confiabilidade na comunicação serial. O TeraTerm foi configurado para estabelecer conexão com a porta COM correspondente ao Arduino, utilizando os mesmos parâmetros de comunicação definidos no firmware (9600 bps, 8 bits de dados, 1 bit de parada, sem paridade).

O software foi programado para capturar automaticamente todo o fluxo de dados recebido pela porta serial e salvá-los em arquivos de texto (.txt) com timestamp de criação. Esta funcionalidade de *logging* contínuo permitiu a coleta ininterrupta de dados durante os experimentos de longa duração, alguns dos

quais se estenderam por mais de 14 horas. Os arquivos de texto gerados continham os dados brutos no formato transmitido pelo Arduino, preservando a estrutura original das informações e garantindo a integridade dos dados coletados.

Posteriormente, os arquivos .txt foram convertidos para o formato .csv (*Comma-Separated Values*) para facilitar o tratamento e análise dos dados no software Origin (“OriginLab - Origin and OriginPro - Data Analysis and Graphing Software”, 2025). Esta conversão envolveu a reorganização dos dados em colunas estruturadas, com separação adequada dos valores de cada sensor e seus respectivos timestamps, permitindo a importação direta para ferramentas de análise gráfica e estatística.

### **3.5 Validação do Sistema Integrado**

#### **3.5.1 Metodologia de Testes**

Os ensaios de caracterização e validação dos sensores foram conduzidos seguindo protocolos padronizados que garantiram a simultaneidade das medições e a comparabilidade dos resultados. Durante todos os experimentos, os sensores foram operados simultaneamente, permitindo a correlação temporal entre os diferentes parâmetros monitorados e a avaliação do comportamento integrado do sistema.

Para os ensaios em soluções padrão, cada sensor foi imerso em sua respectiva solução de referência, mantendo as condições ambientais controladas (temperatura de 24°C e agitação constante). Esta abordagem permitiu a calibração individual de cada sensor em condições otimizadas, garantindo a máxima precisão das medições de referência. Nos ensaios em água destilada, todos os sensores foram imersos simultaneamente na mesma amostra sob agitação contínua, proporcionando condições experimentais idênticas para todos os parâmetros monitorados.

A validação dos resultados foi realizada através da comparação com equipamentos de referência externos. Para o sensor de temperatura, utilizou-se

um termômetro de vidro sem mercúrio (Alla France) como padrão de comparação. As medições de pH foram validadas utilizando fitas indicadoras de pH universal (pH 0-14, MQuant® Merck), que forneceram valores de referência para comparação com as leituras do sensor eletrônico. Para os sensores de turbidez e condutividade, não foi possível realizar comparações simultâneas com equipamentos externos devido à indisponibilidade de instrumentos de referência no momento dos experimentos, sendo a validação baseada na consistência interna dos dados e na linearidade das curvas de calibração obtidas.

## **4 RESULTADOS E DISCUSSÃO**

### **4.1 Desempenho do Sistema de Aquisição de Dados**

Depois de observar como os sensores se comportam com medidas obtidas com tempo de delay de 1000 ms e 3600000 ms optou-se por fazer as medidas de estabilidade com obtenção de dados de hora em hora, pois as medidas se mostraram estáveis em ambos os casos, mas no segundo, a incidência do ruído na medida é menor.

#### **4.1.1 Medição de Temperatura**

Os resultados obtidos nas medições com a solução padrão, tanto no ensaio original quanto na replicata, demonstraram uma notável estabilidade térmica. Conforme observado nos gráficos abaixo (Figura 4.1), a temperatura da solução permaneceu praticamente constante ao longo das 15 horas de monitoramento, com flutuações mínimas contidas dentro de uma faixa estreita de aproximadamente 0,2 °C. Este comportamento atesta a robustez do sistema de medição e a eficácia do controle de temperatura ambiental, indicando que, sob condições estáveis, o conjunto de sensores opera de forma confiável e sem gerar interferências térmicas ou elétricas significativas que pudessem comprometer a leitura do sensor DS18B20, confirmando a estabilidade intrínseca do sensor, servindo como base que indica o equilíbrio térmico do sistema durante experimentos.

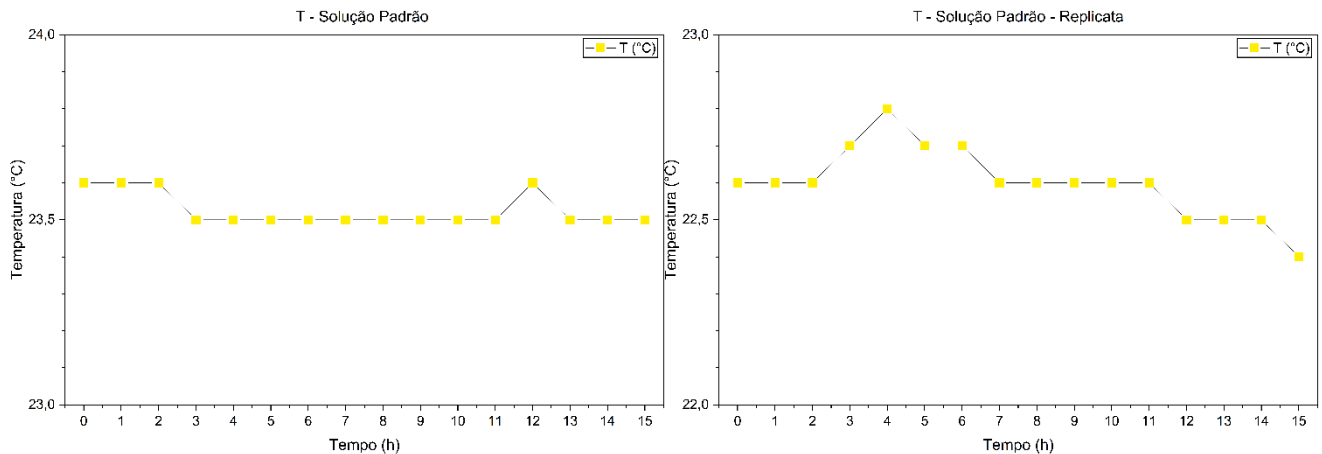


FIGURA 4. 1 – Medida e Replicata da Temperatura em função do tempo para uma solução Padrão 1413  $\mu\text{S}/\text{cm}$   
 Fonte: A autora.

Em contrapartida, os ensaios realizados com água destilada apresentaram resultados divergentes entre si. Enquanto a replicata exibiu um perfil de estabilidade térmica similar ao observado com a solução padrão, o ensaio original (Figura 4.2) revelou uma anomalia: um leve aquecimento gradual e contínuo da amostra. A temperatura, que iniciou em aproximadamente 23,4 °C, elevou-se de forma sistemática até atingir cerca de 24,1 °C ao final do período experimental. Este aumento de 0,7 °C não pode ser atribuído a ruído eletrônico ou ao auto aquecimento do sensor, pois tais efeitos não se manifestaram nos outros três ensaios.

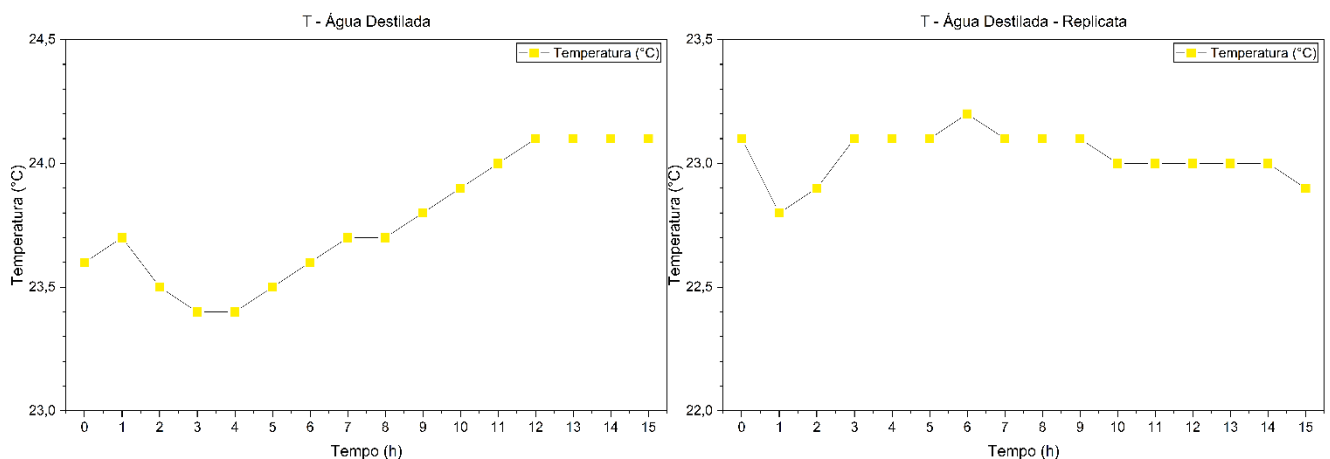


FIGURA 4. 2 – Medida e Replicata da Variação da Temperatura em função do tempo para água destilada  
 Fonte: A autora.

Uma hipótese plausível para explicar esta discrepância reside em um fator experimental externo, especificamente o calor dissipado pelo agitador

magnético utilizado para homogeneizar a amostra, o chamado efeito Joule. A transferência de calor do motor do agitador para a solução foi provavelmente a fonte de energia térmica responsável pelo aquecimento observado. Esta hipótese é corroborada pelo fato de que, neste ensaio específico, foi utilizado um béquer de vidro. O vidro, possuindo uma condutividade térmica superior à de polímeros como o polipropileno (comumente utilizado em recipientes de laboratório), teria facilitado a transferência de calor da placa do agitador para a água destilada. A ausência deste fenômeno na replicata e nos ensaios com a solução padrão sugere que, nestas ocasiões, e/ou se utilizou um recipiente com maior isolamento térmico (e.g., plástico) ou a configuração do agitador (potência, posicionamento) foi diferente, minimizando a dissipação de calor para a amostra.

A análise comparativa dos dados demonstra que o sistema eletrônico de medição é inerentemente estável. A anomalia de aquecimento registrada em um dos ensaios com água destilada não se origina de uma falha ou limitação dos sensores, mas sim de uma variável não controlada no procedimento experimental. Este estudo de caso ressalta a importância crítica do controle rigoroso de todas as variáveis experimentais, incluindo as fontes de calor parasitas como equipamentos auxiliares, e da escolha adequada de materiais (e.g., material do recipiente) para garantir a integridade de medições térmicas sensíveis e de longa duração.

#### **4.1.2 Medição de Condutividade**

A curva de calibração resultante (Figura 4.3) demonstrou uma excelente linearidade, com valores de R-Square (COD) de 0,99886 e R-Square ajustado (Adj. R-Square) de 0,99849. Esses valores confirmam a alta correlação entre o sinal do sensor e a condutividade das soluções padrão, indicando um modelo de calibração robusto.

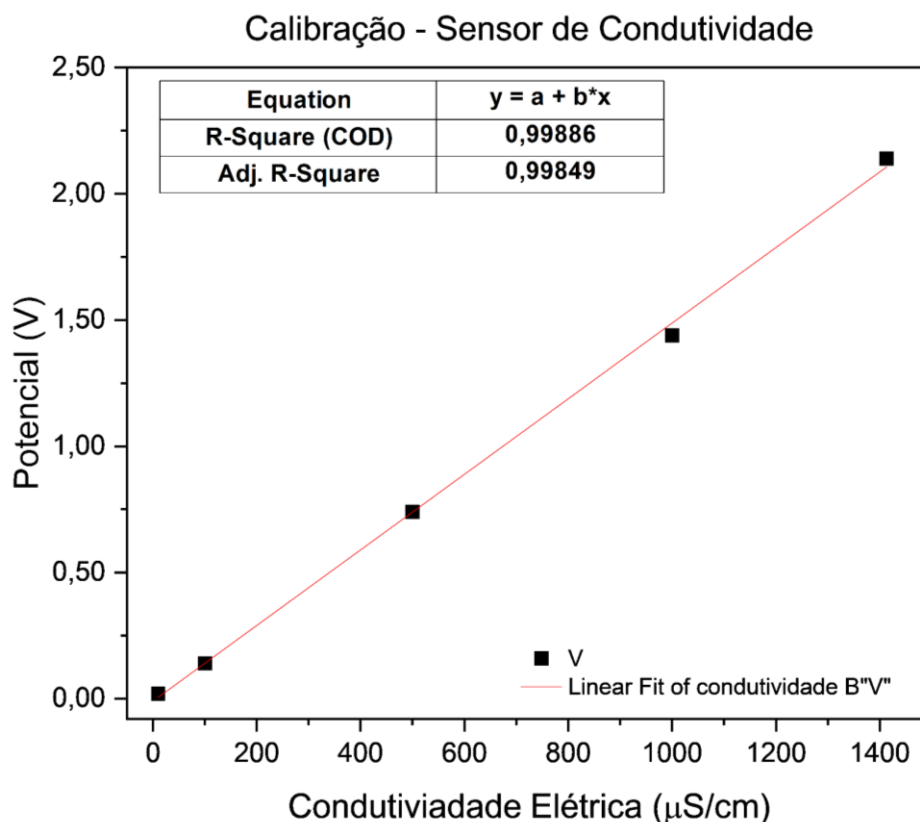


FIGURA 4. 3 - Relação entre a CE Padrão e o Potencial medido pelo sensor de TDS  
 Fonte: A autora.

A calibração do sensor de condutividade demonstrou uma resposta linear satisfatória ( $R^2 > 0,99$ ), validando seu uso para medições quantitativas. Contudo, os ensaios de estabilidade revelaram dinâmicas complexas que dependem tanto do meio quanto das condições experimentais.

Nos ensaios com água destilada, (Figura 4.4) observou-se um aumento consistente e repetível da condutividade elétrica (CE) ao longo do tempo, partindo de  $\sim 3 \mu\text{S}/\text{cm}$  e estabilizando em  $\sim 15 \mu\text{S}/\text{cm}$ . Este comportamento não representa um drift do sensor, mas sim um fenômeno químico bem documentado: a absorção de dióxido de carbono ( $\text{CO}_2$ ) atmosférico pela água. A agitação contínua da amostra maximiza a interface ar-líquido, acelerando a formação de ácido carbônico e sua subsequente dissociação em íons condutores.

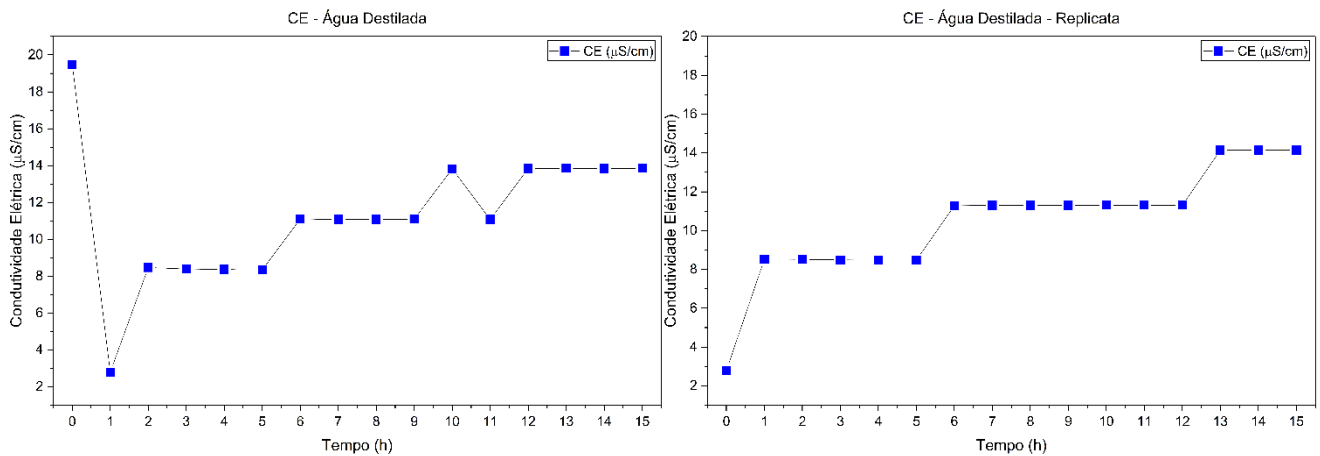


FIGURA 4. 4 - Medida e Replicata da Variação da Condutividade Elétrica em função do tempo para água destilada  
Fonte: A autora.

A boa repetibilidade entre os ensaios confirma que o sensor estava medindo com precisão uma mudança real na composição química da amostra.

Os ensaios na solução padrão de 1413  $\mu\text{S}/\text{cm}$  apresentaram resultados díspares. A replicata demonstrou boa estabilidade e exatidão razoável, com a CE permanecendo constante em torno de 1500  $\mu\text{S}/\text{cm}$ . Em contraste, o ensaio original (Figura 4.5) exibiu um drift positivo significativo, com a CE aumentando de  $\sim 1650$   $\mu\text{S}/\text{cm}$  para  $\sim 1700$   $\mu\text{S}/\text{cm}$ . Correlacionando este resultado com análises térmicas prévias do mesmo sistema, é provável que este drift esteja associado a uma instabilidade térmica presente durante o primeiro ensaio, possivelmente causada pelo aquecimento da amostra pelo agitador magnético. Variações de temperatura não compensadas adequadamente ou a dinâmica de um sistema em não-equilíbrio podem levar a tais desvios.

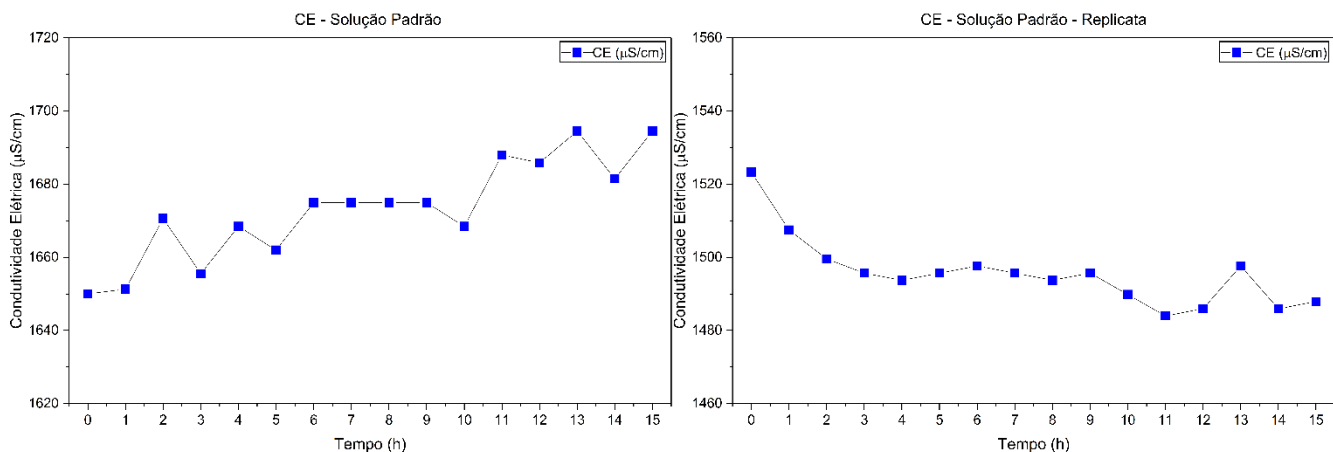


FIGURA 4. 5 - Medida e Replicata da Variação da Condutividade Elétrica em função do tempo para Solução Padrão 1413  $\mu\text{S}/\text{cm}$

Fonte: A autora.

Em suma, o sensor de condutividade mostrou-se capaz de rastrear com fidelidade tanto processos químicos dinâmicos (absorção de CO<sub>2</sub>) quanto condições de estado estacionário. As instabilidades observadas em um dos ensaios não são atribuídas a uma falha intrínseca do sensor, mas sim a variáveis experimentais (provavelmente térmicas) não totalmente controladas naquela ocasião específica, reforçando a necessidade de garantir o equilíbrio térmico para medições de condutividade de alta precisão.

### 4.1.3 Medição de pH

A performance do sensor de pH foi avaliada em dois cenários distintos para caracterizar sua estabilidade, precisão e resposta a fenômenos químicos. A calibração prévia do sensor revelou uma forte correlação linear ( $R^2=0,99234$ ), validando sua capacidade de fornecer medições quantitativas fidedignas (Figura 4.6).

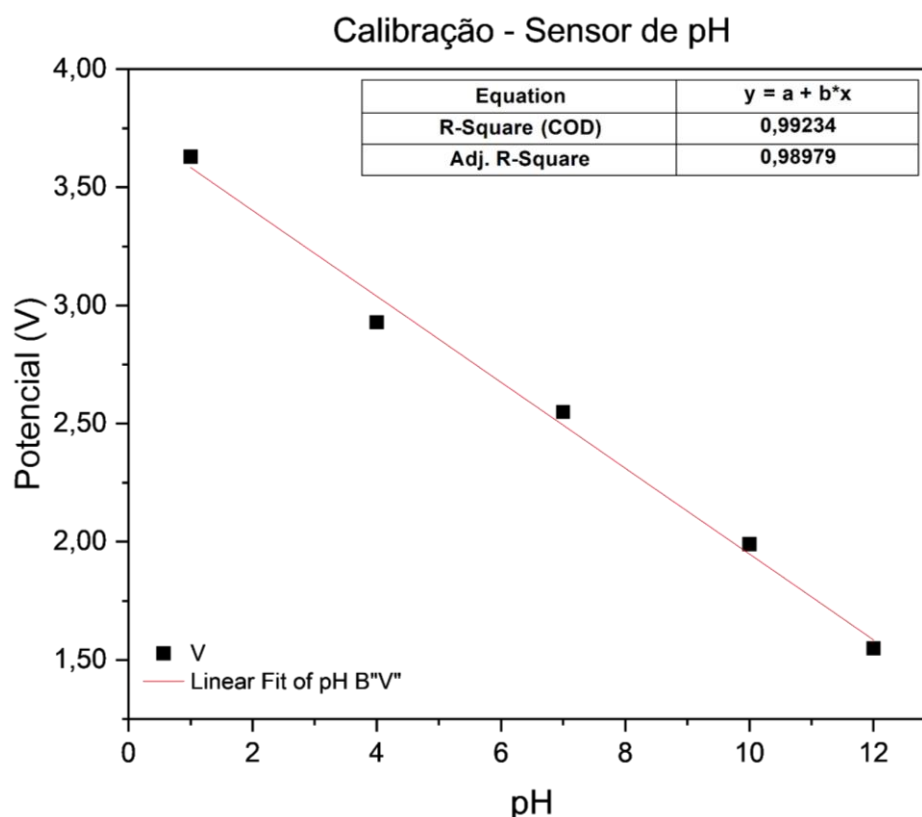


FIGURA 4. 6 - Relação entre o pH da Solução Padrão e o Potencial medido pelo sensor de pH  
Fonte: A autora.

A análise do sensor em solução tampão de pH 7 demonstrou uma estabilidade notável (Figura 4.7). Ao longo de 14 horas de monitoramento contínuo, as leituras de pH mantiveram-se constantes, com um desvio padrão mínimo e ausência de drift significativo. A estabilidade do sensor observada nos ensaios independentes, embora apresentem um pequeno desvio em relação ao valor nominal de 7,0, confirmam a alta precisão e a robustez do eletrodo em um meio de alta força iônica. Este comportamento esperado em uma solução tamponada serve como linha de base, atestando a estabilidade intrínseca do sistema de medição quando a química da amostra é controlada.

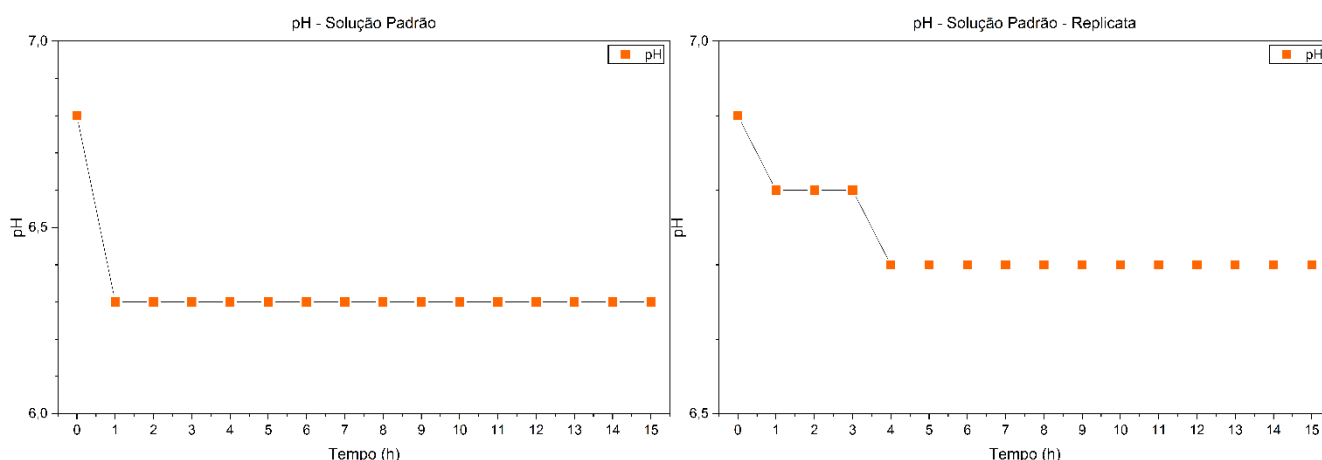


FIGURA 4. 7 - Medida e Replicata da Variação do pH em função do tempo para Solução Padrão pH 7  
Fonte: A autora.

Os ensaios realizados com água de alta pureza (Milli-Q), um meio não tamponado, revelaram a alta sensibilidade do sistema a interações com o ambiente. No primeiro ensaio, observou-se uma clara tendência de acidificação. O pH inicial, próximo a 8,5, decaiu ao longo das primeiras horas até atingir um estado de equilíbrio em torno de 6,5.

Este perfil é consistente com o processo de absorção de dióxido de carbono ( $\text{CO}_2$ ) atmosférico, que ao se dissolver na água forma ácido carbônico, acidificando o meio. O resultado não indica, portanto, uma instabilidade do sensor, mas sim sua capacidade de monitorar com precisão uma reação química fundamental que ocorre em águas não tamponadas expostas ao ar.

De forma contrastante, a réplica deste ensaio apresentou um

comportamento radicalmente diferente e oposto (Figura 4.8). O pH da amostra, partindo de um valor já elevado de 10, exibiu uma tendência de alcalinização contínua, alcançando aproximadamente 10,8 ao final do experimento. Tal comportamento anômalo exclui a absorção de CO<sub>2</sub> como fator dominante e aponta para a introdução de um contaminante alcalino na amostra.

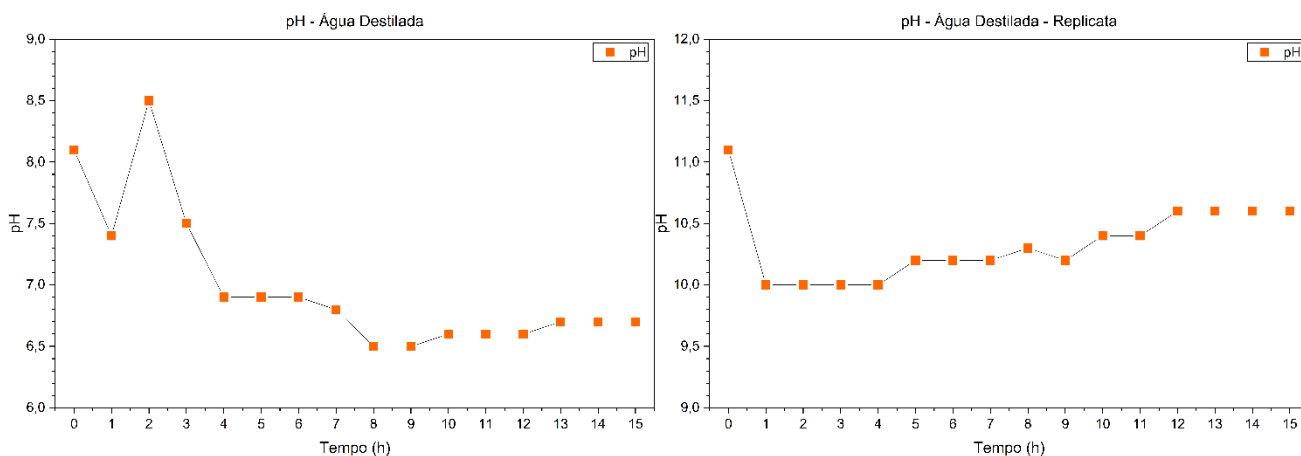


FIGURA 4. 8 - Medida e Replicata da Variação do pH em função do tempo para Água Destilada  
Fonte: A autora.

A hipótese mais plausível para esta alcalinização é a contaminação do ambiente laboratorial por aerossóis ou vapores de uma base forte, como o hidróxido de potássio (KOH), que era utilizado em outros experimentos no local. A água de alta pureza, por sua natureza não tamponada, é extremamente suscetível à absorção de tais contaminantes atmosféricos. A agitação contínua da amostra teria potencializado a absorção desses vapores alcalinos, resultando no aumento progressivo da concentração de íons hidroxila (OH<sup>-</sup>) e, conseqüentemente, na elevação do pH. A notável diferença entre os dois ensaios em água pura, realizados em dias distintos, evidencia a variabilidade das condições ambientais do laboratório e seu impacto direto em medições sensíveis.

Em suma, os resultados demonstram que o sensor de pH PH-4502C é estável e preciso em meios quimicamente estáveis. Mais importante, ele se mostrou uma ferramenta eficaz para monitorar reações químicas dinâmicas em tempo real. A divergência observada nos ensaios com água pura não representa uma falha do dispositivo, mas sim um registro autêntico de duas condições ambientais distintas: uma dominada pela acidificação por CO<sub>2</sub> atmosférico e outra

por uma contaminação anômala por vapores alcalinos. Este estudo de caso reforça que, em sistemas de baixa capacidade de tamponamento, o controle do ambiente é tão crucial quanto a calibração do instrumento para a correta interpretação dos resultados de pH.

#### **4.1.4 Medição de Turbidez**

A avaliação do sensor de turbidez foi conduzida para determinar sua curva de resposta, estabilidade temporal e comportamento em diferentes matrizes aquosas. Os resultados revelam um sistema complexo, cuja precisão é influenciada por múltiplos fatores, incluindo a natureza da amostra, a estabilidade eletrônica e as condições de iluminação ambiente.

A calibração do sensor com padrões de formazina confirmou que a relação entre a turbidez (NTU) e a voltagem de saída não é linear. Um modelo polinomial de segunda ordem ( $y = a + bx + cx^2$ ) demonstrou um ajuste excelente aos dados empíricos, com um coeficiente de determinação ( $R^2$ ) de 0,99847. Este achado é consistente com a teoria de operação de sensores nefelométricos de baixo custo, onde a relação entre a concentração de partículas e a luz espalhada pode desviar da linearidade, especialmente em faixas de medição mais amplas. A alta aderência do modelo polinomial valida o método de calibração e permite a conversão precisa dos sinais do sensor em unidades de turbidez. (Figura 4.9)

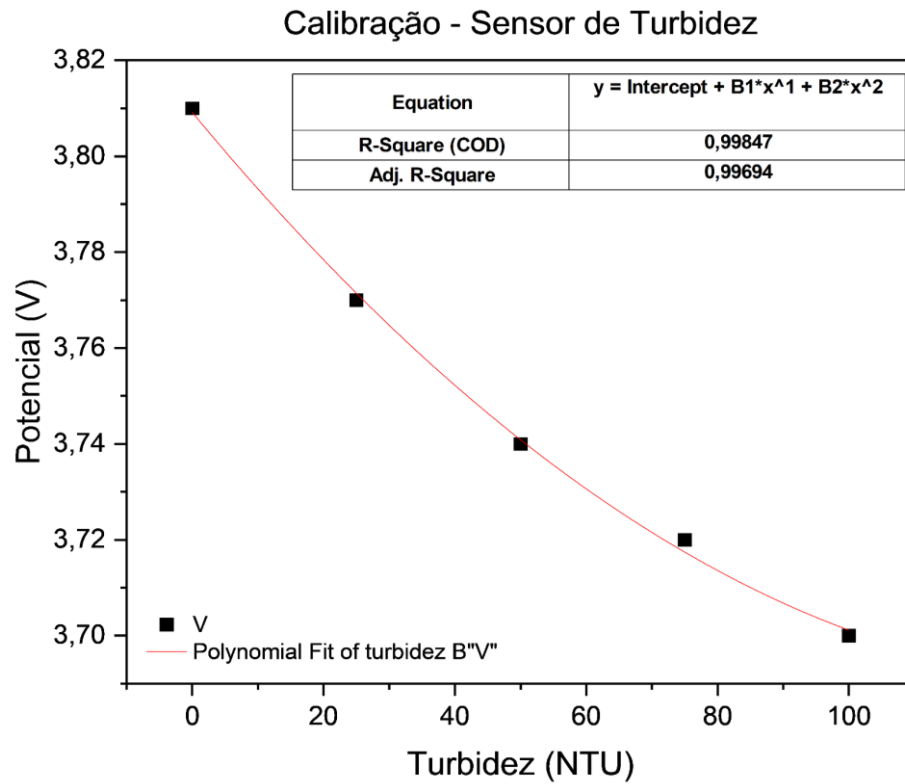


FIGURA 4. 9 - Relação entre a Turbidez da Solução Padrão e o Potencial medido pelo sensor de Turbidez  
 Fonte: A autora.

Os ensaios de estabilidade em solução padrão de 100 NTU apresentaram resultados díspares e revelaram a suscetibilidade do sensor a instabilidades. No primeiro ensaio o sensor registrou valores em torno de 125 NTU, exibindo uma estabilidade razoável, embora com um desvio positivo em relação ao valor nominal, em contrapartida, a replicata (Figura 4.10) exibiu uma instabilidade marcante.

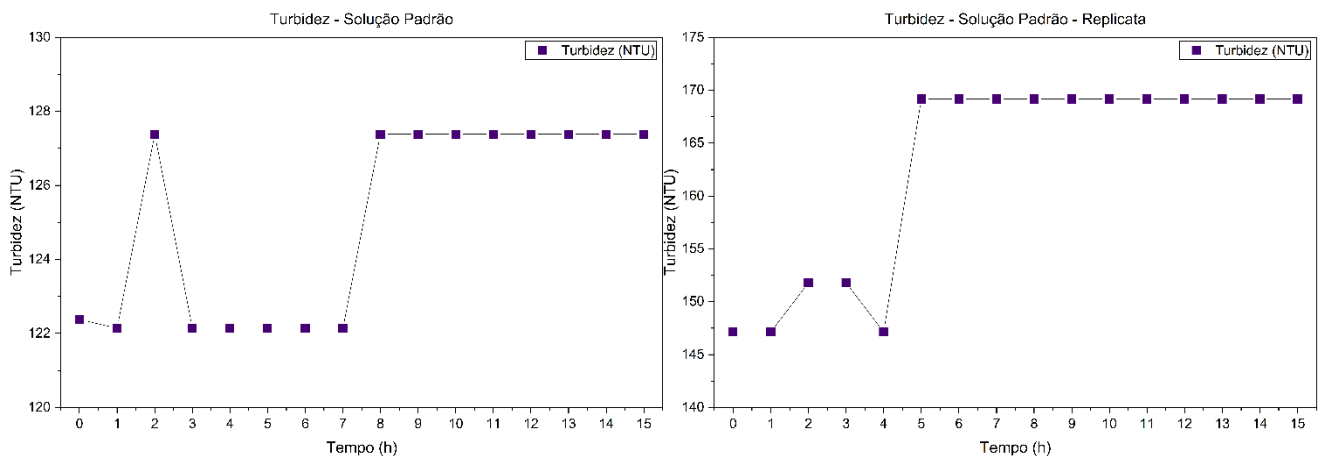


FIGURA 4. 10 - Medida e Replicata da Variação da Turbidez em função do tempo para Solução Padrão 100 NTU  
 Fonte: A autora.

O sinal apresentou uma grande dispersão e um comportamento em bandas distintas de medição, com valores oscilando entre aproximadamente 150 e 180 NTU. Este perfil, caracterizado por saltos abruptos no sinal, é um forte indicativo de problemas de natureza eletrônica, como mau contato nas conexões do sensor com a placa Arduino ou ruído na alimentação elétrica. Adicionalmente, a ausência de um controle sobre a iluminação ambiente durante os ensaios introduz uma variável de confusão significativa. Sensores ópticos como o de turbidez são intrinsecamente sensíveis à luz externa, que pode penetrar na câmara de medição e ser interpretada como luz espalhada, gerando leituras erráticas e artificialmente elevadas. A combinação de instabilidade eletrônica e variações na luz ambiente explica a baixa repetibilidade e a alta dispersão dos dados observada neste ensaio.

Os resultados mais inesperados foram obtidos durante os ensaios com água destilada, que, idealmente, deveria apresentar uma turbidez próxima de zero. No entanto, em ambos os ensaios, o sensor registrou valores de turbidez significativamente elevados e não nulos.

No primeiro ensaio a turbidez estabilizou-se em um patamar surpreendentemente alto, em torno de 230 NTU. Na replicata (Figura 4.11), embora o valor tenha sido menor, ainda se estabilizou em um platô considerável de aproximadamente 85 NTU, com uma dispersão de sinal que novamente sugere a presença de ruído eletrônico.

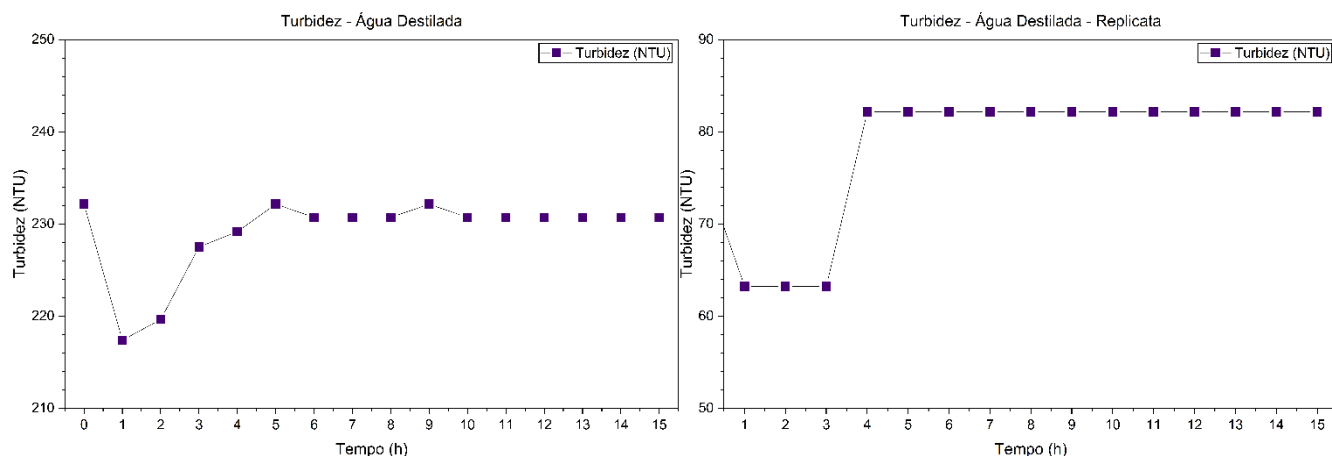


FIGURA 4. 11 - - Medida e Replicata da Variação da Turbidez em função do tempo para Água Destilada

Fonte: A autora.

A medição de uma turbidez elevada e estável em um meio sem partículas em suspensão é uma anomalia clara. Uma vez que a turbidez é uma medida óptica, este resultado pode ser explicado pela presença de microbolhas de ar na amostra. A agitação contínua da água, utilizada para homogeneização, pode ter induzido a cavitação ou a supersaturação do ar na água, gerando uma suspensão estável de microbolhas. Estas bolhas, com diâmetros na ordem de micrômetros, atuam como partículas em suspensão, espalhando a luz do emissor do sensor de forma similar às partículas de formazina. O sensor, incapaz de distinguir entre uma partícula sólida e uma bolha de gás, interpreta essa dispersão de luz como turbidez.

A diferença no nível do platô entre os dois ensaios (230 NTU vs. 85 NTU) pode ser atribuída a diferenças na intensidade da agitação, na temperatura da água ou na própria estabilidade eletrônica do sensor no dia do experimento, fatores que influenciam diretamente a formação e a densidade das microbolhas.

A avaliação do sensor de turbidez revelou que, embora possa ser calibrado para fornecer medições precisas sob condições ideais, sua aplicação prática é desafiadora. A performance do sensor é comprometida por uma combinação de fatores: **1) Instabilidade Eletrônica**, que se manifesta como ruído e saltos no sinal, exigindo conexões robustas e uma fonte de alimentação estável; **2) Sensibilidade à Luz Ambiente**, que demanda o isolamento óptico da câmara de

medição para evitar artefatos; e 3) Suscetibilidade a Interferentes Físicos, como microbolhas, que podem levar a leituras falsamente positivas em amostras agitadas. O comportamento anômalo em água pura destaca a importância de compreender as limitações do método de medição e de controlar as condições físicas da amostra, especialmente em aplicações de baixa turbidez.

## **4.2 Avaliação do Sistema de Processamento e Controle**

### **4.2.1 Desempenho da Plataforma Arduino**

A plataforma Arduino, especificamente o modelo utilizado (Arduino Uno R3, conforme mencionado anteriormente), foi selecionada devido à sua acessibilidade, vasta comunidade de suporte e capacidade de processamento adequada para as tarefas de aquisição e pré-processamento de dados dos sensores. O microcontrolador ATmega328P, que opera a 16 MHz, demonstrou ser suficiente para gerenciar as rotinas de leitura de múltiplos sensores (temperatura, pH, condutividade e turbidez) de forma sequencial e em intervalos definidos. A capacidade de processamento foi avaliada pela fluidez na execução do código e pela ausência de atrasos perceptíveis na aquisição de dados, mesmo quando todos os sensores estavam ativos simultaneamente. A arquitetura de 8 bits e a memória limitada (32 KB de Flash, 2 KB de SRAM, 1 KB de EEPROM) exigiram otimização do código para garantir a eficiência e evitar sobrecarga do processador.

### **Estabilidade do Sistema**

A estabilidade do sistema foi um fator crítico avaliado durante os experimentos de longa duração. Observou-se que a plataforma Arduino manteve um funcionamento contínuo e estável por períodos superiores a 15 horas, sem apresentar travamentos, reinicializações inesperadas ou corrupção de dados. A robustez do hardware e a simplicidade do design do circuito contribuíram para

essa estabilidade. Pequenas flutuações nas leituras dos sensores foram atribuídas principalmente a variações ambientais ou a fenômenos químicos inerentes às amostras, e não a instabilidades do sistema de processamento. A alimentação via porta USB do computador ou fonte externa regulada garantiu um fornecimento de energia consistente, minimizando ruídos elétricos que poderiam afetar a estabilidade.

#### **4.2.2 Eficiência do Firmware**

O firmware desenvolvido para o Arduino foi projetado para ser leve e eficiente, otimizando o uso dos recursos limitados do microcontrolador. A eficiência foi avaliada sob os seguintes aspectos:

##### **Tempo de Resposta**

O tempo de resposta do sistema, definido como o intervalo entre a solicitação de uma leitura e a disponibilização do dado processado, foi considerado adequado para aplicações de monitoramento contínuo. As rotinas de leitura de cada sensor foram implementadas de forma não bloqueante sempre que possível, permitindo que o microcontrolador realizasse outras tarefas enquanto aguardava a estabilização do sinal ou a conversão analógico-digital. O intervalo de leitura de 1 minuto, quando adotado, foi facilmente atendido pelo sistema, com os dados sendo transmitidos via serial de forma quase instantânea após a aquisição. Em testes com intervalos de leitura menores (e.g., 1 segundo), o sistema também demonstrou capacidade de resposta, embora com maior demanda de recursos.

##### **Consumo de Memória**

O consumo de memória (Flash para o código e SRAM para variáveis) foi monitorado durante o desenvolvimento para garantir que o firmware se encaixasse nas limitações do ATmega328P. Estratégias como a utilização de tipos

de dados eficientes, a minimização de variáveis globais e a otimização de bibliotecas foram empregadas. O código final ocupou uma fração razoável da memória Flash disponível, deixando espaço para futuras expansões. O consumo de SRAM, crucial para a execução em tempo real, foi mantido em níveis baixos, evitando estouros de pilha ou lentidão no processamento.

### **Robustez contra Falhas**

A robustez do firmware contra falhas foi testada implicitamente durante os longos períodos de operação. O código foi estruturado para lidar com condições anômalas, como leituras de sensores fora da faixa esperada ou interrupções na comunicação serial. Embora não tenham sido implementados mecanismos complexos de tratamento de erros (e.g., watchdog timers em todas as rotinas), a simplicidade do código e a ausência de operações complexas de alocação dinâmica de memória contribuíram para a sua estabilidade. Em caso de falhas de comunicação ou leituras errôneas, o sistema foi projetado para continuar operando, priorizando a coleta de dados e minimizando a interrupção do monitoramento.

### **4.2.3 Integração do Sensores**

A integração de múltiplos sensores em uma única plataforma exigiu considerações cuidadosas para garantir a compatibilidade e minimizar interferências. A abordagem modular, onde cada sensor foi inicialmente caracterizado individualmente, facilitou a identificação de potenciais problemas antes da integração completa.

#### **4.2.3.1 Análise de Interferência entre Sensores**

Durante a operação simultânea dos sensores, foram realizadas análises para identificar possíveis interferências elétricas ou físicas. A proximidade dos sensores na protoboard e a utilização de uma única fonte de alimentação para o Arduino e os módulos dos sensores poderiam, em tese, gerar

ruídos. No entanto, não foram observadas interferências significativas que comprometessem a qualidade dos dados. A utilização de pinos digitais e analógicos distintos para cada sensor, juntamente com a implementação de pequenos atrasos entre as leituras, ajudou a isolar os sinais. A única interferência notável foi a já discutida influência do agitador magnético na temperatura, que não é uma interferência eletrônica entre sensores, mas sim uma questão de controle experimental, a influência dos agitadores magnéticos na reprodutibilidade dos estudos é algo que vem sendo discutido na literatura como mostra (Cherepanova; Gordeev; Ananikov, 2025).

#### **4.2.3.2 Eficácia das Estratégias de Integração**

As estratégias de integração adotadas, baseadas em uma arquitetura centralizada com o Arduino como hub de dados, mostraram-se eficazes. A comunicação serial padronizada e a estrutura de dados consistente permitiram a fácil adição de novos sensores e a escalabilidade do sistema. A capacidade de pré-processar os dados no próprio microcontrolador (e.g., correção de temperatura para condutividade) reduziu a carga de processamento no computador host e simplificou a análise posterior. A integração física dos sensores em um arranjo compacto também demonstrou a viabilidade de uma solução miniaturizada para monitoramento em campo.

#### **4.2.3.3 Desafios Superados e Limitações Remanescentes**

##### **Desafios Superados:**

- **Compatibilidade de Hardware:** A seleção de sensores comerciais com interfaces compatíveis com o Arduino (analógicas e digitais) simplificou a conexão física e a programação.
- **Otimização de Código:** A necessidade de otimizar o firmware para operar dentro das limitações de memória e processamento do Arduino foi superada através de práticas de programação eficientes.

- **Sincronização de Dados:** A implementação de um sistema de timestamping e a leitura sequencial dos sensores garantiram a sincronização dos dados, crucial para análises correlacionadas.

#### **Limitações Remanescentes:**

- **Ausência de Calibração Automática:** O sistema atual requer calibração manual dos sensores, o que pode ser um desafio em aplicações de longo prazo sem intervenção humana.
- **Dependência de Alimentação Externa:** A plataforma ainda depende de uma fonte de alimentação externa (USB ou bateria), limitando sua autonomia em locais remotos sem infraestrutura.
- **Vulnerabilidade a Ruído:** Embora não tenham sido observadas interferências significativas, sistemas em ambientes mais ruidosos podem exigir blindagem adicional ou filtros de sinal.
- **Falta de Interface Gráfica Embarcada:** A ausência de uma interface gráfica no próprio dispositivo limita a visualização em tempo real dos dados sem a conexão a um computador.

### **4.3 Desempenho da Transmissão e Armazenamento de Dados**

#### **4.3.1 Sistema de Comunicação**

O sistema de comunicação implementado na plataforma baseia-se na interface serial USB entre o Arduino e o computador host, utilizando o protocolo UART (Universal Asynchronous Receiver-Transmitter) a 9600 bps. Esta escolha foi motivada pela simplicidade de implementação, confiabilidade e compatibilidade universal com diferentes sistemas operacionais.

#### **Taxa de Sucesso na Transmissão**

Durante os experimentos realizados, a taxa de sucesso na transmissão de dados foi consistentemente alta, aproximando-se de 100% em condições normais de operação. A comunicação serial via USB demonstrou ser

extremamente confiável, com raros casos de perda de dados ou corrupção de pacotes. Os poucos casos de falha na transmissão foram atribuídos a desconexões físicas acidentais do cabo USB ou a interrupções momentâneas na alimentação do sistema. A estrutura simples dos dados transmitidos (valores numéricos separados por delimitadores) contribuiu para a robustez da comunicação, minimizando a probabilidade de erros de interpretação.

A implementação de um formato de dados padronizado, com identificadores únicos para cada sensor e timestamps relativos, facilitou a detecção de eventuais perdas de dados durante a análise posterior. Em nenhum dos experimentos de longa duração (superiores a 15 horas) foram observadas falhas sistemáticas na transmissão que compromettesse a integridade do conjunto de dados coletados.

### **Alcance e Confiabilidade**

O alcance do sistema de comunicação é limitado pelo comprimento do cabo USB utilizado, tipicamente entre 1 a 5 metros para cabos padrão. Esta atual limitação física restringe a aplicação da plataforma a cenários onde o computador host pode ser posicionado próximo ao local de monitoramento. Para aplicações em campo que exigem maior alcance, seria necessária a implementação de sistemas de comunicação sem fio (e.g., Wi-Fi, Bluetooth, LoRa).

A confiabilidade da comunicação serial USB é amplamente reconhecida e foi confirmada durante os testes. A interface USB fornece não apenas o canal de dados, mas também a alimentação para o Arduino, simplificando a infraestrutura necessária. A ausência de interferências eletromagnéticas significativas no ambiente de laboratório contribuiu para a estabilidade da comunicação. Em ambientes industriais ou com maior ruído eletromagnético, poderiam ser necessárias medidas adicionais de blindagem ou protocolos de comunicação mais robustos.

## **Consumo Energético durante Transmissão**

O consumo energético durante a transmissão de dados é relativamente baixo, uma característica inerente à comunicação serial de baixa velocidade. O Arduino Uno R3, quando alimentado via USB, consome aproximadamente 50-100 mA durante a operação normal, incluindo a transmissão de dados. Este consumo é dominado pelos circuitos de processamento e pelos sensores ativos, sendo a contribuição específica da transmissão serial uma fração mínima do total.

A taxa de transmissão de 9600 bps é suficiente para a aplicação, considerando que os dados são transmitidos em intervalos de 1 minuto e consistem em valores numéricos relativamente simples. A baixa frequência de transmissão contribui para a eficiência energética do sistema, tornando-o adequado para aplicações alimentadas por bateria, embora a autonomia seja limitada pela capacidade da bateria e pelo consumo dos sensores.

### **4.3.2 Armazenamento Local**

O armazenamento local de dados na plataforma é realizado principalmente no computador host, utilizando o software TeraTerm para captura e gravação dos dados transmitidos via serial. Esta abordagem centralizada simplifica o gerenciamento de dados e permite o uso de sistemas de armazenamento com maior capacidade.

#### **Capacidade e Eficiência**

A capacidade de armazenamento é virtualmente ilimitada, dependendo apenas do espaço disponível no disco rígido do computador host. Durante os experimentos, os arquivos de dados gerados foram relativamente pequenos, tipicamente alguns megabytes para sessões de 15 horas, devido à natureza simples dos dados (valores numéricos de sensores) e à frequência de amostragem moderada (1 minuto).

A eficiência do armazenamento é alta, uma vez que os dados são

salvos em formato de texto simples (.txt), sem compressão ou processamento adicional. Este formato, embora não seja o mais compacto possível, oferece vantagens em termos de legibilidade humana, facilidade de processamento posterior e compatibilidade universal com diferentes softwares de análise.

O processo de conversão dos arquivos .txt para .csv, realizado posteriormente para facilitar a análise no software Origin, demonstrou ser eficiente e sem perda de dados. A estrutura simples dos dados facilitou esta conversão, que pôde ser automatizada através de scripts simples.

### **Integridade dos Dados**

A integridade dos dados foi mantida de forma consistente ao longo de todos os experimentos. O formato de texto simples utilizado é menos suscetível à corrupção em comparação com formatos binários complexos. A ausência de compressão ou criptografia elimina potenciais pontos de falha que poderiam comprometer a integridade dos dados.

Estratégias implícitas de verificação de integridade incluem a análise da continuidade temporal dos dados (detecção de lacunas nos timestamps) e a verificação da consistência dos valores dos sensores (detecção de valores anômalos que poderiam indicar corrupção). Durante a análise dos dados coletados, não foram identificadas evidências de corrupção ou perda de integridade.

A redundância natural do sistema, onde múltiplos sensores operam simultaneamente, fornece uma forma adicional de verificação cruzada. Inconsistências entre sensores relacionados (e.g., temperatura e condutividade) poderiam indicar problemas de integridade, mas tais inconsistências não foram observadas de forma sistemática.

### **Limitações do Sistema de Armazenamento:**

Embora o sistema de armazenamento tenha demonstrado eficácia para os propósitos do estudo, algumas limitações foram identificadas:

- **Dependência de Computador Host:** A necessidade de um computador dedicado para armazenamento limita a portabilidade e autonomia do sistema.
- **Ausência de Backup Automático:** O sistema atual não implementa mecanismos automáticos de backup, dependendo de procedimentos manuais para garantir a segurança dos dados.
- **Falta de Armazenamento Local no Arduino:** A capacidade limitada de armazenamento do Arduino (1KB de EEPROM) impede o armazenamento local significativo de dados, tornando o sistema dependente da comunicação contínua com o host.

### **Oportunidades de Melhoria:**

Para futuras versões da plataforma, algumas melhorias no sistema de armazenamento poderiam ser implementadas:

- **Adição de Cartão SD:** A integração de um módulo de cartão SD ao Arduino permitiria armazenamento local independente, aumentando a autonomia do sistema.
- **Implementação de Protocolos de Backup:** Sistemas automáticos de backup e sincronização poderiam aumentar a segurança dos dados coletados.
- **Compressão de Dados:** Para aplicações com grandes volumes de dados, algoritmos de compressão poderiam ser implementados para otimizar o uso do espaço de armazenamento.

## **4.4 Análise Comparativa e Discussão Geral**

### **4.4.1 Comparação com Sistemas Comerciais**

A plataforma de monitoramento hídrico desenvolvida neste trabalho, baseada na integração de sensores comerciais de baixo custo com uma plataforma Arduino, apresenta vantagens e limitações distintas quando comparada a sistemas comerciais de monitoramento de qualidade da água. Os sistemas comerciais,

frequentemente, oferecem alta precisão, robustez e certificações para uso em campo, mas geralmente a um custo elevado e com menor flexibilidade para personalização.

## **Vantagens e Limitações da Plataforma Construída**

### **Vantagens:**

- **Custo-benefício:** A principal vantagem da plataforma desenvolvida é o seu baixo custo de implementação. A utilização de sensores comerciais de prato e uma plataforma de hardware aberto como o Arduino reduz significativamente os gastos em comparação com sistemas comerciais equivalentes, tornando a tecnologia mais acessível para instituições de pesquisa com orçamentos limitados, pequenas comunidades ou projetos de monitoramento cidadão.
- **Flexibilidade e Personalização:** A natureza de hardware e software abertos do Arduino e a modularidade dos sensores permitem uma alta flexibilidade. A plataforma pode ser facilmente adaptada para monitorar diferentes parâmetros, integrar novos sensores ou modificar algoritmos de processamento de dados, sem a necessidade de licenças de software proprietárias ou hardware específico.
- **Manutenção Simplificada:** A disponibilidade de componentes de reposição no mercado e a facilidade de programação do Arduino simplificam a manutenção e o reparo do sistema, reduzindo a dependência de fornecedores específicos.

### **Limitações:**

- **Precisão e Exatidão:** Embora os sensores comerciais utilizados tenham demonstrado desempenho satisfatório em ambiente de laboratório, sua precisão e exatidão podem ser inferiores às de equipamentos de nível

laboratorial ou industrial. A ausência de calibração multiponto formal para alguns sensores e a falta de comparação com equipamentos de referência para turbidez e condutividade em campo são limitações que podem afetar a confiabilidade dos dados em aplicações mais rigorosas.

- **Robustez em Campo:** A plataforma, em sua configuração atual, não foi projetada para resistir a condições ambientais adversas (*e.g.*, imersão prolongada, variações extremas de temperatura, umidade, vibração) sem proteção adicional. A miniaturização e o encapsulamento adequado seriam necessários para uso em ambientes externos.
- **Autonomia Energética:** A dependência de uma fonte de alimentação externa (USB ou bateria de curta duração) limita a autonomia da plataforma em locais remotos sem infraestrutura elétrica.
- **Interface de Usuário:** A ausência de uma interface gráfica embarcada e a dependência de um computador para visualização e armazenamento de dados podem dificultar o uso por usuários não técnicos em campo.

### **Análise de Custo-Benefício**

A análise de custo-benefício da plataforma desenvolvida é favorável para aplicações onde o orçamento é uma restrição significativa e a flexibilidade é uma prioridade. O investimento inicial em hardware é consideravelmente menor do que o de sistemas comerciais, e os custos de desenvolvimento de software são minimizados pela utilização de plataformas de código aberto. Embora a precisão possa ser ligeiramente inferior à de equipamentos de ponta, a capacidade da plataforma de fornecer dados de tendência e identificar anomalias em tempo real oferece um benefício substancial para o monitoramento contínuo de baixo custo.

### **4.4.2 Potenciais Aplicações**

A plataforma de monitoramento hídrico desenvolvida possui um vasto potencial de aplicação em diversos cenários, dada a sua flexibilidade e baixo custo. Os principais cenários de uso incluem:

- **Monitoramento Ambiental:** Ideal para o monitoramento contínuo de rios, lagos, reservatórios e efluentes industriais, permitindo a detecção precoce de poluição ou variações na qualidade da água. Pode ser utilizada em projetos de pesquisa para coletar grandes volumes de dados ao longo do tempo.
- **Agricultura e Aquicultura:** No setor agrícola, a plataforma pode auxiliar no monitoramento da qualidade da água para irrigação, otimizando o uso de recursos hídricos e garantindo a saúde das culturas. Na aquicultura, é fundamental para manter as condições ideais da água em tanques de criação de peixes e camarões, prevenindo doenças e perdas.
- **Educação e Pesquisa:** Devido à sua natureza de hardware e software abertos, a plataforma é uma excelente ferramenta para fins educacionais, permitindo que estudantes e pesquisadores desenvolvam e testem suas próprias soluções de monitoramento. Pode ser utilizada em laboratórios didáticos para demonstrações práticas de princípios de sensoriamento e aquisição de dados.
- **Monitoramento Comunitário/Cidadão:** O baixo custo e a facilidade de uso tornam a plataforma acessível para comunidades que desejam monitorar a qualidade da água em suas regiões, promovendo a conscientização ambiental e a participação cidadã na gestão dos recursos hídricos.
- **Sistemas de Alerta Precoce:** A capacidade de monitoramento contínuo e a transmissão de dados em tempo real permitem a implementação de sistemas de alerta precoce para eventos de poluição ou variações abruptas na qualidade da água, possibilitando ações corretivas rápidas.

### **Adaptabilidade a Diferentes Contextos**

A modularidade da plataforma e a flexibilidade do firmware permitem sua adaptação a uma ampla gama de contextos. A adição ou substituição de sensores pode ser feita com relativa facilidade, permitindo que a plataforma

seja configurada para atender a requisitos específicos de diferentes aplicações. Por exemplo, a integração de sensores de oxigênio dissolvido, carga orgânica ou outros parâmetros relevantes pode expandir o escopo de monitoramento. A possibilidade de ajustar os intervalos de amostragem e os métodos de transmissão de dados (e.g., para comunicação sem fio) aumenta ainda mais sua versatilidade.

### **4.4.3 Limitações Identificadas e Oportunidades de Melhorias**

Durante o desenvolvimento e a avaliação da plataforma, diversas limitações foram identificadas, as quais representam oportunidades valiosas para desenvolvimentos futuros. Abordar essas limitações pode aprimorar significativamente o desempenho, a robustez e a aplicabilidade da plataforma.

#### **Desafios Técnicos Não Completamente Resolvidos**

**Calibração Automática:** A necessidade de calibração manual dos sensores é um desafio para a operação autônoma de longo prazo. A implementação de rotinas de calibração automática, talvez utilizando soluções padrão embarcadas ou algoritmos de auto-calibração, seria um avanço significativo.

**Compensação de Temperatura:** Embora a correção de temperatura tenha sido aplicada para alguns sensores (e.g., condutividade), a otimização e a generalização desses algoritmos para todos os parâmetros, considerando as especificidades de cada sensor, ainda representam um desafio.

**Miniaturização e Encapsulamento:** A plataforma, em sua forma atual, não é totalmente miniaturizada ou encapsulada para uso em campo. O desenvolvimento de um invólucro robusto, à prova d'água e com design otimizado para imersão, é crucial para aplicações externas.

**Autonomia Energética:** A dependência de alimentação externa limita a implantação em locais remotos. A integração de sistemas de gerenciamento de energia, como painéis solares ou baterias de alta capacidade com modos de baixo consumo, é essencial para aumentar a autonomia.

**Interferência da Luz Ambiente (Turbidez):** Conforme discutido na seção 4.1.4, a variação da luz ambiente pode influenciar as leituras do sensor de turbidez. Soluções como blindagem óptica ou algoritmos de compensação de luz ambiente precisam ser exploradas.

### **Sugestões para Desenvolvimento Futuro**

**Comunicação Sem Fio:** A implementação de módulos de comunicação sem fio (e.g., LoRa, NB-IoT, Wi-Fi) permitiria o monitoramento remoto e a transmissão de dados para a nuvem, expandindo o alcance e a acessibilidade da plataforma.

**Interface de Usuário Embarcada:** O desenvolvimento de uma interface de usuário simples (e.g., display LCD com botões) no próprio dispositivo permitiria a visualização em tempo real dos dados e a configuração de parâmetros sem a necessidade de um computador.

**Armazenamento de Dados Local:** A integração de um módulo de cartão SD ou memória flash no Arduino permitiria o armazenamento de dados localmente, garantindo a integridade dos dados mesmo em caso de falha na comunicação com o computador host.

**Análise de Dados e Visualização:** O desenvolvimento de um software de análise de dados mais robusto, com funcionalidades de visualização gráfica e relatórios automatizados, facilitaria a interpretação dos resultados.

**Integração com Plataformas IoT:** A conexão da plataforma a plataformas de Internet das Coisas (IoT) permitiria a integração com outros sistemas de monitoramento, análise de dados em larga escala e acesso remoto via dashboards online.

**Testes de Campo:** A validação da plataforma em ambientes reais, sob condições variadas de temperatura, umidade e interferências, é crucial para avaliar sua robustez e desempenho em aplicações práticas.

**Desenvolvimento de Sensores Miniaturizados:** A pesquisa e o desenvolvimento de novos sensores miniaturizados, como o sensor de pH de IrO<sub>2</sub>,

que já vem sendo desenvolvido e testado (Figura 4.12).

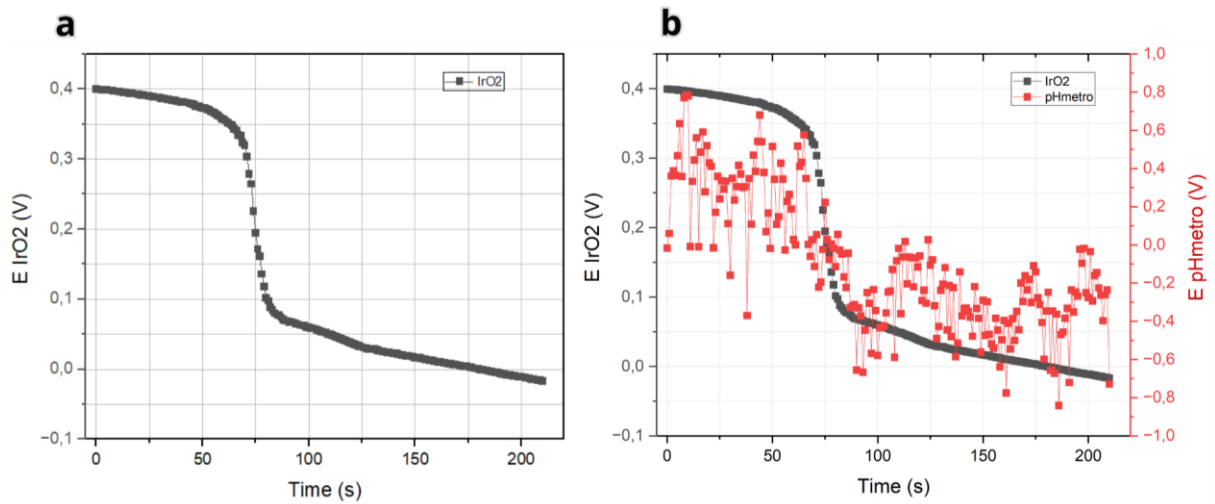


FIGURA 4. 12 - Titulação de HCl (0,1 M) com NaOH (0,5 M)  
Fonte: A autora.

Ou o sensor de condutividade miniaturizado que apresenta uma resposta linear em estudos preliminares (Figura 4.13) podem levar a soluções mais compactas e eficientes para o monitoramento hídrico.

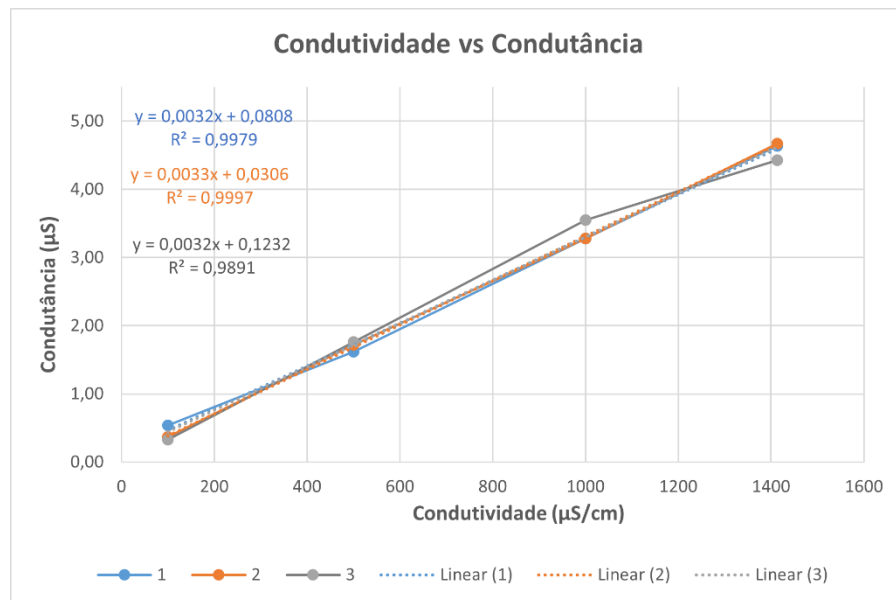


FIGURA 4. 13 - Relação entre condutividade de soluções padrão e a condutância (µS) obtida pelo sensor miniaturizado de condutividade (µS/cm)  
Fonte: A autora.

## 5 CONCLUSÃO

Esta dissertação descreveu o desenvolvimento de uma plataforma de baixo custo para o monitoramento da qualidade da água, integrando sensores comerciais de temperatura, condutividade elétrica, pH e turbidez com o microcontrolador Arduino UNO R3. O objetivo principal foi criar uma ferramenta acessível e eficiente para a coleta de dados ambientais, contribuindo para a gestão de recursos hídricos e a detecção precoce de anomalias.

O sistema demonstrou a viabilidade da utilização de componentes de baixo custo para a construção de um dispositivo de monitoramento multifuncional. A abordagem modular do firmware permitiu uma integração sequencial e robusta dos sensores, com funcionalidades de calibração interativa e persistência de dados via EEPROM. A análise dos códigos desenvolvidos e a descrição detalhada da metodologia de montagem e programação fornecem um guia para a replicação e futuras expansões do sistema.

Embora a fase de validação e calibração detalhada dos sensores ainda esteja em andamento, os resultados preliminares indicam o potencial da plataforma para aplicações em campo. A capacidade de coletar dados de múltiplos parâmetros simultaneamente e a flexibilidade para adaptação a diferentes cenários de comunicação e armazenamento de dados reforçam a relevância deste trabalho. Futuras pesquisas incluirão a validação extensiva em campo, a otimização do consumo de energia para operação autônoma e a implementação de módulos de comunicação sem fio para transmissão de dados em tempo real.

## 6. BIBLIOGRAFIA

1. ANA. **Conjuntura de Recursos Hídricos do Brasil: Informe Anual**. Agência Nacional de Águas e Saneamento Básico, , 2024.
2. ANA. **HidroSat - Monitoramento Hidrológico e de Qualidade de Água por Satélite**. Disponível em: <<https://app.powerbi.com/view?r=eyJrIjojN2FkODY5OTYtNmU1NC00N2JlWjZmQmMDdlMjEyMDdlZmVhliwidCI6ImUwYmI0MDEyLTgxMGItNDY5YS04YjRkLTY2N2ZjZDFiYWY4OCJ9>>. Acesso em: 22 jun. 2025.
3. ANA, Agência Nacional de Águas. **Qualidade da água**. governamental. Disponível em: <<https://www.gov.br/ana/pt-br/assuntos/monitoramento-e-eventos-criticos/qualidade-da-agua/qualidade-da-agua>>. Acesso em: 13 abr. 2023.
4. ANDERSON, Chauncey. Turbidity 6.7. *In: National Field Manual for the Collection of Water-Quality Data*. [S.l.: S.n.].
5. BATES, Roger G. Electrometric pH Determinations. **Soil Science**, v. 79, n. 3, p. 227, mar. 1955.
6. BENCS, László *et al.* Comparative Study on the Use of Some Low-Cost Optical Particulate Sensors for Rapid Assessment of Local Air Quality Changes. **Atmosphere**, v. 13, n. 8, p. 1218, ago. 2022.
7. BERGVELD, P. Thirty years of ISFETOLOGY. **Sensors and Actuators B: Chemical**, v. 88, n. 1, p. 1–20, jan. 2003.
8. BOGDAN, Razvan *et al.* Low-Cost Internet-of-Things Water-Quality Monitoring System for Rural Areas. **Sensors**, v. 23, n. 8, p. 3919, 12 abr. 2023.
9. BRAND, Oliver; POURKAMALI, Siavash. Electrothermal Excitation of Resonant MEMS. *In: Resonant MEMS*. [S.l.]: John Wiley & Sons, Ltd, 2015. p. 173–201.
10. CHEREPANOVA, Vera A.; GORDEEV, Evgeniy G.; ANANIKOV, Valentine P. Magnetic Stirring May Cause Irreproducible Results in Chemical Reactions. **JACS Au**, p. jacsau.5c00412, 11 jun. 2025.
11. CHILDS, Paul *et al.* Measurement of crack formation in concrete using embedded optical fibre sensors and differential strain analysis. **Measurement Science and Technology**, v. 19, n. 6, p. 065301, 1 jun. 2008.
12. DE CAMARGO, Edson Tavares *et al.* Low-Cost Water Quality Sensors for IoT: A Systematic Review. **Sensors**, v. 23, n. 9, p. 4424, 30 abr. 2023.
13. DELAUNEY, L.; COMPÈRE, C.; LEHAITRE, M. Biofouling protection for marine environmental sensors. **Ocean Science**, v. 6, n. 2, p. 503–511, 18 maio 2010.
14. DOWNING, John. Twenty-five years with OBS sensors: The good, the bad, and the ugly. **Continental Shelf Research**, Special Issue in Honor of Richard W. Sternberg's Contributions to Marine Sedimentology. v. 26, n. 17, p. 2299–2318, 1 nov. 2006.
15. **Efeito Seebeck. Alutal Measure & Trust**, 2025. Disponível em: <<https://www.alutal.com.br/br/wiki/termopares/04-efeito-seebeck/>>. Acesso em: 12 ago. 2025
16. EMPINOTTI, Vanessa Lucena; BUDDS, Jessica; AVERSA, Marcelo. Governance and water security: The role of the water institutional framework in the 2013–15 water crisis in São Paulo, Brazil. **Geoforum**, v. 98, p. 46–54, jan. 2019.
17. FORHAD, H. M. *et al.* IoT based real-time water quality monitoring system in water treatment plants (WTPs). **Heliyon**, v. 10, n. 23, p. e40746, dez. 2024.
18. GLASGOW, Howard B. *et al.* Real-time remote monitoring of water quality: a review of current applications, and advancements in sensor, telemetry, and computing technologies. **Journal of Experimental Marine Biology and Ecology**, VOLUME 300 Special Issue. v. 300, n. 1, p. 409–448, 31 mar. 2004.
19. GONZÁLEZ-PÉREZ, María I. *et al.* A novel optical sensor system for the automatic

- classification of mosquitoes by genus and sex with high levels of accuracy. **Parasites & Vectors**, v. 15, n. 1, dez. 2022.
20. GOVERNO DE MINAS GERAIS. **Histórico do rompimento das barragens da Vale na Mina Córrego do Feijão**. Disponível em: <<https://www.mg.gov.br/probrumadinho/pagina/historico-do-rompimento-das-barragens-da-vale-na-mina-corrego-do-feijao>>. Acesso em: 16 jul. 2025.
21. GUPTA, Vrinda; PANDEY, Rajoo. An improved energy aware distributed unequal clustering protocol for heterogeneous wireless sensor networks. **Engineering Science and Technology, an International Journal**, v. 19, n. 2, p. 1050–1058, jun. 2016.
22. HAMEL, Perrine *et al.* Low-cost monitoring systems for urban water management: Lessons from the field. **Water Research X**, v. 22, p. 100212, jan. 2024.
23. HART, Jane K.; MARTINEZ, Kirk. Environmental Sensor Networks: A revolution in the earth system science? **Earth-Science Reviews**, v. 78, n. 3, p. 177–191, 1 out. 2006.
24. HAYASHI, Masaki. Temperature-Electrical Conductivity Relation of Water for Environmental Monitoring and Geophysical Data Inversion. **Environmental Monitoring and Assessment**, v. 96, n. 1–3, p. 119–128, ago. 2004.
25. HONG, Wong *et al.* Water Quality Monitoring with Arduino Based Sensors. **Environments**, v. 8, n. 1, p. 6, 14 jan. 2021.
26. IN-SITU. **Monitoramento Remoto de Qualidade da Água - In-Situ do Brasil**. Disponível em: <<https://in-situdobrasil.com.br/monitoramento-remoto-qualidade-agua>>. Acesso em: 20 jun. 2025.
27. KHATRI, Punit; GUPTA, Karunesh Kumar; GUPTA, Raj Kumar. Raspberry Pi-based smart sensing platform for drinking-water quality monitoring system: a Python framework approach. **Drinking Water Engineering and Science**, v. 12, n. 1, p. 31–37, 25 jun. 2019.
28. KHORASANI, Fereshteh; NAJI, Hamid. Energy efficient data aggregation in wireless sensor networks using neural networks. **International Journal of Sensor Networks**, v. 24, p. 26, 1 jan. 2017.
29. KOESTOER, R. A. *et al.* A simple method for calibration of temperature sensor DS18B20 waterproof in oil bath based on Arduino data acquisition system. *In: THE 10TH INTERNATIONAL MEETING OF ADVANCES IN THERMOFLUIDS (IMAT 2018): SMART CITY: ADVANCES IN THERMOFLUID TECHNOLOGY IN TROPICAL URBAN DEVELOPMENT. Anais...* Bali, Indonesia: 2019. Disponível em: <<https://pubs.aip.org/aip/acp/article/793222>>. Acesso em: 20 jun. 2025
30. KOROSTYNSKA, Olga; MASON, Alex; AL-SHAMMA, A. Flexible Microwave Sensors for Realtime Analysis of Water Contaminants. **Journal of Electromagnetic Waves and Applications**, v. 27, p. 2075–2089, 1 nov. 2013.
31. KUMAR, Vipin; ROY, Debesh R.; BANO, Amreen. First-Principles Calculations of SiBi Nanosheets as Sensors for Oxygen-Containing Gases. **ACS Applied Nano Materials**, v. 4, n. 3, p. 2440–2451, 26 mar. 2021.
32. LAKARD, Boris *et al.* Potentiometric miniaturized pH sensors based on polypyrrole films. **Sensors and Actuators B: Chemical**, v. 122, n. 1, p. 101–108, 8 mar. 2007.
33. LAMBROU, Theofanis *et al.* A Low-Cost Sensor Network for Real-Time Monitoring and Contamination Detection in Drinking Water Distribution Systems. **Sensors Journal, IEEE**, v. 14, p. 2765–2772, 1 ago. 2014.
34. LEHMANN, Moritz K. *et al.* GLORIA - A globally representative hyperspectral in situ dataset for optical sensing of water quality. **Scientific Data**, v. 10, n. 1, 16 fev. 2023.
35. MATZEU, Giusy; FLOREA, Larisa; DIAMOND, Dermot. Advances in wearable chemical sensor design for monitoring biological fluids. **Sensors and Actuators B: Chemical**, v. 211, p. 403–418, 1 maio 2015.
36. MAXIM INTEGRATED PRODUCTS, INC. **DS18B20 - Programmable Resolution 1-Wire Digital Thermometer.** , 2019.

37. MCCLESKEY, R. Blaine *et al.* Solute and geothermal flux monitoring using electrical conductivity in the Madison, Firehole, and Gibbon Rivers, Yellowstone National Park. **Applied Geochemistry**, v. 27, n. 12, p. 2370–2381, dez. 2012.
38. MICHALSKI, Marie-Caroline; BRIARD, Valérie; MICHEL, Françoise. Optical parameters of milk fat globules for laser light scattering measurements. **Le Lait**, v. 81, n. 6, p. 787–796, nov. 2001.
39. MONK, Simon. **Programação com Arduino: começando com sketches**. 2º ed. [S.l.]: Bookman, 2021.
40. OMAR, Ahmad Fairuz Bin; MATJAFRI, Mohd Zubir Bin. Turbidimeter design and analysis: a review on optical fiber sensors for the measurement of water turbidity. **Sensors (Basel, Switzerland)**, v. 9, n. 10, p. 8311–8335, 2009.
41. **OriginLab - Origin and OriginPro - Data Analysis and Graphing Software**. Disponível em: <<https://www.originlab.com/>>. Acesso em: 9 ago. 2025.
42. **Partnerships and cooperation for water | UN World Water Development Report 2023**. Disponível em: <<https://www.unesco.org/reports/wwdr/2023/en>>. Acesso em: 25 jun. 2025.
43. RADIOMETER ANALYTICAL. **Radiometer M231Pt-9 and M251Pt-9 Platinum Electrodes-D31M096**. , 2004.
44. SADAR, Michael J. Turbidity science. technical information seriesbooklet no. 11. **Hach Co. Loveland CO**, v. 7, n. 8, 1998.
45. SAYAHI, T. *et al.* Development of a calibration chamber to evaluate the performance of low-cost particulate matter sensors. **Environmental Pollution**, v. 255, p. 113131, 1 dez. 2019.
46. SCHULZ, Uh; COSTA, Pf. The effects of press and pulse disturbance by long and short-term pollution on the fish community in the Sinos River, RS, Brazil. **Brazilian Journal of Biology**, v. 75, n. 2 suppl, p. 36–44, maio 2015.
47. SIMPSON, Nicholas Philip; SHEARING, Clifford D.; DUPONT, Benoit. ‘Partial functional redundancy’: An expression of household level resilience in response to climate risk. **Climate Risk Management**, v. 28, p. 100216, 2020.
48. SNIS. **SNIS**. Disponível em: <<https://www.gov.br/cidades/pt-br/aceso-a-informacao/acoes-e-programas/saneamento/snis/snis-1>>. Acesso em: 22 jun. 2025.
49. **Tera Term - Terminal Emulator for Windows**. Disponível em: <<https://teratermproject.github.io/>>. Acesso em: 9 ago. 2025.
50. UNESCO. **Statistics | UN World Water Development Report**. Disponível em: <<https://www.unesco.org/reports/wwdr/en/2024/s>>. Acesso em: 22 jun. 2025.
51. WAGNER, Richard J. *et al.* Guidelines and standard procedures for continuous water-quality monitors: Station operation, record computation, and data reporting. **Techniques and Methods**, 2006.
52. WHITESIDES, George. Whitesides, G.M. The origins and the future of microfluidics. **Nature** 442, 368-373. **Nature**, v. 442, p. 368–73, 1 ago. 2006.
53. WILLNER, Marjorie R.; VIKESLAND, Peter J. Nanomaterial enabled sensors for environmental contaminants. **Journal of Nanobiotechnology**, v. 16, n. 1, p. 95, 22 nov. 2018.
54. WILSON, A. N. *et al.* Embedded Sensors, Communication Technologies, Computing Platforms and Machine Learning for UAVs: A Review. **IEEE Sensors Journal**, v. 22, n. 3, p. 1807–1826, fev. 2022.
55. WIRMAN, Rahmi Putri; WARDHANA, Indrawata; ISNAINI, Vandri Ahmad. Kajian Tingkat Akurasi Sensor pada Rancang Bangun Alat Ukur Total Dissolved Solids (TDS) dan Tingkat Kekeruhan Air. **Jurnal Fisika**, v. 9, n. 1, p. 37–46, 11 maio 2019.
56. WMO. **State of Global Water Resources 2023**. Disponível em: <<https://wmo.int/publication-series/state-of-global-water-resources-2023>>. Acesso em: 22 jun. 2025.

57. WURTS, William; DURBOROW, Robert. Interactions of pH, Carbon Dioxide, Alkalinity and Hardness in Fish Ponds. **Southern Regional Aquaculture Center Publication**, v. 464, 1 jan. 1992.
58. YANG, Meijian; WANG, Guiling. Heat stress to jeopardize crop production in the US Corn Belt based on downscaled CMIP5 projections. **Agricultural Systems**, v. 211, p. 103746, out. 2023.
59. YSI. **Aquaculture Sensors, Monitoring and Control Instrumentation with Data Management**. Disponível em: <<https://www.ysi.com/products/aquaculture-process-monitors-and-sensors>>. Acesso em: 22 jun. 2025.
60. ZHANG, Yuanyuan *et al.* Review of the field environmental sensing methods based on multi-sensor information fusion technology. **International Journal of Agricultural and Biological Engineering**, v. 17, n. 2, p. 1–13, 2024.
61. [ZIMER, Alexsandro M. \*et al.\* Needle-like IrO/Ag combined pH microelectrode. \*\*Electrochemistry Communications\*\*, v. 12, n. 12, p. 1703–1705, dez. 2010.](#)

## 7. APÊNDICE

### APÊNDICE A – Código Inicial do Sensor de Temperatura

```
// Bibliotecas
#include <OneWire.h>           // Inclui a biblioteca OneWire para
comunicação com dispositivos OneWire
#include <DallasTemperature.h> // Inclui a biblioteca DallasTemperature
para comunicação com sensores de temperatura Dallas

#define ONE_WIRE_BUS 2 // Pino do Sensor - Define o pino digital ao qual
o sensor de temperatura está conectado

// Define o sensor
OneWire oneWire(ONE_WIRE_BUS); // Cria um objeto OneWire para
gerenciar a comunicação no barramento OneWire
DallasTemperature sensors(&oneWire); // Cria um objeto DallasTemperature
para controlar o sensor de temperatura

void setup(void) {
  sensors.begin(); // Inicia os sensores - Inicializa a comunicação com o
sensor de temperatura
}
void loop(void) {
  sensors.requestTemperatures(); // Envia o comando para ler a
temperatura - Solicita ao sensor que faça uma leitura de temperatura

  float temperatureC = sensors.getTempCByIndex(0); // Obtém a temperatura
em Celsius - Lê a temperatura do sensor em graus Celsius

  // Aqui você pode adicionar o código para usar a temperatura lida,
  // como imprimir no Serial Monitor ou fazer alguma outra ação.
  // Por exemplo:
  Serial.begin(9600); // Inicializa a comunicação serial com uma taxa de
9600 bauds
  Serial.print("Temperatura: "); // Imprime o texto "Temperatura: " no
Serial Monitor
  Serial.print(temperatureC); // Imprime o valor da temperatura no
Serial Monitor
  Serial.println(" °C"); // Imprime a unidade de medida " °C" e
pula para a próxima linha no Serial Monitor

  delay(1000); // Aguarda 1 segundo antes de ler novamente - Pausa a
execução do programa por 1000 milissegundos (1 segundo)
}
```

## APÊNDICE B – Código Inicial do Sensor de TDS (CE)

```
#include <EEPROM.h>
#include "GravityTDS.h"

#define TdsSensorPin A0
GravityTDS gravityTds;

float temperature = 25,tdsValue = 0;

void setup()
{
  Serial.begin(9600);
  gravityTds.setPin(TdsSensorPin);
  gravityTds.setAref(5.0); //reference voltage on ADC, default 5.0V on
Arduino UNO
  gravityTds.setAdcRange(1024); //1024 for 10bit ADC;4096 for 12bit
ADC
  gravityTds.begin(); //initialization
}

void loop()
{
  //temperature = readTemperature(); //add your temperature sensor and
read it
  gravityTds.setTemperature(temperature); // set the temperature and
execute temperature compensation
  gravityTds.update(); //sample and calculate
  tdsValue = gravityTds.getTdsValue(); // then get the valunt
  Serial.print(A0);
  Serial.print(" ");
  Serial.print(tdsValue,0);
  Serial.println("ppm");
  delay(1000);
}
```

## APÊNDICE C – Código Inicial do Sensor de pH

```
#include <Wire.h>
float calibracao_ph7 = 0.0; // Tensão obtida em solução de calibração
pH 7
float calibracao_ph4 = 0.0; // Tensão obtida em solução de calibração
pH 4
float calibracao_ph10 = 0.0; // Tensão obtida em solução de calibração
pH 10

#define UTILIZAR_PH_10 false // Habilita calibração entre pH 7 e 10, \
                             caso contrário utiliza pH 7 e 4.

float m;
float b;
int buf[10];

void setup() {
  Serial.begin(9600);
  if (calibracao_ph7 == 0 && calibracao_ph4 == 0 && calibracao_ph10 == 0)
  {
    delay(500);
    Serial.println();
    Serial.println("Erro - Necessário colocar os valores de calibração no
código!");
    while (1)
      ;
  }

  if (!UTILIZAR_PH_10 && calibracao_ph4 == 0 && calibracao_ph10 != 0 &&
calibracao_ph7 != 0) {
    delay(500);
    Serial.println();
    Serial.println("Erro - Você não marcou a opção UTILIZAR_PH_10 como
true para utilizar pH 7 e 10.");
    while (1)
      ;
  }

  if (UTILIZAR_PH_10) {
    m = (7.0 - 10.0) / (calibracao_ph7 - calibracao_ph10);
    b = 10.0 - m * calibracao_ph10;
  } else {
    m = (4.0 - 7.0) / (calibracao_ph4 - calibracao_ph7);
    b = 7.0 - m * calibracao_ph7;
  }
}
```

```

void loop() {
  for (int i = 0; i < 10; i++) { // 11 amostras
    buf[i] = analogRead(A0); // Ler o sensor PH
    delay(10);
  }
  for (int i = 0; i < 9; i++) { // Ordena em ordem crescente
    for (int j = i + 1; j < 10; j++) {
      if (buf[i] > buf[j]) {
        int temp = buf[i];
        buf[i] = buf[j];
        buf[j] = temp;
      }
    }
  }
  int valorMedio = 0;
  for (int i = 2; i < 8; i++) { // Realiza o valor médio utilizando 6
amostras
    valorMedio += buf[i];
  }

  float tensao = (valorMedio * 5.0) / 1024.0 / 6; // Realiza a média e
transforma o valor analógico em volt
  float ph = m * tensao + b; // Converte para pH
  Serial.println(ph);

  delay(1000); // Aguarda para próxima leitura
}

```

## APÊNDICE D – Código Inicial do Sensor de Turbidez

```
// Define o pino de Leitura do Sensor
int SensorTurbidez = A0;
// Inicia as variáveis
int i;
float voltagem;
float NTU;
void setup() {
  // Inicia a comunicação serial
  Serial.begin(9600);
}
void loop() {
  // Inicia a leitura da voltagem em 0
  voltagem = 0;
  // Realiza a soma dos "i" valores de voltagem
  for (i = 0; i < 800; i++) {
    voltagem += ((float)analogRead(SensorTurbidez) / 1024) * 5;
  }
  // Realiza a média entre os valores lidos na função for acima
  voltagem = voltagem / 800;
  voltagem = ArredondarPara(voltagem, 2);
  // Se Voltagem menor que 2.5 fixa o valor de NTU
  if (voltagem < 2.5) {
    NTU = 3000;
  }
  else if (voltagem > 3.81) {
    NTU = 0;
    voltagem = 3.81;
  }
  // Senão calcula o valor de NTU através da fórmula
  else {
    NTU = -595.2380952383246 * square(voltagem) + 3779.761904763616 *
    voltagem - 5760.357142860333;
  }
  // Imprime as informações no monitor serial
  Serial.print("Leitura: ");
  Serial.print(voltagem);
  Serial.print(" V, NTU: ");
  Serial.println(NTU);
  delay(10);
}
// Sistema de arredondamento para Leitura
float ArredondarPara( float ValorEntrada, int CasaDecimal ) {
  float multiplicador = powf( 10.0f, CasaDecimal );
  ValorEntrada = roundf( ValorEntrada * multiplicador ) / multiplicador;
```

```
    return ValorEntrada;  
}
```

## APÊNDICE E – Código da Integração TDS e Temperatura

```
// Bibliotecas necessárias
#include <OneWire.h>
#include <DallasTemperature.h>

// Definição dos pinos
#define TDS_PIN A1           // Sensor TDS no pino analógico A1
#define TEMP_PIN 4          // Sensor de temperatura no pino digital 4

// Constantes de calibração
#define CAL_SOLUTION_EC 500 // Solução de calibração 1413 µS/cm (comum
para calibração)
#define TEMP_COEF 0.019     // Coeficiente de temperatura (1.9% por °C)

// Inicialização dos sensores
OneWire oneWire(TEMP_PIN);
DallasTemperature sensors(&oneWire);

// Variáveis globais
float calibrationFactor = 721.127; // Fator de calibração inicial

void setup() {
    Serial.begin(9600);           // Inicia comunicação serial
    sensors.begin();             // Inicia sensor de temperatura

    Serial.println("Sistema iniciado!");
    Serial.println("Digite 'c' para calibrar ou aguarde as leituras...");
}

void loop() {
    // Verifica se há comando de calibração
    if (Serial.available()) {
        if (Serial.read() == 'c') {
            calibrateSensor();
        }
    }

    // Realiza leitura normal a cada 2 segundos
    static unsigned long lastTime = 0;
    if (millis() - lastTime >= 2000) {
        readAndDisplayEC();
        lastTime = millis();
    }
}

void readAndDisplayEC() {
    // Lê temperatura
```

```

sensors.requestTemperatures();

float temperature = sensors.getTempCByIndex(0)*25.0/23.7; // 25/23.7
serve para corrigir a temperatura medida

// Lê valor analógico do TDS
float voltage = analogRead(TDS_PIN) * (5.0 / 1024.0);

// Calcula EC com compensação de temperatura
float ecValue = voltage * calibrationFactor;
float ecValue25 = ecValue / (1.0 + TEMP_COEF * (temperature - 25.0));

// Mostra resultados
Serial.print("Tensão: ");
Serial.print(voltage);
Serial.print("V ");
Serial.print("Temperatura: ");
Serial.print(temperature, 1);
Serial.print("°C   EC: ");
Serial.print(ecValue25, 0);
Serial.println(" µS/cm");
}

void calibrateSensor() {
  Serial.println("\nIniciando calibração...");
  Serial.println("Coloque o sensor na solução de 1413 µS/cm");
  Serial.println("Aguarde 10 segundos...");
  delay(10000); // Aguarda estabilização

  // Faz média de 10 leituras
  float sumVoltage = 0;
  for(int i = 0; i < 10; i++) {
    sumVoltage += analogRead(TDS_PIN) * (5.0 / 1024.0);
    delay(100);
  }
  float avgVoltage = sumVoltage / 10;

  // Calcula novo fator de calibração
  calibrationFactor = CAL_SOLUTION_EC / avgVoltage;

  Serial.println("Calibração concluída!");
  Serial.print("Novo fator de calibração: ");
  Serial.println(calibrationFactor, 3);
}

```

## APÊNDICE F – Código da Integração TDS, Temperatura e pH

```
#include <EEPROM.h>
#include "GravityTDS.h"
#include <OneWire.h>
#include <DallasTemperature.h>

#define TdsSensorPin A0 // azul-tds
#define ph_pin A1 // laranja-ph
#define PINO_ONEWIRE 2 // Define pino do sensor amarelo-temperatura

// Endereços EEPROM para calibração pH
#define ADDR_PH7 0x10
#define ADDR_PH4 0x14
#define ADDR_PH10 0x18

GravityTDS gravityTds;
OneWire oneWire(PINO_ONEWIRE);
DallasTemperature sensor(&oneWire);
DeviceAddress endereco_temp;

float tdsValue = 0;
float temperature = 25;
float EC;

// Variáveis pH
float calibracao_ph7 = 2.65;
float calibracao_ph4 = 2.92;
float calibracao_ph10 = 0.0;
#define UTILIZAR_PH_10 false
float m;
float b;
int buf[10];

// Variável para controle do modo de calibração
bool modoCalibracaoAtivo = false;

void setup()
{
    Serial.begin(9600);
    gravityTds.setPin(TdsSensorPin);
    gravityTds.setAref(5.0);
    gravityTds.setAdcRange(1024);
    gravityTds.begin();
    sensor.begin();

    // Carrega valores de calibração da EEPROM
    EEPROM.get(ADDR_PH7, calibracao_ph7);
```

```

EEPROM.get(ADDR_PH4, calibracao_ph4);
EEPROM.get(ADDR_PH10, calibracao_ph10);

calcularConstantesPH();

Serial.println("Sistema iniciado!");
Serial.println("Digite 'enter' para modo de calibração");
}

void calcularConstantesPH() {
    if (UTILIZAR_PH_10) {
        m = (7.0 - 10.0) / (calibracao_ph7 - calibracao_ph10);
        b = 10.0 - m * calibracao_ph10;
    } else {
        m = (4.0 - 7.0) / (calibracao_ph4 - calibracao_ph7);
        b = 7.0 - m * calibracao_ph7;
    }
}

float lerTensaoMedia() {
    long soma = 0;
    for(int i = 0; i < 10; i++) {
        soma += analogRead(ph_pin);
        delay(10);
    }
    float tensao = (soma * 5.0) / (1024.0 * 10.0);
    return tensao;
}

void processarComandos() {
    if (Serial.available()) {
        String cmd = Serial.readStringUntil('\n');
        cmd.trim();

        if (cmd == "enter") {
            modoCalibracaoAtivo = true;
            Serial.println("\n=== MODO DE CALIBRAÇÃO ===");
            Serial.println("Comandos disponíveis:");
            Serial.println("Para TDS:");
            Serial.println("  cal:707 -> Calibrar TDS (coloque em solução
707ppm)");
            Serial.println("Para pH:");
            Serial.println("  ph7 -> Calibrar com solução pH 7");
            Serial.println("  ph4 -> Calibrar com solução pH 4");
            Serial.println("  ph10 -> Calibrar com solução pH 10");
            Serial.println("exit -> Sair do modo de calibração");
            Serial.println("=====\n");

```

```

    }

    if (cmd.startsWith("cal:") && modoCalibracaoAtivo) {
        String valorStr = cmd.substring(4);
        float valor = valorStr.toFloat();
        if (valor > 0) {
            gravityTds.setTemperature(temperature); // Atualiza
temperatura
            gravityTds.update(); // Atualiza leituras
            Serial.println("Calibração TDS realizada!");
        }
    }

    if (cmd == "ph7" && modoCalibracaoAtivo) {
        Serial.println("calibrando pH");
        calibracao_ph7 = lerTensaoMedia();
        EEPROM.put(ADDR_PH7, calibracao_ph7);
        Serial.print("pH 7 calibrado. Tensão: ");
        Serial.println(calibracao_ph7);
        calcularConstantesPH();
    }

    if (cmd == "ph4" && modoCalibracaoAtivo) {
        Serial.println("calibrando pH");
        calibracao_ph4 = lerTensaoMedia();
        EEPROM.put(ADDR_PH4, calibracao_ph4);
        Serial.print("pH 4 calibrado. Tensão: ");
        Serial.println(calibracao_ph4);
        calcularConstantesPH();
    }

    if (cmd == "ph10" && modoCalibracaoAtivo) {
        Serial.println("calibrando pH");
        calibracao_ph10 = lerTensaoMedia();
        EEPROM.put(ADDR_PH10, calibracao_ph10);
        Serial.print("pH 10 calibrado. Tensão: ");
        Serial.println(calibracao_ph10);
        calcularConstantesPH();
    }

    if (cmd == "exit" && modoCalibracaoAtivo) {
        modoCalibracaoAtivo = false;
        Serial.println("Calibração finalizada");
        Serial.println("Valores salvos na EEPROM");
    }
}
}

```

```

void loop()
{
    processarComandos();

    // Leitura de temperatura
    sensor.requestTemperatures();
    if (!sensor.getAddress(endereco_temp,0)) {
        Serial.println("SENSOR NAO CONECTADO");
    }
    temperature = sensor.getTempC(endereco_temp);

    // Leitura TDS
    gravityTds.setTemperature(temperature);
    gravityTds.update();
    tdsValue = gravityTds.getTdsValue();
    EC = (tdsValue*2);

    // Leitura pH
    for (int i = 0; i < 10; i++) {
        buf[i] = analogRead(ph_pin);
        delay(10);
    }

    // Ordenação das leituras pH
    for (int i = 0; i < 9; i++) {
        for (int j = i + 1; j < 10; j++) {
            if (buf[i] > buf[j]) {
                int temp = buf[i];
                buf[i] = buf[j];
                buf[j] = temp;
            }
        }
    }

    // Cálculo pH
    int valorMedio = 0;
    for (int i = 2; i < 8; i++) {
        valorMedio += buf[i];
    }

    float tensao = (valorMedio * 5.0) / 1024.0 / 6;
    float ph = m * tensao + b;

    // Exibição dos resultados se não estiver em calibração
    if (!modoCalibracaoAtivo) {
        Serial.println("\n=== LEITURAS ATUAIS ===");
    }
}

```

```
    Serial.print("TDS: ");
    Serial.print(tdsValue,0);
    Serial.println(" ppm");
    Serial.print("EC: ");
    Serial.print(EC);
    Serial.print(" uS/cm");
    Serial.print(" em ");
    Serial.print(temperature, 1);
    Serial.println("°C");
    Serial.print("pH: ");
    Serial.println(ph, 1);
    Serial.println("=====\n");
}

delay(1000);
}
```

## APÊNDICE G – Código da Integração Completa

```
#include <EEPROM.h> // Inclui a biblioteca para leitura/escrita na EEPROM
(memória não volátil)
#include "GravityTDS.h" // Inclui a biblioteca para o sensor TDS da
Gravity
#include <OneWire.h> // Inclui a biblioteca para comunicação com sensores
OneWire (como o de temperatura)
#include <DallasTemperature.h> // Inclui a biblioteca para sensores de
temperatura Dallas
// Define os pinos dos sensores
#define SensorTurbidez A0 // Define o pino analógico A0 para o sensor de
turbidez (roxo)
#define ph_pin A1 // Define o pino analógico A1 para o sensor de pH
(laranja)
#define TdsSensorPin A2 // Define o pino analógico A2 para o sensor de
TDS (azul)
#define PINO_ONEWIRE 2 // Define o pino digital 2 para o sensor de
temperatura (amarelo)
// Endereços EEPROM para calibração pH
#define ADDR_PH7 0x10 // Define o endereço na EEPROM para o valor de
calibração do pH 7
#define ADDR_PH4 0x14 // Define o endereço na EEPROM para o valor de
calibração do pH 4
#define ADDR_PH10 0x18 // Define o endereço na EEPROM para o valor de
calibração do pH 10
GravityTDS gravityTds; // Cria um objeto para controlar o sensor TDS
OneWire oneWire(PINO_ONEWIRE); // Cria um objeto para comunicação OneWire
no pino definido
DallasTemperature sensor(&oneWire); // Cria um objeto para o sensor de
temperatura Dallas
DeviceAddress endereco_temp; // Variável para armazenar o endereço do
sensor de temperatura
float tdsValue = 0; // Variável para armazenar o valor de TDS
float temperature = 25; // Variável para armazenar a temperatura
(inicializada com 25)
float EC; // Variável para armazenar a condutividade elétrica
// Variáveis pH
float calibracao_ph7 = 2.65; // Valor de calibração para pH 7 (inicial)
float calibracao_ph4 = 2.92; // Valor de calibração para pH 4 (inicial)
float calibracao_ph10 = 0.0; // Valor de calibração para pH 10 (inicial)
#define UTILIZAR_PH_10 false // Define se a calibração com pH 10 será
utilizada
float m; // Variável para a inclinação da reta de calibração do pH
float b; // Variável para o ponto de intersecção da reta de calibração do
pH
int buf[10]; // Buffer para armazenar leituras do pH
```

```

// Variáveis Turbidez
int i; // Variável de loop
float voltagem; // Variável para armazenar a voltagem lida do sensor de
turbidez
float NTU; // Variável para armazenar o valor de turbidez em NTU
// Variável para controle do modo de calibração
bool modoCalibracaoAtivo = false; // Indica se o modo de calibração está
ativo
// Protótipo da função ArredondarPara
float ArredondarPara(float ValorEntrada, int CasaDecimal); // Declara a
função para arredondar valores
void setup() {
    Serial.begin(9600); // Inicializa a comunicação serial
    gravityTds.setPin(TdsSensorPin); // Define o pino do sensor TDS
    gravityTds.setAref(5.0); // Define a tensão de referência para 5V
    gravityTds.setAdcRange(1024); // Define a faixa do ADC para 1024
    gravityTds.begin(); // Inicializa o sensor TDS
    sensor.begin(); // Inicializa o sensor de temperatura
    // Carrega valores de calibração da EEPROM
    EEPROM.get(ADDR_PH7, calibracao_ph7); // Lê o valor de calibração do
pH 7 da EEPROM
    EEPROM.get(ADDR_PH4, calibracao_ph4); // Lê o valor de calibração do
pH 4 da EEPROM
    EEPROM.get(ADDR_PH10, calibracao_ph10); // Lê o valor de calibração
do pH 10 da EEPROM
    calcularConstantesPH(); // Calcula as constantes da calibração do pH
    Serial.println("Sistema iniciado!"); // Imprime mensagem de
inicialização
    Serial.println("Digite 'enter' para modo de calibração"); //
Instruções para entrar no modo de calibração
}
void calcularConstantesPH() {
    if (UTILIZAR_PH_10) { // Se estiver usando calibração com pH 10
        m = (7.0 - 10.0) / (calibracao_ph7 - calibracao_ph10); // Calcula
a inclinação usando pH 7 e pH 10
        b = 10.0 - m * calibracao_ph10; // Calcula o ponto de intersecção
usando pH 10
    } else { // Se não estiver usando calibração com pH 10
        m = (4.0 - 7.0) / (calibracao_ph4 - calibracao_ph7); // Calcula a
inclinação usando pH 4 e pH 7
        b = 7.0 - m * calibracao_ph7; // Calcula o ponto de intersecção
usando pH 7
    }
}
float lerTensaoMedia() {
    long soma = 0; // Variável para somar as leituras
    for(int i = 0; i < 10; i++) { // Loop para ler 10 amostras

```

```

        soma += analogRead(ph_pin); // Lê o valor analógico do pino do pH
e soma
        delay(10); // Pequeno delay entre as leituras
    }
    float tensao = (soma * 5.0) / (1024.0 * 10.0); // Calcula a tensão
média
    return tensao; // Retorna a tensão média
}

void processarComandos() {
    if (Serial.available()) { // Se houver dados disponíveis na Serial
        String cmd = Serial.readStringUntil('\n'); // Lê a string até
encontrar uma nova linha
        cmd.trim(); // Remove espaços em branco no início e fim da string

        if (cmd == "enter") { // Se o comando for "enter"
            modoCalibracaoAtivo = true; // Ativa o modo de calibração
            Serial.println("\n=== MODO DE CALIBRAÇÃO ==="); // Imprime
cabeçalho
            Serial.println("Comandos disponíveis:"); // Lista os comandos
disponíveis
            Serial.println("Para TDS:"); // Comandos para TDS
            Serial.println("  cal:707 -> Calibrar TDS (coloque em solução
707ppm)"); // Instrução para calibrar o TDS
            Serial.println("Para pH:"); // Comandos para pH
            Serial.println("  ph7 -> Calibrar com solução pH 7"); //
Instrução para calibrar com pH 7
            Serial.println("  ph4 -> Calibrar com solução pH 4"); //
Instrução para calibrar com pH 4
            Serial.println("  ph10 -> Calibrar com solução pH 10"); //
Instrução para calibrar com pH 10
            Serial.println("exit -> Sair do modo de calibração"); //
Instrução para sair do modo de calibração
            Serial.println("=====\n"); // Imprime
rodapé
        }
        if (cmd.startsWith("cal:") && modoCalibracaoAtivo) { // Se o
comando começar com "cal:" e o modo de calibração estiver ativo
            String valorStr = cmd.substring(4); // Extraí o valor após
"cal:"
            float valor = valorStr.toFloat(); // Converte o valor para
float

            if (valor > 0) { // Se o valor for maior que 0
                gravityTds.setTemperature(temperature); // Atualiza a
temperatura no sensor TDS
                gravityTds.update(); // Atualiza a leitura do sensor TDS
                Serial.println("Calibração TDS realizada!"); // Imprime

```

```

mensagem de sucesso
    }
}
    if (cmd == "ph7" && modoCalibracaoAtivo) { // Se o comando for
"ph7" e o modo de calibração estiver ativo
        Serial.println("calibrando pH"); // Imprime mensagem
        calibracao_ph7 = lerTensaoMedia(); // Lê a tensão média do
sensor de pH
        EEPROM.put(ADDR_PH7, calibracao_ph7); // Salva o valor na
EEPROM

        Serial.print("pH 7 calibrado. Tensão: "); // Imprime mensagem
        Serial.println(calibracao_ph7); // Imprime o valor da tensão
        calcularConstantesPH(); // Recalcula as constantes de
calibração
    }
    if (cmd == "ph4" && modoCalibracaoAtivo) { // Se o comando for
"ph4" e o modo de calibração estiver ativo
        Serial.println("calibrando pH"); // Imprime mensagem
        calibracao_ph4 = lerTensaoMedia(); // Lê a tensão média do
sensor de pH
        EEPROM.put(ADDR_PH4, calibracao_ph4); // Salva o valor na
EEPROM

        Serial.print("pH 4 calibrado. Tensão: "); // Imprime mensagem
        Serial.println(calibracao_ph4); // Imprime o valor da tensão
        calcularConstantesPH(); // Recalcula as constantes de
calibração
    }
    if (cmd == "ph10" && modoCalibracaoAtivo) { // Se o comando for
"ph10" e o modo de calibração estiver ativo
        Serial.println("calibrando pH"); // Imprime mensagem
        calibracao_ph10 = lerTensaoMedia(); // Lê a tensão média do
sensor de pH
        EEPROM.put(ADDR_PH10, calibracao_ph10); // Salva o valor na
EEPROM

        Serial.print("pH 10 calibrado. Tensão: "); // Imprime
mensagem

        Serial.println(calibracao_ph10); // Imprime o valor da tensão
        calcularConstantesPH(); // Recalcula as constantes de
calibração
    }
    if (cmd == "exit" && modoCalibracaoAtivo) { // Se o comando for
"exit" e o modo de calibração estiver ativo
        modoCalibracaoAtivo = false; // Desativa o modo de calibração
        Serial.println("Calibração finalizada"); // Imprime mensagem
        Serial.println("Valores salvos na EEPROM"); // Imprime
mensagem
    }
}

```

```

    }
}
void loop() {
    processarComandos(); // Processa os comandos recebidos pela Serial
    // Leitura de temperatura
    sensor.requestTemperatures(); // Solicita a leitura da temperatura
    if (!sensor.getAddress(endereco_temp,0)) { // Se não conseguir obter
o endereço do sensor
        Serial.println("SENSOR NAO CONECTADO"); // Imprime mensagem de
erro
    }
    temperature = sensor.getTempC(endereco_temp); // Obtém a temperatura
em graus Celsius
    // Leitura TDS
    gravityTds.setTemperature(temperature); // Define a temperatura para
compensação
    gravityTds.update(); // Atualiza a leitura do TDS
    tdsValue = gravityTds.getTdsValue(); // Obtém o valor de TDS
    EC = (tdsValue*2); // Calcula a condutividade elétrica
    // Leitura pH
    for (int i = 0; i < 10; i++) { // Loop para ler 10 amostras do pH
        buf[i] = analogRead(ph_pin); // Lê o valor analógico do pino do
pH
        delay(10); // Pequeno delay entre as leituras
    }
    // Ordenação das leituras pH
    for (int i = 0; i < 9; i++) { // Loop para ordenar o array de
leituras
        for (int j = i + 1; j < 10; j++) { // Loop interno para comparar
            if (buf[i] > buf[j]) { // Se o valor atual for maior que o
próximo
                int temp = buf[i]; // Troca os valores de posição
                buf[i] = buf[j];
                buf[j] = temp;
            }
        }
    }
    // Cálculo pH
    int valorMedio = 0; // Variável para armazenar a soma dos valores
    for (int i = 2; i < 8; i++) { // Loop para somar os valores do meio
(removendo outliers)
        valorMedio += buf[i]; // Soma os valores
    }
    float tensao = (valorMedio * 5.0) / 1024.0 / 6; // Calcula a tensão
média
    float ph = m * tensao + b; // Calcula o pH usando a equação de
calibração
}
}

```

```

// Leitura Turbidez
voltage = 0; // Zera a voltage
for (i = 0; i < 800; i++) { // Loop para ler 800 amostras
    voltage += ((float)analogRead(SensorTurbidez) / 1024) * 5; //
Soma as leituras
}
voltage = voltage / 800; // Calcula a média da voltage
voltage = ArredondarPara(voltage, 2); // Arredonda a voltage para
2 casas decimais
if (voltage < 2.5) { // Se a voltage for menor que 2.5
    NTU = 3000; // Define o NTU como 3000 (valor máximo)
} else if (voltage > 3.81) { // Se a voltage for maior que 3.81
    NTU = 0; // Define o NTU como 0 (valor mínimo)
    voltage = 3.81; // Limita a voltage
} else { // Caso contrário
    NTU = -595.2380952383246 * square(voltage) + 3779.761904763616 *
voltage - 5760.357142860333; // Calcula o NTU usando uma equação
}
// Exibição dos resultados se não estiver em calibração
if (!modoCalibracaoAtivo) { // Se o modo de calibração não estiver
ativo
    Serial.println("\n=== LEITURAS ATUAIS ==="); // Imprime cabeçalho
    Serial.print("TDS: "); // Imprime o rótulo
    Serial.print(tdsValue, 0); // Imprime o valor do TDS
    Serial.println(" ppm"); // Imprime a unidade
    Serial.print("EC: "); // Imprime o rótulo
    Serial.print(EC); // Imprime o valor da condutividade elétrica
    Serial.print(" uS/cm"); // Imprime a unidade
    Serial.print(" em "); // Imprime o rótulo
    Serial.print(temperature, 1); // Imprime a temperatura
    Serial.println("°C"); // Imprime a unidade
    Serial.print("pH: "); // Imprime o rótulo
    Serial.println(ph, 1); // Imprime o valor do pH
    Serial.print("Turbidez: "); // Imprime o rótulo
    Serial.print(voltage); // Imprime a voltage
    Serial.print(" V, NTU: "); // Imprime o rótulo
    Serial.println(NTU); // Imprime o valor do NTU
    Serial.println("=====\n"); // Imprime rodapé
}
    delay(1000); // Delay de 1 segundo
}
// Sistema de arredondamento para Leitura
float ArredondarPara(float ValorEntrada, int CasaDecimal) { // Função
para arredondar um valor
    float multiplicador = powf(10.0f, CasaDecimal); // Calcula o
multiplicador
    ValorEntrada = roundf(ValorEntrada * multiplicador) / multiplicador;

```

```
// Arredonda o valor
    return ValorEntrada; // Retorna o valor arredondado
}
```